



## ZGOUBI USERS' GUIDE

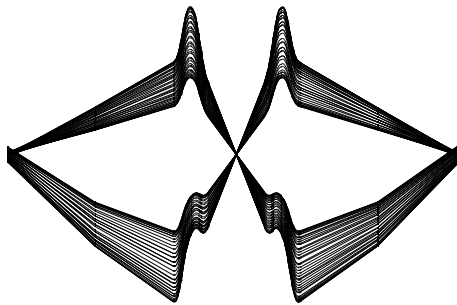
Latest version : <https://sourceforge.net/p/zgoubi/code/HEAD/tree/trunk/guide/Zgoubi.pdf>  
Zgoubi on web : <http://sourceforge.net/projects/zgoubi/>

**François Méot**

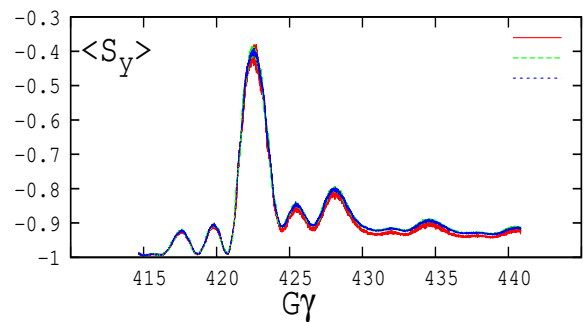
*Brookhaven National Laboratory  
Collider-Accelerator Department  
Upton, NY 11973*

22 January 2024  
(sourceforge r-2013)

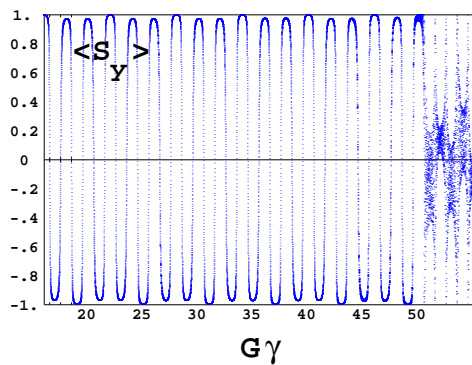
ATLAS & CMS IRs, LHC



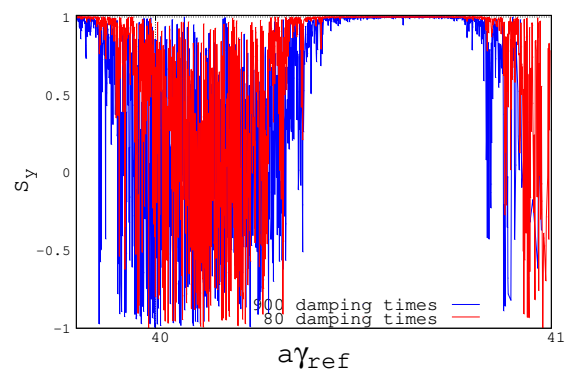
Snake Resonance Crossing in RHIC



Spin Flip During Acceleration Ramp  
AGS With Partial Snakes



An Ergodic Approach of Polarization Lifetime  
in the EIC Electron Storage Ring



## Cover figures

- Top: spin precession in a cyclotron [3],

- Bottom:

*upper left* : collision optics at ATLAS and CMS [84],

*upper right* : average bunch polarization upon crossing of  $393+Q_y$  spin resonance in RHIC [17],

*lower left* : spin-flipping at integer  $G\gamma$  during acceleration in the AGS with two partial snakes [16],

*lower right* : ergodic simulation of 18 GeV electron beam polarization lifetime at the EIC [23].

## Table of contents

<b>PART A Description of software contents</b>	<b>7</b>
<b>GLOSSARY OF KEYWORDS, PART A</b>	<b>9</b>
<b>OPTICAL ELEMENTS VERSUS KEYWORDS</b>	<b>12</b>
<b>PREFACE TO THE BNL EDITION (2015+)</b>	<b>16</b>
<b>PREFACE TO THE BNL EDITION (2012)</b>	<b>17</b>
<b>INTRODUCTION TO THE 4th EDITION (1997)</b>	<b>19</b>
<b>INTRODUCTION</b>	<b>21</b>
<b>1 NUMERICAL CALCULATION OF MOTION AND FIELDS</b>	<b>23</b>
1.1 <b>zgoubi</b> Frame . . . . .	23
1.2 Integration of the Lorentz Equation . . . . .	23
1.2.1 Integration in Magnetic Fields . . . . .	25
1.2.2 Integration in Electric Fields . . . . .	25
1.2.3 Integration in Combined Electric and Magnetic Fields . . . . .	28
1.2.4 Calculation of the Time of Flight . . . . .	28
1.3 Calculation of $\vec{E}$ and $\vec{B}$ Fields and their Derivatives . . . . .	28
1.3.1 1-D (Axial) Analytical Field Models and Extrapolation . . . . .	29
1.3.2 Extrapolation from 1-D axial field map . . . . .	29
1.3.3 Extrapolation From Median Plane Field Models . . . . .	29
1.3.4 Extrapolation from Arbitrary 2-D Field Maps . . . . .	30
1.3.5 Interpolation in 3-D Field Maps . . . . .	30
1.3.6 2-D Analytical Field Models and Extrapolation . . . . .	30
1.3.7 3-D Analytical Models of Fields . . . . .	30
1.4 Calculation of $\vec{E}$ and $\vec{B}$ from Field Maps . . . . .	31
1.4.1 1-D Axial Map, with Cylindrical Symmetry . . . . .	31
1.4.2 2-D Median Plane Map, with Median Plane Antisymmetry . . . . .	32
1.4.3 Arbitrary 2-D Map, no Symmetry . . . . .	35
1.4.4 3-D Field Map . . . . .	37
<b>2 SPIN TRACKING</b>	<b>39</b>
2.1 Introduction . . . . .	39
2.2 Integration in Magnetic Fields . . . . .	40
2.3 Integration in Electric Fields . . . . .	41
2.4 Integration in Combined Electric and Magnetic Fields . . . . .	43
2.5 Radiative polarization . . . . .	43
2.6 Spin Diffusion . . . . .	43
Diffusion Integrals . . . . .	44
<b>3 SYNCHROTRON RADIATION</b>	<b>45</b>
3.1 Energy Loss and Dynamical Effects . . . . .	45
Synchrotron Radiation Integrals . . . . .	47
3.2 Spectral-Angular Radiated Densities . . . . .	48
3.2.1 Calculation of the Radiated Electric Field . . . . .	48
3.2.2 Calculation of the Fourier Transform of the Electric Field . . . . .	51
<b>4 SPACE CHARGE</b>	<b>53</b>
<b>5 ORBIT CORRECTION</b>	<b>55</b>

<b>6</b>	<b>WHAT KEYWORDS DO, AND OTHER AVAILABLE PROCEDURES</b>	<b>57</b>
6.1	Introduction . . . . .	57
6.2	Definition of an Object . . . . .	59
	MCOBJET : Monte-Carlo generation of a 6-D object . . . . .	60
	OBJET : Generation of an object . . . . .	63
	OBJETA : Object from Monte-Carlo simulation of decay reaction . . . . .	67
6.3	Optical Elements and Related Numerical Procedures . . . . .	69
	AGSQUAD : AGS quadrupole . . . . .	70
	AIMANT : Generation of dipole mid-plane 2-D map, polar frame . . . . .	71
	AUTOREF : Transport beam into a new reference frame . . . . .	76
	BEAMBEAM : Beam-beam lens . . . . .	77
	BEND : Bending magnet, Cartesian frame . . . . .	78
	BETATRON : Betatron core (momentum kick) . . . . .	80
	BREVOL : 1-D uniform mesh magnetic field map . . . . .	81
	CARTEMES : 2-D Cartesian uniform mesh magnetic field map . . . . .	82
	CAVITE : Accelerating cavity . . . . .	84
	CHAMBR : Long transverse aperture limitation . . . . .	91
	CHANGREF : Transformation to a new reference frame . . . . .	92
	CIBLE or TARGET : Generate a secondary beam following target interaction . . . . .	94
	COLLIMA : Collimator . . . . .	95
	CYCLOTRON : Spiral setor cyclotron . . . . .	96
	DECAPOLE : Decapole magnet . . . . .	98
	DIPOLE : Dipole magnet, polar frame . . . . .	99
	DIPOLE-M : Generation of dipole mid-plane 2-D map, polar frame . . . . .	102
	DIPOLES : Dipole magnet $N$ -tuple, polar frame . . . . .	104
	EBMULT : Electro-magnetic multipole . . . . .	110
	FFAG : FFAG magnet, $N$ -tuple . . . . .	119
	OCTUPOLE : Octupole magnet . . . . .	131
	QUADRUPO : Quadrupole magnet . . . . .	136
	SOLENOID : Solenoid . . . . .	140
	TOSCA : 2-D and 3-D Cartesian or cylindrical mesh field map . . . . .	143
	YMY : Reverse signs of $Y$ and $Z$ reference axes . . . . .	154
6.4	Functional Options . . . . .	155
	BINARY : $BINARY \leftrightarrow FORMATTED$ data converter . . . . .	156
	END or FIN : End of input data list . . . . .	157
	ERRORS : Injecting errors in optical elements . . . . .	158
	FINISH : End of execution . . . . .	159
	FIT, FIT2 : Fitting procedure . . . . .	160
	GASCAT : Gas scattering . . . . .	169
	GETFITVAL : Get values of <i>variables</i> as saved from former FIT[2] run . . . . .	170
	GOTO : Branching statement . . . . .	171
	INCLUDE : File include statement . . . . .	172
	MCDESINT : Monte-Carlo simulation of in-flight decay . . . . .	173
	OPTICS : Write out optical functions. Log to zgoubi.OPTICS.out . . . . .	175
	OPTIONS : Global or special options . . . . .	176
	ORDRE : Taylor expansions order . . . . .	178
	PARTICUL : Particle characteristics . . . . .	179
	REBELOTE : 'Do it again' . . . . .	180
	RESET : Reset counters and flags . . . . .	182
	SCALING : Power supplies and R.F. function generator . . . . .	183
	SPACECHARG : Space charge correction kick . . . . .	187
	SPNTRK : Spin tracking switch, and spin coordinate initialization . . . . .	188
	SRLOSS : Synchrotron radiation energy loss . . . . .	190
	SVDOC : SVD orbit correction . . . . .	191
	SYNRAD : Synchrotron radiation spectral-angular densities . . . . .	193
	SYSTEM : System call . . . . .	194
6.5	Output Procedures . . . . .	195
	FAISCEAU, FAISCNL, FAISTORE : Print/Store particle coordinates . . . . .	196
	FOCALE, IMAGE[S] : Particle coordinates and beam size ; localization and size of horizontal waist . . . . .	198
	FOCALEZ, IMAGE[S]Z : Particle coordinates and beam size ; localization and size of vertical waist . . . . .	199
	HISTO : 1-D histogram . . . . .	200

IMAGE[S][Z] : Localization and size of vertical waists . . . . .	201
MATRIX : Calculation of transfer coefficients, periodic parameters . . . . .	202
PICKUPS : Bunch centroid path; orbit . . . . .	205
PLOTDATA : Intermediate output for the PLOTDATA graphic software . . . . .	206
SPNPRNL, SPNSTORE : Print/Store spin coordinates . . . . .	207
SPNPRT : Print spin coordinates . . . . .	208
SRPRNT : Print SR loss statistics . . . . .	209
TWISS : Compute periodic optical parameters. Log to zgoubi.TWISS.out . . . . .	210
SR Integrals, DK Spin Integrals . . . . .	210
<b>7 COMPLEMENTS REGARDING VARIOUS FUNCTIONALITIES</b>	<b>212</b>
7.1 Units in <b>zgoubi</b> . . . . .	212
7.2 Reference Rigidity . . . . .	213
7.3 Reference Trajectory, Closed Orbit . . . . .	213
7.4 Time Varying Fields . . . . .	213
7.5 Backward Ray-Tracing . . . . .	213
7.6 Checking Trajectories and Fields Across Optical Elements - the <i>IL</i> , <i>.plt</i> and <i>IC</i> flags . . . . .	213
7.7 Labeling Keywords: <i>LABEL1</i> , <i>LABEL2</i> . . . . .	215
7.8 Multi-turn Tracking in Circular Machines . . . . .	215
7.9 Multi-turn Tracking in Circular Machines . . . . .	215
7.10 <i>KPOS</i> : Positioning, (Mis-)Alignment, of Optical Elements and Field Maps . . . . .	216
7.11 Coded Integration Step . . . . .	219
7.12 Ray-tracing of an Arbitrarily Large Number of Particles . . . . .	219
7.13 Stopped Particles : The <i>IEX</i> Flag . . . . .	220
7.14 Negative Rigidity . . . . .	220
7.15 Fiel maps . . . . .	220
<b>8 ZGOUBI OUTPUT FILES</b>	<b>221</b>
8.1 <i>zgoubi.res</i> . . . . .	222
8.2 <i>zgoubi.fai</i> . . . . .	223
8.3 <i>zgoubi.plt</i> . . . . .	224
8.4 <i>zgoubi.TWISS.out</i> . . . . .	226
<b>PART B Keywords and input data formatting</b>	<b>227</b>
<b>GLOSSARY OF KEYWORDS, PART B</b>	<b>229</b>
<b>OPTICAL ELEMENTS VERSUS KEYWORDS</b>	<b>232</b>
<b>INTRODUCTION</b>	<b>235</b>
<b>PART C Examples of input data files and output result files</b>	<b>359</b>
<b>INTRODUCTION</b>	<b>361</b>
<b>1 MONTE CARLO IMAGES IN SPES 2</b>	<b>363</b>
<b>2 TRANSFER MATRICES ALONG A TWO-STAGE SEPARATION KAON BEAM LINE</b>	<b>366</b>
<b>3 IN-FLIGHT DECAY IN SPES 3</b>	<b>369</b>
<b>4 USE OF THE FITTING PROCEDURE</b>	<b>372</b>
<b>5 MULTITURN SPIN TRACKING IN SATURNE 3 GeV SYNCHROTRON</b>	<b>374</b>
<b>6 MICRO-BEAM FOCUSING WITH <math>\vec{E} \times \vec{B}</math> QUADRUPOLES</b>	<b>376</b>
<b>PART D Running zgoubi and its post-processor/graphic interface zpop</b>	<b>381</b>
<b>INTRODUCTION</b>	<b>383</b>

<b>1</b>	<b>GETTING TO RUN zgoubi AND zpop</b>	<b>383</b>
1.1	Making the Executable Files <b>zgoubi</b> and <b>zpop</b> . . . . .	383
1.1.1	The transportable package <b>zgoubi</b> . . . . .	383
1.1.2	The post-processor and graphic interface package <b>zpop</b> . . . . .	383
1.2	Running <b>zgoubi</b> . . . . .	383
1.3	Running <b>zpop</b> . . . . .	383
<b>2</b>	<b>zgoubi OUTPUT FILES</b>	<b>383</b>
	<b>REFERENCES</b>	<b>387</b>
	<b>INDEX</b>	<b>391</b>

# **PART A**

## **Description of software contents**





## Glossary of Keywords, Part A (page 229, PART B)

---

### Available keywords and where they are to be found in Part A

<b>AGSMM</b>	AGS main magnet .....	69
<b>AGSQAD</b>	AGS quadrupole .....	70
<b>AIMANT</b>	Generation of dipole mid-plane 2-D map, polar frame .....	71
<b>AUTOREF</b>	Transport beam into a new reference frame .....	76
<b>BEAMBEAM</b>	Beam-beam lens .....	77
<b>BEND</b>	Bending magnet, Cartesian frame .....	78
<b>BETATRON</b>	Betatron core (momentum kick) .....	80
<b>BINARY</b>	<i>BINARY</i> ↔ <i>FORMATTED</i> data converter .....	156
<b>BREVOL</b>	1-D uniform mesh magnetic field map .....	81
<b>CARTEMES</b>	2-D Cartesian uniform mesh magnetic field map .....	82
<b>CAVITE</b>	Accelerating cavity .....	84
<b>CHAMBR</b>	Long transverse aperture limitation .....	91
<b>CHANGREF</b>	Transformation to a new reference frame .....	92
<b>CIBLE</b>	Generate a secondary beam following target interaction .....	94
<b>COLLIMA</b>	Collimator .....	95
<b>CYCLOTRON</b>	Spiral setor cyclotron .....	96
<b>DECAPOLE</b>	Decapole magnet .....	98
<b>DIPOLE</b>	Dipole magnet, polar frame .....	99
<b>DIPOLE-M</b>	Generation of dipole mid-plane 2-D map, polar frame .....	102
<b>DIPOLES</b>	Dipole magnet <i>N</i> -tuple, polar frame .....	104
<b>DODECAPO</b>	Dodecapole magnet .....	108
<b>DRIFT</b>	Field free drift space .....	109
<b>EBMULT</b>	Electro-magnetic multipole .....	110
<b>EL2TUB</b>	Two-tube electrostatic lens .....	111
<b>ELCYLDEF</b>	Electrostatic cylindrical deflector .....	112
<b>ELMIR</b>	Electrostatic N-electrode mirror/lens, straight slits .....	113
<b>ELMIRC</b>	Electrostatic N-electrode mirror/lens, circular slits .....	114
<b>ELMULT</b>	Electric multipole .....	115
<b>ELREVOL</b>	1-D uniform mesh electric field map .....	117
<b>EMMA</b>	2-D Cartesian or cylindrical mesh field map for EMMA FFAG .....	118
<b>END</b>	End of input data list .....	157
<b>ERRORS</b>	Injecting errors in optical elements .....	158
<b>ESL</b>	Field free drift space .....	109
<b>FAISCEAU</b>	Print particle coordinates .....	196
<b>FAISCNL</b>	Store particle coordinates in file FNAME .....	196
<b>FAISTORE</b>	Store coordinates every <i>IP</i> other pass at labeled elements .....	196
<b>FINISH</b>	End of execution .....	159
<b>FFAG</b>	FFAG magnet, <i>N</i> -tuple .....	119
<b>FFAG-SPI</b>	Spiral FFAG magnet, <i>N</i> -tuple .....	122
<b>FIN</b>	End of input data list .....	157
<b>FIT</b>	Fitting procedure .....	160
<b>FOCALE</b>	Particle coordinates and horizontal beam size at distance <i>XL</i> .....	198
<b>FOCALEZ</b>	Particle coordinates and vertical beam size at distance <i>XL</i> .....	199
<b>GASCAT</b>	Gas scattering .....	169
<b>GETFITVAL</b>	Get values of <i>variables</i> as saved from former FIT[2] run .....	170

<b>GOTO</b>	Branching statement .....	171
<b>HALBACH</b>	Halbach style permanent magnet .....	125
<b>HELIX</b>	Helical dipole .....	126
<b>HISTO</b>	1-D histogram .....	200
<b>IMAGE</b>	Localization and size of horizontal waist .....	198
<b>IMAGES</b>	Localization and size of horizontal waists .....	198
<b>IMAGESZ</b>	Localization and size of vertical waists .....	199
<b>IMAGEZ</b>	Localization and size of vertical waist .....	199
<b>INCLUDE</b>	File include statement .....	172
<b>MAP2D</b>	2-D Cartesian uniform mesh field map - arbitrary magnetic field .....	127
<b>MAP2D-E</b>	2-D Cartesian uniform mesh field map - arbitrary electric field .....	128
<b>MARKER</b>	Marker .....	129
<b>MATRIX</b>	Calculation of transfer coefficients, periodic parameters .....	202
<b>MCDESINT</b>	Monte-Carlo simulation of in-flight decay .....	173
<b>MCOBJET</b>	Monte-Carlo generation of a 6-D object .....	60
<b>MULTIPOL</b>	Magnetic multipole .....	130
<b>OBJET</b>	Generation of an object .....	63
<b>OBJETA</b>	Object from Monte-Carlo simulation of decay reaction .....	67
<b>OCTUPOLE</b>	Octupole magnet .....	131
<b>OPTICS</b>	Write out optical functions. Log to zgoubi.OPTICS.out .....	175
<b>OPTIONS</b>	Global or special options .....	176
<b>ORDRE</b>	Taylor expansions order .....	178
<b>PARTICUL</b>	Particle characteristics .....	179
<b>PICKUPS</b>	Bunch centroid path; orbit .....	205
<b>PLOTDATA</b>	Intermediate output for the PLOTDATA graphic software .....	206
<b>POISSON</b>	Read magnetic field data from <i>POISSON</i> output .....	132
<b>POLARMES</b>	2-D polar mesh magnetic field map .....	133
<b>PS170</b>	Simulation of a round shape dipole magnet .....	134
<b>QUADISEX</b>	Sharp edge magnetic multipoles .....	135
<b>QUADRUPO</b>	Quadrupole magnet .....	136
<b>REBELOTE</b>	'Do it again' .....	180
<b>RESET</b>	Reset counters and flags .....	182
<b>SCALING</b>	Power supplies and R.F. function generator .....	183
<b>SEPARA</b>	Wien Filter - analytical simulation .....	138
<b>SEXQUAD</b>	Sharp edge magnetic multipole .....	135
<b>SEXTUPOL</b>	Sextupole magnet .....	139
<b>SOLENOID</b>	Solenoid .....	140
<b>SPACECHARG</b>	Space charge correction kick .....	187
<b>SPINR</b>	Spin rotation .....	142
<b>SPNPRNL</b>	Store spin coordinates into file FNAME .....	207
<b>SPNPRT</b>	Print spin coordinates .....	208
<b>SPNSTORE</b>	Store spin coordinates every <i>IP</i> other pass at labeled elements .....	207
<b>SPNTRK</b>	Spin tracking switch, and spin coordinate initialization .....	188
<b>SRLOSS</b>	Synchrotron radiation energy loss .....	190
<b>SRPRNT</b>	Print SR loss statistics .....	209
<b>SVDOC</b>	SVD orbit correction .....	191
<b>SYNRAD</b>	Synchrotron radiation spectral-angular densities .....	193
<b>SYSTEM</b>	System call .....	194
<b>TARGET</b>	Generate a secondary beam following target interaction .....	94
<b>TOSCA</b>	2-D and 3-D Cartesian or cylindrical mesh field map .....	143
<b>TRANSMAT</b>	Matrix transfer .....	147
<b>TRAROT</b>	Translation-Rotation of the reference frame .....	148
<b>TWISS</b>	Compute periodic optical parameters. Log to zgoubi.TWISS.out .....	210

<b>UNDULATOR</b>	Undulator magnet .....	149
<b>UNIPOT</b>	Unipotential cylindrical electrostatic lens .....	150
<b>VENUS</b>	Simulation of a rectangular shape dipole magnet .....	151
<b>VFFA</b>	Simulation of a VFFA dipole, or series of .....	152
<b>WIENFILT</b>	Wien filter .....	153
<b>YMY</b>	Reverse signs of $Y$ and $Z$ reference axes .....	154

## Optical Elements Versus Keywords

---

### Which optical component can be simulated Which keyword(s) can be used for that purpose

---

This glossary gives a list of keywords suitable for the simulation of common optical elements. These are classified in three categories: magnetic, electric and combined electro-magnetic elements.

Field map procedures are also listed; they provide a means for ray-tracing through measured or simulated electric and/or magnetic fields.

#### MAGNETIC ELEMENTS

AGS main magnet	AGSMM
Betatron core	BETATRON, CAVITE
Cyclotron magnet	CYCLOTRON, DIPOLE[S], DIPOLE-M, FFAG, FFAG-SPI
Decapole	DECAPOLE, MULTIPOL
Dipole[s], spectrometer dipole	AIMANT, BEND, DIPOLE[S][-M], MULTIPOL, QUADISEX
Dodecapole	DODECAPO, MULTIPOL
FFAG magnet	DIPOLAS, FFAG, FFAG-SPI, MULTIPOL
Helical dipole	HELIX
Multipole	MULTIPOL, QUADISEX, SEXQUAD
Octupole	OCTUPOLE, MULTIPOL, QUADISEX, SEXQUAD
Quadrupole	QUADRUPO, MULTIPOL, SEXQUAD, AGSQUAD
Sextupole	SEXTUPOL, MULTIPOL, QUADISEX, SEXQUAD
Skew multipoles	MULTIPOL
Solenoid	SOLENOID
Spectrometer dipole	DIPOLE, DIPOLE-M, DIPOLES
Undulator	UNDULATOR

#### Using field maps

1-D, cylindrical symmetry	BREVOL
2-D, mid-plane symmetry	CARTEMES, POISSON, TOSCA
2-D, no symmetry	MAP2D
2-D, polar mesh, mid-plane symmetry	POLARMES
3-D, no symmetry	TOSCA
EMMA FFAG quadrupole doublet	EMMA
linear composition of field maps	TOSCA

#### ELECTRIC ELEMENTS

2-tube (bipotential) lens	EL2TUB
3-tube (unipotential) lens	UNIPOT
Decapole	ELMULT
Dipole	ELMULT
Dodecapole	ELMULT
Multipole	ELMULT
N-electrode mirror/lens, straight slits	ELMIR
N-electrode mirror/lens, circular slits	ELMIRC
Octupole	ELMULT

Quadrupole	ELMULT
R.F. (kick) cavity	CAVITE
Sextupole	ELMULT
Skew multipoles	ELMULT

### Using field maps

1-D, cylindrical symmetry	ELREVOL
2-D, no symmetry	MAP2D-E

### ELECTRO-MAGNETIC ELEMENTS

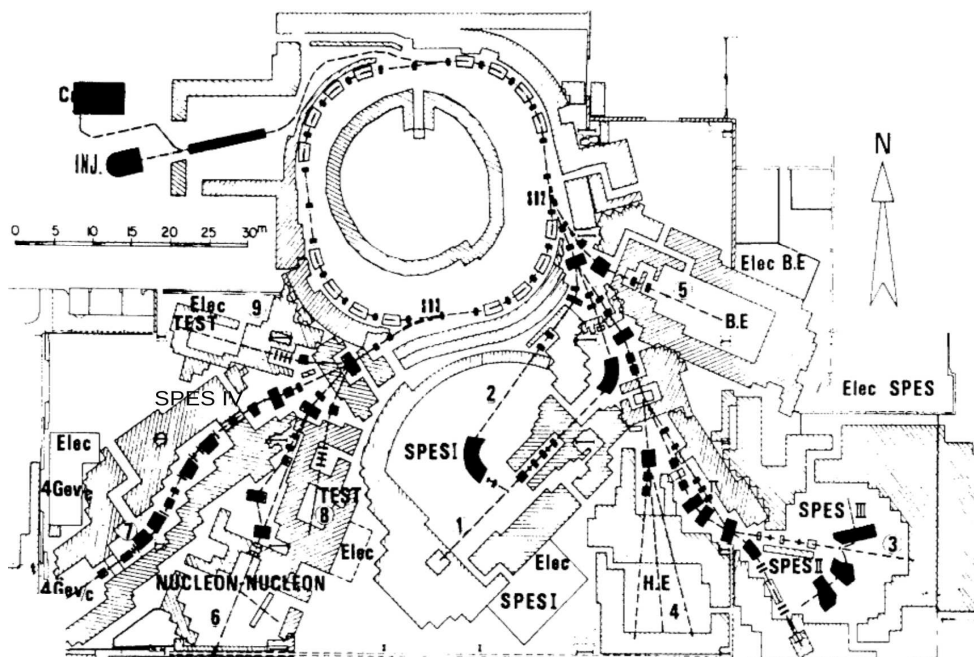
Decapole	EBMULT
Dipole	EBMULT
Dodecapole	EBMULT
Multipole	EBMULT
Octupole	EBMULT
Quadrupole	EBMULT
Sextupole	EBMULT
Skew multipoles	EBMULT
Wien filter	SEPARA, WIENFILT



## OVER A HALF-CENTURY OF RAYTRACING

“The history of SATURNE 1 and SATURNE 2 accelerators includes the construction of 4 septrometers as particle detectors. The first, SPES I designed in 1968 is of high resolution. The resolution permits the selection of one nuclear level which is necessary even for elastic scattering. A program, ZGOUBI (Jean Claude Faivre, Denis Garreta) for precise orbit computing had to be written and was the first one to be fast enough to permit new spectrometer design.”<sup>1</sup>

The initial version of **zgoubi**, dedicated to ray-tracing in magnetic fields, was written by D. Garreta and J.C. Faivre at the Laboratoire National SATURNE, CEN-Saclay in the late 1960s-early 1970s [1, 2]. The program was aimed at the design of SATURNE 2 spectrometers SPES I and SPES II, and further served for the study, design and operation of the four SATURNE 2 spectrometers, SPES I, II, III and IV. It has been in use since, in many nuclear and particle physics national and foreign laboratories.



*SATURNE 2 synchrotron and its experimental areas, including mass spectrometers SPES I to SPES IV, a typical nuclear physics accelerator facility. The polarized ion sources Dioné and Hypérian are at the top left, followed by a 20 MeV linac. In the early 1980s a synchrotron booster, MIMAS, was added for higher polarized ion performance*

<sup>1</sup>Jacques Thirion, Preface, in “The 20 years of the synchrotron Saturne-2”. A. Boudard, P.-A. Chamouard Eds. World Scientific, May 1998.

**PREFACE TO THE BNL EDITION, 2015+**

New functionalities have been introduced :

- *GOTO* keyword, a conditional goto, useful for RLA simulations,
- *INCLUDE* keyword, working like the Fortran, similar as well to latex's, "include", more or less equivalent to a "call file" as found in various optics codes, useful for long or repetitive sequences. The developed optical sequence is printed out, in the first place, in zgoubi.res. The latter can be run as is, as the '*END*' or '*FIN*' statement which concludes an input sequence causes the execution to ignore whatever follows in that zgoubi.res listing.



## PREFACE TO THE BNL EDITION, 2012

The previous release of the Zgoubi Users' Guide as a Lab. report dates from 1997, making the present one the latest in a series of five [4]-[7].

**zgoubi** has undergone substantial developments since the 4th edition of the Users' Guide, in the frame of a number of projects and of beam dynamics studies, as the Neutrino Factory, lepton and hadron colliders, spin dynamics investigations at SuperB, RHIC, etc. As well the list of optical elements has grown, and so did the "Glossary of Keywords" list, pp. 9 and 229, including new simulation and computing procedures, which range from constraint-matching to overlapping magnetic fields capabilities to spin manipulations and other synchrotron radiation damping recipes.

The output files `zgoubi.fai` (local particle data) and `zgoubi.plt` (step-by-step-particle data across an optical element) have been reformatted in columns, making them readily readable by graphic programs as "gnuplot". Ancillary output files, less voluminous than the latter two and useful for producing plots or for `zgoubi` functioning checks/debugging, with names of the form "`zgoubi.*.out`" are available from various keywords, usually by specifying "PRINT" in the keyword's arguments, for instance in the case of "CAVITE" it will produce "`zgoubi.CAVITE.Out`" containing time dependence of RF parameters as experienced by the particles at traversal of the cavity, whereas "`zgoubi.OPTICS.out`" can be obtained from *OPTICS* keyword.

This evolution of **zgoubi** allows embedding "*FIT[2]*" procedure within "*REBELOTE*" do-loop, whereas the latter has been conferred the capability of changing values of any numerical data across `zgoubi.dat`. This allows parameter scans, while possibly using "*SYSTEM*" to store the outcomes in dedicated external files.

A series of auxiliary computing tools and graphic tools (the latter based on "gnuplot") have been developed, aimed at making the designer's life easier, as, search for closed orbits in periodic machines, computation of optical functions and parameters, tune scans, dynamic aperture scans, spin dynamics data treatment, graphic scripts, etc., and including dedicated ones regarding, *e.g.*, FFAG and cyclotron design, AGS and RHIC studies, synchrotron radiation energy losses and their effects. In addition "python" interfaces are being developed by several users, possibly made available on web by their authors.

The code has been installed on SourceForge with the collaboration of J. S. Berg, in Sept. 2007, there it is fully and freely available [8]. The SourceForge package evolves and is maintained at the pace of on-going projects and design studies. It includes the source files, the post-processor program **zpop**, as well as many examples with their input data files ("`zgoubi.dat`") and output result files ("`zgoubi.res`") and the auxiliary toolbox.

The Users' Guide is intended to describe the contents of the most recent version of **zgoubi**. The code and its Guide are both far from being "finished products".

J. S. Berg and N. Tsoupas, BNL, have reviewed the final proof of the present document.



## INTRODUCTION TO THE 4th EDITION, 1997

The initial version of **zgoubi**, dedicated to ray-tracing in magnetic fields, was developed by D. Garreta and J.C. Faivre at CEN-Saclay in the early 1970s. It was perfected for the purpose of studying the four spectrometers SPES I, II, III, IV at the Laboratoire National SATURNE (CEA-Saclay, France), and SPEG at GANIL (Caen, France). It is being used since long in several national and foreign laboratories.

The first manual was in French [4]. Accounting for many developments and improvements, and in order to facilitate access to the program an English version of the manual was written at TRIUMF with the assistance of J. Doornbos. P. Stewart prepared the manuscript for publication [5]

An updating of the latter was necessary for accompanying the third version of the code which included developments regarding spin tracking and ray-tracing in combined electric and magnetic fields ; this was done with the help of D. Bunel (SATURNE Laboratory, Saclay) for the preparation of the document and lead to the third release [6].

In the mid-1990s, the computation of synchrotron radiation electromagnetic impulse and spectra was introduced for the purpose of studying interference effects at the LEP synchrotron radiation based diagnostic mini-wiggler. In the mean time, several new optical elements were added, such as electro-magnetic and other electrostatic lenses. Used since several years for special studies in periodic machines (*e.g.*, SATURNE at Saclay, COSY at Julich, LEP and LHC at CERN, Neutrino Factory rings), **zgoubi** has also undergone extensive developments regarding storage ring related features.

These developments of **zgoubi** have strongly benefited of the environment of the Groupe Théorie, Laboratoire National SATURNE, CEA/DSM-Saclay, in the years 1985-1995.

The graphic interface to **zgoubi** (addressed in Part D) has also undergone concomitant extensive developments, which make it a performing tool for the post-processing of **zgoubi** outputs.

The Users' Guide is intended to describe the contents of the most recent version of **zgoubi**, which is far from being a "finished product".



## INTRODUCTION

The computer code **zgoubi** calculates trajectories of charged particles in magnetic and electric fields. At the origin specially adapted to the definition and adjustment of beam lines and magnetic spectrometers, it has so evolved that it allows the study of systems including complex sequences of optical elements such as dipoles, quadrupoles, arbitrary multipoles, FFAG magnets and other magnetic or electric devices, and is able as well to handle periodic structures. Compared to other codes, it presents several peculiarities, as follows - a non-exhaustive list :

- a numerical method for integrating the Lorentz equation, based on Taylor series, which optimizes computing time and provides high accuracy and strong symplecticity,
- spin tracking, using the same numerical method as for the Lorentz equation,
- account of stochastic photon emission, and its effects on particle dynamics,
- calculation of the synchrotron radiation electric field and spectra in arbitrary magnetic fields, from the ray-tracing outcomes,
- the possibility of using a mesh, which allows ray-tracing from simulated or measured (1-D, 2-D or 3-D) electric and magnetic field maps,
- a number Monte Carlo procedures : unlimited number of trajectories, in-flight decay, stochastic radiation, etc.,
- built-in fitting procedures allowing arbitrary variables and a large variety of constraints, easily expandable,
- multi-turn tracking in circular accelerators including features proper to machine parameter calculation and survey,
- simulation of time-varying power supplies,
- simulation of arbitrary radio-frequency programs.

The initial version of **zgoubi** was dedicated to ray-tracing in magnetic elements, beam lines, spectrometers. It was perfected for the purpose of studying, and operating, the four spectrometers SPES I, II, III, IV at the Laboratoire National SATURNE (CEA-Saclay, France), and, later, SPEG at GANIL (Caen, France).

Developments regarding spin tracking and ray-tracing in combined electric and magnetic fields were implemented, in the late 1980s and early 1990s respectively.

In the mid-1990s, the computation of synchrotron radiation electromagnetic impulse and spectra was introduced, for the purpose of synchrotron radiation diagnostic R&D at LEP, and further applied to the design of the Synchrotron radiation diagnostics installations at LHC in the early 2000s. In the mean time, several new optical elements were added, such as electro-magnetic and other electrostatic lenses. Used since several years for special studies in periodic machines (*e.g.*, SATURNE at Saclay, COSY at Julich, LEP and LHC at CERN), **zgoubi** has also undergone extensive developments regarding storage ring related features.

Many developments have been accomplished since the early 2000s in the frame of a number of project design and beam dynamics studies, as the neutrino factory, lepton and hadron colliders, spin studies at AGS and RHIC, etc. As a consequence the list of optical elements and the compendium of numerical methods, so-called “Glossary of Keywords” list, pp. 9 and 229, has stretched with new simulation and computing procedures, ranging from fitting (*FIT2* procedure ; additional constraints) to overlapping magnetic field capabilities (*DIPOLLES*, *FFAG*), spin manipulation (*SPINR*), optical elements (*BEAMBEAM*), radiation damping tools (*SRLOSS*) and many others (*FAISTORE*, *MAP2D-E*, *OPTICS*, *PICKUPS*, *TWISS*, etc.).

The graphic interface to **zgoubi** (**zpop**, Part D) has undergone extensive developments, making it a convenient companion tool to the use of **zgoubi**.



## 1 NUMERICAL CALCULATION OF MOTION AND FIELDS

### 1.1 zgoubi Frame

The reference frame of **zgoubi** is presented in Fig. 1. Its origin and orientation are arbitrary. However, its origin is generally in the median plane, considered to be  $(O; X, Y)$ .

In optical elements defined in a Cartesian frame, OX generally coincides with the optical axis of the element (for instance, the longitudinal axis in a multipole). In optical elements defined in a polar frame, OX is tangent to a reference arc (for instance, a circle with reference radius RM in a bending magnet) and the  $(O; X, Y)$  frame moves with the particle (located in the  $(Y, Z)$  plane) across the optical element.

### 1.2 Integration of the Lorentz Equation

The Lorentz equation, which governs the motion of a particle of charge  $q$ , relativistic mass  $m$  and velocity  $\vec{v}$  in electric and magnetic fields  $\vec{e}$  and  $\vec{b}$ , is written

$$\frac{d(m\vec{v})}{dt} = q(\vec{e} + \vec{v} \times \vec{b}) \quad (1.2.1)$$

Noting  $()' = \frac{d()}{ds}$  and taking

$$\vec{u} = \frac{\vec{v}}{v}, \quad ds = v dt, \quad \vec{u}' = \frac{d\vec{u}}{ds}, \quad m\vec{v} = mv\vec{u} = q B\rho \vec{u} \quad (1.2.2)$$

where  $B\rho$  is the rigidity of the particle, this equation can be rewritten

$$(B\rho)' \vec{u} + B\rho \vec{u}' = \frac{\vec{e}}{v} + \vec{u} \times \vec{b} \quad (1.2.3)$$

From position  $\vec{R}(M_0)$  and unit velocity  $\vec{u}(M_0)$  at point  $M_0$ , position  $\vec{R}(M_1)$  and unit velocity  $\vec{u}(M_1)$  at point  $M_1$  following a displacement  $\Delta s$ , are obtained from truncated Taylor expansions (Fig. 2)

$$\begin{aligned} \vec{R}(M_1) &\approx \vec{R}(M_0) + \vec{u}(M_0) \Delta s + \vec{u}'(M_0) \frac{\Delta s^2}{2!} + \dots + \vec{u}''''(M_0) \frac{\Delta s^6}{6!} \\ \vec{u}(M_1) &\approx \vec{u}(M_0) + \vec{u}'(M_0) \Delta s + \vec{u}''(M_0) \frac{\Delta s^2}{2!} + \dots + \vec{u}''''(M_0) \frac{\Delta s^5}{5!} \end{aligned} \quad (1.2.4)$$

The rigidity at  $M_1$ , in case it varies (in electric fields), is obtained in a similar manner using

$$(B\rho)(M_1) \approx (B\rho)(M_0) + (B\rho)'(M_0) \Delta s + \dots + (B\rho)''''(M_0) \frac{\Delta s^5}{5!} \quad (1.2.5)$$

The equation of time of flight is written in a similar manner

$$T(M_1) \approx T(M_0) + T'(M_0) \Delta s + T''(M_0) \frac{\Delta s^2}{2} + \dots + T''''(M_0) \frac{\Delta s^5}{5!} \quad (1.2.6)$$

The derivatives  $\vec{u}^{(n)} = \frac{d^n \vec{u}}{ds^n}$  and  $(B\rho)^{(n)} = \frac{d^n (B\rho)}{ds^n}$  involved in these expressions are calculated as described in the next sections. For the sake of computing speed, three distinct software procedures are involved, depending on whether  $\vec{e}$  or  $\vec{b}$  is zero, or  $\vec{e}$  and  $\vec{b}$  are both non-zero.

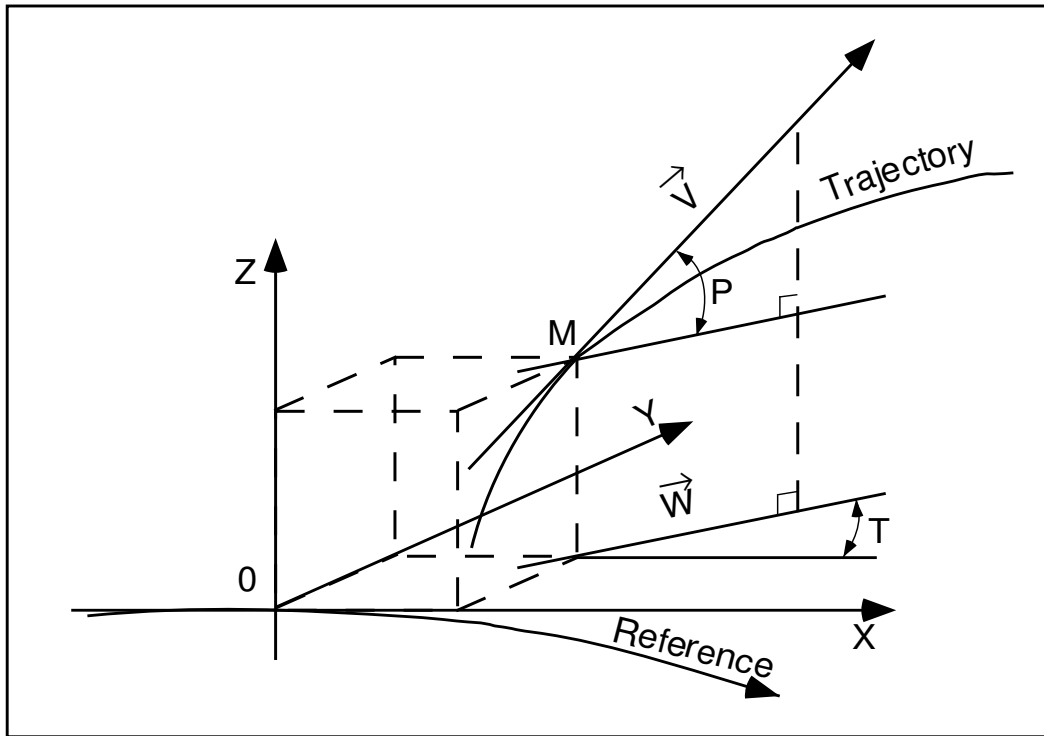


Figure 1: Reference frame and coordinates  $(Y, T, Z, P)$  in **zgoubi**.

Reference curve: a straight axis in optical elements defined in a Cartesian frame; an arc of a circle in optical elements defined in a cylindrical frame.

$OX$  : in the direction of motion, tangent to the Reference curve,

$OY$  : normal to  $OX$ ,

$OZ$  : orthogonal to the  $(X, Y)$  plane,

$\vec{W}$  : projection of the velocity,  $\vec{v}$ , in the  $(X, Y)$  plane,

$T$  : in the  $(X, Y)$  plane, angle between  $\vec{W}$  and the  $X$ -axis,

$P$  : in the  $(\vec{W}, Z)$  plane, angle between  $\vec{W}$  and  $\vec{v}$ .

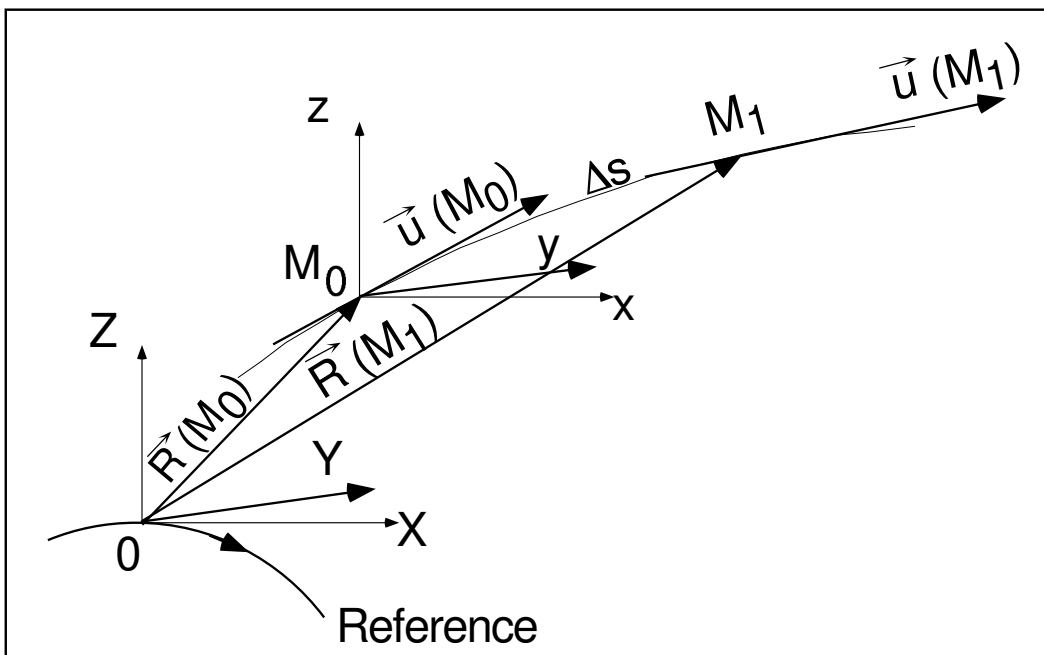


Figure 2: Position and velocity of a particle in the reference frame.



### 1.2.1 Integration in Magnetic Fields

Considering  $\vec{e} = 0$  and given  $\vec{B} = \frac{\vec{b}}{B\rho}$ , eq. (1.2.3) reduces to

$$\vec{u}' = \vec{u} \times \vec{B} \quad (1.2.7)$$

The successive derivatives  $\vec{u}^{(n)} = \frac{d^n \vec{u}}{ds^n}$  of  $\vec{u}$  needed in the Taylor expansions (eqs. 1.2.4) are calculated by differentiating  $\vec{u}' = \vec{u} \times \vec{B}$

$$\begin{aligned} \vec{u}'' &= \vec{u}' \times \vec{B} + \vec{u} \times \vec{B}' \\ \vec{u}''' &= \vec{u}'' \times \vec{B} + 2\vec{u}' \times \vec{B}' + \vec{u} \times \vec{B}'' \\ \vec{u}'''' &= \vec{u}''' \times \vec{B} + 3\vec{u}'' \times \vec{B}' + 3\vec{u}' \times \vec{B}'' + \vec{u} \times \vec{B}''' \\ \vec{u}''''' &= \vec{u}'''' \times \vec{B} + 4\vec{u}''' \times \vec{B}' + 6\vec{u}'' \times \vec{B}'' + 4\vec{u}' \times \vec{B}''' + \vec{u} \times \vec{B}'''' \end{aligned} \quad (1.2.8)$$

where  $\vec{B}^{(n)} = \frac{d^n \vec{B}}{ds^n}$ . From

$$d\vec{B} = \frac{\partial \vec{B}}{\partial X} dX + \frac{\partial \vec{B}}{\partial Y} dY + \frac{\partial \vec{B}}{\partial Z} dZ = \sum_{i=1,3} \frac{\partial \vec{B}}{\partial X_i} dX_i \quad (1.2.9)$$

and by successive differentiation, we get

$$\begin{aligned} \vec{B}' &= \sum_i \frac{\partial \vec{B}}{\partial X_i} u_i \\ \vec{B}'' &= \sum_{ij} \frac{\partial^2 \vec{B}}{\partial X_i \partial X_j} u_i u_j + \sum_i \frac{\partial \vec{B}}{\partial X_i} u_i' \\ \vec{B}''' &= \sum_{ijk} \frac{\partial^3 \vec{B}}{\partial X_i \partial X_j \partial X_k} u_i u_j u_k + 3 \sum_{ij} \frac{\partial^2 \vec{B}}{\partial X_i \partial X_j} u_i' u_j + \sum_i \frac{\partial \vec{B}}{\partial X_i} u_i'' \\ \vec{B}'''' &= \sum_{ijkl} \frac{\partial^4 \vec{B}}{\partial X_i \partial X_j \partial X_k \partial X_l} u_i u_j u_k u_l + 6 \sum_{ijk} \frac{\partial^3 \vec{B}}{\partial X_i \partial X_j \partial X_k} u_i' u_j u_k \\ &\quad + 4 \sum_{ij} \frac{\partial^2 \vec{B}}{\partial X_i \partial X_j} u_i'' u_j + 3 \sum_{ij} \frac{\partial^2 \vec{B}}{\partial X_i \partial X_j} u_i' u_j' + \sum_i \frac{\partial \vec{B}}{\partial X_i} u_i''' \end{aligned} \quad (1.2.10)$$

From the knowledge of  $\vec{u}(M_0)$  and  $\vec{B}(M_0)$  at point  $M_0$  of the trajectory, we calculate alternately the derivatives of  $\vec{u}(M_0)$  and  $\vec{B}(M_0)$ , by means of eqs. (1.2.8) and (1.2.10), and inject these into eq. (1.2.4) so yielding  $\vec{R}(M_1)$  and  $\vec{u}(M_1)$ .

### 1.2.2 Integration in Electric Fields [9]

Admitting that  $\vec{b} = 0$ , eq. (1.2.3) reduces to

$$(B\rho)' \vec{u} + B\rho \vec{u}' = \frac{\vec{e}}{v} \quad (1.2.11)$$

which, by successive differentiations, gives the recursive relations

$$\begin{aligned}
(B\rho)' \vec{u} + B\rho \vec{u}' &= \frac{\vec{e}}{v} \\
(B\rho)'' \vec{u} + 2(B\rho)' \vec{u}' + B\rho \vec{u}'' &= \left(\frac{1}{v}\right)' \vec{e} + \frac{\vec{e}'}{v} \\
(B\rho)''' \vec{u} + 3(B\rho)'' \vec{u}' + 3(B\rho)' \vec{u}'' + B\rho \vec{u}''' &= \left(\frac{1}{v}\right)'' \vec{e} + 2 \left(\frac{1}{v}\right)' \vec{e}' + \left(\frac{1}{v}\right) \vec{e}'' \\
(B\rho)'''' \vec{u} + 4(B\rho)''' \vec{u}' + 6(B\rho)'' \vec{u}'' + 4(B\rho)' \vec{u}''' + B\rho \vec{u}'''' &= \\
&\left(\frac{1}{v}\right)''' \vec{e} + 3 \left(\frac{1}{v}\right)'' \vec{e}' + 3 \left(\frac{1}{v}\right)' \vec{e}'' + \frac{1}{v} \vec{e}''' \\
(B\rho)''''' \vec{u} + 5(B\rho)'''' \vec{u}' + \text{etc.} &
\end{aligned} \tag{1.2.12}$$

that provide the derivatives  $\frac{d^n \vec{u}}{ds^n}$  needed in the Taylor expansions (eq. 1.2.4)

$$\begin{aligned}
\vec{u}' &= \left(\frac{1}{v}\right) \vec{E} - \frac{(B\rho)'}{B\rho} \vec{u} \\
\vec{u}'' &= \left(\frac{1}{v}\right)' \vec{E} + \left(\frac{1}{v}\right) \vec{E}'|_{B\rho} - 2 \frac{(B\rho)'}{B\rho} \vec{u}' - \frac{(B\rho)''}{B\rho} \vec{u} \\
\vec{u}''' &= \left(\frac{1}{v}\right)'' \vec{E} + 2 \left(\frac{1}{v}\right)' \vec{E}'|_{B\rho} + \frac{1}{v} \vec{E}''|_{B\rho} - 3 \frac{(B\rho)'}{B\rho} \vec{u}'' - 3 \frac{(B\rho)''}{B\rho} \vec{u}' - \frac{(B\rho)'''}{B\rho} \vec{u} \\
\vec{u}'''' &= \left(\frac{1}{v}\right)''' \vec{E} + 3 \left(\frac{1}{v}\right)'' \vec{E}'|_{B\rho} + 3 \left(\frac{1}{v}\right)' \vec{E}''|_{B\rho} + \left(\frac{1}{v}\right) \vec{E}'''|_{B\rho} \\
&\quad - 4 \frac{(B\rho)'}{B\rho} \vec{u}''' - 6 \frac{(B\rho)''}{B\rho} \vec{u}'' - 4 \frac{(B\rho)'''}{B\rho} \vec{u}' - \frac{(B\rho)''''}{B\rho} \vec{u} \\
\vec{u}''''' &= \left(\frac{1}{v}\right)'''' \vec{E} + 4 \left(\frac{1}{v}\right)''' \vec{E}'|_{B\rho} + 6 \left(\frac{1}{v}\right)'' \vec{E}''|_{B\rho} + 4 \left(\frac{1}{v}\right)' \vec{E}'''|_{B\rho} + \left(\frac{1}{v}\right) \vec{E}''''|_{B\rho} \\
&\quad - 5 \frac{(B\rho)'}{B\rho} \vec{u}'''' - 10 \frac{(B\rho)''}{B\rho} \vec{u}''' - 10 \frac{(B\rho)'''}{B\rho} \vec{u}'' - 5 \frac{(B\rho)''''}{B\rho} \vec{u}' - \frac{(B\rho)''''' }{B\rho} \vec{u}
\end{aligned} \tag{1.2.13}$$

where  $\vec{E} = \frac{\vec{e}}{B\rho}$ , and  $(\ )^{(n)}|_{B\rho}$  denotes differentiation at constant  $B\rho$ :  $\vec{E}^{(n)}|_{B\rho} = \frac{1}{B\rho} \frac{d^n \vec{e}}{ds^n}$ . These derivatives of the electric field are obtained from the total derivative

$$d\vec{E} = \frac{\partial \vec{E}}{\partial X} dX + \frac{\partial \vec{E}}{\partial Y} dY + \frac{\partial \vec{E}}{\partial Z} dZ \tag{1.2.14}$$

by successive differentiations

$$\begin{aligned}
\vec{E}' &= \sum_i \frac{\partial \vec{E}}{\partial X_i} u_i \\
\vec{E}'' &= \sum_{ij} \frac{\partial^2 \vec{E}}{\partial X_i \partial X_j} u_i u_j + \sum_i \frac{\partial \vec{E}}{\partial X_i} u_i' \\
\vec{E}''' &= \sum_{ijk} \frac{\partial^3 \vec{E}}{\partial X_i \partial X_j \partial X_k} u_i u_j u_k + 3 \sum_{ij} \frac{\partial^2 \vec{E}}{\partial X_i \partial X_j} u_i' u_j + \sum_i \frac{\partial \vec{E}}{\partial X_i} u_i''
\end{aligned} \tag{1.2.15}$$

etc. as in eq. 1.2.10. The eqs. (1.2.13), as well as the calculation of the rigidity, eq. (1.2.5), involve derivatives  $(B\rho)^{(n)} = \frac{d^n (B\rho)}{ds^n}$ , which are obtained in the following way. Considering that

$$\frac{dp^2}{dt} = \frac{d\vec{p}^2}{dt} \quad \text{i.e.,} \quad \frac{dp}{dt} p = \frac{d\vec{p}}{dt} \vec{p} \tag{1.2.16}$$

with  $\frac{d\vec{p}}{dt} = q(\vec{e} + \vec{v} \times \vec{b})$  (eq. 1.2.1), we obtain

$$\frac{dp}{dt} p = q(\vec{e} + v \times \vec{b}) \cdot \vec{p} = q\vec{e} \cdot \vec{p} \quad (1.2.17)$$

since  $(\vec{v} \times \vec{b}) \cdot \vec{p} = 0$ . Normalizing as previously with  $\vec{p} = p\vec{u} = qB\rho\vec{u}$  and  $ds = vdt$ , and by successive differentiations, eq. (1.2.17) leads to the  $(B\rho)^{(n)}$

$$\begin{aligned} (B\rho)' &= \frac{1}{v} (\vec{e} \cdot \vec{u}) \\ (B\rho)'' &= \left(\frac{1}{v}\right)' (\vec{e} \cdot \vec{u}) + \frac{1}{v} (\vec{e} \cdot \vec{u})' \\ (B\rho)''' &= \left(\frac{1}{v}\right)'' (\vec{e} \cdot \vec{u}) + 2 \left(\frac{1}{v}\right)' (\vec{e} \cdot \vec{u})' + \frac{1}{v} (\vec{e} \cdot \vec{u})'' \\ (B\rho)'''' &= \left(\frac{1}{v}\right)''' (\vec{e} \cdot \vec{u}) + 3 \left(\frac{1}{v}\right)'' (\vec{e} \cdot \vec{u})' + 3 \left(\frac{1}{v}\right)' (\vec{e} \cdot \vec{u})'' + \frac{1}{v} (\vec{e} \cdot \vec{u})''' \\ (B\rho)''''' &= \text{etc.} \end{aligned} \quad (1.2.18)$$

The derivatives  $(\vec{e} \cdot \vec{u})^{(n)} = \frac{d^n(\vec{e} \cdot \vec{u})}{ds^n}$  are obtained by recursive differentiation

$$\begin{aligned} (\vec{e} \cdot \vec{u})' &= \vec{e}' \cdot \vec{u} + \vec{e} \cdot \vec{u}' \\ (\vec{e} \cdot \vec{u})'' &= \vec{e}'' \cdot \vec{u} + 2\vec{e}' \cdot \vec{u}' + \vec{e} \cdot \vec{u}'' \\ &\text{etc.} \end{aligned} \quad (1.2.19)$$

Note that they can be related to the derivatives of the kinetic energy  $W$  by  $dW = \frac{d\vec{p}}{dt} \cdot \vec{v} dt = q\vec{e} \cdot \vec{v} dt$  which leads to

$$\frac{d^{n+1}W}{ds^{n+1}} = q \frac{d^n(\vec{e} \cdot \vec{u})}{ds^n} \quad (1.2.20)$$

Finally, the derivatives  $\left(\frac{1}{v}\right)^{(n)} = \frac{d^n\left(\frac{1}{v}\right)}{ds^n}$  involved in eqs. (1.2.13,1.2.18) are obtained from

$$p = \frac{vW + m_0c^2}{c}$$

( $m_0$  is the rest mass) by successive differentiations, that give the recursive relations

$$\begin{aligned} \left(\frac{1}{v}\right) &= \frac{1}{c^2} \frac{W + m_0c^2}{qB\rho} \\ \left(\frac{1}{v}\right)' &= \frac{1}{c^2} \frac{(\vec{e} \cdot \vec{u})}{B\rho} - \frac{1}{v} \frac{(B\rho)'}{B\rho} \\ \left(\frac{1}{v}\right)'' &= \frac{1}{c^2} \frac{(\vec{e} \cdot \vec{u})'}{B\rho} - 2 \left(\frac{1}{v}\right)' \frac{(B\rho)'}{B\rho} - \frac{1}{v} \frac{(B\rho)''}{B\rho} \\ \left(\frac{1}{v}\right)''' &= \frac{1}{c^2} \frac{(\vec{e} \cdot \vec{u})''}{B\rho} - 3 \left(\frac{1}{v}\right)'' \frac{(B\rho)'}{B\rho} - 3 \left(\frac{1}{v}\right)' \frac{(B\rho)''}{B\rho} - \frac{1}{v} \frac{(B\rho)'''}{B\rho} \\ \left(\frac{1}{v}\right)'''' &= \text{etc.} \end{aligned} \quad (1.2.21)$$

### 1.2.3 Integration in Combined Electric and Magnetic Fields

When both  $\vec{e}$  and  $\vec{b}$  are non-zero, the complete eq. (1.2.3) must be considered. Recursive differentiations give the following relations

$$\begin{aligned}
(B\rho)' \vec{u} + B\rho \vec{u}' &= \frac{\vec{e}}{v} + \vec{u} \times \vec{b} \\
(B\rho)'' \vec{u} + 2(B\rho)' \vec{u}' + B\rho \vec{u}'' &= \left(\frac{1}{v}\right)' \vec{e} + \left(\frac{1}{v}\right) \vec{e}' + (\vec{u} \times \vec{b})' \\
(B\rho)''' \vec{u} + 3(B\rho)'' \vec{u}' + 3(B\rho)' \vec{u}'' + B\rho \vec{u}''' &= \left(\frac{1}{v}\right)'' \vec{e} + 2\left(\frac{1}{v}\right)' \vec{e}' + \left(\frac{1}{v}\right) \vec{e}'' + (\vec{u} \times \vec{b})'' \\
(B\rho)'''' \vec{u} + 4(B\rho)''' \vec{u}' + 6(B\rho)'' \vec{u}'' + 4(B\rho)' \vec{u}''' + B\rho \vec{u}'''' &= \\
\left(\frac{1}{v}\right)''' \vec{e} + 3\left(\frac{1}{v}\right)'' \vec{e}' + 3\left(\frac{1}{v}\right)' \vec{e}'' + \frac{1}{v} \vec{e}''' + (\vec{u} \times \vec{b})''' &
\end{aligned} \tag{1.2.22}$$

that provide the derivatives  $\frac{d^n \vec{u}}{ds^n}$  needed in the Taylor expansions (1.2.4)

$$\begin{aligned}
\vec{u}' &= \left(\frac{1}{v}\right) \vec{E} + (\vec{u} \times \vec{B}) - \frac{(B\rho)'}{B\rho} \vec{u} \\
\vec{u}'' &= \left(\frac{1}{v}\right)' \vec{E} + \left(\frac{1}{v}\right) \vec{E}'|_{B\rho} + (\vec{u} \times \vec{B})'|_{B\rho} - 2\frac{(B\rho)'}{B\rho} \vec{u}' - \frac{(B\rho)''}{B\rho} \vec{u} \\
\vec{u}''' &= \left(\frac{1}{v}\right)'' \vec{E} + 2\left(\frac{1}{v}\right)' \vec{E}'|_{B\rho} + \frac{1}{v} \vec{E}''|_{B\rho} + (\vec{u} \times \vec{B})''|_{B\rho} - 3\frac{(B\rho)'}{B\rho} \vec{u}'' - 3\frac{(B\rho)''}{B\rho} \vec{u}' - \frac{(B\rho)'''}{B\rho} \vec{u} \\
\vec{u}'''' &= \left(\frac{1}{v}\right)''' \vec{E} + 3\left(\frac{1}{v}\right)'' \vec{E}'|_{B\rho} + 3\left(\frac{1}{v}\right)' \vec{E}''|_{B\rho} + \left(\frac{1}{v}\right) \vec{E}'''|_{B\rho} \\
&\quad + (\vec{u} \times \vec{B})'''|_{B\rho} - 4\frac{(B\rho)'}{B\rho} \vec{u}''' - 6\frac{(B\rho)''}{B\rho} \vec{u}'' - 4\frac{(B\rho)'''}{B\rho} \vec{u}' - \frac{(B\rho)''''}{B\rho} \vec{u}
\end{aligned} \tag{1.2.23}$$

where  $\vec{E} = \frac{\vec{e}}{B\rho}$ ,  $\vec{B} = \frac{\vec{b}}{B\rho}$ , and  $(n)|_{B\rho}$  denotes differentiation at constant  $B\rho$

$$\vec{E}^{(n)}|_{B\rho} = \frac{1}{B\rho} \frac{d^n \vec{e}}{ds^n} \quad \text{and} \quad (\vec{u} \times \vec{B})^{(n)}|_{B\rho} = \frac{1}{B\rho} (\vec{u} \times \vec{b})^{(n)}. \tag{1.2.24}$$

These derivatives  $\vec{E}^{(n)}$  and  $\vec{B}^{(n)}$  of the electric and magnetic fields are calculated from the vector fields  $\vec{E}(X, Y, Z)$ ,  $\vec{B}(X, Y, Z)$  and their derivatives  $\frac{\partial^{i+j+k} \vec{E}}{\partial X^i \partial Y^j \partial Z^k}$  and  $\frac{\partial^{i+j+k} \vec{B}}{\partial X^i \partial Y^j \partial Z^k}$ , following eqs. (1.2.14, 1.2.15) and eqs. (1.2.9, 1.2.10), respectively.

### 1.2.4 Calculation of the Time of Flight

The time of flight eq. (1.2.6) involves the derivatives  $dT/ds = 1/v$ ,  $d^2T/ds^2 = d(1/v)/ds$ , etc. that are obtained from eq. (1.2.21). In the absence of electric field, however, eq. (1.2.6) reduces to the simple form

$$T(M_1) = T(M_0) + \Delta s/v \tag{1.2.25}$$

### 1.3 Calculation of $\vec{E}$ and $\vec{B}$ Fields and their Derivatives

In this section, unless otherwise stated,  $\vec{B} = (B_X(X, Y, Z), B_Y(X, Y, Z), B_Z(X, Y, Z))$  stands indifferently for electric field  $\vec{E}$  or magnetic field  $\vec{B}$ .

$\vec{B}(X, Y, Z)$  and derivatives are calculated in various ways, depending whether field maps or analytic representations of optical elements are used. The basic means are the following.

### 1.3.1 1-D (Axial) Analytical Field Models and Extrapolation

This procedure assumes cylindrical symmetry with respect to the  $X$ -axis. The longitudinal field component  $B_X$  or  $E_X(X, r = 0)$  ( $r = (Y^2 + Z^2)^{1/2}$ ), along that axis is derived by differentiation of an appropriate model of the potential  $V(X)$  (e.g., magnetic in *SOLENOID*, electrostatic in *EL2TUB*, *UNIPOT*). The longitudinal and radial field components  $B_X(X, r)$ ,  $B_r(X, r)$  and their derivatives off-axis  $\frac{\partial^{i+j} B_X}{\partial X^i \partial r^j}$  and  $\frac{\partial^{i+j} B_r}{\partial X^i \partial r^j}$  are obtained by Taylor expansions to the second to fifth order in  $r$  (depending on the optical element) assuming cylindrical symmetry (eq. (1.3.1)), and then transformed to the  $(X, Y, Z)$  Cartesian frame of **zgoubi** in order to provide the derivatives  $\frac{\partial^{i+j+k} \vec{B}}{\partial X^i \partial Y^j \partial Z^k}$  needed in eq. (1.2.15).

### 1.3.2 Extrapolation from 1-D axial field map [10]

A cylindrically symmetric field (e.g., using *BREVOL*, *ELREVOL*) can be described by an axial 1-D field map of its longitudinal component  $B_X(X, r = 0)$  ( $r = (Y^2 + Z^2)^{1/2}$ ), while the radial component on axis  $B_r(X, r = 0)$  is assumed to be zero.  $B_X(X, r = 0)$  is obtained at any point along the  $X$ -axis by a polynomial interpolation from the map mesh (see section 1.4.1). Then the field components  $B_X(X, r)$ ,  $B_r(X, r)$  at the position of the particle,  $(X, r)$  are obtained from Taylor expansions truncated at the fifth order in  $r$  (hence, up to the fifth order derivative  $\frac{\partial^5 B_X}{\partial X^5}(X, 0)$ ), assuming cylindrical symmetry

$$\begin{aligned} B_X(X, r) &= B_X(X, 0) - \frac{r^2}{4} \frac{\partial^2 B_X}{\partial X^2}(X, 0) + \frac{r^4}{64} \frac{\partial^4 B_X}{\partial X^4}(X, 0) \\ B_r(X, r) &= -\frac{r}{2} \frac{\partial B_X}{\partial X}(X, 0) + \frac{r^3}{16} \frac{\partial^3 B_X}{\partial X^3}(X, 0) - \frac{r^5}{384} \frac{\partial^5 B_X}{\partial X^5}(X, 0) \end{aligned} \quad (1.3.1)$$

Then, by differentiation with respect to  $X$  and  $r$ , up to the second order, these expressions provide the derivatives of  $\vec{B}(X, r)$ . Finally a conversion from the  $(X, r)$  coordinates to the  $(X, Y, Z)$  Cartesian coordinates of **zgoubi** is performed, thus providing the expressions  $\frac{\partial^{i+j+k} \vec{B}}{\partial X^i \partial Y^j \partial Z^k}$  needed in the eq. (1.2.10).

### 1.3.3 Extrapolation From Median Plane Field Models

In the median plane,  $B_Z(X, Y, 0)$  and its derivatives with respect to  $X$  or  $Y$  may be derived from analytical models (e.g., in Venus magnet - *VENUS*, and sharp edge multipoles *SEXQUAD* and *QUADISEX*) or numerically by polynomial interpolation from 2-D field maps (e.g., *CARTEMES*, *TOSCA*).

Median plane antisymmetry is assumed, which results in

$$\begin{aligned}
 B_X(X, Y, 0) &= 0 \\
 B_Y(X, Y, 0) &= 0 \\
 B_X(X, Y, Z) &= -B_X(X, Y, -Z) \\
 B_Y(X, Y, Z) &= -B_Y(X, Y, -Z) \\
 B_Z(X, Y, Z) &= B_Z(X, Y, -Z)
 \end{aligned} \tag{1.3.2}$$

Accommodated with Maxwell's equations, this results in Taylor expansions below, for the three components of  $\vec{B}$  (here,  $B$  stands for  $B_Z(X, Y, 0)$ )

$$\begin{aligned}
 B_X(X, Y, Z) &= Z \frac{\partial B}{\partial X} - \frac{Z^3}{6} \left( \frac{\partial^3 B}{\partial X^3} + \frac{\partial^3 B}{\partial X \partial Y^2} \right) \\
 B_Y(X, Y, Z) &= Z \frac{\partial B}{\partial Y} - \frac{Z^3}{6} \left( \frac{\partial^3 B}{\partial X^2 \partial Y} + \frac{\partial^3 B}{\partial Y^3} \right) \\
 B_Z(X, Y, Z) &= B - \frac{Z^2}{2} \left( \frac{\partial^2 B}{\partial X^2} + \frac{\partial^2 B}{\partial Y^2} \right) + \frac{Z^4}{24} \left( \frac{\partial^4 B}{\partial X^4} + 2 \frac{\partial^4 B}{\partial X^2 \partial Y^2} + \frac{\partial^4 B}{\partial Y^4} \right)
 \end{aligned} \tag{1.3.3}$$

which are then differentiated one by one with respect to  $X$ ,  $Y$ , or  $Z$ , up to second or fourth order (depending on optical element or *IORDRE* option, see section 1.4.2) so as to get the expressions involved in eq. (1.2.10).

### 1.3.4 Extrapolation from Arbitrary 2-D Field Maps

2-D field maps that give the three components  $B_X(X, Y, Z_0)$ ,  $B_Y(X, Y, Z_0)$  and  $B_Z(X, Y, Z_0)$  at each node  $(X, Y)$  of a  $Z_0$   $Z$ -elevation map may be used.  $\vec{B}$  and its derivatives at any point  $(X, Y, Z)$  are calculated by polynomial interpolation followed by Taylor expansions in  $Z$ , without any hypothesis of symmetries (see section 1.4.3 and keywords *MAP2D*, *MAP2D-E*).

### 1.3.5 Interpolation in 3-D Field Maps [11]

In 3-D field maps  $\vec{B}$  and its derivatives up to the second order with respect to  $X$ ,  $Y$  or  $Z$  are calculated by means of a second order polynomial interpolation, from 3-D  $3 \times 3 \times 3$ -point grid (see section 1.4.4).

### 1.3.6 2-D Analytical Field Models and Extrapolation

Several optical elements such as *BEND*, *WIENFILT* (that uses the *BEND* procedures), *QUADISEX*, *VENUS*, etc., are defined from the expression of the field and derivatives in the median plane. 3-D extrapolation of these off the median plane is drawn from Taylor expansions and Maxwell's equations.

### 1.3.7 3-D Analytical Models of Fields

In many optical elements such as *QUADRUPOL*, *SEXTUPOL*, *MULTIPOL*, *EBMULT*, etc., the three components of  $\vec{B}$  and their derivatives with respect to  $X$ ,  $Y$  or  $Z$  are obtained at any step along trajectories from analytical expression drawn from the scalar potential  $V(X, Y, Z)$ , namely

$$B_X = \frac{\partial V}{\partial X}, \quad B_Y = \frac{\partial V}{\partial Y}, \quad B_Z = \frac{\partial V}{\partial Z}, \quad \frac{\partial B_X}{\partial X} = \frac{\partial^2 V}{\partial X^2}, \quad \frac{\partial B_X}{\partial Y} = \frac{\partial^2 V}{\partial X \partial Y}, \quad \text{etc.} \tag{1.3.4}$$

and similarly for  $\vec{E}$  with opposite sign for the gradients.

### Multipoles

The scalar potential used for the calculation of  $\frac{\partial^{i+j+k}\vec{B}_n(X, Y, Z)}{\partial X^i \partial Y^j \partial Z^k}$  ( $i + j + k = 0$  to 4) in the case of magnetic and electro-magnetic multipoles with  $2n$  poles (namely, *QUADRUPO* ( $n = 2$ ) to *DODECAPO* ( $n = 6$ ), *MULTIPOL* ( $n = 1$  to 10), *EBMULT* ( $n = 1$  to 10)) is [12]

$$V_n(X, Y, Z) = (n!)^2 \left( \sum_{q=0}^{\infty} (-1)^q \frac{G^{(2q)}(X)(Y^2 + Z^2)^q}{4^q q! (n+q)!} \right) \left( \sum_{m=0}^n \frac{\sin\left(m\frac{\pi}{2}\right) Y^{n-m} Z^m}{m!(n-m)!} \right) \quad (1.3.5)$$

where  $G(X)$  is a longitudinal form factor, defined at the entrance or exit of the optical element by

$$G(s) = \frac{G_0}{1 + \exp(P(s))}, \quad G_0 = \frac{B_0}{R_0^{n-1}} \quad (1.3.6)$$

wherein  $B_0$  is the field at pole tip radius  $R_0$ , and

$$P(s) = C_0 + C_1 \left(\frac{s}{\lambda}\right) + C_2 \left(\frac{s}{\lambda}\right)^2 + C_3 \left(\frac{s}{\lambda}\right)^3 + C_4 \left(\frac{s}{\lambda}\right)^4 + C_5 \left(\frac{s}{\lambda}\right)^5$$

and  $s$  is the distance to the EFB.

### Skew Multipoles

A multipole component with arbitrary order  $n$  can be tilted independently of the others by an arbitrary angle  $A_n$  around the  $X$ -axis. If so, the calculation of the field and derivatives in the rotated axis  $(X, Y_R, Z_R)$  is done in two steps. First, they are calculated at the rotated position  $(X, Y_R, Z_R)$ , in the  $(X, Y, Z)$  frame, using the expression (1.3.5) above. Second,  $\vec{B}$  and its derivatives at  $(X, Y_R, Z_R)$  in the  $(X, Y, Z)$  frame are transformed into the new,  $(X, Y_R, Z_R)$  frame, by a rotation with angle  $A_n$ .

In particular a skew  $2n$ -pole component is created by taking  $A_n = \pi/2n$ .

### A Note on Electrostatic Multipoles

A right electric multipole has the same field equations as the like-order skew magnetic multipole. Therefore, calculation of right or skew electric or electro-magnetic multipoles (*ELMULT*, *EBMULT*, *ELMULT*) uses the same eq. (1.3.5) together with the rotation process as described in section 1.3.7. The same method is used for arbitrary rotation of any multipole component around the  $X$ -axis.

## 1.4 Calculation of $\vec{E}$ and $\vec{B}$ from Field Maps

In this section, unless otherwise stated,  $\vec{B} = (B_X(X, Y, Z), B_Y(X, Y, Z), B_Z(X, Y, Z))$  stands indifferently for electric field  $\vec{E}$  or magnetic field  $\vec{B}$ .

### 1.4.1 1-D Axial Map, with Cylindrical Symmetry

Let  $B_i$  be the value of the longitudinal component  $B_X(X, r = 0)$  of the field  $\vec{B}$ , at node  $i$  of a uniform mesh that defines a 1-D field map along the symmetry  $X$ -axis, while  $B_r(X, r = 0)$  is assumed to be zero ( $r = (Y^2 + Z^2)^{1/2}$ ). The field component  $B_X(X, r = 0)$  is calculated by a polynomial interpolation of the fifth degree in  $X$ , using a 5 points grid centered at the node of the 1-D map which is closest to the actual coordinate  $X$  of the particle.

The interpolation polynomial is

$$B(X, 0) = A_0 + A_1 X + A_2 X^2 + A_3 X^3 + A_4 X^4 + A_5 X^5 \quad (1.4.1)$$

and the coefficients  $A_i$  are calculated by expressions that minimize the quadratic sum

$$S = \sum_i (B(X, 0) - B_i)^2 \quad (1.4.2)$$

Namely, the source code contains the explicit analytical expressions of the coefficients  $A_i$  solutions of the normal equations  $\partial S/\partial A_i = 0$ .

The derivatives  $\frac{\partial^n B}{\partial X^n}(X, 0)$  at the actual position  $X$ , as involved in eqs. (1.3.1), are then obtained by differentiation of the polynomial (1.4.1), giving

$$\begin{aligned} \frac{\partial B}{\partial X}(X, 0) &= A_1 + 2A_2X + 3A_3X^2 + 4A_4X^3 + 5A_5X^4 \\ \frac{\partial^2 B}{\partial X^2}(X, 0) &= 2A_2 + 6A_3X + 12A_4X^2 + 20A_5X^3 \\ &\dots \\ \frac{\partial^5 B}{\partial X^5}(X, 0) &= 120A_5 \end{aligned} \quad (1.4.3)$$

#### 1.4.2 2-D Median Plane Map, with Median Plane Antisymmetry

Let  $B_{ij}$  be the value of  $B_Z(X, Y, 0)$  at the nodes of a mesh which defines a 2-D field map in the  $(X, Y)$  plane while  $B_X(X, Y, 0)$  and  $B_Y(X, Y, 0)$  are assumed to be zero. Such a map may have been built or measured in either Cartesian or polar coordinates. Whenever polar coordinates are used, a change to Cartesian coordinates (described below) provides the expression of  $\vec{B}$  and its derivatives as involved in eq. (1.2.10).

**zgoubi** provides three types of polynomial interpolation from the mesh (option *IODRE*) ; namely, a second order interpolation, with either a 9- or a 25-point grid, or a fourth order interpolation with a 25-point grid (Fig. 3).

If the 2-D field map is built up from computer simulation, the grid in principle simply aims at interpolating the field and derivatives at a given point from its 9 or 25 neighbors. On the other hand if the map results from field measurements, the grid also has the virtue of smoothing field fluctuations.

The mesh may be defined in Cartesian coordinates, (Figs. 3A and 3B) or in polar coordinates (Fig. 3C).

The interpolation grid is centered on the node which is closest to the projection in the  $(X, Y)$  plane of the actual point of the trajectory.

The interpolation polynomial is

$$B(X, Y, 0) = A_{00} + A_{10}X + A_{01}Y + A_{20}X^2 + A_{11}XY + A_{02}Y^2 \quad (1.4.4)$$

to the second order, or

$$\begin{aligned} B(X, Y, 0) &= A_{00} + A_{10}X + A_{01}Y + A_{20}X^2 + A_{11}XY + A_{02}Y^2 \\ &\quad + A_{30}X^3 + A_{21}X^2Y + A_{12}XY^2 + A_{03}Y^3 \\ &\quad + A_{40}X^4 + A_{31}X^3Y + A_{22}X^2Y^2 + A_{13}XY^3 + A_{04}Y^4 \end{aligned} \quad (1.4.5)$$

to the fourth order. The coefficients  $A_{ij}$  are calculated by expressions that minimize, with respect to  $A_{ij}$ , the quadratic sum

$$S = \sum_{ij} (B(X, Y, 0) - B_{ij})^2 \quad (1.4.6)$$

The source code contains the explicit analytical expressions of the coefficients  $A_{ij}$  solutions of the normal equations  $\partial S/\partial A_{ij} = 0$ .



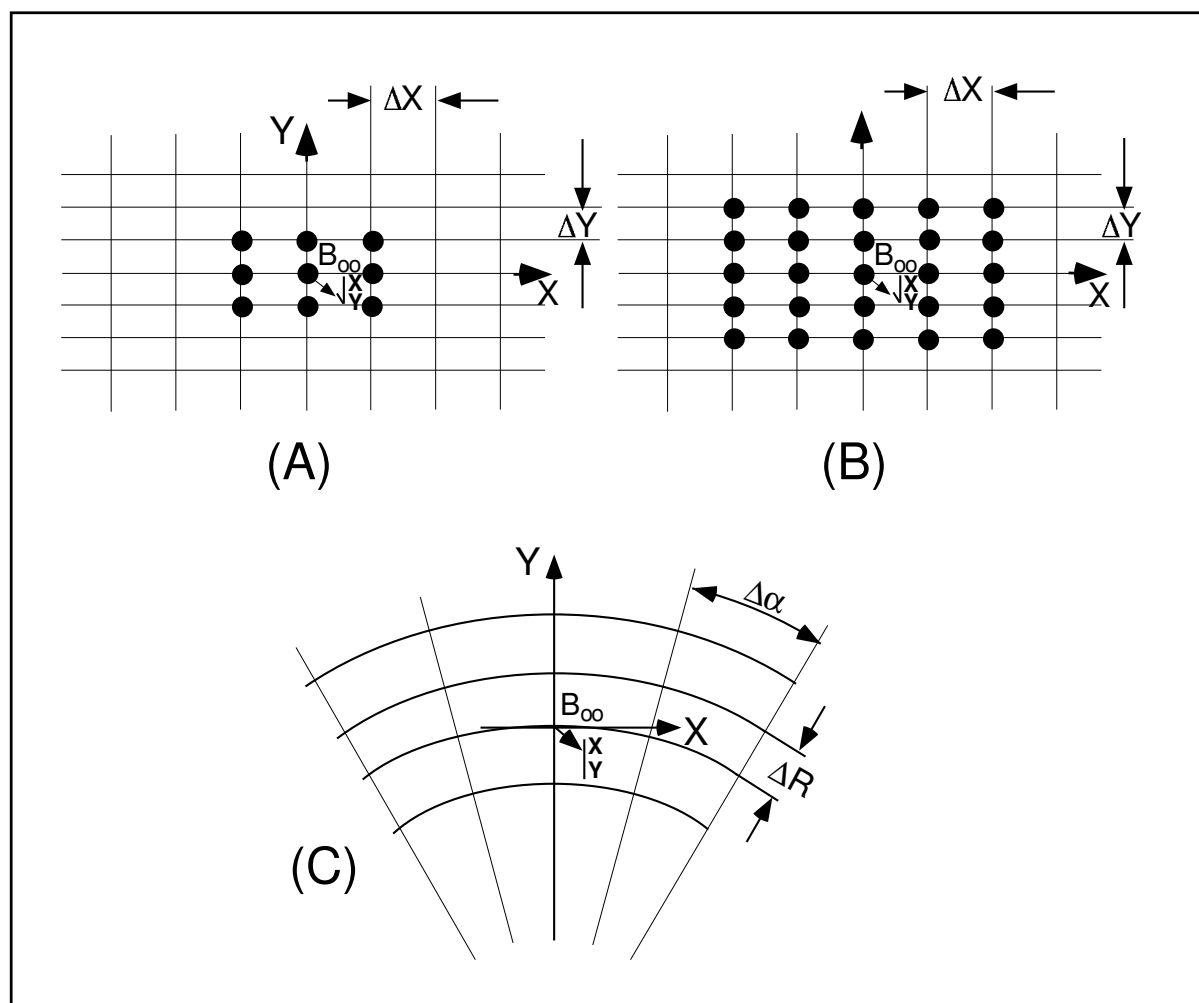


Figure 3: Mesh in the  $(X, Y)$  plane. The grid is centered on the node closest to the actual position of the particle.

A : Cartesian mesh, 9-point interpolation grid.

B : Cartesian mesh, 25-point interpolation grid.

C : Polar mesh and moving Cartesian frame.

The  $A_{ij}$  may then be identified with the derivatives of  $B(X, Y, 0)$  at the central node of the grid

$$A_{ij} = \frac{1}{i!j!} \frac{\partial^{i+j} B}{\partial X^i \partial Y^j} (0, 0, 0) \quad (1.4.7)$$

The derivatives of  $B(X, Y, 0)$  with respect to  $X$  and  $Y$ , at the actual point  $(X, Y, 0)$  are obtained by differentiation of the interpolation polynomial, which gives (*e.g.*, from (1.4.4) in the case of second order interpolation)

$$\begin{aligned} \frac{\partial B}{\partial X} (X, Y, 0) &= A_{10} + 2A_{20}X + A_{11}Y \\ \frac{\partial B}{\partial Y} (X, Y, 0) &= A_{01} + A_{11}X + 2A_{02}Y \\ &\text{etc.} \end{aligned} \quad (1.4.8)$$

This allows stepping to the calculation of  $\vec{B}(X, Y, Z)$  and its derivatives as described in subsection 1.3.3 (eq. 1.3.3).

### The Case of Polar Maps

In some optical elements (*e.g.*, *POLARMES*, *DIPOLE[S]*) the field is given in polar coordinates. It is thus necessary to transform the field and derivatives from the polar frame of the map,  $(R, \alpha, Z)$  to the Cartesian moving frame  $(X, Y, Z)$ , Fig. 3C. This is done as follows.

In second order calculations the correspondence is (we note  $B \equiv B_Z(Z = 0)$ )

$$\begin{aligned}
\frac{\partial B}{\partial X} &= \frac{1}{R} \frac{\partial B}{\partial \alpha} \\
\frac{\partial B}{\partial Y} &= \frac{\partial B}{\partial R} \\
\frac{\partial^2 B}{\partial X^2} &= \frac{1}{R^2} \frac{\partial^2 B}{\partial \alpha^2} + \frac{1}{R} \frac{\partial B}{\partial R} \\
\frac{\partial X \partial Y}{\partial^2 B} &= \frac{1}{R} \frac{\partial \alpha \partial R}{\partial^2 B} - \frac{1}{R^2} \frac{\partial B}{\partial \alpha} \\
\frac{\partial Y^2}{\partial^2 B} &= \frac{\partial R^2}{\partial^2 B} \\
\frac{\partial^3 B}{\partial X^3} &= \frac{3}{R^2} \frac{\partial^2 B}{\partial \alpha \partial R} - \frac{2}{R^3} \frac{\partial B}{\partial \alpha} \\
\frac{\partial X^2 \partial Y}{\partial^3 B} &= \frac{-2}{R^3} \frac{\partial^2 B}{\partial \alpha^2} - \frac{1}{R^2} \frac{\partial B}{\partial R} + \frac{1}{R} \frac{\partial^2 B}{\partial R^2} \\
\frac{\partial X \partial Y^2}{\partial^3 B} &= \frac{2}{R^3} \frac{\partial B}{\partial \alpha} - \frac{2}{R^2} \frac{\partial^2 B}{\partial \alpha \partial R} \\
\frac{\partial^3 B}{\partial Y^3} &= 0
\end{aligned} \tag{1.4.9}$$

In fourth order calculations the relations above are pushed to fourth order in  $X$ ,  $Y$  whereas

$$\begin{aligned}
\frac{\partial^3 B}{\partial X^3} &= \frac{1}{R^3} \frac{\partial^3 B}{\partial \alpha^3} + \frac{3}{R^2} \frac{\partial^2 B}{\partial \alpha \partial R} - \frac{2}{R^3} \frac{\partial B}{\partial \alpha} \\
\frac{\partial X^2 \partial Y}{\partial^3 B} &= \frac{1}{R^2} \frac{\partial^3 B}{\partial \alpha^2 \partial R} - \frac{2}{R^3} \frac{\partial^2 B}{\partial \alpha^2} - \frac{1}{R^2} \frac{\partial B}{\partial R} + \frac{1}{R} \frac{\partial^2 B}{\partial R^2} \\
\frac{\partial X \partial Y^2}{\partial^3 B} &= \frac{1}{R} \frac{\partial \alpha \partial R^2}{\partial^3 B} + \frac{2}{R^3} \frac{\partial B}{\partial \alpha} - \frac{2}{R^2} \frac{\partial^2 B}{\partial \alpha \partial R} \\
\frac{\partial Y^3}{\partial^3 B} &= \frac{\partial R^3}{\partial^3 B} \\
\frac{\partial^4 B}{\partial X^4} &= \frac{1}{R^4} \frac{\partial^4 B}{\partial \alpha^4} - \frac{8}{R^4} \frac{\partial^2 B}{\partial \alpha^2} + \frac{6}{R^3} \frac{\partial^3 B}{\partial \alpha^2 \partial R} + \frac{3}{R^2} \frac{\partial^2 B}{\partial R^2} - \frac{3}{R^3} \frac{\partial B}{\partial R} \\
\frac{\partial X^3 \partial Y}{\partial^4 B} &= \frac{1}{R^3} \frac{\partial^4 B}{\partial \alpha^3 \partial R} - \frac{3}{R^4} \frac{\partial^3 B}{\partial \alpha^3} + \frac{2}{R^2} \frac{\partial \alpha \partial R^2}{\partial^2 B} - \frac{8}{R^3} \frac{\partial^2 B}{\partial \alpha \partial R} + \frac{6}{R^4} \frac{\partial B}{\partial \alpha} \\
\frac{\partial X^2 \partial Y^2}{\partial^4 B} &= \frac{1}{R^4} \frac{\partial^4 B}{\partial \alpha^2} - \frac{3}{R^3} \frac{\partial \alpha^2 \partial R}{\partial^3 B} - \frac{2}{R^2} \frac{\partial R^2}{\partial^2 B} + \frac{2}{R^3} \frac{\partial B}{\partial R} + \frac{1}{R^2} \frac{\partial \alpha^2 \partial R^2}{\partial^2 B} + \frac{1}{R} \frac{\partial^3 B}{\partial R^3} \\
\frac{\partial X \partial Y^3}{\partial^4 B} &= \frac{1}{R} \frac{\partial^4 B}{\partial \alpha \partial R^3} - \frac{3}{R^2} \frac{\partial^3 B}{\partial \alpha \partial R^2} + \frac{6}{R^3} \frac{\partial^2 B}{\partial \alpha \partial R} - \frac{6}{R^4} \frac{\partial^4 B}{\partial \alpha^4} \\
\frac{\partial Y^4}{\partial^4 B} &= \frac{\partial R^4}{\partial^4 B}
\end{aligned} \tag{1.4.10}$$

**NOTE :** In case a particle goes beyond the limits of the field map, the field and its derivatives are extrapolated using a grid at the border of the map, which is the closest to the actual position of the particle. The flag *IEX* attached to the particle (section 7.13, p. 220) is then given the value  $-1$ .

### 1.4.3 Arbitrary 2-D Map, no Symmetry

The map is assumed to describe the field  $\vec{B}(B_X, B_Y, B_Z)$  in the  $(X, Y)$  plane at elevation  $Z_0$ . It provides the components  $B_{X,ij}$ ,  $B_{Y,ij}$ ,  $B_{Z,ij}$  at each node  $(i, j)$  of a 2-D mesh.

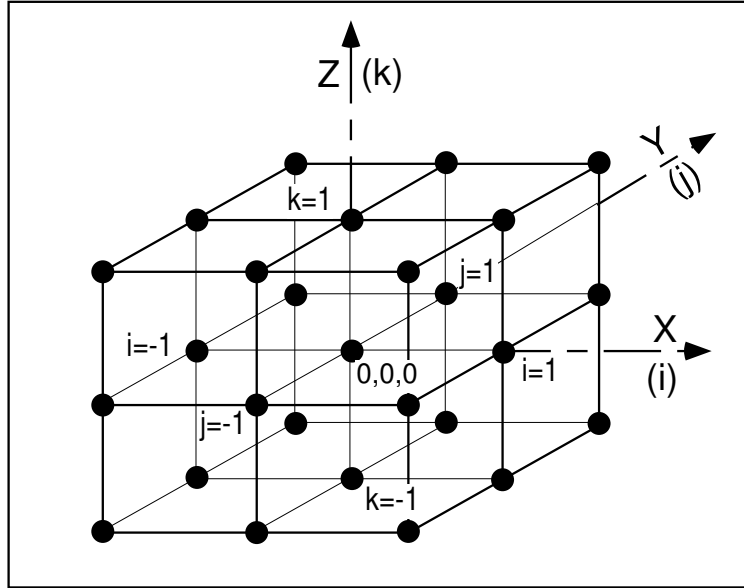


Figure 4: A 3-D, 27-point grid is used for interpolation of magnetic or electric fields and derivatives up to second order. The central node of the grid ( $i = j = k = 0$ ) is the closest to the actual position of the particle.

The value of  $\vec{B}$  and its derivatives at the projection  $(X, Y, Z_0)$  of the actual position  $(X, Y, Z)$  of a particle is obtained by means of (parameter *IORBRE* in the keyword argument list - see for instance *MAP2D*, *MAP2D-E*) either a second degree polynomial interpolation from a  $3 \times 3$  points grid (*IORBRE*=2), or a fourth degree polynomial interpolation from a  $5 \times 5$  points grid (*IORBRE*=4), centered at the node  $(i, j)$  closest to the position  $(X, Y)$ .

To second order for instance

$$B_\ell(X, Y, Z_0) = A_{00} + A_{10}X + A_{01}Y + A_{20}X^2 + A_{11}XY + A_{02}Y^2 \quad (1.4.11)$$

where  $B_\ell$  stands for any of the three components  $B_X$ ,  $B_Y$  or  $B_Z$ . Differentiating then gives the derivatives

$$\begin{aligned} \frac{\partial B_\ell}{\partial X}(X, Y, Z_0) &= A_{10} + 2A_{20}X + A_{11}Y \\ \frac{\partial^2 B_\ell}{\partial X \partial Y}(X, Y, Z_0) &= A_{11} \\ &\text{etc.} \end{aligned} \quad (1.4.12)$$

Then follows a procedure of extrapolation from  $(X, Y, Z_0)$  to the actual position  $(X, Y, Z)$ , based on Taylor series development.

No special symmetry is assumed, which allows the treatment of arbitrary field distribution (*e.g.*, solenoid, helical snake).

#### 1.4.4 3-D Field Map

When using a 3-D field map, the vector field  $\vec{B}(X, Y, Z)$  and its derivatives necessary for the calculation of position and velocity of the particle are obtained via second degree polynomial interpolation,

$$B_\ell(X, Y, Z) = A_{000} + A_{100}X + A_{010}Y + A_{001}Z + A_{200}X^2 + A_{020}Y^2 + A_{002}Z^2 + A_{110}XY + A_{101}XZ + A_{011}YZ \quad (1.4.13)$$

$B_\ell$  stands for any of the three components,  $B_X$ ,  $B_Y$  or  $B_Z$ . By differentiation of  $B_\ell$  one gets

$$\begin{aligned} \frac{\partial B_\ell}{\partial X} &= A_{100} + 2A_{200}X + A_{110}Y + A_{101}Z \\ \frac{\partial^2 B_\ell}{\partial X^2} &= 2A_{200} \end{aligned} \quad (1.4.14)$$

and so on up to second order derivatives with respect to  $X$ ,  $Y$  or  $Z$ .

The interpolation involves a  $3 \times 3 \times 3$ -point parallelepipedic grid (Fig. 4), the origin of which is positioned at the node of the 3-D field map which is closest to the actual position of the particle.

Let  $B_{ijk}^\ell$  be the value of the — measured or computed — magnetic field at each one of the 27 nodes of the 3-D grid ( $B^\ell$  stands for  $B_X$ ,  $B_Y$  or  $B_Z$ ), and  $B_\ell(X, Y, Z)$  be the value at a position  $(X, Y, Z)$  with respect to the central node of the 3-D grid. Thus, any coefficient  $A_i$  of the polynomial expansion of  $B_\ell$  is obtained by means of expressions that minimize, with respect to  $A_i$ , the sum

$$S = \sum_{ijk} (B_\ell(X, Y, Z) - B_{ijk}^\ell)^2 \quad (1.4.15)$$

where the indices  $i$ ,  $j$  and  $k$  take the values -1, 0 or +1 so as to sweep the 3-D grid. The source code contains the explicit analytical expressions of the coefficients  $A_{ijk}$  solutions of the normal equations  $\partial S / \partial A_{ijk} = 0$ .



## 2 SPIN TRACKING

Spin motion computation was implemented in **zgoubi** in the SATURNE years [13], it has since been used in a number of studies, including the transport of polarized muons in the Neutrino Factory [14] ; polarization survival in the superB project [15] ; studies on proton polarization with helical snakes at AGS [16] and RHIC [17] ; in the magic-energy electrostatic ring of the proton-EDM project [18] ; in the Booster [19], AGS, RHIC hadron rings of the Electron Ion Collider (EIC) (Fig. 5); for polarization transmission in rapid cycling injector synchrotrons [20, 21] ; at the EIC electron injector [22] : for an ergodic approach of spin diffusion and polarization lifetime in the EIC electron storage ring [23].

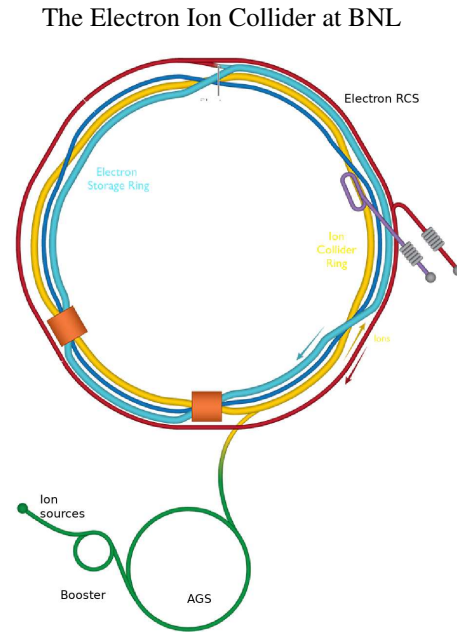


Figure 5: Zgoubi has tackled polarization in all the rings of the future EIC at BNL: polarized electron bunches in the RCS injector and the downstream ESR storage ring; polarized proton and  $^3\text{He}$  bunches in the AGS Booster, the AGS, and the HSR storage ring.

### 2.1 Introduction

The precession of the spin  $\vec{S}$  of a charged particle in electric and magnetic fields is governed by the Thomas-BMT first order differential equation [24]

$$\frac{d\vec{S}}{dt} = \frac{q}{m} \vec{S} \times \vec{\omega} \quad (2.1.1)$$

with, in the laboratory frame,

$$\begin{aligned} \vec{\omega} &= (1 + \gamma G)\vec{b} + G(1 - \gamma)\vec{b}_{\parallel} + \gamma \left( G + \frac{1}{1 + \gamma} \right) \frac{\vec{e} \times \vec{v}}{c^2} \\ &= (1 + G)\vec{b}_{\parallel} + (1 + \gamma G)\vec{b}_{\perp} + \gamma \left( G + \frac{1}{1 + \gamma} \right) \frac{\vec{e} \times \vec{v}}{c^2} \end{aligned} \quad (2.1.2)$$

$t$ ,  $q$ ,  $m$ ,  $c$ ,  $\gamma$  and  $G$  are respectively the time, charge, relativistic mass, speed of light, Lorentz relativistic factor and anomalous magnetic moment of the particle,  $\vec{b}$  and  $\vec{e}$  are the fields in the laboratory,  $\vec{b}_{\parallel}$  is the component of  $\vec{b}$  parallel to the velocity  $\vec{v}$  of the particle,  $\vec{b} = \vec{b}_{\parallel} + \vec{b}_{\perp}$ .

Equation (2.1.1) is normalized by introducing the same notation as for the Lorentz equation, page 23, namely  $b = \|\vec{b}\|$ ,  $v = \|\vec{v}\|$ ,  $\vec{v} = v\vec{u}$ ,  $ds = vdt$  the differential path,  $\frac{\gamma mv}{q} = B\rho$  the rigidity of the particle, whereas  $\vec{S}' = \frac{d\vec{S}}{ds} = \frac{1}{v} \frac{d\vec{S}}{dt}$  is the derivative of the spin with respect to the path. This yields

$$(B\rho) \vec{S}' = \vec{S} \times \vec{\omega} \quad \text{or} \quad \vec{S}' = \vec{S} \times \vec{\Omega} \quad (2.1.3)$$

where, noting  $\vec{B} = \vec{b}/B\rho$ ,  $\vec{E} = \vec{e}/B\rho$ ,

$$\vec{\Omega} = \frac{\vec{\omega}}{B\rho} = (1 + \gamma G)\vec{B} + G(1 - \gamma)\vec{B}_{\parallel} + \frac{\beta\gamma}{c} \left( G + \frac{1}{1 + \gamma} \right) \vec{E} \times \vec{u} \quad (2.1.4)$$

This equation is solved in the same way as the reduced Lorentz equation (1.2.3). From the values of the precession factor  $\vec{\omega}(M_0)$  and spin  $\vec{S}(M_0)$  of the particle at position  $M_0$ , the spin  $\vec{S}(M_1)$  at position  $M_1$ , following a displacement  $\Delta s$  (fig. 2), is obtained from truncated Taylor expansion

$$\vec{S}(M_1) \approx \vec{S}(M_0) + \frac{d\vec{S}}{ds}(M_0) \Delta s + \frac{d^2\vec{S}}{ds^2}(M_0) \frac{\Delta s^2}{2} + \frac{d^3\vec{S}}{ds^3}(M_0) \frac{\Delta s^3}{3!} + \frac{d^4\vec{S}}{ds^4}(M_0) \frac{\Delta s^4}{4!} + \frac{d^5\vec{S}}{ds^5}(M_0) \frac{\Delta s^5}{5!} \quad (2.1.5)$$

## 2.2 Integration in Magnetic Fields

In purely magnetic fields  $\vec{e} = 0$  thus eq. (2.1.4) reduces to

$$\vec{\Omega}_b = (1 + \gamma G)\vec{B} + G(1 - \gamma)\vec{B}_{\parallel} \quad (2.2.1)$$



$\vec{S}$  and its derivatives  $\vec{S}^{(n)} = d^n \vec{S} / ds^n$  satisfy the recursive differentiation relations

$$\begin{aligned}\vec{S}' &= \vec{S} \times \vec{\Omega}_b \\ \vec{S}'' &= \vec{S}' \times \vec{\Omega}_b + \vec{S} \times \vec{\Omega}'_b \\ \vec{S}''' &= \vec{S}'' \times \vec{\Omega}_b + 2\vec{S}' \times \vec{\Omega}'_b + \vec{S} \times \vec{\Omega}''_b \\ &\text{etc.}\end{aligned}\tag{2.2.2}$$

with the derivatives  $d^n \vec{\Omega}_b / ds^n$  obtained by differentiation of eq. (2.2.1). This requires  $\vec{B}_{\parallel}$  and derivatives, obtained in the following way,

$$\begin{aligned}\vec{B}_{\parallel} &= (\vec{B} \cdot \vec{u}) \vec{u} \\ \vec{B}'_{\parallel} &= (\vec{B}' \cdot \vec{u} + \vec{B} \cdot \vec{u}') \vec{u} + (\vec{B} \cdot \vec{u}) \vec{u}' \\ \vec{B}''_{\parallel} &= (\vec{B}'' \cdot \vec{u} + 2\vec{B}' \cdot \vec{u}' + \vec{B} \cdot \vec{u}'') \vec{u} + 2(\vec{B}' \cdot \vec{u} + \vec{B} \cdot \vec{u}') \vec{u}' + (\vec{B} \cdot \vec{u}) \vec{u}'' \\ &\text{etc.}\end{aligned}\tag{2.2.3}$$

The quantities  $\vec{u}$ ,  $\vec{B}$  and their derivatives as involved in these equations are known, being sub-products of the integration of the motion of the particle, eqs. (1.2.8, 1.2.10) p. 25.

### 2.3 Integration in Electric Fields

In purely electric fields  $\vec{b} = 0$ , thus eq. (2.1) reduces to

$$\omega_e = \frac{\beta\gamma}{c} \left( G + \frac{1}{1+\gamma} \right) \vec{e} \times \vec{u}\tag{2.3.1}$$

$\vec{S}$  and its derivatives  $\vec{S}^{(n)} = d^n \vec{S} / ds^n$  satisfy the recursive differentiation relations

$$\begin{aligned}B\rho \vec{S}' &= \vec{S} \times \vec{\omega}_e \\ (B\rho)' \vec{S}' + B\rho \vec{S}'' &= \vec{S}' \times \vec{\omega}_e + \vec{S} \times \vec{\omega}'_e \\ (B\rho)'' \vec{S}' + 2(B\rho)' \vec{S}'' + B\rho \vec{S}''' &= \vec{S}'' \times \vec{\omega}_e + 2\vec{S}' \times \vec{\omega}'_e + \vec{S} \times \vec{\omega}''_e \\ &\text{etc.}\end{aligned}\tag{2.3.2}$$

that provide the derivatives  $d^n \vec{S} / ds^n$  needed in the Taylor expansion (eq. 2.1.5). The derivatives  $(B\rho)^{(n)} = d^n (B\rho) / ds^n$  above are a sub-product of the integration of the force law (eq. 1.2.18). The derivatives of  $\omega_e$  (eq. 2.3.1) are obtained by recursive differentiation,

$$\begin{aligned}\vec{\omega}'_e &= \left( \frac{\beta\gamma}{c} \left( G + \frac{1}{1+\gamma} \right) \right)' \vec{e} \times \vec{u} + \frac{\beta\gamma}{c} \left( G + \frac{1}{1+\gamma} \right) (\vec{e} \times \vec{u})' \\ \vec{\omega}''_e &= \left( \frac{\beta\gamma}{c} \left( G + \frac{1}{1+\gamma} \right) \right)'' \vec{e} \times \vec{u} + 2 \left( \frac{\beta\gamma}{c} \left( G + \frac{1}{1+\gamma} \right) \right)' (\vec{e} \times \vec{u})' + \frac{\beta\gamma}{c} \left( G + \frac{1}{1+\gamma} \right) (\vec{e} \times \vec{u})'' \\ &\text{etc.}\end{aligned}\tag{2.3.3}$$

The quantities  $\left( \frac{1}{1+\gamma} \right)^{(n)}$  in the  $(\omega_e)^{(n)}$  above are obtained as follows.

From  $\gamma^2 = 1/(1 - \beta^2)$  it comes  $(\gamma + 1)(\gamma - 1)\beta^2/(1 - \beta^2)$  and then

$$\frac{1}{1 + \gamma} = \left(\frac{1}{\beta} - 1\right) \left(\frac{1}{\beta} + 1\right) (\gamma - 1)$$

which is differentiated, recursively,

$$\begin{aligned} \left(\frac{1}{1 + \gamma}\right)' &= \left(\frac{1}{\beta}\right)' \left(\frac{1}{\beta} + 1\right) (\gamma - 1) + \left(\frac{1}{\beta} - 1\right) \left(\frac{1}{\beta}\right)' (\gamma - 1) + \left(\frac{1}{\beta} - 1\right) \left(\frac{1}{\beta} + 1\right) \gamma' \\ \left(\frac{1}{1 + \gamma}\right)'' &= \left(\frac{1}{\beta}\right)'' \left(\frac{1}{\beta} + 1\right) (\gamma - 1) + \left(\frac{1}{\beta} - 1\right) \left(\frac{1}{\beta}\right)'' (\gamma - 1) + \left(\frac{1}{\beta} - 1\right) \left(\frac{1}{\beta} + 1\right) \gamma'' + \\ &\quad 2 \left(\frac{1}{\beta}\right)' \left(\frac{1}{\beta}\right)' (\gamma - 1) + 2 \left(\frac{1}{\beta}\right)' \left(\frac{1}{\beta}\right)' (\gamma - 1) + 2 \left(\frac{1}{\beta} - 1\right) \left(\frac{1}{\beta}\right)' \gamma' \\ \text{etc.} \end{aligned} \tag{2.3.4}$$

The interest of that formulation is that the  $\left(\frac{1}{\beta}\right)^{(n)}$  are already known from the particle dynamics, eq. (1.2.21), as well as the derivatives

$$\frac{d^{n+1}\gamma}{ds^{n+1}} = \frac{q}{m} \frac{d^n(\vec{e} \cdot \vec{u})}{ds^n}$$

following eq. (1.2.20), whereas the  $d^n(\vec{e} \cdot \vec{u})/ds^n$  are given by eq. (1.2.19).

## 2.4 Integration in Combined Electric and Magnetic Fields

When both  $\vec{e}$  and  $\vec{b}$  are non-zero, the complete eqs. (2.1.3, 2.1.4) must be considered.

The precession vector (eq. 2.1.4) and its derivatives can be split into independent, magnetic and electric components, namely

$$\vec{\omega} = \vec{\omega}_b + \vec{\omega}_e, \quad \vec{\omega}' = \vec{\omega}'_b + \vec{\omega}'_e, \quad \text{etc.}$$

As a consequence, the spin vector  $\vec{S}(s)$  and its derivatives at location  $M_0$  prior to a  $\Delta s$  push to location  $M_1$  (fig. 2) can be obtained by linear superposition of the separate solutions for the magnetic case (eq. 2.2.2) and for the electric case (eq. 2.3.2), namely

$$\begin{aligned} B\rho \vec{S}' &= \vec{S} \times \vec{\omega}_b && + \vec{S} \times \vec{\omega}_e \\ B\rho \vec{S}'' &= \vec{S}' \times \vec{\omega}_b + \vec{S} \times \vec{\omega}'_b && + \vec{S}' \times \vec{\omega}_e + \vec{S} \times \vec{\omega}'_e - (B\rho)' \vec{S}' \\ B\rho \vec{S}''' &= \underbrace{\vec{S}'' \times \vec{\omega}_b + 2\vec{S}' \times \vec{\omega}'_b + \vec{S} \times \vec{\omega}''_b}_{\text{Magnetic field component, eq. (2.2.2)}} && + \underbrace{\vec{S}'' \times \vec{\omega}_e + 2\vec{S}' \times \vec{\omega}'_e + \vec{S} \times \vec{\omega}''_e - \left( (B\rho)'' \vec{S}' + 2(B\rho)' \vec{S}'' \right)}_{\text{Electric field component, eq. (2.3.2)}} \end{aligned}$$

etc.

The process is then completed by applying the  $\Delta s$  push, eq. (2.1.5).

## 2.5 Radiative polarization

Radiative spin transition is accounted for in a classical manner with an additional term in the Thomas-BMT differential equation for the motion of the polarization vector [25], namely, see Eqs. 2.1.1, 2.1 (in the absence of electric fields),

$$\frac{d\vec{S}}{dt} = \frac{q}{m} \vec{S} \times \left( (1 + a\gamma)\vec{b} + a(1 - \gamma)\vec{b}_{\parallel} \right) - \frac{1}{\tau_{ST}} \left( \vec{S} - \frac{2}{9}(\vec{S} \cdot \vec{u})\vec{u} + \frac{8}{5\sqrt{3}} \frac{\vec{u} \times \dot{\vec{u}}}{|\dot{\vec{u}}|} \right) \quad (2.5.1)$$

where the electron notation for the magnetic anomaly,  $a$ , replaces hadron's  $G$  (and  $q = -e$  with  $e$  the elementary charge, for electrons),  $\vec{u}$  is the unit velocity vector of the particle and  $\dot{\vec{u}} = d\vec{u}/dt = v\vec{u}'$  (eq. 1.2.2). The Sokolov-Ternov polarization equilibrium time constant has been introduced,

$$\tau_{ST} = \frac{4}{5\sqrt{3}} \frac{E}{\hbar\omega_c} \tau_0 \quad (2.5.2)$$

with  $E = \gamma m_0$  the particle energy and  $\hbar\omega_c = h \times 3\gamma^3 c / 2\rho$  the critical photon energy.

This radiative term is only added if the particle does radiate [26], that event being determined by the Monte Carlo photon emission simulation process described in Sec. 3.1, switched on using the keyword **SRLOSS**. It has the effect of slightly changing the electron polarization, under the effect of radiation induced spin transition.

This radiative correction process is under installation in **zgoubi**. Its benchmarking includes comparison with statistical averages of quantum spin physics.

As to the numerical integration of spin motion with diffusion (*i.e.*, solving the differential eq. 2.5.1), the technique follows the **zgoubi** Taylor expansion method, as detailed in Sec. 2.1, using instead the successive derivatives of  $\vec{S}$  with respect to the path variable  $s$  drawn from Eq. 2.5.1.

## 2.6 Spin Diffusion

Spin diffusion is a sub-product of the Monte Carlo synchrotron radiation technique. The spin vector of a particle will diffuse under the effect of stochastic photon emission.

*TO BE DOCUMENTED FURTHER.*

### Spin Diffusion Integrals

They write

$$DK\ spinintegrals \quad (2.6.1)$$

These intergals are evaluated by stepwise integration<sup>2</sup> along a reference orbit (using a mid-point method), therefore requiring prior knowledge - or computation - of the latter and of the spin precession direction along. If the latter two quantities are known, they may just be specified as the initial coordinates in *OBJET* and in *SPNTRK*. If not, then (see Sec. 7.3) a *FIT[2]* of both initial orbit and initial spin conditions will update both *OBJET* and *SPNTRK* data to the required values. Note that, in the case of a ring sequence, these initial conditions are expected to be the periodic orbit coordianates and periodic (aka “closed”) spin precession axis coordinates, which *FIT[2]* can find.

*TWISS* keyword manages the all process (possibly preceded by a prior *FIT[2]* following the periodicity considerations above). Namely, *TWISS* will trigger DK spin integral computation if both *SRLOSS* and *SPNTRK* are present. Note that the presence of *SRLOSS* also results in the numerical, stepwise, computation of the synchrtron radiation integrals (Sec 3.1).

*UNDER INSTALLATION/BENCHMARKING.*  
*TO BE DOCUMENTED FURTHER.*

<sup>2</sup>For the geek : The process is managed in the same way as Monte Carlo Synchrotron radiation, *CHAMBER*, etc., by the “event” flag

### 3 SYNCHROTRON RADIATION

**zgoubi** provides the simulation of two distinct types of synchrotron radiation (SR) manifestations namely, on the one hand energy loss by stochastic emission of photons and the ensuing perturbation on particle dynamics and, on the other hand the time-dependent radiated electro-magnetic field impulse, and its spectral-angular energy density as observed in the laboratory.

**SR loss simulation** was first installed in **zgoubi** in view of beam dynamics studies in the beam delivery system of the Next Linear Collider [27], based on a method developed in the frame of the ELFE project (“EU Lab. For Electrons”) [28, 29, 30]. It was later used, and benchmarked including coupling, to account for damping effects in beam studies for various electron-ring and recirculator projects [15, 31, 32], and to account for spin diffusion studies in eRHIC collider lattice - which includes spin rotators [23].

**SR electromagnetic impulse and spectrum** computation was first installed in **zgoubi** for the study of interference effects at the LEP beam diagnostics miniwiggler [34]. These simulation tools were next used to design the LHC SR beam diagnostics systems, located in the IR4 RF section [35].

#### 3.1 Energy Loss and Dynamical Effects

Most of the contents of the present section are drawn from Ref. [27].

Given a particle wandering in the magnetic field of an arbitrary optical element or field map, **zgoubi** computes the energy loss undergone, and its effect on the particle motion. The energy loss is calculated in a classical manner, by invoking two random processes that accompany the emission of a photon namely,

- the probability of emission,
- the energy of the photon.

The effects on the dynamics of the emitting particle either account for the sole alteration of its energy, or, if requested, include the angle kick. Particle position is supposed not to change upon emission of a photon. These calculations and ensuing dynamics corrections are performed at each integration step. In a practical manner, this requires centimeters to tens of centimeters steps in smoothly varying magnetic fields (a quantity to be determined before any simulation, from convergence trials).

Main aspects of the method are developed in the following.

#### Probability of Emission of a Photon

Given that the number of photons emitted within a step  $\Delta s$  can be very small (units or fractions of a unit)<sup>3</sup> a Poisson probability law

$$p(k) = \frac{\lambda^k}{k!} \exp(-\lambda) \quad (3.1.1)$$

is considered.  $k$  is the number of photons emitted over a  $\Delta\theta$  (circular) arc of trajectory such that, the average number of photons per radian expresses as<sup>4</sup>

$$\lambda = \frac{20er_0}{8\hbar\sqrt{3}} \beta^2 B\rho \frac{\Delta s}{\rho} \quad (3.1.2)$$

where  $r_0 = e^2/4\pi\epsilon_0 m_0 c^2$  is the classical radius of the particle of rest-mass  $m_0$ ,  $e$  is the elementary charge,  $\hbar = h/2\pi$ ,  $h$  is the Planck constant,  $\beta = v/c$ ,  $B\rho$  is the particle stiffness.  $\lambda$  is evaluated at each integration step from the current values  $\beta$ ,  $B\rho$  and  $\Delta s$ , then a value of  $k$  is drawn by a rejection method.

<sup>3</sup>For instance, a 1 GeV electron will emit about 20.6 photons per radian ; an integration step size  $\Delta s = 0.1$  m upon  $\rho = 10$  m bending radius results in 0.2 photon per step.

<sup>4</sup>This leads for instance, in the case of electrons, to the classical formula  $\lambda/\Delta\theta \approx 129.5E(\text{GeV})/2\pi \approx \gamma/94.9$ .

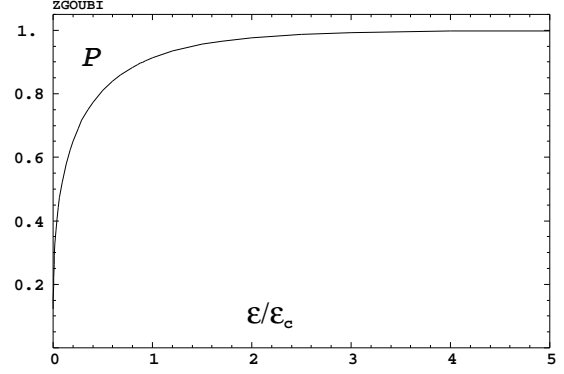
### Energy of the Photons

These  $k$  photons are assigned energies  $\epsilon = h\nu$  at random, in the following way. The cumulative distribution of the energy probability law  $p(\epsilon/\epsilon_c)d\epsilon/\epsilon_c$  (i.e., the probability of emission of a photon with energy in  $[0, \epsilon]$ ) writes <sup>5</sup>

$$\mathcal{P}(\epsilon/\epsilon_c) = \frac{3}{5\pi} \int_0^{\epsilon/\epsilon_c} \frac{d\epsilon}{\epsilon_c} \int_{\epsilon/\epsilon_c}^{\infty} K_{5/3}(x) dx \quad (3.1.3)$$

where  $K_{5/3}$  is a modified Bessel function and,  $\epsilon_c = \hbar\omega_c$ ,  $\omega_c = 2\pi \times 3\gamma^3 c / 2\rho$  being the critical frequency in the presence of bending radius  $\rho$ ;  $\omega_c$  is evaluated at each integration step from the current values  $\gamma$  and  $\rho$ , in other words, this energy loss calculation assumes constant magnetic field over the trajectory arc  $\Delta s$ . In the low frequency region ( $\epsilon/\epsilon_c \ll 1$ ) it can be approximated by

$$\mathcal{P}(\epsilon/\epsilon_c) = \frac{12\sqrt{3}}{2^{1/3} 5 \Gamma(1/3)} \left( \frac{\epsilon}{\epsilon_c} \right)^{1/3} \quad (3.1.4)$$



Cumulative energy distribution  $\mathcal{P}(\epsilon/\epsilon_c)$ .

About 40 values of  $\mathcal{P}(\epsilon/\epsilon_c)$  computed from eq. 3.1.3 [33], honestly spread over a range  $\epsilon/\epsilon_c \leq 10$  are tabulated in **zgoubi** source file (see figure). In order to get  $\epsilon/\epsilon_c$ , first a random value  $0 < \mathcal{P} < 1$  is generated uniformly, then  $\epsilon/\epsilon_c$  is drawn either by simple inverse linear interpolation of the tabulated values if  $\mathcal{P} > 0.26$  (corresponding to  $\epsilon/\epsilon_c > 10^{-2}$ ), or, if  $\mathcal{P} < 0.26$  from eq. 3.1.4 that gives  $\epsilon/\epsilon_c = \left( \frac{2^{1/3} 5 \Gamma(1/3)}{12\sqrt{3}\mathcal{P}} \right)^3$  with precision no less than 1% at  $\mathcal{P} \rightarrow 0.26$ .

When SR loss tracking is requested, several optical elements that contain a dipole magnetic field component (e.g., **MULTIPOL**, **BEND**) provide a printout of various quantities related to SR emission, as drawn from classical theoretical expressions, such as for instance,

- energy loss per particle  $\Delta E(eV) = \frac{2}{3} r_0 c \gamma^3 B(T) \Delta\theta$ , with  $B$  the dipole field exclusive of any other multipole component in the magnet,  $\Delta\theta$  the total deviation as calculated from  $B$ , from the magnet length, and from the reference rigidity **BORO** (as defined with, e.g., **[MC]OBJET**)

- critical energy  $\epsilon_c(eV) = \frac{3\gamma^3 c \hbar}{2\rho e}$ , with  $\rho = \text{BORO} / B$

- average energy of the photons radiated  $\langle \epsilon \rangle = \frac{8}{15\sqrt{3}} \epsilon_c$ ,

- rms energy of radiated photons  $\epsilon_{rms} = 0.5591 \epsilon_c$ ,

- number of radiated photons per particle  $N = \Delta E / \langle \epsilon \rangle$ .

This is done in order to facilitate verifications, since on the other hand statistics regarding those values are drawn from the tracking and may be printed using the dedicated keyword **SRPRNT**.

Finally, upon user's request as well, SR loss can be limited to particular classes of optical elements, for instance dipole magnets alone, or dipole + quadrupole magnets, etc. This option is made available in order to allow further inspection, or easier comparison with other codes, for instance.

<sup>5</sup>From a practical viewpoint, the value of the magnetic field first computed for a one-step push of the particle (eqs. 1.2.4, 1.2.8) is next used to obtain  $\rho$  and perform SR loss corrections afterwards.

### Synchrotron Radiation Integrals

Numerical computation of Synchrotron radiation integrals, if requested, is a subproduct of the stepwise integration of particle motion<sup>6</sup>. The process is managed by the *TWISS* procedure, SR integrals computation is triggered if (and only if) *SRLOSS* is present in the input sequence. The integral values so obtained are part of the data inventory in the *zgoubi.TWISS.out* output file header.

Synchrotron radiation integrals are evaluated by stepwise integration (using a mid-point method) along a reference orbit, therefore requiring prior knowledge - or computation - of the latter. If that quantity is known, it may just be specified as the initial coordinates in *OBJET*. If not, then a *FIT* of the initial orbit conditions will update *OBJET* data to the required values. Note that, in the case of a ring sequence, these initial conditions are expected to be the periodic orbit coordinates, which *FIT[2]* can find.

*UNDER INSTALLATION/BENCHMARKING.*

*TO BE DOCUMENTED FURTHER.*

<sup>6</sup>For the geek : The process is managed in the same way as Monte Carlo Synchrotron radiation, *CHAMBER*, etc., by the “event” flag

### 3.2 Spectral-Angular Radiated Densities

Most of the content in the present section is drawn from Refs. [34, 35, 36].

The ray-tracing procedures provide the ingredients necessary for the determination of the electric field radiated by the particle subject to acceleration, as shown in Fig. 6, this is developed in section 3.2.1. These ingredients further allow calculating the spectral-angular density of the radiation<sup>7</sup>, this is developed in section 3.2.2.

#### 3.2.1 Calculation of the Radiated Electric Field

The expression for the radiated electric field  $\vec{\mathcal{E}}(\vec{n}, \tau)$  as seen by the observer in the long distance approximation is [37]

$$\vec{\mathcal{E}}(\vec{n}, \tau) = \frac{q}{4\pi\epsilon_0 c} \frac{\vec{n}(t) \times \left[ \left( \vec{n}(t) - \vec{\beta}(t) \right) \times d\vec{\beta}/dt \right]}{r(t) \left( 1 - \vec{n}(t) \cdot \vec{\beta}(t) \right)^3} \quad (3.2.1)$$

where  $t$  is the time in which the particle motion is described and  $\tau$  is the observer time. Namely, when at position  $\vec{r}(t)$  with respect to the observer (or as well at position  $\vec{R}(t) = \vec{X} - \vec{r}(t)$  in the  $(O, x, y, z)$  frame) the particle emits a signal which reaches the observer at time  $\tau$ , such that  $\tau = t + r(t)/c$  where  $r(t)/c$  is the delay necessary for the signal to travel from the emission point to the observer, which also leads by differentiation to the well-known relation

$$\frac{d\tau}{dt} = 1 - \vec{n}(t) \cdot \vec{\beta}(t) \quad (3.2.2)$$

The vectors  $\vec{R}(t)$  and  $\vec{\beta}(t) = \frac{v}{c}\vec{u}$  (eq. 1.2.2) that describe the motion are obtained from the ray-tracing (eqs. 1.2.4). The acceleration is calculated from (a form of eq. 1.2.1 with  $\vec{e} = 0$ )

$$\frac{d\vec{\beta}}{dt} = \frac{q}{m} \vec{\beta}(t) \times \vec{b}(t) \quad (3.2.3)$$

<sup>7</sup>These calculations have been installed in the post-processor **zpop**.



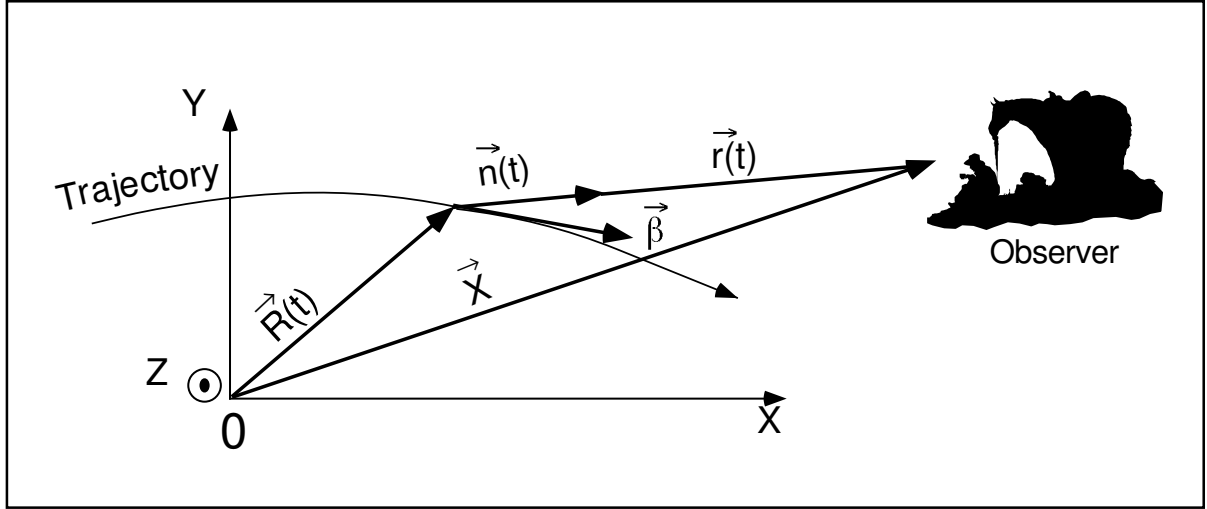


Figure 6: A scheme of the reference frame in **zgoubi** together with the vectors entering in the definition of the electric field radiated by the accelerated particle :

- $(x, y)$  : horizontal plane ;  $z$  : vertical axis.
- $\vec{R}(t)$  = particle position in the fixed frame  $(O, x, y, z)$  ;
- $\vec{X}$  (time-independent) = position of the observer in the  $(O, x, y, z)$  frame ;
- $\vec{r}(t) = \vec{X} - \vec{R}(t)$  = position of the particle with respect to the observer ;
- $\vec{n}(t)$  = (normalized) direction of observation =  $\vec{r}(t)/|\vec{r}(t)|$  ;
- $\vec{\beta}$  = normalized velocity vector of the particle  $\vec{v}/c = (1/c)d\vec{R}/dt$ .

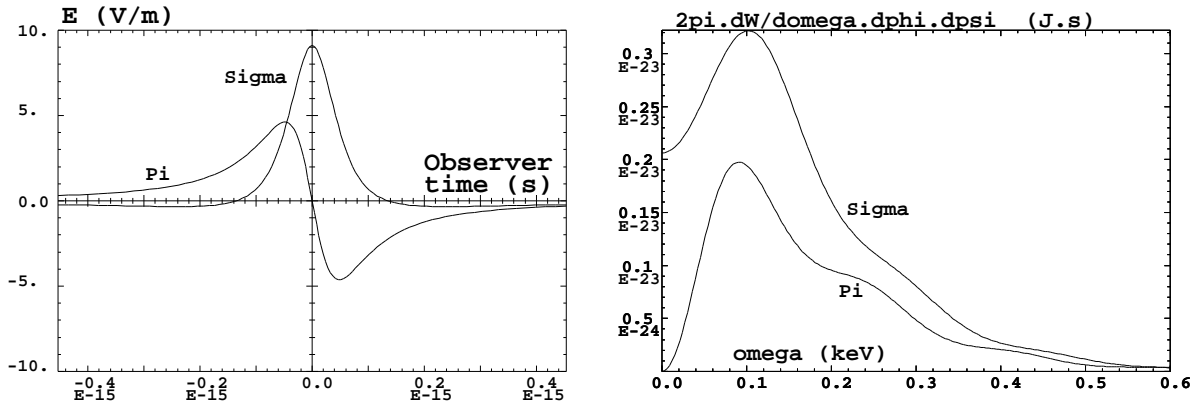


Figure 7: *Left* : typical shapes of the  $\mathcal{E}_\sigma(t)$  and  $\mathcal{E}_\pi(t)$  components of the electric field impulse (eq. 3.2.1) emitted by a 2.5 GeV electron on a  $\rho = 53.6$  m circular trajectory in a  $l = 20$  cm-long dipole, as observed in the direction of the centre of the dipole,  $\phi = 0$ ,  $\gamma\psi = 5$  (observation distance  $r = 10$  m). *Right* : the related spectral angular densities  $\partial^3 W_{\sigma, \pi} / \partial\phi \partial\psi \partial\omega$  (eq. 3.2.10). After Ref. [36].

Then, given the observer position  $\vec{X}$  in the fixed frame, it is possible to calculate

$$\vec{r}(t) = \vec{X} - \vec{R}(t) \quad \text{and} \quad \vec{n}(t) = \frac{\vec{r}(t)}{|\vec{r}(t)|} \quad (3.2.4)$$

As an illustration, Fig. 7 has been produced using **zpop**, it shows the typical shape of the electric field impulse at the observer, with central peak width [36]

$$2\tau_c = 2 \frac{2\rho}{3\gamma^3 c} (1 + \gamma^2\psi^2)^{3/2}$$

### Computation of $\vec{n} - \vec{\beta}$ and $1 - \vec{n} \cdot \vec{\beta}$

Owing to computer precision the crude computation of  $\vec{n} - \vec{\beta}$  and  $1 - \vec{n} \cdot \vec{\beta}$  may lead to

$$\vec{n} - \vec{\beta} = 0 \quad \text{and} \quad 1 - \vec{n} \cdot \vec{\beta} = 0$$

since the preferred direction of observation is generally almost parallel to  $\vec{\beta}$  (in that case, parallel in the sense of computer precision), while  $\beta \approx 1$  as soon as particle energies of a few hundred times the rest mass are concerned.

It is therefore necessary to express  $\vec{n} - \vec{\beta}$  and  $1 - \vec{n} \cdot \vec{\beta}$  in an adequate form for achieving accurate software computation.

The expression for  $\vec{n}$  is

$$\begin{aligned} \vec{n} = (n_x, n_y, n_z) &= (\cos \psi \cos \phi, \cos \psi \sin \phi, \sin \psi) \\ &= [1 - 2(\sin^2 \phi/2 + \sin^2 \psi/2) + 4 \sin^2 \phi/2 \sin^2 \psi/2, \sin \phi(1 - 2 \sin^2 \psi/2), \sin \psi] \end{aligned} \quad (3.2.5)$$

where  $\phi$  and  $\psi$  are the observation angles, given by

$$\phi = \text{Atg} \left( \frac{r_y}{r_x} \right) \quad \text{and} \quad \psi = \text{Atg} \left( \frac{r_z}{\sqrt{r_x^2 + r_y^2}} \right) \quad (3.2.6)$$

with  $\vec{r} = (r_x, r_y, r_z)$ , while  $\vec{\beta}$  can be written under the form

$$\begin{aligned} \vec{\beta} = (\beta_x, \beta_y, \beta_z) &= \left[ \sqrt{(\beta^2 - \beta_y^2 - \beta_z^2)}, \beta_y, \beta_z \right] \\ &= \left[ \sqrt{(1 - 1/\gamma^2 - \beta_y^2 - \beta_z^2)}, \beta_y, \beta_z \right] = (1 - a/2 + a^2/8 - a^3/16 + \dots, \beta_y, \beta_z) \end{aligned} \quad (3.2.7)$$

where  $a = 1/\gamma^2 + \beta_y^2 + \beta_z^2$ . This leads to

$$n_x = 1 - \varepsilon_x \text{ and } \beta_x = 1 - \xi_x$$

with

$$\varepsilon_x = 2(\sin^2 \phi/2 + \sin^2 \psi/2) - 4 \sin^2 \phi/2 \sin^2 \psi/2$$

and

$$\xi_x = a/2 - a^2/8 + a^3/16 + \dots$$

All this provides, on the one hand,

$$\vec{n} - \vec{\beta} = (-\varepsilon_x + \xi_x, n_y - \beta_y, n_z - \beta_z), \quad (3.2.8)$$

whose components are combinations of terms of the same order of magnitude ( $\varepsilon_x$  and  $\xi_x \sim 1/\gamma^2$  while  $n_y, \beta_y, n_z$  and  $\beta_z \sim 1/\gamma$ ) and, on the other hand,

$$1 - \vec{n} \cdot \vec{\beta} = \varepsilon_x + \xi_x - n_y \beta_y - n_z \beta_z - \varepsilon_x \xi_x, \quad (3.2.9)$$

that combines terms of the same order of magnitude ( $\varepsilon_x, \xi_x, n_y \beta_y$  and  $n_z \beta_z \sim 1/\gamma^2$ ), plus  $\varepsilon_x \beta_x \sim 1/\gamma^4$ . The precision of these expressions is directly related to the order at which the series

$$\xi_x = a/2 - a^2/8 + a^3/16 + \dots \quad (a = 1/\gamma^2 + \beta_y^2 + \beta_z^2)$$

is pushed, however the convergence is fast since  $a \sim 1/\gamma^2 \ll 1$  in situations of concern.

### 3.2.2 Calculation of the Fourier Transform of the Electric Field

The Fourier transforms

$$FT_\omega[\vec{\mathcal{E}}(\tau)] = \int \vec{\mathcal{E}}(\tau) e^{-i\omega\tau} d\tau$$

of the  $\sigma$  and  $\pi$  electric field components provide the spectral angular energy density

$$\frac{\partial^3 W}{\partial \phi \partial \psi \partial \omega} = \frac{r^2}{\mu_0 c} \left| FT_\omega \left( \vec{\mathcal{E}}(\tau) \right) \right|^2 \quad (3.2.10)$$

Fig. 7-right gives a typical example in the case of a short magnet. These Fourier transforms are computed without resorting to FFT techniques, namely from

$$FT_\omega \left[ \vec{\mathcal{E}}(\tau) \right] \approx \sum \vec{\mathcal{E}}(\tau_k) e^{-i\omega\tau_k} \Delta\tau_k \quad (3.2.11)$$

for two reasons. On the one hand, the number of integration steps  $\Delta s$  that define the trajectory (eqs. 1.2.4), is arbitrary and therefore in general not of order  $2^n$ . On the other hand, the integration step defines a constant time differential element  $\Delta t_k = \Delta s/\beta c$  which results in the observer differential time element  $\Delta\tau_k$ , which is also the differential element of the Fourier transform, being non-constant, since both are related by eq. 3.2.2 in which  $\vec{\beta}$  and  $\vec{n}$  vary as a function of the integration step number  $k$ .

An additional issue is that  $\Delta\tau_k$  may reach drastically small values in the region of the central peak of the electric impulse emitted in a dipole, *i.e.*,  $1 - \vec{n}(t) \cdot \vec{\beta}(t) \rightarrow 1/2\gamma^2$ , whereas the total integrated time  $\sum_{k=1}^N \Delta\tau_k$  may be several orders of magnitude larger. In terms of the physical phenomenon, the total duration of the electric field impulse as seen by the observer corresponds to the time delay  $\sum_{k=1}^N \Delta\tau_k$  that separates photons emitted at the entrance of the magnet from photons emitted at the exit, but the significant part of it (in terms of energy density) which can be represented by the width  $2\tau_c = 2 \frac{2\rho}{3\gamma^3 c} (1 + \gamma^2 \psi^2)^{3/2}$  of the radiation peak,

is a very small fraction of  $\sum_{k=1}^N \Delta\tau_k$ .

The consequence is that, again in relation to computer precision, the differential element  $\Delta\tau_k$  involved in the computation of eq. 3.2.11 cannot be derived from such relation as  $\Delta\tau_k = \sum_{k=1}^n \Delta\tau_k - \sum_{k=1}^{n-1} \Delta\tau_k$  but instead must be stored as is, in the course of the ray-tracing process, for subsequent data treatment.



#### 4 SPACE CHARGE

Zgoubi provides two models for the simulation of space charge effects: KV (effect of space is linear), and Gaussian (non linear). This capability has been first developed in the context of a PhD work regarding FFAG proton driver design studies and accelerator-driven subcritical reactor applications. This section of the guide is essentially drawn from there [38, 40], which can be referred to for additional details - the document provides benchmarking proofs in both KV and Gaussian density models, including bunch transport in a drift, transport in a beam line, bunch tracking in an FFAG ring represented using either **zgoubi**'s *FFAG* keyword or magnetic field maps. Some of these benchmarking simulations are also part of **zgoubi** source-forge package, at

[pathTo]/branches//exemples/KEYWORDS/SPACECHARG .

*TO BE DOCUMENTED FURTHER. Refer to [38, 40, Chap. 5] for details regarding the implementation in zgoubi.*



## 5 ORBIT CORRECTION

Orbit correction in simple cases can be attempted using the *FIT[2]* procedure.

However, **zgoubi** provides an SVD orbit correction method. It is activate by the keyword *SVDOC*. Details can be found in Sec. 6.4, p. 191.

*TO BE DOCUMENTED FURTHER.*





## 6 WHAT KEYWORDS DO, AND OTHER AVAILABLE PROCEDURES

### 6.1 Introduction

This chapter gives an inventory of the procedures available in **zgoubi**, their associated “keyword”, and a brief description of the way they function.

The chapter has been split into several sections. Sections 6.2 to 6.5 explain the underlying content - physics and numerical methods - behind the keywords, they are organized by topics :

- How to defined an object (a set of initial coordinates),
- Available options,
- Optical elements and procedures,
- Output procedures.

Section 7 addresses further a series of functionalities that may be accessed by means of special input data or flags.



## 6.2 Definition of an Object

The description of the object, *i.e.*, initial coordinates of the ensemble of particles, must be the first procedure in the **zgoubi** input data file, `zgoubi.dat`.

Several types of automatically generated objects are available, they are described in the following pages and include,

- non-random object, with various distributions : individual particles, grids, object for *MATRIX*, etc.
- Monte Carlo distribution, with various distributions as well : 6-D window, ellipsoids, etc.

A recurrent quantity appearing in these procedures is *IMAX*, the number of particles to be ray-traced. The maximum value allowed for *IMAX* can be changed at leisure in the include file '`MAXTRA.H`' where it is defined (that requires re-compiling **zgoubi**).

**MCOBJET : Monte-Carlo generation of a 6-D object**

(page 304 in PART B)

*MCOBJET* generates a set of *IMAX* random 6-D initial conditions (the maximum value for *IMAX* is defined in the include file 'MAXTRA.H'). It can be used in conjunction with the keyword *REBELOTE* which either allows generating an arbitrarily high number of initial conditions, or, in the hypothesis of a periodic structure, allows multi-turn tracking with initial conditions at pass number *IPASS* identical to the conditions at end of pass number *IPASS*-1.

The first datum in *MCOBJET* is the reference rigidity (a negative value is allowed)

$$BORO = \frac{p_0}{q} \text{ (kG.cm)}$$

Depending on the value of the next datum, *KOBJ*, the *IMAX* particles have their initial random conditions *Y*, *T*, *Z*, *P*, *X* and *D* (relative rigidity,  $B\rho/BORO$ ) generated on 3 different types of supports, as described below.

Next come the data

$$KY, KT, KZ, KP, KX, KD$$

that specify the type of probability density for the 6 coordinates.

*KY*, *KT*, *KZ*, *KP*, *KX* can take the following values :

1. uniform density,  $p(x) = 1/2\delta x$  if  $-\delta x \leq x \leq \delta x$ ,  $p(x) = 0$  elsewhere,

2. Gaussian density,  $p(x) = \frac{1}{\delta x \sqrt{2\pi}} e^{-\frac{x^2}{2\delta x^2}}$ ,

3. parabolic density,  $p(x) = \frac{3}{4\delta x} (1 - \frac{x^2}{\delta x^2})$  if  $-\delta x \leq x \leq \delta x$ ,  $p(x) = 0$  elsewhere.

*KD* can take the following values :

1. uniform density,  $p(D) = 1/2\delta D$  if  $-\delta D \leq D \leq \delta D$ ,  $p(D) = 0$  elsewhere,
2. exponential density,  $p(D) = N_0 \exp(C_0 + C_1 l + C_2 l^2 + C_3 l^3)$  with  $0 \leq l \leq 1$  and  $-\delta D \leq D \leq \delta D$ ,
3.  $p(D)$  is determined by a kinematic relation, namely, with  $T =$  horizontal angle,  $D = \delta D \times T$ .

Next come the central values for the random sorting,

$$Y_0, T_0, Z_0, P_0, X_0, D_0$$

namely, the probability density laws  $p(x)$  ( $x = Y, T, Z, P$  or  $X$ ) and  $p(D)$  described above apply to the variables  $x - x_0$  ( $\equiv Y - Y_0, T - T_0, \dots$ ) and  $D - D_0$  respectively. Negative value for  $D_0$  is allowed (see section 7.14, page 220).

**KOBJ = 1** : Random generation of *IMAX* particles in a hyper-window with widths (namely the half-extent for uniform or parabolic distributions (*KY*, *KT*, ... = 1 or 3), and the r.m.s. width for Gaussian distributions (*KY*, *KT*, ... = 2))

$$\delta Y, \delta T, \delta Z, \delta P, \delta X, \delta D$$

Then follow the cut-off values, in units of the r.m.s. widths  $\delta Y, \delta T, \dots$  (used only for Gaussian distributions, *KY*, *KT*, ... = 2)

$$N_{\delta Y}, N_{\delta T}, N_{\delta Z}, N_{\delta P}, N_{\delta X}, N_{\delta D}$$

The last data are the parameters

$$N_0, C_0, C_1, C_2, C_3$$

needed for generation of the *D* coordinate upon option *KD* = 2 (unused if *KD* = 1, 3) and a set of three integer seeds for initialization of random sequences,

$$IR1, \quad IR2, \quad IR3 \quad (\text{all} \simeq 10^6)$$

All particles generated by *MCOBJET* are tagged with a (non-S) character, for further statistic purposes (*e.g.*, with *HISTO*, *MCDESINT*).

**KOBJ = 2** : Random generation of  $IMAX = IY \times IT \times IZ \times IP \times IX \times ID$  particles on a hyper-grid. The input data are the number of bars in each coordinate

$$IY, \quad IT, \quad IZ, \quad IP, \quad IX, \quad ID$$

the spacing of the bars

$$PY, \quad PT, \quad PZ, \quad PP, \quad PX, \quad PD$$

the width of each bar

$$\delta Y, \quad \delta T, \quad \delta Z, \quad \delta P, \quad \delta X, \quad \delta D$$

the cut-offs, used with Gaussian densities (in units of the r.m.s. widths)

$$N_{\delta Y}, \quad N_{\delta T}, \quad N_{\delta Z}, \quad N_{\delta P}, \quad N_{\delta X}, \quad N_{\delta D}$$

This is illustrated in Fig. 8.

The last two sets of data in this option are the parameters

$$N_0, \quad C_0, \quad C_1, \quad C_2, \quad C_3$$

needed for generation of the  $D$  coordinate upon option  $KD=2$  (unused if  $KD=1, 3$ ) and a set of three integer seeds for initialization of random sequences,  $IR1$ ,  $IR2$ , and  $IR3$  (all  $\simeq 10^6$ ).

All particles generated by *MCOBJET* are tagged with a (non-S) character, for further statistic purposes (see *HISTO* and *MCDESINT*).

**KOBJ = 3** : Distribution of  $IMAX$  particles in a 6-D ellipsoid (with possible momentum correlation via dispersion) defined by the three sets of data (one set per 2-D phase-space)

$$\begin{aligned} \alpha_Y, \quad \beta_Y, \quad \frac{\varepsilon_Y}{\pi}, \quad N_{\varepsilon_Y} \quad [ \quad N'_{\varepsilon_Y}, \quad \text{if } N_{\varepsilon_Y} < 0 \quad ] \quad [ \quad D_Y, \quad D_T \quad ] \\ \alpha_Z, \quad \beta_Z, \quad \frac{\varepsilon_Z}{\pi}, \quad N_{\varepsilon_Z} \quad [ \quad N'_{\varepsilon_Z}, \quad \text{if } N_{\varepsilon_Z} < 0 \quad ] \quad [ \quad D_Z, \quad D_P \quad ] \\ \alpha_X, \quad \beta_X, \quad \frac{\varepsilon_X}{\pi}, \quad N_{\varepsilon_X} \quad [ \quad N'_{\varepsilon_X}, \quad \text{if } N_{\varepsilon_X} < 0 \quad ], \end{aligned}$$

where  $\alpha_{X,Y,Z}$ ,  $\beta_{X,Y,Z}$ ,  $D_{Y,T}$ ,  $D_{Z,P}$  are the ellipse parameters and the dispersions and derivatives,  $\varepsilon_{X,Y,Z}/\pi$  are the *rms* emittances. This defines for each plane an elliptical frontier :

$$\frac{1 + \alpha_Y^2}{\beta_Y} Y^2 + 2\alpha_Y Y T + \beta_Y T^2 = \varepsilon_Y / \pi,$$

$$\frac{1 + \alpha_Z^2}{\beta_Z} Z^2 + 2\alpha_Z Z P + \beta_Z P^2 = \varepsilon_Z / \pi,$$

$$\frac{1 + \alpha_l^2}{\beta_l} l^2 + 2\alpha_l l \delta + \beta_l \delta^2 = \varepsilon_l / \pi,$$

with  $l$  the bunch length variable (unit is cm) and  $\delta$  a short notation for  $\delta p/p$ , momentum spread.

Sorting cut-offs  $N_{\varepsilon_Y}$ ,  $N_{\varepsilon_Z}$  and  $N_{\varepsilon_X}$  are accounted for if distributions are Gaussian ( $KY, KT, \dots = 2$ ); in this case *rms* values are

$$\sigma_Y = \sqrt{\beta_Y \times \varepsilon_Y / \pi} \text{ and } \sigma_T = \sqrt{\gamma_Y \times \varepsilon_Y / \pi},$$

$$\sigma_Z = \sqrt{\beta_Z \times \varepsilon_Z / \pi} \text{ and } \sigma_P = \sqrt{\gamma_Z \times \varepsilon_Z / \pi},$$

$$\sigma_l = \sqrt{\beta_l \times \varepsilon_l / \pi} \text{ (bunch length, unit cm), and } \sigma_\delta = \sqrt{\gamma_l \times \varepsilon_l / \pi} \text{ momentum spread.}$$

For uniform sorting, the distribution extends over the intervals, for respectively the radial, axial and longitudinal distribution,

$$\pm \sqrt{\varepsilon_Y / \pi \times \beta_Y} \text{ and } \pm \sqrt{\varepsilon_Y / \pi \times \gamma_Y},$$

$$\pm \sqrt{\varepsilon_Z / \pi \times \beta_Z} \text{ and } \pm \sqrt{\varepsilon_Z / \pi \times \gamma_Z},$$

$$\pm \sqrt{\varepsilon_l / \pi \times \beta_l} \text{ and } \pm \sqrt{\varepsilon_l / \pi \times \gamma_l}.$$

$N_\varepsilon$  may be negative. The meaning of this is for the sorting to fill up an elliptical ring that extends from  $|N_\varepsilon|$  to  $N'_\varepsilon$  (rather than the inner region determined by the  $N_\varepsilon$  cut-off as discussed above, when  $N_\varepsilon > 0$ ).

Random seeds are part of *MCOBJET* data (last line); however, in order for different runs to produce different series (meaning, overriding these seeds) *zgoubi* can be launched using 'irank' argument: [pathTo]/*zgoubi* -irank someNonZeroInteger -in mySequence.dat, or [pathTo]/*zgoubi* -irank someNonZeroInteger (input file is *zgoubi.dat*) ... The random seed is then generated from the date and time<sup>8</sup>.

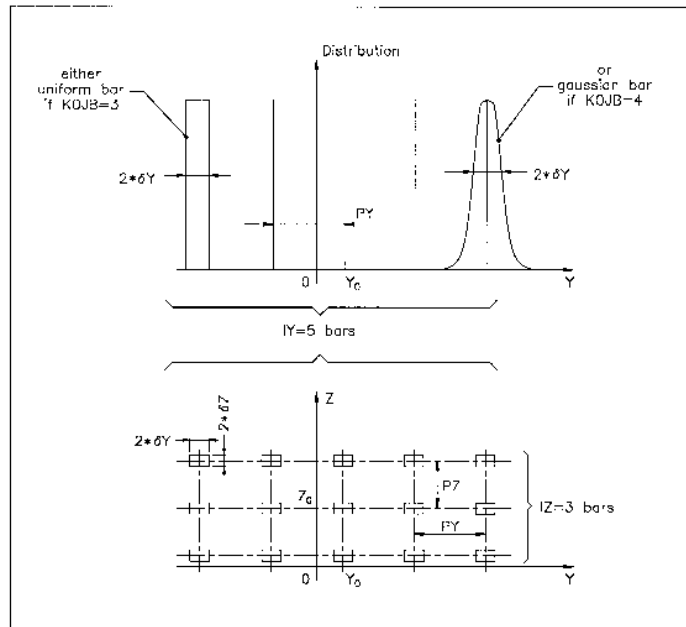


Figure 8: A scheme of input parameters to *MCOBJET* when  $KOBJ= 2$ .  
 Top : Possible distributions of the  $Y$  coordinate.  
 Bottom : A 2-D grid in  $(Y, Z)$  space.

<sup>8</sup>For the geek: rank is managed in *zgoubi\_main* which passes the instruction to *mcobj3.f* (via the latter's ENTRY *MCOBJB*); *mcobj3* does the 'CALL DATE\_AND\_TIME'.

**OBJET : Generation of an object**

(page 308 in PART B)

*OBJET* is dedicated to the construction of sets of initial coordinates, in several ways.

The first datum is the reference rigidity (a negative value is allowed)

$$BORO = \frac{p_0}{q}$$

At the object, the beam is defined by a set of *IMAX* particles (the maximum value for *IMAX* is defined in the include file 'MAXTRA.H') with the initial conditions (*Y, T, Z, P, X, D*) with  $D = B\rho/BORO$  the relative rigidity.

Depending on the value of the next datum *KOBJ*, these initial conditions may be generated in different ways, as follows.

**KOBJ = 1** : Defines a grid in the *Y, T, Z, P, X, D* space. One defines the number of points desired

$$IY, IT, IZ, IP, IX, ID$$

with  $IY \leq n_Y \dots ID \leq n_D$  such that  $n_Y \times n_T \times \dots \times n_D \leq \max(IMAX)$ . One defines the sampling range in each coordinate

$$dY, dT, dZ, dP, dX, dD$$

**zgoubi** then generates  $IMAX = IY \times IT \times IZ \times IP \times IX \times ID$  particles with initial coordinates

$$\begin{aligned} 0, & \pm dY, \pm 2 \times dY, \dots, \pm IY/2 \times dY, \\ 0, & \pm dT, \pm 2 \times dT, \dots, \pm IT/2 \times dT, \\ 0, & \pm dZ, \pm 2 \times dZ, \dots, \pm IZ/2 \times dZ, \\ 0, & \pm dP, \pm 2 \times dP, \dots, \pm IP/2 \times dP, \\ 0, & \pm dX, \pm 2 \times dX, \dots, \pm IX/2 \times dX, \\ 0, & \pm dD, \pm 2 \times dD, \dots, \pm ID/2 \times dD, \end{aligned}$$

In this option relative rigidities will be classified automatically in view of possible further use of *IMAGES[Z]* for momentum analysis and image formation.

The particles are tagged with an index *IREP* possibly indicating a symmetry with respect to the (*X,Y*) plane, as explained in option *KOBJ= 3*. If two trajectories have mid-plane symmetry, only one will be ray-traced, while the other will be deduced using the mid-plane symmetries. This is done for the purpose of saving computing time. It may be non-compatible with the use of some procedures (*e.g. MCDESINT*, which involves random processes) or with the use of non mid-plane symmetric fields.

The last datum is a reference in each coordinate, *YR, TR, ZR, PR, XR, DR*. For instance the reference rigidity is  $DR \times BORO$ , resulting in the rigidity of a particle defined by  $I \times dD$  to be  $(DR + I \times dD) \times BORO$ .

**KOBJ = 1.1** : The initial *Z* and *P* conditions are

$$\begin{aligned} 0, & dZ, 2 \times dZ, \dots, (IZ - 1) \times dZ, \\ 0, & dP, 2 \times dP, \dots, (IP - 1) \times dP, \end{aligned}$$

This object results in shorter outputs/CPU-time when studying problems with *Z* symmetry.

**KOBJ = 1.2** : Same as *KOBJ= 1*, but the mid-plane symmetry of the tracking is lifted, all particles are actually tracked.

**KOBJ = 2** : Next data : *IMAX, IDMAX*. Initial coordinates are entered explicitly for each trajectory. *IMAX* is the total number of particles. These may be classified in groups of equal number for each value of momentum, in order to fulfill the requirements of image calculations by *IMAGES[Z]*. *IDMAX* is the number of groups of momenta. The following initial conditions defining a particle are specified for each one of the *IMAX* particles

$$Y, T, Z, P, X, D, 'A'$$

where  $D \times BORO$  is the rigidity (negative value allowed) and '*A*' is a (arbitrary) tagging character.

The last record *IEX* ( $I=1$ , *IMAX*) contains *IMAX* times either the character “1” to indicates that the particle has to be tracked, or “-9” to indicates that the particle should not be tracked.

This option *KOBJ*= 2 may be be useful for the definition of objects including kinematic effects.

**KOBJ = 2.1** : Same as *KOBJ*= 2 except for the units, meter and radian in that case.

Note that in this option **KOBJ = 2[.]**, only the first 7 trajectories can possibly have their coodinates varied when using the *FIT[2]* procedure.

**KOBJ = 3** : This option allows the reading of initial conditions, including spin coordinates, from an external input file *FNAME*.

The next three data lines are :

```
IT1, IT2, ITStep
IP1, IP2, IPStep
YF, TF, ZF, PF, SF, DPF, TiF, TAG
YR, TR, ZR, PR, SR, DPR, TiR
InitC
```

followed by the storage file name *FNAME*.

*IT1*, *IT2*, *ITStep* cause the code to read coordinates of particles number *IT1* through *IT2* by step *ITStep*.

*IP1*, *IP2*, *IPStep* cause the code to read coordinates belonging in the passes range *IP1* through *IP2*, step *IPStep*.

*YF*, *TF*, *ZF*, *PF*, *SF*, *DPF*, *TiF* are scaling factors whereas *YR*, *TR*, *ZR*, *PR*, *SR*, *DPR*, *TiR* are references added to the values of respectively Y, T, Z, P, S, DP as read in file *FNAME*, so that any coordinate  $C = Y, T, Z...$  is changed into  $CF * C + CR$ . In addition a flag character TAG allows retaining only particles with identical tagging letter LET, unless TAG='\*' in which case it has no selection effect - for instance TAG='S' can be used to retain only secondary particles following in-flight decay simulations.

If *InitC*= 1 ray-tracing starts from the current coordinates  $F(J, I)$ ,

if *InitC*= 0 ray-tracing starts from the initial coordinates  $FO(J, I)$ , as read from file *FNAME*.

The file *FNAME* must be formatted in the same way as a *zgoubi.fai* ouptput file, following the *FORTTRAN* sequence below - details and possible updates are to be found in the source file *obj3.f* :

```
OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD')
DO I = 1, IMAX
  READ (NL, 100)
  1 IEX(I), -1.D0+FO(1, I), (FO(J, I), J=2, MXJ),
  2 -1.D0+F(1, I), F(2, I), F(3, I),
  3 (F(J, I), J=4, MXJ),
  4 (SI(J, I), J=1, 4), (SF(J, I), J=1, 4),
  5 ENKI, ENERG,
  6 I, IREP(I), SORT(I), (AMQ(J, I), J=1, 5), PH(I), DPR(I), PS,
  7 BORO, IPASS, NOEL,
  8 KLEY, LBL1C, LBL2C, LETC,
  9 SRLT(I), DPREF, HDPRF, DMY, HMS,
  1 NI, MXITR, NBFT, KBYD, LAST
ENDDO
```

The meaning of some of the parameters above is the following (see Sec. 8.2 for the detailed content of a *zgoubi.fai* style of file) :

<i>IEX(I)</i>	: flag, see <i>KOBJ</i> = 2 and page 220
<i>FO(1-6,I)</i>	: coordinates <i>D</i> , <i>Y</i> , <i>T</i> , <i>Z</i> , <i>P</i> and path length of particle number <i>I</i> , at the origin. $D \times BORO = \text{rigidity}$
<i>F(1-6,I)</i>	: <i>idem</i> , at the current position
<i>(SI(J,I),J=1,4),(SF(J,I),J=1,4)</i>	: initial and current spin coordinates
... etc.	:
<i>LETC</i>	: one-character string (for tagging)
... etc.	:



*IREP* is an index which indicates a symmetry with respect to the median plane. For instance, if  $Z(I+1) = -Z(I)$ , then normally  $IREP(I+1) = IREP(I)$ . Consequently the coordinates of particle  $I+1$  will not be obtained from ray-tracing but instead deduced from those of particle  $I$  by simple symmetry. This saves on computing time.

*KOBJ*= 3 can be used for reading files filled by *FAISCNL* or *FAISTORE*.  
If more than *MaxIMAX*<sup>9</sup> particles are to be read from a file, use *REBELOTE*.

Note : In this option, one has to make sure that input data do not conflict with possible use of the keyword *PARTICUL* that assigns mass and charge.

**KOBJ = 3.1** : Same as **KOBJ = 3**, except for the formatting of trajectory coordinate data in *FNAME*, namely, according to the following *FORTTRAN* sequence

```

      OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD')
1     CONTINUE
      READ (NL, *, END=10, ERR=99) Y, T, Z, P, S, D
      GOTO 1
10    CALL ENDFIL
99    CALL ERREAD

```

**KOBJ = 3.2 to 3.6** : As for **KOBJ=3.1**, except for a different format, as follows:

3.2: READ(NL,\*) X, Y, Z, PX, PY, PZ

with PX, PY, and PZ, the momenta in MeV/c.

3.3: READ(NL,\*) DP, Y, T, Z, P, S, TIME, MASS, CHARGE

with MASS the mass in MeV/c<sup>2</sup> and CHARGE the charge in units of the elementary charge.

3.4: READ(NL,\*) Y, T, Z, P, S, DP, LET, IEN, IT

with LET a tag letter, IEN an integer, IT the particle number.

3.5: READ(NL, \*, ERR=97, END=95) Y, T, Z, P, TIME, GAMMA

with TIME the particle time and GAMMA its Lorentz relativistic factor.

3.6: READ(NL,\*) Y, Z, X, PY, PZ, PX

similar to 3.2, just a different ordering of the coordinates.

**Note 1** : Conversion from the input file to proper units (namely, Y, Z in cm, T, P in mrad, path variable in cm, time in  $\mu$ s), can be performed using the scaling factors *YF*, *YF*, etc. (see above).

**Note 2** : For details and possible updates in the formatted read of concern in the *FORTTRAN*, regarding options 3.1-3.5 and possibly beyond, see the source file *obj3.f*.

#### RECOVERING FROM A CRASH :

*OBJET* with option *KOBJ*=3 can be used to recover from a job crash, which would occur when multi-turn tracking - due to computer crash or else. This is discussed in the *SCALING* section, page 183, an example of job resuming is given page 310.

**KOBJ = 5** : Mostly dedicated to the calculation of first order transport matrix and various other optical parameters, using for instance *MATRIX* or *TWISS*. The user defines the coordinate sampling

$$dY, dT, dZ, dP, dS, dD$$

These values should be small enough, so that the paraxial ray approximation be valid. The code generates 13 particles, with initial coordinates

$$0, YR \pm dY, TR \pm dT, ZR \pm dZ, PR \pm dP, SR \pm dS, DR \pm dD$$

given the reference trajectory

$$YR, TR, ZR, PR, DR$$

(with  $DR \times BORO$  the reference rigidity - negative value allowed).

<sup>9</sup>Max(IMAX) here stands for the maximum number of particles allowed in one go. The reason for a limitation is the dimensioning of arrays in the source code: Max(IMAX) is controlled, and can be changed, in the include file [pathTo]/include/MAXTRA.H.

**KOBJ = 5.1** : Same as  $KOBJ = 5$ , except for an additional data line giving initial beam ellipse parameters and dispersions,  $\alpha_Y, \beta_Y, \alpha_Z, \beta_Z, \alpha_X, \beta_X, D_Y, D'_Y, D_Z, D'_Z$ , for further transport of these using *MATRIX* or *OPTICS*, or for possible use by the *FIT[2]* procedure.

**KOBJ = 5.N** ( $N \geq 2$ ) : Like  $KOBJ = 5$  except for its allowing  $N \geq 2$  reference trajectories, rather than just one as in  $KOBJ = 5$  case (thus  $N-1$  additional input data lines are needed in this case). These reference trajectories can be for instance chromatic orbits. Zgoubi will generate  $N$  sets of 13 particles with initial coordinates in each set taken wrt. one of the  $N$  references.

A subsequent use of *MATRIX* would then cause the computation of  $N$  transport matrices.

Note that in the option  $KOBJ=5.N, N \geq 2$ , only the first 7 reference trajectories can possibly have their coordinates varied when using the *FIT[2]* procedure.

**KOBJ = 6** : Dedicated to the calculation of first and second order transport matrices, and some additional higher order transport coefficients, using for instance *MATRIX*. The input data are a reference trajectory and a coordinate sampling with respect to that reference (normally taken paraxial)

$$dY, dT, dZ, dP, dD$$

to allow the building up of an object containing 61 particles (note : their coordinates can be checked by printing out into zgoubi.res using *FAISCEAU*), whereas a last data line gives the reference

$$YR, TR, ZR, PR, DR$$

(with  $DR \times BORO$  the reference rigidity - negative value allowed), which adds to the previous coordinate values.

**KOBJ = 6.1** : Dedicated to the calculation of first, second and third order transport matrices, using for instance *MATRIX*. It builds fort that a 102 particle initial object. The sampling and functioning are similar to the 61 particle ( $KOBJ=6$ ) and 13 particle ( $KOBJ=5$ ) object cases.

**KOBJ = 7** : Object with kinematics

The data and functioning are the same as for  $KOBJ=1$ , except for the following

- $ID$  is not used,
- $PD$  is the kinematic coefficient, such that for particle number  $I$ , the initial relative rigidity  $D_I$  is calculated from the initial angle  $T_I$  following

$$D_I = DR + PD \times T_I$$

while  $T_I$  is in the range

$$0, \pm PT, \pm 2 \times PT, \dots, \pm IT/2 \times PT$$

as when using  $KOBJ=1$ .

**KOBJ = 8** : Generation of phase-space coordinates on ellipses.

The ellipses are defined by the three sets of data (one set per ellipse)

$$\begin{aligned} \alpha_Y, \beta_Y, \varepsilon_Y/\pi \\ \alpha_Z, \beta_Z, \varepsilon_Z/\pi \\ \alpha_X, \beta_X, \varepsilon_X/\pi \end{aligned}$$

where  $\alpha, \beta$  are the ellipse parameters and  $\varepsilon/$  is the ellipse surface, corresponding to an ellipse with equation

$$\frac{1 + \alpha_Y^2}{\beta_Y} Y^2 + 2\alpha_Y Y T + \beta_Y T^2 = \varepsilon_Y/\pi$$

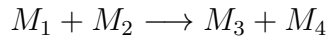
(idem for the  $(Z, P)$  or  $(X, D)$  planes).

The ellipses are centered respectively on  $(Y_0, T_0), (Z_0, P_0), (X_0, D_0)$ .

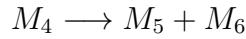
The number of samples per plane is respectively  $IX, IY, IZ$ . If that value is zero, the central value above is assigned.

**OBJETA : Object from Monte-Carlo simulation of decay reaction**

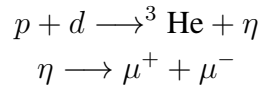
This generator simulates the reactions[41]



and then



where  $M_1$  is the mass of the incoming body ;  $M_2$  is the mass of the target ;  $M_3$  is an outgoing body ;  $M_4$  is the rest mass of the decaying body ;  $M_5$  and  $M_6$  are decay products. Example :



The first input data are the reference rigidity

$$BORO = p_0/q$$

an index *IBODY* which specifies the particle to be ray-traced, namely M3 (*IBODY* = 1), M5 (*IBODY* = 2) or M6 (*IBODY* = 3). In this last case, initial conditions for M6 must be generated by a first run of *OBJETA* with *IBODY* = 2 ; they are then stored in a buffer array, and restored as initial conditions at the next occurrence of *OBJETA* with *IBODY* = 3. Note that **zgoubi** by default assumes positively charged particles.

Another index, *KOBJ*, specifies the type of distribution for the initial transverse coordinates *Y*, *Z* ; namely either uniform (*KOBJ*= 1) or Gaussian (*KOBJ*= 2). The other three coordinates *T*, *P* and *D* are deduced from the kinematic of the reactions.

The next data are the number of particles to be generated, *IMAX*, the masses involved in the two previous reactions.

$$M_1, \quad M_2, \quad M_3, \quad M_4, \quad M_5, \quad M_6$$

and the kinetic energy  $T_1$  of the incoming body ( $M_1$ ).

Then one gives the central value of the distribution for each coordinate

$$Y_0, \quad T_0, \quad Z_0, \quad P_0, \quad D_0$$

and the width of the distribution around the central value

$$\delta Y, \quad \delta T, \quad \delta Z, \quad \delta P, \quad \delta D$$

so that only those particles in the range

$$Y_0 - \delta Y \leq Y \leq Y_0 + \delta Y \quad \dots \quad D_0 - \delta D \leq D \leq D_0 + \delta D$$

will be retained. The longitudinal initial coordinate is uniformly sorted in the range

$$-XL \leq X_0 \leq XL$$

The random sequences involved may be initialized with different values of the two integer seeds  $IR_1$  and  $IR_2$  ( $\simeq 10^6$ ).

Possible use of *PARTICUL* will have no effect : it will not change the mass and charge assumptions as set by *OBJETA*.



### 6.3 Optical Elements and Related Numerical Procedures

#### AGSMM : AGS main magnet

(page 237 in PART B)

The AGS main magnet is a combined function dipole with straight axis (lines of constant field are straight lines).

The field computation routines for *AGSMM* are the same as for *MULTIPOL* (details in section 1.3.7, page 30), however *AGSMM* has the following four particularities :

- There are only three multipole components present in *AGSMM* : dipole, quadrupole and sextupole.
- The dipole field  $B_0$  is drawn from the reference rigidity,  $B\rho_{ref}$ , and follows the latter so to preserve  $\rho = B\rho_{ref}/B_0$  and the orbit deviation  $L/\rho$ . In particular,
  - in the absence of acceleration,  $B\rho_{ref} \equiv BORO$ , with *BORO* the quantity appearing in the object definition using *[MC]OBJET*,
  - in presence of acceleration using *CAVITE*,  $B\rho_{ref}$  is changed to  $BORO \times D_{ref}$  at each passage in the cavity, with  $D_{ref}$  the relative synchronous momentum increase, a quantity that **zgoubi** updates at cavity traversal.
- The field indices, quadrupole  $K1$  and sextupole  $K2$ , are derived from the reference rigidity,  $B\rho_{ref}$ , via momentum-dependent polynomials, taken from Ref. [46], and following in that the methods found in the MAD model of the AGS [47].
- The AGS main dipole has back-leg windings, used for instance for injection and extraction orbit bumps. The number of winding turns and the number of ampere-turns are part of the data in the keyword argument list. The intensity in the windings is accounted for in the conversion from total ampere-turns in the magnet to momentum and then to magnetic field.

Note : A consequence of items 2 and 3 above is that no field value is required in defining the AGS main magnets using *AGSMM*.

**AGSQUAD : AGS quadrupole**

The AGS quadrupoles are regular quadrupoles. The simulation of *AGSQUAD* uses the same field modelling as *MULTIPOL*, section 1.3.7, page 30. However amperes are provided as input to *AGSQUAD* rather than fields, the reason being that some of the AGS quadrupoles have two superimposed coil circuits, with separate power supplies. It has been dealt with this particularity by allowing for an additional set of quadrupole data in *AGSQUAD*, compared to *MULTIPOL*.

The field in *AGSQUAD* is computed using transfer functions, from the specified ampere-turns in the coils to magnetic field at pole-tip, that account for the non-linearity of the magnetic permeability [47].

**AIMANT : Generation of dipole mid-plane 2-D map, polar frame**

The keyword *AIMANT* provides an automatic generation of a dipole median plane field map in polar coordinates.

A more recent and improved version will be found in *DIPOLE-M*. In addition, a similar modelling, that however skips the stage of an intermediate mid-plane field map, can be found in *DIPOLE[S]*.

The extent of the map is defined by the following parameters, as shown in Figs. 9A and 9B,

- $AT$  : total angular aperture
- $RM$  : mean radius used for the positioning of field boundaries
- $RMIN, RMAX$  : minimum and maximum radial boundaries of the map

The 2 or 3 effective field boundaries (EFB) inside the map are defined from geometric boundaries, the shape and position of which are determined by the following parameters,

- $ACENT$  : arbitrary angle, used for the positioning of the EFBs.
- $\omega$  : azimuth of an EFB with respect to  $ACENT$
- $\theta$  : angle of a boundary with respect to its azimuth (wedge angle)
- $R_1, R_2$  : radius of curvature of an EFB
- $U_1, U_2$  : extent of the linear part of the EFB.

At any node of the map mesh, the value of the  $Z$  component of the field is calculated as

$$B_Z = \mathcal{F}(R, \theta) \times B_0 \times \left( 1 + N \times \left( \frac{R - RM}{RM} \right) + B \times \left( \frac{R - RM}{RM} \right)^2 + G \times \left( \frac{R - RM}{RM} \right)^3 \right) \quad (6.3.1)$$

where  $N$ ,  $B$  and  $G$  are respectively the first, second and third order field indices and  $\mathcal{F}(R, \theta)$  is the fringe field coefficient (it determines the “flutter” in periodic structures).

**Calculation of the Fringe Field Coefficient**

With each EFB a realistic extent of the fringe field,  $\lambda$ , is associated (Figs. 9A and 9B), and a fringe field coefficient  $F$  is calculated. In the following  $\lambda$  stands for either  $\lambda_E$  (Entrance),  $\lambda_S$  (Exit) or  $\lambda_L$  (Lateral EFB).

If a node of the map mesh is at a distance of the EFB larger than  $\lambda$ , then  $F = 0$  outside the field map and  $F = 1$  inside. If a node is inside the fringe field zone, then  $F$  is calculated as follows.

Two options are available, for the calculation of  $F$ , depending on the value of  $\xi$ .

**If  $\xi \geq 0$** ,  $F$  is a second order type fringe field (Fig. 10) given by

$$F = \frac{1}{2} \frac{(\lambda - s)^2}{\lambda^2 - \xi^2} \quad \text{if } \xi \leq s \leq \lambda \quad (6.3.2)$$

$$F = 1 - \frac{1}{2} \frac{(\lambda - s)^2}{\lambda^2 - \xi^2} \quad \text{if } -\lambda \leq s \leq -\xi \quad (6.3.3)$$

where  $s$  is the distance to the EFB, and

$$F = \frac{1}{2} + \frac{s}{\lambda + \xi} \quad \text{if } 0 \leq s \leq \xi \quad (6.3.4)$$

$$F = \frac{1}{2} - \frac{s}{\lambda + \xi} \quad \text{if } -\xi \leq s \leq 0 \quad (6.3.5)$$

This simple model allows a rapid calculation of the fringe field, but may lead to erratic behavior of the field when extrapolating out of the median plane, due to the discontinuity of  $d^2B/ds^2$ , at  $s = \pm\xi$  and  $s = \pm\lambda$ . For better accuracy it is advised to use the next option.

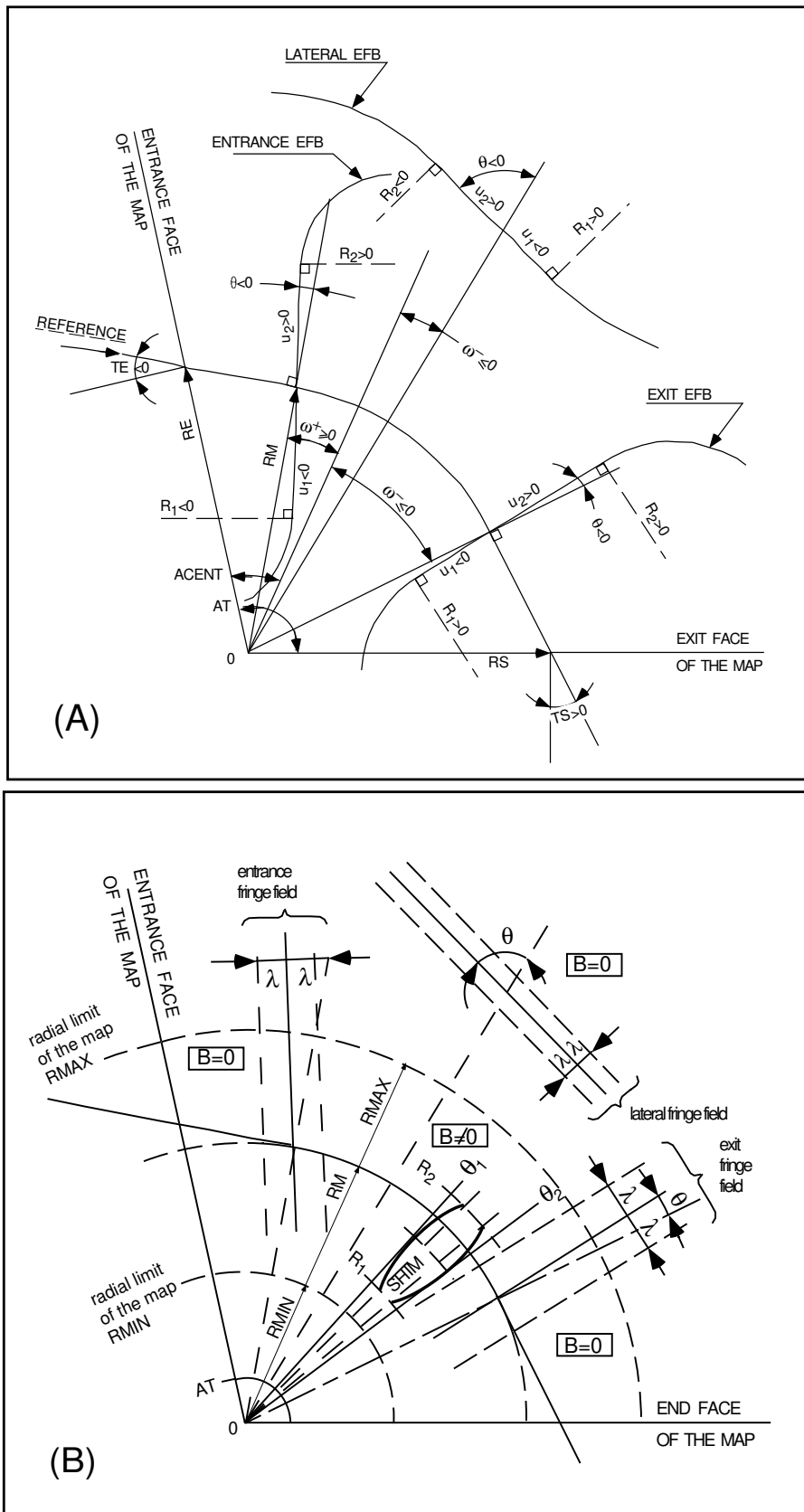


Figure 9: A : Parameters used to define the field map and geometrical boundaries.

B : Parameters used to define the field map and fringe fields.



If  $\xi = -1$ ,  $F$  is an exponential type fringe field (Fig. 10) given by [48]

$$F = \frac{1}{1 + \exp P(s)} \quad (6.3.6)$$

where  $s$  is the distance to the EFB, and

$$P(s) = C_0 + C_1 \left(\frac{s}{\lambda}\right) + C_2 \left(\frac{s}{\lambda}\right)^2 + C_3 \left(\frac{s}{\lambda}\right)^3 + C_4 \left(\frac{s}{\lambda}\right)^4 + C_5 \left(\frac{s}{\lambda}\right)^5 \quad (6.3.7)$$

The values of the coefficients  $C_0$  to  $C_5$  should be such that the derivatives of  $B_Z$  with respect to  $s$  be negligible at a distance  $s$  of a few  $\lambda$  from the EFB, so to allow a safe extrapolation of  $\vec{B}$  out of the median plane.

It is also possible to simulate a shift of the EFB, by giving a non zero value to the parameter *shift*.  $s$  is then changed to  $s - \text{shift}$  in the previous equation. This allows small variations of the total magnetic length.

Let  $F_E$  (respectively  $F_S$ ,  $F_L$ ) be the fringe field coefficient attached to the entrance (respectively exit, lateral) EFB following the equations above. At any node of the map mesh, the resulting value of the fringe field coefficient (eq. 6.3.1) is (Fig. 11)

$$\mathcal{F}(R, \theta) = F_E \times F_S \times F_L$$

In particular,  $F_L \equiv 1$  if no lateral EFB is requested.

#### The Mesh of the Field Map

The magnetic field is calculated at the nodes of a mesh with polar coordinates, in the median plane. The radial step is given by

$$\delta R = \frac{RMAX - RMIN}{IRMAX - 1}$$

and the angular step by

$$\delta\theta = \frac{AT}{IAMAX - 1}$$

where,  $RMIN$  and  $RMAX$  are the lower and upper radial limits of the field map, and  $AT$  is its total angular aperture (Fig. 9B).  $IRMAX$  and  $IAMAX$  are the total number of nodes in the radial and angular directions.

#### Simulating Field Defects and Shims

Once the initial map is calculated, it is possible to perturb it by means of the parameter  $NBS$ , so as to simulate field defects or shims.

If  $NBS = -2$ , the map is globally modified by a perturbation proportional to  $R - R_0$ , where  $R_0$  is an arbitrary radius, with an amplitude  $\Delta B_Z/B_0$ , so that  $B_Z$  at the nodes of the mesh is replaced by

$$B_Z \times \left(1 + \frac{\Delta B_Z}{B_0} \frac{R - R_0}{RMAX - RMIN}\right)$$

If  $NBS = -1$ , the perturbation is proportional to  $\theta - \theta_0$ , and  $B_Z$  is replaced by

$$B_Z \times \left(1 + \frac{\Delta B_Z}{B_0} \frac{\theta - \theta_0}{AT}\right)$$

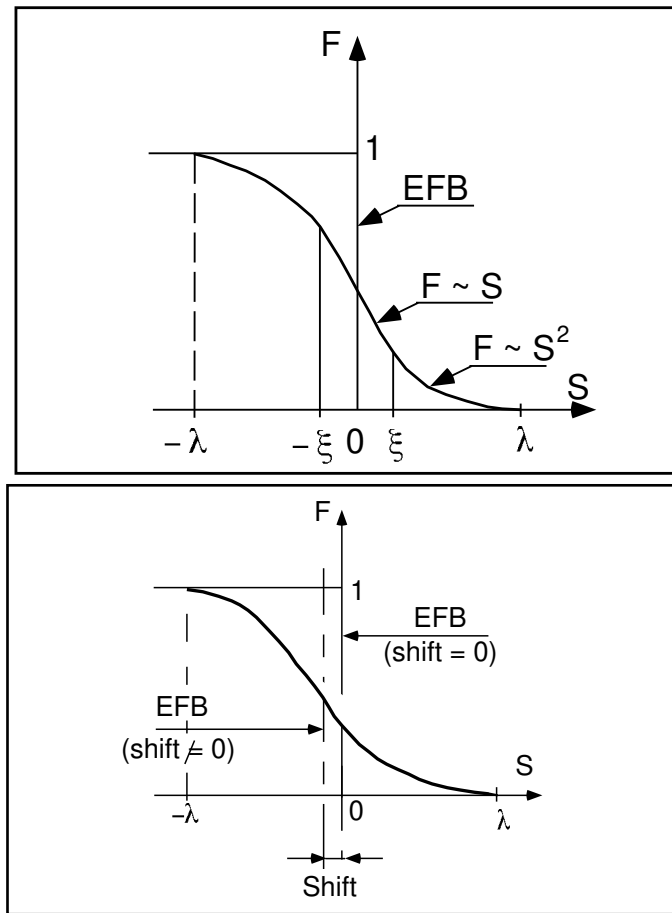


Figure 10: Second order type fringe field (upper plot) and exponential type fringe field (lower plot).

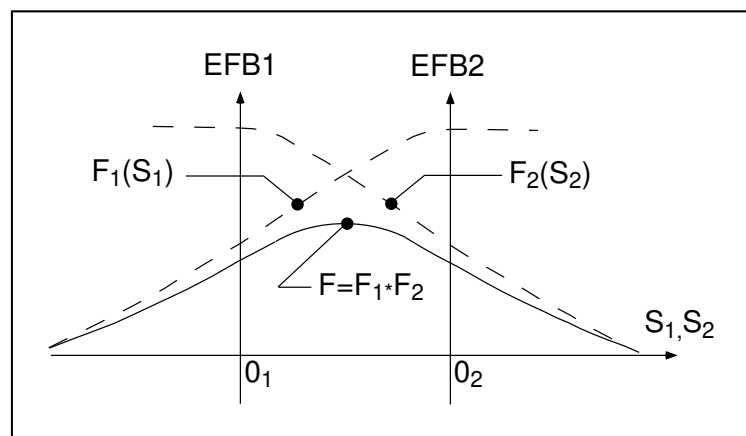


Figure 11: Effective value of  $\mathcal{F}(R, \theta)$  for overlapping fringe fields  $F_1$  and  $F_2$  centered at  $O_1$  and  $O_2$ .

If  $NBS \geq 1$ , then  $NBS$  shims are introduced at positions  $\frac{R_1 + R_2}{2}$ ,  $\frac{\theta_1 + \theta_2}{2}$  (Fig. 12) [49]  
The initial field map is modified by shims with second order profiles given by

$$\theta = \left( \gamma + \frac{\alpha}{\mu} \right) \beta \frac{X^2}{\rho^2}$$

where  $X$  is shown in Fig. 12,  $\rho = \frac{R_1 + R_2}{2}$  is the central radius,  $\alpha$  and  $\gamma$  are the angular limits of the shim,  $\beta$  and  $\mu$  are parameters.

At each shim, the value of  $B_Z$  at any node of the initial map is replaced by

$$B_Z \times \left( 1 + F\theta \times FR \times \frac{\Delta B_Z}{B_0} \right)$$

where  $F\theta = 0$  or  $FR = 0$  outside the shim, and  $F\theta = 1$  and  $FR = 1$  inside.

### Extrapolation Off Median Plane

The vertical field  $\vec{B}$  and its derivatives in the median plane are calculated by means of a second or fourth order polynomial interpolation, depending on the value of the parameter  $IORDRE$  ( $IORDRE=2, 25$  or  $4$ , see section 1.4.2). The transformation from polar to Cartesian coordinates is performed following eqs. (1.4.9 or 1.4.10). Extrapolation off median plane is then performed by means of Taylor expansions following the procedure described in section 1.3.3.

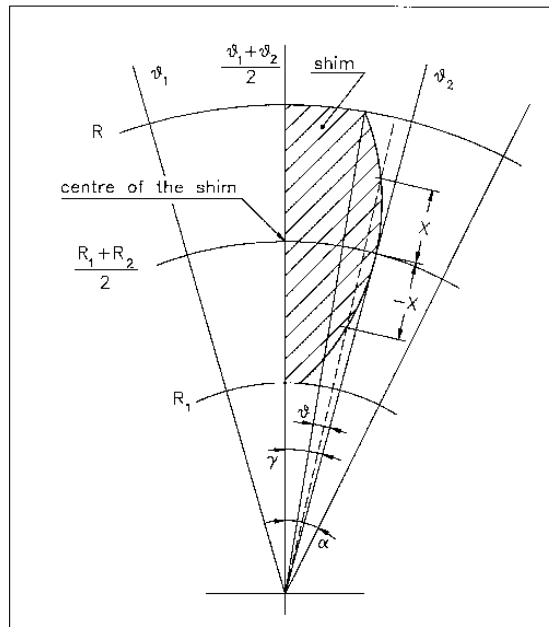


Figure 12: A second order profile shim. The shim is centered at  $\frac{(R_1 + R_2)}{2}$  and  $\frac{(\theta_1 + \theta_2)}{2}$ .

**AUTOREF : Transport beam into a new reference frame**

*AUTOREF* positions the new reference frame following four different options (apart from  $I = 0$  which is just an “off” switch) :

**If  $I = 1$** , *AUTOREF* is equivalent to

$$CHANGREF[XCE = 0, YCE = Y(1), ALE = T(1)]$$

so that the new reference frame is at the exit of the last element, with particle 1 at the origin with its horizontal angle set to  $T = 0$ .

**If  $I = 2$** , it is equivalent to

$$CHANGREF[XW, YW, T(1)]$$

so that the new reference frame is at the position  $(XW, YW)$  of the waist (calculated automatically in the same way as for *IMAGE*) of the three rays number 1, 4 and 5 (compatible for instance with *OBJET*[*KOBJ*=5, 6], together with the use of *MATRIX*) while  $T(1)$ , the horizontal angle of particle number *I1*, is set to zero.

**If  $I = 3$** , it is equivalent to

$$CHANGREF[XW, YW, T(I1)]$$

so that the new reference frame is at the position  $(XW, YW)$  of the waist (calculated automatically in the same way as for *IMAGE*) of the three rays number *I1*, *I2* and *I3* specified as data, while  $T(I1)$  is set to zero.

**If  $I = 4$**  : new horizontal beam centroid positioning *XCE*, *YCE*, *ALE* is provided. The beam is moved by *XCE* and then centered on *YCE*, *ALE*.

**If  $I = 4.1$**  : new beam centroid positioning *XCE*, *YCE*, *ALE*, *DCE*, *TIME* is provided. The beam is moved by *XCE* and then centered on *YCE*, *ALE*. In addition, the beam is centered on a new relative momentum *DCE* and new timing value *TIME*.

**If  $I = 4.2$**  : same as 4.1, except that particles all have their timing set to *TIME*.

**If  $I = 5$**  : new vertical beam centroid positioning *ZCE*, *PLE* (position, angle) is provided. The beam is centered on vertical position and angle *ZCE*, *PLE*.

**BEAMBEAM : Beam-beam lens**

*BEAMBEAM* is a beam-beam lens simulation, a thin-lens transform [50], in the weak-strong approximation.

Upon option using *SPNTRK*, *BEAMBEAM* will include spin kicks, after modelling as described in Ref. [51].

***For software developers***

`rbf.f` reads the data from `zgoubi.dat`; `bb.f` and programs therein do the beam (and spin when requested) dynamics.

**BEND : Bending magnet, Cartesian frame**

(page 245 in PART B)

*BEND* is one of several keywords available for the simulation of dipole magnets. It presents the interest of easy handling, and is well adapted for the simulation of synchrotron dipoles and such other regular dipoles as sector magnets with wedge angles.

The field in *BEND* is defined in a Cartesian coordinate frame (unlike for instance *DIPOLE[S]* that uses a polar frame). As a consequence, having particle coordinates at entrance or exit of the magnet referring to the curved main direction of motion may require using *KPOS*, in particular *KPOS=3* (in a circular accelerator cell for instance, see section 7.10, p. 216).

The dipole simulation accounts for the magnet geometrical length  $XL$ , for a possible skew angle (X-rotation, useful for obtaining vertical deviation magnet), and for the field  $B1$  such that in absence of fringe field the deviation  $\theta$  satisfies  $XL = 2 \frac{BORO}{B1} \sin \frac{\theta}{2}$ .

Then follows the description of the entrance and exit EFBs and fringe fields. The wedge angles  $W_E$  (entrance) and  $W_S$  (exit) are defined with respect to the sector angle, with the signs as described in Fig. 13. Within a distance  $\pm X_E$  ( $\pm X_S$ ) on both sides of the entrance (exit) EFB, the fringe field model is used (same as for *QUADRUPO*, Fig. 36, p. 137) ; elsewhere, the field is supposed to be uniform.

If  $\lambda_E$  (resp.  $\lambda_S$ ) is zero sharp edge field model is assumed at entrance (resp. exit) of the magnet and  $X_E$  (resp.  $X_S$ ) is forced to zero. In this case, the wedge angle vertical first order focusing effect (if  $\vec{B}1$  is non zero) is simulated at magnet entrance and exit by a kick  $P_2 = P_1 - Z_1 \tan(\epsilon/\rho)$  applied to each particle ( $P_1$ ,  $P_2$  are the vertical angles upstream and downstream the EFB,  $Z_1$  the vertical particle position at the EFB,  $\rho$  the local horizontal bending radius and  $\epsilon$  the wedge angle experienced by the particle ;  $\epsilon$  depends on the horizontal angle T).

Magnet (mis-)alignment is assured by *KPOS*. *KPOS* also allows some degrees of automatic alignment useful for periodic structures (section 7.10).

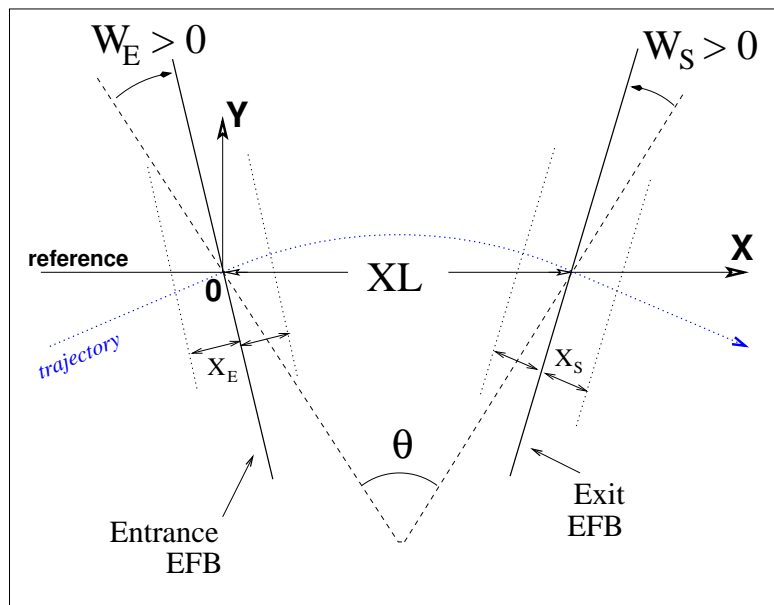


Figure 13: Geometry and parameters of *BEND* :  $XL$  = length,  $\theta$  = deviation,  $W_E$ ,  $W_S$  are the entrance and exit wedge angles. The motion is computed in the Cartesian frame ( $O, X, Y, Z$ ).

**From matrix-style code modeling, to zgoubi :** Fig. 14 illustrates the conversion of matrix method style of data where the magnet is defined by its length and deviation angle (see MAD input below, using 'SBEND'), to **zgoubi** input data using *BEND* (in this particular case of an illustration of a rectangular magnet, *MULTIPOL* could be used, as well).

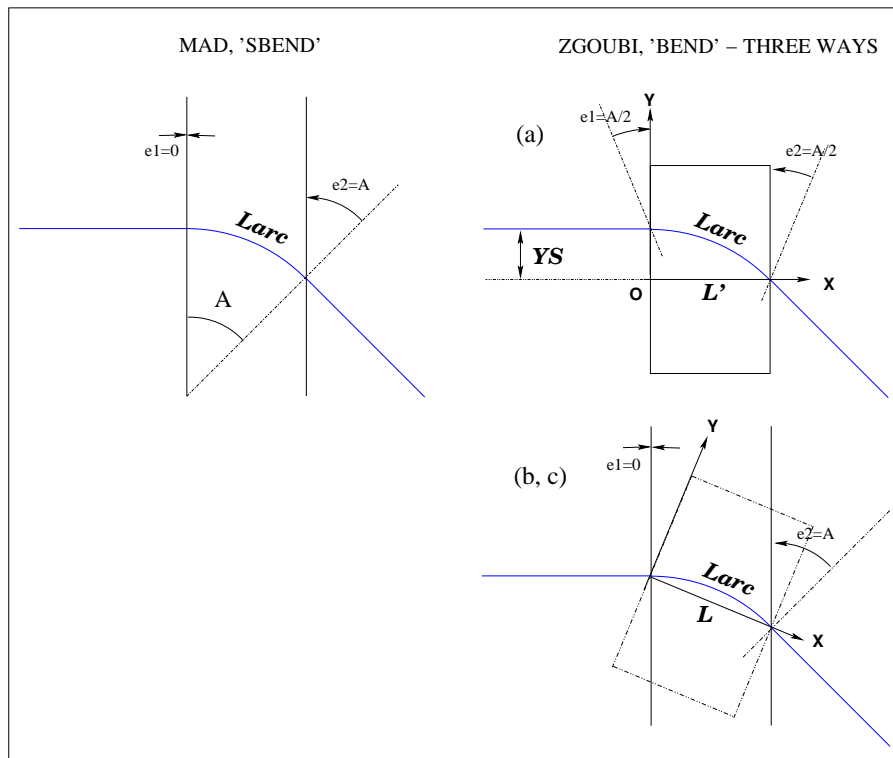


Figure 14: Simulating a matrix method style of rectangular magnet (e.g., MAD's 'SBEND') by means of *BEND*, in three different ways, see text. In case (a), the straight length in **zgoubi**'s *BEND* is  $L' = \rho \sin(A)$ , whereas in cases (b, c) it is  $L = 2\rho \sin(A/2)$ .

• MAD style 'BEND' definition :

```
L = 1.
A = 0.1
Larc = L*(A/2.) / sin(A/2.)
e1 = 0.
e2 = A
B: SBEND, L=Larc, ANGLE=A, e1 = e1, e2 = e2
```

• **zgoubi**, case (a) :

(a)  $YS = -\rho \cdot (1 - \cos(A))$  translated, non-rotated *BEND*, followed by A Z-rotation of frame

```
'CHANGREF'
YS -4.997916927068
'BEND'
1
99.8750260394 0. 1.
0. 0. 0.05 ! e1=A/2
0 0. 0. 0. 0. 0. 0.
0. 0. 0.05 ! e2=A/2
0 0. 0. 0. 0. 0. 0.
.1
1 0. 0. 0.
'CHANGREF'
ZR -5.72957795131
```

• **zgoubi**, case (b) or (c) using  $KPOS=1$  or  $3$  :

(b) A/2 Z-rotated *BEND* followed by A/2 Z-rotation of frame

```
'CHANGREF'
ZR -2.86478897565
'BEND'
1
100. 0. 1.
0. 0. 0. ! e1=0
0 0. 0. 0. 0. 0. 0.
0. 0. 0.1 ! e2=A
0 0. 0. 0. 0. 0. 0.
.1
1 0. 0. 0.
'CHANGREF'
ZR -2.86478897565
```

(c) same as (b), using  $KPOS=3$  instead

```
'BEND'
1
100. 0. 1.
0. 0. 0. ! e1=0
0 0. 0. 0. 0. 0. 0.
0. 0. 0.1 ! e2=A
0 0. 0. 0. 0. 0. 0.
.1
3 0. 0. -0.05
```

**Negative bend, vertical bend, tricks :** Use *YMY* for the former, *TRAROT* with a  $\pm\pi$  X-rotation for the latter. The two can be combined, so that a vertical negative bend can be represented by the sequence *TRAROT*[ $\pi$ ], *YMY*, *BEND*[ $B > 0$ ], *YMY*, *TRAROT*[ $-\pi$ ], with positioning methods for *BEND* as discussed above (Fig. 14) still applying.

**BETATRON : Betatron core (momentum kick)**

*BETATRON* simulates a betatron yoke, for instance for slow extraction. It just imparts the particle a momentum kick, normally small.



**BREVOL : 1-D uniform mesh magnetic field map**

*BREVOL* reads a 1-D axial field map from a storage data file, whose content must match the following *FORTRAN* reading sequence (possible *FORMAT* updates are to be found in *fmapw.f*).

```

OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD' [,FORM='UNFORMATTED'])
DO 1 I = 1, IX
  IF (BINARY) THEN
    READ(NL) X(I), BX(I)
  ELSE
    READ(NL,*) X(I), BX(I)
  ENDF
1 CONTINUE

```

where *IX* is the number of nodes along the (symmetry) *X*-axis, *X(I)* their coordinates, and *BX(I)* are the values of the *X* component of the field. *BX* is normalized with *BNORM* factor prior to ray-tracing, as well *X* is normalized with the coefficient *XNORM* (useful to convert to centimeters, the working units in **zgoubi**). For binary files, *FNAME* must begin with 'B\_' or 'b\_', a flag 'BINARY' will thus be set to '.TRUE.' by the *FORTRAN*.

*X*-cylindrical symmetry is assumed, resulting in *BY* and *BZ* taken to be zero on axis.  $\vec{B}(X, Y, Z)$  and its derivatives along a particle trajectory are calculated by means of a 5-point polynomial interpolation followed by second order off-axis extrapolation (see sections 1.3.2, 1.4.1).

Entrance and/or exit integration boundaries may be defined in the same way as in *CARTEMES* by means of the flag *ID* and coefficients *A*, *B*, *C*, etc.

Including 'FLIP' in *TITL* causes the field map to be X-extended using its X-flipped copy.

**CARTEMES : 2-D Cartesian uniform mesh magnetic field map**

*CARTEMES* was originally dedicated to the reading and processing of the measured median plane field maps of the QDD spectrometer SPES2 at Saclay, assuming mid-plane dipole symmetry. However, it can be used for the reading of any 2-D median plane maps, provided that the format of the field data storage file fits the following *FORTRAN* sequence

```

OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD' [,FORM='UNFORMATTED'])
  IF (BINARY) THEN
    READ(NL) (Y(J), J=1, JY)
  ELSE
    READ(NL,FMT='(10F8.2)') (Y(J), J=1, JY)
  ENDIF
  DO 1 I=1, IX
  IF (BINARY) THEN
    READ(NL) X(I), (BMES(I,J), J=1, JY)
  ELSE
    READ(NL,FMT='(10F8.1)') X(I), (BMES(I,J), J=1, JY)
  ENDIF
1    CONTINUE

```

where,  $IX$  and  $JY$  are the number of longitudinal and transverse horizontal nodes of the uniform mesh, and  $X(I)$ ,  $Y(J)$  their coordinates.  $FNAME$  is the name of the file containing the field data. For binary files,  $FNAME$  must begin with 'B\_' or 'b\_', a flag 'BINARY' will thus be set to '.TRUE.' by the *FORTRAN*.

The measured field  $BMES$  is normalized with  $BNORM$ ,

$$B(I, J) = BMES(I, J) \times BNORM$$

As well the longitudinal coordinate  $X$  is normalized with a  $XNORM$  coefficient (useful to convert to centimeters, the working units in **zgoubi**).

The vector field,  $\vec{B}$ , and its derivatives out of the median plane are calculated by means of a second or fourth order polynomial interpolation, depending on the value of the parameter  $IODRE$  ( $IODRE = 2, 25$  or  $4$ , see section 1.4.2).

In case a particle exits the mesh, its  $IEX$  flag is set to  $-1$  (see section 7.13, p. 220), however it is still tracked with the field being *extrapolated* from the closest nodes of the mesh. Note that such extrapolation process may induce erratic behavior if the distance from the mesh gets too large.

Entrance and/or exit integration boundaries (so-called "droite de coupure" in the code) can be defined with the flag  $ID$ , as follows (Fig. 15).

**If  $ID = 1$**  : the integration in the field is terminated on a boundary with equation  $A'X + B'Y + C' = 0$ , and then the trajectories are extrapolated linearly onto the exit border of the map.

**If  $ID = -1$**  : an entrance boundary is defined, with equation  $A'X + B'Y + C' = 0$ , up to which trajectories are first extrapolated linearly from the map entrance border, prior to being integrated in the field.

**If  $ID \geq 2$**  : one entrance boundary, and  $ID - 1$  exit boundaries are defined, as above. The integration in the field terminates on the last ( $ID - 1$ ) exit boundary. No extrapolation onto the map exit border is performed in this case.

Including 'FLIP' in *TITL* causes the field map to be X-extended using its X-flipped copy.

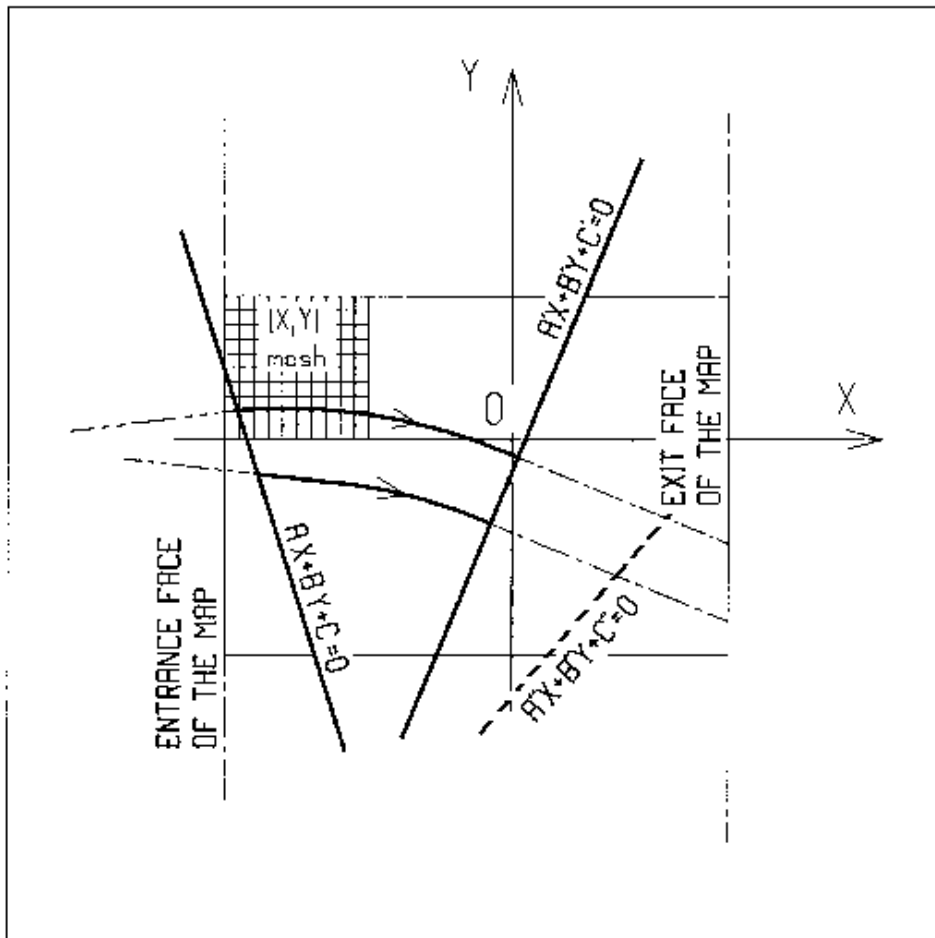


Figure 15:  $OXY$  is the coordinate system of the mesh. Integration boundaries may be defined, using  $ID \neq 0$ : particle coordinates are extrapolated linearly from the entrance face of the map, onto the boundary  $A'X + B'Y + C' = 0$ ; after ray-tracing inside the map and terminating on the boundary  $AX + BY + C = 0$ , coordinates are extrapolated linearly onto the exit face of the map if  $ID = 2$ , or terminated on the last  $(ID - 1)$  boundary if  $ID > 2$ .

**CAVITE : Accelerating cavity**

(page 252 in PART B)

*CAVITE* provides a simulation of a (zero length) accelerating cavity ; it can be used in conjunction with keywords *REBELOTE* and *SCALING* for the simulation of multi-turn tracking with synchrotron or fixed field (FFAG, cyclotron) acceleration (see section 7.12). It can also be used to simulate recirculating linacs and energy-recovery linacs. *CAVITE* must be preceded by *PARTICUL* for the definition of mass  $M$  and charge  $q$ .

*CAVITE* can be placed at any location in an optical sequence, and in any number. However, when RF phase matters, proper *IOPT* option should be used, and the RF phase at *CAVITE*(s) specified if necessary for its operation. Proper placement of *SCALING* command(s) relative to *CAVITE* have to be considered as well. In many cases, for instance a simulation of a pulsed synchrotron, *CAVITE* may just be placed at the downstream end of the sequence - no explicit phase specification needed in that case - and *SCALING* placed upstream of the optical sequence.

A major effect of *CAVITE* on optics settings is the following :

The reference rigidity  $B\rho_{ref}$  in **zgoubi**, as used for instance when determining optical strengths from field values (see sections 1.2.1-1.2.3), is defined when creating the object by means of *[MC]OBJET*. However, in many cases - options as described below - that reference rigidity will be updated upon crossing a cavity, consistently with the amount of synchronous rigidity increase as induced by the energy change, namely,

$$B\rho_{ref} = BORO \longrightarrow B\rho_{ref} = BORO + \delta B\rho_s \quad (6.3.8)$$

Note as an illustration of the process, that, in this case, a simple way to have the optical elements have their *strengths* maintained constant is to use *SCALING* with the option  $NTIM = -1$ .

- **If  $IOPT = 0$  :** *CAVITE* is switched off.
- **If  $IOPT = 1$  :** *CAVITE* simulates the RF cavity of a synchrotron accelerator : the periodic motion over  $IP = 1$ ,  $NPASS + 1$  turns (passes through the optical sequence) is obtained using the keyword *REBELOTE*, option  $K = 99$ , while RF and optical elements time dependent functions are simulated by means of *SCALING* – see section 7.12. *CAVITE* may conveniently be located *at one end* of the optical sequence, otherwise its phasing has to be provided. The synchrotron motion of any of the *IMAX* particles of a beam is obtained from the following mapping

$$\begin{cases} \phi_2 - \phi_1 = 2\pi f_{RF} \left( \frac{\ell}{\beta c} - \frac{\mathcal{L}}{\beta_s c} \right) \\ W_2 - W_1 = q\hat{V} \sin \phi_1 \end{cases}$$

where

- $\phi$  = RF phase ;  $\phi_2 - \phi_1$  = variation of  $\phi$  between two traversals
- $\hat{V}$  = peak RF voltage
- $W$  = kinetic energy ;  $W_2 - W_1$  = energy gain at a traversal of *CAVITE*
- $\mathcal{L}$  = length of the synchronous closed orbit (to be calculated by prior ray-tracing, see '*CLOSED ORBIT COMPUTATION*' below)
- $\beta_s c$  = velocity associated with the synchronous energy
- $\ell$  = particle trajectory length between two traversals
- $\beta c$  = particle velocity
- $c$  = velocity of light.

The RF frequency  $f_{RF}$  is a multiple of the synchronous revolution frequency, and is obtained from the input data, following

$$f_{RF} = \frac{hc}{\mathcal{L}} \frac{q(B\rho)_s}{\sqrt{q^2(B\rho)_s^2 + (Mc)^2}}$$

where  $h$  = harmonic number of the RF  
 $B\rho_s$  = rigidity associated with the synchronous energy

The synchronous rigidity  $(B\rho)_s$  is obtained from the timing law specified by means of *SCALING* following  $(B\rho)_s = BORO \cdot SCALE(TIMING)$  (see *SCALING* for the meaning and calculation of the scale factor *SCALE(TIMING)*). If *SCALING* is not used,  $(B\rho)_s$  is assumed to keep the constant value *BORO* as given in the object description (see *OBJET* for instance).

The velocity  $\beta c$  of a particle is calculated from its current rigidity

$$\beta = \frac{q(B\rho)}{\sqrt{q^2(B\rho)^2 + (Mc)^2}}$$

The velocity  $\beta_s c$  of the synchronous particle is obtained in the same way from

$$\beta_s = \frac{q(B\rho)_s}{\sqrt{q^2(B\rho)_s^2 + (Mc)^2}}$$

The kinetic energies and rigidities involved in these formulae are related by

$$q(B\rho) = \sqrt{W(W + 2Mc^2)}$$

Finally, the initial conditions for the mapping, at the first turn, are the following

- For the (virtual) synchronous particle

$$\begin{aligned} \phi_1 &= \phi_s = \text{synchronous phase} \\ (B\rho)_{1s} &= BORO \end{aligned}$$

- For any of the  $I = 1, IMAX$  particles of the beam

$$\begin{aligned} \phi_{1I} &= \phi_s = \text{synchronous phase} \\ (B\rho)_{1I} &= BORO \times D_I \end{aligned}$$

where the quantities *BORO* and  $D_I$  are given in the object description.

### Calculation of the Coordinates

Let  $p_I = [p_{XI}^2 + p_{YI}^2 + p_{ZI}^2]^{1/2}$  be the momentum of particle  $I$  at the exit of the cavity, while  $p_{I_0} = [p_{XI_0}^2 + p_{YI_0}^2 + p_{ZI_0}^2]^{1/2}$  is its momentum at the entrance. The kick in momentum is assumed to be fully longitudinal, resulting in the following relations between the coordinates at the entrance (denoted by the index zero) and at the exit

$$\begin{aligned}
p_{XI} &= [p_I^2 - (p_{I_0}^2 - p_{X_{I_0}}^2)]^{1/2} \\
p_{YI} &= p_{Y_{I_0}}, \quad \text{and} \quad p_{ZI} = p_{Z_{I_0}} \quad (\text{longitudinal kick}) \\
X_I &= X_{I_0}, \quad Y_I = Y_{I_0} \quad \text{and} \quad Z_I = Z_{I_0} \quad (\text{zero length cavity})
\end{aligned}$$

and for the angles (see Fig. 1)

$$\left. \begin{aligned}
T_I &= \text{Atg} \left( \frac{p_{YI}}{p_{XI}} \right) \\
P_I &= \text{Atg} \left( \frac{P_{ZI}}{(p_{XI}^2 + p_{YI}^2)^{1/2}} \right)
\end{aligned} \right\} \quad (\text{damping of the transverse motion})$$

- **If IOPT = 2** : the same simulation of a synchrotron RF cavity as for IOPT = 1 is performed, except that the keyword *SCALING* (family *CAVITE*) is not taken into account in this option : the increase in kinetic energy at each traversal, for the synchronous particle, is

$$\Delta W_s = q\hat{V} \sin \phi_s$$

where the synchronous phase  $\phi_s$  is part of the input data to *CAVITE*. From this, the calculation of the law  $(B\rho)_s$  and the RF frequency  $f_{RF}$  follows, according to the formulae given in the *IOPT* = 1 case.

*IOPT* = 2 can be used for SR loss compensation in *storage ring* mode. (use *IOPT*=11 for accelerated mode). SR loss is switched on by means of the keyword *SRLOSS/KSR*=1. In that case  $\hat{V}$  and  $\phi_s$  should be given the appropriate value for compensation of SR induced energy loss ( $\hat{V} \times \sin \phi_s$  should be equal to the energy loss). The same functioning (stationary bucket with SR loss) can be obtained using *IOPT*=11, which further allows acceleration in the presence of SR energy loss.

*IOPT* = 2 handles double RF systems. Respective harmonic value and voltage amplitudes are  $h_1, h_2$  and  $V_1, V_2$ . It follows CERN ISR-TH-RF/80-26 [52]. In particular, the stable phase  $\phi_n$  of the higher harmonic voltage is taken to be

$$\phi_n = \text{atan}(\tan(\phi_s)/n)/n$$

wherein  $n = h_2/h_1$ . The synchronous energy gain is

$$\Delta W_s = q\hat{V} (\sin(\phi_s) + k \sin(n\phi_n))$$

with  $k = V_2/V_1$ .

The option *IOPT* = 2 handles synchrotron radiation energy loss in the case of stationary bucket (see *IOPT* = 11 for accelerating bucket). Compensation is derived from the synchronous phase  $\phi_s$ , via  $\Delta W_{SR} = q\hat{V} \sin \phi_s$ , therefore assuming appropriate value

$$\phi_s = \text{asin}(\Delta W_{SR}/q\hat{V})$$

with  $\Delta W_{SR}$  = the energy loss between passes in the cavity.

- **If IOPT = 3** : sine RF law, acceleration without synchrotron motion. Any particle will be given a kick

$$\Delta W = q\hat{V} \sin \phi_s$$

where  $\hat{V}$  and  $\phi_s$  are input data.

The option *IOPT* = 3 handles synchrotron radiation energy loss in the case of stationary bucket (as in *IOPT* = 2).

- **If IOPT = 4 :** to be documented.
  
- **If IOPT = 5 :** to be documented.
  
- **If IOPT = 6 :** allows reading the RF frequency and/or phase law from an external ascii file (with name normally `zgoubi.freqLaw.In` - the formatting of these ascii data file is given page 185). See routines `cavite.f` and `scal.in.f` for details. Was first used for acceleration in scaling FFAG [60].
  
- **If IOPT = 7 :** fixed frequency RF, quasi- or isochronous acceleration. Was first used for quasi-isochronous, fixed frequency acceleration in the EMMA prototype linear FFAG [61, 62].

*IOPT = 7* can be used for cyclotron acceleration.

- **If IOPT = 8 :** to be documented.
  
- **If IOPT = 9 :** to be documented.
  
- **If IOPT = 10 :** fixed frequency RF. Was first installed/used for CEBAF and eRHIC ERL simulations, later used for Cornell's CBETA FFAG ERL simulaitons.

A “*Facility=CEBAF*” flag allows some specificities regarding the management of reference phase and individual particle phase in the case of CEBAF cavities, which includes the following:

- the RF crest voltage is at  $\phi_s = \pi/2$  (otherwise, it is at  $\phi_s = 0$ ),
- the RF phase of any particle is offset by an amount equal to the phase of the bunch centroid.

It is equivalent to set “*CEBAF*”, or to set both “*SHIFT\_PH0*” (with RF in this case  $\phi_s = 0$  instead) and “*CENTROID*” (see below). If “*CEBAF*” is set, then it overrides the previous two, whatever their state.

Two additional flags are available with

- “SHIFT\_PH0”: this causes an offset of particle phases, by  $\pi/2$  (it would be equivalent to offset  $\phi_s$  datum, instead),
- “CENTROID”: this causes the RF phase of any particle to be offset by an amount equal to the phase of the bunch centroid. The bunch centroid is computed as an average over the *IMAX* particles launched by *[MC]OBJET*.

A Chambers matrix method is used to get the transverse focusing effect. Particles undergo a longitudinal boost in energy

$$\Delta W = q\hat{V} \cos(2\pi f_{RF}t + \phi_s)$$

with  $t$  the arrival time at the center of the cavity. The phase datum  $\phi_s$  in this option has two roles, (i) that played in the equation above, (ii) determining the updating of **zgoubi**'s reference rigidity, namely,

$$B\rho_{ref} = BORO \rightarrow B\rho_{ref} = BORO + \delta B\rho_s \quad \text{with} \quad \delta B\rho_s = \frac{\Delta(\beta_s(W_s + M))}{qc}, \quad \Delta W_s = q\hat{V} \cos(\phi_s) \quad (6.3.9)$$

The cavity length  $L$  as specified in the input data is also used to update the path length of the particles (linearly). An optional 'IOP' flag allows small  $\Delta W$  approximation of Chambers matrices, as well as removing the longitudinal damping (thus geometrical emittance is conserved), as follows :

- IOP = 0 : Default value ; the transverse matrix is that of a drift with length  $L$ .
- IOP = 1 : Chambers matrix with  $\Delta W/W$  approximation, namely, for both planes,

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{out} = \begin{pmatrix} \sqrt{(W_i/W_o)} & L_{cav} \times \sqrt{(W_i/W_o)} \\ 0 & \sqrt{(W_i/W_o)} \end{pmatrix} \begin{pmatrix} x \\ x' \end{pmatrix}_{in} \quad (6.3.10)$$

with  $W_i, W_o$  respectively the incoming and outgoing kinetic energies.

- IOP = -1 : Same as 1, with matrix in both planes re-normalized to determinant=1 (i.e.,  $R_{ij} \rightarrow R_{ij}/\sqrt{R_{11}R_{12} - R_{21}R_{22}}$ ),
- IOP = 2 : Chambers matrix without approximation on  $\Delta W/W$ , namely, for both planes,

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{out} = \begin{pmatrix} \cos(u) - \sqrt{2} \sin(u) \cos(\phi) & vW_i \sin(u) \cos(\phi) \\ -\frac{\sin(u)}{vW_o} (2 \cos(\phi) + \frac{1}{\cos(\phi)}) & \frac{W_i}{W_o} (\cos(u) + \sqrt{2} \sin(u) \cos(\phi)) \end{pmatrix} \begin{pmatrix} x \\ x' \end{pmatrix}_{in} \quad (6.3.11)$$

with  $u = \log(W_o/W_i)/(\sqrt{8} \cos(\phi))$ ,  $v = \sqrt{8}L/(W_o - W_i)$ ,  $\phi$  the particle phase at the cavity.

- IOP = -2 : Same as 2, with matrix in both planes re-normalized to determinant=1.

• **If IOPT = 11** : under development.

The same simulation of a synchrotron RF cavity as for **IOPT = 2** is performed, **IOPT = 11** handles synchrotron radiation energy loss as well, yet both stationary or accelerated bucket.

**IOPT = 11** allows specifying the machine concerned, for instance 'CornellSynch' (Cornell RCS; the RF time-law is hard-coded in the **zgoubi** source file `cavite.f`), 'EICRCS' (Electron Ion Collider electron RCS), EICRCS\_File (Electron Ion Collider electron RCS, reading RF program from an external file). Additional RF laws, in existing or additional **IOPT** cases, can easily be hard-coded in `cavite.f`.



In the 'CornellSynch' option, the RF peak voltage input data is not used, the voltage is hard-coded (in `cavite.f`), - the acceleration voltage applied is [88]

$$V(t) = 4.4 \sin(2\pi ft) + 8.8 \sin^8(2\pi ft/2), \quad f = 60 \text{ Hz}$$

*CAVITE* in this option accounts for the Synchrotron radiation energy loss,  $U_0$ , to be recovered at this *CAVITE*; in particular if only one cavity is used, then  $U_0$  is the energy loss per turn.  $U_0$  adds to the peak voltage (unless 'CornellSynch' is specified), namely, the actual RF peak voltage follows the evolution of  $U_0$  to provide

$$\hat{V}_{\text{effective}} = \hat{V} + U_0$$

The initial value of the energy to be recovered needs be specified,  $U_{00}$ , it is then updated at the subsequent passes using

$$U_0 = U_{00} \left( \frac{E}{E_1} \right)^4$$

with  $E$  the current energy, and  $E_1$  the energy at the first pass. If  $U_{00}$  is left null whereas the keyword *SRLOSS* (pp. 190, 339) is set for energy loss ( $KSR=1$ ), then  $U_{00}$  is set, at the first pass, in a hard-coded way, using the curvature radius and assuming an iso-field lattice, namely

$$U_0 = 88.46276 \times E_{[\text{MeV}]}^4 / \rho_{[\text{m}]}$$

This is done in `cavite.f`, at the moment this setting takes  $\rho = 235$  m (case of the US EIC RCS electron injector in RHIC tunnel), this can be changed and the Fortran recompiled.

Positioning of *CAVITE*: knowledge of cavity location azimuth along the actual closed orbit (which may not be null, in the presence of orbit defects for instance) is necessary if the cavity is not located at an end of the periodic optical sequence. In that case a prior run, possibly using *FIT* to find the closed orbit (see Sec. 7.3), will deliver that azimuth.

### CLOSED ORBIT COMPUTATION

Due to possible dipole type of optical defects (*e.g.*, fringe fields, straight axis combined function dipoles), the closed orbit may not coincide with the ideal axis of the optical elements and will instead be almost everywhere non-zero (see also Sec. 7.3, p. 213).

One method in order to get the closed orbit at the beginning of the optical sequence (*i.e.*, where the initial particle coordinates or spin coordinates are defined) consists in using the *FIT*[2] procedure. Use in this case the "3.1" (orbit) or "10.1" (spin) periodicity constraints.

Another method consists in ray-tracing a single particle over a sufficiently large number of turns, starting with initial conditions taken near the reference orbit, so as to obtain statistically well-defined transverse phase-space ellipses. The local closed orbit coincides with the coordinates  $Y_c, T_c, Z_c, P_c$  of the center of the ellipses. A few iterations are usually sufficient (avoid near-integer tunes) to ensure accuracy. Next, ray-tracing over one turn a particle starting with the initial condition  $(Y_c, T_c, Z_c, P_c)$  will provide the closed orbit along the all optical sequence, and as a sub-product its length  $\mathcal{L}$  (the  $F(6, 1)$  coordinate in the source *FORTRAN*).

### RECOVERING FROM A CRASH

When accelerating a bunch in a pulsed ring, *SCALING* is used to ramp magnetic fields, so following the rigidity increase by *CAVITE*. *CAVITE* may or may not be part of the *SCALING* argument list, this is discussed in the *SCALING* section, page 186. Now, if the job is stopped - due to computer crash or else - it is possible to resume the tracking from the latest records (in `[b_]zboubi.fai` storage file), using *OBJET*, *KOBJ*= 3. In the presence of acceleration using *CAVITE*, since there is no reference particle in **zgoubi**, *CAVITE* will read from `[b_]zboubi.fai` the reference rigidity at the corresponding time/turn when tracking crashed/resumes. See the *SCALING* section for more (page 183) and the examples there, as well as the example page 310.

**PRINT** : The optional command 'PRINT', following the first argument, 'IOPT', in *CAVITE* keyword, will cause logging of computational data (synchronous time and momentum, RF phase, and much more) in the file `zgoubi.CAVITE.Out`.

***For software developers***

Cavity simulations are done in `cavite.f`. Management of the reference rigidity  $B\rho_{ref}$  is ensured for part in `cavite.f`, and for part in routines associated with 'SCALING', essentially `scaler.f`.

**CHAMBR : Long transverse aperture limitation**

*CHAMBR* causes the identification, counting and stopping of particles that reach the transverse limits of the vacuum chamber. The chamber can be either rectangular (*IFORM* = 1) or elliptic (*IFORM* = 2). The chamber is centered at *YC*, *ZC* and has transverse dimensions  $\pm YL$  and  $\pm ZL$  such that any particle will be stopped if its coordinates *Y*, *Z* satisfy

$$(Y - YC)^2 \geq YL^2 \text{ or } (Z - ZC)^2 \geq ZL^2 \quad \text{if } IFORM = 1$$

$$\frac{(Y - YC)^2}{YL^2} + \frac{(Z - ZC)^2}{ZL^2} \geq 1 \quad \text{if } IFORM = 2$$

The conditions introduced with *CHAMBR* are valid along the optical sequence until the next occurrence of the keyword *CHAMBR*. Then, if *IL* = 1 the aperture is possibly modified by introducing new values of *YC*, *ZC*, *YL* and *ZL*, or, if *IL* = 2 the chamber ends and information is printed concerning those particles that have been stopped.

The testing is done in optical elements at each integration step, between the *EFB*s. For instance, in *QUADRUPO* there will be no testing from  $-X_E$  to 0 and from *XL* to  $XL + X_S$ , but only from 0 to *XL*; in *DIPOLE*, there is no testing as long as the *ENTRANCE EFB* is not reached, and testing is stopped as soon as the *EXIT* or *LATERAL EFB*s are passed.

In optical elements defined in polar coordinates, *Y* stands for the radial coordinate (e.g., *DIPOLE*, see Figs. 3C, p. 33, and 20, p. 99). Thus, centering *CHAMBR* at  $YC = RM$  simulates a chamber curved with radius *RM*, and having a radial acceptance  $RM \pm YL$ . In *DRIFT*, the testing is done at the beginning and at the end, and only for positive drifts. There is no testing in *CHANGREF*.

When a particle is stopped, its index *IEX* (see *OBJET* and section 7.13) is set to the value -4, and its actual path length is stored in the array *SORT* for possible further use.

**CHANGREF : Transformation to a new reference frame**

*CHANGREF* transports particles from a reference plane ( $O, Y, Z$ ) at path distance  $S$ , to a new one by a combination of translations and/or rotations. It essentially aims at positioning optical elements with respect to one another, as setting a reference frame at the entrance or exit of field maps, or to simulate misalignments (see also *KPOS* option). *CHANGREF* can be placed at any location in an optical sequence.

Spin tracking, particle decay and gas-scattering are taken into account in *CHANGREF*. Energy loss by synchrotron radiation (*SRLOSS* keyword) is not.

There are two “styles” of *CHANGREF*, as follows.

**The “old style” *CHANGREF*** requires the three data *XCE*, *YCE*, *ALE* and then gets the new particle coordinates  $Y_2, T_2, Z_2, P_2$  and path length  $S_2$  from the old ones  $Y_1, T_1, Z_1, P_1$  and  $S_1$  using

$$T_2 = T_1 - ALE$$

$$Y_2 = \frac{(Y_1 - YCE) \cos T_1 + XCE \sin T_1}{\cos T_2}$$

$$DL^2 = (XCE - Y_2 \sin ALE)^2 + (YCE - Y_1 + Y_2 \cos ALE)^2$$

$$Z_2 = Z_1 + DL \tan P_1$$

$$S_2 = S_1 + \frac{DL}{\cos P_1}$$

$$P_2 = P_1$$

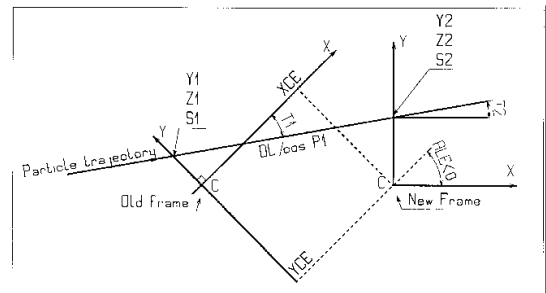


Figure 16: Scheme of the *CHANGREF* procedure.

where, *XCE* and *YCE* are shifts in the horizontal plane along, respectively, *X*- and *Y*-axis, and *ALE* is a rotation around the *Z*-axis. *DL* is given the sign of  $XCE - Y_2 \sin(ALE)$ .

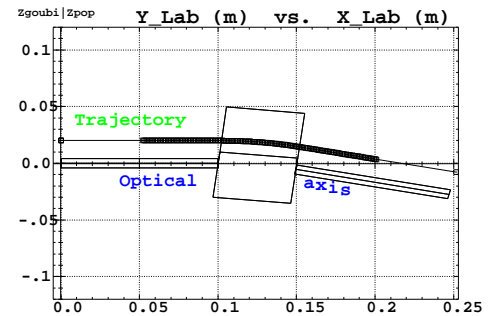
**EXAMPLE** - The example below illustrates the use of *CHANGREF*. This is a case of symmetric positioning of a combined function dipole+quadrupole magnet in a drift-bend-drift geometry with 12.691 degrees deviation (obtained upon combined effect of a dipole component and of quadrupole axis shifted 1 cm off optical axis).

## Zgoubi data file :

```

Using CHANGREF, "Old style"
'OBJET'
51.71103865921708      ! Electron, Ekin=15MeV.
2
1 1                    ! One particle, with
2. 0.  0.0 0.0 0.0 1. 'R' ! Y_0=2 cm.
1 1 1 1 1 1
'MARKER'   BEG      .plt ! Print into zgoubi.plt.
'DRIFT'    ! 10 cm drift.
10.
'CHANGREF'
0. 0. -6.34165        ! 1/ half Z-rotation.
'CHANGREF'
0. 1. 0.              ! 2/ Y-shift.
'MULTIPOL' ! Combined function dipole + quadrupole.
  2                ! Print into zgoubi.plt.
5 10. 2.064995867082342 2. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0 0 5. 1.1 1.00 1.00 1.00 1.00 1.00 1. 1. 1. 1.
4 .1455 2.2670 -.6395 1.1558 0. 0.
  0 0 5. 1.1 1.00 1.00 1.00 1.00 1.00 1. 1. 1. 1.
4 .1455 2.2670 -.6395 1.1558 0. 0.
0 0 0 0 0 0 0 0 0 0
.1  step size
1 0. 0. 0.
'CHANGREF'
0. -1. -6.34165      ! 1/ Y-shift, 2/ half Z-rotate.
'DRIFT'   - .plt     ! 10 cm drift. Print into zgoubi.plt.
10.
'FAISCEAU'
'END'

```



Note : The square markers scheme the stepwise integration in case of  $\pm 5$  cm additional fringe field extent upstream and downstream of the 5 cm long multipole.

The “new style” *CHANGREF* allows all 6 degrees of freedom rather than just 3, namely, X-, Y-, Z-shift, X-, Y-, Z-rotation. In addition, *CHANGREF* “new style” allows up to 9 successive such elementary transformations, in arbitrary order. The “old style” example above is transposed into “new style”, hereafter.

## Zgoubi data file :

```

Using CHANGREF, "New Style"
'OBJET'
51.71103865921708      ! Electron, Ekin=15MeV.
2
1 1                    ! One particle, with
2. 0.  0.0 0.0 0.0 1. 'R' ! Y_0=2 cm.
1 1 1 1 1 1
'MARKER'   BEG      .plt ! Print into zgoubi.plt.
'DRIFT'    ! 10 cm drift.
10.
'CHANGREF'
ZR -6.34165 YS 1.      ! 1/ half Z-rotate, 2/ Y-shift.
'MULTIPOL' ! Combined function dipole + quadrupole.
  2                ! Print into zgoubi.plt.
5 10. 2.064995867082342 2. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0 0 5. 1.1 1.00 1.00 1.00 1.00 1.00 1. 1. 1. 1.
4 .1455 2.2670 -.6395 1.1558 0. 0.
  0 0 5. 1.1 1.00 1.00 1.00 1.00 1.00 1. 1. 1. 1.
4 .1455 2.2670 -.6395 1.1558 0. 0.
0 0 0 0 0 0 0 0 0 0
.1  step size
1 0. 0. 0.
'CHANGREF'
YS -1. ZR -6.34165    ! 1/ Y-shift, 2/ half Z-rotate.
'DRIFT'   - .plt     ! 10 cm drift. Print into zgoubi.plt.
10.
'FAISCEAU'
'END'

```

*SCALING* can be used to scale positioning parameters in *CHANGREF*.

**CIBLE or TARGET : Generate a secondary beam following target interaction**

The reaction is  $1 + 2 \rightarrow 3 + 4$  with the following parameters

Laboratory momentum	$p_1 \equiv 0$	$p_2$	$p_3$	$p_4$
Rest mass	$M_1$	$M_2$	$M_3$	$M_4$
Total energy in laboratory	$M_1 c^2$	$W_2$	$W_3$	$W_4$

The geometry of the interaction is shown in Fig. 17.

The angular sampling at the exit of the target consists of the  $NT$  coordinates  $0, \pm TS, \pm 2 \times TS \dots \pm (NT - 1) \times TS/2$  in the median plane, and the  $NP$  coordinates  $0, \pm PS, \pm 2 \times PS \dots \pm (NP - 1) \times PS/2$  in the vertical plane.

The position of  $B$  downstream is deduced from that of  $A$  upstream by a transformation equivalent to two transformations using *CHANGREF*, namely

$$\text{CHANGREF}(XCE = YCE = 0, \quad ALE = \beta)$$

followed by

$$\text{CHANGREF}(XCE = YCE = 0, \quad ALE = \theta - \beta).$$

Particle 4 is discarded, while particle 3 continues. The energy loss  $Q$  is related to the variable mass  $M_4$  by

$$Q = M_1 + M_2 - (M_3 + M_4) \quad \text{and} \quad dQ = -dM_4$$

The momentum sampling of particle 3 is derived from conservation of energy and momentum, according to

$$M_1 c^2 + W_2 = W_3 + W_4$$

$$p_4^2 = p_2^2 + p_3^2 - 2p_2 p_3 \cos(\theta - T)$$

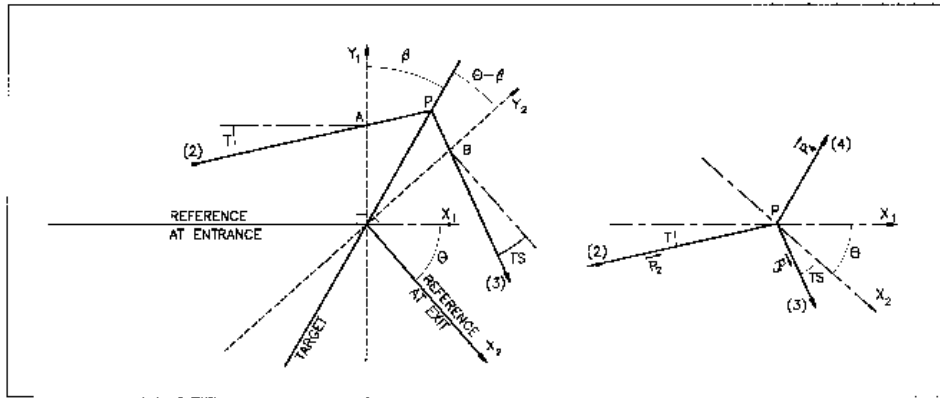


Figure 17: Scheme of the principles of *CIBLE (TARGET)*

- $A, T$  = position, angle of incoming particle 2 in the entrance reference frame
- $P$  = position of the interaction
- $B, T$  = position, angle of the secondary particle in the exit reference frame
- $\theta$  = angle between entrance and exit frames
- $\beta$  = tilt angle of the target

**COLLIMA : Collimator**

*COLLIMA* acts as a mathematical aperture of zero length. It causes the identification, counting and stopping of particles that reach the aperture limits.

**Physical Aperture**

A physical aperture can be either rectangular (*IFORM* = 1) or elliptic (*IFORM* = 2). The collimator is centered at *YC*, *ZC* and has transverse dimensions  $\pm YL$  and  $\pm ZL$  such that any particle will be stopped if its coordinates *Y*, *Z* satisfy

$$(Y - YC)^2 \geq YL^2 \text{ or } (Z - ZC)^2 \geq ZL^2 \quad \text{if } \textit{IFORM} = 1$$

$$\frac{(Y - YC)^2}{YL^2} + \frac{(Z - ZC)^2}{ZL^2} \geq 1 \quad \text{if } \textit{IFORM} = 2$$

**Longitudinal Collimation**

*COLLIMA* can act as a longitudinal phase-space aperture, coordinates acted on are selected with *IFORM.J*. Any particle will be stopped if its horizontal (*h*) and vertical (*v*) coordinates satisfy

$$(h \leq h_{min} \text{ or } h \geq h_{max}) \text{ or } (v \leq v_{min} \text{ or } v \geq v_{max})$$

wherein, *h* is either path length *S* if *IFORM*=6 or time if *IFORM*=7, and *v* is either 1+DP/P if *J*=1 or kinetic energy if *J*=2 (provided mass and charge have been defined using the keyword *PARTICUL*).

**Phase-Space Elliptical Collimation**

*COLLIMA* can act as a phase-space elliptical aperture. Any particle will be stopped if its coordinates satisfy

$$\begin{aligned} \gamma_Y Y^2 + 2\alpha_Y Y T + \beta_Y T^2 &\geq \epsilon_Y / \pi & \text{if } \textit{IFORM} = 11 \text{ or } 14 \\ \gamma_Z Z^2 + 2\alpha_Z Z P + \beta_Z P^2 &\geq \epsilon_Z / \pi & \text{if } \textit{IFORM} = 12 \text{ or } 15 \\ \gamma_S S^2 + 2\alpha_S S D + \beta_S D^2 &\geq \epsilon_S / \pi & \text{if } \textit{IFORM} = 13 \text{ or } 16 \text{ (under development)} \end{aligned}$$

If *IFORM*=11 (respectively 12, 13) then  $\epsilon_Y / \pi$  (respectively  $\epsilon_Z / \pi$ ,  $\epsilon_S / \pi$ ) is to be specified by the user as well as  $\alpha_{Y,Z,S}$ ,  $\beta_{Y,Z,S}$ . If *IFORM*=14 (respectively 15, 16) then  $\alpha_Y$  and  $\beta_Y$  (respectively  $\alpha_{Z,S}$ ,  $\beta_{Z,S}$ ) are determined by **zgoubi** by prior computation of the matched ellipse to the particle population, so only  $\epsilon_{Y,Z,S} / \pi$  need be specified by the user.

When a particle is stopped, its index *IEX* (see *OBJET* and section 7.13) is set to the value -4, and its actual path length is stored in the array *SORT* for possible further use (e.g., by *HISTO*, or for loss studies, etc.).

**CYCLOTRON : Spiral setor cyclotron**

(page 258 in PART B)

*CYCLOTRON* provides a model of a sector dipole, or a N-tuple of neighboring sector dipoles, with straight or spiraled EFBs [38, Appendix C, p. 149] [39]. The source code has been derived from *FFAG-SPI*, derived itself from *DIPOLÉS*, thus there is many similarities in the capabilities and operation of these keywords. The field model assumes mid-plane symmetry, with mid-plane normal field component writing

$$B_Z = BN \sum_{i=1}^{N \text{ dipoles}} \mathcal{F}_i(R, \theta) \mathcal{R}_i(R) \quad (6.3.12)$$

with

- $R$  the distance to the geometrical center of the cyclotron,  $\theta$  the azimuthal angle,
- $BN$  a normalization coefficient,
- $\mathcal{R}_i(R) = B_{0,i} + B_{1,i} \times R + B_{2,i} \times R^2 + B_{3,i} \times R^3 + \dots$  (up to  $R^{20}$ )
- $\mathcal{F}(R, \theta)$  the flutter, which accounts in particular for fringe fields.

The field along a particle trajectory is computed as the particle motion proceeds: at any position  $(R, \theta)$  along the trajectory (Fig 18), the flutter is calculated from the distance to the effective field boundaries of the  $i=1, N$  magnet(s), in the following way.

$$\mathcal{F}(R, \theta) = \mathcal{F}_{entr}(R, \theta) \times \mathcal{F}_{exit}(R, \theta) = \frac{1}{1 + \exp(P_{entr}(d_{entr}))} \times \frac{1}{1 + \exp(P_{exit}(d_{exit}))} \quad (6.3.13)$$

where

$$P(d) = C_0 + C_1 \left(\frac{d}{g}\right) + C_2 \left(\frac{d}{g}\right)^2 + C_3 \left(\frac{d}{g}\right)^3 + C_4 \left(\frac{d}{g}\right)^4 + C_5 \left(\frac{d}{g}\right)^5 + C_6 \left(\frac{d}{g}\right)^6 + C_7 \left(\frac{d}{g}\right)^7 \quad (6.3.14)$$

and  $d$  is the distance from the Effective Field Boundary (EFB) either at the entrance or at the exit of the magnet ( $d_{entr}$  and  $d_{exit}$  as shown in Fig 18).

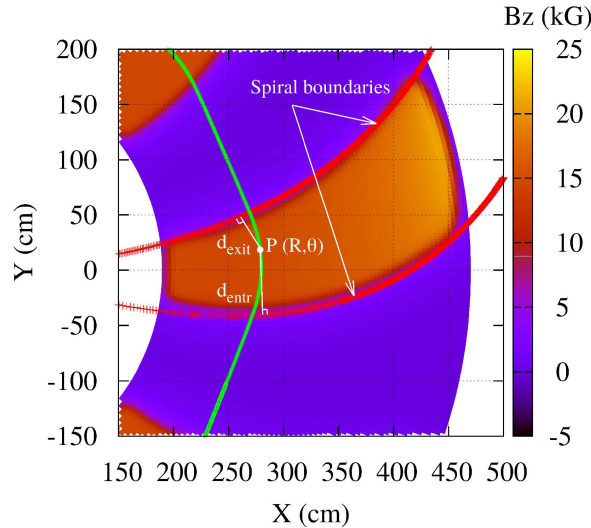


Figure 18: Definition of the EFB in *CYCLOTRON*: the trajectory of the moving particle (anti-clockwise motion, here) is in green while the EFB are in red.

An EFB is either straight or spiraled. In the latter case it is modelled by a logarithmic spiral for which the angle  $\xi$  is allowed to increase radially, namely:

$$R = R_0 \times \exp\left(\frac{\theta + \omega}{\tan \xi(R)}\right) \quad (6.3.15)$$

where  $\xi(R) = \xi_0 + \xi_1 \times R + \xi_2 \times R^2 + \xi_3 \times R^3$ ,  $\theta$  is the azimuthal angle and  $\omega$  is a parameter used to position the EFB with respect to the azimuthal position  $\theta=0$ .



In this model, the magnet gap is also allowed to vary with  $R$ , following

$$g(R) = g_0 + g_1 \times R + g_2 \times R^2 \quad (6.3.16)$$

and this determines a radial dependence of the fringe field extent, as follows from P(d) model, Eq. 6.3.14. The field is then extrapolated off median plane by means of Taylor series: for that, the median plane antisymmetry is assumed and Maxwell's equations are accounted for.

*CYCLOTRON* allows the overlapping of up to  $N=5$  such dipole fields. This follows the method described in [53]. Strictly speaking, the dipoles may even be superimposed (to the point, for instance, that a single dipole ( $N=1$ ) in an AT sector, with some plateau field value  $B_{Z0}$ , can be simulated by superimposing  $N \leq 5$  identical dipoles, with identical positioning within AT and field  $B_{Z0}/N$  each).

In the case of a cyclotron machine, the isochronicity is a crucial point: Because the revolution time has to be constant ( $f_{rev} = \frac{qB}{2\pi\gamma m_0}$ ), this implies that the radial dependence of the field must be proportional to  $\gamma$ , so that  $\mathcal{R}(\bar{R}) \propto \gamma(\bar{R})$ , where  $\bar{R}$  is the average radius of the orbit. Since  $f_{rev} = \frac{v}{C}$ , where  $C$  is the path length of the particle for one closed orbit, one obtains, with a good approximation, that  $R \propto \beta$ . Thus,

$$\mathcal{R}(R) \approx \frac{1}{\sqrt{1 - \left(\frac{R}{R_0}\right)^2}} \quad (6.3.17)$$

*EXAMPLE* - An example of the set up and use of *CYCLOTRON* is given in PART. B, p. 259.

**DECAPOLE : Decapole magnet (Fig. 19)**

(page 261 in PART B)

The meaning of parameters for *DECAPOLE* is the same as for *QUADRUPO*.

In fringe field regions the magnetic field  $\vec{B}(X, Y, Z)$  and its derivatives up to fourth order are derived from the scalar potential expressed to the 5th order in  $Y$  and  $Z$

$$V(X, Y, Z) = G(X) \left( Y^4 Z - 2Y^2 Z^3 + \frac{Z^5}{5} \right)$$

$$\text{with } G_0 = \frac{B_0}{R_0^4}$$

The modelling of the fringe field form factor  $G(X)$  is described under *QUADRUPO*, p. 136.

Outside fringe field regions, or everywhere in sharp edge decapole ( $\lambda_E = \lambda_S = 0$ ),  $\vec{B}(X, Y, Z)$  in the magnet is given by

$$B_X = 0$$

$$B_Y = 4G_0(Y^2 - Z^2)YZ$$

$$B_Z = G_0(Y^4 - 6Y^2Z^2 + Z^4)$$

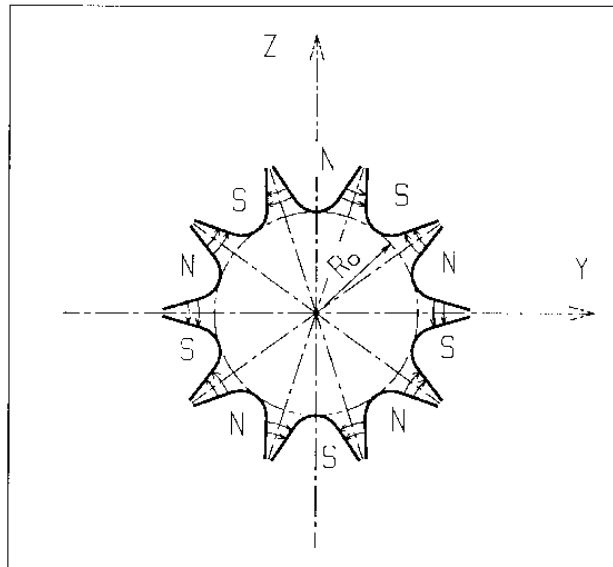


Figure 19: Decapole magnet

**DIPOLE : Dipole magnet, polar frame**

(page 262 in PART B)

*DIPOLE* provides a model of a dipole field, and allows transverse indices. The field along a particle trajectory is computed as the particle motion proceeds, straightforwardly from the dipole geometrical boundaries. Field simulation in *DIPOLE* is the same as used in *DIPOLE-M* and *AIMANT* for computing a field map (Fig. 20) ; the essential difference in *DIPOLE* is in its skipping that intermediate stage of field map generation found in the former two.

*DIPOLE* is available in an alternate version, *DIPOLÉS*, that allows overlapping of fringe fields in a configuration where several neighboring magnets are comprised within the AT angular extent (such as in Fig. 21).

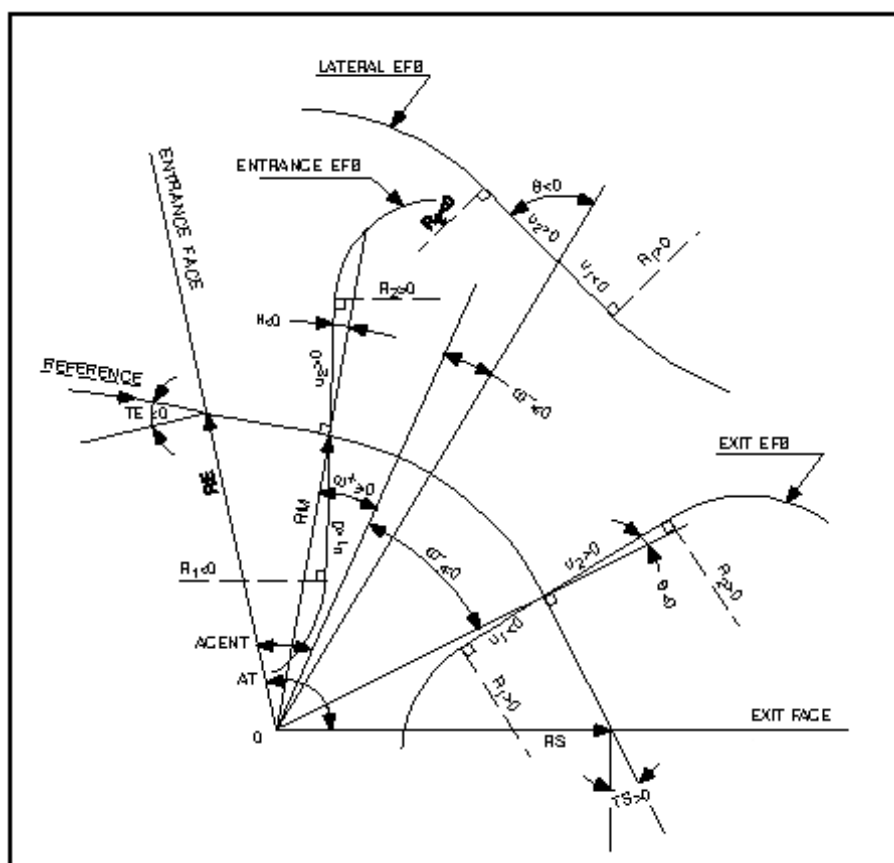


Figure 20: Parameters used to define the geometry of a dipole magnet with index, using *DIPOLE[S]*

The dimensioning of the magnet is defined by (Fig. 20)

*AT* : total angular aperture

*RM* : mean radius used for the positioning of field boundaries

The 2 or 3 effective field boundaries (EFB), from which the dipole field is drawn, are defined from geometric boundaries, the shape and position of which are determined by the following parameters.

*ACENT* : arbitrary inner angle, used for EFBs positioning

$\omega$  : azimuth of an EFB with respect to *ACENT*

$\theta$  : angle of an EFB with respect to its azimuth (wedge angle)

$R_1, R_2$  : radius of curvature of an EFB

$U_1, U_2$  : extent of the linear part of an EFB.

The magnetic field is calculated in polar coordinates. At any position  $(R, \theta)$  along the particle trajectory the value of the vertical component of the mid-plane field is calculated using

$$B_Z(R, \theta) = \mathcal{F}(R, \theta) \times B_0 \times \left( 1 + N \times \frac{R - RM}{RM} + B \times \left( \frac{R - RM}{RM} \right)^2 + G \times \left( \frac{R - RM}{RM} \right)^3 \right) \quad (6.3.18)$$

where  $N$ ,  $B$  and  $G$  are respectively the first, second and third order field indices and  $\mathcal{F}(R, \theta)$  is a fringe field factor (it determines the “flutter” in periodic structures).

### Calculation of the Fringe Field Factor

With each EFB a realistic extent of the fringe field,  $\lambda$  (normally equal to the gap size), is associated and a fringe field factor  $F$  is calculated. In the following  $\lambda$  stands for either  $\lambda_E$  (Entrance),  $\lambda_S$  (Exit) or  $\lambda_L$  (Lateral EFB).

$F$  is an exponential type fringe field (Fig. 10, p. 74) given by [48]

$$F = \frac{1}{1 + \exp P(s)}$$

wherein  $s$  is the distance to the EFB and depends on  $(R, \theta)$ , and

$$P(s) = C_0 + C_1 \left( \frac{s}{\lambda} \right) + C_2 \left( \frac{s}{\lambda} \right)^2 + C_3 \left( \frac{s}{\lambda} \right)^3 + C_4 \left( \frac{s}{\lambda} \right)^4 + C_5 \left( \frac{s}{\lambda} \right)^5$$

It is also possible to simulate a shift of the *EFB*, by giving a non zero value to the parameter *shift*.  $s$  is then changed to  $s - \text{shift}$  in the previous equation. This allows small variations of the magnetic length.

Let  $F_E$  (respectively  $F_S$ ,  $F_L$ ) be the fringe field factor attached to the entrance (respectively exit, lateral) EFB. At any position on a trajectory the resulting value of the fringe field factor (eq. 6.3.18) is

$$\mathcal{F}(R, \theta) = F_E \times F_S \times F_L$$

In particular,  $F_L \equiv 1$  if no lateral EFB is requested.

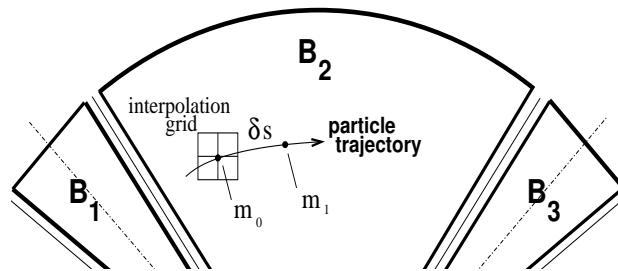


Figure 21: Interpolation method.  $m_0$  and  $m_1$  are the projections in the median plane of particle positions  $M_0$  and  $M_1$  and separated by  $\delta s$ , projection of the integration step.

### Calculation of the Mid-plane Field Derivatives

$B_Z(R, \theta)$  in Eq. 6.3.18 is computed at the  $n \times n$  nodes ( $n = 3$  or  $5$  in practice) of a “flying” interpolation grid in the median plane centered on the projection  $m_0$  of the actual particle position  $M_0$  as schemed in Fig. 21. A polynomial interpolation is involved, of the form

$$B_Z(R, \theta) = A_{00} + A_{10}\theta + A_{01}R + A_{20}\theta^2 + A_{11}\theta R + A_{02}R^2$$

that yields the requested derivatives, using

$$A_{kl} = \frac{1}{k!l!} \frac{\partial^{k+l} B_Z}{\partial \theta^k \partial r^l}$$

Note that, the source code contains the explicit analytical expressions of the coefficients  $A_{kl}$  solutions of the normal equations, so that the operation *is not* CPU time consuming.

### Extrapolation Off Median Plane

From the vertical field  $\vec{B}$  and derivatives in the median plane, first a transformation from polar to Cartesian coordinates is performed, following eqs (1.4.9 or 1.4.10), then, extrapolation off median plane is performed by means of Taylor expansions, following the procedure described in section 1.3.3.

**DIPOLE-M : Generation of dipole mid-plane 2-D map, polar frame**

(page 264 in PART B)

*DIPOLE-M* is a simpler and improved version of *AIMANT*.

The keyword *DIPOLE-M* provides an automatic generation of a dipole field map in polar coordinates. The extent of the map is defined by the following parameters, as shown in Figs. 9A and 9B.

- AT* : total angular aperture  
*RM* : mean radius used for the positioning of field boundaries  
*RMIN, RMAX* : minimum and maximum radii

The 2 or 3 effective field boundaries (EFB) inside the map are defined from geometric boundaries, the shape and position of which are determined by the following parameters.

- ACENT* : arbitrary inner angle, used for EFBs positioning  
 $\omega$  : azimuth of an EFB with respect to *ACENT*  
 $\theta$  : angle of an EFB with respect to its azimuth (wedge angle)  
 $R_1, R_2$  : radius of curvature of an EFB  
 $U_1, U_2$  : extent of the linear part of an EFB.

At any node of the map mesh, the value of the field is calculated as

$$B_Z(R, \theta) = \mathcal{F}(R, \theta) \times B_0 \times \left( 1 + N \times \left( \frac{R - RM}{RM} \right) + B \times \left( \frac{R - RM}{RM} \right)^2 + G \times \left( \frac{R - RM}{RM} \right)^3 \right) \quad (6.3.19)$$

where  $N$ ,  $B$  and  $G$  are respectively the first, second and third order field indices and  $\mathcal{F}$  is the fringe field factor.

**Calculation of the Fringe Field Factor**

With each EFB a realistic extent of the fringe field,  $\lambda$  (normally equal to the gap size), is associated and a fringe field factor  $F$  is calculated. In the following  $\lambda$  stands for either  $\lambda_E$  (Entrance),  $\lambda_S$  (Exit) or  $\lambda_L$  (Lateral EFB).

$F$  is an exponential type fringe field (Fig. 10) given by [48]

$$F = \frac{1}{1 + \exp P(s)}$$

where  $s$  is the distance to the EFB, and

$$P(s) = C_0 + C_1 \left( \frac{s}{\lambda} \right) + C_2 \left( \frac{s}{\lambda} \right)^2 + C_3 \left( \frac{s}{\lambda} \right)^3 + C_4 \left( \frac{s}{\lambda} \right)^4 + C_5 \left( \frac{s}{\lambda} \right)^5$$

It is also possible to simulate a shift of the *EFB*, by giving a non zero value to the parameter *shift*.  $s$  is then changed to  $s - \text{shift}$  in the previous equation. This allows small variations of the total magnetic length.

Let  $F_E$  (respectively  $F_S$ ,  $F_L$ ) be the fringe field factor attached to the entrance (respectively exit, lateral) EFB. At any node of the map mesh, the resulting value of the fringe field factor (eq. 6.3.19) is

$$\mathcal{F}(R, \theta) = F_E \times F_S \times F_L$$

In particular,  $F_L \equiv 1$  if no lateral EFB is requested.

**The Mesh of the Field Map**

The magnetic field is calculated at the nodes of a mesh with polar coordinates, in the median plane. The radial step is given by

$$\delta R = \frac{RMAX - RMIN}{IRMAX - 1}$$

and the angular step by

$$\delta\theta = \frac{AT}{IAMAX - 1}$$

where  $RMIN$  and  $RMAX$  are the lower and upper radial limits of the field map, and  $AT$  is its total angular aperture (Fig. 9B).  $IRMAX$  and  $IAMAX$  are the total number of nodes in the radial and angular directions.

### Simulating Field Defects and Shims

Once the initial map is calculated, it is possible to modify it by means of the parameter  $NBS$ , so as to simulate field defects or shims.

**If  $NBS = -2$ ,** the map is globally modified by a perturbation proportional to  $R - R_0$ , where  $R_0$  is an arbitrary radius, with an amplitude  $\Delta B_Z/B_0$ , so that  $B_Z$  at the nodes of the mesh is replaced by

$$B_Z \times \left( 1 + \frac{\Delta B_Z}{B_0} \frac{R - R_0}{RMAX - RMIN} \right)$$

**If  $NBS = -1$ ,** the perturbation is proportional to  $\theta - \theta_0$ , and  $B_Z$  is replaced by

$$B_Z \times \left( 1 + \frac{\Delta B_Z}{B_0} \frac{\theta - \theta_0}{AT} \right)$$

**If  $NBS \geq 1$ ,** then  $NBS$  shims are introduced at positions  $\frac{R_1 + R_2}{2}, \frac{\theta_1 + \theta_2}{2}$  (Fig. 12) [49]

The initial field map is modified by shims with second order profiles given by

$$\theta = \left( \gamma + \frac{\alpha}{\mu} \right) \beta \frac{X^2}{\rho^2}$$

where  $X$  is shown in Fig. 10,  $\rho = \frac{R_1 + R_2}{2}$  is the central radius,  $\alpha$  and  $\gamma$  are the angular limits of the shim,  $\beta$  and  $\mu$  are parameters.

At each shim, the value of  $B_Z$  at any node of the initial map is replaced by

$$B_Z \times \left( 1 + F\theta \times FR \times \frac{\Delta B_Z}{B_0} \right)$$

where  $F\theta = 0$  or  $FR = 0$  outside the shim, and  $F\theta = 1$  and  $FR = 1$  inside.

### Extrapolation Off Median Plane

The vector field  $\vec{B}$  and its derivatives in the median plane are calculated by means of a second or fourth order polynomial interpolation, depending on the value of the parameter  $IODRE$  ( $IODRE=2, 25$  or  $4$ , see section 1.4.2). The transformation from polar to Cartesian coordinates is performed following eqs (1.4.9 or 1.4.10). Extrapolation off median plane is then performed by means of Taylor expansions, following the procedure described in section 1.3.3.

**DIPOLLES : Dipole magnet  $N$ -tuple, polar frame [53, 54]**

(page 266 in PART B)

*DIPOLLES* works much like *DIPOLE* as to the field modelling, yet with the particularity that it allows positioning up to 5 such dipoles within the angular sector with full aperture  $AT$  thus allowing accounting for overlapping fringe fields. This is done in the following way<sup>10</sup>.

The dimensioning of the magnet is defined by

$AT$  : total angular aperture  
 $RM$  : mean radius used for the positioning of field boundaries

For each one of the  $N = 1$  to 5 dipoles of the  $N$ -tuple, the 2 effective field boundaries (entrance and exit EFBs) from which the dipole field (eqs. 6.3.20, 6.3.22) is drawn are defined from geometrical boundaries, the shape and position of which are determined by the following parameters (in the same manner as in *DIPOLE*, *DIPOLE-M*) (see Fig. 20, p. 99, and Fig. 22)

$ACN_i$  : arbitrary inner angle, used for EFBs positioning  
 $\omega$  : azimuth of an EFB with respect to  $ACN$   
 $\theta$  : angle of an EFB with respect to its azimuth (wedge angle)  
 $R_1, R_2$  : radius of curvature of an EFB  
 $U_1, U_2$  : extent of the linear part of an EFB

#### Calculation of the Field From a Single Dipole

The magnetic field is calculated in polar coordinates. At all  $(R, \theta)$  in the median plane ( $Z = 0$ ), the magnetic field due a single one (index  $i$ ) of the dipoles of a  $N$ -tuple magnet can take one of the forms below, upon option,



$$(i) \quad B_{Z_i}(R, \theta) = B_{Z_{0,i}} \mathcal{F}_i(R, \theta) (1 + b_{1_i}(R - RM_i)/RM_i + b_{2_i}(R - RM_i)^2/RM_i^2 + \dots) \quad (6.3.20)$$

$$(ii) \quad B_Z(R, \theta) = B_{Z_{0,i}} + \sum_{i=1}^N \mathcal{F}_i(R, \theta) (b_{1_i}(R - RM_i) + b_{2_i}(R - RM_i)^2 + \dots) \quad (6.3.21)$$

(iii) Sandro Ruggiero's FFAG style of field, experimental, see routine `dipsi.f`

wherein  $B_{Z_{0,i}}$  is a reference field, at reference radius  $RM_i$ , and  $\mathcal{F}_i(R, \theta)$  is a fringe field factor, see below. This field model is proper to simulate for instance chicane dipoles, cyclotron or FFAG magnets, etc.

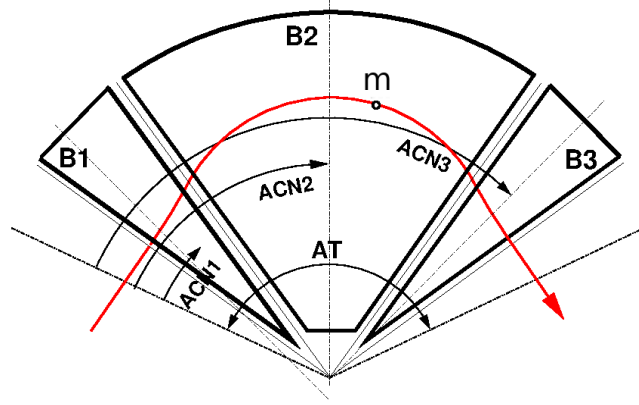


Figure 22: Definition of a dipole triplet using the *DIPOLES* or *FFAG* procedures.  $m$  is the projection of the particle position in the median plane, the red curve is the projected trajectory

### Calculation of the Fringe Field Factor

In a dipole, a realistic extent of the fringe field,  $g$ , is associated with each EFB, and a fringe field factor  $F$  is calculated.

$F$  is an exponential type fringe field (Fig. 10, page 74) given by [48]

$$F = \frac{1}{1 + \exp P(d)}$$

wherein  $d$  is the distance to the EFB and depends on  $(R, \theta)$ , and

$$P(d) = C_0 + C_1 \left(\frac{d}{g}\right) + C_2 \left(\frac{d}{g}\right)^2 + C_3 \left(\frac{d}{g}\right)^3 + C_4 \left(\frac{d}{g}\right)^4 + C_5 \left(\frac{d}{g}\right)^5$$

In addition,  $g$  is made dependent of  $R$  (a way to simulate the effect of variable gap size on fringe field extent), under the form

$$g(R) = g_0 (RM/R)^\kappa$$

This dependence is accounted for rigorously if the interpolation method (see below) is used, or else to order zero (derivatives of  $g(R)$  are not considered) if the analytic method (below) is used.

Let  $F_E$  (respectively  $F_S$ ) be the fringe field factor attached to the entrance (respectively exit) EFB ; at any position on a trajectory the resulting value of the fringe field factor is taken to be

$$\mathcal{F}_i(R, \theta) = F_E \times F_S \quad (6.3.22)$$

### Calculation of the Field Resulting From all $N$ Dipoles

Now, accounting for  $N$  neighboring dipoles in an  $N$ -tuple, the mid-plane field and field derivatives are obtained by addition of the contributions of the  $N$  dipoles taken separately, namely

<sup>10</sup>FFAG can be referred to as another instance of a procedure based on such method.

$$B_Z(R, \theta) = \sum_{i=1, N} B_{Zi}(R, \theta) \quad (6.3.23)$$

$$\frac{\partial^{k+l} \vec{B}_Z(R, \theta)}{\partial \theta^k \partial r^l} = \sum_{i=1, N} \frac{\partial^{k+l} \vec{B}_{Zi}(R, \theta)}{\partial \theta^k \partial r^l} \quad (6.3.24)$$

Note that, in doing so it is not meant that field superposition does apply in reality, it is just meant to provide the possibility of obtaining a realistic field shape, that would for instance closely match (using appropriate  $C_0 - C_5$  sets of coefficients) 3-D field simulations obtained from magnet design codes.

### Calculation of the Mid-plane Field Derivatives

Two methods have been implemented to calculate the field derivatives in the median plane (Eq. 6.3.23), based on either analytical expressions derived from the magnet geometrical description, or classical numerical interpolation.

The first method has the merit of insuring best symplecticity in principle and fastest tracking. The interest of the second method is in its facilitating possible changes in the mid-plane magnetic field model  $B_Z(R, \theta)$ , for instance if simulations of shims, defects, or special  $R, \theta$  field dependence need to be introduced.

*Analytical method [55] :*

The starting ingredients are, on the one hand distances to the EFBs,

$$d(R, \theta) = \sqrt{(x(R, \theta) - x_0(R, \theta))^2 + (y(R, \theta) - y_0(R, \theta))^2}$$

to be computed for the two cases  $d_{\text{Entrance}}$ ,  $d_{\text{Exit}}$ , and on the other hand the expressions of the coordinates of particle position  $M$  and its projection  $P$  on the EFB in terms of the magnet geometrical parameters, namely

$$\begin{aligned} x(R, \theta) &= \cos(ACN - \theta) - RM \\ y(R, \theta) &= R \sin(ACN - \theta) \\ x_P(R, \theta) &= \sin(u) (y(R, \theta) - y_b)/2 + x_b \sin^2(u) + x(R, \theta) \cos^2(u) \\ y_P(R, \theta) &= \sin(u) (x(R, \theta) - x_b)/2 + y_b \cos^2(u) + y(R, \theta) \sin^2(u) \end{aligned}$$

with  $x_b$ ,  $y_b$ ,  $u$  parameters drawn from the magnet geometry (sector angle, wedge angle, face curvatures, etc.).

These ingredients allow calculating the derivatives  $\frac{\partial^{u+v} x(R, \theta)}{\partial \theta^u \partial r^v}$ ,  $\frac{\partial^{u+v} y(R, \theta)}{\partial \theta^u \partial r^v}$ ,  $\frac{\partial^{u+v} x_0(R, \theta)}{\partial \theta^u \partial r^v}$ ,  $\frac{\partial^{u+v} y_0(R, \theta)}{\partial \theta^u \partial r^v}$ , which, in turn, intervene in the derivatives of the compound functions  $\frac{\partial^{u+v} F(R, \theta)}{\partial \theta^u \partial r^v}$ ,  $\frac{\partial^{u+v} p(R, \theta)}{\partial \theta^u \partial r^v}$ ,  $\frac{\partial^{u+v} d(R, \theta)}{\partial \theta^u \partial r^v}$ .

*Interpolation method :*

The expression  $B_Z(R, \theta)$  in Eq. 6.3.23 is, in this case, computed at the  $n \times n$  nodes ( $n = 3$  or  $5$  in practice) of a “flying” interpolation grid in the median plane centered on the projection  $m_0$  of the actual particle position  $M_0$  as schemed in Fig. 23. A polynomial interpolation is involved, of the form

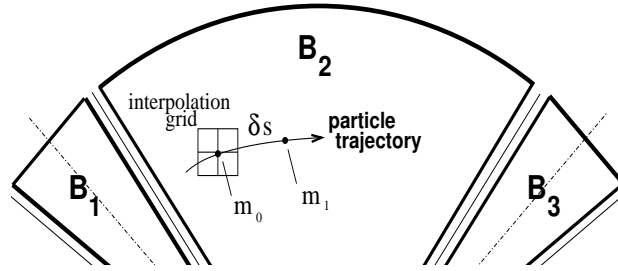


Figure 23: Interpolation method.  $m_0$  and  $m_1$  are the projections in the median plane of particle positions  $M_0$  and  $M_1$  and separated by  $\delta s$ , projection of the integration step.

$$B_Z(R, \theta) = A_{00} + A_{10}\theta + A_{01}R + A_{20}\theta^2 + A_{11}\theta R + A_{02}R^2$$

that yields the requested derivatives, using

$$A_{kl} = \frac{1}{k!l!} \frac{\partial^{k+l} B}{\partial \theta^k \partial R^l}$$

Note that, the source code contains the explicit analytical expressions of the coefficients  $A_{kl}$  solutions of the normal equations, so that the operation *is not* CPU time consuming.

#### Extrapolation Off Median Plane

From the vertical field  $\vec{B}$  and derivatives in the median plane, first a transformation from polar to Cartesian coordinates is performed, following eqs (1.4.9 or 1.4.10), then, extrapolation off median plane is performed by means of Taylor expansions, following the procedure described in section 1.3.3.

#### Sharp Edge

Sharp edge field fall-off at a field boundary can only be simulated if the following conditions are fulfilled :

- entrance (resp. exit) field boundary coincides with entrance (resp. exit) dipole limit (it means in particular, see Fig. 20,  $\omega^+ = ACENT$  (resp.  $\omega^- = -(AT - ACENT)$ ), together with  $\theta = 0$  at entrance (resp. exit) EFBs),
- analytical method for calculation of the mid-plane field derivatives is used.

**DODECAPO : Dodecapole magnet (Fig. 24)**

The meaning of parameters for *DODECAPO* is the same as for *QUADRUPO*.

In fringe field regions the magnetic field  $\vec{B}(X, Y, Z)$  and its derivatives up to fourth order are derived from the scalar potential approximated to the 6th order in  $Y$  and  $Z$

$$V(X, Y, Z) = G(X) \left( Y^4 - \frac{10}{3} Y^2 Z^2 + Z^4 \right) Y Z$$

$$\text{with } G_0 = \frac{B_0}{R_0^5}$$

The modelling of the fringe field form factor  $G(X)$  is described under *QUADRUPO*, p. 136.

Outside fringe field regions, or everywhere in sharp edge dodecapole ( $\lambda_E = \lambda_S = 0$ ),  $\vec{B}(X, Y, Z)$  in the magnet is given by

$$B_X = 0$$

$$B_Y = G_0(5Y^4 - 10Y^2Z^2 + Z^4)Z$$

$$B_Z = G_0(Y^4 - 10Y^2Z^2 + 5Z^4)Y$$

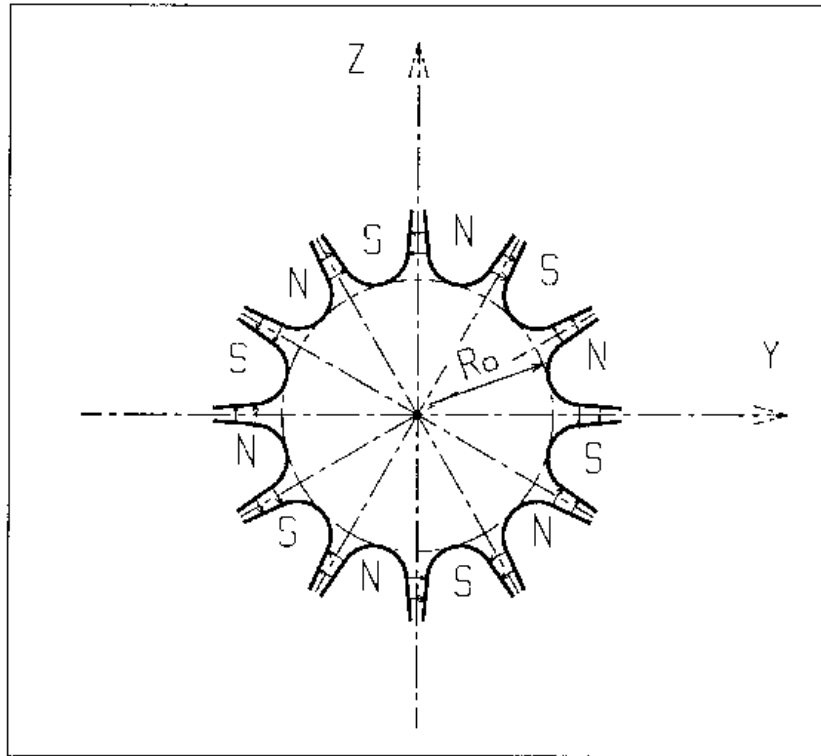


Figure 24: Dodecapole magnet

**DRIFT or ESL : Field free drift space**

*DRIFT* or *ESL* allow introduction of a drift space with length  $XL$  with positive or negative sign, anywhere in an optical sequence. This is an exact transport: the associated equations of motion are (Fig. 25)

$$Y_2 = Y_1 + XL \times \operatorname{tg} T$$

$$Z_2 = Z_1 + \frac{XL}{\cos T} \operatorname{tg} P$$

$$SAR_2 = SAR_1 + \frac{XL}{\cos T \times \cos P}$$

whereas the velocity vector is left unchanged.

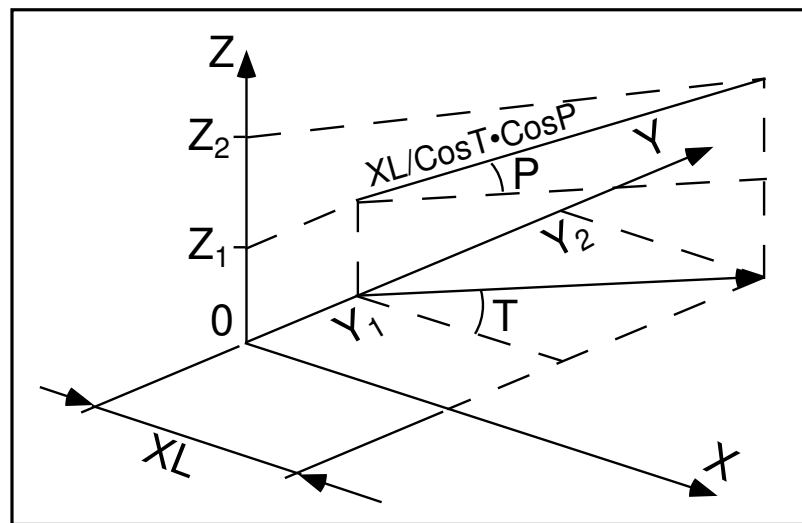


Figure 25: Transport of particles in a drift space.

The flag '.plt' can be added as a *LABEL2* (second label) following *DRIFT* keyword. It causes local coordinates to be logged to zgoubi.res or to zgoubi.plt (in the same manner as *IL=1*, or *2*, respectively, does in the case optical elements).

A drift can be split into pieces, using the 'split' flag. This may be used for graphic purposes for instance: a sub-product of 'split' is its allowing using the *IL* flag, so logging coordinates to either zgoubi.res or to zgoubi.plt (in the same manner as *IL=1* or *2*, respectively, does in optical elements, step by step).

**EBMULT : Electro-magnetic multipole**

(page 269 in PART B)

*EBMULT* simulates an electro-magnetic multipole, by addition of electric ( $\vec{E}$ ) and magnetic ( $\vec{B}$ ) multipole components (dipole to 20-pole).  $\vec{E}$  and its derivatives  $\frac{\partial^{i+j+k}\vec{E}}{\partial X^i \partial Y^j \partial Z^k}$  ( $i + j + k \leq 4$ ) are derived from the general expression of the multipole scalar potential (eq. 1.3.5), followed by a  $\frac{\pi}{2n}$  rotation ( $n = 1, 2, 3, \dots$ ) (see also *ELMULT*).  $\vec{B}$  and its derivatives are derived from the same general potential, as described in section 1.3.7 (see also *MULTIPOL*).

The entrance and exit fringe fields of the  $\vec{E}$  and  $\vec{B}$  components are treated separately, in the same way as described under *ELMULT* and *MULTIPOL*, for each one of these two fields. Wedge angle correction is applied in sharp edge field model if  $\vec{B}1$  is non zero, as in *MULTIPOL*. Any of the  $\vec{E}$  or  $\vec{B}$  multipole field component can be *X*-rotated independently of the others.

Use *PARTICUL* prior to *EBMULT*, for the definition of particle mass and charge.

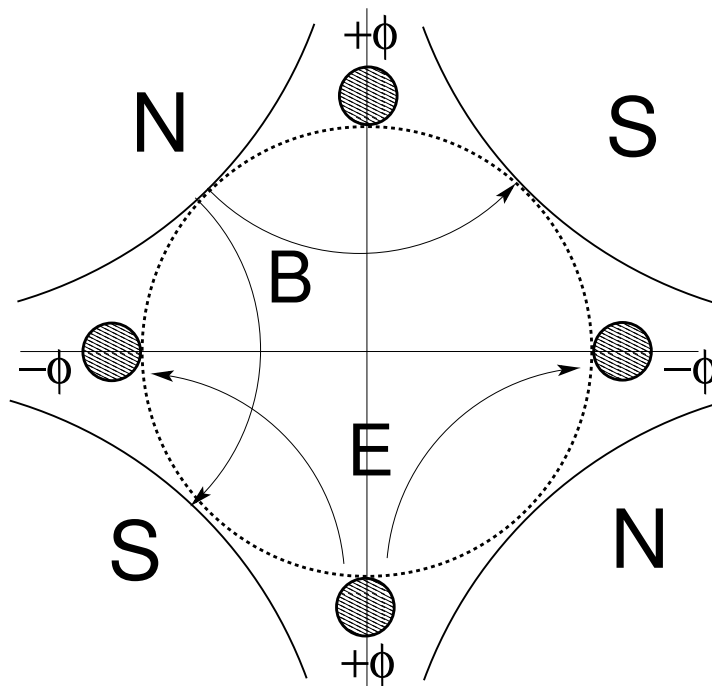


Figure 26: An example of an  $\vec{E}$ ,  $\vec{B}$  multipole : the achromatic quadrupole (known for its allowing null second order chromatic aberrations [56]).

**EL2TUB : Two-tube electrostatic lens**

The lens is cylindrically symmetric about the  $X$ -axis.

The length and potential of the first (resp. second) electrode are  $X1$  and  $V1$  ( $X2$  and  $V2$ ). The distance between the two electrodes is  $D$ , and their inner radius is  $R_0$  (Fig. 27). The model for the electrostatic potential along the axis is [57]

$$V(X) = \frac{V_2 - V_1}{2} \operatorname{th} \frac{\omega x}{R_0} \left[ + \frac{V_1 + V_2}{2} \right] \quad \text{if } D \rightarrow 0$$

$$V(X) = \frac{V_2 - V_1}{2} \frac{1}{2\omega D/R_0} \ln \frac{\operatorname{ch} \omega \frac{x+D}{R_0}}{\operatorname{ch} \omega \frac{x-D}{R_0}} \left[ + \frac{V_1 + V_2}{2} \right] \quad \text{if } D \neq 0$$

where  $x$  = distance from half-way between the electrodes ;  $\omega = 1.318$  ; th = hyperbolic tangent ; ch = hyperbolic cosine. The second form ( $D \neq 0$ ) assumes that  $D$  is large enough that the field fall-offs from the two lenses do not overlap. An upstream end lens (respectively downstream) is modeled using  $V_1 = 0$  (resp.  $V_2 = 0$ ).

From these expressions the field  $\vec{E}(X, Y, Z)$  and its derivatives are derived following the procedure described in section 1.3.1 (note that they don't depend on the constant term  $\left[ \frac{V_1 + V_2}{2} \right]$  which disappears when differentiating).

Use *PARTICUL* prior to *EL2TUB*, for the definition of particle mass and charge.

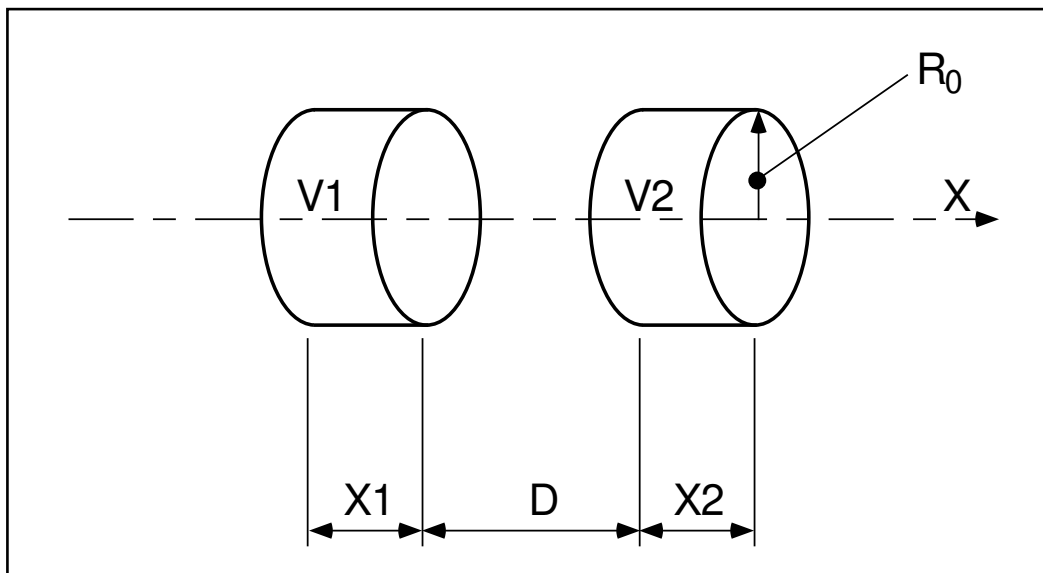


Figure 27: Two-electrode cylindrical electric lens.

**ELCYLDEF : Electrostatic cylindrical deflector**

(page 272 in PART B)

*TO BE DOCUMENTED.*



**ELMIR : Electrostatic N-electrode mirror/lens, straight slits**

(page 273 in PART B)

The device works as mirror or lens, horizontal or vertical. It is comprised of  $N = 2$  up to 7 parallel-plate electrodes. Mid-plane symmetry is assumed.

Electrode lengths are  $L1, L2, \dots, LN$ .  $D$  is the mirror/lens gap. The model for the  $Y$ -independent electrostatic potential is (after Ref. [58, p.412])

$$V(X, Z) = \sum_{i=2}^N \frac{V_i - V_{i-1}}{\pi} \arctan \frac{\sinh(\pi(X - X_{i-1})/D)}{\cos(\pi Z/D)}$$

where  $V_i$  are the potential at the  $N$  electrodes (and normally  $V1 = 0$  refers to the incident beam energy),  $X_i$  are the locations of the zero-length slits,  $X$  is the distance from the origin taken at the first slit (located at  $X1 \equiv 0$  between the first and second electrodes). From  $V(X, Z)$  the field  $\vec{E}(X, Y, Z)$  and derivatives are deduced following the procedure described in section 1.3.7 (page 30).

The total X-extent of the mirror/lens is  $L = \sum_{i=1}^N L_i$ .

In the mirror mode (option  $MT = 11$  for vertical mid-plane or  $MT = 12$  for horizontal mid-plane) stepwise integration starts at  $X = -L1$  (entrance of the first electrode) and terminates either when back to  $X = -L1$  or when reaching  $X = L - L1$  (end of the  $N - th$  electrode). In the latter case particles are stopped with their index  $IEX$  set to  $-8$  (see section 7.13 on page 220). Normally  $X1$  should exceed  $3D$  (enough that  $V(X < X1)$  have negligible effect in terms of trajectory behavior).

In the lens mode (option flag  $MT = 21$  for vertical mid-plane or  $MT = 22$  for horizontal mid-plane) stepwise integration starts at  $X = -L1$  (entrance of the first electrode) and terminates either when reaching  $X = L - L1$  (end of the  $N - th$  electrode) or when the particle deflection exceeds  $\pi/2$ . In the latter case the particle is stopped with their index  $IEX$  set to  $-3$ .

Use *PARTICUL* prior to *ELMIR*, for the definition of particle mass and charge.

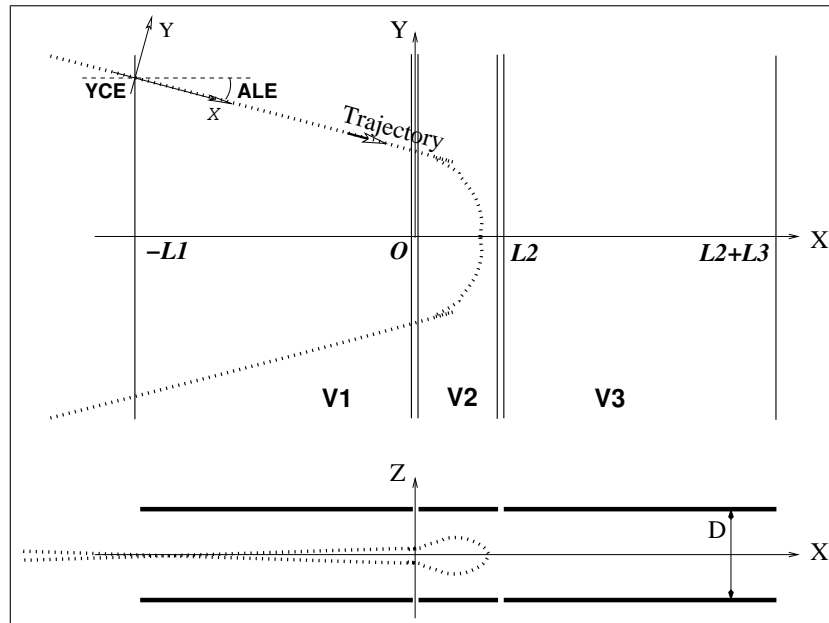


Figure 28: Electrostatic N-electrode mirror/lens, straight slits, in the case  $N = 3$ , in horizontal mirror mode ( $MT = 11$ ).

Possible non-zero entrance quantities  $YCE$ ,  $ALE$  should be specified using *CHANGREF*, or using  $KPOS=3$  with  $YCE$  and  $ALE$ =half-deviation matched to the reference trajectory.

**ELMIRC : Electrostatic N-electrode mirror/lens, circular slits**

(page 273 in PART B)

The device works as mirror or lens, horizontal or vertical. It is comprised of  $N = 2$  up to 7 parallel-plate electrodes. Mid-plane symmetry is assumed<sup>11</sup>.

Electrode slits are circular, concentric with radii  $R_1, R_2, \dots, R_{N-1}$ ,  $D$  is the mirror/lens gap. The model for the mid-plane ( $Z = 0$ ) radial electrostatic potential is (after Ref. [58, p.443])

$$V(r) = \sum_{i=2}^N \frac{V_i - V_{i-1}}{\pi} \arctan \left( \sinh \frac{\pi(r - R_{i-1})}{D} \right)$$

where  $V_i$  are the potential at the  $N$  electrodes (and normally  $V_1 = 0$  refers to the incident beam energy).  $r$  is the current radius.

The mid-plane field  $\vec{E}(r)$  and its  $r$ -derivatives are first derived by differentiation, then  $\vec{E}(r, Z)$  and derivatives are obtained from Taylor expansions and Maxwell relations. Eventually a transformation to the rotating frame provides  $\vec{E}(X, Y, Z)$  and derivatives as involved in eq. 1.2.15.

Stepwise integration starts at entrance (defined by  $RE, TE$ ) of the first electrode and terminates when rotation of the reference rotating frame ( $RM, X, Y$ ) has reached the value  $AT$ . Normally,  $R_1 - RE$  and  $R_1 - RS$  should both exceed  $3D$  (so that potential tails have negligible effect in terms of trajectory behavior).

Positioning of the element is performed by means of  $KPOS$  (see section 7.10).

Use  $PARTICUL$  prior to  $ELMIRC$ , for the definition of particle mass and charge.

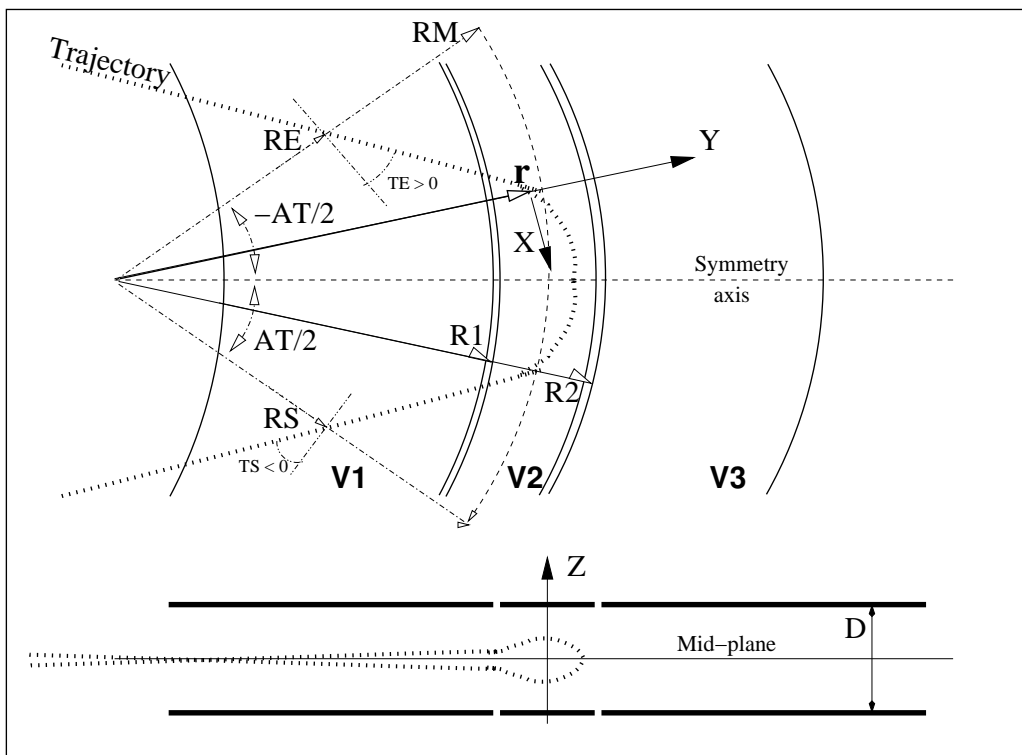


Figure 29: Electrostatic N-electrode mirror/lens, circular slits, in the case  $N = 3$ , in horizontal mirror mode.

<sup>11</sup>NOTE : in the present version of the code, the sole horizontal mirror mode is operational, and  $N$  is limited to 3.

**ELMULT : Electric multipole**

The simulation of multipolar electric field  $\vec{M}_E$  proceeds by addition of the dipolar ( $\vec{E}1$ ), quadrupolar ( $\vec{E}2$ ), sextupolar ( $\vec{E}3$ ), etc., up to 20-polar ( $\vec{E}10$ ) components, and of their derivatives up to fourth order, following

$$\begin{aligned}\vec{M}_E &= \vec{E}1 + \vec{E}2 + \vec{E}3 + \dots + \vec{E}10 \\ \frac{\partial \vec{M}_E}{\partial X} &= \frac{\partial \vec{E}1}{\partial X} + \frac{\partial \vec{E}2}{\partial X} + \frac{\partial \vec{E}3}{\partial X} + \dots + \frac{\partial \vec{E}10}{\partial X} \\ \frac{\partial^2 M_E}{\partial X \partial Z} &= \frac{\partial^2 \vec{E}1}{\partial X \partial Z} + \frac{\partial^2 \vec{E}2}{\partial X \partial Z} + \frac{\partial^2 \vec{E}3}{\partial X \partial Z} + \dots + \frac{\partial^2 \vec{E}10}{\partial X \partial Z} \\ &\text{etc.}\end{aligned}$$

The independent components  $\vec{E}1$  to  $\vec{E}10$  and their derivatives up to the fourth order are calculated by differentiating the general multipole potential given in eq. 1.3.5 (page 31), followed by a  $\frac{\pi}{2n}$  rotation about the  $X$ -axis, so that the so defined right electric multipole of order  $n$ , and of strength [56, 59]

$$K_n = \frac{1}{2} \frac{\gamma}{\gamma^2 - 1} \frac{V_n}{R_0^n}$$

( $V_n$  = potential at the electrode,  $R_0$  = radius at pole tip,  $\gamma$  = relativistic Lorentz factor of the particle) has the same focusing effect as the right magnetic multipole of order  $n$  and strength  $K_n = \frac{B_n}{R_0^{n-1} B\rho}$  ( $B_n$  = field at pole tip,  $B\rho$  = particle rigidity, see *MULTIPOL*).

The entrance and exit fringe fields are treated separately. They are characterized by the integration zone  $X_E$  at entrance and  $X_S$  at exit, as for *QUADRUPO*, and by the extent  $\lambda_E$  at entrance,  $\lambda_S$  at exit. The fringe field extents for the dipole component are  $\lambda_E$  and  $\lambda_S$ . The fringe field extent for the quadrupolar (sextupolar, ..., 20-polar) component is given by a coefficient  $E_2$  ( $E_3$ , ...,  $E_{10}$ ) at entrance, and  $S_2$  ( $S_3$ , ...,  $S_{10}$ ) at exit, such that the fringe field extent is  $\lambda_E \times E_2$  ( $\lambda_E \times E_3$ , ...,  $\lambda_E \times E_{10}$ ) at entrance and  $\lambda_S \times S_2$  ( $\lambda_S \times S_3$ , ...,  $\lambda_S \times S_{10}$ ) at exit.

If  $\lambda_E = 0$  ( $\lambda_S = 0$ ) the multipole lens is considered to have a sharp edge field at entrance (exit), and then,  $X_E$  ( $X_S$ ) is forced to zero (for the mere purpose of saving computing time).

If  $E_i = 0$  ( $S_i = 0$ ) ( $i = 2, 10$ ), the entrance (exit) fringe field for multipole component  $i$  is considered as a sharp edge field.

Any multipole component  $\vec{E}i$  can be rotated independently by an angle  $RXi$  around the longitudinal  $X$ -axis, for the simulation of positioning defects, as well as skew lenses.

Use *PARTICUL* prior to *ELMULT*, for the definition of particle mass and charge.

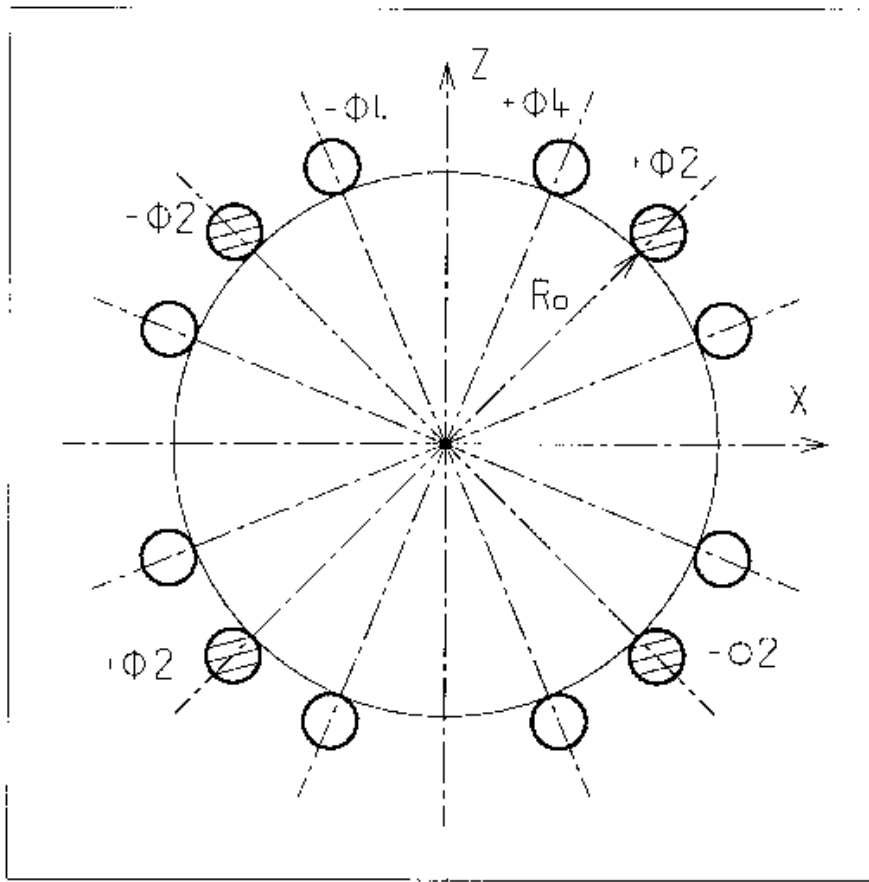


Figure 30: An electric multipole combining skew-quadrupole ( $\vec{E}_2 \neq \vec{0}$ ,  $\vec{R}_2 = \pi/4$ ) and skew-octupole ( $\vec{E}_4 \neq \vec{0}$ ,  $\vec{R}_4 = \pi/8$ ) components ( $\vec{E}_1 = \vec{E}_3 = \vec{E}_5 = \dots = \vec{E}_{10} = \vec{0}$ ) [59].

**ELREVOL : 1-D uniform mesh electric field map**

*ELREVOL* reads a 1-D axial field map from a storage data file, whose content must fit the following *FORTRAN* reading sequence

```

OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD' [,FORM='UNFORMATTED'])
DO 1 I=1, IX
  IF (BINARY) THEN
    READ (NL) X(I), EX(I)
  ELSE
    READ (NL, *) X(I), EX(I)
  ENDIF
1   CONTINUE

```

where  $IX$  is the number of nodes along the (symmetry)  $X$ -axis,  $X(I)$  their coordinates, and  $EX(I)$  are the values of the  $X$  component of the field.  $EX$  is normalized with  $ENORM$  prior to ray-tracing. As well the longitudinal coordinate  $X$  is normalized with a  $XNORM$  coefficient (useful to convert to centimeters, the working units in **zgoubi**).

$X$ -cylindrical symmetry is assumed, resulting in  $EY$  and  $EZ$  taken to be zero on axis.  $\vec{E}(X, Y, Z)$  and its derivatives along a particle trajectory are calculated by means of a 5-points polynomial interpolation followed by second order off-axis extrapolation (see sections 1.3.1 and 1.4.1).

Entrance and/or exit integration boundaries may be defined in the same way as in *CARTEMES* by means of the flag  $ID$  and coefficients  $A, B, C, A', B', C'$ .

Use *PARTICUL* prior to *ELREVOL*, for the definition of particle mass and charge.

Including 'FLIP' in *TITL* causes the field map to be X-extended using its X-flipped copy.

**EMMA : 2-D Cartesian or cylindrical mesh field map for EMMA FFAG**

*EMMA* is dedicated to the reading and treatment of 2-D or 3-D Cartesian mesh field maps representing the EMMA FFAG cell quadrupole doublet<sup>12</sup> [62, 63].

*EMMA* can sum up independent field maps of each of the two quadrupoles, with each its scaling coefficient. The two maps can be radially positioned independently of one another at  $Y_F$ ,  $Y_D$  respectively, just like the actual *EMMA* quadrupoles. In particular,

*MOD* : operational and map *FORMAT* reading mode ;

*MOD* ≤ 19 : Cartesian mesh ;

*MOD* ≥ 20 : cylindrical mesh.

*MOD*=0 : two 2D maps, one representing QF, one representing QD. A single map, superimposition of both, is built prior to tracking and used for tracking.

*MOD*=1 : two 2D maps, one representing QF, one representing QD, a resulting single map is devised in the following way : QF<sub>new</sub> is interpolated from QF with  $dr=x_F$ , QD<sub>new</sub> is interpolated from QD with  $dr=x_D$ . A single map, superimposition of both, is built prior to tracking and used for tracking.

The parameters that move/position the maps, as  $(Y_F, Y_D)$ , are accessible from the FIT, allowing to adjust the cell tunes.

*EMMA* works much like *TOSCA*. Refer to that keyword, and to the *FORTRAN* file `emma_c.f`, for details.

<sup>12</sup>The stepwise ray-tracing code *Zgoubi* is the on-line model code for the world's first non-scaling FFAG experiment.

**FFAG : FFAG magnet,  $N$ -tuple [53, 54]**

(page 280 in PART B)

smallskip

*FFAG* works much like *DIPOLLES* as to the field modelling, apart from the radial dependence of the field,  $B = B_0(r/r_0)^k$ , so-called “scaling”. Note that *DIPOLLES* does similar job by using a Taylor  $r$ -expansion of  $B_0(r/r_0)^k$ .

smallskip

The *FFAG* procedure allows overlapping of fringe fields of neighboring dipoles, thus simulating in some sort the field in a dipole  $N$ -tuple - as for instance in an FFAG doublet or triplet. A detailed application, with five dipoles, can be found in Ref. [53]. This is done in the way described below.

smallskip

The dimensioning of the magnet is defined by

smallskip

$AT$  : total angular aperture

$RM$  : mean radius used for the positioning of field boundaries

smallskip

For each one of the  $N = 1$  to (maximum) 5 dipoles of the  $N$ -tuple, the two effective field boundaries (entrance and exit EFBs) from which the dipole field is drawn are defined from geometric boundaries, the shape and position of which are determined by the following parameters (in the same manner as in *DIPOLE*, *DIPOLE-M*) (see Fig. 20 page 99, and Fig. 31)

smallskip

- $ACN_i$  : arbitrary inner angle, used for EFBs positioning  
 $\omega$  : azimuth of an EFB with respect to  $ACN$   
 $\theta$  : angle of an EFB with respect to its azimuth (wedge angle)  
 $R_1, R_2$  : radius of curvature of an EFB  
 $U_1, U_2$  : extent of the linear part of an EFB

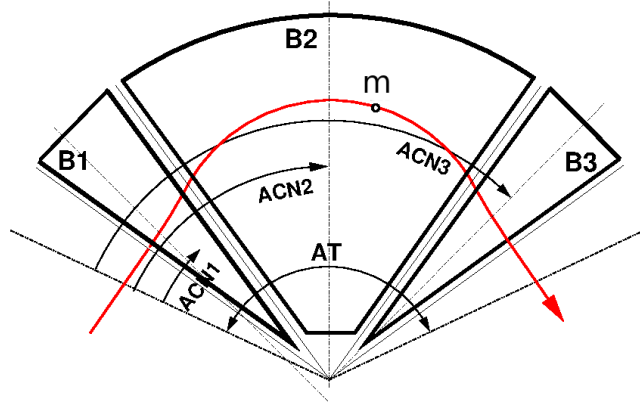


Figure 31: Definition of a dipole  $N$ -tuple ( $N = 3$ , a triplet here) using the *DIPOLLES* or *FFAG* procedures.  $m$  is the projection of the particle position in the median plane, the red curve is the projected trajectory

#### Calculation of the Field From a Single Dipole

The magnetic field is calculated in polar coordinates. At all  $(R, \theta)$  in the median plane ( $z = 0$ ), the magnetic field due a single one (index  $i$ ) of the dipoles of a  $N$ -tuple FFAG magnet is written

$$B_{Z_i}(R, \theta) = B_{Z_{0,i}} \mathcal{F}_i(R, \theta) (R/R_M)^{K_i} \quad (6.3.25)$$

wherein  $B_{Z_{0,i}}$  is a reference field, at reference radius  $RM_i$ , whereas  $\mathcal{F}(R, \theta)$  is calculated as described below.

#### Calculation of $\mathcal{F}_i(R, \theta)$

The fringe field factor  $\mathcal{F}_i(R, \theta)$  associated with a dipole is computed as in the procedure *DIPOLLES* (eq. 6.3.22), including (rigorously if the interpolation method is used, see page 106, or to order zero if the analytic method is used, see page 106) radial dependence of the gap size

$$g(R) = g_0 (RM/R)^\kappa \quad (6.3.26)$$

so to simulate the effect of gap shaping on  $B_{Z_i}(R, \theta)|_R$  field fall-off, over the all radial extent of a scaling FFAG dipole (with normally - but not necessarily in practice -  $\kappa \approx K_i$ ).

#### Calculation of the Field Resulting From All $N$ Dipoles

For the rest, namely, calculation of the full field at particle position from the  $N$  dipoles, analytical calculation or numerical interpolation of the mid-plane field derivatives, extrapolation off median plane, etc., things are performed exactly as in the case of the *DIPOLLES* procedure (see page 105).

#### Sharp Edge

Sharp edge field fall-off at a field boundary can only be simulated if the following conditions are fulfilled :



- entrance (resp. exit) field boundary coincides with entrance (resp. exit) dipole limit (it means in particular, see Fig. 20,  $\omega^+ = ACENT$  (resp.  $\omega^- = -(AT - ACENT)$ ), together with  $\theta = 0$  at entrance (resp. exit) EFBs),
- analytical method for calculation of the mid-plane field derivatives is used.

**FFAG-SPI : Spiral FFAG magnet,  $N$ -tuple [54, 60]**

*FFAG-SPI* works much like *FFAG* as to the field modelling, with essentially a different axial dependence.

The *FFAG-SPI* procedure allows overlapping of fringe fields of neighboring dipoles, thus simulating in some sort the field in a dipole  $N$ -tuple (similar to Fig. 31, page 120). This allows for instance accounting for fringe field effects, or clamps, as schemed in Fig. 32.

The dimensioning of the magnet is defined by

$AT$  : total angular aperture

$RM$  : mean radius used for the positioning of field boundaries

For each one of the  $N = 1$  to (maximum) 5 dipoles of the  $N$ -tuple, the two effective field boundaries (entrance and exit EFBs) from which the dipole field is drawn are defined from geometric boundaries, the shape and position of which are determined by the following parameters

$ACN_i$  : arbitrary inner angle, used for EFBs positioning

$\omega$  : azimuth of an EFB with respect to  $ACN$

$\xi$  : spiral angle

with  $ACN_i$  and  $\omega$  as defined in Fig. 32 (similar to what can be found in Figs. 31 and 20).

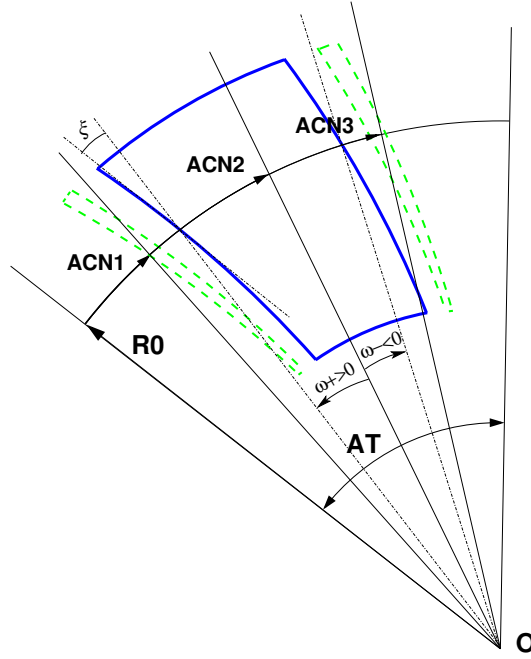


Figure 32: A  $N$ -tuple spiral sector FFAG magnet ( $N = 3$  here, simulating active field clamps at entrance and exit side of a central dipole).

### Calculation of the Field From a Single Dipole

The magnetic field is calculated in polar coordinates. At all  $(R, \theta)$  in the median plane ( $Z = 0$ ), the magnetic field due a single one (index  $i$ ) of the dipoles of a  $N$ -tuple spiral FFAG magnet is written

$$B_{Z_i}(R, \theta) = B_{Z_{0,i}} \mathcal{F}_i(R, \theta) (R/R_M)^{K_i} \quad (6.3.27)$$

wherein  $B_{Z_{0,i}}$  is a reference field, at reference radius  $RM_i$ , whereas  $\mathcal{F}(R, \theta)$  is calculated as described below.

### Calculation of $\mathcal{F}_i(R, \theta)$

The fringe field factor  $\mathcal{F}_i(R, \theta)$  associated with a dipole is computed as in the procedure *DIPOLLES* (eq. 6.3.22), including radial dependence of the gap size

$$g(R) = g_0 (RM/R)^\kappa \quad (6.3.28)$$

so to simulate the effect of gap shaping on  $B_{Z_i}(R, \theta)|_R$  field fall-off, over the all radial extent of the dipole (with normally - yet not necessarily in practice -  $\kappa \approx -1$ ).

In the case  $\kappa = -1$ , the scaling condition for sector spiral rings is satisfied. In that case the fringe field function may be transformed to

$$\mathcal{F}(R, \theta) = \mathcal{F}(\psi)$$

where

$$\psi = \theta - \tan \xi \ln \frac{R}{RM}$$

is the *generalised* azimuthal angle. The distance between spirals is constant in terms of  $\psi$ . The fringe field extent  $g(R)$  can be converted into an angular extent  $\Delta\psi_g$  via,

$$\Delta\psi_g = \frac{g_0}{RM \cos \xi}$$

Then the fringe field dependence is given by  $P(\Delta\psi_d/\Delta\psi_g)$  where  $\Delta\psi_d$  is evaluated at the current particle location, replacing the  $P(d/g)$  described in the *DIPOL* procedure. On the other hand, if  $\kappa \neq -1$ , the scaling condition is not fulfilled and so the distance to the magnet edge  $d$  must be found numerically. At the moment, the generalised azimuthal angle approach is applied in the code only if the flying mesh interpolation method is used. The numerical method can be forced by including the DNEWT flag (see page 281).

#### Calculation of the Full Field From All $N$ Dipoles

For the rest, namely calculation of the full field at particle position, as resulting from the  $N$  dipoles, calculation of the mid-plane field derivatives, extrapolation off median plane, etc., things are performed in the same manner as for the *DIPOL* procedure (see page 105).

**HALBACH : Halbach style permanent magnet**

\*\*\*\*\* UNDER DEVELOPMENT \*\*\*\*\*

*HALBACH* generates magnetic field from a Halbach style permanent magnet [65].

Fig. 3 is an example of a cross section of a focusing quadrupole on the left, and a defocusing combined function dipole on the right (combined quadrupole + dipole components) [67]. The black lines on the figures are the equipotential vector lines. QF is made of 16 wedges and its quadrupole component is given by [68]

$$2B_{rm} \cos^2 \left( \frac{\pi}{M} \right) \frac{\sin \left( \frac{2\pi}{M} \right)}{\frac{2\pi}{M}} \left( \frac{1}{r_i} - \frac{1}{r_o} \right) \quad (6.3.29)$$

with  $B_{rm}$  is the remnant magnetization,  $r_i$  and  $r_o$  the inner and outer radii of the annulus made by the wedges, and  $M$  the number of magnetized wedges. The magnetization direction  $\alpha$  of each of the wedges is given by  $\alpha = 3\theta + \pi/2$ , where  $\theta$  is the azimuthal angle of each wedge.

**HELIX : Helical dipole**

(page 293 in PART B)

\*\*\*\*\* UNDER DEVELOPMENT \*\*\*\*\*

*HELIX* provides magnetic field from a helical dipole [66]. It can be used for instance to simulate light source helical wigglers, or siberian snakes in polarized beam accelerators.

The field components are computed following

$$\begin{cases} B_Y/B_0 = -(1 + \frac{k^2}{8}(3Y^2 + Z^2)) \sin(kX) + \frac{k^2}{4}YZ \cos(kX) \\ B_Z/B_0 = (1 + \frac{k^2}{8}(Y^2 + 3Z^2)) \cos(kX) - \frac{k^2}{4}YZ \sin(kX) \\ B_X/B_0 = -k(1 + \frac{k^2}{8}(Y^2 + Z^2))(Y \cos(kX) + Z \sin(kX)) \end{cases} \quad (6.3.30)$$

wherein

$$k = \epsilon \frac{2\pi}{\lambda} \quad (6.3.31)$$

the pitch

$\epsilon = \pm 1$  the helicity (+ for right-handed)

$X$  is the distance along the helix axis,  $B_0$  is the peak transverse field value.

Two techniques are available to compute the field and derivatives:

- the field is computed from Eq. 6.3.30, its derivatives are hard-coded, up to the second order in the coordinates
- the field at particle location and at the 26 neighboring nodes of a cubic mesh (Fig. 4) are computed from Eq. 6.3.30, numerical values of the derivatives are obtained by interpolation from that mesh (see Sec. 1.4.4).

**MAP2D : 2-D Cartesian uniform mesh field map - arbitrary magnetic field [64]**

*MAP2D* reads a 2-D field map that provides the three components  $B_X$ ,  $B_Y$ ,  $B_Z$  of the magnetic field at all nodes of a 2-D Cartesian uniform mesh in an  $(X, Y)$  plane. No particular symmetry is assumed, which allows the treatment of any type of field (e.g., solenoidal, or dipole, helical dipole, at arbitrary  $Z$  elevation - the map needs not be a mid-plane map).

The field map data file has to be filled with a format that satisfies the *FORTTRAN* reading sequence below (in principle compatible with *TOSCA* code outputs), details and possible updates are to be found in the source file `fmapw.f` :

```

OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD' [,FORM='UNFORMATTED'])
DO 1 J=1,JY
  DO 1 I=1,IX
    IF (BINARY) THEN
      READ(NL) Y(J), Z, X(I), BY(I,J), BZ(I,J), BX(I,J)
    ELSE
100    READ(NL,100) Y(J), Z, X(I), BY(I,J), BZ(I,J), BX(I,J)
      FORMAT (1X, 6E11.4)
    ENDF
1    CONTINUE

```

$IX$  ( $JY$ ) is the number of longitudinal (transverse horizontal) nodes of the 2-D uniform mesh,  $Z$  is the considered  $Z$ -elevation of the map. For binary files, `FNAME` must begin with 'B\_' or 'b\_', a flag 'BINARY' will thus be set to '.TRUE.'. The field  $\vec{B} = (B_X, B_Y, B_Z)$  is next normalized with `BNORM`, prior to ray-tracing. As well the coordinates  $X, Y$  are normalized with  $X$ -,  $Y$ -*NORM* coefficients (useful to convert to centimeters, the working units in **zgoubi**).

At each step of the trajectory of a particle, the field and its derivatives are calculated using a second or fourth degree polynomial interpolation followed by a  $Z$  extrapolation (see sections 1.3.4 page 30, 1.4.3 page 35). The interpolation grid is 3\*3-node for 2nd order (option `IORDRE = 2`) or 5\*5 for 4th order (option `IORDRE = 4`).

Entrance and/or exit integration boundaries may be defined, in the same way as for *CARTEMES*.

Including 'FLIP' in *TITL* causes the field map to be X-extended using its X-flipped copy.

**MAP2D-E : 2-D Cartesian uniform mesh field map - arbitrary electric field**

*MAP2D-E* reads a 2-D field map that provides the three components  $E_X$ ,  $E_Y$ ,  $E_Z$  of the electric field at all nodes of a 2-D Cartesian uniform mesh in an  $(X, Y)$  plane. No particular symmetry is assumed, which allows the treatment of any type of field (e.g., field of a parallel-plate mirror with arbitrary  $Z$  elevation - the map needs not be a mid-plane map).

The field map data file has to be filled with a format that satisfies the *FORTTRAN* reading sequence below (in principle compatible with *TOSCA* code outputs), details and possible updates are to be found in the source file `fmapw.f` :

```

OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD' [,FORM='UNFORMATTED'])
DO 1 J=1,JY
  DO 1 I=1,IX
    IF (BINARY) THEN
      READ(NL) Y(J), Z, X(I), EY(I,J), EZ(I,J), EX(I,J)
    ELSE
100    READ(NL,100) Y(J), Z, X(I), EY(I,J), EZ(I,J), EX(I,J)
      FORMAT (1X, 6E11.4)
    ENDF
1    CONTINUE

```

$IX$  ( $JY$ ) is the number of longitudinal (transverse horizontal) nodes of the 2-D uniform mesh,  $Z$  is the considered  $Z$ -elevation of the map. For binary files, `FNAME` must begin with 'E\_' or 'b\_', a flag 'BINARY' will thus be set to '.TRUE.'. The field  $\vec{E} = (E_X, E_Y, E_Z)$  is next normalized with `ENORM`, prior to ray-tracing. As well the coordinates  $X$ ,  $Y$  are normalized with  $X$ -,  $Y$ -*NORM* coefficients (useful to convert to centimeters, the working units in **zgoubi**).

At each step of the trajectory of a particle, the field and its derivatives are calculated using a second or fourth degree polynomial interpolation followed by a  $Z$  extrapolation (see sections 1.3.4 page 30, 1.4.3 page 35). The interpolation grid is 3\*3-node for 2nd order (option `IORDRE = 2`) or 5\*5 for 4th order (option `IORDRE = 4`).

Entrance and/or exit integration boundaries may be defined, in the same way as for *CARTEMES*.

Including 'FLIP' in *TITL* causes the field map to be X-extended using its X-flipped copy, as in:

```

'MAP2D-E'                                ! Electrostatic field map defined in a Cartesian frame.
0 2                                        ! IL=2 writes partice data to zgoubi.plt.
68.6204499993e-8 1. 1.                    ! Normalization so as to get ~-10.5MV/m on good radius.
FLIP HEADER_0 Condenser end               ! FLIP: causes the full field map to be obtained by
71 111                                    ! complementing it by an X-flipped copy.
al_151x21x71.map2D                        ! BNL pEDM electrostatic bend field map.
0 0. 0. 0.
2
0.1                                        ! Integration step size (cm).
1 0. 0. 0.

```

In that example, the full field map actually operated by **zgoubi** has twice the X-extent of 'al\_151x21x71.map2D', and is symmetric with respect to an  $X=0$  plane.



**MARKER : Marker**

*MARKER* does nothing. Just a marker. No data.

As any other keyword, *MARKER* is allowed two *LABELS*. Using '.plt' as a second *LABEL* will cause storage of current coordinates into zgoubi.plt.

**MULTIPOL : Magnetic multipole**

The simulation of multipolar magnetic field  $\vec{M}$  by *MULTIPOL* proceeds by addition of the dipolar ( $\vec{B}1$ ), quadrupolar ( $\vec{B}2$ ), sextupolar ( $\vec{B}3$ ), etc., up to 20-polar ( $\vec{B}10$ ) components, and of their derivatives up to fourth order, following

$$\begin{aligned}\vec{M} &= \vec{B}1 + \vec{B}2 + \vec{B}3 + \dots + \vec{B}10 \\ \frac{\partial \vec{M}}{\partial X} &= \frac{\partial \vec{B}1}{\partial X} + \frac{\partial \vec{B}2}{\partial X} + \frac{\partial \vec{B}3}{\partial X} + \dots + \frac{\partial \vec{B}10}{\partial X} \\ \frac{\partial^2 \vec{M}}{\partial X \partial Z} &= \frac{\partial^2 \vec{B}1}{\partial X \partial Z} + \frac{\partial^2 \vec{B}2}{\partial X \partial Z} + \frac{\partial^2 \vec{B}3}{\partial X \partial Z} + \dots + \frac{\partial^2 \vec{B}10}{\partial X \partial Z} \\ &\text{etc.}\end{aligned}$$

The independent components  $\vec{B}1, \vec{B}2, \vec{B}3, \dots, \vec{B}10$  and their derivatives up to the fourth order are calculated as described in section 1.3.7.

The entrance and exit fringe fields are treated separately. They are characterized by the integration zone  $X_E$  at entrance and  $X_S$  at exit, as for *QUADRUPO*, and by the extent  $\lambda_E$  at entrance,  $\lambda_S$  at exit. The fringe field extents for the dipole component are  $\lambda_E$  and  $\lambda_S$ . The fringe field for the quadrupolar (sextupolar, ..., 20-polar) component is given by a coefficient  $E_2$  ( $E_3, \dots, E_{10}$ ) at entrance, and  $S_2$  ( $S_3, \dots, S_{10}$ ) at exit, such that the extent is  $\lambda_E \times E_2$  ( $\lambda_E \times E_3, \dots, \lambda_E \times E_{10}$ ) at entrance and  $\lambda_S \times S_2$  ( $\lambda_S \times S_3, \dots, \lambda_S \times S_{10}$ ) at exit.

If  $\lambda_E = 0$  ( $\lambda_S = 0$ ) the multipole lens is considered to have a sharp edge field at entrance (exit), and then,  $X_E$  ( $X_S$ ) is forced to zero (for the mere purpose of saving computing time). If  $E_i = 0$  ( $S_i = 0$ ) ( $i = 2, 10$ ), the entrance (exit) fringe field for the multipole component  $i$  is considered as a sharp edge field. In sharp edge field model, the wedge angle vertical first order focusing effect (if  $\vec{B}1$  is non zero) is simulated at magnet entrance and exit by a kick  $P_2 = P_1 - Z_1 \tan(\epsilon/\rho)$  applied to each particle ( $P_1, P_2$  are the vertical angles upstream and downstream of the EFB,  $Z_1$  is the vertical particle position at the EFB,  $\rho$  the local horizontal bending radius and  $\epsilon$  the wedge angle experienced by the particle ;  $\epsilon$  depends on the horizontal angle T).

Any multipole component  $\vec{B}i$  can be rotated independently by an angle  $RXi$  around the longitudinal  $X$ -axis, for the simulation of positioning defects, as well as skew lenses.

Magnet (mis-)alignment is assured by *KPOS*. *KPOS* also allows some degrees of automatic alignment useful for periodic structures (section 7.10).

**OCTUPOLE : Octupole magnet (Fig. 33)**

(page 312 in PART B)

The meaning of parameters for *OCTUPOLE* is the same as for *QUADRUPO*. In fringe field regions the magnetic field  $\vec{B}(X, Y, Z)$  and its derivatives up to fourth order are derived from the scalar potential approximated to the 8-th order in  $Y$  and  $Z$

$$V(X, Y, Z) = \left( G(X) - \frac{G''(X)}{20} (Y^2 + Z^2) + \frac{G''''(X)}{960} (Y^2 + Z^2)^2 \right) (Y^3 Z - Y Z^3)$$

with  $G_0 = \frac{B_0}{R_0^3}$

The modelling of the fringe field form factor  $G(X)$  is described under *QUADRUPO*, p. 136.

Outside fringe field regions, or everywhere in sharp edge dodecapole ( $\lambda_E = \lambda_S = 0$ ),  $\vec{B}(X, Y, Z)$  in the magnet is given by

$$\begin{aligned} B_X &= 0 \\ B_Y &= G_0(3Y^2 - Z^2) Z \\ B_Z &= G_0(Y^2 - 3Z^2) Y \end{aligned}$$

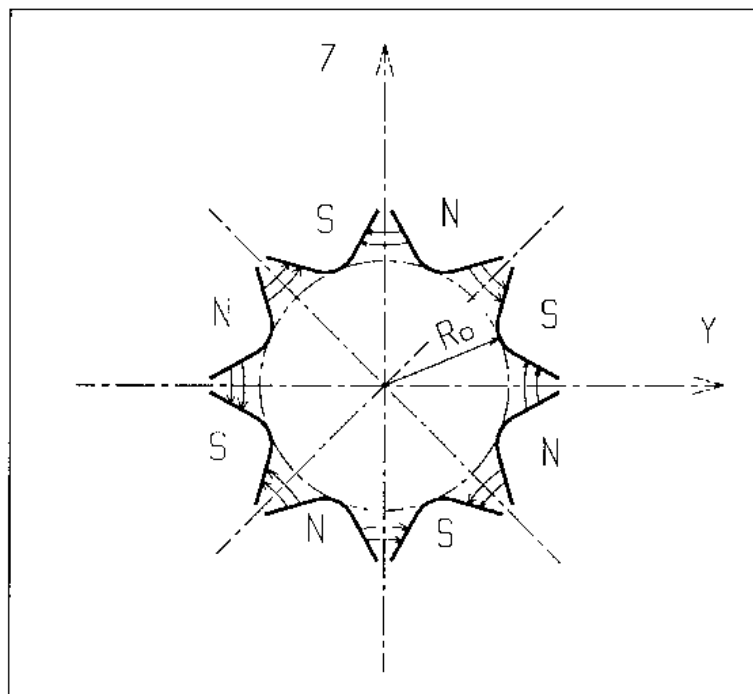


Figure 33: Octupole magnet

**POISSON :Read magnetic field data from *POISSON* output**

This keyword allows reading a field profile  $B(X)$  from *POISSON* output. Let *FNAME* be the name of this output file (normally, *FNAME* = outpoi.lis) ; the data are read following the *FORTRAN* statements here under

```

I = 0
11  CONTINUE
I = I + 1
    READ(LUN,101,ERR=10,END=10) K, K, K, R, X(I), R, R, B(I)
101  FORMAT(I1, I3, I4, E15.6, 2F11.5, 2F12.3)
    GOTO 11
10  CONTINUE
    ...

```

where  $X(I)$  is the longitudinal coordinate, and  $B(I)$  is the  $Z$  component of the field at a node ( $I$ ) of the mesh.  $K$ s and  $R$ s are dummy variables appearing in the *POISSON* output file outpoi.lis but not used here.

From this field profile, a 2-D median plane map is built, with a rectangular and uniform mesh ; mid-plane symmetry is assumed. The field at each node  $(X_i, Y_j)$  of the map is  $B(X_i)$ , independent of  $Y_j$  (*i.e.*, the distribution is uniform in the  $Y$  direction).

For the rest, *POISSON* works in a way similar to *CARTEMES*.

**POLARMES : 2-D polar mesh magnetic field map**

medskip

Similar to *CARTEMES*, apart from the polar mesh frame :  $IX$  is the number of angular nodes,  $JY$  the number of radial nodes ;  $X(I)$  and  $Y(J)$  are respectively the angle and radius of a node (these parameters are similar to those entering in the definition of the field map in *DIPOLE-M*).

Including 'FLIP' in *TITL* causes the field map to be X-extended using its X-flipped copy.

**PS170 : Simulation of a round shape dipole magnet**

*PS170* is dedicated to a 'rough' simulation of CERN *PS170* spectrometer dipole.

The field  $B_0$  is constant inside the magnet, and zero outside. The pole is a circle of radius  $R_0$ , centered on the  $X$  axis. The output coordinates are generated at the distance  $XL$  from the entrance (Fig. 34).

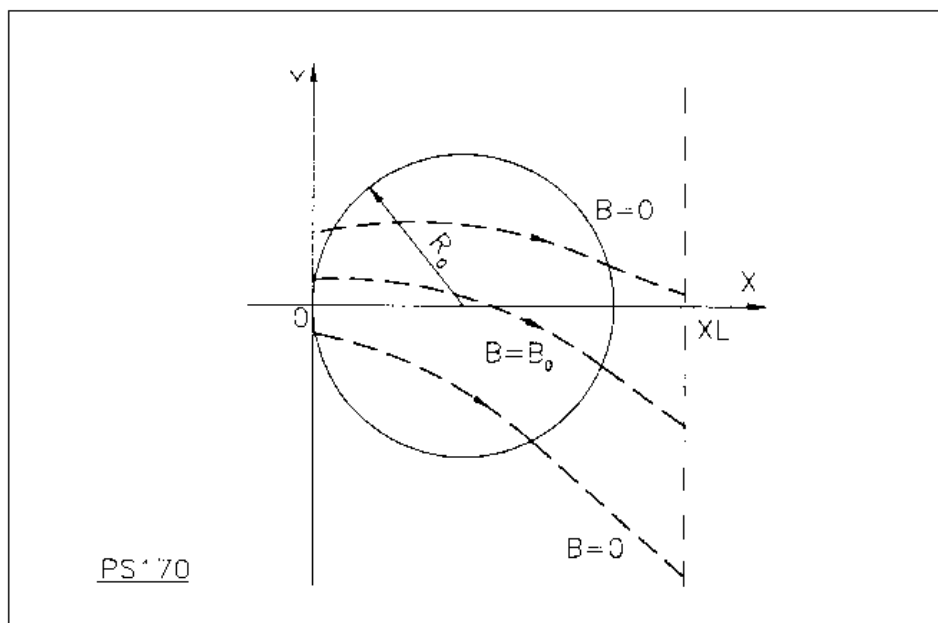


Figure 34: Scheme of the PS170 magnet simulation.

Including 'FLIP' in *TITL* causes the field map to be X-extended using its X-flipped copy.

**QUADISEX, SEXQUAD : Sharp edge magnetic multipoles**

*SEXQUAD* defines in a simple way a sharp edge field with quadrupolar, sextupolar and octupolar components. *QUADISEX* adds a dipole component. The length of the element is  $XL$ . The vertical component  $B \equiv B_z(X, Y, Z = 0)$  of the field and its derivatives in median plane are calculated at each step using the following expressions

$$\begin{aligned}
 B &= B_0 \left( U + \frac{N}{R_0} Y + \frac{B}{R_0^2} Y^2 + \frac{G}{R_0^3} Y^3 \right) \\
 \frac{\partial B}{\partial Y} &= B_0 \left( \frac{N}{R_0} + 2 \frac{B}{R_0^2} Y + 3 \frac{G}{R_0^3} Y^2 \right) \\
 \frac{\partial^2 B}{\partial Y^2} &= B_0 \left( 2 \frac{B}{R_0^2} + 6 \frac{G}{R_0^3} Y \right) \\
 \frac{\partial^3 B}{\partial Y^3} &= 6 B_0 \frac{G}{R_0^3}
 \end{aligned}$$

and then extrapolated out of the median plane by Taylor expansion in  $Z$  (see section 1.3.3).

With option *SEXQUAD*,  $U = 0$ , while with *QUADISEX*,  $U = 1$ .

**QUADRUPO : Quadrupole magnet (Fig. 35)**

(page 324 in PART B)

The length of the magnet  $XL$  is the distance between the effective field boundaries (EFB), Fig. 36. The field at the pole tip  $R_0$  is  $B_0$ .

The extent of the entrance (exit) fringe field is characterized by  $\lambda_E(\lambda_S)$ . The distance of ray-tracing on both sides of the EFBs, in the field fall off regions, will be  $\pm X_E$  at the entrance, and  $\pm X_S$  at the exit (Fig. 36), by prior and further automatic change of frame.

In the fringe field regions  $[-X_E, X_E]$  and  $[-X_S, X_S]$  on both sides of the EFBs,  $\vec{B}(X, Y, Z)$  and its derivatives up to fourth order are calculated at each step of the trajectory from the analytical expressions of the three components  $B_X, B_Y, B_Z$  obtained by differentiation of the scalar potential (see section 1.3.7) expressed to the 8th order in  $Y$  and  $Z$ .

$$V(X, Y, Z) = \left( G(X) - \frac{G''(X)}{12} (Y^2 + Z^2) + \frac{G''''(X)}{384} (Y^2 + Z^2)^2 - \frac{G''''''(X)}{23040} (Y^2 + Z^2)^3 \right) YZ$$

(  $G^{(n)}(X) = d^n G(X)/dX^n$ , etc.)

where  $G(X)$  is the gradient on axis [48] :

$$G(X) = \frac{G_0}{1 + \exp P(d(X))} \quad \text{with} \quad G_0 = \frac{B_0}{R_0}$$

and,

$$P(d) = C_0 + C_1 \left( \frac{d}{\lambda} \right) + C_2 \left( \frac{d}{\lambda} \right)^2 + C_3 \left( \frac{d}{\lambda} \right)^3 + C_4 \left( \frac{d}{\lambda} \right)^4 + C_5 \left( \frac{d}{\lambda} \right)^5$$

where  $d(X)$  is the distance to the field boundary and  $\lambda$  stands for  $\lambda_E$  or  $\lambda_S$  (normally,  $\lambda \simeq 2 \times R_0$ ). When fringe fields overlap inside the magnet ( $XL \leq X_E + X_S$ ), the gradient  $G$  is expressed as

$$G = G_E + G_S - 1$$

where,  $G_E$  is the entrance gradient and  $G_S$  is the exit gradient.

If  $\lambda_E = 0$  ( $\lambda_S = 0$ ), the field at entrance (exit) is considered as sharp edged, and then  $X_E(X_S)$  is forced to zero (for the mere purpose of saving computing time).

Outside of the fringe field regions (or everywhere when  $\lambda_E = \lambda_S = 0$ )  $\vec{B}(X, Y, Z)$  in the magnet is given by

$$\begin{aligned} B_X &= 0 \\ B_Y &= G_0 Z \\ B_Z &= G_0 Y \end{aligned}$$



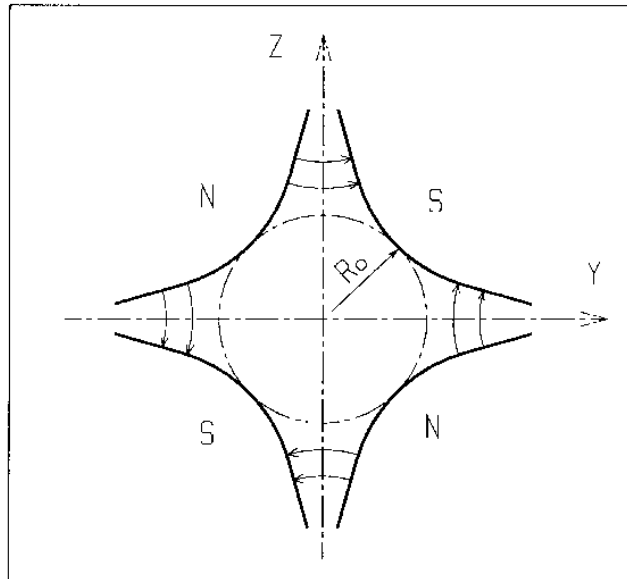


Figure 35: Quadrupole magnet

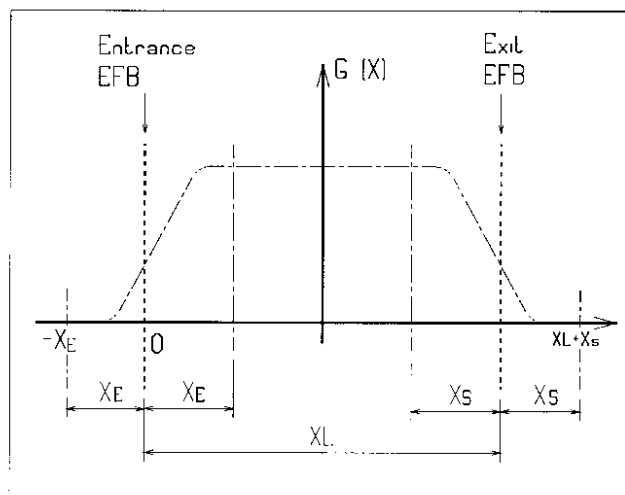


Figure 36: Scheme of the longitudinal field gradient  $G(X)$ .  $(OX)$  is the longitudinal axis of the reference frame  $(0, X, Y, Z)$  of **zgoubi**. The length of the element is  $XL$ . Trajectories are ray-traced from  $-X_E$  to  $XL + X_S$ , by means of respectively prior and final automatic change of frame.

**SEPARA : Wien Filter - analytical simulation**

Note : *SEPARA* is an analytical transport: simulation by stepwise integration may use *WIENFILTER*.

*SEPARA* provides an analytic simulation of an electrostatic separator. Input data are the length  $L$  of the element, the electric field  $E$  and the magnetic field  $B$ . The mass  $m$  and charge  $q$  of the particles are entered by means of the keyword *PARTICUL*.

The subroutines involved in *SEPARA* solve the following system of three equations with three unknown variables  $S$ ,  $Y$ ,  $Z$  (while  $X \equiv L$ ), that describe the cycloidal motion of a particle in  $\vec{E}$ ,  $\vec{B}$  static fields (Fig. 37).

$$\begin{aligned} X &= -R \cos\left(\frac{\omega S}{\beta c} + \epsilon\right) - \frac{\alpha S}{\omega \beta c} + \frac{C_1}{\omega} \\ Y &= R \sin\left(\frac{\omega S}{\beta c} + \epsilon\right) - \frac{\alpha}{\omega^2} - \frac{C_2}{\omega} + Y_0 \\ Z &= S \sin(P_0) + Z_0 \end{aligned}$$

where,  $S$  is the path length in the separator,  $\alpha = -\frac{Ec^2}{\gamma}$ ,  $\omega = -\frac{Bc^2}{m\gamma}$ ,  $C_1 = \beta \sin(T_0) \cos(P_0)$  and  $C_2 = \beta c \cos(T_0) \cos(P_0)$  are initial conditions.  $c$  = velocity of light,  $\beta c$  = velocity of the particle,  $\gamma = (1 - \beta^2)^{-\frac{1}{2}}$  and  $\tan \epsilon = (C_2 + \frac{\alpha}{\omega})/C_1$ .  $Y_0$ ,  $T_0$ ,  $Z_0$ ,  $P_0$  are the initial coordinates of the particle in the **zgoubi** reference frame. Here  $\beta c$  and  $\gamma$  are assumed constant, which is true as long as the change of momentum due to the electric field remains negligible all along the separator.

The option index  $IA$  in the input data allows switching to inactive element (thus equivalent to *ESL*), horizontal or vertical separator. Normally,  $E$ ,  $B$  and the value of  $\beta_W$  for wanted particles are related by

$$B_{[T]} = -\frac{E_{[V/m]}}{\beta_W \cdot c_{[m/s]}}$$

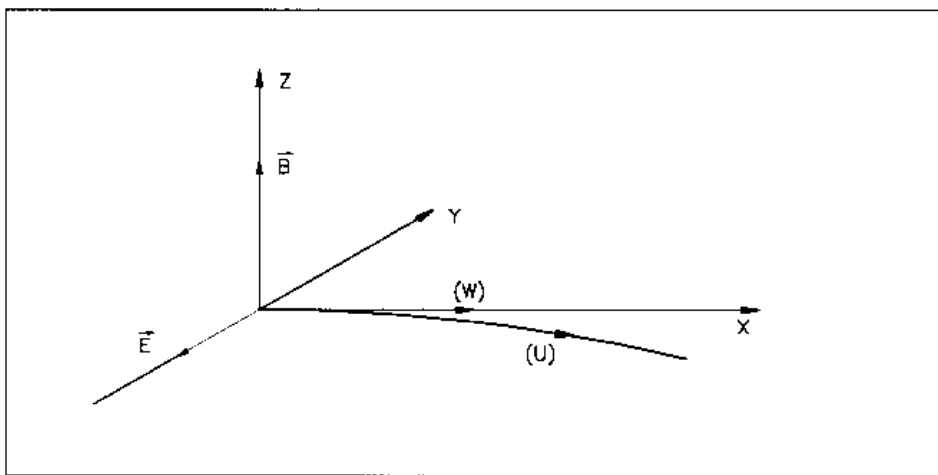


Figure 37: Horizontal separation between a wanted particle, ( $W$ ), and an unwanted particle, ( $U$ ). ( $W$ ) undergoes a linear motion while ( $U$ ) undergoes a cycloidal motion.

**SEXTUPOL : Sextupole magnet (Fig. 38)**

The meaning of parameters for *SEXTUPOL* is the same as for *QUADRUPO*.

In fringe field regions the magnetic field  $\vec{B}(X, Y, Z)$  and its derivatives up to fourth order are derived from the scalar potential approximated to 7th order in  $Y$  and  $Z$

$$V(X, Y, Z) = \left( G(X) - \frac{G''(X)}{16} (Y^2 + Z^2) + \frac{G''''(X)}{640} (Y^2 + Z^2)^2 \right) \left( Y^2 Z - \frac{Z^3}{3} \right)$$

with  $G_0 = \frac{B_0}{R_0^2}$

The modelling of the fringe field form factor  $G(X)$  is described under *QUADRUPO*, p. 136.

Outside fringe field regions, or everywhere in sharp edge sextupole ( $\lambda_E = \lambda_S = 0$ ),  $\vec{B}(X, Y, Z)$  in the magnet is given by

$$\begin{aligned} B_X &= 0 \\ B_Y &= 2G_0 Y Z \\ B_Z &= G_0 (Y^2 - Z^2) \end{aligned}$$

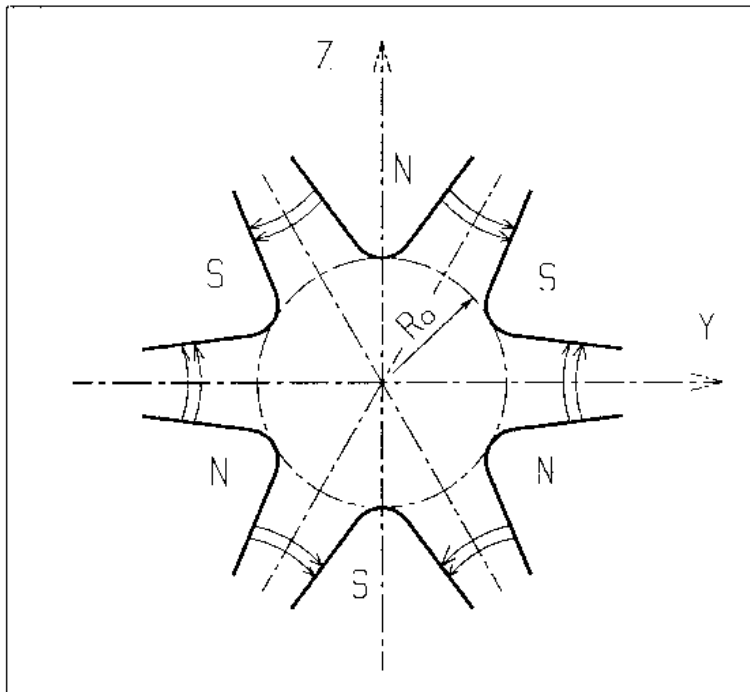


Figure 38: Sextupole magnet

**SOLENOID : Solenoid (Fig. 39)**

(page 333 in PART B)

The solenoidal magnet has an effective length  $XL$  and a mean radius  $R_0$  (Fig. 39). Its asymptotic field is  $B_0 = \mu_0 NI / XL$ , namely,

$$\int_{-\infty}^{\infty} B_X(X, r) dX \xrightarrow{R_0/XL \rightarrow 0} \mu_0 NI, \quad \forall r < R_0$$

In this expression,  $NI$  = number of ampere-turn,  $\mu_0 = 4\pi 10^{-7}$ .

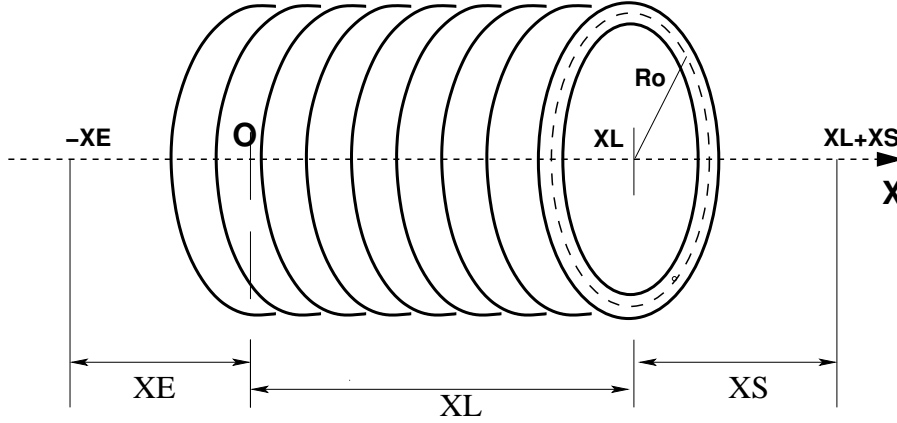


Figure 39: Solenoidal magnet. Note that in **zgoubi** the origin is taken at the entrance EFB, whereas Eq. 6.3.32 is for origin taken at the center.

*Field fall-off extents:*

The distance of ray-tracing beyond the effective length  $XL$ , is  $X_E$  upstream of the entrance effective field boundary (located at  $X = 0$  in **zgoubi** reference frame), and  $X_S$  downstream of the exit field boundary (located at  $X = X_L$  in **zgoubi** frame). This adding entrance and exit integration extents is transparent to the user, namely, prior to being pushed, a particle is first moved back to  $X = -X_E$  by a **CHANGREF** style change of frame; it is then pushed all the way to  $X = X_L + X_S$ , whereafter a second change of frame brings it back to the exit effective field boundary at  $X = X_L$ .

Two methods are available for the computation of the field  $\vec{B}(X, r)$  and its derivatives.

**Method 1**

This is the classical on-axis field model [12]

$$B_X(X, r = 0) = \frac{B_0}{2} \left[ \frac{XL/2 - X}{\sqrt{(XL/2 - X)^2 + R_0^2}} + \frac{XL/2 + X}{\sqrt{(XL/2 + X)^2 + R_0^2}} \right] \quad (6.3.32)$$

with origin  $X = r = 0$  taken at the center of the solenoid. This model assumes that the coil thickness is small compared to its mean radius  $R_0$ . The effective magnetic length is

$$L_{mag} \equiv \frac{\int_{-\infty}^{\infty} B_X(X, r < R_0) dX}{B_X(X = r = 0)} = XL \sqrt{1 + \frac{4R_0^2}{XL^2}} > XL \quad (6.3.33)$$

with in addition

$$B_X(\text{center}) \equiv B_X(X = r = 0) = \frac{\mu_0 NI}{XL \sqrt{1 + \frac{4R_0^2}{XL^2}}} \xrightarrow{R_0 \ll XL} \frac{\mu_0 NI}{XL}$$

The field  $\vec{B}(s)$  and its derivatives  $d^n \vec{B}(s) / ds^n$  at all  $(X, Y, Z)$ , as required for pushing a particle (Eqs. 1.2.4, 1.2.8) and possibly its spin (Eqs. 2.1.5, 2.2.2), are extrapolated from eq. 6.3.32, following the method described in section 1.3.1. Note that the model validity limit is  $\sqrt{Y^2 + Z^2} = r < R_0$ .

**Method 2**

The computation of the radial,  $B_r(X, r)$ , and axial  $B_X(X, r)$  components of the field vector  $\vec{B}(X, r)$  (with  $r = (Y^2 + Z^2)^{1/2}$ ) follows the technique described in Ref. [69]. Summarizing:  $B_r(X, r)$ , and  $B_X(X, r)$  are constructed from respectively

$$\begin{aligned} b_X(X, r) &= \frac{\mu_0 N I c k}{4\pi r} \mathcal{X} \left[ K + \frac{r_0 - r}{2r_0} (\Pi - K) \right] \\ b_r(X, r) &= \mu_0 N I \frac{1}{k} \sqrt{\frac{r_0}{r}} [2(K - E) - k^2 K] \end{aligned} \quad (6.3.34)$$

wherein  $K$ ,  $E$  and  $\Pi$  are the three complete elliptic integrals,  $\mathcal{X}$  is an X- and XL-dependent quantity, and

$$k = 2\sqrt{r_0 r} / \sqrt{(r_0 + r)^2 + \mathcal{X}^2}; \quad c = 2\sqrt{r_0 r} / (r_0 + r)$$

$K$ ,  $E$  and  $\Pi$  are computed using fast converging algorithms proposed in that very Ref. [69]. Their derivatives are obtained from recursive relations [70], as needed in order to derive the field derivatives at all  $(X, Y, Z)$  (up to second order in the present version of the code).

**For software developers**

*Solenoid* parameters are read from `zgoubi.dat` by `rsolen.f`. Initializations prior to tracking are done in `solenof.f`. Field computation during tracking is by `solenf.f`. The initial  $\delta X = -X_E$  (prior to pushing a particle) and final  $\delta X = -X_S$  (once that particle has reached  $X = X_L + X_S$ ) change of frames to account for field fall-offs, transparent to the user, are performed in `transf.f`.

**SPINR : Spin rotation [71]**

(page 335 in PART B)

*SPINR* causes spin precession, a local transformation on the spin vector of the particles, as, *e.g.*, in a local helical spin rotator. Two options are available, as follows.

**IOPT = 0** inhibits the keyword.

**If IOPT = 1 :** the precession is defined by its axis, at angle  $\phi$  with respect to **zgoubi**'s X-axis and assumed in the horizontal plane, and by its value,  $\mu$ .

**If IOPT = 2 :** given the spin precession axis direction  $\phi$  in the (X,Y) plane, the spin precession angle follows a function of the *reference* Lorentz factor :

$$\mu(\gamma) = \left( \frac{B}{B_0} \right)^2 \times \left( C_0 + \frac{C_1}{\gamma} + \frac{C_2}{\gamma^2} + \frac{C_3}{\gamma^3} \right)$$

with  $B$  and  $B_0$  normally a nominal and a reference magnetic field value, and  $C_i$  empirical coefficients. The *reference* Lorentz factor corresponds to the reference rigidity *BORO*, as defined for instance with *[MC]OBJET*, possibly updated in the presence of acceleration (see section 7.2, page 213).

*SCALING* can be used to scale the precession angle  $\mu$ .

**TOSCA : 2-D and 3-D Cartesian or cylindrical mesh field map**

(page 346 in PART B)

TOSCA is dedicated to the reading and treatment of

- 2-D Cartesian or cylindrical mesh mid-plane field maps (mid-plane is assumed anti-symmetry plane, as in an upright multipole),
- 3-D Cartesian or cylindrical mesh field maps.

TOSCA was first installed to read TOSCA magnet computer code style of fieldmap output and formatting, hence the name. However the actual origin of the fieldmap, TOSCA or other computer code, does not matter.

TOSCA has been developed over the years to allow reading many different input formats, including 3-D maps given under the form of a series of files, partial 3-D maps with particular symmetry plane(s), etc. In order to select between the many possibilities, two input data, IZ and MOD.MOD2, are used. This is addressed in the following, and table 1, page 145, lists all possibilities as a function of the values given to IZ and MOD.MOD2.

A double-flag,  $MOD[.MOD2]$ , determines whether a Cartesian mesh (Fig. 40-A or -B) or a Z-axis cylindrical mesh (Fig. 40-C) is used, and the nature and/or form of the field map data set.

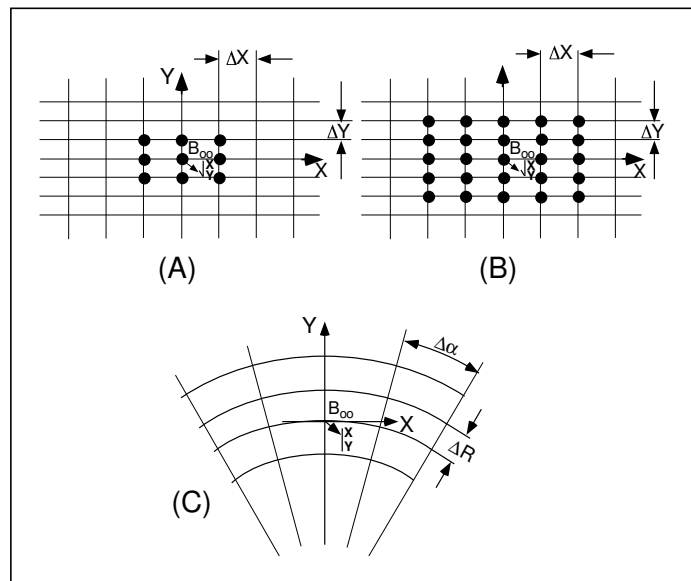


Figure 40: Principle 2-D field map mesh as used by TOSCA, and the (O;X,Y) particle coordinate frame. (A), (B): Cartesian mesh in the (X,Y) plane, the mesh increments are  $\Delta X$  and  $\Delta Y$ ; (A) illustrates a 9-point interpolation grid and (B) a 25-point interpolation grid. (C) : Polar mesh and moving Cartesian frame (O;X,Y) along a reference arc with radius  $R_M$ , as used during stepwise integration; the mesh increments are  $\Delta\alpha$  and  $\Delta R$ ; In all three cases, during integration the interpolation grid is centered on the mesh node closest to particle location

The total number of field data files to be read is determined by the *MOD[.MOD2]* flag (see below) and by the parameter *IZ* that appears in the data list following the keyword. Each of these files contains the field components  $B_X$ ,  $B_Y$ ,  $B_Z$  on an  $(X, Y)$  mesh.  $IZ = 1$  for a 2-D map, and in this case  $B_X$  and  $B_Y$  are assumed zero all over the map<sup>13</sup>. For a 3-D map with mid-plane symmetry, described with a set of 2-D maps at various  $Z$ , then  $MOD=0$  and  $IZ \geq 2$ , and thus, the first data file whose name follows in the data list is supposed to contain the median plane field (assuming  $Z = 0$  and  $B_X = B_Y = 0$ ), while the remaining  $IZ - 1$  file(s) contain the  $IZ - 1$  additional planes in increasing  $Z$  order. For arbitrary 3-D maps, no symmetry assumed, then  $MOD=1$  and the total number of maps (whose names follow in the data list) is  $IZ$ , such that map number  $[IZ/2] + 1$  is the  $Z = 0$  elevation one.

The field map data file has to be filled with a format that fits the *FORTRAN* reading sequence. The following is an instance, details and possible updates are to be found in the source file `fmapw.f` :

```
DO 1 K = 1, KZ
  OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD' [,FORM='UNFORMATTED'])
  DO 1 J = 1, JY
    DO 1 I = 1, IX
      IF (BINARY) THEN
        READ(NL) Y(J), Z(K), X(I), BY(J,K,I), BZ(J,K,I), BX(J,K,I)
      ELSE
100    READ(NL,100) Y(J), Z(K), X(I), BY(J,K,I), BZ(J,K,I), BX(J,K,I)
        FORMAT(1X,6E11.2)
      ENDIF
    1 CONTINUE
```

$IX$  ( $JY$ ,  $KZ$ ) is the number of longitudinal (transverse horizontal, vertical) nodes of the 3-D uniform mesh. For letting **zgoubi** know in case these are binary files, `FNAME` must begin with 'B\_' or 'b\_'.

Other *MOD[.MOD2]* cases in addition to the above, are listed in table 1. The *FORTRAN* subroutine `fmapw.f` and its entries `FMAPR`, `FMAPR2` can be looked up for more details, in particular regarding the formatting of the field map data file(s).

Once the field map(s) reading is completed, the field  $\vec{B} = (B_X, B_Y, B_Z)$  is normalized by means of *BNORM* in a similar way as in *CARTEMES*. As well the coordinates  $X$  and  $Y$  (and  $Z$  in the case of 3-D field maps) are normalized by the  $X$ -[,  $Y$ -,  $Z$ -]*NORM* coefficient (useful to convert to centimeters, the working units in **zgoubi**).

At each step of the trajectory of a particle inside the map, the field and its derivatives are calculated as follows :

- in the case of a 2-D map, by means of a second or fourth order polynomial interpolation, depending on *IODRE* ( $IODRE = 2, 25$  or  $4$ ), as for *CARTEMES*,
- in the case of a 3-D map, by means of a second order polynomial interpolation with a  $3 \times 3 \times 3$ -point parallelepipedic grid, as described in section 1.4.4.

In the Cartesian mesh case, entrance and/or exit integration boundaries between which the trajectories are integrated in the field may be defined, in the same way as in *CARTEMES*.

A '*TITL*' (a line of comments) is part of the arguments of the keyword *TOSCA*. It allows introducing necessary field map handling data, or some options, for instance :

- ◇ including 'HEADER N' allows specifying the number of header lines ( $N$ ) at the top of the field map file. In the case of cylindrical field maps one line is necessary ( $N \geq 1$ ) to indicate the reference radius in particular (but also the mesh size, etc., lookup 'HEADER' in `fmapw.f` in case of doubt),
- ◇ including 'FLIP' in *TITL* causes the field map to be X-flipped,
- ◇ including 'ZroBXY' forces  $B_X = B_Y = 0$  at all  $Z=0$  nodes of the field map mesh (only applies with  $MOD=15$  and  $MOD=24$ ),

<sup>13</sup>Use *MAP2D* in case non-zero  $B_X$ ,  $B_Y$  are to be taken into account in a 2-D map.



Table 1: The various *IZ*, *MOD* and *MOD2* options, when using *TOSCA*.

**IZ** : number of nodes of the *complete* field map along the *Z* direction (*IZ*=1 for 2-D)

**MOD, MOD2** : determine the coordinate system, symmetries, reading format and column sequence, etc.

**NF** : number of field map input data files to be declared. Always include mid-plane map.

**Expected columns** : formatting of the coordinates and field data columns in the field map data file(s)

**'Example' example folder** : examples of *zgoubi* runs using field maps can be found in the subfolders of

*zgoubi-code/examples/KEYWORDS/TOSCA/cartesian* (case  $MOD \leq 19$ ) or

*zgoubi-code/examples/KEYWORDS/TOSCA/cylindrical* (case  $MOD \geq 20$ ).

The rightmost column below indicates the subfolder of concern, following (*IZ*, *MOD*, *MOD2*) options of interest.

**MOD  $\leq$  19 : Cartesian mesh**

<b>IZ</b>	<b>MOD</b>	<b>.MOD2</b>		<b>NF</b>	<b>Expected columns</b>	<b>Example folder</b>
1	0, 1	none or .1, .2, .3	<b>2-D</b> map. File contains $B_Z(X, Y) _{Z=0}$ , mid-plane antisymmetry assumed. Several different reading formats (see <i>fmapw.f/fmapr3</i> ).	1	Y, Z, X, BY, BZ, BX	IZ-MOD-.MOD2_1-0-none/ (GSI KAOS spectrometer)
1	3	none or .1	<b>2-D</b> map. Used for AGS main magnet. If <i>MOD2</i> =1, $B_Z(X, Y, Z = 0)$ field is perturbed by $(1 + n_1 Y + n_2 Y^2 + n_3 Y^3)$ factor.	1	Special - see example	AGS/usingMainMagnetsMaps (AGS with main magnet maps)
1	15	.1 - .4	<b>2-D</b> map. Up to 4 files to be combined linearly into a new map : field at all node of new map is $\vec{B} = \sum_{i=1}^{MOD2} a_i \vec{B}_i$ . Mid-plane antisymmetry is assumed : each file has to contain $B_Z(X, Y, Z = 0)$ .	1 to 4	Y, Z, X, BY, BZ, BX	IZ-MOD-.MOD2_1-15-.1-.4 (EMMA FFAG cell)
>1	0	none	<b>3-D</b> map. Files span upper half of magnet, one per $(X, Y) _{0 \leq Z \leq Z_{max}}$ plane including median plane, mid-plane antisymmetry assumed.	1+ <i>IZ</i> /2	Y, Z, X, BY, BZ, BX	IZ-MOD-.MOD2_gt1-0-none/ (GSI KAOS spectrometer)
odd, $\geq 3$	1	none	<b>3-D</b> map. Files span full magnet volume, one file per $(X, Y)$ plane, no symmetry assumed.	<i>IZ</i>	Y, Z, X, BY, BZ, BX	IZ-MOD-.MOD2_gt1-1-none/ (AGS warm helix snake)
>1	12	none	<b>3-D</b> map. Single file, upper half of magnet, mid-plane antisymmetry assumed.	1		
>1	12	.1	<b>3-D</b> map. Single file, whole magnet volume, no symmetry assumed.	1	Y, Z, X, BY, BZ, BX	IZ-MOD-.MOD2_gt1-12-.1/ (AGS warm helix snake)
>1	12	.2	<b>3-D</b> map. Single file, 1/8th of the magnet, symmetry wrt. $(X, Y)_{Z=0}$ , $(X, Z)_{Y=0}$ , $(Y, Z)_{X=0}$ planes.	1		
odd, $\geq 3$	15	.1 - .4	<b>3-D</b> map. Up to 4 files to be combined linearly into a new map, field at all node of new map is $\vec{B} = \sum_{i=1}^{MOD2} a_i \vec{B}_i$ . Each file has to contain $B_{X, Y, Z}(X, Y, Z)$ data over <i>IZ</i> equally <i>Z</i> -spaced $(X, Y)$ planes (no symmetry assumed).	1 to 4		
odd, $\geq 3$	16	.1 - .4	<b>3-D</b> map. Fields from up to 4 maps to be combined linearly into a new field value at particle location, $\vec{B} = \sum_{i=1}^{MOD2} a_i \vec{B}_i$ . Each file has to contain $B_{X, Y, Z}(X, Y, Z)$ data over <i>IZ</i> equally <i>Z</i> -spaced $(X, Y)$ planes.	1 to 4	UNDER DEVELOPMENT	

**MOD  $\geq$  20 : Cylindrical mesh**

<b>IZ</b>	<b>MOD</b>	<b>.MOD2</b>				
1	25	.1 - .4	<b>2-D</b> map. Up to 4 files to be combined linearly into a new map : at all node of new map $\vec{B} = \sum_{i=1}^{MOD2} a_i \vec{B}_i$ . Each file contains mid-plane $B_Z(X, Y, Z = 0)$ data, mid-plane antisymmetry is assumed.	1 to 4	Y, Z, X, BY, BZ, BX	IZ-MOD-.MOD2_1-15-.1-.4 (EMMA FFAG cell)
>1	20, 21		<b>3-D</b> map. Single file. <i>MOD</i> =20 : 1/4 magnet, cylindrical symmetry with respect to $(Y, Z)$ plane and antisymmetry wrt $(X, Y)$ plane. <i>MOD</i> =21 : another type of symmetry (to be documented - see <i>fmapw.f</i> ).	1	$Y(r, \theta), Z, X(r, \theta),$ BY, BZ, BX	IZ-MOD-.MOD2_gt1-20 (KEK 150 MeV FFAG)
odd, $\geq 1$	22, 23	.1 - .4	<b>2-D</b> or <b>3-D</b> map. Mid-plane antisymmetry assumed. Up to 4 files can be combined linearly into a new one; each file is $Z \geq 0$ half-magnet. Field at all node of new map is $\vec{B} = \sum_{i=1}^{MOD2} a_i \vec{B}_i$ . Each file has to contain $B_{X, Y, Z}(X, Y, Z)$ data over $(IZ+1)/2$ equally <i>Z</i> -spaced $(X, Y)$ planes. <i>MOD</i> =23 : special, test code (see <i>fmapw.f</i> ).	1 to 4	$Y(r, \theta) = r \cos \theta, Z,$ $X(r, \theta) = r \sin \theta,$ BY, BZ, BX	IZ-MOD-.MOD2_gt1-22-.1-.4 (ERIT FFAG)
odd, $\geq 3$	24		<b>3-D</b> map, full volume. No symmetry assumed. The file contains $B_{X, Y, Z}(X, Y, Z)$ data over <i>IZ</i> equally <i>Z</i> -spaced $(X, Y)$ planes.	1	$\theta, R, Z, B_\theta, BR, BZ$	

◇ ‘RHIC\_helix’ can be mentioned as part of the ‘TITL’ string. That concerns RHIC snakes and rotators field maps [72]. In that case *BNORM* (assuming *SCALING* is not used, see below) will no longer just be a scaling factor to the field read from the map, instead it will be the coil current (in Amps); the effective field at particle location will be derived from the OPERA map accounting for a normalization to the *measured* magnetic field (measured B(I) data used for that are hard-coded in the program `toscac.f`) [72]. The coefficient  $a(1)$  in mode *MOD.MOD2=15.1* can be used to normalize the field map (namely,  $a(1) = 1/B_{max}$  with  $B_{max}$  the maximum field value on helix axis). If *SCALING* is used, then actually *BNORM*×*SCAL* (not just *BNORM*) *has to represent* the coil current value, in Amps; note also: in that case,  $sign(BNORM \times SCAL) == sign(BNORM)$ , the sign of *SCAL* is not taken into account.

An example of the two modes of utilization (with or without “RHIC\_helix” flag) is given in PART. B, p. 347.

**TRANSMAT : Matrix transfer**

*TRANSMAT* performs a second order transport of the particle coordinates in the following way

$$X_i = \sum_j R_{ij} X_j^0 + \sum_{j,k} T_{ijk} X_j^0 X_k^0$$

where,  $X_i$  stands for any of the current coordinates  $Y, T, Z, P$ , path length and momentum dispersion, and  $X_i^0$  stands for any of the initial coordinates.  $[R_{ij}]$  ( $[T_{ijk}]$ ) is the first order (second order) transport matrix as usually involved in second order beam optics [42]. Second order transport is optional. The length of the element represented by the matrix may be introduced for the purpose of path length updating.

Note : *MATRIX* delivers  $[R_{ij}]$  and  $[T_{ijk}]$  matrices in a format suitable for straightforward use with *TRANSMAT*.

**TRAROT : Translation-Rotation of the reference frame**

UNDER DEVELOPMENT. Check before use.

This procedure performs translation and rotation of the local **zgoubi** frame. It can be used for instance for skewing multipoles.

Relationship to spin tracking, particle decay or gas-scattering may not be fully installed, to be checked before use.

**UNDULATOR : Undulator magnet**

*UNDULATOR* magnet. UNDER DEVELOPMENT.

**UNIPOT : Unipotential cylindrical electrostatic lens**

The lens is cylindrically symmetric about the  $X$ -axis.

The length of the first (resp. second, third) electrode is  $L_1$  (resp.  $L_2, L_3$ ). The distance between the electrodes is  $D$ . The potentials are  $V_1$  and  $V_2$ . The inner radius is  $R_0$  (Fig. 41). The model for the electrostatic potential along the axis is [73]

$$V(X) = \frac{V_2 - V_1}{2\omega D} \left[ \ln \frac{\cosh \frac{\omega \left( x + \frac{L_2}{2} + D \right)}{R_0}}{\cosh \frac{\omega \left( x + \frac{L_2}{2} \right)}{R_0}} + \ln \frac{\cosh \frac{\omega \left( x - \frac{L_2}{2} - D \right)}{R_0}}{\cosh \frac{\omega \left( x - \frac{L_2}{2} \right)}{R_0}} \right]$$

( $x$  = distance from the center of the central electrode ;  $\omega = 1,318$  ; cosh = hyperbolic cosine), from which the field  $\vec{E}(X, Y, Z)$  and its derivatives are deduced following the procedure described in section 1.3.1.

Use *PARTICUL* prior to *UNIPOT*, for the definition of particle mass and charge.

The total length of the lens is  $L_1 + L_2 + L_3 + 2D$  ; stepwise integration starts at entrance of the first electrode and terminates at exit of the third one.

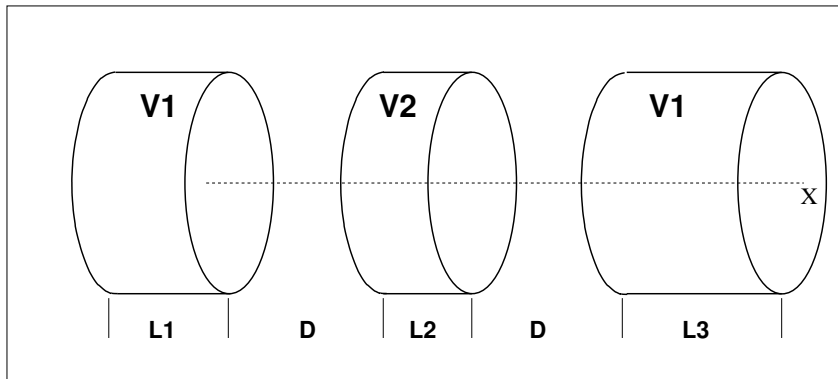


Figure 41: Three-electrode cylindrical unipotential lens.

**VENUS : Simulation of a rectangular shape dipole magnet**

*VENUS* is dedicated to a 'rough' simulation of SATURNE Laboratory's *VENUS* dipole. The field  $B_0$  is constant inside the magnet, with longitudinal extent  $XL$  and transverse extent  $\pm YL$  ; outside these limits,  $B_0 = 0$  (Fig. 42).

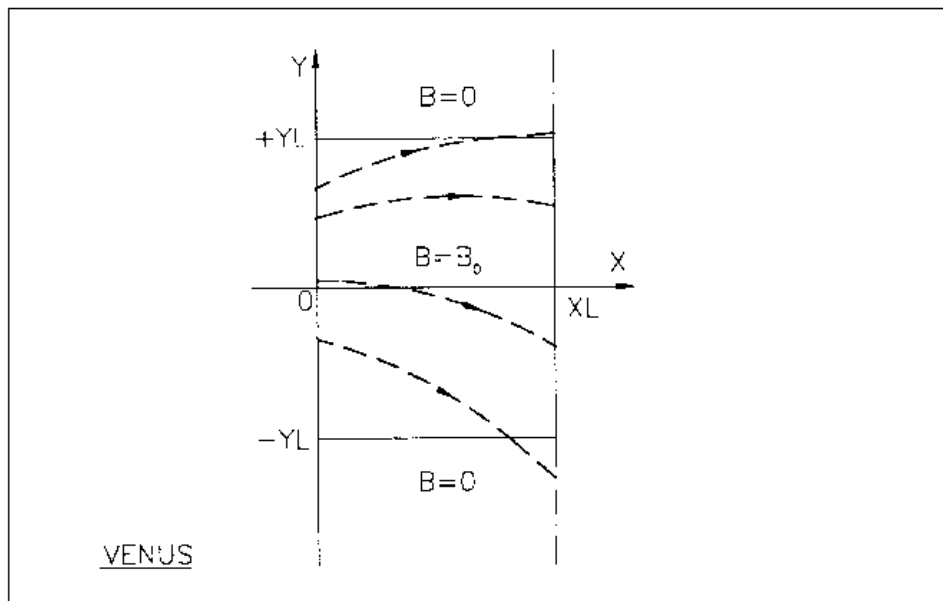


Figure 42: A scheme of a few trajectories through *VENUS* rectangular dipole.

**VFFA : Simulation of a VFFA dipole, or series of**

(page 356 in PART B)

VFFA simulates a vertical FFAG dipole, or a series of nearby dipoles (Fig. 43).

*TO BE DOCUMENTED FURTHER.*

Examples, including input data files, are available at  
<https://sourceforge.net/p/zgoubi/code/HEAD/tree/trunk/exemples/FFAG/VFFA/>

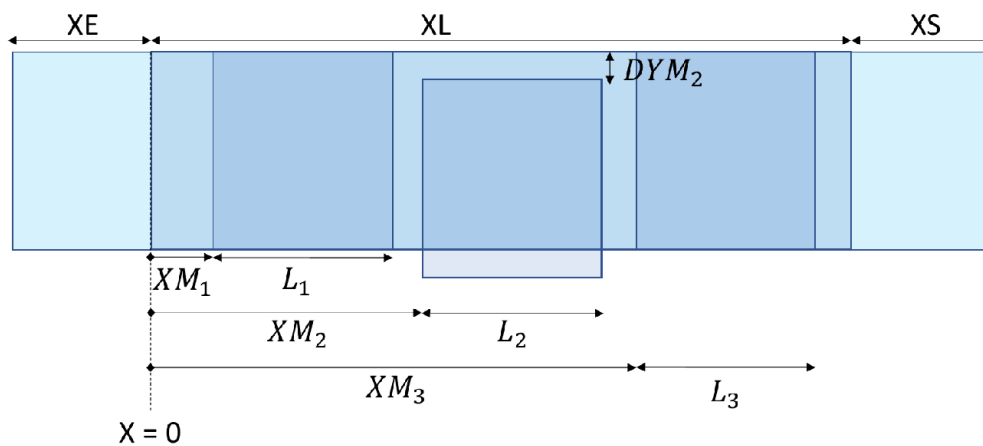


Figure 43: Multiple-dipole VFFA design parameters in **zgoubi** formalism.



**WIENFILT : Wien filter**

(page 357 in PART B)

**WIENFILT** simulates a Wien filter, with transverse and orthogonal electric and magnetic fields  $\vec{E}_Y, \vec{B}_Z$  (index KHV=1) or  $\vec{E}_Z, \vec{B}_Y$  (index KHV=1) (Fig. 37). It must be preceded by **PARTICUL** for the definition of particle mass and charge.

The length  $XL$  of the element is the distance between its entrance and exit EFBs. The electric and magnetic field intensities  $E_0$  and  $B_0$  in the central, uniform field region, normally satisfy the relation

$$B_0 = -\frac{E_0}{\beta_W c}$$

for the selection of “wanted” particles of velocity  $\beta_W c$ . Ray-tracing in field fall-off regions extends over a distance  $X_E$  ( $X_S$ ) beyond the entrance (exit) EFB by means of prior and further automatic change of frame. Four sets of coefficients  $\lambda, C_0 - C_5$  allow the description of the entrance and exit fringe fields outside the uniform field region, following the model [48]

$$F = \frac{1}{1 + \exp(P(s))}$$

where  $P(s)$  is of the term

$$P(s) = C_0 + C_1 \left(\frac{s}{\lambda}\right) + C_2 \left(\frac{s}{\lambda}\right)^2 + C_3 \left(\frac{s}{\lambda}\right)^3 + C_4 \left(\frac{s}{\lambda}\right)^4 + C_5 \left(\frac{s}{\lambda}\right)^5$$

and  $s$  is the distance to the EFB. When fringe fields overlap inside the element (*i.e.*,  $XL \leq X_E + X_S$ ), the field fall-off is expressed as

$$F = F_E + F_S - 1$$

where  $F_E$  ( $F_S$ ) is the fringe field factor for the entrance (exit) EFB.

If  $\lambda_E = 0$  ( $\lambda_S = 0$ ) for either the electric or magnetic component, then both are considered as sharp edge fields and  $X_E$  ( $X_S$ ) is forced to zero (for the purpose of saving computing time). In this case, the magnetic wedge angle vertical first order focusing effect is simulated at entrance and exit by a kick  $P_2 = P_1 - Z_1 \tan(\epsilon/\rho)$  applied to each particle ( $P_1, P_2$  are the vertical angles upstream and downstream the EFB,  $Z_1$  the vertical particle position at the EFB,  $\rho$  the local horizontal bending radius and  $\epsilon$  the wedge angle experienced by the particle ;  $\epsilon$  depends on the horizontal angle T). This is not done for the electric field, however it is advised not to use a sharp edge electric dipole model since this entails non symplectic mapping, and in particular precludes accounting for momentum effects of the non zero longitudinal electric field component.

**YMY : Reverse signs of  $Y$  and  $Z$  reference axes**

(page 358 in PART B)

$YMY$  performs a  $180^\circ$  rotation of particle coordinates with respect to the  $X$ -axis, as shown in Fig. 44. This is done by means of a change of sign of  $Y$  and  $Z$  axes, and therefore coordinates, as follows

$$Y_2 = -Y_1, \quad T_2 = -T_1, \quad Z_2 = -Z_1 \quad \text{and} \quad P_2 = -P_1$$

Also performs  $180^\circ$  X-rotation of spins.

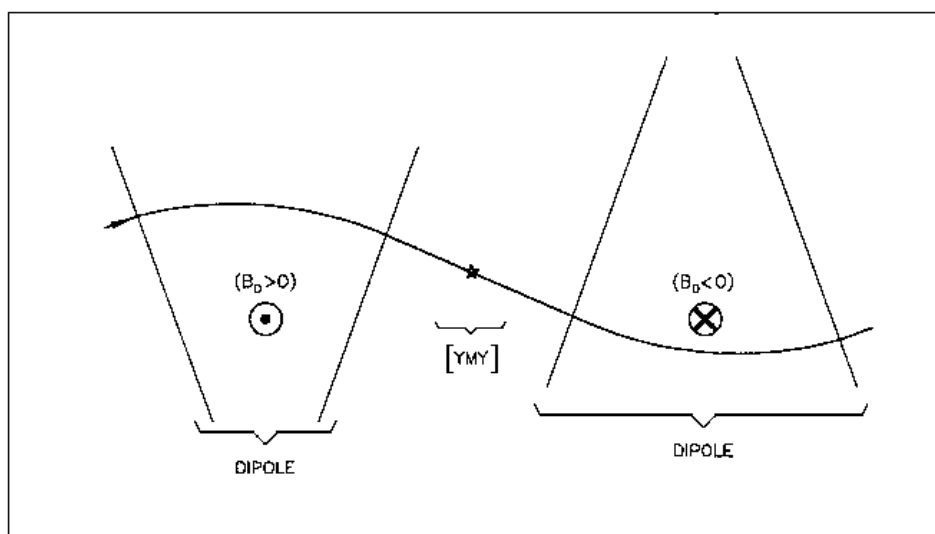


Figure 44: The use of  $YMY$  in a sequence of two identical dipoles of opposite signs.

## 6.4 Functional Options

A series of options are available which allow the control of various of the procedures and functionalities of the code.

Some of these options are normally declared right after the object definition, for instance

- *SPNTRK* : switch-on spin tracking,
- *PARTICUL* to declare particle mass and charge, if for instance tracking in electric fields, or tracking spin, or in presence of synchrotron radiation energy loss simulations,

some may appear further down in the optical sequence (in *zgoubi.dat*), for instance

- *MCDESINT* : switch-on in-flight decay, could be after a target,
- *REBELOTE* : for multi-turn tracking, including an extraction line section for instance,

others may normally be declared at the end of *zgoubi.dat* data pile, for instance

- *END* : end of a problem,
- *FIT* : fitting procedure - can also appear before *REBELOTE*

*GETFITVAL* may appear *before* the object definition (keyword *[MC]OBJET*, normally the first keyword in **zgoubi** optical sequence). This is the case if variables prior saved following a '*FIT[2]*' procedure and then read back using *GETFITVAL*, happen to belong in the *[MC]OBJET* keyword data list.

*SYSTEM* as well, may appear anywhere in the optical sequence.

**BINARY : *BINARY*↔*FORMATTED* data converter**

This procedure translates field map data files from “BINARY” to “FORMATTED” – in the *FORTTRAN* sense, or the other way.

The keyword is followed by, next data line,

$$NF[.J], NCOL, NHDR$$

the number of files to be translated [*READ* format option, a single digit integer, optional], number of data columns in the file, number of header lines in the file.

Fortran *READ* format, case of formatted input file :

If *J* is not given, the *NCOL* arrangement should be consistent with the following *FORTTRAN READ* statement :

```
READ (unit=ln, *) (X7(I), I=1, NCOL)
```

If *J* = 1, *NCOL* should be consistent with the following *FORTTRAN READ* statement :

```
READ (unit=ln, fmt='(1x, ncol*E11.2)') (X7(I), I=1, NCOL)
```

Fortran *READ* format, case of formatted input file :

Data are expected to have been recorded under the form of *Ncol* column rows.

Then follow, line by line, the *NF* names of the files to be translated.

*Iff* a file name begins with the prefix “B\_” or “b\_”, it is assumed “binary”, and hence converted to “formatted”, and given the same name after suppression of the prefix “B\_” or “b\_”. Conversely, *iff* the file name does not begin with “B\_” or “b\_”, the file is presumed “formatted” and hence translated to “binary”, and is given the same name after addition of the prefix “b\_”.

In its present state, the procedure *BINARY* only supports a limited number of read/write formats. Details concerning I/O formatting can be found in the *FORTTRAN* file `binary.f`.

**END or FIN : End of input data list**

The end of a problem, or of a set of several problems stacked in the data file, should be stated by means of the keywords *FIN* or *END*.

Any information following these keywords will be ignored.

The execution of a problem will also stop when *FIN* or *END* is met, unless *FINISH* appears in the sequence, prior.

In some cases, these keywords may cause some information to be printed in *zgoubi.res*, for instance when the keyword *PICKUPS* is used.

**ERRORS : Injecting errors in optical elements**

(page 284 in PART B)

*UNDER DEVELOPMENT*

The keyword *ERRORS* allows injecting diverse types of errors in optical elements, including field and positioning defects. Uniform and Gaussian distribution laws are available.

Optical elements concerned can be discriminated using one or both of their labels.

*ERRORS* is under development, it is only installed for some of the optical elements of **zgoubi** library (listed below) <sup>14</sup>, functionalities at the moment are the following :

*MULTIPOL* can be injected the following random defects:

Field defect on any one of its 2-pole (dipole) to 20-pole component. The defect can be relative (to the current field value) or absolute (adds to the current field value);

X-rotation of any of the 20 poles, the defect adds-up to possibly existing skew angle of pole;

Z-offset, independently of KPOS value; the defect adds-up to possibly existing Z-offset.

*TOSCA*, case of Cartesian mesh ( $MOD \leq 19$ ), can be injected a field defect that applies to the field map as a whole. The defect can be relative (to the current field value) or absolute (adds to the current field value).

Some rules :

- if *REBELOTE* is used in multi-turn mode (*i.e.*, its *K* argument value is 99), then the defect series, generated upon first pass, applies at all subsequent passes, by default.

Note: one might want instead to simulate a N-cell ring by repeating N times a cell subset, using *REBELOTE*. In which case defects of random type in that cell subset should be changed at each pass over the first N passes - and that N-cell sequence repeat identically beyond that. This possibility is not implemented, it can be, pending modification in `errors.f` program.

<sup>14</sup>For the geek: adding new elements takes place in the source file of the element concerned, for instance, `multpo.f` for *MULTIPOL*, `toscac.f` for *TOSCA[MOD ≤ 19]*.

**FINISH : End of execution**

The execution of a problem will stop if *FINISH* is met.

*FINISH* is useful in order to stop the execution before *END* (or *FIN*), for instance (but not exclusively) when the sequence uses *GOTO* keyword(s), or *REBELOTE*.

**FIT, FIT2 : Fitting procedure**

(page 284 in PART B)

The keywords *FIT*, *FIT2* allow the automatic adjustment of up to 20 variables, for fitting up to 20 constraints.

*FIT* was implemented in 1985, drawn from the matrix transport code BETA [42]. *FIT2* is a simplex method (Nelder-Mead method), it has been implemented in 2007 [43]. One or the other may converge faster, or may have some advantages/disadvantages, depending on the problem.

Any physical parameter of any element in *zgoubi.dat* optical sequence may be varied. Examples of available constraints are, amongst others :

- trajectory coordinates in the  $F(J, I)$  array,  $I$  = particle number,  $J$  = coordinate number = 1 to 7 for respectively  $D, Y, T, Z, P, S$  = path length, time ;
- spin coordinates ;
- any of the  $6 \times 6$  coefficients of the first order transport matrix  $[R_{ij}]$  as defined in the keyword *MATRIX* ;
- any of the  $6 \times 6 \times 6$  coefficients of the second order array  $[T_{ijk}]$  as defined in *MATRIX* ;
- any of the  $4 \times 4$  coefficients of the beam  $\sigma$ -matrix
- transmission efficiency of an optical channel.
- tunes  $\nu_{Y,Z}$  and periodic betatron functions  $\beta_{Y,Z}, \alpha_{Y,Z}, \gamma_{Y,Z}$ , as computed in the coupled hypothesis [44].

A full list of the constraints available is given in the table page 165.

*FIT*, *FIT2* are compatible with the use of (*i.e.*, can be encompassed in) *REBELOTE* for successive fitting trials using various sets of parameters (see pp. 167 181).

**VARIABLES**

The first input data in *FIT[2]* is the number of variables *NV*. A variable is defined by a line of data comprised of

$IR =$  number of the varied element in the optical sequence

$IP =$  number of the physical parameter to be varied in this element

$XC =$  coupling parameter. Normally  $XC = 0$ . If  $XC \neq 0$ , coupling will occur (see below)

and of the allowed variation range, either under the form

$DV =$  allowed relative range of variation of the physical parameter  $IP$

or under the form

$[Vmin, Vmax] =$  allowed interval of variation of the physical parameter  $IP$

**Numbering of the Elements (IR) :**

The elements (*i.e.*, keywords *DIPOLE*, *QUADRUPO*, etc.) as read by **zgoubi** in the *zgoubi.dat* sequence are assigned a number. which follows their sequence in the data file. It is that very number,  $IR$ , that the *FIT[2]* procedure uses. A simple way to get  $IR$  once the *zgoubi.dat* file has been built, is to do a preliminary run, since the first thing **zgoubi** does is copy the sequence from *zgoubi.dat* into the result file *zgoubi.res*, with all elements numbered.

**Numbering of the Physical Parameters (IP) :**

All the data that follow a keyword are numbered - except for *SCALING*, see below.



With most of the keywords, the numbering follows the principle hereafter :

<b>Input data</b>	<b>Numbering for FIT</b>
'KEYWORD'	
first line	1, 2, 3, ..., 9
second line	10, 11, 12, 13, ..., 19
this is a comment	a line of comments is skipped
next line	20, 21, 22, ..., 29
and so on...	

The examples of *QUADRUPO* (quadrupole) and *TOSCA* (Cartesian or cylindrical mesh field map) are as follows.

<b>Input data</b>	<b>Numbering for FIT</b>
'QUADRUPO'	
<i>IL</i>	1
<i>XL, R<sub>0</sub>, B</i>	10, 11, 12
<i>X<sub>E</sub>, λ<sub>E</sub></i>	20, 21
<i>NCE, C<sub>0</sub>, C<sub>1</sub>, C<sub>2</sub>, C<sub>3</sub>, C<sub>4</sub>, C<sub>5</sub></i>	30, 31, 32, 33, 34, 35, 36
<i>X<sub>S</sub>, λ<sub>S</sub></i>	40, 41
<i>NCS, C<sub>0</sub>, C<sub>1</sub>, C<sub>2</sub>, C<sub>3</sub>, C<sub>4</sub>, C<sub>5</sub></i>	50, 51, 52, 53, 54, 55, 56
<i>XPAS</i>	60
<i>KPOS, XCE, YCE, ALE</i>	70, 71, 72, 73

<b>Input data</b>	<b>Numbering for FIT</b>
'TOSCA'	
<i>IC, IL</i>	1, 2
<i>BNORM, X- [, Y-, Z-]NORM</i>	10, 11 [, 12, 13]
<i>TIT</i>	This is text
<i>IX, IY, IZ, MOD</i>	20, 21, 22, 23
<i>FNAME</i>	This is text
<i>ID, A, B, C [A', B', C', etc. if ID ≥ 2]</i>	30, 31, 32, 33 [34, 35, 36 [, 37, 38, 39] if ID ≥ 2]
<i>IORDRE</i>	40
<i>XPAS</i>	50
if <i>KPOS=1</i> :	
<i>KPOS, XCE, YCE, ALE</i>	60,61,62,63
if <i>KPOS=2</i> :	
<i>KPOS</i>	60
<i>KPOS, XCE, YCE, ALE</i>	70,71,72,73

A different numbering, fully sequential, has been adopted in the following elements :

*AIMANT, DIPOLE, EBMULT, ELMULT, MULTIPOL.*

It is illustrated here after in the case of *MULTIPOL* and *DIPOLE-M*.

<b>Input data</b>	<b>Numbering for FIT</b>
'MULTIPOL'	
0	1
365.760 10.0 7.5739 1.4939 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0	2, 3, 4, 5, ..., 13
10.0 4.0 0.80 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0	14, 15, ..., 24
<i>NC, C<sub>0</sub>, C<sub>1</sub>, C<sub>2</sub>, C<sub>3</sub>, C<sub>4</sub>, C<sub>5</sub></i>	25, 26, 27, 28, 29, 30, 31
10.0 4.0 0.80 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0	32, 33, ..., 42
<i>NC, C<sub>0</sub>, C<sub>1</sub>, C<sub>2</sub>, C<sub>3</sub>, C<sub>4</sub>, C<sub>5</sub></i>	43, 44, ..., 49
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.	50, 51, ..., 59
step size	60 [and 61, 62 reserved]
<i>KPOS, XCE, YCE, ALE</i>	63, 64, 65, 66

**Input data**

'DIPOLE-M'

NFACE, IC, IL

IAMAX, IRMAX

B<sub>0</sub>, N, B, G

AT, ACENT, RM, RMIN, RMAX

λ, ξ

NC, C<sub>0</sub>, C<sub>1</sub>, C<sub>2</sub>, C<sub>3</sub>, C<sub>4</sub>, C<sub>5</sub>, shiftω, θ, R<sub>1</sub>, U<sub>1</sub>, U<sub>2</sub>, R<sub>2</sub>

etc.

**Numbering for FIT**

1, 2, 3

4, 5

6, 7, 8, 9

10, 11, 12, 13, 14

15, 16

17, 18, 19, 20, 21, 22, 23, 24

25, 26, 27, 28, 29, 30

Parameters in *SCALING* also have a sequential numbering, yet some positions are skipped, this is illustrated in the example hereafter which covers all possible working modes of *SCALING* (all details regarding the numbering can be found in the *FORTRAN* subroutine *rscal.f*):

Input data	Numbering in FIT	Quantities to be varied (see SCALING for details)
'SCALING'		
<input type="checkbox"/> <input type="checkbox"/> 9	1 2	Non relevant
AGSMM *AF *BF		Keywords concerned, their labels
-1 3 12 <input type="checkbox"/> 13 <input type="checkbox"/> 14 <input type="checkbox"/>	3 4 5	dB1, dB2, dB3 parameters in AGSMM
<input type="checkbox"/> 7.2135	6	Field factor
<input type="checkbox"/> 1	7	Timing
AGSMM *AD *BD		
-1 3 12 <input type="checkbox"/> 13 <input type="checkbox"/> 14 <input type="checkbox"/>	8 9 10	
<input type="checkbox"/> 7.2135	11	
<input type="checkbox"/> 1	12	
AGSMM *CF		
-1 3 12 <input type="checkbox"/> 13 <input type="checkbox"/> 14 <input type="checkbox"/>	13 14 15	
<input type="checkbox"/> 7.2135	16	
<input type="checkbox"/> 1	17	
AGSQUAD QH_*		
3		
<input type="checkbox"/> 0.605 <input type="checkbox"/> 0.77 <input type="checkbox"/> 0.879	18 19 20	Field factor
<input type="checkbox"/> 1 <input type="checkbox"/> 2000 <input type="checkbox"/> 10000	21 22 23	Timing
AGSQUAD QV_*		
3		
<input type="checkbox"/> 0.587 <input type="checkbox"/> 0.83 <input type="checkbox"/> 0.83	24 25 26	
<input type="checkbox"/> 1 <input type="checkbox"/> 2000 <input type="checkbox"/> 10000	27 28 29	
MULTIPOL		General rule. Precedes labeled cases if any.
<input type="checkbox"/> -1	30	(meaningless)
<input type="checkbox"/> 0.72135154291	31	
<input type="checkbox"/> 1	32	(meaningless)
MULTIPOL COH1		
1.10		No numbering if 1.10 type of option
./Csnk3D/bump_centered.scal		
1 2		
MULTIPOL COH2		
1.10		
./Csnk3D/bump_centered.scal		
1 4		
MULTIPOL KICKH KICKV		
2		
<input type="checkbox"/> 0.1 <input type="checkbox"/> 0.3	33 34	Field factor
<input type="checkbox"/> 1 <input type="checkbox"/> 10	35 36	Timing

**Coupled Variables (XC)**

Coupling a variable parameter to any other parameter in the optical sequence is possible. This is done by giving *XC* a value of the form *int.jnt* (which can be either negative or positive, see below) where the integer part “int” is the number of the coupled element in the optical sequence (equivalent to *IR*, see above), and the decimal part “jnt” is the number of its parameter of concern (equivalent to *IP*, see above). For example,  $XC = 20.10$  or  $-20.10$  is a request for coupling with the parameter number 10 of element number 20 of the optical sequence,  $XC = 20.100$  or  $-20.100$  is a request for coupling with the parameter number 100 of element 20.

An element of the optical sequence which is coupled (by means of  $XC \neq 0$ ) to a variable declared in the data list of the *FIT[2]* keyword, needs not appear as one of the *NV* variables in that data list (this would amount to redundant information).

*XC* can be either positive or negative:

- if  $XC > 0$ , the sum of the two parameters stays constant [for example, two drifts in a quadrupole doublet can be coupled so that their total length stays constant (equal to its value before *FIT[2]* starts), while both drift lengths are varied];

- if  $XC < 0$ , the difference between the two parameters stays constant [for example, two homologous quadrupoles in a triplet can be coupled and have their fields stay equal while varied (assuming they are equal before *FIT[2]* starts)].

**Coupled Variables using SCALING**

Because it applies a scaling factor to families of optical elements (as determined by their keyword and possible *LABEL1*, *LABEL2*), *SCALING* can be used to couple *FIT* variables.

**Variable Variation Range**

There are two ways to define the allowed range for a variable, as follows.

(i) *DV* : For a variable (parameter number *IP* under some keyword) with initial value *v*, the *FIT[2]* procedure is allowed to explore the range  $v \times (1 \pm DV)$ .

(i) [*v<sub>min</sub>*, *v<sub>max</sub>*] : This specifies the allowed interval of variation.

**CONSTRAINTS**

The next input data in *FIT[2]* is the number of constraints, *NC*. A list of the available constraints is given in the table page 165 ; adding or changing a constraint resorts to the *FORTRAN* file *ff.f*.

Each constraint is defined by the following list of data :

<i>IC</i> [. <i>IC2</i> ]	=	type of constraint (see table p. 165).
<i>I</i> , <i>J</i>	=	constraint ( <i>i.e.</i> , $R_{ij}$ , determinant, tune ; $T_{ijk}$ ; $\sigma_{ij}$ ; trajectory # <i>I</i> and coordinate # <i>J</i> )
<i>IR</i>	=	number of the keyword at the exit of which the constraint applies ; instead of numeral, can also be '#End', so the constraint applies at the last keyword preceding ' <i>FIT[2]</i> '
<i>V</i>	=	desired value of the constraint
<i>W</i>	=	weight of the constraint (smaller <i>W</i> for higher weight)
<i>NP</i>	=	number of additional parameters ( $NP = 0$ unless otherwise specified in Tab. page 165); <i>NP</i> values $PR_{1-NP}$ follow, defining the constraint

*IC=0* : The coefficients  $\sigma_{11}$  ( $\sigma_{33}$ ) = horizontal (vertical) beta values and  $\sigma_{22}$  ( $\sigma_{44}$ ) = horizontal (vertical) derivatives ( $\alpha = -\beta'/2$ ) are obtained by transport of their initial values at line start as introduced using for instance *OBJET*, *KOBJ=5.1*.

*IC=0.1* : Beam parameters :  $\sigma_{11} = \beta_Y$ ,  $\sigma_{12} = \sigma_{21} = -\alpha_Y$ ,  $\sigma_{22} = \gamma_Y$ ,  $\sigma_{33} = \beta_Z$ ,  $\sigma_{34} = \sigma_{43} = -\alpha_Z$ ,  $\sigma_{44} = \gamma_Z$  ; periodic dispersion :  $\sigma_{16} = D_Y$ ,  $\sigma_{26} = D'_Y$ ,  $\sigma_{36} = D_Z$ ,  $\sigma_{46} = D'_Z$ , all quantities derived by assuming periodic optical sequence and identifying the first order transport matrix with the form  $I \cos \mu + J \sin \mu$ .

$IC=1, 2$  : The coefficients  $R_{ij}$  and  $T_{ijk}$  are calculated following the procedures described in *MATRIX*, option *IFOC* = 0. The fitting of the  $[R_{ij}]$  matrix coefficients supposes the tracking of particles with paraxial coordinates, normally defined using *OBJET* option *KOBJ*= 5 or 6.

$IC=3$  : If  $1 \leq I \leq IMAX$  then the value of coordinate type  $J$  ( $J = 1, 6$  for respectively  $D, Y, T, Z, P, S$ ) of particle number  $I$  ( $1 \leq I \leq IMAX$ ) is constrained. However  $I$  can take special meaning, as follows.

$I = -1$  : the constraint is the average value of coordinate of type  $J$  over particles  $[I_1, I_2]$ ,

$I = -2$  : the constraint is the maximum value of coordinate of type  $J$ ,

$I = -3$  : the constraint is the distance between particles  $I_1 - I_2$  (step  $\Delta I$ ).

$I = -4$  : take the average of the  $F(J, i)$  coordinate ( $J=1-7$  for  $D, Y, T, Z, P, S, \text{time}$ ) over  $(I_2 - I_1 + 1)$  particles in the range  $i = I_1 - I_2$ , get it at  $N$  different pickup locations  $PU_1, PU_2, \dots, PU_N$ . The constraint is, minimizing the distance between these averages,  $Dist [ < F(J, i) >_{i=I_1, I_2} ]_{PU_1-N}$  (Ex. : minimize the difference between PU records, for a series of PUs along a beamline). If the number of locations is 997 (*i.e.*,  $N+2=999$ ) then all PUs upstream of IR are accounted for. This constraint requires declaring pick-ups using *PICKUPS* keyword, their locations are as numbered in the optical sequence in *zgoubi.res*.

$IC=3.1$  : Absolute value of the difference between local and initial  $J$ -coordinate of particle  $I$  (convenient *e.g.* for closed orbit search).

$IC=3.2$  : Absolute value of the sum of the local and initial  $J$ -coordinate of particle number  $I$ .

$IC=3.4$  : Absolute value of the difference between local  $J$ -coordinates of particles respectively  $I$  and  $K$ .

$IC=3.5$  : Difference between local  $J$ -coordinates of particles respectively  $I$  and  $K$ , relative to  $J$ -coordinate of particle  $K$ .

$IC=4$  : The coefficients  $\sigma_{11}$  ( $\sigma_{33}$ ) = horizontal (vertical) beta values and  $\sigma_{22}$  ( $\sigma_{44}$ ) = horizontal (vertical) derivatives ( $\alpha = -\beta'/2$ ) are derived from an ellipse match of the current particle population (as generated for instance using *MCOBJET*, *KOBJ*=3).

The fitting of the  $[\sigma_{ij}]$  coefficients supposes the tracking of a relevant population of particles within an appropriate emittance.

$IC=5$  : The constraint value is the ratio of particles (over *IMAX*). Three cases possible :

$I = -1$ , ratio of particles still on the run.

$I = 1, 2, 3$ , *maximization* of the number of particles encompassed within a phase-space ellipse with given emittance value ( $I=1-3$  for respectively  $Y, Z, D$ ). Then,  $NP=1$ , followed by the emittance value. The center and shape of the ellipse are determined by an rms match to the position and shape of the particle distribution.

$I = 4, 5, 6$ , same as previous case, except for the surface of the ellipse, taken to be the surface of rms matched ellipse to the distribution. Thus  $NP=0$ .

$IC=7$  series : The constraint applies *inside* the optical element with number *IR*. This constraint may be (possibly combinations of) the minimum or maximum values of the coordinates or fields, or the value of field integrals, along the particle trajectory over the extent  $[X_0, X_{LIM}]$  of the element (regardless of the transverse trajectory coordinates  $Y$  and  $Z$ ). Parameters  $PR_2 = X_1$  and  $PR_3 = X_2$  impose that the constraint only be applied within a limited interval  $[X_1, X_2] \subset [X_0, X_{LIM}]$  inside the element. Note in that latter case, the element extent  $[X_0, X_{LIM}]$ , in proper units, can be determined from a preliminary run with  $IL=1$  (see Sec. 7.6, p. 213).

$IC=7.1$  : Minimum or maximum value as reached by coordinate  $J$  of particle number  $I$ .

$IC=7.2$  : Maximum minus minimum of the value of coordinate  $J$  of particle number  $I$ .

$IC=7.3$  : Maximum plus minimum of the value of coordinate  $J$  of particle number  $I$ .

$IC=7.6$  : Minimum ( $PR_1 = 1$ ) or maximum ( $PR_1 = 2$ ) value as reached by the  $X, Y$  or  $Z$  component of the field along the trajectory of particle number  $I$ .

$IC=7.7$  : Maximum minus minimum of the values taken by the  $X, Y$  or  $Z$  component of the field along the trajectory of particle number  $I$ .

Type of constraint	Parameters defining the constraints (see also footnotes)					Additional parameter(s) NP	Param. values, $pr_1 - pr_{NP}$	Recommended [MC]OBJET (not an exclusive list) ; comments
	IC	I	J	Constraint (all are defined and managed in ff.f)				
<b>Transported <math>\sigma</math>-matrix</b> ( $\sigma(s) = T\sigma(0)\bar{T}$ )	0	1-6	1-6	$\sigma_{IJ}$ ( $\sigma_{11} = \beta_Y, \sigma_{21, 12} = \alpha_Y$ , etc.)				OBJET[KOBJ=5.1]
<b>Periodic <math>\sigma</math>-matrix</b> ( $\sigma = I \cos\mu + J \sin\mu$ ), <b>tunes</b> ( $N = 1 - 9$ for MATRIX block 1-9)	0.N ( $N \leq 9$ )	1-6 7 8 9 10	1-6 [0, 1] [0, 1] [-1, 1] [-1, 1]	$\sigma_{IJ}$ ( $\sigma_{11} = \cos\mu_Y + \alpha_Y \sin\mu_Y$ , etc.) $\mu_Y/2\pi$ $\mu_Z/2\pi$ $\cos(\mu_Y)$ $\cos(\mu_Z)$				OBJET[KOBJ=5] if $N = 1$ , OBJET[KOBJ=5.N] if $N \geq 2$ .
<b>First order transport coeffs. &amp; determinants.</b>	1	1-6 7 8	1-6 j j	Transport coeff. $R_{IJ}$ $j \neq 8$ : YY-determinant ; $j=8$ : YZ-det. $j \neq 7$ : ZZ-determinant ; $j=7$ : ZY-det.				OBJET[KOBJ=5]
<b>Chromas.</b>	1.1	7 8	7 8	$dQ_Y / dp/p$ $dQ_Z / dp/p$				OBJET[KOBJ=5.2] or KOBJ=5.3
<b>Second order transport coeffs.</b>	2	1-6	11-66	Transport coeff. $T_{I,j,k}$ ( $j = [J/10], k = J - 10[J/10]$ )				OBJET[KOBJ=6]
<b>Trajectory coordinates</b>	3	1-IMAX -1	1-7 1-7	$F(J, I)$ $\langle F(J, i) \rangle$ $i = \begin{cases} 1, IMAX & \text{if } NP=0 \\ 11, 12 & \text{if } NP=2 \end{cases}$	0 [or 2]	$[I_1, I_2]$		[MC]OBJET $1 \leq I_1 \leq I_2 \leq IMAX$
(I = particle number; J=1-7 for D, Y, T, Z, P, S, time)		-2 -3 -4	1-7 1-7 1-7	$Sup( F(J, i) )_{i=1, IMAX}$ $Dist [F(J, I)_{i=I_1, I_2}]$ $Dist [PU_i, i = 1, N]$	3 2	$I_1, I_2, \Delta I$ $NOEL_A, NOEL_B$		$1 \leq I_1 \leq I_2 \leq IMAX$ PU range
	3.1	1-IMAX	1-7	$ F(J, I) - FO(J, I) $	1	K		$K \leq IMAX$
	3.2	1-IMAX	1-7	$ F(J, I) + FO(J, I) $	1	K		$K \leq IMAX$
	3.4	1-IMAX	1-7	$ F(J, I) - F(J, K) $	1	K		$K \leq IMAX$
	3.5	1-IMAX	1-7	$(F(J, I) - F(J, K))/F(J, K)$	1	K		$K \leq IMAX$
<b>Ellipse parameters</b>	4	1-6	1-6	$\sigma_{IJ}$ ( $\sigma_{11} = \beta_Y$ , $\sigma_{12} = \sigma_{21} = \alpha_Y$ , etc.)				OBJET[KOBJ=8] ; MCOBJET[KOBJ=3]
<b>Number of particles</b>	5	-1 1-3 4-6	$\leq 1$ $\leq 1$ $\leq 1$	$N_{survived}/IMAX$ $N_{in \epsilon_{Y,Z,X}}/N_{survived}$ $N_{in \text{ best } \epsilon_{Y,Z,X,rms}}/N_{survived}$	1	$\epsilon_{Y,Z,X}/\pi$		OBJET MCOBJET MCOBJET
<b>Coordinates &amp; fields, across optical elements</b>	7.1	1-IMAX	1-7	min. (param.=1) or max. (2) of $F(J, I)$	1 [or 3]	1 or 2 [, X1, X2]		[MC]OBJET
	7.2	1-IMAX	1-7	$\max(F(J, I)) - \min(F(J, I))$	0 [or 2]	[X1, X2]		
	7.3	1-IMAX	1-7	$\min(F(J, I)) + \max(F(J, I))$	id.	id.		
(J=1, 2, 3 for $B_X, Y, Z$ )	7.6	1-IMAX	1-3	min. or max. value of $B_J$	1 [or 3]	1 or 2 [, X1, X2]		
	7.7	1-IMAX	1-3	$\max(B_J) - \min(B_J)$	0 [or 2]	[X1, X2]		
	7.8	1-IMAX	1-3	$\min(B_J) + \max(B_J)$	id.	id.		
	7.9	1-IMAX	1-3	$\int B_J ds$	id.	id.		
	7.10	1-IMAX	1-7	to be documented	id.	id.		
<b>Spin</b>	10	1-IMAX	1-4	$S_{X,Y,Z}(I),  \vec{S}(I) $				SPNTRK+
	10.1	1-IMAX	1-3	$ S_{X,Y,Z}(I) - SO_{X,Y,Z}(I) $				[MC]OBJET
	10.2	1-IDMAX	any	spin precession angle ( $[-\pi, \pi]$ )				OBJET[KOBJ=2]
	10.3	1-IDMAX	1-3	X, Y or Z component of precession axis	0 [or 1]	[1]		OBJET[KOBJ=1,2]
	10.4	1-IDMAX	any	fractional spin tune				OBJET[KOBJ=2]
	10.5	1-IMAX	1-3	$\langle S_{X,Y,Z} \rangle_{N=1, I \text{ particles}}$				OBJET[KOBJ=2]

◇ IMAX: total number of particles tracked.

◇  $F(J, I)$  : particle coordinate array ; J: coordinate (1-7 for respectively D, Y, T, Z, P, S, time, resp. units: none, cm, mrad, cm, mrad, cm,  $\mu s$ ) ; I: particle number.

◇  $FO(J, I)$  : initial particle coordinates.

◇  $S(J, I)$  : spin coordinate array ; J: coordinate (1, 2, 3 for respectively X, Y, Z) ; I: particle number.

◇  $SO(J, I)$  : initial spin coordinates.

◇ Constraints 7.1 - 7.9: X1 and X2, optional, define the longitudinal coordinate interval inside the optical element, over which it applies.

◇ Constraints 10.2 - 10.4: I is the number of the concerned 3-particle momentum group, among the IDMAX momentum groups defined by OBJET[KOBJ=2, IDMAX]; if only one group needed, OBJET[KOBJ=1, IMAX=3] may be used as well.

◇ Constraint 10.3: If NP=1 and  $pr_{NP}=1$ , then the absolute value of the X, Y or Z component is accounted for.

$IC=7.8$  : Maximum plus minimum of the values taken by the X, Y or Z component of the field along the trajectory of particle number  $I$ .

$IC=7.9$  : Integral of the X, Y or Z component of the field as experienced by particle number  $I$ , across the optical element.

$IC=7.10$  : constraint on minimum value of coordinate  $J$  of particle  $I$  reached inside optical element  $K$ , and on maximum value of coordinate  $J2$  of particle  $I2$  inside very optical element  $K$ . Example of use: in a linear FFAG quadrupole doublet cell, it allows centering the orbits in the QF quadrupole such that extreme excursions (for instance, coordinate  $(J,I)$  [resp.  $(J2,I2)$ ] reaching extreme negative (resp. positive) value) are equal in absolute value and opposite in sign.

$IC=10$  : If  $1 \leq I \leq IMAX$  then the constraint is the value of coordinate type  $J$  ( $J = 1, 3$  for respectively  $S_X, S_Y, S_Z$ ) of particle number  $I$ .

$IC=10.1$  : Difference between final and initial  $J$ -spin coordinate of particle  $I$  (convenient *e.g.* for  $\vec{n}_0$  spin vector search).

$IC=10.2$  : Constraint on spin rotation angle for particles of momentum group # $K$ . This requires  $OBJET/KOBJ=2$ , with groups of 3 particles, all particles in a group have the same momentum, and their spins in direction, respectively, X, Y, Z.

$IC=10.3$  : Constraint on spin rotation axis for particles of momentum group # $K$ . This requires  $OBJET/KOBJ=2$ , with groups of 3 particles, all particles in a group have the same momentum, and their spins in direction, respectively, X, Y, Z.

### Constraining tunes

The example in the left column, Tab. 2, matches the (fractional) tunes in RHIC ring. It requires  $OBJET[KOBJ=5]$ , which creates a 13-particle sample proper to allow computation of transport coefficients (as done by *MATRIX*), from what optical functions and tunes are drawn.

### Constraining chromaticities

The example in the right column, Tab. 2, matches both the (fractional) tunes and the chromaticities in RHIC ring. It requires  $OBJET[KOBJ=5.3]$  to create three 13-particle sets of sample particles, one set centered on the on-momentum closed orbit, the other two centered on respectively  $\pm\delta p/p$  chromatic closed orbits ( $OBJET[KOBJ=5.2]$  also works and creates two sets only, on-momentum and at  $\delta p/p$ ). On-momentum as well as off-momentum 1st order transport coefficients are computed from these 13-particle sets (the same way as *MATRIX* does), from what optical functions and tunes are drawn. Chromaticities are given by the difference in the tunes obtained from the on-momentum, 13-particle set and the off-momentum, 13-particle set(s); this interpolation yields better accuracy in the 3-set case ( $OBJET[KOBJ=5.3]$ ).

### OBJECT DEFINITION

Depending on the type of constraint (see table p. 165), constraint calculations are performed either from transport coefficient calculation and in such case require *OBJET* with either  $KOBJ = 5$  or  $KOBJ = 6$ , or from particle distributions and in this case need object definition using for instance *OBJET* with  $KOBJ = 8$ , *MCOBJET* with  $KOBJ = 3$ .

### THE FITTING METHODS

The *FIT* procedure was drawn from the matrix transport code BETA [42]. It is a direct sequential minimization of the quadratic sum of all errors (*i.e.*, differences between desired and actual values for the *NC* constraints), each normalized by its specified weight  $W$  (the smaller  $W$ , the stronger the constraint).

Table 2: Fitting tunes (left column), or tunes and chromaticities (right column), using *SCALING*. Note the following in the latter case: *OBJET* specifies on-momentum as well as chromatic closed orbit coordinates for the three reference trajectories (respectively  $D=1$ ,  $D=1.0001$  and  $D=0.999$ ); however, a different way is to include the search for closed orbit coordinates as part of the *FIT* request, just specifying the  $D$  value for each one of the the references.

```

Fitting SCALING parameters
'OBJET'
555.739008456e3
5
! Create one set of 13 particles.
.001 .01 .001 .01 .001 .0001
0. 0. 0. 0. 0. 1.

'PARTICUL'
HELION
'FAISCEAU'
'SCALING'
1 6
BEND
-1
555.739008456
1
MULTIPOL
-1
555.739008456
1
MULTIPOL QUADQF ! QF family quads all have LABEL!=QUADQF
-1
555.569221809 ! FIT variable #12
1
MULTIPOL QUADQD ! QD family quads all have LABEL!=QUADQD
-1
554.852506166 ! FIT variable #16
1
MULTIPOL SEXTSF ! SF family sextus all have LABEL!=SEXTSF
-1
555.739008456 ! FIT variable #20
1
MULTIPOL SEXTSD ! SD family sextus all have LABEL!=SEXTSD
-1
555.739008456 ! FIT variable #24
1

'INCLUDE'
1
./RHICSequence[RHIC$START:RHIC#END]

'FIT'
2
4 12 0 .2 ! Vary QF family
4 16 0 .2 ! Vary QD family
4 20 0 .2 ! Vary SF family
4 24 0 .2 ! Vary SD family
4
0.1 7 7 #End 0.6850251017 1. 0 ! frac. QY
0.1 8 8 #End 0.6730125830 1. 0 ! frac. QZ
'END'

Fitting SCALING parameters
'OBJET'
555.739008456e3
5.3 ! Create three sets of 13 particles.
.001 .01 .001 .01 .001 .0001
0. 0. 0. 0. 0. 1. ! Define on-momentum 13-particle set.
0.022675 0.034243 0. 0. 0. 1.0001 ! Define dp/p=+1e-4 off-momentum 13-particle set,
-0.022675 -0.034243 0. 0. 0. 0.9999 ! define dp/p=-1e-4 off-momentum 13-particle set,
! initial coordinates are chromatic coordinates.

'PARTICUL'
HELION
'FAISCEAU'
'SCALING'
1 6
BEND
-1
555.739008456
1
MULTIPOL
-1
555.569221809
1
MULTIPOL QUADQF ! QF family quads all have LABEL!=QUADQF
-1
555.569221809 ! FIT variable #12
1
MULTIPOL QUADQD ! QD family quads all have LABEL!=QUADQD
-1
554.852506166 ! FIT variable #16
1
MULTIPOL SEXTSF ! SF family sextus all have LABEL!=SEXTSF
-1
555.739008456 ! FIT variable #20
1
MULTIPOL SEXTSD ! SD family sextus all have LABEL!=SEXTSD
-1
555.739008456 ! FIT variable #24
1

'INCLUDE'
1
./RHICSequence[RHIC$START:RHIC#END]

'FIT'
4
4 12 0 .2 ! Vary QF family
4 16 0 .2 ! Vary QD family
4 20 0 .2 ! Vary SF family
4 24 0 .2 ! Vary SD family
4
0.1 7 7 #End 0.6850251017 1. 0 ! frac. QY
0.1 8 8 #End 0.6730125830 1. 0 ! frac. QZ
1.1 7 7 #End 1. 1. 0 ! xi_Y
1.1 8 8 #End 1. 1. 0 ! xi_Z
'END'

```

The step sizes for the variation of the physical parameters depend on their initial values, and cannot be accessed by the user. At each iteration, the optimum value of the step size, as well as the optimum direction of variation, is determined for each one of the  $NV$  variables. Then follows an iterative global variation of all  $NV$  variables, until the minimization fails which results in a next iteration on the optimization of the step sizes.

The *FIT2* procedure is based on the Nelder-Mead method, it has various specificities, details can be found in Ref. [43].

The optimization process may be stopped by means of a penalty value, or a maximum number of iterations on the step size or on the call to the function.

### COMBINING FIT[2] AND REBELOTE

*FIT[2]* may be followed by the keyword *REBELOTE*. This allows for instance executing again the fitting procedure, following a change, by *REBELOTE*, in the value of some parameter(s) (about any parameter) in *zgoubi.dat* optical sequence.

The values of the variables in *FIT[2]* itself (and their weights as well) can be changed by *REBELOTE*.

More on that can be found under *REBELOTE*, Sec. 6.4, page 181.

*EXAMPLE* - Examples are given page 285.

Conversely, *REBELOTE* may be followed by the keyword *FIT[2]*. This allows for instance optimization of constraints that require multi-turn tracking (*e.g.*, optimized tune-jump quadrupole settings for beam acceleration through depolarizing resonances).

*AN UPDATED SEQUENCE FILE*

When a zgoubi run includes *FIT[2]*, an updated input data file, ready to use, is saved under the name *zgoubi.FIT.out.dat* at the end of the run.

***FIT[2] OPTIONS***

*FIT* and *FIT2* accept various options, as follows.

- 'nofinal' option : a request to avoid a final run that would normally follow the completion of the fit, allowing for the variable values as just determined by *FIT[2]*. By default, *i.e.*, if 'nofinal' does not appear in the *FIT[2]* keyword data list, then that final run does occur. The latter is necessary in particular when it is desired to store resulting particle data, in two cases for instance:

- when using *FAISTORE*, particle data desired at the sole very final pass.

- when using *IL=2* in some optical element(s): in that case a final run would allow print out of trajectories with final FIT variable states.

- 'save' [, *FileName*] option : a request for the variable values as resulting from *FIT[2]* to be saved, by default in *zgoubi.FITVALS.out*, or in *FileName* file if the latter is specified following the 'save' instruction.

- Penalty : value of the penalty to be reached, following what the *FIT[2]* procedure will be considered completed, and thus stopped.

- ITER : maximum allowed number of "calls to the function" (namely, (recursive) calls to *zgoubi*). Once that number is reached, *FIT[2]* is stopped, regardless of a possible "penalty" value.

*EXAMPLE* - An example using all five options, "nofinal", "save [*FileName*]", "penalty" and "ITER", is given in page 288.



**GASCAT : Gas scattering**

Modification of particle momentum and velocity vector, performed at each integration step, under the effect of scattering by residual gas.

*INSTALLATION IS TO BE COMPLETED.*

**GETFITVAL : Get values of variables as saved from former FIT[2] run**

This keyword allows reading, from a file whose name needs be specified, parameter values to be assigned to optical elements in zgoubi.dat.

That file is expected to contain a copy-paste of the data under the *FIT[2]* procedure as displayed in zgoubi.res, normally under the form

```

STATUS OF VARIABLES (Iteration # 95)
LMNT  VAR  PARAM  MINIMUM  INITIAL  FINAL  MAXIMUM  STEP  NAME  LBL1  LBL2
 145   1    4    -3.000E+03  762.    761.9484791  3.000E+03  1.254E-05  MULTIPOL  HKIC  DHCB02
 182   2    4    -1.000E+03 -231.   -230.9846875  1.000E+03  4.182E-06  MULTIPOL  HKIC  DHCB08
 146   3    4    -1.000E+03 -320.   -319.8554171  1.000E+03  4.182E-06  MULTIPOL  VKIC  DVCB02
 183   4    4    -1.000E+03  528.    527.7249064  1.000E+03  4.182E-06  MULTIPOL  VKIC  DVCB08
 615   5    4    -3.000E+03  308.    307.6860565  3.000E+03  1.254E-05  MULTIPOL  HKIC  DHCF02
 651   6    4    -1.000E+03 -114.   -113.8490362  1.000E+03  4.182E-06  MULTIPOL  HKIC  DHCF08
 616   7    4    -1.000E+03 -78.9   -78.88730937  1.000E+03  4.182E-06  MULTIPOL  VKIC  DVCF02
 652   8    4    -1.000E+03  212.    211.8789183  1.000E+03  4.182E-06  MULTIPOL  VKIC  DVCF08
# STATUS OF CONSTRAINTS
# TYPE  I  J  LMNT#  DESIRED  WEIGHT  REACHED  KI2  *  Parameter(s)
# 3  1  2  127  0.0000000E+00  1.0000E+00  1.0068088E-08  6.0335E-01 *  0 :
# 3  1  3  127  0.0000000E+00  1.0000E+00  7.0101405E-09  2.9250E-01 *  0 :
# 3  1  4  127  0.0000000E+00  1.0000E+00  2.9184383E-10  5.0696E-04 *  0 :
# 3  1  5  127  0.0000000E+00  1.0000E+00  3.1142381E-10  5.7727E-04 *  0 :
# 3  1  2  436  0.0000000E+00  1.0000E+00  3.8438378E-09  8.7944E-02 *  0 :
# 3  1  3  436  0.0000000E+00  1.0000E+00  1.5773011E-09  1.4808E-02 *  0 :
# 3  1  4  436  0.0000000E+00  1.0000E+00  2.2081272E-10  2.9022E-04 *  0 :
# 3  1  5  436  0.0000000E+00  1.0000E+00  5.7930552E-11  1.9975E-05 *  0 :
# Function called 1859 times
# Xi2 = 1.68006E-16 Busy...

```

A '#' at the beginning of a line means it is commented, thus it will not be taken into account. However a copy-paste from zgoubi.res (which is the case in the present example) would not need any commenting.

Since some of the *FIT[2]* variables may belong in *[MC]OBJET*, *GETFITVAL* may appear upstream of *[MC]OBJET* in zgoubi.dat, to allow updating the latter.

Once completed, *FIT[2]* will be followed by a final run of zgoubi.dat with variable values updated as resulting from the fit. This can be inhibited by indicating 'nofinal' option in *FIT[2]* (see page 168). For that final run *GETFITVAL* will be inhibited so avoid overriding the updated variable values.

**GOTO : Branching statement**

*TO BE DOCUMENTED FURTHER.*

*GOTO* is a conditional goto, working much like the fortran branching statement. It sends the execution pointer to some keyword (*e.g.*, “*MARKER*”) in the optical sequence, characterized by one or two specific labels.

Available options :

- pass number

*GOTO* accepts *GOBACK* as an argument, this sends the execution pointer back to the next keyword immediately following the calling *GOTO*.

*GOTO/GOBACK* for instance can be used in RLA type of optical sequence, to direct the **zgoubi** amongst the series of spreaders of mergers.

*EXAMPLE* - An example of an input data file of a recirculating linac, structured using *GOTO* statements, can be found in page 298.

**INCLUDE : File include statement**

(page 245 in PART B)

*INCLUDE* works much like the fortran “include” statement : it allows a fragment of an ancillary input data file to be taken from a separate file : when *INCLUDE* is met, the content of the ancillary file, or part of it as stated, is inserted in the data sequence. That file can be any ascii file, including e.g. your latest latex report, from which *INCLUDE* is requested to find a sequence of lines of text (presumably, a zgoubi.dat style sequence of lines).

The *INCLUDE* statement takes the following arguments :

- the number of ancillary input files to include a fragment from,
- their names, in as many lines, one name per line, and possibly, for each file, a preceding repeat factor - an integer - telling how many times that file has to be INCLUDED.

The file name may be followed by '*[LBL\_1A,LBL\_2A:LBL\_1B,LBL\_2B]*', with any of these four labels optional, as follows :

- *LBL\_1A* [*LBL\_2A*] identify to the first [and possibly second] label of some keyword in the include file, from which the include process will start (whatever precedes will be ignored). If none of '*LBL\_1A*, *\_2A*' is declared, the include process starts from the beginning of the include file. Declaring the second label only takes the form '*[LBL\_2A:LBL\_1B,LBL\_2B]*'.

- *LBL\_1B* [*LBL\_2B*] identify to the first [and second] label of a *MARKER* or some data-less keyword (such as *FAISCEAU*, ...) in the include file, at which the include process will end (whatever follows will be ignored). If none of '*LBL\_1B*, *\_2B*' is declared, the include file will be included down to its end. Declaring the second label only takes the form '*[LBL\_1A,LBL\_2A: ,LBL\_2B]*'.

Because **zgoubi** uses the reserved name *zgoubi.dat* as its standard working data file name, and as the presence of *INCLUDE* will cause the input data list to be changed (it is expanded to include the content of all INCLUDED files), the user's input data file is no longer allowed the name *zgoubi.dat* if it contains *INCLUDE* keyword(s) (*zgoubi.dat* is reserved as the name of the expanded input file to **zgoubi**). This is overcome by using the command '*zgoubi -fileIn myFile*' instead, with *myFile*≠*zgoubi.dat* (see Sec. 1.2 in PART D of the guide, page 383).

An *INCLUDE* file may contain one or more *INCLUDE*. There is a limit in the allowed depth of *INCLUDE* statements within *INCLUDE* files, however high. **zgoubi** will warn the user when it is exceeded. It can be changed in *prdata.f*.

By contrast with other keywords, *INCLUDE* is not part on the LSTKEY.H keyword list (found in the '*include*' folder). Instead, possible presence of *INCLUDE* in the input data file is dealt with by *prdata* program (*prdata.f* source file) in a simple way : the input data file is expanded so eliminating all *INCLUDE* occurrences, and the expanded, complete, data list is stored in the standard **zgoubi** input data file *zgoubi.dat*. Thus **zgoubi** eventually actually works using a regular, complete, data list in a single input file, *zgoubi.dat*.

*EXAMPLE* - An example of an input data file of a recirculating linac, structured using *INCLUDE* statements, can be found in page 298.

**MCDESINT : Monte-Carlo simulation of in-flight decay**

(page 303 in PART B)

When *MCDESINT*[45] is met in an optical sequence (normally, after *OBJET* or after *CIBLE*), in-flight decay simulation starts. It must be preceded by *PARTICUL* for the definition of mass  $M_1$  and *COM* lifetime  $\tau_1$  of the parent particle.

The two-body decay simulated is

$$1 \longrightarrow 2 + 3$$

The decay is isotropic in the center of mass. 1 is the incoming particle, with mass  $M_1$ , momentum  $p_1 = \gamma_1 M_1 \beta_1 c$  (relative momentum  $D_1 = \frac{p_1}{q} \frac{1}{BORO}$  with *BORO* = reference rigidity, defined in [*MC*]*OBJET*).

2 and 3 are decay products with respective masses and momenta  $M_2, M_3$  and  $p_2 = \gamma_2 M_2 \beta_2 c, p_3 = \gamma_3 M_3 \beta_3 c$ . The mean decay distance  $s_1$  is related to the center of mass lifetime  $\tau_1$  by

$$s_1 = c\tau_1 \sqrt{\gamma_1^2 - 1}$$

The individual distance  $s_d(I)$  to the decay point is calculated for each one of the  $I=1, I_{MAX}$  particles defined by [*MC*]*OBJET*, from a random number  $0 < R_1 \leq 1$ , using the exponential decay rule

$$_d s = -s_1 \ln R_1$$

After decay of the parent particle 1, particle 2 will be ray-traced with assumed positive charge, while particle 3 is discarded. Its scattering angles in the center of mass  $\theta^*$  and  $\phi$  are generated from two other random numbers  $0 < R_2 \leq 1$  and  $0 < R_3 \leq 1$  using

$$\begin{aligned} \theta^* &= \text{acos}(1 - 2R_2) & (0 < \theta^* \leq \pi) \\ \phi &= 2\pi R_3 & (0 < \phi \leq 2\pi) \end{aligned}$$

$\phi$  is a relativistic invariant, and  $\theta$  in the laboratory frame (Fig. 45) is given by

$$\tan \theta = \frac{1}{\gamma_1} \frac{\sin \theta^*}{\frac{\beta_1}{\beta_2^*} + \cos \theta^*}$$

$\beta_2^*$  and momentum  $p_2$  are given by

$$\begin{aligned} \gamma_2^* &= \frac{M_1^2 + M_2^2 - M_3^2}{2M_1 M_2} \\ \gamma_2 &= \gamma_1 \gamma_2^* (1 + \beta_1 \beta_2^* \cos \theta^*) \\ \beta_2 &= \left(1 - \frac{1}{\gamma_2^2}\right)^{1/2} \\ p_2 &= M_2 \sqrt{\gamma_2^2 - 1} \end{aligned}$$

Finally,  $\theta$  and  $\phi$  are transformed into the angles  $T_2$  and  $P_2$  in the **zgoubi** frame, and the relative momentum takes the value  $D_2 = \frac{p_2}{q} \frac{1}{BORO}$  (where *BORO* is the reference rigidity, see *OBJET*), while the starting position of  $M_2$  is the very location of the parent particle decay,  $(Y_1, Z_1, s_1)$ .

The decay simulation by **zgoubi** satisfies the following procedures. In optical elements and field maps, after each integration step *XPAS*, the current path length of the particle ( $F(6, I)$  in the source code) is compared to decay point location  $s_d$ . If  $s_d$  is passed, then the particle is considered to have decayed at  $F(6, I) - \frac{XPAS}{2}$  (*XPAS* is the integration step), at a position obtained by a linear translation from the position at  $F(6, I)$ . Presumably, the smaller *XPAS*, the smaller the error on position and angles at the decay point.

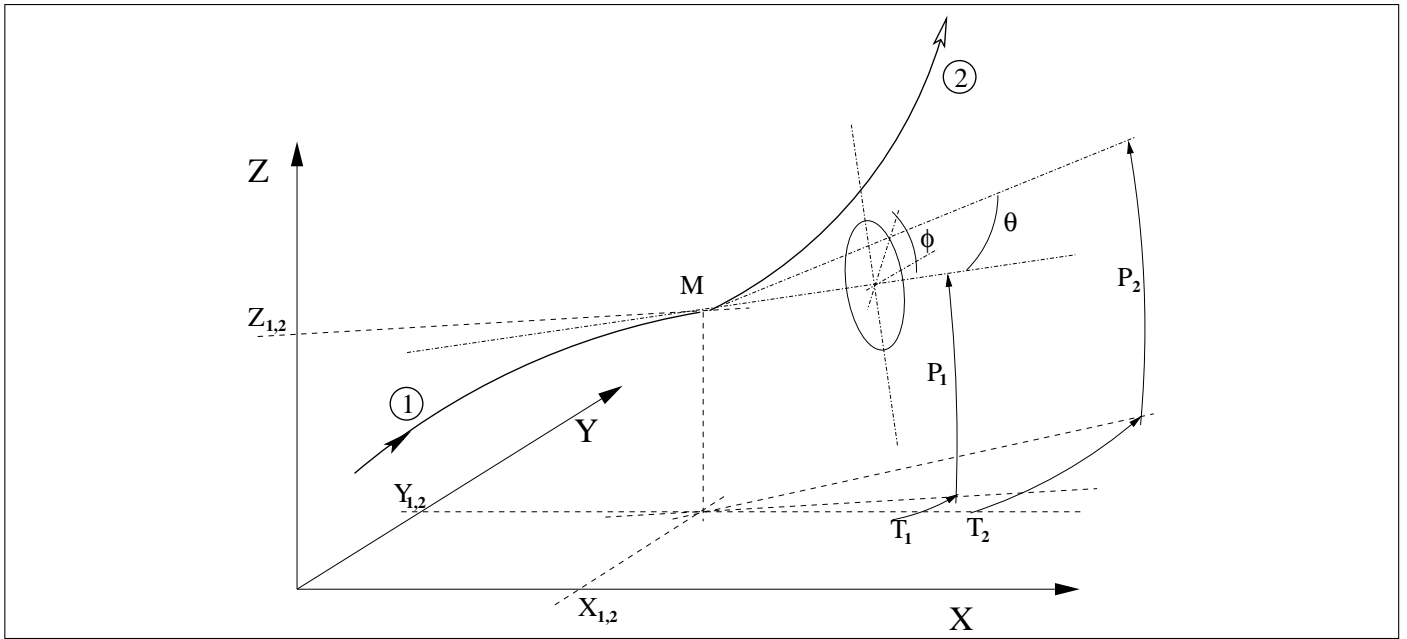


Figure 45: At position  $M(X_1, Y_1, Z_1)$ , particle 1 decays into 2 and 3 ; **zgoubi** then proceeds with the computation of the trajectory of 2, while 3 is discarded.  $\theta$  and  $\phi$  are the scattering angles of particle 2 relative to the direction of the incoming particle 1 ; they transform to  $T_2$  and  $P_2$  in **zgoubi** frame.

In *ESL* and *CHANGREF*, the current path length  $s = F(6, I)$  at the end of the element is compared to  $s_d$ . If the decay distance  $s_d$  happens to be comprised within the element then the particle coordinates at  $s_d$  are recomputed by linear translation from  $s$ .

The  $s_d(I = 1, IMAX)$  values are stored in the array  $FDES(6, I)$ . For further statistical purposes (*e.g.*, use of *HISTO*) the daughter particle 2 is tagged with an 'S' standing for "secondary". When a particle decays, its coordinates  $D, Y, T, Z, P, s$ , time at the decay point are stored in  $FDES(J, I)$ ,  $J = 1, 7$ .

#### A note on negative drifts :

The use of negative drifts with *MCDESINT* is allowed. For instance, negative drifts may occur in an optical structure : for some of the particles when using *CHANGREF* (due for instance to the  $Z$ -axis rotation or a negative  $XCE$ ); or when using *DRIFT* with  $XL < 0$ . Provision has been made to take it into account during the *MCDESINT* procedure, as follows.

If, due to a negative drift, a secondary particle reaches back the decay location of its parent particle, then the parent particle is "resurrected" with its original coordinates at that location, the secondary particle is discarded, and ray-tracing resumes in a regular way for the parent particle which is again allowed to decay when  $s_d$  is reached. This procedure is made possible by prior storage of the coordinates of the parent particles (in array  $FDES(J, I)$ ) each time a decay occurs.

Negative steps ( $XPAS < 0$ ) in optical elements are not compatible with *MCDESINT*.

#### Backward option:

By default, a particle moving backward upon decay is stopped (its *IEX* flag (see Sec. 7.13 is set to -11). This can be reversed using the option `back=1`.

**OPTICS : Write out optical functions. Log to zgoubi.OPTICS.out**

*OPTICS* normally appears next to object definition, it normally works in conjunction with element label(s). *OPTICS* causes the transport and write out, in *zgoubi.res*, of the  $6 \times 6$  beam matrix, following options *KOPT* and '*label*', below.

IF *KOPT*=0 : Off

IF *KOPT*=1 : Will transport the optical functions with initial values as specified in *OBJET*, option *KOBJ*=5.1. Note that *OPTICS* assumes the first particle of the set of 13 as defined by *OBJET*, to be the reference particle for computation of the transport coefficients.

Note : In the case of a periodic optical sequence, the initial coordinate values in *OBJET*[*KOBJ*=5.1] may be the periodic ones (as obtained, for instance, from a first run using *MATRIX*[*IFOC*=11]). Using *TWISS* keyword instead may be considered in that case.

A second argument, '*label*', allows

- if *label* = *all* : printing out, into *zgoubi.res*, after all keywords of the *zgoubi.dat* optical sequence,
- otherwise, printing out at all keyword featuring *LABEL1*  $\equiv$  *label* as a first label (see section 7.7, page 215, regarding the labelling of keywords).

*PRINT* : An optional third argument, *IMP*=1, or equivalently the command '*PRINT*', will cause saving of the transported optical functions into file *zgoubi.OPTICS.out*.

About the source code :

The program *beamat* ensures the transport of the beam matrix along the *zgoubi.dat* sequence. *beamat* is called by *opticc*, itself called after each optical element, in the program *zgoubi*.

The starting beam matrix value is in the array *F1*, initialized by *OBJET*[*KOBJ*=5.1], program *obj5*. The array *F0* contains the running beam matrix.

The program *beaimp* prints to *zgoubi.res*, whereas *optimp* prints to *zgoubi.OPTICS.out*. Both are called from *opticc*.

**OPTIONS : Global or special options**

(page 314 in PART B)

*OPTIONS* allows various on/off switches and options<sup>15</sup>.

Available options :

- '.plt -1', '.plt 0', '.plt 1', '.plt 2' or '.plt 7'

Role : force IL value to respectively -1, 0, 1, 2 or 7 (*cf.* Sec. 7.6, p. 213) in all optical elements.

Interest : avoids changing IL value element by element, which can be a hurdle in long optical sequences.

- 'AGSQUADS ON'

Role : change winding currents in some AGS quadrupole families. The program `agsquf.f` is resorted to to achieve that.

- 'CONSTY ON' or 'CONSTY OFF'

Role : force to constant *Y* and *Z* coordinates during stepwise integration.

This allows for instance checking fields, or fabricating field maps from **zgoubi**'s analytical models of optical elements, or smoothing/changing the mesh density of an existing field map, as follows.

Concurrent use with  $IL = 1, 2, \text{ or } 7$  (*cf.* '.plt' option here, or Sec. 7.6, p. 213) will allow logging  $\vec{B}$  and/or  $\vec{E}$  field(s) (and their derivatives) experienced across optical elements, in `zgoubi.res`, `zgoubi.plt` or `zgoubi.impdev.out` respectively.

As a consequence,

- forcing *Y* and *Z* to constant values, using 'CONSTY ON', allows checking fields and derivatives at constant *Y* and *Z*, as determined by *OBJET*, across (individual or short series of) optical elements. This may be helpful when setting up input data in **zgoubi**. It can be for instance of great help in case of complex geometrical and field data inputs, as in *DIPOLES*, *FFAG*;

- furthermore, particle coordinates (which are evenly distributed in space as an effect of CONSTY) and field data can be read from `zgoubi.plt` (requiring  $IL = 2$ ) and re-written in a dedicated file with proper formatting for use as a field map file by *TOSCA* (or *CARTEMES*, or *POLARMES*).

When using CONSTY, initial coordinates are set in a regular manner using *OBJET*. The *Y* coordinate will stand for the radius value in the case of optical elements defined in polar coordinates such as *DIPOLES* and *TOCA[MOD ≥ 20]*. In the latter case, if the field map raytraced through is a measured field map, CONSTY may be used to smooth it, and/or create a new field map with a different mesh density. Note: to generate a field map with uniform mesh in the longitudinal direction, it is necessary to adjust the integration step size so that the last step falls exactly on the exit boundary of the optical element which is raytraced through.

- 'MONITOR IDLUNI'

Role : monitor **zgoubi** execution, a sort of debugging tool. Only one functionality installed so far: IDLUNI option monitors the IDLUNI counter.

- 'RAND\_SEED MCOBJET3' or 'RAND SRLOSS'

Two roles : (i) substitute the use of current date-and-time for initializing the random sequence, to the use of IR values specified in the input data file, and, if the random series has already been initialized (by some prior random process (such as *ERRORS*, *MCOBJET[KOBJ=3]*, *SPNTRK[KSO=5]*, etc) then (ii) preclude re-initialization of the series.

Interest of (ii) : if single particle jobs are launched in parallel on a CPU farm, then they will use different random series, for generation of respectively particle coordinates and stochastic emission of photons.

- 'WRITE ON' or 'WRITE OFF'

Role : inhibit write statements to `zgoubi.res`. WRITE options, either 'ON' or 'OFF', may be inserted anywhere along an optical sequence.

<sup>15</sup>For the Geek: more options can be added if desired, that takes place in `options.f`.



Interest : 'WRITE OFF' may allow substantial savings on CPU time and/or zgoubi.res memory volume. 'WRITE ON' sets back to normal (the 'normal' state of write outs may depend on the status of other commands, such as *REBELOTE*, *FIT[2]*, etc.).

**ORDRE : Taylor expansions order**

The position  $\vec{R}$  and velocity  $\vec{u}$  of a particle are obtained from Taylor expansions as described in eq. (1.2.4). By default, these expansions are up to the fifth order derivative of  $\vec{u}$ ,

$$\begin{aligned}\vec{R}_1 &\approx \vec{R}_0 + \vec{u}\Delta s + \dots + \vec{u}^{(5)} \frac{\Delta s^6}{6!} \\ \vec{u}_1 &\approx \vec{u} + \vec{u}'\Delta s + \dots + \vec{u}^{(5)} \frac{\Delta s^5}{5!}\end{aligned}$$

which corresponds to fourth order derivatives of fields  $\vec{B}$ , eq. (1.2.8). and of  $\vec{E}$ , eq. (1.2.13).

However, third or higher order derivatives of fields may be zero in some optical elements, for instance in a sharp edge quadrupole. Also, in several elements, no more than first and second order field derivatives are implemented in the code. One may also wish to save on computation time by limiting the time-consuming calculation of lengthy (while possibly ineffective in terms of accuracy) Taylor expansions.

In that spirit, the purpose of *ORDRE*, option  $IO = 2 - 5$ , is to allow for expansions to the  $\vec{u}^{(n)}$  term in eq. 1.2.4. Default functioning is  $IO = 4$ , stated in *FORTTRAN* file `block.f`.

Note the following :

As concerns the optical elements

*DECAPOLE, DODECAPO, EBMULT, ELMULT, MULTIPOL, OCTUPOLE,  
QUADRUPO, SEXTUPOL*

field derivatives (see eq. 1.2.10 p. 25, eq. 1.2.15 p. 26,) have been installed in the code according to  $\vec{u}^{(5)}$  Taylor development order ; it may not be as complete for other optical elements. In particular, in electric optical elements field derivatives (eq. 1.2.15) are usually provided to no more than second order, which justifies saving on computing time by means of *ORDRE*, so to avoid pushing Taylor expansions as high as  $\vec{u}^{(5)}$ .

**NOTE** : see also the option *IORDRE* in field map declarations (*DIPOLE-M, TOSCA*, etc.).

**PARTICUL : Particle characteristics**

(page 317 in PART B)

Since **zgoubi** works using the rigidity, (“*BORO*”, as declared in *[MC]OBJET*), the keyword *PARTICUL* is in general not necessary.

However, it needs be introduced (normally, following *[MC]OBJET* in the input data file *zgoubi.dat*) when the definition of some characteristics of the particles (such as mass, charge, gyromagnetic factor, life-time in the center of mass) is needed, as is the case when using the following procedures :

<i>CAVITE</i>	: mass, charge
<i>MCDESINT</i>	: mass, COM life-time
<i>SPNTRK</i>	: mass, gyromagnetic factor
<i>SRLOSS</i>	: mass, charge
<i>SYNRAD</i>	: mass, charge
<i>Electric and Electro-Magnetic elements</i>	: mass, charge

The declaration of *PARTICUL* must **precede** these keywords.

If *PARTICUL* is omitted, which is in general the case when ray-tracing ions in purely magnetic optical assemblies, then **zgoubi** only knows the particle rigidity and would thus skip the computation of such quantities as time of flight, spin precession, synchrotron radiation, etc.

The expected attributes under *PARTICUL* are  
mass, charge, gyromagnetic factor, life-time

On the other hand, particle name can be substituted to that parameter list. Known particles are:

ELECTRON  
POSITRON  
MUON+  
MUON-  
PION+  
PION-  
PROTON  
HELION  
DEUTERON

Actual particle parameters are found listed in *zgoubi.res* when using *PARTICUL*. They may also be found in *rpart.i.f.*

**REBELOTE : 'Do it again'**

*REBELOTE* command operates as a do-loop, it has to be placed at the bottom of the loop in question, in the input data sequence; the program execution pointer will quietly continue beyond the command once it is completed.

When *REBELOTE* is encountered during execution, the program execution pointer jumps,

- either back to the beginning of the data file - the default behavior,
- or, if option  $K=99.1$  or  $K=99.2$ , back to a particular *LABEL1*, located upstream.

Then *NPASS-1* passes (from *LABEL1* to *REBELOTE*) follow.

As to the last pass, number *NPASS+1*, there are two possibilities :

- either it also encompasses the whole *LABEL1* to *REBELOTE* range,
- or, upon request (option  $K=99.2$ ), execution may exit that final pass upstream of *REBELOTE*, at a location defined by a second dedicated *LABEL2* placed between the first aforementioned *LABEL1*, and *REBELOTE*. In both cases, following the end of this “multiple-pass” procedure, the execution continues from the keyword which follows *REBELOTE*, until 'END' is encountered.

Amongst other, these procedures allow for instance raytracing through the following sequence:  
injection line + multiturn in a ring + extraction line.

The three functionalities of *REBELOTE* are the following :

1. **Beam lines:** *REBELOTE* can be used for Monte Carlo simulations in beam lines (in which case  $K = 0$ ), when more than the maximum allowed  $\text{Max}(IMAX)$  particles<sup>16</sup> are to be tracked. In that functioning mode, when procedures such as *HISTO*, *MCDESINT*, *MCOBJET*, *OBJETA*, *SPNTRK*[ $KSO = 5$ ] are used, their random seeds are not reset and independent statistics will add up pass after pass.

This allows for instance high statistics Monte Carlo simulations. *NPASS* runs through the optical sequence, from *MCOBJET* to *REBELOTE* will result from the latter, resulting in the calculation of  $(1 + NPASS) \times IMAX$  trajectories, with as many random sets of initial coordinates.

2. **Multiturn tracking:** *REBELOTE* can be used for multi-turn ray-tracing in circular machines ; normally  $K = 99$  in that case (or  $K = 99.2$ , see below). *NPASS* turns in the optical sequence will follow, resulting in the tracking of *IMAX* particles over  $1 + NPASS$  turns. For the simulation of pulsed power supplies, synchrotron motion, and other Q-jump manipulation, see *SCALING*.

For instance, using option described  $K=99.2$  above, a full “injection line + ring + extraction line” installation can be simulated - kicker firing and other magnet ramping can be simulated using *SCALING*.

Using the double-*LABEL1-2* method discussed above with option  $K=99.2$ , it is possible to encompass the ring between an injection line section (namely, with the element sequence of the latter extending from *OBJET* to the first *LABEL1*), and an extraction line (its description will then follow *REBELOTE*), whereas the ring description extends from to the first *LABEL1* down to *REBELOTE*, with possible extraction, at the last pass, at the location of the second *LABEL2*, located anywhere between the *LABEL1* and *REBELOTE*,

<sup>16</sup> $\text{Max}(IMAX)$  here stands for the maximum number of particles allowed in one go. The reason for a limitation is the dimensioning of arrays in the source code:  $\text{Max}(IMAX)$  is controlled, and can be changed, in the include file [pathTo]/include/MAXTRA.H.

Note : Some *CAVITE* options cause a reset-to-zero of individual particle path length<sup>17</sup>. This is for reasons of cumulated path length accuracy (a delta-path below computer accuracy compared to total path length would not be resolved - it can be for instance bunch length as compared to cumulated multi-turn distance around a large ring). This reset may not be desirable, it depends on the multi-turn problem dealt with using *REBELOTE*, it may be necessary for instance in long-term tracking in large rings, it is not in recirculating linacs. Option *K=99* in *REBELOTE* will cause this reset, whereas option *K=98* (UNDER DEVELOPMENT, In *cavite.f*, IN THIS VERSION OF THE CODE) will avoid it everything else left unchanged.

3. **Case *IOPT=1***: *REBELOTE* can in that case change the value of any parameter in *zgoubi.dat* optical sequence (see page 326). *NPRM* tells the number of parameters to be changed. *NPRM* lines follow. Each line specifies the respective *NRBLT* values which the parameter concerned is to be assigned, over the *NRBLT* passes of the *REBELOTE* process. The parameter values are specified either

- in the form of a list of *NRBLT* numerical values,
- or in the form ***first\_value:last\_value***.

*IOPT=1* is under development. *TO BE DOCUMENT FURTHER*.

### **Printouts**

Printouts over *NPASS+1* passes might result in a prohibitively big *zgoubi.res* file. They may be switched on/off by means of the option *KWRIT=i,j*, with *i = 1/0* respectively. The *j* flag allows a printout of the pass number and of some additional information to the video output, every  $10^{j-1}$  turns if  $j > 0$  ; video output is essentially switched off during *REBELOTE* execution if  $j = 0$ .

*REBELOTE* also provides information : statistical calculations and related data regarding particle decay (*MCDESINT*), spin tracking (*SPNTRK*), stopped particles (*CHAMBR*, *COLLIMA*), etc.

### **COMBINING REBELOTE AND FIT[2]**

The keyword *REBELOTE* can follow *FIT[2]*. This allows executing again the fit procedure, after changing the value of some parameter(s) in *zgoubi.dat* using *REBELOTE* with option *IOPT=1*. That's the interest of the game : *REBELOTE* changes that (these) parameter(s), and then sends the *zgoubi* execution pointer back to the top of *zgoubi.dat* for a new *FIT[2]* run. Note that parameters in *FIT[2]* itself can be changed, if a parameter scan so requires for instance.

Conversely, *REBELOTE* may be followed by the keyword *FIT[2]*. This allows for instance optimization of constraints that require multi-turn tracking (e.g., optimized tune-jump quadrupole settings for beam acceleration through depolarizing resonances).

### **EXAMPLES**

- An example of the use of *REBELOTE* loop including data changes is given page 327.
- An example of *FIT* embedded in a *REBELOTE* loop is given page 285.

### **RECOVERING FROM A CRASH**

When accelerating a bunch, *REBELOTE* is used to ensure the multi-turn process. Now, if the job is stopped - due to computer crash or else - it is possible to resume the tracking from the latest records (in *[b\_]zboubi.fai* storage file), using *OBJET*, *KOBJ= 3*. *CAVITE* and *SCALING* aspects of the data file modifications necessary to ensure the recovery, in the case of pulsed magnets, are discussed respectively in the *SCALING* section, pp. 183, 186,

**EXAMPLE** - An example of job resuming is given page 310.

<sup>17</sup>For the geek: the "f(6,I)=0.d0" statements in *cavite.f*.

**RESET : Reset counters and flags**

Resets counters involved in *CHAMBR*, *COLLIMA*, *HISTO* and *INTEG* procedures.

Switches off *CHAMBR*, *MCDESINT*, *SCALING* and *SPNTRK* options.

**SCALING : Power supplies and R.F. function generator**

(page 329 in PART B)

*SCALING* acts as a function generator dedicated to

- varying fields in optical elements, potentials in electrostatic devices, possibly in correlation with the reference rigidity  $B\rho_{ref}$  (Sec. 7.2) for instance by *CAVITE*.

- varying one or more parameters (so, not necessarily field values), in an arbitrary optical element. This capabilities is developed *on demand*, thus at present available in a limited number of elements (namely, *CHANGREF*, *SPINR*, *AGSMM*, *AGSQUAD*, a couple more may be...). *TO BE DOCUMENTED FURTHER*.

*SCALING* is normally intended to be declared after the object definition. Nothing precludes declaring it later though, it will be taken into account from the moment that it is met within *zgoubi.dat* optical sequence.

Used in conjunction with *REBELOTE* for the simulation of multi-turn tracking, *SCALING* allows to change fields turn-by-turn in an arbitrary, user defined manner during an acceleration multi-turn. Fields (for instance main magnets fields) may instead be forced to follow a rigidity change caused by *CAVITE*, this can be obtained by simply using *NT=-1* as the first argument in *SCALING*, see below.

The latter case (*i.e.*, fields being correlated to *CAVITE*), allows the simulation of a recirculating linac (thus optical elements between cavities are scaled, following the rigidity increase cavity by cavity), or multiple RF stations in a ring.

*SCALING* acts on families of elements, a family being designated by its name that coincides with the keyword of the corresponding element. For instance, declaring *MULTIPOL* as to be scaled will result in the same timing law being applied to all *MULTIPOLs* in *zgoubi.dat* optical sequence. Subsets can be selected by labeling keywords in the data file (section 7.7, page 215) and adding the corresponding *LABEL(s)* in the *SCALING* declarations (9 *LABELs* maximum - for the geek: this can be changed in *MXFS.H*, parameter *MLF*). The family name of concern, as well as the scaling function for that family, are given as input data to the keyword *SCALING*. There is an upper limit to the number *NF* of families that can be declared as subject to a scaling law (this can be changed in the *FORTTRAN* include file *MXFS.H*, parameter *MXF*).

A scaling law can be comprised of up to *NT* successive timings. *NT* is thus normally a positive integer, its value tells the number of timings used in simulating a time dependent power supply or RF function. The scaling factor at all turns is obtained by linear interpolation from the so declared timing series, see below. However, *NT* can also be a real and/or take negative values, being in this case an option switch, see below.

If a family appears more than once in the *SCALING* list (*i.e.*, identical keyword and label list), the following will occur :

- if each instance addresses a different parameter in the optical elements concerned, both scaling rules are applied,
- if two (or more) instances address the same parameter in the optical elements, the last instance will prevail.

If a family addressed is a subset of a previously declared one, the *SCALING* rule will override the previous one, for that subset only.

The following illustrates typical data formatting for the simulation of an acceleration cycle in a circular accelerator:

**SCALING**

1	5		Active. $NF = 5$ families of elements are concerned, as listed below :
		<b>QUADRUPO QFA QFB</b>	- <b>QUADRUPO</b> keywords labeled 'QFA' and Quadrupoles labeled 'QFB'
2			$NT = 2$ timings
18131.E-3	24176.E-3		The field increases (linearly) from 18131E-3* $B_0$ to 24176E-3* $B_0$
1	6379		from turn 1 to turn 6379
		<b>MULTIPOL</b>	- General rule. Precedes labeled cases if any.
2			$NT = 2$ timings
18131.E-3	24176.E-3		Fields increase from 18131E-3* $B_i$ to 24176E-3* $B_i$ ( $\forall i = 1, 10$ poles)
1	6379		from turn 1 to turn 6379
		<b>MULTIPOL QDA QDB</b>	- <b>MULTIPOL</b> keywords labeled 'QDA' and Multipoles labeled 'QDB'
2			$NT = 2$ timings
18131.E-3	24176.E-3		Fields increase from 18131E-3* $B_i$ to 24176E-3* $B_i$ ( $\forall i = 1, 10$ poles)
1	6379		from turn 1 to turn 6379
		<b>BEND</b>	- All <b>BEND</b> keywords, regardless of any <b>LABEL</b>
2			$NT = 2$ timings
18131.E-3	24176.E-3		As above
1	6379		
		<b>CAVITE</b>	- Accelerating cavity
3			$NT = 3$ timings
1	1.22	1.33352	The synchronous rigidity $(B\rho)_s$ increases linearly,
1	1200	6379	from $(B\rho)_{s_0}$ to $1.22 * (B\rho)_{s_0}$ from turn 1 to 1200, and from $1.22 * (B\rho)_{s_0}$ to $1.33352 (B\rho)_{s_0}$ from turn 1200 to 6379

The timing is in unit of turns. In this example, **TIMING** = 1 to 6379 (turns). Therefore, at turn number  $N$ ,  $B$  and  $B_i$  are updated in the following way. Let  $SCALE(TIMING = N)$  be the updating scale factor

$$SCALE(N) = 18.131 \frac{24.176 - 18.131}{1 + 6379 - 1} (N - 1)$$

and then

$$B(N) = SCALE(N)B_0$$

$$B_i(N) = SCALE(N)B_{i0}$$

The RF frequency is computed using

$$f_{RF} = \frac{hc}{\mathcal{L}} \frac{q(B\rho)_s}{(q^2(B\rho)_s^2 + (Mc^2)^2)^{1/2}}$$

where the rigidity is updated in the following way. Let  $(B\rho)_{s_0}$  be the initial rigidity (namely,  $(B\rho)_{s_0} = BORO$  as defined in the keyword **OBJET** for instance). Then, at turn number  $N$ ,

$$\text{if } 1 \leq N \leq 1200 \text{ then, } SCALE(N) = 1 + \frac{1.22 - 1}{1 + 1200 - 1} (N - 1)$$

$$\text{if } 1200 \leq N \leq 6379 \text{ then, } SCALE(N) = 1.22 + \frac{1.33352 - 1.22}{1 + 6379 - 1200} (N - 1200)$$

and then,

$$(B\rho)_s(N) = SCALE(N) \cdot (B\rho)_{s_0}$$

from which value the calculations of  $f_{RF}(N)$  follow.

Beyond the regular  $NT \geq 1$  (or the short-cut  $NT = -1$ ), there are coded "values" for  $NT$ , as follows.

Case of  $NT.Int$  :  $NT=1$  and  $Int$  is an integer. This is for special scaling laws and/or reading from external files, as follows:

- $NT = 1.10$  : allows taking the scaling law from an external data file, as in the following example :



<i>MULTIPOL COH1</i>	Scaling applies to all multipole field components, in <i>MULTIPOL</i> keywords having 'COH1' as first label.
1.10 [ <i>FAC</i> ]	Scaling is read from a file, file name e.g. 'scalingFile'; <i>FAC</i> is an additional scaling factor.
scalingFile	File name (file is expected to contain at least N2 columns).
1 n	Column number n lists the scaling factor at rigidity values listed in column number 1; current scaling is obtained by interpolation in this table (and multiplied by <i>FAC</i> if specified).

- *NT* = 1.11 : *TO BE DOCUMENTED FURTHER.*

*someKEYWORD* [*someLABEL1*]  
1.11

- *NT* = 1.12 : allows taking the scaling law from an external data file, as in the following example :

<i>AGSQUAD QH.A17</i>	Scaling applies to <i>AGSQUAD</i> keyword having 'COH1' as first label.
1.12	Scaling is read from a file, file name e.g. 'scalingFile'.
scalingFile	File name (file is expected to contain at least 2+2*N2 columns).
N1, N2, p11, p12, p21, p22, ...	N1=number of pairs, see next; N2= ; Scaling is obtained from interpolation in column pi1 and applied to parameter pi2 of <i>AGSQUAD</i> .

- *NT* = 1.13 : *TO BE DOCUMENTED FURTHER.*

*someKEYWORD* [*someLABEL1*]  
1.13

*NT* can take negative values, then acting as a switch for various types of hard-coded function generators, as follows :

- *NT* = -1 : this is convenient for synchrotron acceleration. In this case the next two lines both contain a single data (as for *NT* = 1), respectively the starting scaling factor value, and 1. The current field scaling factor will then be updated from the energy kick by the cavity if for instance *CAVITE/IOPT=2* is used, namely,

$$\text{SCALE}(N) = \text{SCALE}(N - 1) * \frac{B\rho(N)}{B\rho(N - 1)}$$

- *NT* = -2 : allows reading an RF law (voltage, frequency, etc.) for *CAVITE* from an external ascii file (with default name *zgoubi.freqLaw.In*). Useful for acceleration in fixed field rings for instance. The formatting of the ascii data is in six columns, as follows (the numerical data are just for illustration):

#	1	2	3	4	5	6
#	turn#,	tauCell/micro-second,	phi/rad,	oclock/micro-second,	Ekin/MeV,	one
1	5.07794047E-02	0.00000000E+00	6.09352856E-01	1.20000000E+01	1	
4	5.07064127E-02	1.88495559E+01	2.43571260E+00	1.20410424E+01	1	
7	5.06336342E-02	3.76991118E+01	4.25952433E+00	1.20820848E+01	1	
1	5.05610722E-02	5.65486678E+01	6.08078864E+00	1.21231273E+01	1	

etc.

The column content is the following: Col. 1: turn number: the increment does not have to be 1, **zgoubi** will interpolate during acceleration. Col. 2: synchronous time of flight,  $T_{\text{rev}}$ . Col. 3: RF phase,  $\omega_{\text{rf}}T_{\text{rev}}$ . Col. 4: cumulated time. Col. 5: synchronous energy. Column 6 is unused.

- *NT* = -60 : field scaling follows  $B\rho$  change by *CAVITE*, accounting for some starting value.
- *NT* = -77 : field scaling law for FNAL proton driver - to be documented.
- *NT* = -87 : AGS tune-jump quadrupoles - to be documented.
- *NT* = -88 : AC dipole function generator. An AC dipole provides a bending field, thus for a magnetic dipole *MULTIPOL* can be used for instance, or *BEND*. Four values determine the functioning of this optical element : phase of the AC dipole; AC dipole tune Q1 at the start of the sweep; AC dipole tune Q2 at the end of the sweep; and a scale factor to be applied to the field.

Note that in the case of the previous example (assuming, though, that the ramping of the magnetic fields strictly follows the *CAVITE* kick, which was not strictly the case), the *SCALING* argument list can be simplified, including by suppressing 'CAVITE' from the list, as follows :

<i>SCALING</i>	- Scaling
1 3	Active. $NF = 3$ families of elements are concerned, as listed below
<i>QUADRUPO QFA QFB</i>	- Quadrupoles labeled 'QFA' and Quadrupoles labeled 'QFB'
-1	
18131.E-3	
1	
<i>MULTIPOL QDA QDB</i>	- Multipoles labeled 'QDA' and Multipoles labeled 'QDB'
-1	
18131.E-3	
1	
<i>BEND</i>	- All <i>BEND</i> s (regardless of any <i>LABEL</i> )
-1	
18131.E-3	
1	

**Notes :**

1. In causing, via *CAVITE*, a change of the synchronous rigidity, *SCALING* causes a change of the reference rigidity (Sec. 7.2), following (see *CAVITE*, p. 84)

$$B\rho_{ref} = BORO \longrightarrow B\rho_{ref} = BORO + \delta B\rho_s$$

2. It may happen that some optical elements won't scale, for source code development or updating reasons. This should be paid attention to by preliminary simulation tests.

**SCALING AND THE FIT PROCEDURE****Coupling variables**

Because it applies a scaling factor to families of optical elements (as determined by their keyword and possible *LABEL1*, *LABEL2*), *SCALING* can be used to couple FIT variables.

**Tunes and chromaticities**

*FIT* can vary any numerical data under *SCALING*, this allows scaling fields in optical element families (by specifying one or more *LABEL1*s to address one or more families as described above). Examples are given in Tab. 2 (p. 167).

**RECOVERING FROM A CRASH**

When accelerating a bunch in a pulsed ring, *SCALING* is used to ramp magnetic fields, so following the rigidity increase by *CAVITE*. *CAVITE* may or may not be part of the *SCALING* argument list, see the two previous examples. Now, if the job stops during the multiturn tracking - due to computer crash or else - it is possible to resume the tracking from the latest records (in [b\_]zboubi.fai storage file), using *OBJET*, *KOBJ= 3*. In the presence of acceleration using *CAVITE*, since there is no reference particle in **zgoubi**, *CAVITE* will read from [b\_]zboubi.fai the reference rigidity at the corresponding time/turn when tracking crashed/resumes. Note that, as far as *CAVITE* is concerned, this recovery process is only installed for *CAVITE/Option=2*, a option often used for synchronous acceleration (recovery for additional *CAVITE* options can be added in `cavite.f` program).

**SPACECHARG : Space charge correction kick**

(page 334 in PART B)

## UNDER DEVELOPMENT

The keyword *SPACECHARG* enables (or disables) the calculation of space charge effects (see Sec. 4). It must be preceded by *PARTICUL* for definition of mass and charge values, as they enter in the definition of space charge parameters.

*SPACECHARG* provides two models for the simulation of space charge effects: KV (space charge is a linear effect on motion) and Gaussian (non-linear effect).

Optical elements for which space charge has to be accounted for are specified as *SPACECHARG* user's input data. It can be all elements of the optical sequence, or some of them specified by their keyword. The Fortran has provision to account for LABEL1-2s; at the time of these writings this requires further installation.

*SPACECHARG* is expected to appear a first time at the location where space charge effects should start being taken into account, with the first data *LMNT* set to 'all' or as a list of up to 10 keywords.

*SPACECHARG* is expected to appear a second time at the location where corrections should stop, with *LMNT* set to 'none'. It results in summary outputs in *zgoubi.res*.

Occurrence of *PRINT* in *SPACECHARG* argument results in storage of computational data and other relevant informations in *zgoubi.SPACECHARG.out*, as space charge computation proceeds.

The way the effect is implemented in **zgoubi**: a space charge correction is applied (in *integr.f*) on *T* and *P* particle angle at each integration step across the optical elements concerned. These correction kicks are prepared prior to entering the optical element (by *spachk*, an entry in *spach.f*, called in *zgoubi.f*).

*TO BE DOCUMENTED FURTHER.*

**SPNTRK : Spin tracking switch, and spin coordinate initialization**

(page 338 in PART B)

The keyword *SPNTRK* allows switching spin tracking on (index  $KSO=1$ ) or off ( $KSO=0$ ), or resuming (index  $KSO=-1$ , following an occurrence  $KSO=0$ ).

*SPNTRK* will also attribute initial spin coordinates to each one of the *IMAX* particles defined by *[MC]OBJET* (unless *OBJET[KOBJ=3]* is used, in which case the latter takes care of spin coordinates initialization). Initial spin coordinates depend on the option index  $KSO$ .

*SPNTRK* It must be preceded by *PARTICUL* for the definition of mass and gyromagnetic factor.

**KSO = 1 (respectively 2, 3) :** the *IMAX* particles defined with *[MC]OBJET* are given a longitudinal (1,0,0) spin component (respectively transverse horizontal (0,1,0), vertical (0,0,1)).

**KSO = 4 :** initial spin components are entered explicitly for each one of the *IMAX* particles of the beam.

**KSO = 4.1 :** three initial spin components  $S_X$ ,  $S_Y$ ,  $S_Z$  are entered explicitly just once, they are then assigned to each one of the *IMAX* particles of the beam.

**KSO = 4.2 :** *under development*. Read spin components from a file, file name has to be provided. (that file at present can be *zgoubi.res* :  $KSO=4.2$  looks for key “(deg) (deg)“ that appears at start of the spin coordinates list under the keyword *SPNPRT*).

**KSO = 5 :** *under development* - to be completed in *spn.f*. Random generation of *IMAX* initial spin conditions as sketched in Fig. 46. Given a mean polarization axis (S) defined by its angles  $T_0$  and  $P_0$ , and a cone of angle  $A$  with respect to this axis, the *IMAX* spins are sorted randomly in a Gaussian distribution

$$p(a) = \exp \left[ -\frac{(A - a)^2}{2\delta A^2} \right] / \delta A \sqrt{2\pi}$$

and within a cylindrical uniform distribution around the (S) axis. Examples of simple distributions available by this mean are given in Fig. 47.

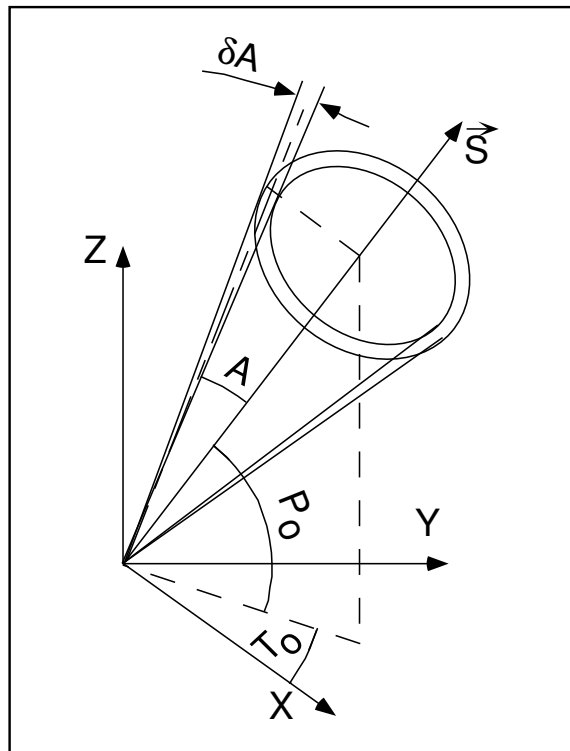


Figure 46: Spin distribution as obtained with option  $KSO = 5$ .  
The spins are distributed within an annular strip  $\delta A$  (standard deviation) at an angle  $A$  with respect to the axis of mean polarization ( $\vec{S}$ ) defined by  $T_0$  and  $P_0$ .

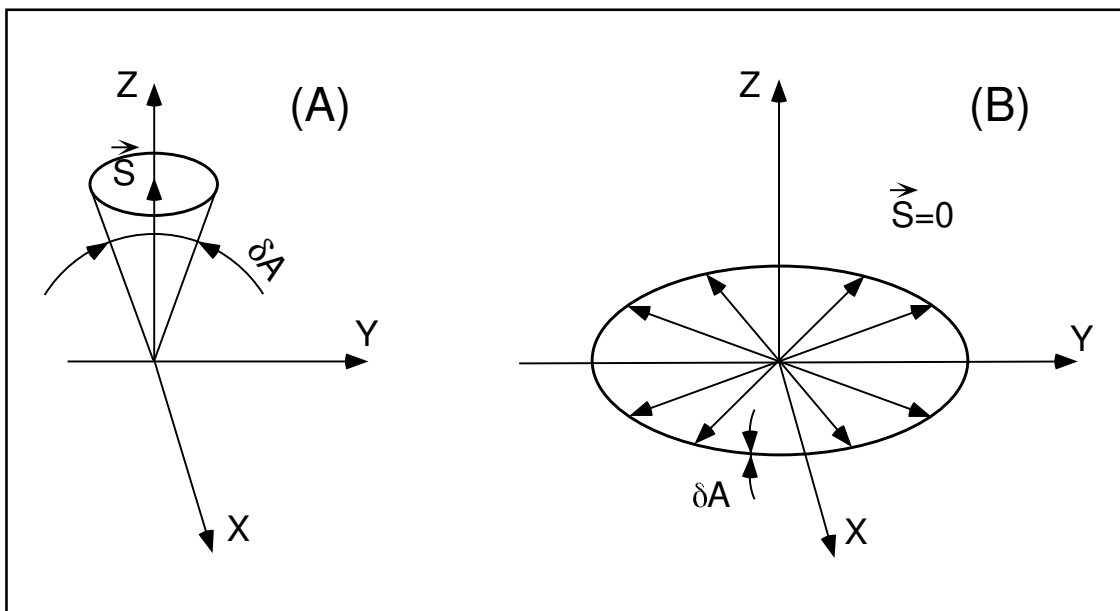


Figure 47: Examples of the use of  $KSO = 5$ .

A : Gaussian distribution around a mean vertical polarization axis, obtained with  $T_0 = \text{arbitrary}$ ,  $P_0 = \pi/2$ ,  $A = 0$  and  $\delta A \neq 0$ .

B : Isotropic distribution in the median plane, obtained with  $P_0 = \pm\pi/2$ ,  $A = \pi/2$ , and  $\delta A = 0$ .

**SRLOSS : Synchrotron radiation energy loss**

(page 339 in PART B)

The keyword *SRLOSS* allows activating or stopping (option *KSR* = 1, 0 respectively) stepwise tracking of energy loss by stochastic emission of photons in magnetic fields, following the method described in section 3.1[27].

It can be chosen to allow radiation in the sole dipole fields, or in all types of fields regardless of their multipole composition. It can also be chosen to allow for the radiation induced transverse kick (this is yet to be installed in the code - not done as it is generally considered a negligible effect).

*SRLOSS* must be preceded by *PARTICUL* for defining mass and charge values as they enter in the definition of SR parameters.

Statistics on SR parameters are computed and updated while tracking, the results of which can be obtained by means of the keyword *SRPRNT*.

**SVDOC : SVD orbit correction**

(page 341 in PART B)

**UNDER DEVELOPMENT - Feb. 2019**

*SVDOC* command operates as a do-loop, it has to be placed at the bottom of the loop in question, in the input data sequence; the program execution pointer will quietly continue beyond the command once it is completed.

ERRORS or not ERRORS? Two cases:

- the optical sequence already has an defect orbit included (quad positioning, field defects, or whatever else), then ERRORS is not necessary. Yet nothing precludes using it to generate errors on top of the existing defect orbit, see below;

- the optical has no orbit errors. Then ERRORS can be used to inject random ones. It will do so before the SVD correction process is launched. Details below.

In the case on a periodic structure, *SVDOC* can be preceded by *FIT[2]*. *FIT[2]* will find the closed orbit before launching the *SVDOC* procedure, and after completion of the latter (it is then expected to be zero if *SVDOC* went well).

The way *SVDOC* operates is the following :

Step 1. Define/determine the reference trajectory (case of a beam line) or closed orbit (case of a periodic optical sequence) : the initial coordinate(s) in *OBJET* are considered the reference(s) - multiple references are allowed, see below. A reference does not have to be zero, however *FIT* can be used to find the reference orbit initial coordinates, in that case *FIT* should appear before *SVDOC* in the sequence. For instance,

- (i) in a spectrometer or in a beam line, *FIT[IC=3]* would allow to determine the necessary initial coordinates (*FIT* variables) for a particle with a given momentum to acquire some requested radial excursion at some requested location down the line (*FIT* constraint(s));

- (ii) if the sequence is periodic, *FIT[IC=3.1]* will find the closed orbit (possibly multiple closed orbits, e.g. in the case of a multiple-pass recirculating linac, or a fixed-field ring such as a cyclotron or FFAG). A practical example of periodic orbit finding using *FIT[2]* is given page 352.

Step 2. The algorithm computes the *A* matrix (in  $A\vec{x} = \vec{y} - \vec{y}_0$ , with  $\vec{x}$  the correctors signal,  $\vec{y}$  the pickups signal,  $\vec{y}_0$  a reference PU signal - the reference orbit), by injecting a non-zero signal (angle) in each orbit corrector, one after the other successively, and saving the successive PU readings (as rows of the *A* matrix). There can be up to 20 different particles tracked (meaning 20 different reference orbits, and 20 times more columns (PUs) in *A*), each PU is “multiple-beam”, capable of discriminating each orbit in the SVD process (case for instance of chromatic orbits in a synchrotron, or multiple momenta in a multiple-pass ERL, or multiple orbits in a scaling FFAG lattice).

Step 3. The inverse matrix  $A^{-1}$  is computed and logged to *zgoubi.SVD\_Amatrix.out*, together with the *A* matrix and the result of an  $A^{-1} \times A$  sanity test, expected to be the identity matrix.

Step 4. *SVDOC* applies the correction. A last pass through the sequence delivers the corrected orbit. The latter is logged in *zgoubi.SVDOrbits.out*.

A version of the initial data file with corrected orbit, i.e. including existing initial orbit or ERRORS defects and with correctors updated, is saved in *zgoubi.SVD.out.dat*.

Step 5. Optionally, *SVDOC* can carry on with error trials (for statistics purposes for instance), in the following way:

- (i) *SVDOC* applies orbit defects,

- (ii) then applies the compensation signal (determined from the SVD-inverted  $A^{-1}$  matrix) on the correctors,

- (iii) finds the corrected [multiple-]orbit (expected near-zero; logged to *zgoubi.SVDOrbit.out*),

- (iv) repeats (i)-(iv).

The method is the following for instance in the case of a periodic sequence:

- introduce *ERRORS* in the optical sequence, it will be used by *SVDOC* to activate orbit defects. Note that *ERRORS* must be set to OFF in the *zgoubi* sequence, as 1/ the first pass through the sequence is used to determine the  $\vec{y}_0$  reference orbit mentioned earlier (using *FIT[2]*, placed at the end of the sequence), 2/ the ON/OFF switch in *ERRORS* is managed by *SVDOC* in this procedure;

- *SVDOC* resets to zero the strength in the correctors that the SVD used;
- *FIT[2]* finds the closed orbit resulting from the defects generated by *ERRORS*; this orbit is recorded at the PUs that the SVD is told to use;
- *SVDOC* then sets the strengths in the correctors, using the SVD matrix;
- *FIT[2]* finds the new orbit - presumably improved ! and delivers it (logged to *zgoubi.SVDObits.out*, together with the uncorrected orbit).

A recap of the output files produced by [*SVDOC*]:

file	comment
<i>zgoubi.SVD_Amatrix.out</i>	contains $A^{-1}$ and $A$ matrices, as computed by <i>SVDOC</i>
<i>zgoubi.SVD_PuCoList.out</i>	
<i>zgoubi.SVD.out.dat</i>	a copy of <i>zgoubi.dat</i> sequence with updated corrector values as found by <i>SVDOC</i> . Default name, can be changed
<i>zgoubi.SVDObits.out</i>	Pickup records, and corrector strengths

*EXAMPLE* - An example of the set up and use of *SVDOC* is given in PART. B, p. 342.



**SYNRAD : Synchrotron radiation spectral-angular densities**

The keyword *SYNRAD* enables (or disables) the calculation of synchrotron radiation (SR) electric field and spectral angular energy density. It must be preceded by *PARTICUL* for defining mass and charge values, as they enter in the definition of SR parameters.

*SYNRAD* is supposed to appear a first time at the location where SR should start being taken into account, with the first data *KSR* set to 1. It results in on-line storage of the electric field vector and other relevant quantities in *zgoubi.sre*, as step by step integration proceeds. The observer position (*XO*, *YO*, *ZO*) is specified next to *KSR*.

Data stored in *zgoubi.sre* :

(*ELx*, *ELy*, *ELz*) : electric field vector  $\vec{\mathcal{E}}$  (eq. 3.2.1)

(*btx*, *bty*, *btz*) =  $\vec{\beta} = \frac{1}{c} \times$  particle velocity

(*gx*, *gy*, *gz*) =  $\frac{d\vec{\beta}}{dt}$  = particle acceleration (eq. 3.2.3)

$\Delta\tau$  = observer time increment (eq. 3.2.2)

$t' = \tau - r(t')/c$  = retarded (particle) time

(*rtx*, *rty*, *rtz*) :  $\vec{R}(t)$ , particle to observer vector (eq. 3.2.4)

(*x*, *y*, *z*) = particle coordinates

$\Delta s$  = step size in the magnet (fig. 2)

*NS* = step number

*I* = particle number

*LET(I)* = tagging letter

*IEX(I)* = stop flag (see section 7.13)

*SYNRAD* is supposed to appear a second time at the location where SR calculations should stop, with *KSR* set to 2. It results in the output of the angular energy density  $\int_{\nu_1}^{\nu_2} \partial^3 W / \partial \phi \partial \psi \partial \nu$  (eq. 3.2.11) as calculated from the Fourier transform of the electric field (eq. 3.2.11). The spectral range of interest and frequency sampling ( $\nu_1$ ,  $\nu_2$ , *N*) are specified next to *KSR*.

**SYSTEM : System call**

The keyword *SYSTEM* allows one or a series of system calls. It can appear anywhere, an arbitrary number of times, along the *zgoubi.dat* optical sequence. It is effective at the very location where it appears.

*SYSTEM* keyword is followed by the list of the desired system commands. That can be saving *zgoubi* output files, calling again **zgoubi** at the end of a run so allowing dependent consecutive jobs, etc.

*EXAMPLES* - Examples of the use of *SYSTEM* are given in pp. 285, 345.

**6.5 Output Procedures**

These procedures are dedicated to the storage or printing of particle coordinates, histograms, spin coordinates, etc. They may be called for at any spot in the data pile.

**FAISCEAU, FAISCNL, FAISTORE : Print/Store particle coordinates**

• *FAISCEAU* can be introduced anywhere along an optical sequence. It produces a print (into zgoubi.res) of initial and actual coordinates of the *IMAX* particles at the location where it stands, together tagging indices and letters, etc.

• *FAISCNL* is an older, less performing, version of *FAISTORE*, see below for the latter. It produces a lot more information on particles at current location, including spin components, decay distance, mass, charge, etc. (see list below), and stores it in a dedicated file *FNAME* (advised name is *FNAME* = 'zgoubi.fai' (formatted write) or 'b\_zgoubi.fai' (binary write) if post-processing with **zpop** should follow). This file may further on be read by means of *OBJET*, option *KOBJ*= 3, or used for other purposes such as graphics (see Part D of the Guide).

The data written to that file are formatted and ordered according to the *FORTRAN* sequence in the subroutine *impfai.f*, where details and possible updates are to found. The following is an instance :

```

OPEN (UNIT = NL, FILE = FNAME, STATUS = 'NEW')
IF (BINARY) THEN
  DO 2 I=1, IMAX
    P = BORO*CL9 *F(1, I) * AMQ(2, I)
    ENERG = SQRT(P*P + AMQ(1, I)*AMQ(1, I))
    ENEKI = ENERG - AMQ(1, I)
    WRITE (NFAI)
1    IEX(I), -1.D0+FO(1, I), (FO(J, I), J=2, MXJ),          tag, initial D-1, Y, T, Z, P, S, Time
2    -1.D0+F(1, I), F(2, I), F(3, I), (F(J, I), J=4, MXJ),  current D_i, Y, T, Z, P, S, Time
>    (SI(J, I), J=1, 4), (SF(J, I), J=1, 4),                spin components, initial and current
>    ENEKI, ENERG,                                          energy
4    I, IREP(I), SORT(I), (AMQ(J, I), J=1, 5), RET(I), DPR(I), PS, particle#, loss S, mass, Q, G, life time
5    BORO, IPASS, NOEL, KLEY, LBL1, LBL2, LET(I)           BORO, pass#, element#, keyword, labels
2  CONTINUE
ELSE
  DO 1 I=1, IMAX
    P = BORO*CL9 *F(1, I) * AMQ(2, I)
    ENERG = SQRT(P*P + AMQ(1, I)*AMQ(1, I))
    ENEKI = ENERG - AMQ(1, I)
    WRITE (NFAI, 110)
1    IEX(I), -1.D0+FO(1, I), (FO(J, I), J=2, MXJ),
2    -1.D0+F(1, I), F(2, I), F(3, I),
3    (F(J, I), J=4, MXJ),
4    (SI(J, I), J=1, 4), (SF(J, I), J=1, 4),
5    ENEKI, ENERG,
6    I, IREP(I), SORT(I), (AMQ(J, I), J=1, 5), RET(I), DPR(I), PS,
7    BORO, IPASS, NOEL, KLEY, LBL1, LBL2, LET(I)
    INCLUDE "FRMFAI.H"
1  CONTINUE
ENDIF

110  FORMAT(1X,
C1      KEX,      XXXO, (FO(J, I), J=2, MXJ)
> 1P,      1X, I2,      7(1X, E16.8)

C2      XXX, Y, T*1.D3,
>      , 3(1X, E24.16)

C3      Z, P*1.D3, SAR, TAR
>      , 4(1X, E24.16)

C4      SXo, SYo, SZo, So, SX, SY, SZ, S
>      , 8(1X, E15.7)

C5      ENEKI, ENERG
>      , 2(1X, E16.8)

C6      IT, IREP(IT), SORT(IT), (AMQ(J, I), J=1, 5), RET(IT), DPR(IT), PS
>      , 2(1X, I6),      9(1X, E16.8)

C7      BORO, IPASS, NOEL,
>      , 1X, E16.8,      2(1X, I6)

C8      'KLEY',      (' LABEL(NOEL, I)', I=1, 2),      'LET(IT)'
>      , 1X, A1, A10, A1,      2(1X, A1, A , A1),      1X, 3A1)

```

The meaning of the main data is the following (see the keyword *OBJET*)

$LET(I)$	: one-character string, for tagging particle number $I$
$IEX, I, IREP(I)$	: flag, particle number, index
$FO(1 - 6, I)$	: coordinates $D, Y, T, Z, P$ and path length at the origin of the optical sequence
$F(1 - 6, I)$	: <i>idem</i> at the current position
$SORT(I)$	: path length at which the particle has possibly been stopped (see <i>CHAMBR</i> or <i>COLLIMA</i> )
$RET(I), DPR(I)$	: synchrotron phase space coordinates ; $RET$ =phase (radian), $DPR$ = momentum dispersion (MeV/c) (see <i>CAVITE</i> )
$IPASS$	: turn number (see <i>REBELOTE</i> )
<i>etc.</i>	:

• *FAISTORE* has an effect similar to *FAISCNL*, with more features.

- On the first data line, *FNAME* may be followed by a series of up to 10 *LABELS*. If there is no label, the print occurs by default at the location of *FAISTORE* ; if there are labels the print occurs right downstream of all optical elements wearing those labels (and no longer at the *FAISTORE* location).

- The next data line gives a parameter,  $IP$  : printing will occur at pass 1 and then at every  $IP$  other pass, if using *REBELOTE* with  $NPASS \geq IP - 1$ .

For instance the following data list in *zgoubi.dat* :

```
FAISTORE
zgoubi.fai   HPCUP   VPCKUP
12
```

will result in output prints into *zgoubi.fai*, at pass 1 and then every 12 other pass, each time elements of the *zgoubi.dat* optical sequence labeled either *HPCUP* or *VPCKUP* are encountered.

**Case of Binary storage :** it can be obtained from *FAISCNL* and *FAISTORE*. This is for the sake of compactness and access speed, for instance in case voluminous amounts of data would have to be manipulated using **zpop**.

This is achieved by giving the storage file a name of the form  $b\_FNAME$  or  $B\_FNAME$  (e.g., 'b\_zgoubi.fai'). The *FORTTRAN WRITE* list is the same as in the *FORMATTED* case above.

This is compatible with the *READ* statements in **zpop** that will recognize binary storage from that very radical 'b\_' or 'B\_'.

**Case of FIT[2] procedure :** it may not be desired to store during the FITting process, but instead only when the FITting is completed. It is sufficient for that to (i) put 'AtFITfinal' as the first label following *FAISTORE* keyword, this will inhibit all storage until the final run following a FIT procedure, and (ii) avoid using the 'nofinal' instruction in *FIT[2]* (see p. 168).

**FOCALE, IMAGE[S] : Particle coordinates and beam size ; localization and size of horizontal waist**

*FOCALE* calculates the dimensions of the beam and its mean transverse position, at a longitudinal distance *XL* from the position corresponding to the keyword *FOCALE*.

*IMAGE* computes the location and size of the closest horizontal waist.

*IMAGES* has the same effect as *IMAGE*, but, in addition, for a non-monochromatic beam it calculates as many waists as there are distinct momenta in the beam, provided that the object has been defined with a classification of momenta (see *OBJET[KOBJ= 1or2]* for instance).

Optionally, for each of these three procedures, **zgoubi** can list a trace of the coordinates in the *X*, *Y* and in the *Y*, *Z* planes.

The following quantities are calculated for the *N* particles of the beam (*IMAGE*, *FOCALE*) or of each group of momenta (*IMAGES*)

- Longitudinal position :

$$\text{FOCALE : } X = XL$$

$$\text{IMAGE[S] : } X = - \frac{\sum_{i=1}^N Y_i \times \text{tg}T_i - \left( \sum_{i=1}^N Y_i \times \sum_{i=1}^N \text{tg}T_i \right) / N}{\sum_{i=1}^N \text{tg}^2T_i - \left( \sum_{i=1}^N \text{tg}T_i \right)^2 / N}$$

$$Y = Y_1 + X \times \text{tg}T_1$$

where  $Y_1$  and  $T_1$  are the coordinates of the first particle of the beam (*IMAGE*, *FOCALE*) or the first particle of each group of momenta (*IMAGES*).

- Transverse position of the center of mass of the waist (*IMAGE[S]*) or of the beam (*FOCALE*), with respect to the reference trajectory

$$YM = \frac{1}{N} \sum_{i=1}^N (Y_i + X \text{tg}T_i) - Y = \frac{1}{N} \sum_{i=1}^N Y M_i$$

- FWHM of the image (*IMAGE[S]*) or of the beam (*FOCALE*), and total width, respectively, *W* and *WT*

$$W = 2.35 \left( \frac{1}{N} \sum_{i=1}^N Y M_i^2 - Y M^2 \right)^{\frac{1}{2}}$$

$$WT = \max(YM_i) - \min(YM_i)$$

**FOCALEZ, IMAGE[S]Z : Particle coordinates and beam size ; localization and size of vertical waist**

Similar to *FOCALE* and *IMAGE[S]*, but the calculations are performed with respect to the vertical coordinates  $Z_i$  and  $P_i$ , in place of  $Y_i$  and  $T_i$ .

**HISTO : 1-D histogram**

(page 295 in PART B)

Any of the coordinates used in **zgoubi** may be histogrammed, namely initial  $Y_0, T_0, Z_0, P_0, S_0, D_0$  or current  $Y, T, Z, P, S, D$  particle coordinates ( $S =$  path length ;  $D$  may change in decay process simulation with *MCDESINT*, or when ray-tracing in  $\vec{E}$  fields), and also spin coordinates and modulus  $S_X, S_Y, S_Z$  and  $\|\vec{S}\|$ .

*HISTO* can be used in conjunction with *MCDESINT*, for statistics on the decay process, by means of *TYP*. *TYP* is a one-character string. If it is set equal to 'S', only secondary particles (they are tagged with an 'S') will be histogrammed. If it is set equal to 'P', then only parent particles (non-'S') will be histogrammed. For no discrimination between S-econdary and P-arent particles, *TYP* = 'Q' must be used.

The dimensions of the histogram (number of lines and columns) may be modified. It can be normalized with *NORM* = 1, to avoid saturation.

Histograms are indexed with the parameter *NH*. This allows making independent histograms of the same coordinate at several locations along an optical sequence. This is also useful when piling up problems in a single input data file (see also *RESET*). *NH* is in the range 1-5.

If *REBELOTE* is used, the statistics on the  $1+NPASS$  runs through the optical sequence will add up.

**PRINT** : The optional command 'PRINT', under *HISTO* keyword, will cause logging of computation outcomes in the file *zgoubi.HISTO.Out*.



**IMAGE[S][Z] : Localization and size of vertical waists**

See FOCAL[Z].

**MATRIX : Calculation of transfer coefficients, periodic parameters**

(page 302 in PART B)

*MATRIX* allows computation of transport matrices through the optical sequence up to third order, from *OBJET* down to the location(s) where *MATRIX* is introduced in the optical sequence (which can be at arbitrary locations), or, upon option, down to the horizontal focus closest to *MATRIX* keyword. In the latter case the position of the focus is calculated automatically in the same way as the position of the waist in *IMAGE*.

An automatic generation of an appropriate object for the use of *MATRIX* can be obtained from the procedure *OBJET* (pages 63, 308), as follows:

- if *IOR* = 1, use *OBJET*[*KOBJ*= 5[.N, 2 ≤ N ≤ 99]], that generates up to 99\*13 initial coordinate sets (*Y, T, Z, P, S, D*). Up to 99 transport matrices are computed, each one wrt. to the reference trajectory of concern (trajectory number 1, 14, 27, ..., (13N-12), respectively).
- if *IOR* = 2, use *OBJET*[*KOBJ*= 6] that generates 61 initial coordinate sets.
- if *IOR* = 3, use *OBJET*[*KOBJ*= 6.1] that generates 102 initial coordinate sets.

The different procedures available are as follows:

- If *IOR* = 0, *MATRIX* is inhibited (equivalent to *FAISCEAU*, whatever *IFOC*).
- If *IOR* = 1, using *OBJET*[*KOBJ*= 5[.N]], the first order transport matrix [*R<sub>ij</sub>*] is calculated, from a third order approximation on the coordinates. For instance

$$Y^+ = \left(\frac{Y}{T_0}\right) T_0 + \left(\frac{Y}{T_0^2}\right) T_0^2 + \left(\frac{Y}{T_0^3}\right) T_0^3 + \mathcal{O}(T_0^4), \quad Y^- = - \left(\frac{Y}{T_0}\right) T_0 + \left(\frac{Y}{T_0^2}\right) T_0^2 - \left(\frac{Y}{T_0^3}\right) T_0^3 + \mathcal{O}(T_0^4)$$

will yield, neglecting third order terms,

$$R_{11} \approx \left(\frac{Y}{T_0}\right) = \frac{Y^+ - Y^-}{2T_0}$$

Matrix computation is repeated *N* times if *OBJET*[*KOBJ*= 5.N] (2 ≤ *N* ≤ 99) is used, so delivering a first order transport matrix for each 13-particle subset, with for each subset the reference trajectory being trajectory number 1+(*N*-1)×13 (*FAISCEAU* will print out the *N*×13 particle list in *zgoubi.res*, if desired).

In the particular case  $KOBJ= 5.1$ , *MATRIX* will also transport optical functions:

- this requires the initial optical function values

$$\alpha_{Y,Z}, \alpha'_{Y,Z}, D_{Y,Z}, D'_{Y,Z}, \quad \text{at } OBJET$$

to be provided, by *OBJET*;

- then, using the  $[R_{ij}]$  matrix above, optical functions and phase advances  $\phi_Y, \phi_Z$  are transported following

$$\begin{pmatrix} \beta \\ \alpha \\ \gamma \end{pmatrix}_{\text{at } MATRIX} = \begin{pmatrix} R_{11}^2 & -2R_{11}R_{12} & R_{12}^2 \\ -R_{11}R_{21} & R_{12}R_{21} & R_{11}R_{12} \\ R_{21}^2 & -2R_{21}R_{22} & R_{22}^2 \end{pmatrix} \begin{pmatrix} \beta \\ \alpha \\ \gamma \end{pmatrix}_{\text{at } OBJET}$$

$$\Delta\phi_Y = A \tan \frac{R_{12}}{(R_{11}\beta_{Y,objet} - R_{12}\alpha_{Y,objet})}, \quad \Delta\phi_Z = A \tan \frac{R_{34}}{(R_{33}\beta_{Z,objet} - R_{34}\alpha_{Z,objet})}, \quad (6.5.1)$$

$$\phi_{Y,Z} \rightarrow \phi_{Y,Z} + 2\pi \quad \text{if } \phi_{Y,Z} < 0, \text{ given } [0, \pi] \text{ A tan determination}$$

Outcomes are printed in *zgoubi.res*, and as well in *zgoubi.MATRIX.out* ancillary output file *iff PRINT* is part of *MATRIX* arguments.

- If  $IOR = 2$ , using *OBJET*[ $KOBJ= 6$ ] for the automatic generation of an *ad hoc* 61-particle initial object, fifth order Taylor expansions are used for the calculation of the first order transport matrix  $[R_{ij}]$  and of the second order matrix  $[T_{ijk}]$ . Other higher order coefficients are also calculated.
- If  $IOR = 3$ , using *OBJET*[ $KOBJ= 6.1$ ] for the automatic generation of an *ad hoc* 102-particle initial object, transport coefficients up to 3rd order are computed (after routines developed for the RAYTRACE computer code [74, 75]).

The next option, *IFOC*, acts as follows

- If  $IFOC = 0$ , the transport coefficients are calculated at the location of *MATRIX*, and with respect to the reference trajectory. For instance,  $Y^+$  and  $T^+$  above are defined for particle number  $i$  as  $Y^+ = Y^+(i) - Y(\text{ref.})$ , and  $T^+ = T^+(i) - T(\text{ref.})$ .
- If  $IFOC = 1$ , the transport coefficients are calculated at the horizontal focus closest to *MATRIX* (determined automatically), while the reference direction is that of the reference particle. For instance,  $Y^+$  is defined for particle number  $i$  as  $Y^+ = Y^+(i) - Y_{\text{focus}}$ , while  $T^+$  is defined as  $T^+ = T^+(i) - T(\text{ref.})$ .
- If  $IFOC = 2$ , no change of reference frame is performed : the coordinates refer to the current frame. Namely,  $Y^+ = Y^+(i)$ ,  $T^+ = T^+(i)$ , etc.

### Periodic Optical Sequences

- If  $IFOC = 10 + N\text{Period}$ , then, from the 1-turn transport matrix as obtained in the way described above, *MATRIX* calculates periodic parameters characteristic of the optical sequence such as optical functions and tune numbers, assuming that it is  $N\text{Period}$ -periodic. In the coupled hypothesis (default), the computation uses the Edwards-Teng method [44]. In non-coupled hypothesis (this requires “*uncoupled*” argument, see below), this is performed by a simple identification

$$[T_{ij}] = I \cos \mu + J \sin \mu$$

Matrix computation is repeated  $N$  times if *OBJET*[ $KOBJ= 5.N$ ] ( $2 \leq N \leq 99$ ) is used, so delivering first order data for each of the  $N \times 13$ -particle sets.

If  $IOR D = 2$  (using  $OBJET[KOBJ= 6]$ ) additional periodic parameters are computed such as chromaticities, momentum dependence of optical functions, etc.

**PRINT** : Addition of “*PRINT*” following  $IOR D$ ,  $IFOC$  will cause stacking of *MATRIX* output data into `zgoubi.MATRIX.out` file (convenient for use with *e.g.* gnuplot type of data treatment software).

Addition of “*uncoupled*” next to  $IOR D$ ,  $IFOC$  [, *PRINT*], in the case of periodic beam matrix request (i.e.,  $IFOC = 10 + NPeriod$ ) will cause use of uncoupled formalism.

About the source code :

The program `matric` computes the transport matrix coefficients, it is called by `zgoubi` when the keyword *MATRIX* is met along the `zgoubi.dat` sequence. `matimp`, called by `matric`, ensures the print out of the matrix (and possibly the beam matrix) in `zgoubi.res`, and upon ‘*PRINT*’ option in `zgoubi.MATRIX.out` as well.

**PICKUPS : Bunch centroid path; orbit**

*PICKUPS* computes the average values of the coordinates of the particles in a bunch (coordinates of the bunch centroid) at one or more keywords specified by their *LABELs*. The list of *LABEL(s)* concerned is specified by the user, as part of the arguments under the keyword *PICKUPS*.

In conjunction with *REBELOTE* in the case of a periodic optical sequence, with *REBELOTE* thus being used to cause multi-turn tracking, *PICKUPS* are reset (their records are zero-ed) at start of each turn.

Pickup data computed during **zgoubi** execution are stored in the file *zgoubi.PICKUP.out*, usable for further analysis, or for plotting (for instance, using *gnuplot*).

A summary of the list of pickups and their individual bunch data is listed in the result file *zgoubi.res* (below the *END* keyword) at the end of **zgoubi** execution. It is concluded with statistics as the *rms*, minimum and maximum values of the bunch centroid data over the ensemble of pickups.

**PLOTDATA : Intermediate output for the PLOTDATA graphic software**

*PLOTDATA* was at the origin implemented for the purpose of plotting particle coordinates using the TRI-UMF *PLOTDATA* package [76]. However nothing precludes using it with a different aim.

The *PLOTDATA* keyword can be introduced at up to 20 locations in *zgoubi.dat*. There, particle coordinates will be stored in a local array, *FF*. They are overwritten at each pass. Usage of *FF* is left to the user, see *FORTTRAN* subroutine `pltdat.f`.

**SPNPRNL, SPNSTORE : Print/Store spin coordinates**

• *SPNPRNL* has similar effect to *SPNPRT* (page 208), except that the information is stored in a dedicated file *FNAME* (should post-processing with **zpop** follow, advised name is *FNAME* = 'zgoubi.spn' (formatted write) or 'b\_zgoubi.spn' (binary write) ). The data are formatted and ordered according to the *FORTTRAN* sequence found in the subroutine `spnprn.f`, with meaning of printed quantities as follows :

*LET(I), IEX(I)* : tagging character and flag (see *OBJET*)  
*SI(1-4,I)* : spin components *SX*, *SY*, *SZ* and modulus, at the origin  
*SF(1-4,I)* : *idem* at the current position  
*GAMMA* : Lorentz relativistic factor  
*I* : particle number  
*IMAX* : total number of particles ray-traced (see *OBJET*)  
*IPASS* : turn number (see *REBELOTE*)

• *SPNSTORE* has an effect similar to *SPNPRNL*, with two more features.

- On the first data line, *FNAME* may be followed by a series of up to 10 *LABELs* proper to the elements of the *zgoubi.dat* data file at the exit of which the print should occur ; if no label is given, the print occurs by default at the very location of *SPNSTORE* ; if labels are given, then print occurs right downstream of all optical elements wearing those labels (and no longer at the *SPNSTORE* location).

- The next data line gives a parameter, *IP* : printing will occur every *IP* other pass, when using *REBELOTE* with  $NPASS \geq IP - 1$ .

For instance the following data input in *zgoubi.dat* :

```
SPNSTORE
zgoubi.spn  HPCUP  VPCUP
12
```

will result in output prints into *zgoubi.spn*, every 12 other pass, each time elements of the *zgoubi.dat* optical sequence labeled either *HPCUP* or *VPCUP* are encountered.

Note

Binary storage can be obtained from *SPNPRNL* and *SPNSTORE*. This is for the sake of compactness and I/O access speed by *zgoubi* or *zpop*, for instance in case voluminous amounts of data should be manipulated. This is achieved by giving the storage file a name of the form *b\_FNAME* or *B\_FNAME* (e.g., 'b\_zgoubi.spn'). The *FORTTRAN WRITE* output list is the same as in the *FORMATTED* case above.

**SPNPRT : Print spin coordinates**

(page 337 in PART B)

*SPNPRT* can be introduced anywhere in an optical sequence. It produces a print out (to *zgoubi.res* by default) of various informations such as the initial and actual coordinates and modulus of the spin of the *IMAX* particles, their Lorentz factor  $\gamma$ , the mean values of the spin components, etc., at the location where it is placed in the *zgoubi.dat* optical sequence.

*PRINT* : If “*PRINT*” appears after *SPNPRT*, then spin data will be stored in *zgoubi.SPNPRT.Out*. The latter is opened at the first occurrence of “*PRINT*”. In particular, if *REBELOTE* is used then data will be stacked in *zgoubi.SPNPRT.Out*.

*MATRIX* : If “*MATRIX*” appears after *SPNPRT*, then spin rotation matrix (or matrices) will be computed. This is done assuming that appropriate initial particle and spin sampling has been defined (by respectively *OBJET* and *SPNTRK*), as follows :

- three particles are needed for *SPNPRT* to compute a spin matrix. All three have identical initial coordinates (for instance in a periodic optical sequence, the local stable precession axis coordinates) and rigidity, and their spins are on a direct trièdre, for instance, respectively,

$$(S_X, S_Y, S_Z) = (1, 0, 0), (0, 1, 0), (0, 0, 1),$$

- there may be several groups of three particles, each group with a particular rigidity (*i.e.*, the three particles in a group have the same momentum). *OBJET[KOBJ=1]* or *[KOBJ=2]* can be used for instance to generate these particles by groups of like rigidity. *SPNPRT[MATRIX]* will thus compute as many matrices.

An example of spin matrix computation is given page 337.



**SRPRNT : Print SR loss statistics**

*SRPRNT* may be introduced anywhere in an optical sequence. It allows switching on synchrotron radiation energy loss computation. It produces in addition a print out (to zgoubi.res) of current state of statistics on several parameters related to SR loss presumably activated beforehand with keyword *SRLOSS*.

**TWISS : Compute periodic optical parameters. Log to zgoubi.TWISS.out**

(page 351 in PART B)

*TWISS* command operates as a do-loop, it has to be placed at the bottom of the loop in question, in the input data sequence; the program execution pointer will quietly continue beyond the command once it is completed.

*TWISS* causes the calculation of transport coefficients and various other global parameters, in particular periodical quantities as tunes and optical functions, in the coupled hypothesis. *TWISS* causes a succession of up to 5 passes through the sequence (by sending the program execution pointer back to the top of the input data file, up to 4 successive times).

The object necessary for these calculations will be generated automatically if one uses *OBJET* with option *KOBJ=5*. This requires knowledge of the reference orbit, as coordinates of trajectory number 1 of the 13-set defined by *OBJET[KOBJ=5]* have to be on the periodic orbit (trajectory number 1 is the reference for transport coefficient calculation).

*FIT* may precede *TWISS* in case the periodic orbit coordinates at the beginning of the sequence are not known, for an automatic search, see example in PART. B, p. 352.

*TWISS* works in a way similar to *MATRIX*, iterating the *MATRIX* process wherever necessary, changing for instance the reference trajectory in *OBJET* for dp/p related computations. In particular :

- It assumes that the reference particle (particle #1 of 13, when using *OBJET[KOBJ=5]*) is located on the closed orbit. *This condition has to be satisfied for TWISS to work consistently*, because *TWISS* does not look for the closed orbit by itself. A simple way to first find the orbit is to have *TWISS* preceded by a dedicated *FIT[2]*, see example in PART B, page 351.

- A first pass (the only one if *KTW=1*) through the optical sequence allows computing the periodic optical functions at the end of the sequence, from the rays.

- The periodic dispersions from the first pass are used to define chromatic closed orbits at  $\pm\delta p/p$  (or at  $\pm FacD \times \delta p/p$ , accounting for a user's parameter  $FacD \neq 1$ ). A second and a third pass (which terminate the process if *KTW=2*) with chromatic objects centered respectively on  $\pm\delta p/p$  chromatic orbits will then compute the chromatic first order transport matrices. From these the chromaticities are deduced. Full accuracy on the chromatic orbits can be obtained using *FIT[2]*.

- Anharmonicities need two additional passes (which terminate the process if *KTW=3*). They are deduced from the difference in tunes for particles tracked on different transverse invariants, horizontal or vertical.

The execution of *TWISS* will cause printout to the file *zgoubi.TWISS.out*, a MADX type "twiss" file, with similar formatting, including periodic optical data as a header (tunes, chromaticities, etc.) and, in sequence element by element : optical functions, element strength, first order transport coefficients, spin  $\vec{n}_0$  vector (requires 'SPNTRK' request).

### SR Integrals, DK Spin Integrals

Presence of "SRINT" in *TWISS* arguments will cause computation of Synchrotron radiation integrals, which will then be added to *zgoubi.TWISS.out* header. Presence of "DKINT" in *TWISS* arguments will cause computation of Derbenev-Kondratenko integrals, which will then be added to *zgoubi.TWISS.out* header. In the case SR and/or DK spin integrals are requested, they are computed along particle number 1 trajectory, it is the responsibility of the user to ensure that particle number 1 is on the closed orbit, and the DK spin integrals case to ensure in addition that its initial spin coordinates in *SPNTRK* are on the spin closed orbit.

SR and DK spin integrals are computed step by step; all optical elements contribute (not just dipoles) as long as trajectory number 1 experiences  $1/\rho \neq 0$ .

About the source code :

The program `beamat` ensures the transport of the beam matrix along the `zgoubi.dat` sequence, at the origin of the content of `zgoubi.TWISS.out`. `beamat` is called by `opticc`, itself called after each optical element, in the program `zgoubi`.

The starting beam matrix value is in the array `F1`, initialized by the program `twiss` when met at the end of the `zgoubi.dat` sequence. The array `F0` contains the running beam matrix.

The program `optimp`, called from `opticc`, prints to `zgoubi.TWISS.out` after each optical element of the `zgoubi.dat` sequence.

## 7 COMPLEMENTS REGARDING VARIOUS FUNCTIONALITIES

### 7.1 Units in zgoubi

Units to be employed in **zgoubi** input data list (in zgoubi.dat) are stated in Part B of this guide. Part B should be carefully referred to about units to be used, when setting up the data list, as some insight in the code indicates :

Working units in **zgoubi** numerical integrator, and as to the normalized fields delivered to the integrator, satisfy to the following (in particular, whatever the units in the input data file, as listed in Part B, quantities of concern end up manipulated as follows) :

- position coordinates (X, Y, Z, S) in centimeter,
- angle coordinates (T, P) in radian,
- time in  $\mu$ second,
- energy in MeV,
- magnetic field in kG
- electrostatic field in MV/cm
- rigidity  $B\rho$  in kG.cm,
- fields delivered to the integration procedure after conversion to strengths, units of  $m^{-\text{integer}}$ , for instance,
  - magnetic dipole strength ( $B/B\rho = 1/\rho$ ) is in  $m^{-1}$  (with  $B$  the field),
  - $2(n+1)$  order magnetic multipole strengths ( $(B/R^n)/B\rho$ ,  $n=1$  (quadrupole),  $n=2$  (sextupole), etc.) are in  $m^{-(n+1)}$  (with  $B$  the field at pole tip radius  $R$ ),
  - etc.,
  - $2(n+1)$  order magnetic multipole strengths ( $(\Phi/R^{(n+1)})/W$ ,  $n=1$  (quadrupole),  $n=2$  (sextupole), etc) are in  $m^{-(n+1)}$  (with  $\Phi$  the potential (in Volt) at pole tip radius  $R$  and  $W$  the kinetic energy of the particle (in electron-Volt)).

For historical reasons, since **zgoubi** was first developed in 1972, implementation of optical elements (as listed in Part B) has resulted in parameter definition resorting to a wide variety of units. For instance :

- element lengths in meter or centimeter,
- positioning angles in rad, mrad, degree,
- magnetic fields in Tesla or kGauss,
- electrostatic potentials in Volt, MVolt,
- electrostatic fields in Volt/m, V/cm, MVolt/m,
- etc.

## 7.2 Reference Rigidity

**zgoubi** computes the strengths of optical elements (they are usually defined by their field) from the reference rigidity *BORO* as defined in *[MC]OBJET*. However using *CAVITE*, and indirectly *SCALING*, may affect the reference rigidity, following

$$B\rho_{ref} = BORO \longrightarrow B\rho_{ref} = BORO + \delta B\rho_s \quad (7.2.1)$$

with  $\delta B\rho_s$  increase (or decrease) of the reference rigidity. A typical configuration where this would occur is that of multi-turn tracking in a pulsed synchrotron (see section 7.9), where in general optical element strengths have to follow the acceleration so that the orbit and lattice optics remain unchanged.

## 7.3 Reference Trajectory, Closed Orbit

Various procedures in **zgoubi** assume prior knowledge of a reference orbit. Or the closed orbit, in the case of an optical sequence assumed to be periodic. This is the case for instance for *MATRIX*, for *TWISS*, when searching the spin closed orbit, when multi-turn tracking an initially centered bunch (see also p. 89), etc.

The initial coordinates of that reference trajectory are specified in *[MC]OBJET*. Initial spin coordinates are specified in *SPNTRK*. They may possibly be found by a prior *FIT[2]* (with for instance “3.1” or “10.1” style periodicity constraints, in the case of a periodic sequence or in the case of a closed spin orbit, respectively).

In the case of the *MATRIX* and *TWISS* procedures, both keywords can be preceded with *FIT[2]* for that purpose of prior finding of the reference trajectory or closed orbit, or spin closed orbit.

## 7.4 Time Varying Fields

Fields can be varied as a function of time (in some cases this may mean as a function of turn number, see section 7.9), by means of the *SCALING* keyword.

Eventually some families of magnets may be given a different timing law for the simulation of special processes (e.g., time varying orbit bump, fast crossing of spin resonances with families of jump quadrupoles).

## 7.5 Backward Ray-Tracing

For the purpose of parameterization for instance, it may be interesting to ray-trace backward from the image toward the object. This can be performed by first reversing the position of optical elements in the optical sequence, and then reversing the integration step sign in all the optical elements.

An illustration of this feature is given in the following Figure 48.

## 7.6 Checking Trajectories and Fields Across Optical Elements - the *IL*, *.plt* and *IC* flags

- In all optical elements, an option *IL* is available. It is normally set to  $IL = 0$  and in this case has no effect. Other possibilities are as follows :

- $IL = 1$  causes a print in *zgoubi.res* of particle coordinates, fields and low-order derivatives, spin vector and other data, at each integration step along trajectories across the optical element. In the meantime, a calculation and summation of the values of  $\vec{\nabla} \cdot \vec{B}$ ,  $\vec{\nabla} \times \vec{B}$  and  $\nabla^2 \vec{B}$  (same for  $\vec{E}$ ) at all integration steps is performed, which allows a check of the behavior of  $\vec{B}$  (and/or  $\vec{E}$ ) in field maps (all these derivatives should normally be zero).

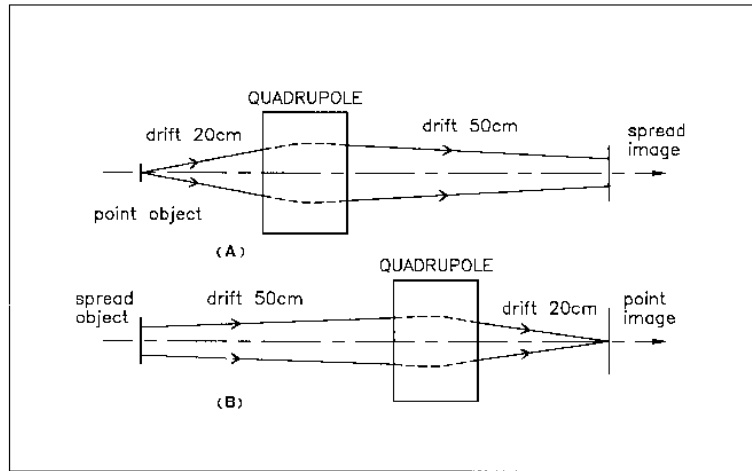


Figure 48: A. Regular forward ray-tracing, from object to image.  
 B. Same optical sequence, with backward ray-tracing from image to object :  
 negative integration step XPAS is used in the quadrupole.

-  $IL = 2$  causes a print of particle data (coordinates,  $\vec{B}$  and/or  $\vec{E}$  field(s) experienced, RF cavity related data, spin, etc.), particle status (e.g., decay status, whether it is still tracked, etc.), into the dedicated file `zgoubi.plt`, one line of data per article, at each integration step. This information can further be processed (using `zpop`<sup>18</sup> or `gnuplot`, for instance). In order to minimize the volume of that storage file (when dealing with small step size, long optical elements, large number of particles, etc.) it is possible to print out every other  $10^n$  integration step by taking  $IL = 2 \times 10^n$  instead (for instance,  $IL = 200$  would cause output into `zgoubi.plt` every 100 other step).

-  $IL = 7$  causes similar outputs to  $IL = 1$ , and more, in particular derivatives to higher order, yet into the dedicated file `zgoubi.impdev.out`, one line of data per particle, at each integration step. This information can be further plotted (e.g., using `zpop`, or `gnuplot`). An example is given in fig. 49, a plot of the quadrupole and sextupole field indices along the reference orbit in a combined function main dipole pertaining to the AGS lattice [77].

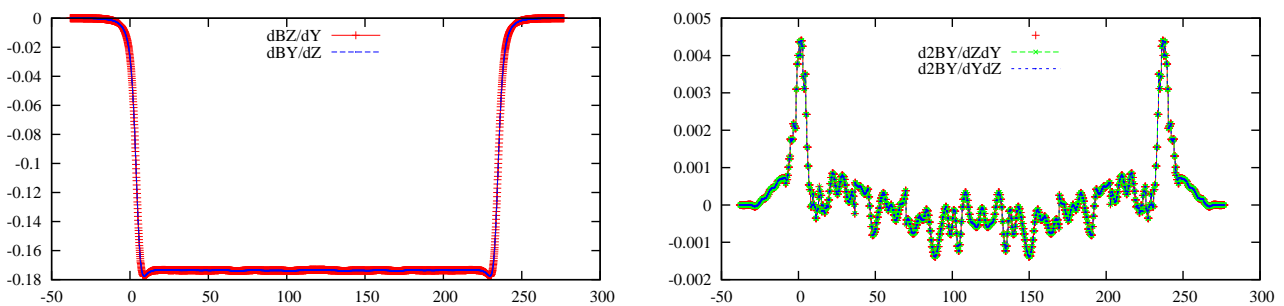


Figure 49: Typical field and derivatives - here, along the 10 GeV proton orbit across an AGS A-type main magnet. Derivatives are obtained from second degree polynomial interpolation from the magnet measured field map. Left :  $dB_Z(s)/dY$  and  $dB_Y(s)/dZ$ , right :  $d^2B_Y(s)/dZdY$ ,  $d^2B_Y(s)/dYdZ$ . These data were stored in `zgoubi.impdev.out` during execution, by stating  $IL = 7$  under `TOSCA` keyword.

- $IL$  can be forced to the aforementioned values, in all optical elements at once, by means of `OPTIONS`.
- The `.plt` flag can be added as a second `LABEL` in the case of `MARKER` or `DRIFT` keywords. It comes in complement to  $IL = 2$  as the latter is not available (actually:  $IL=2$  is available in `DRIFT` in the presence of the `'split'` flag, see page 109), to log local coordinates to `zgoubi.plt` in the same manner as  $IL=2$  does.

<sup>18</sup>See Part D of the Guide.

- *IC*: When dealing with field maps (e.g., *CARTEMES*, *ELREVOL*, *TOSCA*), an option index *IC* is available. It is normally set to  $IC = 0$  and in this case has no effect. Other possibilities are as follows :

- $IC = 1$  causes a print of the field map in *zgoubi.res*.

- $IC = 2$  will cause a print of the field map into the file *zgoubi.map* which can further be processed with **zpop** for plotting, superimposing trajectories, and other data treatment purposes.

### 7.7 Labeling Keywords: LABEL1, LABEL2

Keywords in **zgoubi** input data file *zgoubi.dat* may be LABEL'ed, for the purpose of various options in the execution of such procedures as *REBELOTE*, *PICKUPS*, *FAISCNL*, *FAISTORE*, *SCALING*, or also in order to allow for particle coordinate print out to *zgoubi.plt* (see section 7.6,  $IL = 2$  option).

A keyword in *zgoubi.dat* accepts two LABELs normally referred to as LABEL1, LABEL2. The first one is used for the above mentioned purposes. An instance of the use of the second one is with *MARKER* or with *DRIFT* : it can be set to ".plt" for storage of current particle data into *zgoubi.plt*. The keyword and its LABEL[s] should fit within a 110-character long string on a single line (a quantity set in the FORTRAN file *prdata.f*).

### 7.8 Multi-turn Tracking in Circular Machines

There are two ways to compute chromaticities.

1. Use *OBJET*[KOBJ=6], and *MATRIX* at the end of the sequence.
2. Use *OBJET*[KOBJ=5], and *TWISS* at the end of the sequence.

### 7.9 Multi-turn Tracking in Circular Machines

Multi-turn tracking in circular machines can be performed by means of the keyword *REBELOTE*. *REBELOTE* is introduced in the *zgoubi.dat* optical sequence with its argument  $NPASS+1$  being the number of turns to be performed. It will cause a jump of the execution pointer back to (details on page 180), either the beginning of *zgoubi.dat* data list (default case), or to a particular LABEL in that list. From there on, tracking resumes down to *REBELOTE* again, and so forth until the requested number of passes has been reached. At that point, the execution simply continues beyond *REBELOTE* down to the *END* keyword.

Note : Some *CAVITE* options cause a reset-to-zero of individual particle path length<sup>19</sup>. This is for reasons of cumulated path length accuracy (a delta-path below computer accuracy compared to total path length would not be resolved - it can be for instance bunch length as compared to cumulated multi-turn distance around a large ring). This reset may not be desirable, it depends on the multi-turn problem dealt with using *REBELOTE*, it may be necessary for instance in long-term tracking in large rings, it is not in recirculating linacs. Option  $K=99$  in *REBELOTE* will cause this reset, whereas option  $K=98$  (UNDER DEVELOPMENT, IN *cavite.f*, IN THIS VERSION OF THE CODE) will avoid it, with everything else the same as with  $K=99$ .

Possible magnet timing laws  $B(T)$  during the multi-turn process (with  $T = 1$  to  $NPASS+1$  counted in number of turns) can be introduced by means of *SCALING*.

In order that the *IMAX* particles of the beam start a new pass with the coordinates they had reached at the end of the previous one, the option  $K = 99$  has to be specified in *REBELOTE*.

<sup>19</sup>For the geek: the "f(6,I)=0.d0" statements in *cavite.f*.

**Synchrotron acceleration** can be simulated, using following the procedure :

- *CAVITE* appears in the *zgoubi.dat* optical sequence (normally before *REBELOTE*), with option *IOPT*  $\neq$  0,
- the RF frequency of the cavity may be given a timing law  $f_{RF}(T)$  by means of *SCALING*, family *CAVITE*,
- the magnets are given a field timing law  $B(T)$ , (with  $T = 1$  to  $NPASS+1$  counted in number of turns) by means of *SCALING*.

Eventually some families of magnets may be given a different timing law for the simulation of special processes (e.g., time varying orbit bump, fast crossing of spin resonances with families of jump quadrupoles).

### 7.10 *KPOS* : Positioning, (Mis-)Alignment, of Optical Elements and Field Maps

The last record in most optical elements and field maps is the positioning option *KPOS*. *KPOS* is followed by the positioning parameters.

In most optical elements it operates as follows:

- either *XCE*, *YCE* for translation and *ALE* for Z-rotation, if the optical element is defined in a Cartesian frame ( $O; X, Y, Z$ ),
  - or *RE*, *TE* entrance radius and Z-rotation (E stands for the French “Entrée”) and similar *RS*, *TS* at exit (S stands for the French “Sortie”), if the optical element is defined in a cylindrical frame ( $O; R, \theta, Z$ ).
- The positioning works in two different ways, depending whether the element is defined in Cartesian ( $X, Y, Z$ ) coordinates (e.g., *QUADRUPO*, *TOSCA[MOD $\leq$ 19]*), or polar ( $R, \theta, Z$ ) coordinates (e.g., *DIPOLE*, *TOSCA[MOD $\geq$ POLARMES]*).

*ERRORS* allows random misalignmet of some magnets<sup>20</sup>.

Additional possibilities are available in some other optical elements, see below.

#### Cartesian Coordinates :

If *KPOS* = 1, the optical element is moved (shifted by *XCE*, *YCE* and Z-rotated by *ALE*) with respect to the incoming reference frame. Trajectory coordinates after traversal of the element refer the element frame.

If *KPOS*= 2, the shifts *XCE* and *YCE*, and the tilt angle *ALE* are taken into account, for mis-aligning the element with respect to the incoming reference, as shown in Fig. 50. The effect is equivalent to a *CHANGREF(XCE, YCE, ALE)* upstream of the optical element, followed by *CHANGREF(XCS, YCS, ALS = -ALE)* downstream of it, with computed *XCS*, *YCS* values as schemed in Fig. 50.

*KPOS* = 3 option is available for some optical elements defined in a Cartesian frame (e.g., *BEND*, *MULTIPOL*, *AGSMM*) ; it is effective only if a non zero dipole component *B1* is present, or if *ALE* is non-zero. It positions automatically the magnet in a symmetric manner with respect to the incoming and outgoing reference axis, convenient for periodic optical sequences, as follows (Fig 51).

Both incoming and outgoing reference frames are tilted w.r.t. the magnet,

- either, by an angle *ALE* if  $ALE \neq 0$ ,

• or, if  $ALE=0$  by half the Z-rotation  $\alpha_Z/2$  such that  $L = 2 \frac{BORO}{B1} \sin(\alpha_Z/2)$ , wherein  $L$  = geometrical length, *BORO* = reference rigidity as defined in *[MC]OBJET*.

Next, the optical element is Y-shifted by  $\delta Y = YCE$  (*XCE* is not used) in a direction orthogonal to the new magnet axis (i.e., at an angle  $ALE + \pi/2$  wrt. the *X* axis of the incoming reference frame).

*KPOS* = 4 is UNDER DEVELOPMENT. It has been introduced for *AGSMM* keyword (AGS main magnet), it is used in the on-line AGS model at BNL. It also applies to *MULTIPOL*. It orients the magnet in a way similar to *KPOS* = 3, with reference frame Z-rotated by  $\alpha_Z/2$  as drawn from  $L = 2 \frac{BORO}{B1} \sin(\alpha_Z/2)$ .

<sup>20</sup>For the Geek: Installing misalignments in additional magnets is doable, in the present state of development it involves *errors.f*, *mulerr.f*, *rerror.f*, *toserr.f*, including X-, Y-, Z-offsets and X-, Y-, Z-rotation.



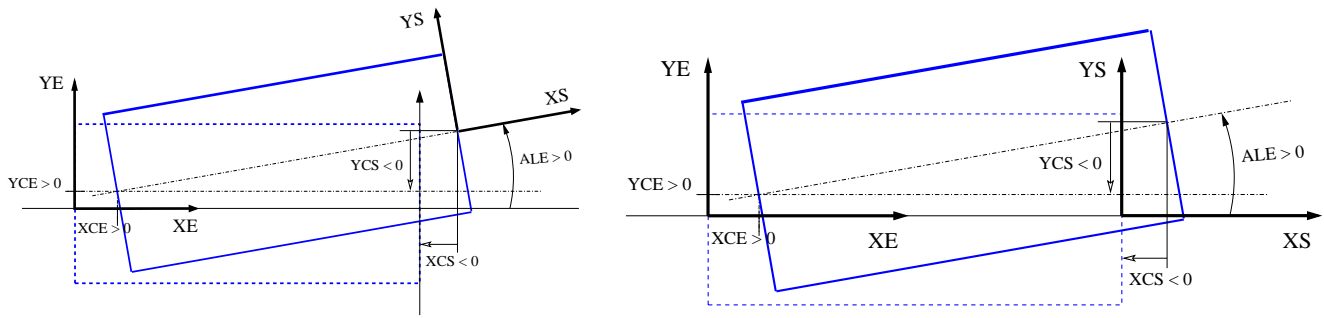


Figure 50: Case of Cartesian frame optical element. Left : positioning an optical element using  $KPOS= 1$ . Right : Mis-aligning an optical element using  $KPOS= 2$ .  $(X_E, Y_E)$  and  $(X_S, Y_S)$  are respectively the incoming and outgoing reference frames.

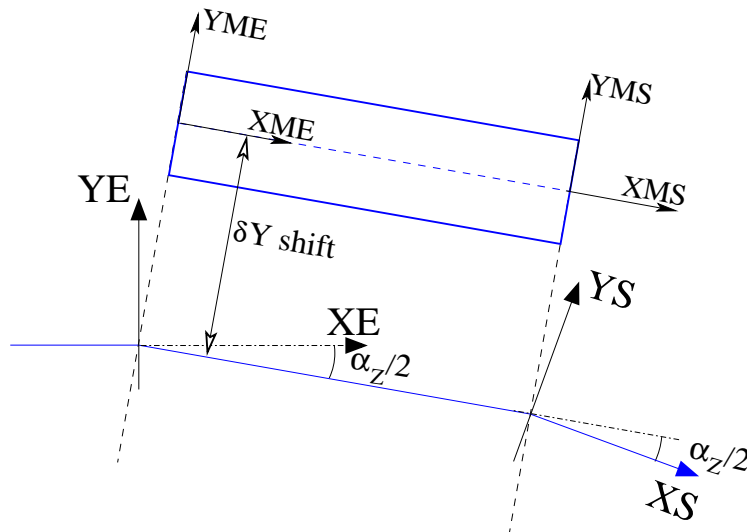


Figure 51: Alignment in *BEND* by half the deviation,  $\alpha_Z/2$ , using  $KPOS= 3$ .  $(X_E, Y_E)$  and  $(X_S, Y_S)$  are respectively the incoming and outgoing reference frames. Trajectory coordinates inside the magnet are defined in the magnet frame,  $(X_{ME}, Y_{ME})$ .

However it provides additional magnet mis-alignments (alignment errors) (Fig. 52) : Z-rotation  $\theta_Z$ , Z-shift  $\delta Z$ , Y-rotation  $\theta_Y$ . Rotations apply at the center of the magnet.

Note that some of these features may not be properly operational due to on-going developments: check before use.

$KPOS = 5$  is UNDER DEVELOPMENT. It is similar to  $KPOS = 4$ , with a different ordering of the offset data list. It allows X-, Y- and Z-offset, as well as X-, Y- and Z-rotation in that order. Note that some of these features may not be properly operational due to on-going developments: check before use.

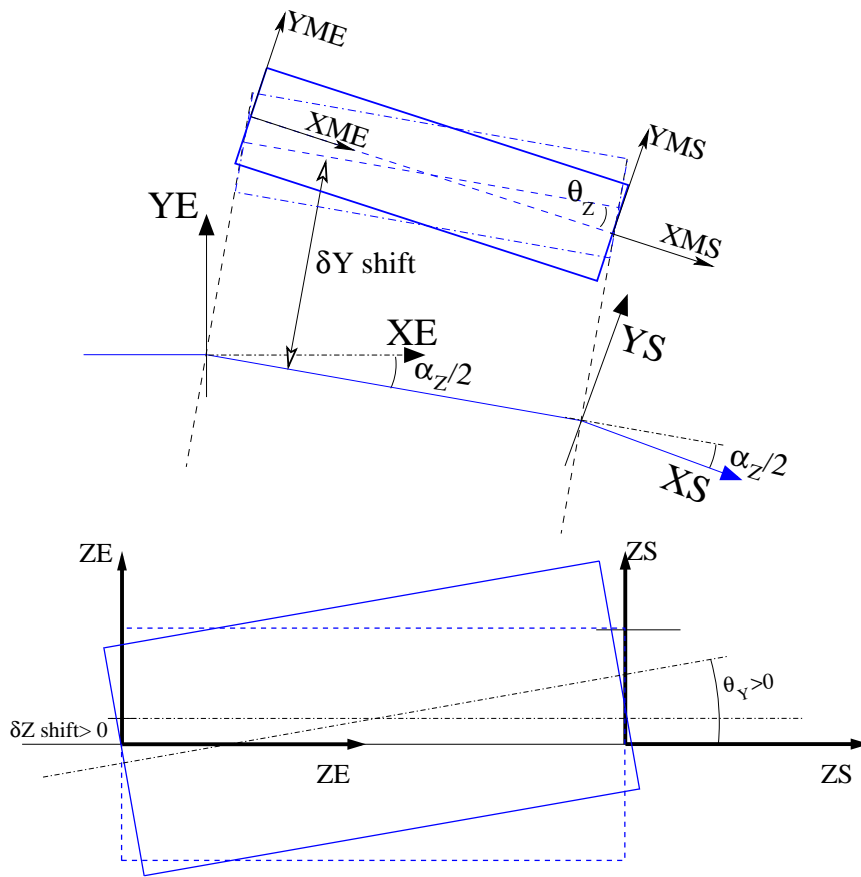


Figure 52: Additional magnet mis-alignment with  $KPOS=4$ .

### Polar Coordinates

If  $KPOS = 1$ , the element is positioned automatically in such a way that a particle entering with zero initial coordinates and  $1 + DP = B\rho/BORO$  relative rigidity will reach position  $(RM, \frac{AT}{2})$  in the element with  $T = 0$  angle with respect to the moving frame in the polar coordinates system of the element ; see *DIPOLE-M* and *POLARMES*).

If  $KPOS = 2$ , the map is positioned in such a way that the incoming reference frame is presented at radius  $RE$  with angle  $TE$ . The exit reference frame is positioned in a similar way with respect to the map, by means of the two parameters  $RS$  (radius) and  $TS$  (angle) (see Fig. 20 page 99).

### 7.11 Coded Integration Step

In several optical elements (*e.g.*, all multipoles, *BEND*) the integration step (in general noted *XPAS*) can be coded under the form  $XPAS = \#E | C | S$ , with  $E, C, S$  integers,  $E$  is the number of steps in the entrance fringe field region,  $C$  is the number of steps in the magnet body, and  $S$  is the number of steps in the exit fringe field region.

### 7.12 Ray-tracing of an Arbitrarily Large Number of Particles

Monte Carlo multi-particle simulations involving an arbitrarily large number of particles can be performed using *MCOBJET* together with *REBELOTE*, put at the end of the optical sequence, with its argument *NPASS* being the number of passes through *REBELOTE*, and  $(NPASS+1) * IMAX$  the number of particles to be ray-traced. In order that new initial conditions ( $D, Y, T, Z, P, X$ ) be generated at each pass,  $K = 0$  has to be specified in *REBELOTE*.

Statistics on coordinates, spins, and other histograms can be performed by means of such procedures as *HISTO*, *SPNTRK*, etc. that stack the information from pass to pass.

### 7.13 Stopped Particles : The *IEX* Flag

As described in *OBJET*, each particle  $I = 1, IMAX$  is attached a value  $IEX(I)$  of the *IEX* flag. Normally,  $IEX(I) = 1$ . Under certain circumstances, *IEX* may be changed to a negative value by **zgoubi**, as follows.

- 1 : the trajectory happened to wander outside the limits of a field map
- 2 : too many integration steps in an optical element (a quantity controlled in `MXSTEP.H` include file)
- 3 : deviation happened to exceed  $\pi/2$  in an optical element not designed to allow that
- 4 : stopped by walls (procedures *CHAMBR*, *COLLIMA*)
- 5 : too many iterations in subroutine `depl.a.f`
- 6 : energy loss exceeds particle energy
- 7 : field discontinuities larger than 50% within a field map
- 8 : reached field limit in an optical element
- 9 : particle not accounted for
- 10 : secondary particle has decayed (whatever that means...)
- 11 : particle moves backward upon decay

Only in the case  $IEX = -1$  will the integration not be stopped since in this case the field outside the map is extrapolated from the map data, and the particle may possibly get back into the map (see section 1.4.2 on page 32). In all other cases the particle will be stopped.

### 7.14 Negative Rigidity

**zgoubi** can handle negative rigidities  $B\rho = p/q$ . This is equivalent to considering either particles with negative charge ( $q < 0$ ) or momentum ( $p < 0$ ), or reversed fields (*wrt.* the field sign that shows in the `zgoubi.dat` optical element data list).

Negative rigidities may be specified in terms of  $BORO < 0$  or  $D = B\rho/BORO < 0$  when defining the initial coordinates with *OBJET* or *MCOBJET*.

### 7.15 Fiel maps

Field maps read by dedicated keywords as *TOSCA*, *CARTEMES*, *POLARMES*, etc., may have, and in some cases *must have* a header. The header is a series of comment lines, say  $N$  lines. The default value is  $N=8$ . If a field map with  $N \neq 8$  is to be used, this need be specified in the field map title, as part of the "TITL" comment (see *TOSCA* or other field map keyword about TITL) under the form 'HEADER\_N' with  $N \geq 0$  placed anywhere within the TITL comment area.

## 8 ZGOUBI OUTPUT FILES

Running zgoubi always creates a default zgoubi.res file. The name can be changed by resorting to the argument `-out` (or `-fileOut` or `>`). In the same manner that the input data file name can be changed (the default is `zgoubi.dat`) by means of `-in` (or `-fileIn` or `<`).

Additional output files can be obtained, upon request:

- `zgoubi.plt` (using  $IL=2$  in one or more optical elements);
- `zgoubi.impdev.out` (using  $IL=7$  in one or more optical elements);
- a copy of current zgoubi executable (launch zgoubi execution with argument `-saveExec`);
- a copy of current zpop executable (launch zgoubi execution with argument `-saveZpop`: this guarantees availability of a version of zpop compatible with the current formatting of zgoubi outputs which zpop reads).

Some keywords produce, by default, one or more specific output files, in principle in order to allow subsequent user's data treatment/analysis, namely,

file	KEYWORD[OPTION]	comment
<code>zgoubi.consty.out</code>	<i>OPTIONS[CONSTY]</i>	
<code>zgoubi.fai</code>	<i>FAISCNL</i> or <i>FAISTORE</i>	
<code>zgoubi.FITVALS.out</code>	<i>FIT[SAVE]</i>	name can be different, if specified
<code>zgoubi.FIT.out.dat</code>	<i>FIT</i>	a copy of <code>zgoubi.dat</code> sequence with updated variables values once FIT is over. Default name, can be changed
<code>zgoubi.IDLUNI.out</code>	<i>OPTIONS[MONITOR]</i>	
<code>zgoubi.SVD_Amatrix.out</code>	<i>SVDOC</i>	contains $A^{-1}$ and $A$ matrices, as computed by SVDOC
<code>zgoubi.SVD_PuCoList.out</code>	<i>SVDOC</i>	
<code>zgoubi.SVD.out.dat</code>	<i>SVDOC</i>	a copy of <code>zgoubi.dat</code> sequence with updated corrector values as found by SVDOC. Default name, can be changed
<code>zgoubi.SVDOrbits.out</code>	<i>SVDOC</i>	Pickup records, and corrector strengths
<code>zgoubi.TWISS.out</code>	<i>TWISS</i>	has similar content formatting to that of the MADX twiss command output file

- in several keyword cases, a listing specific of the computations undergone in that keyword can be obtained by means of a `PRINT` comment, or by means of a specific argument in their data list. The idea behind that feature is to allow checks or some level of debugging, and anyway these outputs can readily be used for plotting, with gnuplot for instance (like, particle RF phase, or boost, versus turn number from `zgoubi.CAVITE.Out`, histograms from `zgoubi.HISTO.out`, etc., *confer* next sections). The output files name is of the form `zgoubi.XXX.out` or `zgoubi.XXX.Out`, "XXX" is in general the keyword itself, or close. Here is a list as of May 2020 (may not be exhaustive, a `'grep "zgoubi.*.out" [pathTo]/zgoubi/*f'` or `'grep "zgoubi.*.Out" [pathTo]/zgoubi/*f'` would tell):

file	KEYWORD[OPTION]	comment
zgoubi.CAVITE.Out	<i>CAVITE</i> [PRINT]	
zgoubi.COLLIMA.out	<i>COLLIMA</i> [IA=2]	
zgoubi.ERRORS.out	<i>ERRORS</i> [PRINT]	
zgoubi.HISTO.out	<i>HISTO</i> [PRINT]	
zgoubi.MATRIX.out	<i>MATRIX</i> [PRINT]	
zgoubi.OPTICS.out	<i>OPTICS</i> [PRINT]	
zgoubi.PICKUP.out	<i>PICKUPS</i> [PRINT]	
zgoubi.SCALING.Out	<i>SCALING</i> [PRINT]	
zgoubi.SPNPRT.Out	<i>SPNPRT</i> [PRINT]	
zgoubi.SRLOSS.Out	<i>SRLOSS</i> [PRINT]	

### 8.1 zgoubi.res

Running zgoubi always creates a default zgoubi.res file. This is a listing of the program execution, the listing follows the zgoubi.dat sequence ... in sequence!

It is recommended to keep an eye in zgoubi.res when running a problem, it contains a lot of informations regarding what happened, numerical results, etc. That's the first place to lookup in case of a crash.

## 8.2 zgoubi.fai

Table 3: A list of the 53 parameters characterizing a particle and the current status of its ray tracing, as printed out to zgoubi.fai when the execution pointer meets *FAISTORE* or *FASICNL*. Zgoubi.fai displays these 46 parameters under the form of as many columns, one line per particle. The particle coordinate frame is the (O;X,Y,Z) Cartesian frame defined in Fig. 1, including T and P angles therein.

column number	parameter name	unit or nature	what's that
1	KEX	int	IEX flag. Set to 1 by <i>[MC]OBJET</i> . Set to negative value when particle has an accident
2	Do-1	float	$(B\rho - BORO)/BORO$ , at <i>[MC]OBJET</i> ( <i>BORO</i> : see 37)
3	Yo	cm	horizontal position, at <i>[MC]OBJET</i>
4	To	mrd	horizontal angle, at <i>[MC]OBJET</i>
5	Zo	cm	vertical position, at <i>[MC]OBJET</i>
6	Po	mrd	vertical angle, at <i>[MC]OBJET</i>
7	So	cm	longitudinal distance, at <i>[MC]OBJET</i>
8	to	$\mu s$	time, at <i>[MC]OBJET</i>
9	D-1	float	$(B\rho - BORO)/BORO$ , current
10	Y	cm	horizontal position, current
11	T	mrd	horizontal angle, current
12	Z	cm	vertical position, current
13	P	mrd	vertical angle, current
14	S	cm	longitudinal distance, current
15	t	$\mu s$	time, current
16	SXo	float	X spin component, at genesis by <i>SPNTRK</i>
17	SYo	float	Y spin component, at genesis by <i>SPNTRK</i>
18	SZo	float	Z spin component, at genesis by <i>SPNTRK</i>
19	modSo	float	spin vector modulus, at genesis by <i>SPNTRK</i>
20	SX	float	X spin component, current
21	SY	float	Y spin component, current
22	SZ	float	Z spin component, current
23	modS	float	spin vector modulus, current
24	ENEKI	MeV	kinetic energy
25	ENERG	MeV	total energy
26	IT	int	particle number
27	IREP	int	flag - more or less obsolete
28	SORT	cm	distance at which particle got lost
29	M	MeV/c2	mass
30	Q	C	charge
31	G	float	anomalous magnetic moment
32	tau	float	life time
33	unused	float	
34	RET	float	$\phi - \phi_{synch.}, \phi_{synch.}$ = synchronous phase. Set by <i>CAVITE</i>
35	DPR	float	$(p - p_{synch.})/p_{synch.}, p_{synch.}$ = synchronous momentum. Set by <i>CAVITE</i>
36	PS	float	synchronous momentum, possibly subject to updating, by <i>CAVITE</i> for instance
37	BORO	kG.cm	$BORO \equiv$ reference $B\rho$ , set by <i>[MC]OBJET</i>
38	IPASS	int	pass number (incremented by <i>REBELOTE</i> )
39	NOEL	int	element number in the zgoubi.dat sequence (as shown in zgoubi.res)
40	KLEY	string	previous keyword
41	LABEL1	string	label 1 of the previous keyword
42	LABEL2	string	label 2 of the previous keyword
43	LET	string	particle tag
44	SRloss	MeV	cumulated synchrotron radiation energy loss
45	DPREF-[DPREF]	MeV	fractional part of the current synchronous momentum $p_{synch.}$ (see 36)
46	[DPREF]	MeV	integer part of the current synchronous momentum $p_{synch.}$ (see 36)
47	DATE		date
48	TIME		time
49	NI	int.	
50	MXITR	int.	
51	NBFT	int.	
52	KBYD	0-1	1 if this print out occurred beyond FIT keyword
53	LAST	0-1	1 if this print out occurred with updated variable values following FIT completion

Examples;

- horizontal phase space (analogous to  $(x, x')$  in matrix transport notations): (Y,T), *i.e.*, columns (10,11),
- vertical phase space (analogous to  $(y, y')$  in matrix transport notations): (Y,T), *i.e.*, columns (12,13),
- longitudinal phase space (analogous to  $(\delta\phi, \delta p/p)$  in matrix transport notations): (RET,DPR), *i.e.*, columns (34,35),
- position as a function of distance along a beam line, or turn by turn around a ring: horizontal Y(S), *i.e.*, columns (10,14); vertical Z(S), *i.e.*, columns (12,14).

### 8.3 zgoubi.plt

Table 4: A list of the 52 parameters characterizing a particle and the current status of its ray tracing, during the step by step integration across an optical element, as printed out to zgoubi.plt when the execution pointer meets  $IL=2$ . Particles are pushed one by one all the way through, so, if it takes N steps for particle number 1 to cross, then the N first lines all concern particle 1, and so on. Zgoubi.plt displays these 50 parameters under the form of as many columns, one line at each integration step. The particle coordinate frame (O;X,Y,Z) is (i) either Cartesian as sketched in Figs. 1, 53 (case of optical elements defined in a Cartesian frame), (ii) or cylindrical, and in that case with Y the radius and X the polar angle (case of optical elements defined in a cylindrical frame, Fig. 53).

column number	parameter name	unit or nature	what's that
1	KEX	int	IEX flag. Set to 1 by [MC]OBJET. Set to negative value when particle has an accident
2	Do-1	float	$(B\rho - BORO)/BORO$ , at [MC]OBJET (BORO: see 37)
3	Yo	cm	horizontal position, at [MC]OBJET
4	To	mrd	horizontal angle, at [MC]OBJET
5	Zo	cm	vertical position, at [MC]OBJET
6	Po	mrd	vertical angle, at [MC]OBJET
7	So	cm	longitudinal distance, at [MC]OBJET
8	to	$\mu s$	time, at [MC]OBJET
9	D-1	float	$(B\rho - BORO)/BORO$ , current
10	Y-DY	cm	transverse horizontal coordinate if Cartesian frame, radius if cylindrical. Normally DY=0
11	T	mrd	horizontal angle (in the moving frame if cylindrical), current
12	Z	cm	vertical position, current
13	P	mrd	vertical angle, current
14	S	cm	longitudinal distance, current
15	time	$\mu s$	time, current
16	beta	-	v/c
17	DS	cm	step size
18	KART	int	current frame either Cartesian (KART=1) or cylindrical (KART=2)
19	IT	int	particle number
20	IREP	int	flag - more or less obsolete
21	SORT	cm	distance at which particle got lost
22	X	cm	X coordinate if Cartesian frame, polar angle if cylindrical (positive clockwise) (Fig. 53)
23	BX	kG	X component of magnetic field, current
24	BY	kG	Y component of magnetic field, current
25	BZ	kG	Z component of magnetic field, current
26	RET	float	$\phi - \phi_{synch.}, phi_{synch.}$ = synchronous phase. Set by CAVITE
27	DPR	float	$(p - p_{synch.})/p_{synch.}, p_{synch.}$ = synchronous momentum. Set by CAVITE
28	PS	float	synchronous momentum, possibly subject to updating, by CAVITE for instance
29	SXo	float	X spin component, at genesis by SPNTRK
30	SYo	float	Y spin component, at genesis by SPNTRK
31	SZo	float	Z spin component, at genesis by SPNTRK
32	modSo	float	spin vector modulus, at genesis by SPNTRK
33	SX	float	X spin component, current
34	SY	float	Y spin component, current
35	SZ	float	Z spin component, current
36	modS	float	spin vector modulus, current
37	EX	V/m	X component of electric field, current
38	EY	V/m	Y component of electric field, current
39	EZ	V/m	Z component of electric field, current
40	BORO	kG.cm	$BORO \equiv$ reference $B\rho$ , set by [MC]OBJET
41	IPASS	int	pass number (incremented by REBELOTE)
42	NOEL	int	element number in the zgoubi.dat sequence (as shown in zgoubi.res)
43	KLEY	string	current keyword
44	LABEL1	string	label 1 of the current keyword
45	LABEL2	string	label 2 of the current keyword
46	LET	string	particle tag
47	NITR	int	fit procedure: iteration number
48	MXIT	int	fit procedure: max. allowed number of iterations
49	FIT#	int	fit procedure: number
50	FITBYD,	0-1	fit procedure: final pass after FIT, 0/1 (Y/N)
51	FITLST	0-1	fit procedure: print out occurred once FIT was completed, 0/1 (Y/N)
52	KFLD	1-3	nature of current field: 1/2/3 [B/E/E+B]

Examples;

- vertical magnetic field component as a function of distance: BZ(S), columns (25,14),
- vertical position as a function of turn number: Z(IPASS), columns (12,41),
- vertical position as a function of turn distance: Z(S), columns (12,14).
- horizontal position (or radial excursion R-RM) as a function of optical element number in zgoubi.dat sequence: Y(NOEL), columns (10,42).



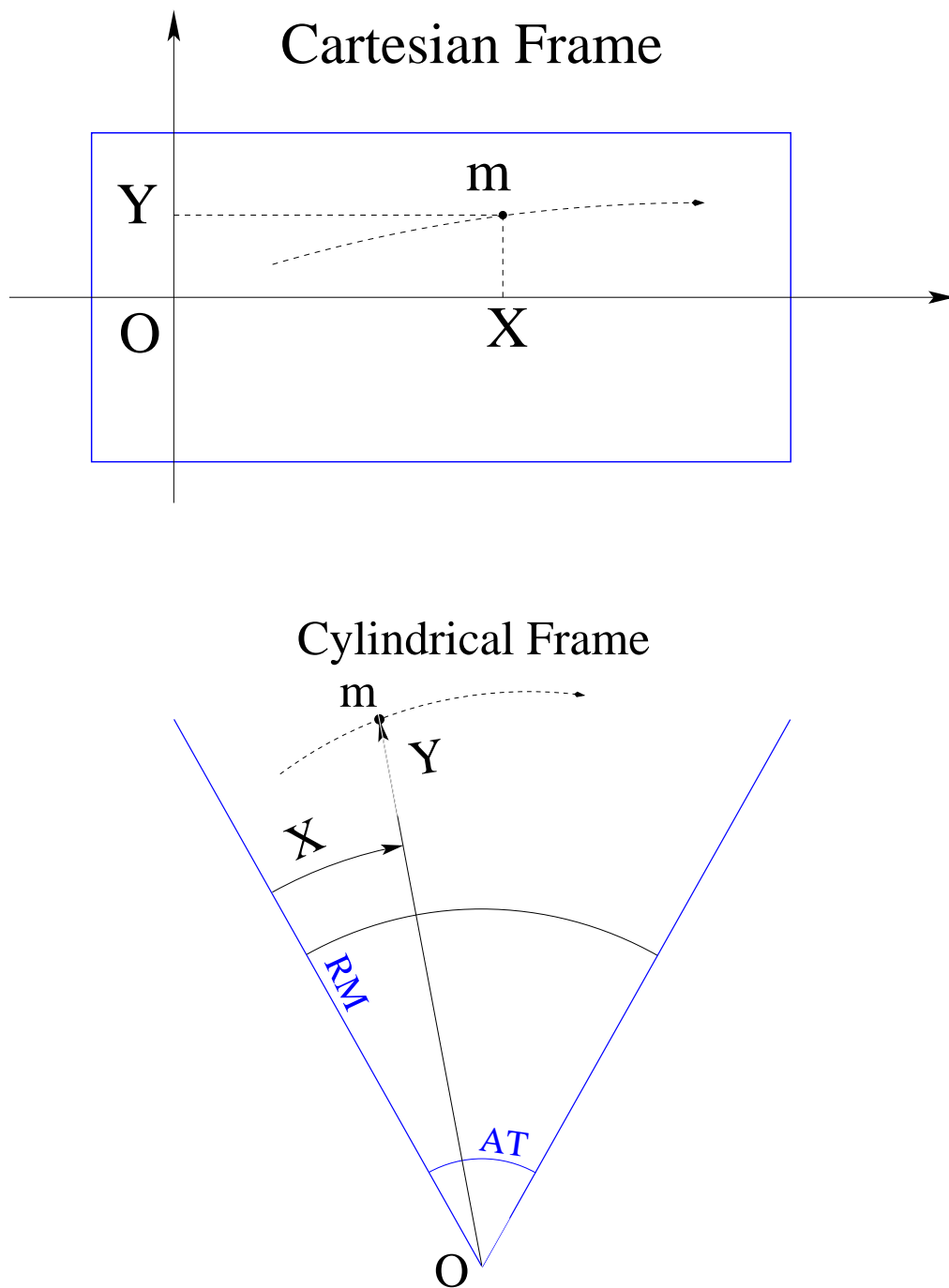


Figure 53: Coordinates, in `zgoubi.plt`, of a particle at location  $M$  along its trajectory in a  $(O; X, Y, Z)$  frame;  $Z$  is normal to the  $(X, Y)$  plane, pointing out of the page. In these graphs, the dashed curve is the projection of the trajectory in the  $(X, Y)$  plane,  $m$  is the projection of  $M$ . Top: case of an optical element (figured as a rectangular box) in which the field is defined in a Cartesian coordinate system (case for instance of *MULTIPOL*, *BEND*, *TOSCA*[ $\text{MOD} \leq 19$ ]). Bottom: case of an optical element (figured as an angular sector  $AT$ , with a reference radius  $RM$ ) in which the field is defined in a cylindrical coordinate system (case of *DIPOLE* for instance, Fig. 20); in that case  $X$  denotes the polar angle, counted positive clockwise

## 8.4 zgoubi.TWISS.out

Table 5: A list of the parameters characterizing the optics at the exit of optical elements, as produced by *TWISS* keyword.

column number	parameter name	unit or nature	what's that
1	alfx		next 6 cols.: Y, Z and s optical functions
2	btx		
3	alfy		
4	bty		
5	alfl		
6	btl		
7	Dx		next 4 cols.: dispersion function
8	Dxp		
9	Dy		
10	Dyp		
11	phix/2pi		Y-motion phase advances
12	phiy/2pi		Z-motion phase advances
13	sum_s		cumulative length of optical elements (erroneous if field maps used)
14	#lmnt		element number
15	x		Y orbit
16	xp		T orbit angle
17	y		Z orbit
18	yp		P orbit angle
19	KEYWORD		
20	label1		
21	label2		
22	FO(6,1)		initial (at [MC]OBJET) path distance of particle 1
23	K0*L		integrated field of preceding optical element
24	K1*L		integrated K1 of preceding optical element
25	K2*L		integrated K2 of preceding optical element
26	—C—		coupling coefficient
27	r		coupling coefficient
28	!		
29	optimp.f		
30	IPASS		pass number
31	DPREF		integer part of reference dp/p (=PS / P0 -HDPRF)
32	HDPREF		fractional part of reference dp/p (= INT(PS / P0) )
33	R11		next 13 cols.: first order transport coefficients, from begining of the sequence
34	R12		
35	R21		
36	R22		
37	R33		
38	R34		
39	R43		
40	R44		
41	R51		
42	R52		
43	R53		
44	R54		
45	R56		
46	F61		path distance of particle 1
47	SX		X spin components
48	SY		Y spin components
49—	SZ		Z spin components
50	—S—		spin vector modulus
51	Length		length of preceding optical element
52	Deviation		orbital deviation by preceding optical element

Examples;

- vertical magnetic field component as a function of distance: BZ(S), columns (25,14),
- vertical position as a function of turn number: Z(IPASS), columns (12,41),
- vertical position as a function of turn distance: Z(S), columns (12,14).
- horizontal position (or radial excursion R-RM) as a function of optical element number in zgoubi.dat sequence: Y(NOEL), columns (10,42).

## **PART B**

### **Keywords and input data formatting**



## Glossary of Keywords, Part B (page 9, PART A)

---

### Keywords : how to format their input data list

<b>AGSMM</b>	AGS main magnet .....	237
<b>AGSQUAD</b>	AGS quadrupole .....	238
<b>AIMANT</b>	Generation of dipole mid-plane 2-D map, polar frame .....	239
<b>AUTOREF</b>	Transport beam into a new reference frame .....	243
<b>BEAMBEAM</b>	Beam-beam lens .....	244
<b>BEND</b>	Bending magnet, Cartesian frame .....	245
<b>BETATRON</b>	Betatron core (momentum kick) .....	247
<b>BINARY</b>	<i>BINARY</i> ↔ <i>FORMATTED</i> data converter .....	248
<b>BREVOL</b>	1-D uniform mesh magnetic field map .....	249
<b>CARTEMES</b>	2-D Cartesian uniform mesh magnetic field map .....	250
<b>CAVITE</b>	Accelerating cavity .....	252
<b>CHAMBR</b>	Long transverse aperture limitation .....	254
<b>CHANGREF</b>	Transformation to a new reference frame .....	255
<b>CIBLE</b>	Generate a secondary beam following target interaction .....	256
<b>COLLIMA</b>	Collimator .....	257
<b>CYCLOTRON</b>	Spiral setor cyclotron .....	258
<b>DECAPOLE</b>	Decapole magnet .....	261
<b>DIPOLE</b>	Dipole magnet, polar frame .....	262
<b>DIPOLE-M</b>	Generation of dipole mid-plane 2-D map, polar frame .....	264
<b>DIPOLES</b>	Dipole magnet <i>N</i> -tuple, polar frame .....	266
<b>DODECAPO</b>	Dodecapole magnet .....	267
<b>DRIFT</b>	Field free drift space .....	268
<b>EBMULT</b>	Electro-magnetic multipole .....	269
<b>EL2TUB</b>	Two-tube electrostatic lens .....	271
<b>ELCYLDEF</b>	Electrostatic cylindrical deflector .....	272
<b>ELMIR</b>	Electrostatic N-electrode mirror/lens, straight slits .....	273
<b>ELMIRC</b>	Electrostatic N-electrode mirror/lens, circular slits .....	274
<b>ELMULT</b>	Electric multipole .....	275
<b>ELREVOL</b>	1-D uniform mesh electric field map .....	276
<b>EMMA</b>	2-D Cartesian or cylindrical mesh field map for EMMA FFAG .....	277
<b>END</b>	End of input data list .....	282
<b>ESL</b>	Field free drift space .....	268
<b>ERRORS</b>	Injecting errors in optical elements .....	278
<b>FAISCEAU</b>	Print particle coordinates .....	279
<b>FAISCNL</b>	Store particle coordinates in file FNAME .....	279
<b>FAISTORE</b>	Store coordinates every <i>IP</i> other pass at labeled elements .....	279
<b>FINISH</b>	End of execution .....	283
<b>FFAG</b>	FFAG magnet, <i>N</i> -tuple .....	280
<b>FFAG-SPI</b>	Spiral FFAG magnet, <i>N</i> -tuple .....	281
<b>FIN</b>	End of input data list .....	282
<b>FIT,FIT2</b>	Fitting procedure .....	284
<b>FOCALE</b>	Particle coordinates and horizontal beam size at distance <i>XL</i> .....	289
<b>FOCALEZ</b>	Particle coordinates and vertical beam size at distance <i>XL</i> .....	289
<b>GASCAT</b>	Gas scattering .....	290
<b>GETFITVAL</b>	Get values of <i>variables</i> as saved from former FIT[2] run .....	291

<b>GOTO</b>	Branching statement	294
<b>HALBACH</b>	Halbach style permanent magnet	294
<b>HELIX</b>	Helical dipole	294
<b>HISTO</b>	1-D histogram	295
<b>IMAGE</b>	Localization and size of horizontal waist	296
<b>IMAGES</b>	Localization and size of horizontal waists	296
<b>IMAGESZ</b>	Localization and size of vertical waists	296
<b>IMAGEZ</b>	Localization and size of vertical waist	296
<b>INCLUDE</b>	File include statement	297
<b>MAP2D</b>	2-D Cartesian uniform mesh field map - arbitrary magnetic field	299
<b>MAP2D-E</b>	2-D Cartesian uniform mesh field map - arbitrary electric field	300
<b>MARKER</b>	Marker	301
<b>MATRIX</b>	Calculation of transfer coefficients, periodic parameters	302
<b>MCDESINT</b>	Monte-Carlo simulation of in-flight decay	303
<b>MCOBJET</b>	Monte-Carlo generation of a 6-D object	304
<b>MULTIPOL</b>	Magnetic multipole	307
<b>OBJET</b>	Generation of an object	308
<b>OBJETA</b>	Object from Monte-Carlo simulation of decay reaction	311
<b>OCTUPOLE</b>	Octupole magnet	312
<b>OPTICS</b>	Write out optical functions. Log to zgoubi.OPTICS.out	313
<b>OPTIONS</b>	Global or special options	314
<b>ORDRE</b>	Taylor expansions order	316
<b>PARTICUL</b>	Particle characteristics	317
<b>PICKUPS</b>	Bunch centroid path; orbit	318
<b>PLOTDATA</b>	Intermediate output for the PLOTDATA graphic software	319
<b>POISSON</b>	Read magnetic field data from <i>POISSON</i> output	320
<b>POLARMES</b>	2-D polar mesh magnetic field map	321
<b>PS170</b>	Simulation of a round shape dipole magnet	322
<b>QUADISEX</b>	Sharp edge magnetic multipoles	323
<b>QUADRUPO</b>	Quadrupole magnet	324
<b>REBELOTE</b>	'Do it again'	326
<b>RESET</b>	Reset counters and flags	328
<b>SCALING</b>	Power supplies and R.F. function generator	329
<b>SEPARA</b>	Wien Filter - analytical simulation	330
<b>SEXQUAD</b>	Sharp edge magnetic multipole	331
<b>SEXTUPOL</b>	Sextupole magnet	332
<b>SOLENOID</b>	Solenoid	333
<b>SPACECHARG</b>	Space charge correction kick	334
<b>SPINR</b>	Spin rotation	335
<b>SPNPRNL</b>	Store spin coordinates into file FNAME	336
<b>SPNSTORE</b>	Store spin coordinates every <i>IP</i> other pass at labeled elements	336
<b>SPNPRT</b>	Print spin coordinates	337
<b>SPNTRK</b>	Spin tracking switch, and spin coordinate initialization	338
<b>SRLOSS</b>	Synchrotron radiation energy loss	339
<b>SRPRNT</b>	Print SR loss statistics	340
<b>SVDOC</b>	SVD orbit correction	341
<b>SYNRAD</b>	Synchrotron radiation spectral-angular densities	343
<b>SYSTEM</b>	System call	344
<b>TARGET</b>	Generate a secondary beam following target interaction	256
<b>TOSCA</b>	2-D and 3-D Cartesian or cylindrical mesh field map	346
<b>TRANSMAT</b>	Matrix transfer	349
<b>TRAROT</b>	Translation-Rotation of the reference frame	350
<b>TWISS</b>	Compute periodic optical parameters. Log to zgoubi.TWISS.out	351

<b>UNDULATOR</b>	Undulator magnet .....	353
<b>UNIPOT</b>	Unipotential cylindrical electrostatic lens .....	354
<b>VENUS</b>	Simulation of a rectangular shape dipole magnet .....	355
<b>VFFA</b>	Simulation of a VFFA dipole, or series of .....	356
<b>WIENFILT</b>	Wien filter .....	357
<b>YMY</b>	Reverse signs of $Y$ and $Z$ reference axes .....	358

## Optical Elements Versus Keywords

### Which optical component can be simulated Which keyword(s) can be used for that purpose

This glossary gives a list of keywords suitable for the simulation of common optical elements. These are classified in three categories: magnetic, electric and combined electro-magnetic elements.

Field map procedures are also listed; they provide a means for ray-tracing through measured or simulated electric and/or magnetic fields.

#### MAGNETIC ELEMENTS

AGS main magnet	AGSMM
Betatron core	BETATRON, CAVITE
Cyclotron magnet	CYCLOTRON, DIPOLE[S], DIPOLE-M, FFAG, FFAG-SPI
Decapole	DECAPOLE, MULTIPOL
Dipole[s], spectrometer dipole	AIMANT, BEND, DIPOLE[S][-M], MULTIPOL, QUADISEX
Dodecapole	DODECAPO, MULTIPOL
FFAG magnets	DIPOLAS, FFAG, FFAG-SPI, MULTIPOL
Helical dipole	HELIX
Multipole	MULTIPOL, QUADISEX, SEXQUAD
Octupole	OCTUPOLE, MULTIPOL, QUADISEX, SEXQUAD
Quadrupole	QUADRUPO, MULTIPOL, SEXQUAD, AGSQUAD
Sextupole	SEXTUPOL, MULTIPOL, QUADISEX, SEXQUAD
Skew multipoles	MULTIPOL
Solenoid	SOLENOID
Undulator	UNDULATOR

#### Using field maps

1-D, cylindrical symmetry	BREVOL
2-D, mid-plane symmetry	CARTEMES, POISSON, TOSCA
2-D, no symmetry	MAP2D
2-D, polar mesh, mid-plane symmetry	POLARMES
3-D, no symmetry	TOSCA
EMMA FFAG quadrupole doublet	EMMA
linear composition of field maps	TOSCA

#### ELECTRIC ELEMENTS

2-tube (bipotential) lens	EL2TUB
3-tube (unipotential) lens	UNIPOT
Decapole	ELMULT
Dipole	ELMULT
Dodecapole	ELMULT
Multipole	ELMULT
N-electrode mirror/lens, straight slits	ELMIR
N-electrode mirror/lens, circular slits	ELMIRC
Octupole	ELMULT
Quadrupole	ELMULT
R.F. (kick) cavity	CAVITE
Sextupole	ELMULT
Skew multipoles	ELMULT

#### Using field maps

1D, cylindrical symmetry	ELREVOL
--------------------------	---------



2-D, no symmetry

MAP2D-E

**ELECTRO-MAGNETIC ELEMENTS**

Decapole	EBMULT
Dipole	EBMULT
Dodecapole	EBMULT
Multipole	EBMULT
Octupole	EBMULT
Quadrupole	EBMULT
Sextupole	EBMULT
Skew multipoles	EBMULT
Wien filter	SEPARA, WIENFILT



## INTRODUCTION

Here after is given a detailed description of input data formatting and units. All available keywords appear in alphabetical order.

Keywords are read from the input data file by an unformatted *FORTTRAN READ* statement. They be enclosed between quotes (e.g., *'DIPOLE'*).

Text string data such as comments or file names, are read by formatted *READ* statements, no quotes should be used in that case.

Numerical variables and indices are read by unformatted *READ*. It may therefore be necessary that integer variables be assigned an integer value.

In the following tables

- the first column shows the expected input parameters (actually, their *values* are expected), indices and text strings,
- the second column gives brief comments regarding their meaning and use,
- the third column gives the units or ranges,
- the fourth column indicates whether the expected parameter types are integer (I), real (E) or text string (A). For example, "I, 3\*E" means that one integer followed by 3 reals is expected. "A80" means that a text string of maximum 80 characters is expected.



<b>AGSMM</b>	<b>AGS main magnet</b>	(page 69 in PART A)	
<i>IL</i>	$IL = 1, 2[\times 10^n]$ , 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2 $[\times 10^n]$ , 7	I
<i>MOD, dL, R<sub>0</sub>, dB1, dB2, dB3</i>	Type of magnet model <sup>1</sup> ; unused ; pole tip radius, 10 cm if set to zero ; relative error on dipole, quadrupole, sextupole component.	2*no dim, cm, 3*no dim	I[.I], 5*E
<i>NBLW[.MOD], NBLW times : NW, I</i>	Number of back-leg windings [back-leg winding model <sup>2</sup> ] ; for each back-leg winding : number of windings, current.	≤2, <i>NBLW</i> × (any, Amp.)	I, <i>NBLW</i> × (I, E)
<i>X<sub>E</sub>, λ<sub>E</sub>, E<sub>2</sub>, E<sub>3</sub></i>	<b>Entrance face</b> Integration zone ; fringe field extent : dipole fringe field extent = λ <sub>E</sub> ; quadrupole fringe field extent = λ <sub>E</sub> * E <sub>2</sub> ; sextupole fringe field extent = λ <sub>E</sub> * E <sub>3</sub> (sharp edge if field extent is zero)	2*cm, 2*no dim	4*E
<i>NCE, C<sub>0</sub> – C<sub>5</sub></i>	same as <i>QUADRUPO</i>	0-6, 6*no dim	I, 6*E
<i>X<sub>S</sub>, λ<sub>S</sub>, S<sub>2</sub>, S<sub>3</sub></i>	<b>Exit face</b> Integration zone ; as for entrance	2*cm, 2*no dim	4*E
<i>NCS, C<sub>0</sub> – C<sub>5</sub></i>		0-6, 6*no dim	I, 6*E
<i>R1, R2, R3</i>	Skew angles of field components	3*rad	10*E
<i>XPAS</i>	Integration step	cm	E
<b>If KPOS = 1-3</b> <i>KPOS, XCE, YCE, ALE</i>	Positioning as follows : <i>KPOS</i> =1 : element aligned ; <i>KPOS</i> =2 : misaligned ; shifts, tilt ; <i>KPOS</i> =3 : effective only if <i>B1</i> ≠ 0 : entrance and exit frames are tilted wrt. the magnet by an angle of • either <i>ALE</i> if <i>ALE</i> ≠0 • or 2 Arcsin( <i>B1 XL</i> / 2 <i>BORO</i> ) if <i>ALE</i> =0	1-3, 2*cm, rad	I, 3*E
<b>If KPOS = 4</b> <i>KPOS, XS, YS, ZR, ZS, YR</i>	X-, Y-shift, Z-rotation, Z-shift, Y-rotation. Automatic ZR-positioning (as for <i>ALE</i> in case <i>KPOS</i> =3).	4, cm, 2*(cm, rad)	I, 5*E

<sup>1</sup>*MOD*=1 : centered multipole model ;*MOD*=2 : long-shifted dipole model ;*MOD*=3 : short-shifted dipole model.<sup>2</sup>*MOD* = 0 (default) : user defined back-leg windings (defined in routine agsblw.f) ;

*MOD* = 1 : actual AGS data are taken, namely : MM.A16AD : NBLW = 1, SIGN = 1.D0, NW = 10 ; MM.A17CF : NBLW = 1, SIGN = 1.D0, NW = 10 ; MM.A18CF : NBLW = 1, SIGN = -1.D0, NW = 10 ; MM.A19BD : NBLW = 1, SIGN = -1.D0, NW = 12 ; MM.A20BD : NBLW = 1, SIGN = 1.D0, NW = 12 ; MM.B02BF : NBLW = 2, SIGN = 1.D0, NW1 = 12, SIGN = 1.D0, NW2 = 6 ; MM.B03CD : NBLW = 1, SIGN = 1.D0, NW = 10 ; MM.B04CD : NBLW = 1, SIGN = -1.D0, NW = 10 ; MM.B05A : NBLW = 1, SIGN = -1.D0, NW = 10 ; MM.K19BD : NBLW = 1, SIGN = 1.D0, NW = 6 ; MM.K20B : NBLW = 1, SIGN = 1.D0, NW = 6 ; MM.L13CF : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM.L14C : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM.A07CD : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM.A08C : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM.B01B : NBLW = 1, SIGN = 1.D0, NW = 6 ; MM.L06A : NBLW = 1, SIGN = 1.D0, NW = 5 ; MM.L07C : NBLW = 1, SIGN = 1.D0, NW = 5 ; MM.A14C : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM.A15A : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM.E06A : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM.E07CD : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM.E20BD : NBLW = 1, SIGN = 1.D0, NW = 6 ; MM.F01BF : NBLW = 1, SIGN = 1.D0, NW = 6 ; MM.F14CF : NBLW = 1, SIGN = 1.D0, NW = 5 ; MM.F15AD : NBLW = 1, SIGN = 1.D0, NW = 5 ; MM.G08CD : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM.G09BF : NBLW = 1, SIGN = -1.D0, NW = 6.

<b>AGSQUAD</b>	<b>AGS quadrupole</b>	dim.	data type	FIT num.
<i>IL</i>	<i>IL</i> = 1, 2[ $\times 10^n$ ], 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2[ $\times 10^n$ ], 7	I	
<i>XL, R<sub>0</sub>, I<sub>W1</sub>, I<sub>W2</sub>, I<sub>W3</sub>, dI<sub>W1</sub>, dI<sub>W2</sub>, dI<sub>W3</sub></i>	Length of element ; radius at pole tip ; current in windings ; relative error on currents.	2*cm, 3*A 3*no dim	5*E 3*E	10 - 17
<i>X<sub>E</sub>, λ<sub>E</sub></i>	<b>Entrance face</b> Integration zone ; fringe field extent. (sharp edge if field extent is zero)	2*cm	11*E	20, 21
<i>NCE, C<sub>0</sub> - C<sub>5</sub></i>	Same as <i>QUADRUPO</i>	0-6, 6*no dim	I, 6*E	30 - 36
<i>X<sub>S</sub>, λ<sub>S</sub></i>	<b>Exit face</b> Integration zone ; as for entrance	2*cm	11*E	40, 41
<i>NCS, C<sub>0</sub> - C<sub>5</sub></i>		0-6, 6*no dim	I, 6*E	50 - 56
<i>R1</i>	Roll angle	rad	E	60
<i>XPAS</i>	Integration step	cm	E	70
<i>KPOS, XCE, YCE, ALE</i>	<i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt.	1-2, 2*cm, rad	I, 3*E	80 - 83

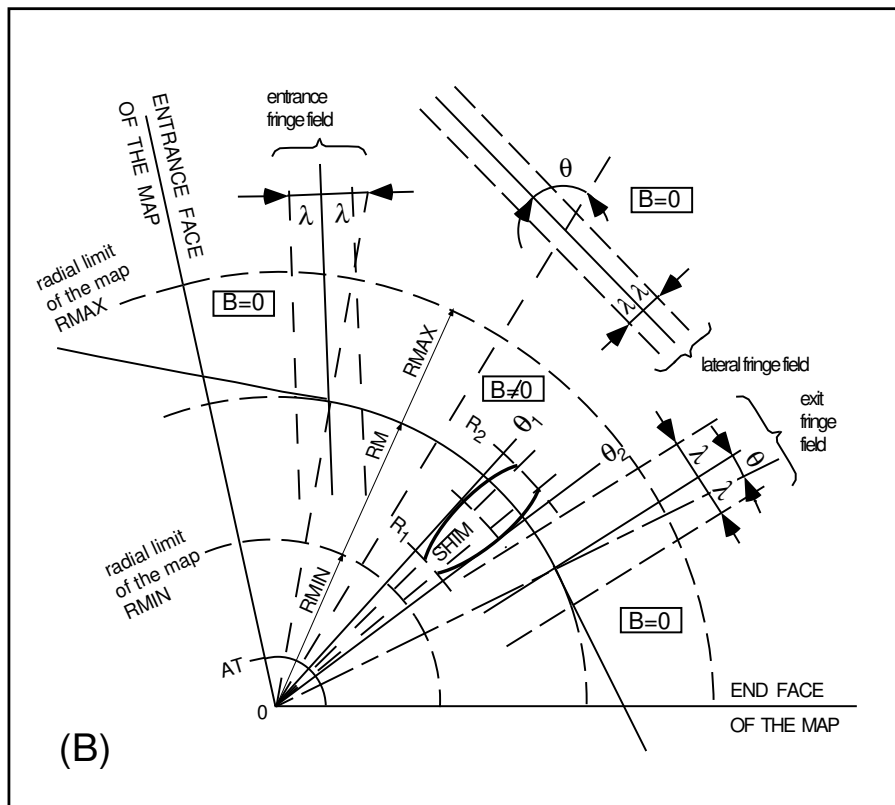
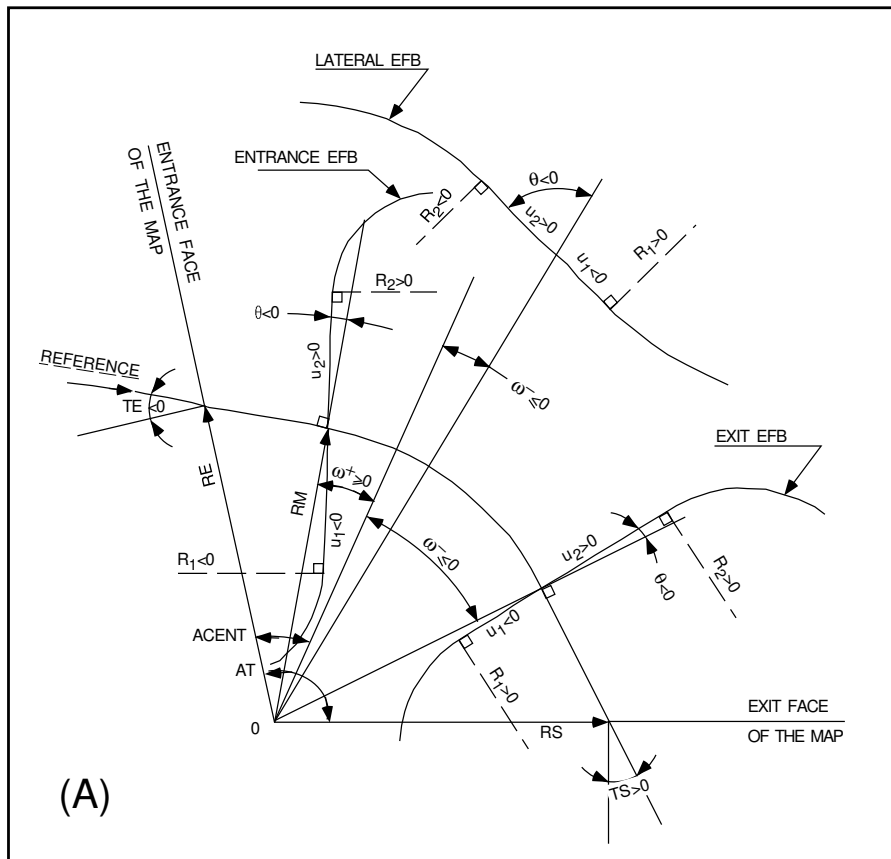
**AIMANT****Generation of dipole mid-plane 2-D map, polar frame**

$$B_Z = \mathcal{F}B_0 \left( 1 - N \left( \frac{R - RM}{RM} \right) + B \left( \frac{R - RM}{RM} \right)^2 + G \left( \frac{R - RM}{RM} \right)^3 \right)$$

<i>NFACE, IC, IL</i>	Number of field boundaries <i>IC</i> = 1, 2 : print field map <i>IL</i> = 1, 2[ $\times 10^n$ ], 7 : print coordinates, fields, etc., step-by-step, in <i>zgoubi.res</i> (1), <i>zgoubi.plt</i> (2), <i>zgoubi.impdev.out</i> (7).	2-3; 0-2; 0-2[ $\times 10^n$ ], 7 3*I	
<i>IAMAX, IRMAX</i>	Azimuthal and radial number of nodes of the mesh	$\leq 400, \leq 10^4$	2*I
<i>B<sub>0</sub>, N, B, G</i>	Field and field indices	kG, 3*no dim	4*E
<i>AT, ACENT, RM, RMIN, RMAX</i>	Mesh parameters : total angle of the map ; azimuth for EFBs positioning ; reference radius ; minimum and maximum radii	2*deg, 3*cm	5*E
ENTRANCE FIELD BOUNDARY			
$\lambda, \xi$	Fringe field extent (normally $\simeq$ gap size) ; flag : - if $\xi \geq 0$ : second order type fringe field with linear variation over distance $\xi$ - if $\xi = -1$ : exponential type fringe field : $F = (1 + \exp(P(s)))^{-1}$ $P(s) = C_0 + C_1\left(\frac{s}{\lambda}\right) + C_2\left(\frac{s}{\lambda}\right)^2 + \dots + C_5\left(\frac{s}{\lambda}\right)^5$	cm, (cm)	2*E
<i>NC, C<sub>0</sub> - C<sub>5</sub>, shift</i>	<i>NC</i> = 1 + degree of $P(s)$ ; $C_0$ to $C_5$ : see above ; EFB shift (ineffective if $\xi \geq 0$ )	0-6, 6*no dim, cm	I, 7*E
$\omega^+, \theta, R_1, U_1, U_2, R_2$	Azimuth of entrance EFB with respect to <i>ACENT</i> ; wedge angle of EFB ; radii and linear extents of EFB (use $U_{1,2} = \mp\infty$ when $R_{1,2} = \infty$ )	2*deg, 4*cm	6*E
(Note : $\lambda = 0, \omega^+ = \text{ACENT}$ and $\theta = 0$ for <u>sharp edge</u> )			
EXIT FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)			
$\lambda, \xi$	Fringe field parameters	cm, (cm)	2*E
<i>NC, C<sub>0</sub> - C<sub>5</sub>, shift</i>	See above	0-6, 6*no dim, cm	1, 7*E
$\omega^-, \theta, R_1, U_1, U_2, R_2$	Positioning and shape of the exit EFB	2*deg, 4*cm	6*E
(Note : $\lambda = 0, \omega^- = -\text{AT} + \text{ACENT}$ and $\theta = 0$ for <u>sharp edge</u> )			

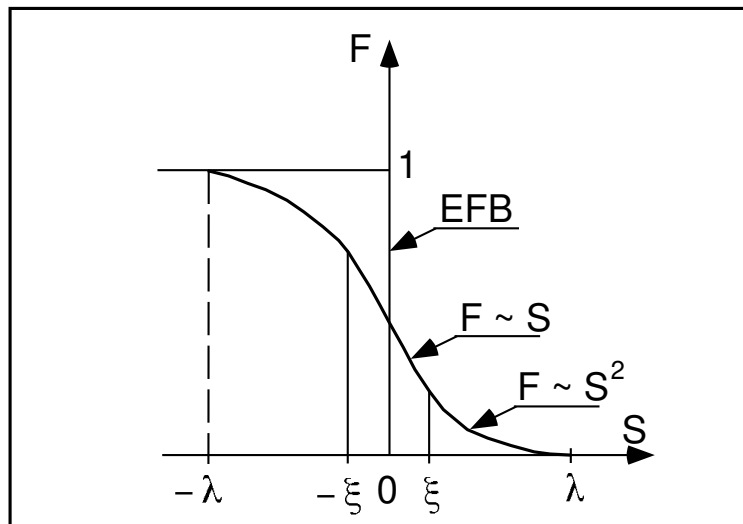
<b>If NFACE = 3</b>	LATERAL FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY) Next 3 records <i>only</i> if NFACE = 3		
$\lambda, \xi$	Fringe field parameters	cm, (cm)	2*E
$NC, C_0 - C_5, shift$	See above	0-6, 6*no dim, cm	I, 7*E
$\omega^-, \theta, R_1, U_1, U_2, R_2, RM3$	Positioning and shape of the lateral EFB ; RM3 is the radial position on azimuth ACENT	2*deg, 5*cm	7*E
<b>NBS</b>	Option index for perturbations to the field map	-2 - 0 or $\geq 1$	I
<b>If NBS = 0</b>	Normal value. No other record required		
<b>If NBS = -2</b>	The map is modified as follows :		
$R_0, \Delta B/B_0$	$B$ transforms to $B * \left(1 + \frac{\Delta B}{B_0} \frac{R - R_0}{RMAX - RMIN}\right)$	cm, no dim	2*E
<b>If NBS = -1</b>	the map is modified as follows :		
$\theta_0, \Delta B/B_0$	$B$ transforms to $B * \left(1 + \frac{\Delta B}{B_0} \frac{\theta - \theta_0}{AT}\right)$	deg, no dim	2*E
<b>If NBS <math>\geq 1</math></b>	Introduction of NBS shims		
<b>For I = 1, NBS</b>	The following 2 records must be repeated NBS times		
$R_1, R_2, \theta_1, \theta_2, \lambda$	Radial and angular limits of the shim ; $\lambda$ is unused	2*cm, 2*deg, cm	5*E
$\gamma, \alpha, \mu, \beta$	geometrical parameters of the shim	2*deg, 2*no dim	4*E
<b>IRDRE</b>	Degree of interpolation polynomial : 2 = second degree, 9-point grid 25 = second degree, 25-point grid 4 = fourth degree, 25-point grid	2, 25 or 4	I
<b>XPAS</b>	Integration step	cm	E
<b>KPOS</b>	Positioning of the map, normally 2. Two options :	1-2	I
<b>If KPOS = 2</b> RE, TE, RS, TS	Positioning as follows : Radius and angle of reference, respectively, at entrance and exit of the map.	cm, rad, cm, rad	4*E
<b>If KPOS = 1</b> DP	Automatic positioning of the map, by means of reference relative momentum	no dim	E



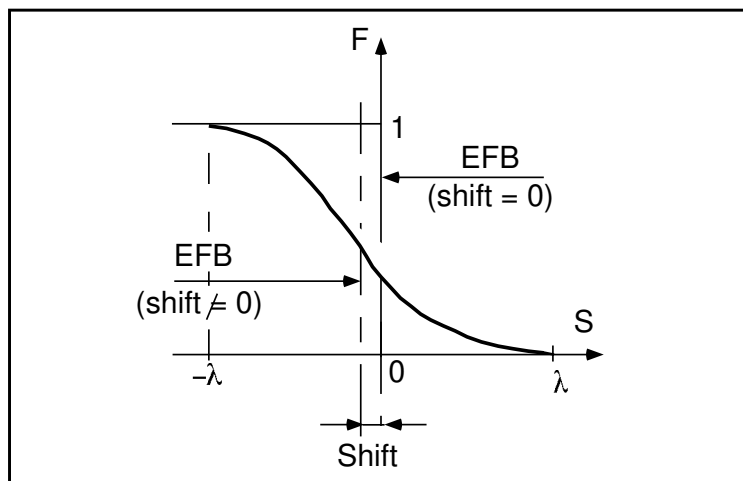


A : Parameters used to define the field map and geometrical boundaries.

B : Parameters used to define the field map and fringe fields.



Second order type fringe field.



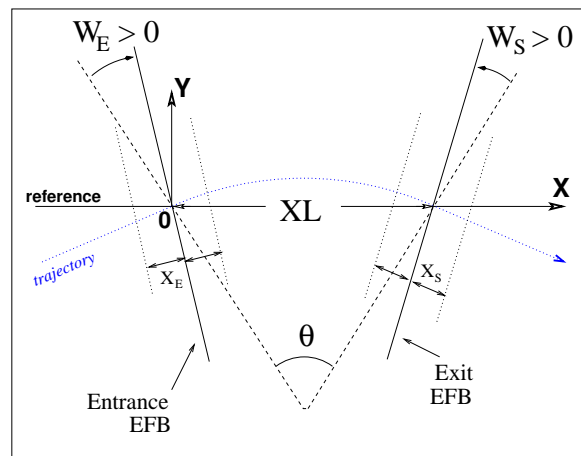
Exponential type fringe field.

<b>AUTOREF</b>	<b>Transport beam into a new reference frame</b>		
<i>I</i>	<p>1 : Equivalent to <i>CHANGREF</i>(<i>XCE</i> = 0, <i>YCE</i> = <i>Y</i>(1), <i>ALE</i> = <i>T</i>(1)), <i>i.e.</i>, recentering of the beam on particle #1.</p> <p>2 : Equivalent to <i>CHANGREF</i>(<i>XW</i>, <i>YW</i>, <i>T</i>(1)), with (<i>XW</i>, <i>YW</i>) the location of the intersection (waist) of particles 1, 4 and 5 (useful with <i>MATRIX</i>, for automatic positioning of the first order focus).</p>	1-4	I
<b>If</b> <i>I</i> = 3 <i>I1, I2, I3</i>	<p>Provide next record if <i>I</i> = 3</p> <p>Three particle numbers</p> <p>Equivalent to <i>CHANGREF</i>(<i>XW</i>, <i>YW</i>, <i>T</i>(<i>I1</i>)), with (<i>XW</i>, <i>YW</i>) the location of the intersection (waist) of particles <i>I1</i>, <i>I2</i> and <i>I3</i> (for instance : <i>I1</i> = central trajectory, <i>I2</i> and <i>I3</i> = paraxial trajectories that intersect at the first order focus).</p>	3*(1- <i>IMAX</i> )	3*I
<b>If</b> <i>I</i> = 4 <i>XCE, YCE, ALE</i>	<p>Provide next record if <i>I</i> = 4</p> <p><i>XCE</i> and beam centroid new coordinates <i>YCE, ALE</i></p> <p>Equivalent to <i>CHANGREF</i>(<i>XCE, YCE, ALE</i>).</p>	2*cm, mrad	3*E
<b>If</b> <i>I</i> = 4.1 <i>XCE, YCE, ALE, DCE, TIME</i>	<p>Provide next record if <i>I</i> = 4.1</p> <p><i>XCE</i> and beam centroid new coordinates <i>YCE, ALE, DCE, TIME</i></p> <p>4Equivalent to <i>CHANGREF</i>(<i>XCE, YCE, ALE</i>) with in addition centering of the beam on a new relative momentum <i>DCE</i>.</p>	2*cm, mrad, -, $\mu$ s	6*E
<b>If</b> <i>I</i> = 4.2 <i>XCE, YCE, ALE, DCE, TIME</i>	<p>Provide next record if <i>I</i> = 4.2</p> <p><i>XCE</i> and beam centroid new coordinates <i>YCE, ALE, DCE</i>, time setting <i>TIME</i> (same for all particles).</p>	2*cm, mrad, -, $\mu$ s	6*E
<b>If</b> <i>I</i> = 5 <i>ZCE, PLE</i>	<p>Provide next record if <i>I</i> = 5</p> <p>New vertical beam centroid coordinates <i>ZCE, PLE</i></p> <p>The beam is centered vertically on <i>ZCE, PLE</i>.</p>	cm, mrad	2*E

**BEAMBEAM****Beam-beam lens**

SW, I	0/1 : off/on ; beam intensity. Use <i>SPNTRK</i> to activate spin kicks.	0-2, Amp	I, E
$\alpha_Y, \beta_Y, \epsilon_{Y,norm}/\pi$	Beam parameters, horizontal.	- , m, m.rad	3*E
$\alpha_Z, \beta_Z, \epsilon_{Z,norm}/\pi$	Beam parameters, vertical.	- , m, m.rad	3*E
$\sigma_X, \sigma_{dp/p}$	<i>rms</i> bunch length ; <i>rms</i> momentum spread.	m, -	2*E
$\mathcal{C}, \alpha$	Ring circumference ; momentum compaction.	m, -	2*E
$Q_Y, Q_Z, Q_s$	Tunes, horizontal, vertical, synchrotron.	-, -, -	3*E
$A_Y, A_Z, A_X$	Amplitudes, horizontal, vertical, longitudinal.	-, -, -	3*E

<b>BEND</b>	<b>Bending magnet, Cartesian frame</b>	(page 78 in PART A)	
$IL$	$IL = 1, 2[\times 10^n], 7$ : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	$0-2[\times 10^n], 7$	I
$XL, Sk, B1$	Length ; skew angle ; field (change ALE and wedge signs, if $B < 0$ ).	cm, rad, kG	3*E
$X_E, \lambda_E, W_E$	<b>Entrance face :</b> Integration zone extent ; fringe field extent (normally $\simeq$ gap height ; zero for sharp edge) ; wedge angle ( $>0$ for rectangle magnet).	cm, cm, rad	3*E
$N, C_0-C_5$	Unused ; fringe field coefficients : $B(s) = B1 F(s)$ with $F(s) = 1/(1 + \exp(P(s)))$ and $P(s) = \sum_{i=0}^5 C_i (s/\lambda)^i$	unused, 6*no dim	I, 6*E
$X_S, \lambda_S, W_S$	<b>Exit face :</b> See entrance face	cm, cm, rad	3*E
$N, C_0-C_5$		unused, 6*no dim	I, 6*E
$XPAS$	Integration step	cm	E
$KPOS, XCE, YCE, ALE$	$KPOS=1$ : element aligned, $2$ : misaligned ; shifts, Z-tilt. $KPOS = 3$ : ( $XCE$ is unused) entrance and exit frames are shifted by $YCE$ and tilted wrt. the magnet by an angle of <ul style="list-style-type: none"> <li>• either <math>ALE</math>, if <math>ALE</math> is non-zero (normally, <math>ALE &gt; 0</math> in that case)</li> <li>• or <math>2 \text{ Arcsin}(B1 \times XL / 2 \times BORO)</math> if <math>ALE=0</math></li> </ul>	1-2, 2*cm, rad	I, 3*E



Geometry and parameters of *BEND* :  $XL$  = length,  $\theta$  = deviation,  $W_E, W_S$  are the entrance and exit wedge angles. The motion is computed in the Cartesian frame ( $O, X, Y, Z$ ).

• **Example** : Positive or negative field *BEND*

The data list at left hand, below, simulates a case of positive field, followed by two different ways to simulate the same magnet with negative field. (This can be copy-pasted to **zgoubi**, as is, for execution.)

The right hand column shows the transport matrix in all three cases.

```

Test signs, with wedges.

! Magnet deviation is
! tta=11.25deg=2asin(L/2rho)=2asin(B.L/2.Brho) with L=247.30039, Brho=1000, B=1.57776

! 1/ Positive B, regular form of input data
'OBJET'
1000.
5
.01 .1 .01 .1 0. .001
0. 0. 0. 0. 0. 1.

'BEND'
0
247.30039 0. 1.57776
20. 8. 0.04276056667 ! wedge is +2.45deg
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
20. 8. 0.04276056667 ! wedge is +2.45deg
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
#30|10|30
3 0. 0. -.1963495408

Reference particle (# 1), path length : 248.88547
TRANSFER MATRIX ORDRE 1 (MKSA units)
0.940559 2.42528 0.00000 0.00000 0.0 0.482374
-4.768802E-02 0.940229 0.00000 0.00000 0.0 0.385904
0.00000 0.00000 0.985447 2.48908 0.0 0.00000
0.00000 0.00000 -1.163098E-02 0.985390 0.0 0.00000
0.385970 0.482384 0.00000 0.00000 1.0 6.370064E-02
0.00000 0.00000 0.00000 0.00000 0.0 1.00000

'FAISCEAU'
'MATRIX'
1 0

! 2/ Negative B. Compared to 1/ : R11-R44 unchanged, R16, R26 change sign.
! 2-a/ By changing sign of B. Note : signs of ALE and wedges have to be reversed.
'OBJET'
1000.
5
.01 .1 .01 .1 0. .001
0. 0. 0. 0. 0. 1.

'BEND'
0
247.30039 0. -1.57776
20. 8. -0.04276056667 ! wedge is +2.45deg
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
20. 8. -0.04276056667 ! wedge is +2.45deg
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
#30|10|30
3 0. 0. .1963495408

Reference particle (# 1), path length : 248.88547
TRANSFER MATRIX ORDRE 1 (MKSA units)
0.940559 2.42528 0.00000 0.00000 0.0 -0.482374
-4.768802E-02 0.940229 0.00000 0.00000 0.0 -0.385904
0.00000 0.00000 0.985447 2.48908 0.0 0.00000
0.00000 0.00000 -1.163098E-02 0.985390 0.0 0.00000
-0.385970 -0.482384 0.00000 0.00000 1.0 6.370064E-02
0.00000 0.00000 0.00000 0.00000 0.0 1.00000

'FAISCEAU'
'MATRIX'
1 0

! 2-a/ Using YMY instead. All data in BEND remain unchanged.
'OBJET'
1000.
5
.01 .1 .01 .1 0. .001
0. 0. 0. 0. 0. 1.

'YMY'
'BEND'
0
247.30039 0. 1.57776
20. 8. 0.04276056667 ! wedge is +2.45deg
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
20. 8. 0.04276056667 ! wedge is +2.45deg
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
#30|10|30
3 0. 0. -.1963495408
'YMY'

Reference particle (# 1), path length : 248.88547
TRANSFER MATRIX ORDRE 1 (MKSA units)
0.940559 2.42528 0.00000 0.00000 0.0 -0.482374
-4.768802E-02 0.940229 0.00000 0.00000 0.0 -0.385904
0.00000 0.00000 0.985447 2.48908 0.0 0.00000
0.00000 0.00000 -1.163098E-02 0.985390 0.0 0.00000
-0.385970 -0.482384 0.00000 0.00000 1.0 6.370064E-02
0.00000 0.00000 0.00000 0.00000 0.0 1.00000

'FAISCEAU'
'MATRIX'
1 0

'END'

```

**BETATRON**

**Betatron core (momentum kick)**

dp-kick

Momentum kick, relative to particle momentum.

-

E

**BINARY*****BINARY*↔*FORMATTED* data converter**

*NF*[.J], *NCol*, *NHDR*      Number of files to convert [READ format type, see below],      ≤ 20, ≤ 7, 0 – 9    3\*I1  
of data columns, of header lines.

**The next *NF* lines :**

*FNAME*      Name of the file to be converted. File content is assumed binary      A80  
*iff* name begins with “B\_” or “b\_”, assumed formatted otherwise.

READ format, case of formatted input file :

If FRM not given      Format is '\*'  
If FRM=1      Format is '1X, 7E11.\*'

READ format, case of binary input file :

Expected format is 7 column rows.



<b>BREVOL</b>		<b>1-D uniform mesh magnetic field map</b>	
		X-axis cylindrical symmetry is assumed	
<i>IC, IL</i>	<i>IC</i> = 1, 2 : print the map <i>IL</i> = 1, 2[ $\times 10^n$ ], 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2; 0-2[ $\times 10^n$ ], 7	2*I
<i>BNORM, XN</i>	Field and X-coordinate unit conversion coeff. Convert values as read from map file, to kG and cm units.	2*UnitConv.	2*E
<i>TITL</i>	Title. Start with "FLIP" to get field map X-flipped.		A80
<i>IX</i>	Number of longitudinal nodes of the map	$\leq 400$	I
<i>FNAME [ , SUM]</i> <sup>1,2</sup>	File name		A80
<i>ID, A, B, C</i> [, <i>A', B', C', B''</i> , etc., if <i>ID</i> $\geq 2$ ]	Integration boundary. Ineffective when <i>ID</i> = 0. <i>ID</i> = -1, 1 or $\geq 2$ : as for <i>CARTEMES</i>	$\geq -1$ , 2*no dim, cm [,2*no dim, cm, etc.]	I,3*E [,3*E,etc.]
<i>IODRE</i>	Unused	2, 25 or 4	I
<i>XPAS</i>	Integration step	cm	E
<i>KPOS, XCE, YCE, ALE</i>	<i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt.	1-2, 2*cm, rad	I, 3*E

<sup>1</sup> *FNAME* (e.g., solenoid.map) contains the field data. These must be formatted according to the following *FORTRAN* sequence :

```
OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD' [,FORM='UNFORMATTED'])
DO 1 I = 1, IX
  IF (BINARY) THEN
    READ(NL) X(I), BX(I)
  ELSE
    READ(NL,*) X(I), BX(I)
  ENDF
  1 CONTINUE
```

where *X(I)* and *BX(I)* are the longitudinal coordinate and field component at node (*I*) of the mesh. Binary file names must begin with *FNAME* 'B\_' or 'b\_'. 'Binary' will then automatically be set to '.TRUE.'.

<sup>2</sup> Superimposing (summing up) field maps is possible. Up to 4 at most.

To do so, pile up file names with 'SUM' following each name but the last one.

In the following example e.g., 3 field maps are read and summed up<sup>2</sup>:

```
myMapFile1 SUM
myMapFile2 SUM
myMapFile3
```

(all maps must all have their mesh defined in identical coordinate frame).

<b>CARTEMES</b>	<b>2-D Cartesian uniform mesh magnetic field map</b> mid-plane symmetry is assumed		
<i>IC, IL</i>	<i>IC</i> = 1, 2 : print the map <i>IL</i> = 1, 2[ $\times 10^n$ ], 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2; 0-2[ $\times 10^n$ ], 7	2*I
<i>BNORM, XN, YN</i>	Field and X-,Y-coordinate unit conversion coeffs. Convert values as read from map file, to kG and cm units.	3*UnitConv.	3*E
<i>TITL</i>	Title. Start with "FLIP" to get field map X-flipped.		A80
<i>IX, JY</i>	Number of longitudinal ( <i>IX</i> ) and transverse ( <i>JY</i> ) nodes of the map	$\leq 400, \leq 200$	2*I
<i>FNAME</i> <sup>1</sup>	File name		A80
<i>ID, A, B, C</i> [, <i>A', B', C', A'', B'', etc.</i> , if <i>ID</i> $\geq 2$ ]	Integration boundary. Normally <i>ID</i> = 0. <i>ID</i> = -1 : integration in the map begins at entrance boundary defined by $AX + BY + C = 0$ . <i>ID</i> = 1 : integration in the map is terminated at exit boundary defined by $AX + BY + C = 0$ . <i>ID</i> $\geq 2$ : entrance ( <i>A, B, C</i> ) and up to <i>ID</i> - 1 exit ( <i>A', B', C', A'', B'', etc.</i> ) boundaries	$\geq -1, 2$ *no dim, cm [ $2$ *no dim, cm, etc.]	I, 3*E [3*E, etc.]
<i>IORBRE</i>	Degree of interpolation polynomial (see <i>DIPOLE-M</i> )	2, 25 or 4	I
<i>XPAS</i>	Integration step	cm	E
<i>KPOS, XCE, YCE, ALE</i>	<i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt.	1-2, 2*cm, rad	I, 3*E

<sup>2</sup> *FNAME* (e.g., spes2.map) contains the field data. These must be formatted according to the following *FORTRAN* sequence :

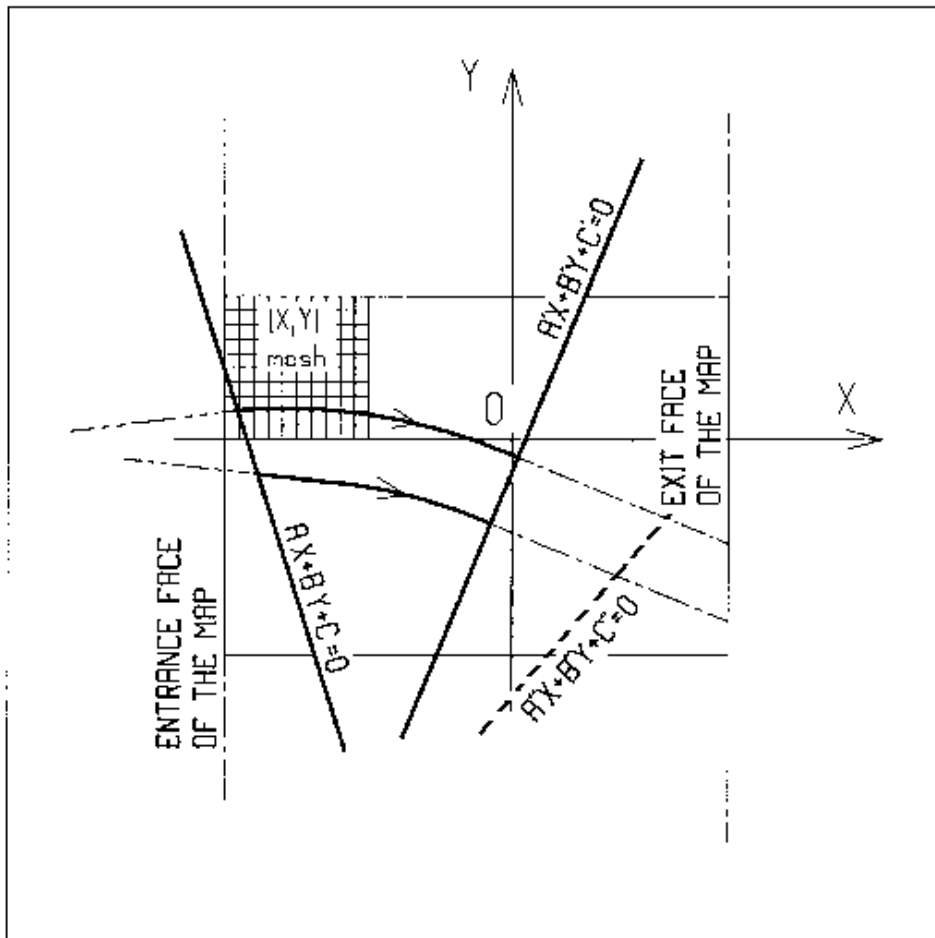
```

OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD' [,FORM='UNFORMATTED'])
IF (BINARY) THEN
  READ(NL) (Y(J), J=1, JY)
ELSE
  READ(NL,100) (Y(J), J=1, JY)
ENDIF
100  FORMAT(10 F8.2)
DO 1 I=1,IX
  IF (BINARY) THEN
    READ(NL) X(I), (BMES(I,J), J=1, JY)
  ELSE
    READ(NL,101) X(I), (BMES(I,J), J=1, JY)
  101  FORMAT(10 F8.2)
ENDIF
1  CONTINUE

```

where  $X(J)$  and  $Y(J)$  are the longitudinal and transverse coordinates and  $BMES$  is the  $Z$  field component at a node  $(I, J)$  of the mesh. For binary files, *FNAME* must begin with 'B.' or 'b.'

'Binary' will then automatically be set to 'TRUE.'



$OXY$  is the coordinate system of the mesh. Integration zone limits may be defined, using  $ID \neq 0$ : particle coordinates are extrapolated linearly from the entrance face of the map, into the plane  $A'X + B'Y + C' = 0$ ; after ray-tracing inside the map and terminating on the integration boundary  $AX + BY + C = 0$ , coordinates are extrapolated linearly to the exit face of the map.

<b>CAVITE</b> <sup>1,2</sup>	<b>Accelerating cavity</b> $\Delta W = q\hat{V} \sin(2\pi h f \Delta t + \phi_s)$ or other voltage and frequency laws.		(page 84 in PART A)
<i>IOPT</i> [ <i>i</i> ] [ <i>PRINT</i> ] [ <i>Facility</i> ] [ <i>SHIFT_PH0</i> ] [ <i>CENTROID</i> ]	Option, see below ; <i>i</i> = 1 (equivalently, presence of 'PRINT') causes info logged to <i>zgoubi.CAVITE.Out</i> ; <i>Facility</i> is accounted for <i>iff IOPT=10</i> , or <i>11</i> (otherwise ignored), it can be : - 'CornellSynch' (hard-coded voltage), or - 'EIC_RCS' (hard-coded synchronous RF phase), or - 'EICRCS_File' (read data from external file <i>zgoubi.RFLaw.In</i> <sup>3</sup> ), or - CEBAF (overrides <i>SHIFT_PH0</i> : imposes phase origin at $\pi/2$ ) ; 'SHIFT_PH0' is accounted for <i>iff IOPT=10</i> (otherwise ignored), it shifts the RF phase of any particle by $\pi/2$ ; 'CENTROID' is accounted for <i>iff IOPT=10</i> (otherwise ignored), it shifts the RF phase of any particle by the bunch centroid phase.	0-11, -, -, -	I[.1] [, A] [, A] [, A] [, A]
<b>If IOPT=0</b>	Element inactive		
<i>X, X</i>	Unused.		
<i>X, X</i>	Unused.		
<b>If IOPT=1</b> <sup>4</sup>	$f_{RF}$ follows the timing law given by <i>SCALING</i>		
<i>L, h</i>	Reference closed orbit length ; harmonic number.	m, no dim	2*E
$\hat{V}, X$	R.F. peak voltage ; unused.	V, unused	2*E
<b>If IOPT=2</b> [ <i>i</i> ]	$f_{RF}$ follows $\Delta W_s = q\hat{V} \sin \phi_s$ . Handles SR loss in store mode. Handles double RF systems		
- either :			
<i>L, h</i> [ <i>POS</i> ]	Reference closed orbit length ; harmonic number ; s location (optional)	m, - [,m]	2*E [,E]
$\hat{V}, \phi_s$	R.F. peak voltage ; synchronous phase.	V, rad	2*E
- or :			
<i>L, {h<sub>1</sub>, h<sub>2</sub>}</i> [ <i>POS</i> ]	Reference closed orbit length ; harmonic numbers ; s location (optional)	m, - [,m]	2*E [,E]
{ $\hat{V}_1, \hat{V}_2$ }, $\phi_s$ [ <i>dphn</i> ]	R.F. peak voltages ; synchronous phase ; phase-shift (default is <i>dphn</i> = $\pi/21$ ).	2*V, rad [, rad]	3*E [, E]
<i>If i=2</i>	<i>a third input data line is needed (case of acceleration range including transition)</i>		
$\gamma_{tr}$	Transition gamma value	-	E
<b>If IOPT=3</b>	No synchrotron motion : $\Delta W = q\hat{V} \sin \phi_s$		
<i>X, X</i>	Unused ; unused.	2*unused.	2*E
$\hat{V}, \phi_s$	R.F. peak voltage ; synchronous phase.	V, rad	2*E
<b>If IOPT=6</b>	Read RF frequency and/or phase law from external file, "zgoubi.freqLaw.In" <sup>5</sup> .		
<i>X, E<sub>kin,s</sub></i>	Unused ; initial synchronous kin.-energy; RF harmonic.	-, -	2*E
$\hat{V}, \Phi_s$	R.F. peak voltage ; synchronous phase.	V, rad	2*E
<b>If IOPT=7</b>	Quasi- or isochronous acceleration.		
<i>X, f<sub>RF</sub></i>	Unused ; RF frequency.	-, Hz	2*E
$\hat{V}, \Phi_s$	R.F. peak voltage ; synchronous phase.	V, rad	2*E
<b>If IOPT=10</b>	Chambers matrix method (origin of phase determined by 'SHIFT_PH0' option in line 1).		
<i>L, f<sub>RF</sub></i>	Cavity length ; RF frequency	m, Hz	2*E
$\hat{V}, \Phi_s$ [ <i>IOP</i> ]	R.F. peak voltage ; synchronous phase ; matrix (damping) options: either <i>IOP=0</i> : default value, transverse matrix is a drift; or <i>IOP</i> = $\pm 1$ : $dE/E \ll 1$ approx. and, resp <sup>ly</sup> , $\left\{ \begin{array}{l} \text{transverse damping} \\ \text{transv. determinant} = 1 \end{array} \right.$ ; or <i>IOP</i> = $\pm 2$ : $\forall dE/E$ and, resp <sup>ly</sup> , $\left\{ \begin{array}{l} \text{transverse damping} \\ \text{transv. determinant} = 1 \end{array} \right.$	V, rad, [ $\in [-2, +2]$ ]	2*E [, I]

<b>If IOPT=11</b>	Acceleration in presence of SR loss (optionally, line 1: <i>Facility= 'CornellSynch'</i> (voltage law is hard-coded), or <i>Facility= 'EIC_RCS'</i> (synchronous phase law is hard-coded).		
$\mathcal{L}, E_k$	Orbit length and kinetic energy at start of acceleration.	m, MeV [,-]	2*E [,-]
$\hat{V}, \Phi_s, U_{00}$ [,POS]	R.F. peak voltage (unused - hard-coded - if <i>Facility= 'CornellSynch'</i> ) ; synchronous phase (will be initial value if <i>Facility= 'EIC_RCS'</i> ) ; energy loss to be recovered here (value at initial energy) <sup>6</sup> ; optional: cavity distance from beg. of sequence (not needed if cavity is at beginning or end).	V, rad, eV [,-]	3*E [,-]

<sup>1</sup>For the geek: details in subroutines *rcavit.f* (read input), *cavite.f* (cavity models).

<sup>2</sup> Use *PARTICUL* to declare mass and charge.

<sup>3</sup> Formatting of *zgoubi.RFLaw.In* file is 5 columns: turn #, energy/MeV, unused, voltage/V, unused

<sup>4</sup> For ramping the R.F. frequency following  $B\rho(t)$ , use *SCALING*, with family *CAVITE*.

<sup>5</sup> The formatting of these ascii data file is given page 185.

<sup>6</sup> Synchrotron radiation loss at subsequent turns is actualized using  $U \propto E^4$ .

<b>CHAMBR</b>	<b>Long transverse aperture limitation</b> <sup>1</sup>		
<i>IA</i>	0 : element inactive 1 : (re)definition of the aperture 2 : stop testing and reset counters, print information on stopped particles.	0-2	I
<i>IFORM</i> [.J], <i>C1</i> , <i>C2</i> , <i>C3</i> , <i>C4</i>	<i>IFORM</i> = 1 : rectangular aperture ; <i>IFORM</i> = 2 : elliptical aperture. <i>J</i> = 0, default : opening is <sup>2</sup> $\pm YL = \pm C1$ , $\pm ZL = \pm C2$ , centered at $YC = C3$ , $ZC = C4$ . <i>J</i> = 1 : opening is <sup>2</sup> , in Y : [ <i>C1</i> , <i>C2</i> ], in Z : [ <i>C3</i> , <i>C4</i> ]	1-2[.0-1]	I[.I], 4*E

<sup>1</sup> Any particle out of limits is stopped.

<sup>2</sup> When used with an optical element defined in polar coordinates (e.g., *DIPOLE*) *YL* is the radius and *YC* stands for the reference radius (normally,  $YC \simeq RM$ ).

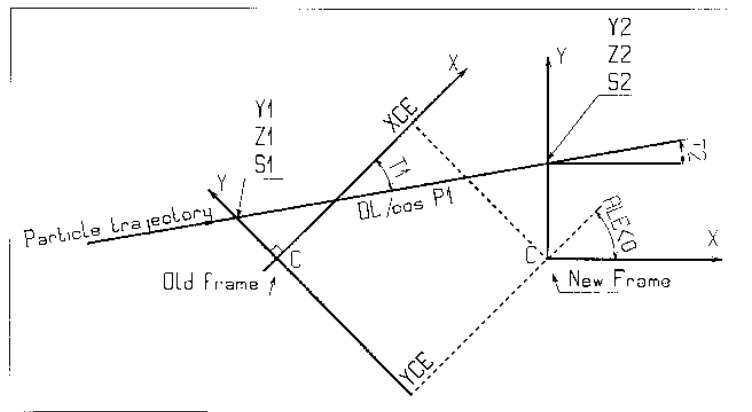
**CHANGREF Transformation to a new reference frame**  
**SCALING** can be used to scale positioning parameters in *CHANGREF*.

“Old Style” (Figure below) :

*XCE, YCE, ALE* Longitudinal and transverse shifts, 2\*cm, deg 3\*E  
 followed by Z-axis rotation

“New Style” (see example below). In an arbitrary order, up to 9 occurrences of :

*XS 'val', YS 'val', ZS 'val', XR 'val', YR 'val', ZR 'val'* cm or deg up to 9\*(A2,E)

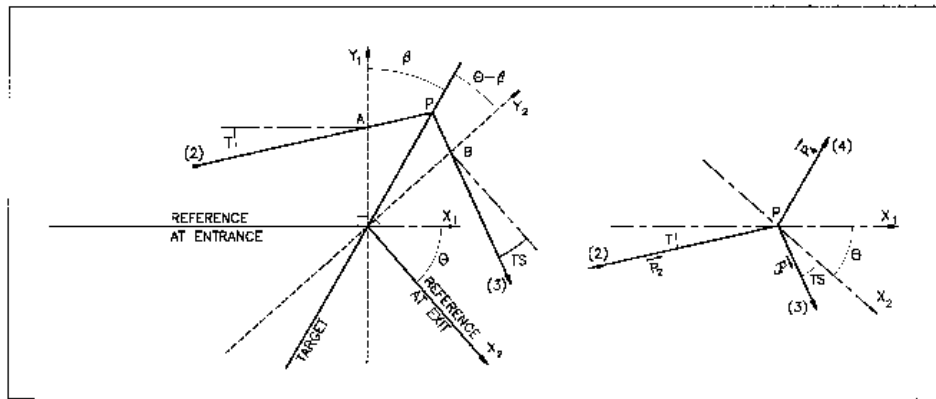


Parameters in the *CHANGREF* procedure.

• **Example : CHANGREF**

```
Using CHANGREF "New Style
'OBJET'
51.71103865921708 Electron, Ekin=15MeV.
2
1 1 One particle, with
2. 0. 0.0 0.0 0.0 1. 'R' Y_0=2 cm, other coordinates zero.
1 1 1 1 1 1
'MARKER' BEG .plt -> list into zgoubi.plt.
'DRIFT' 10 cm drift.
10.
'CHANGREF'
ZR -6.34165 YS 1. First half Z-rotate, Next Y-shift.
'MULTIPOL' Combined function multipole, dipole + quadrupole.
2 -> list into zgoubi.plt.
5 10. 2.064995867082342 2. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0 0 5. 1.1 1.00 1.00 1.00 1.00 1.00 1. 1. 1. 1.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.
0 0 5. 1.1 1.00 1.00 1.00 1.00 1.00 1.00 1. 1. 1. 1.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.
0 0 0 0 0 0 0 0 0 0
.1 step size
1 0. 0. 0.
'CHANGREF'
YS -1. ZR -6.341 First Y-shift back, next half Z-rotate.
'DRIFT' 10 cm drift.
10.
'FAISCEAU'
'END'
```

<b>CIBLE, TARGET</b>	<b>Generate a secondary beam following target interaction</b>		
$M_1, M_2, M_3, Q$ $T_2, \theta, \beta$	Target, incident and scattered particle masses ; $Q$ of the reaction ; incident particle kinetic energy ; scattering angle ; angle of the target	$5 * \frac{MeV}{c^2}, 2 * \text{deg}$	$7 * E$
$NT, NP$	Number of samples in $T$ and $P$ coordinates after <i>CIBLE</i>		$2 * I$
$TS, PS, DT$	Sampling size ; tilt angle	$3 * \text{mrad}$	$3 * E$
<i>BORO</i>	New reference rigidity after <i>CIBLE</i>	kG.cm	$E$



Scheme of the principles of *CIBLE (TARGET)*

$A, T$  = position, angle of incoming particle 2 in the entrance reference frame

$P$  = position of the interaction

$B, T$  = position, angle of the secondary particle in the exit reference frame

$\theta$  = angle between entrance and exit frames

$\beta$  = tilt angle of the target



<b>COLLIMA</b>	<b>Collimator</b> <sup>1</sup>		
<i>IA</i>	0 : element inactive 1 : element active 2 : element active and print information on stopped particles	0-2	I
<b>Physical-space collimation</b>			
<i>IFORM</i> [ <i>J</i> ], <i>C1</i> , <i>C2</i> , <i>C3</i> , <i>C4</i>	<i>IFORM</i> = 1 or 2 : rectangular or elliptical aperture ; <i>J</i> = 0, default : opening is $\pm YL = \pm C1$ , $\pm ZL = \pm C2$ , centered at $YC = C3$ , $ZC = C4$ . <i>J</i> = 1 : opening is, in Y : [ <i>C1</i> , <i>C2</i> ], in Z : [ <i>C3</i> , <i>C4</i> ]	1-2[.0-1], 4*cm	I[,I], 4*E
<b>Longitudinal collimation</b>			
<i>IFORM</i> . <i>J</i> , <i>H<sub>min</sub></i> , <i>H<sub>max</sub></i> , <i>V<sub>min</sub></i> , <i>V<sub>max</sub></i>	<i>IFORM</i> = 6 or 7 for horizontal variable resp <sup>ly</sup> S or Time, <i>J</i> =1 or 2 for vertical variable resp <sup>ly</sup> 1+dp/p, kinetic-E (MeV) ; horizontal and vertical limits	2*cm or 2* $\mu$ s, 2*no.dim or 2*MeV	I, 4*E
<b>Phase-space elliptical collimation</b>			
<i>IFORM</i> , $\alpha$ , $\beta$ , $\epsilon/\pi$ , <i>N<sub><math>\sigma</math></sub></i> , <i>XM</i> , <i>XPM</i>	<i>IFORM</i> = 11, 14 : horizontal collimation ; horizontal ellipse parameters (unused if 14) <sup>2</sup> , emittance, cut-off, ellipse centering (unused if 14) <sup>2</sup> ; <i>IFORM</i> = 12, 15 : vertical collimation ; vertical ellipse parameters (unused if 15) <sup>2</sup> , emittance, cut-off, ellipse centering (unused if 15) <sup>2</sup> ; <i>IFORM</i> = 13, 16 : longitudinal collimation ; <i>to be implemented</i> .	11-16, no.dim, 2*m, no.dim	I, 4*E

<sup>1</sup> Any particle out of limits is stopped.

<sup>2</sup> If *IFORM* = 14, 15 or 16, the *rms* ellipse matched to the particle distribution is first computed (namely, ellipse parameters  $\alpha$ ,  $\beta$  and centering *XM*, *XPM*). It is then considered to be the phase-space collimation aperture.

<b>CYCLOTRON</b>	<b>Spiral setor cyclotron</b>	(page 96 in PART A)
	$B_Z = \sum_{i=1}^{N \text{ dipoles}} BN_i \mathcal{F}_i(R, \theta) \mathcal{R}_i(R) \text{ with } \mathcal{R}_i(R) = B_0 + \sum_{j=1}^{j \leq 20} B_{j,i} \times R^j$	
<i>IL</i>	<i>IL</i> = 1, 2[×10 <sup>n</sup> ], 7 : print coordinates along trajectories, fields, etc., into zgoubi.res (1) or zgoubi.plt (2[×10 <sup>n</sup> ]) or zgoubi.impdev.out (7).	0-2[×10 <sup>n</sup> ], 7 I
<i>N, AT, RM, ITYP</i>	Number of dipoles within AT ; total angular extent of the dipole ; reference radius; sector type: spiral (0), radial (1), both (2)	-, deg, cm, 0-2 I, 2*E, I
<i>Repeat N times the following sequence</i> _____		
<i>ACN, δRM, BN, B<sub>0</sub>, K, R<sub>ref</sub>, [H<sub>i</sub>, i = 1, n]</i>	Dipole positioning ( <i>ACN, RM + δRM</i> ) ; field normalization ; dipole field ; unused ; unused ; optional: up to 20 field indices	deg, cm, -, kG, -, - [ , cm <sup>-j</sup> ]
<b>ENTRANCE FIELD BOUNDARY</b>		
<i>g<sub>0</sub>, κ, g1, g2 NC, C<sub>0</sub> - C<sub>7</sub>, shift ω<sup>+</sup>, ξ<sub>0</sub>, ξ<sub>1</sub>, ξ<sub>2</sub>, ξ<sub>3</sub>, a, b, c</i>	Fringe field extent : <i>g</i> = <i>g</i> <sub>0</sub> if <i>κ</i> = 0, otherwise <i>g</i> = <i>g</i> <sub>1</sub> + <i>g</i> <sub>2</sub> <i>R</i> /100. Unused ; <i>C</i> <sub>0</sub> to <i>C</i> <sub>7</sub> : fringe field coefficients ; unused. Azimuth of entrance EFB with respect to <i>ACN</i> ; ξ <sub>0-3</sub> : definition of spiral EFB ; a-c: definition of radial EFB (unused)	cm, 0-1, cm, - E,I,2*E 0-6, 9*- I, 9*E 2*deg,deg/m,deg/m <sup>2</sup> , 8*E deg/m <sup>3</sup> , 3*-
<b>EXIT FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)</b>		
<i>g<sub>0</sub>, κ, g1, g2 NC, C<sub>0</sub> - C<sub>7</sub>, shift ω<sup>-</sup>, ξ<sub>0</sub>, ξ<sub>1</sub>, ξ<sub>2</sub>, ξ<sub>3</sub>, a, b, c</i>	Fringe field parameters, EFB definition and positioning, see above.	cm, 0-1, cm, - E,I,2*E 0-6, 9*- I, 9*E 2*deg,deg/m,deg/m <sup>2</sup> , 8*E deg/m <sup>3</sup> , 3*-
<b>LATERAL FIELD BOUNDARY to be implemented - following data not used</b>		
<i>g<sub>0</sub>, κ NC, C<sub>0</sub> - C<sub>5</sub>, shift ω<sup>-</sup>, ξ, R<sub>1</sub>, U<sub>1</sub>, U<sub>2</sub>, R<sub>2</sub></i>		2*- E, I 8*- 1, 7*E 6*- 6*E
<i>End of repeat</i> _____		
<b>Integration boundaries</b> - next line is optional, starting with string <i>IntLim</i> :		
<i>IntLim, ID, A, B, C [, A', B', C']</i>	Integration boundary. Line has to start with ' <i>IntLim</i> '. <i>ID</i> = -1 : integration in the magnet begins at entrance boundary defined by A, B, C. <i>ID</i> = 1 : integration is terminated at exit boundary defined by A', B', C'. <i>ID</i> = 2 : both entrance and exit boundaries.	-1, 1, 2; deg; cm; I, 3*E deg [; id.] [,3*E]
<i>KIRD, Resol</i>	<i>KIRD</i> = 2, 4 or 25 : numerical interpolation of field derivatives ; size of flying interpolation mesh is <i>XPAS/Resol</i> . <i>KIRD</i> =2 or 25 : second degree, 9- or 25-point grid <i>KIRD</i> =4 : fourth degree, 25-point grid	0, 2, 25 or 4; - I, E
<i>XPAS</i>	Integration step	cm E
<i>KPOS, RE, TE, RS, TS</i>	Positioning of the magnet, has to be 2. As follows : radius and angle of reference, respectively, at entrance and exit of the magnet.	2, 2*(cm, rad) I, 4*E

• **Example : CYCLOTRON**

A model of a 6-sector cyclotron where one varied the spiral angle is illustrated in Fig 54-left :

- in red, the spiral angle is fixed,  $\xi = 30$  deg, while, in blue, the spiral angle increases with the radius,  $\xi = 30 + 0.02 \times R$  deg.
- the tune calculation is shown in Fig 54-right, where one can see that a proper radial increase of the spiral angle allows stabilizing the vertical wave number.

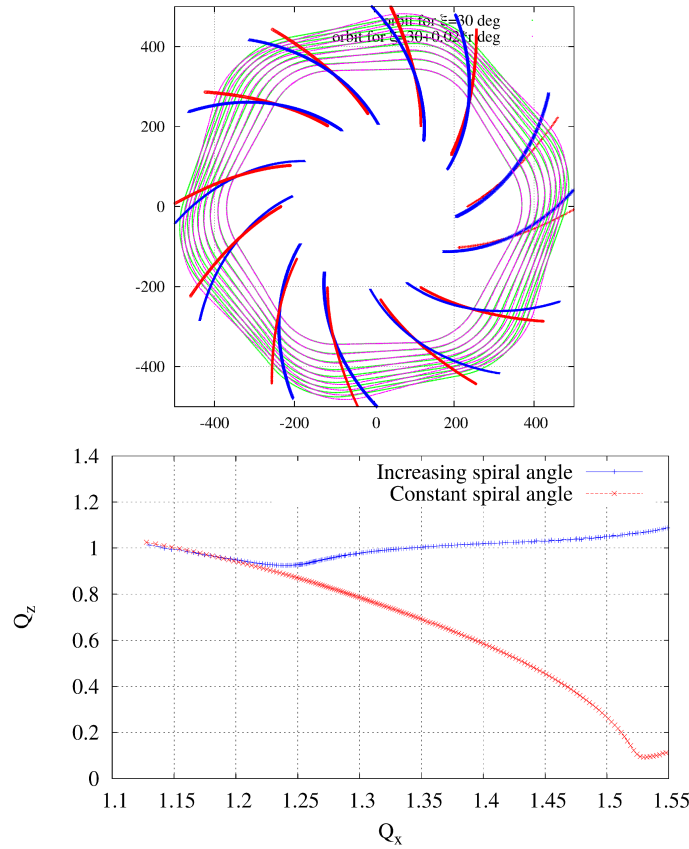


Figure 54: Comparison of cyclotron spiral sector designs: with fixed spiral angle and with varying spiral angle. Left: orbits around the ring; EFB footprints are superimposed, with constant spiral angle (red) or increasing spiral angle with radius (blue). Right: radial and axial wave numbers over the radial span of the orbits, with and without increasing spiral angle (note that  $Q_z$  is close to 1, this requires some more work...)

The input data file below tracks closed orbits for a few different momenta:

```
PSI-style CYCLOTRON spiral sector
'OBJET'
1249.382414 ! Reference rigidity.
2
6 1
268.20121069999999 -15.580272709999999 0. 0. 0. 1.4 'o'
282.46123799999998 -26.403956470000001 0. 0. 0. 1.5 'o'
296.05739569999997 -38.103638570000001 0. 0. 0. 1.6 'o'
309.00757420000002 -50.260491320000000 0. 0. 0. 1.7 'o'
321.32946040000002 -62.556048799999999 0. 0. 0. 1.8 'o'
333.05392440000003 -74.952056110000001 0. 0. 0. 1.9 'o'
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
'PARTICUL'
PROTON

'CYCLOTRON'
2 ! IL=2 to log stepwise integration data in zgoubi.plt.
1 45.0 2.760000E+02 1.0
0. 0. 0.99212277 51.4590015 .5 800. -0.476376328 2.27602517e-03 -4.81955890e-06 3.94715806e-09
18.3000E+00 1. 28. -2.0 ! g0, k, g1, g2 Entrance face.
8 1.10243581, 3.12915071, -3.14287154, 3.0858059, -1.43544992, 0.24047436 0. 0. 0.
11.0 3.5 35.E-3 0.E-4 3.E-8 0. 0. 0. ! Entrance EFB: OMEGA, XI0, XI1, XI2, XI3, a,b,c.
18.3000E+00 1. 28. -2.0 ! lambda=gap, gap's k g10 g11.
8 0.70490173 4.1601305 -4.3309575 3.540416 -1.3472703 0.18261076 0. 0. 0. ! NBCOEF, COEFS_CO-5, SHIPT.
-8.5 2. 12.E-3 75.E-6 0. 0. 0. 0. ! Exit EFB: OMEGA, XI0, XI1, XI2, XI3, a,b,c.
0. -1 ! Lateral face, ignored.
0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0.
2 10. ! Interpolation order to compute derivatives; flying mesh size is step-size/10.
1.0 ! Integration step size.
2
0. 0. 0. 0.
'FAISCEAU'
'END'
```

From zgoubi.res file, final FAISCEAU :

```

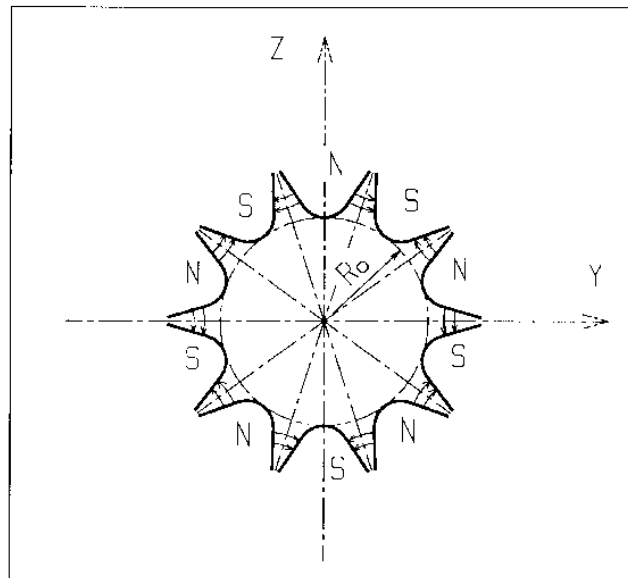
*****
4 Keyword, label(s) : FAISCEAU
*****
TRACE DU FAISCEAU
6 TRAJECTOIRES
OBJET FAISCEAU
D Y (CM) T (MR) Z (CM) P (MR) S (CM) D Y (CM) T (MR) Z (CM) P (MR) S (CM)
o 1 1.4000 268.201 -15.580 0.000 0.000 0.0000 1.4000 268.268 -15.802 0.000 0.000 2.166099E+02 1
Time of flight (mus) : 1.48103721E-02 mass (MeV/c2) : 938.272
o 1 1.5000 282.461 -26.404 0.000 0.000 0.0000 1.5000 282.434 -29.930 0.000 0.000 2.281330E+02 2
Time of flight (mus) : 1.48124804E-02 mass (MeV/c2) : 938.272
o 1 1.6000 296.057 -38.104 0.000 0.000 0.0000 1.6000 295.929 -43.489 0.000 0.000 2.390438E+02 3
Time of flight (mus) : 1.48130438E-02 mass (MeV/c2) : 938.272
o 1 1.7000 309.008 -50.260 0.000 0.000 0.0000 1.7000 308.830 -56.195 0.000 0.000 2.493744E+02 4
Time of flight (mus) : 1.48133252E-02 mass (MeV/c2) : 938.272
o 1 1.8000 321.329 -62.556 0.000 0.000 0.0000 1.8000 321.169 -67.995 0.000 0.000 2.591432E+02 5
Time of flight (mus) : 1.48134077E-02 mass (MeV/c2) : 938.272
o 1 1.9000 333.054 -74.952 0.000 0.000 0.0000 1.9000 332.939 -79.024 0.000 0.000 2.683616E+02 6
Time of flight (mus) : 1.48128373E-02 mass (MeV/c2) : 938.272

Beam characteristics (EMIT,ALP,BET,XM,XPM,NLIV,NINL,RATIN) :
7.9637E-04 75.42 770.0 3.016 -4.8739E-02 6 6 1.000 B-Dim 1 1
0.000 75.42 770.0 0.000 0.000 6 6 1.000 B-Dim 2 1
3.2319E-04 -0.9964 4.1519E-08 1.4813E-02 186.5 6 6 1.000 B-Dim 3 1

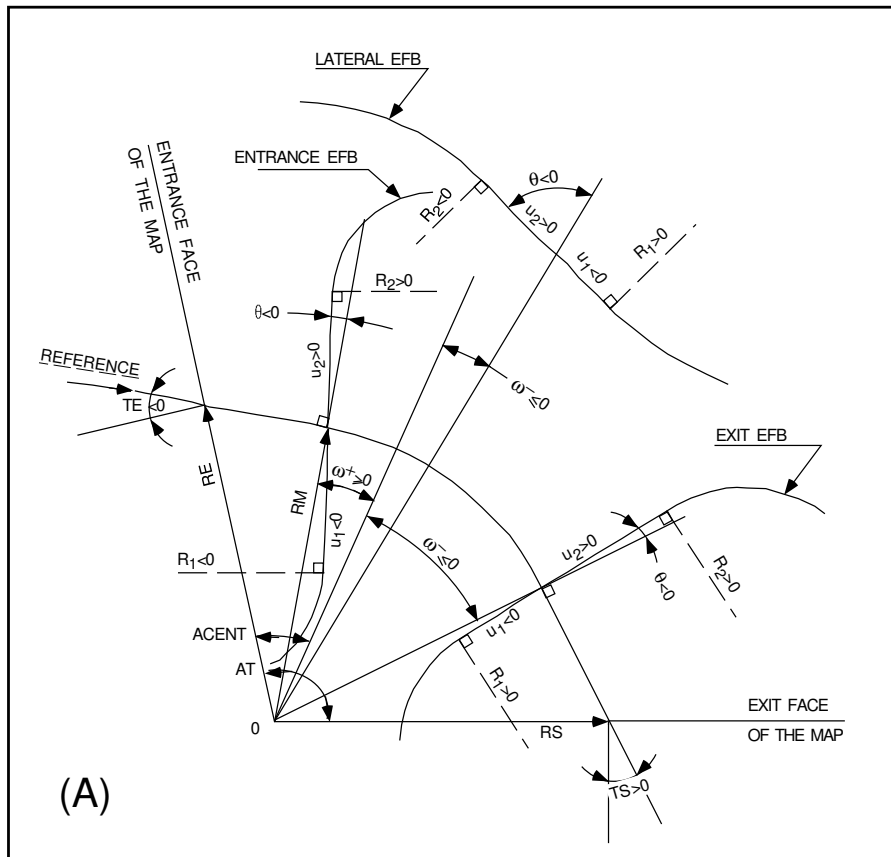
Beam characteristics SIGMA(4,4) :
Ex, Ez = 6.337310E-05 0.000000E+00
AlpX, BetX = -7.542375E+01 7.700024E+02
AlpZ, BetZ = NaN 0.000000E+00
*****

```

<b>DECAPOLE</b>	<b>Decapole magnet</b>	(page 98 in PART A)	
<i>IL</i>	$IL = 1, 2[\times 10^n]$ , 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2 $[\times 10^n]$ , 7	I
<i>XL, R<sub>0</sub>, B<sub>0</sub></i>	Length ; radius and field at pole tip	2*cm, kG	3*E
<i>X<sub>E</sub>, λ<sub>E</sub></i>	Entrance face : Integration zone extent ; fringe field extent ( $\lesssim 2R_0$ , $\lambda_E = 0$ for sharp edge)	2*cm	2*E
<i>NCE, C<sub>0</sub> - C<sub>5</sub></i>	<i>NCE</i> = unused <i>C<sub>0</sub> - C<sub>5</sub></i> = Fringe field coefficients such that $G(s) = G_0/(1 + \exp P(s))$ , with $G_0 = B_0/R_0^4$ and $P(s) = \sum_{i=0}^5 C_i (s/\lambda)^i$	unused, 6*no dim	I, 6*E
<i>X<sub>S</sub>, λ<sub>S</sub></i> <i>NCS, C<sub>0</sub> - C<sub>5</sub></i>	Exit face : see entrance face	2*cm 0-6, 6*no dim	2*E I, 6*E
<i>XPAS</i>	Integration step	cm	E
<i>KPOS, XCE, YCE, ALE</i>	<i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt.	1-2, 2*cm, rad	I, 3*E



<b>DIPOLE</b>		(page 99 in PART A)	
	<b>Dipole magnet, polar frame</b>		
	$B_z = \mathcal{F}B_0 \left( 1 + N \frac{R - RM}{RM} + B \left( \frac{R - RM}{RM} \right)^2 + G \left( \frac{R - RM}{RM} \right)^3 \right)$		
		dim.	data type
			FIT num.
<i>IL</i>	<i>IL</i> = 1, 2[×10 <sup>n</sup> ], 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2[×10 <sup>n</sup> ], 7	I
<i>AT, RM</i>	Total angular extent of the dipole ; reference radius	deg, cm	2*E 2, 3
<i>ACENT, B<sub>0</sub>, N, B, G</i>	Azimuth for positioning of EFBs ; field and field indices at $R \equiv RM$ . ( $N = \frac{RM}{B_0} \frac{dB}{dR}$ ; $B = \frac{RM}{2B_0} \frac{d^2B}{dR^2}$ ; $G = \frac{RM}{6B_0} \frac{d^3B}{dR^3}$ ).	deg., kG, 3*no dim	5*E 4-8
<b>ENTRANCE FIELD BOUNDARY</b>			
$\lambda, \xi$	Fringe field extent (normally $\simeq$ gap size) ; unused. Exponential type fringe field $F = 1 / (1 + \exp(P(s)))$ with $P(s) = C_0 + C_1(\frac{s}{\lambda}) + C_2(\frac{s}{\lambda})^2 + \dots + C_5(\frac{s}{\lambda})^5$	cm, unused	2*E 9, 10
<i>NC, C<sub>0</sub> - C<sub>5</sub>, shift</i>	Unused ; C <sub>0</sub> to C <sub>5</sub> : see above ; EFB shift	0-6, 6*no dim, cm	I, 7*E 11-18
$\omega^+, \theta, R_1, U_1, U_2, R_2$	Azimuth of entrance EFB with respect to <i>ACENT</i> ; wedge angle of EFB ; radii and linear extents of EFB (use $U_{1,2} = \mp\infty$ when $R_{1,2} = \infty$ )	2*deg, 4*cm	6*E 19-24
(Note : $\lambda = 0, \omega^+ = \text{ACENT}$ and $\theta = 0$ for <u>sharp edge</u> )			
<b>EXIT FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)</b>			
$\lambda, \xi$	Fringe field parameters	cm, unused	2*E 25, 26
<i>NC, C<sub>0</sub> - C<sub>5</sub>, shift</i>		0-6, 6*no dim, cm	1, 7*E 27-34
$\omega^-, \theta, R_1, U_1, U_2, R_2$	Positioning and shape of the exit EFB	2*deg, 4*cm	6*E 35-40
(Note : $\lambda = 0, \omega^- = -AT + \text{ACENT}$ and $\theta = 0$ for <u>sharp edge</u> )			
<b>LATERAL FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)</b>			
$\lambda, \xi$	Lateral EFB is inhibited if $\xi = 0$	cm, unused	2*E 41-42
<i>NC, C<sub>0</sub> - C<sub>5</sub>, shift</i>		0-6, 6*no dim, cm	1, 7*E 43-50
$\omega^-, \theta, R_1, U_1, U_2, R_2, RM3$	Positioning and shape of the EFB	2*deg, 5*cm	7*E 51-57
<i>IRDRE, Resol</i>	Degree of interpolation polynomial : 2 = second degree, 9-point grid 25 = second degree, 25-point grid 4 = fourth degree, 25-point grid ; size of flying mesh is <i>XPAS/Resol</i>	2, 25 or 4 ; no dim	I, E 58, 59
<i>XPAS</i>	Integration step	cm	E 60
<i>KPOS</i>	Positioning of the magnet, normally 2. Two options :	1-2	I
<b>If KPOS = 2</b>	Positioning as follows :		
<i>RE, TE, RS, TS</i>	Radius and angle of reference, respectively, at entrance and exit of the magnet.	cm, rad, cm, rad	4*E 64-67
<b>If KPOS = 1</b>	Automatic positioning of the magnet, by means of		
<i>DP</i>	reference relative momentum	no dim	E



Parameters used to define the geometry of a dipole magnet with index, using DIPOLE[S]

<b>DIPOLE-M</b>	<b>Generation of dipole mid-plane 2-D map, polar frame</b>	(page 102 in PART A)	
	$B_Z = \mathcal{F}B_0 \left( 1 + N \frac{R - RM}{RM} + B \left( \frac{R - RM}{RM} \right)^2 + G \left( \frac{R - RM}{RM} \right)^3 \right)$		
<i>NFACE, IC, IL</i>	Number of field boundaries <i>IC</i> = 1, 2 : print field map <i>IL</i> = 1, 2[ $\times 10^n$ ], 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	2-3; 0-2; 0-2[ $\times 10^n$ ], 7 3*I	
<i>IAMAX, IRMAX</i>	Azimuthal and radial number of nodes of the mesh	$\leq 400, \leq 200$	2*I
<i>B<sub>0</sub>, N, B, G</i>	Field and field indices	kG, 3*no dim	4*E
<i>AT, ACENT, RM, RMIN, RMAX</i>	Mesh parameters : total angle of the map ; azimuth for positioning of EFBs ; reference radius ; minimum and maximum radii	2*deg, 3*cm	5*E
ENTRANCE FIELD BOUNDARY			
$\lambda, \xi$	Fringe field extent (normally $\simeq$ gap size) ; unused. Exponential type fringe field $F = 1 / (1 + \exp(P(s)))$ with $P(s) = C_0 + C_1 \left(\frac{s}{\lambda}\right) + C_2 \left(\frac{s}{\lambda}\right)^2 + \dots + C_5 \left(\frac{s}{\lambda}\right)^5$	cm, unused	2*E
<i>NC, C<sub>0</sub> - C<sub>5</sub>, shift</i>	Unused ; <i>C<sub>0</sub></i> to <i>C<sub>5</sub></i> : see above ; EFB shift	0-6, 6*no dim, cm	I, 7*E
$\omega^+, \theta, R_1, U_1, U_2, R_2$	Azimuth of entrance EFB with respect to <i>ACENT</i> ; wedge angle of EFB ; radii and linear extents of EFB (use $U_{1,2} = \mp\infty$ when $R_{1,2} = \infty$ )  (Note : $\lambda = 0, \omega^+ = \text{ACENT}$ and $\theta = 0$ for <u>sharp edge</u> )	2*deg, 4*cm	6*E
EXIT FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)			
$\lambda, \xi$ <i>NC, C<sub>0</sub> - C<sub>5</sub>, shift</i>	Fringe field parameters	cm, unused 0-6, 6*nodim., cm	2*E I, 7*E
$\omega^-, \theta, R_1, U_1, U_2, R_2$	Positioning and shape of the exit EFB  (Note : $\lambda = 0, \omega^- = -AT + \text{ACENT}$ and $\theta = 0$ for <u>sharp edge</u> )	2*deg, 4*cm	6*E
If <b>NFACE = 3</b> LATERAL FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY) Next 3 records <i>only</i> if <i>NFACE</i> = 3			
$\lambda, \xi$ <i>NC, C<sub>0</sub> - C<sub>5</sub>, shift</i>	Fringe field parameters	cm, (cm) 0-6, 6*no dim, cm	2*E I, 7*E
$\omega^-, \theta, R_1, U_1, U_2, R_2,$ <i>RM3</i>	Positioning and shape of the lateral EFB ; <i>RM3</i> is the radial position on azimuth <i>ACENT</i>	2*deg, 5*cm	7*E
<i>NBS</i>	Option index for perturbations to the field map	normally 0	I
<b>If NBS = 0</b>	Normal value. No other record required		
<b>If NBS = -2</b>	The map is modified as follows :		
<i>R<sub>0</sub>, <math>\Delta B/B_0</math></i>	$B$ transforms to $B * \left( 1 + \frac{\Delta B}{B_0} \frac{R - R_0}{RMAX - RMIN} \right)$	cm, no dim	2*E
<b>If NBS = -1</b>	the map is modified as follows :		
$\theta_0, \Delta B/B_0$	$B$ transforms to $B * \left( 1 + \frac{\Delta B}{B_0} \frac{\theta - \theta_0}{AT} \right)$	deg, no dim	2*E

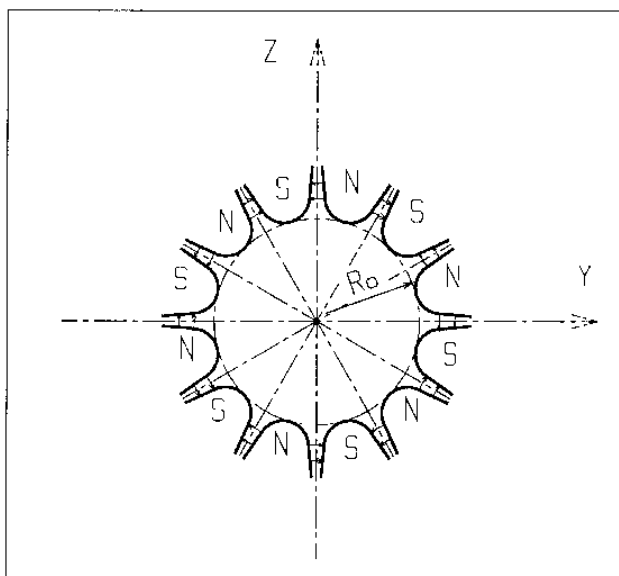


<b>If NBS ≥ 1</b>	Introduction of NBS shims		
<b>For I = 1, NBS</b>	The following 2 records must be repeated NBS times		
$R_1, R_2, \theta_1, \theta_2, \lambda$	Radial and angular limits of the shim ; $\lambda$ is unused	2*cm, 2*deg, cm	5*E
$\gamma, \alpha, \mu, \beta$	geometrical parameters of the shim	2*deg, 2*no dim	4*E
<b>IORPRE</b>	Degree of interpolation polynomial : 2 = second degree, 9-point grid 25 = second degree, 25-point grid 4 = fourth degree, 25-point grid	2, 25 or 4	I
<b>XPAS</b>	Integration step	cm	E
<b>KPOS</b>	Positioning of the map, normally 2. Two options :	1-2	I
<b>If KPOS = 2</b> RE, TE, RS, TS	Positioning as follows : Radius and angle of reference, respectively, at entrance and exit of the map.	cm, rad, cm, rad	4*E
<b>If KPOS = 1</b> DP	Automatic positioning of the map, by means of reference relative momentum	no dim	E

<b>DIPOLES</b>		<b>Dipole magnet <math>N</math>-tuple, polar frame</b>		(page 104 in PART A)	
	Depending on $KIRD[i]$ below:				
	(i) $B_Z = \sum_{i=1}^N B_{Z0,i} \mathcal{F}_i(R, \theta) (1 + b_{1_i}(R - RM_i)/RM_i + b_{2_i}(R - RM_i)^2/RM_i^2 + \dots)$				
	(ii) $B_Z = B_{Z0,i} + \sum_{i=1}^N \mathcal{F}_i(R, \theta) (b_{1_i}(R - RM_i) + b_{2_i}(R - RM_i)^2 + \dots)$				
		dim.		data type	FIT num.
$IL$	$IL = 1, 2[\times 10^n]$ , 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2 $[\times 10^n]$ , 7		I	
$N, AT, RM$	Number of magnets in the $N$ -tuple ; total angular extent of the dipole ; reference radius	no dim, deg, cm		I, 2*E	2-4
<i>Repeat <math>N</math> times the following sequence</i> _____					
$ACN, \delta RM^1, B_0, ind, b_i, (i = 1, ind)$	Positioning of EFBs : azimuth, $RM_i = RM + \delta RM$ ; field ; number of, and field indices at $R \equiv RM$ ( $b_1 = \frac{RM}{B_0} \frac{dB}{dR}$ ; $b_2 = \frac{RM}{2B_0} \frac{d^2B}{dR^2}$ ; etc.).	deg., cm, kG, ( $ind + 1$ )*no dim		3*E, I, ind*E	5-7, 8-(8+ind)
$g_0, \kappa$	<b>ENTRANCE FIELD BOUNDARY</b> Fringe field extent ( $g = g_0 (RM/R)^\kappa$ ) Exponential type fringe field $F = 1 / (1 + \exp(P(s)))$ with $P(s) = C_0 + C_1(\frac{s}{g}) + C_2(\frac{s}{g})^2 + \dots + C_5(\frac{s}{g})^5$	cm, no dim		2*E	(9+ind) to (10+ind)
$NC, C_0 - C_5, \text{shift}$	Unused ; $C_0$ to $C_5$ : see above ; EFB shift	0-6, 6*no dim, cm		I, 7*E	etc.
$\omega^+, \theta, R_1, U_1, U_2, R_2$	Azimuth of entrance EFB with respect to ACN ; wedge angle of EFB; radii and linear extents of EFB (use $U_{1,2} = \mp \infty$ when $R_{1,2} = \infty$ )  (Note : $\lambda = 0, \omega^+ = ACN$ and $\theta = 0$ for <u>sharp edge</u> )	2*deg, 4*cm		6*E	
	<b>EXIT FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)</b>				
$g_0, \kappa$		cm, no dim		2*E	
$NC, C_0 - C_5, \text{shift}$		0 - 6, 6*no dim, cm		1, 7*E	
$\omega^-, \theta, R_1, U_1, U_2, R_2$		2*deg, 4*cm		6*E	
	(Note : $\lambda = 0, \omega^- = -AT + ACN$ and $\theta = 0$ for <u>sharp edge</u> )				
	<b>LATERAL FIELD BOUNDARY to be implemented - following data not used</b>				
$g_0, \kappa$		cm, no dim		2*E	
$NC, C_0 - C_5, \text{shift}$		0-6, 6*no dim, cm		1, 7*E	
$\omega^-, \theta, R_1, U_1, U_2, R_2, RM3$		2*deg, 5*cm		7*E	
<i>End of repeat</i> _____					
$KIRD[i], Resol$	If $KIRD=0$ : analytical computation of field derivatives, and . $i=0$ or absent : default, $B_Z$ formula (i) above, $i=1$ : $B_Z$ formula (ii), . $i=7-9$ : S. Ruggiero's FFAG (see dipsi.f routine), . $Resol = 2/4$ for 2nd/4th order field derivatives computation. If $KIRD=2, 25$ or $4$ : numerical interpolation of field derivatives, and . size of flying interpolation mesh is $XPAS/Resol$ . $KIRD=2$ or $25$ : second degree, 9- or 25-point grid . $KIRD=4$ : fourth degree, 25-point grid	0, 2, 25 or 4 ; no dim		I, E	
$XPAS$	Integration step	cm		E	
$KPOS$	Positioning of the magnet, normally 2. Two options :	1-2		I	
<b>If <math>KPOS = 2</math></b>	Positioning as follows :				
$RE, TE, RS, TS$	Radius and angle of reference, respectively, at entrance and exit of the magnet	cm, rad, cm, rad		4*E	
<b>If <math>KPOS = 1</math></b>	Automatic positioning of the magnet, by means of				
$DP$	reference relative momentum	no dim		E	

<sup>1</sup> Non-zero  $\delta RM$  requires  $KIRD = 2, 4$  or  $25$ .

<b>DODECAPO</b>		<b>Dodecapole magnet</b>	
$IL$	$IL = 1, 2[\times 10^n]$ , 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2 $[\times 10^n]$ , 7	I
$XL, R_0, B_0$	Length ; radius and field at pole tip	2*cm, kG	3*E
$X_E, \lambda_E$	Entrance face : Integration zone extent ; fringe field extent ( $\lesssim 2R_0$ , $\lambda_E = 0$ for sharp edge)	2*cm	2*E
$NCE, C_0 - C_5$	$NCE =$ unused $C_0 - C_5 =$ Fringe field coefficients such that $G(s) = G_0/(1 + \exp P(s))$ , with $G_0 = B_0/R_0^5$ and $P(s) = \sum_{i=0}^5 C_i(s/\lambda)^i$	unused, 6*no dim	I, 6*E
$X_S, \lambda_S$ $NCS, C_0 - C_5$	Exit face : see entrance face	2*cm 0-6, 6*no dim	2*E I, 6*E
$XPAS$	Integration step	cm	E
$KPOS, XCE, YCE, ALE$	$KPOS=1$ : element aligned, 2 : misaligned ; shifts, tilt.	1-2, 2*cm, rad	I, 3*E



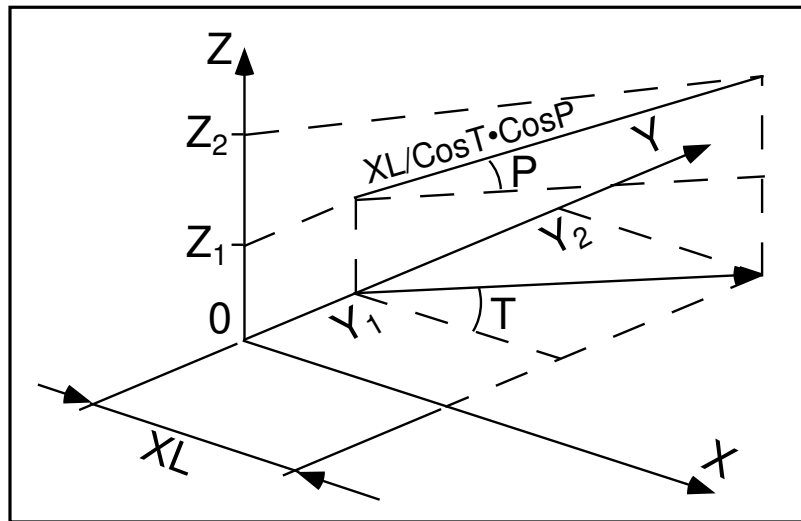
**DRIFT, ESL**

**Field free drift space**

*XL* [, 'split', N [, *IL*]]

length; 'split', optional, will split into N segments ; cm  
*IL*=0, 1 or 2 (0 is default, no logging), optional, causes particle data to be  
 logged in zgoubi.res (*IL*=1) or zgoubi.plt (*IL*=2); *IL*=0 inhibits logging.

E



• **Example : DRIFT**

◇ Default format of *DRIFT*, to push particles 30 cm ahead :

```
'DRIFT'
30.3
```

◇ It can be split into 10, 3 cm long pieces. Optionally, using *IL*=2 will cause particle data print out to zgoubi.plt, at the downstream end of each piece.

```
'DRIFT'
30.3 split 10 2
```

◇ Two equivalent methods to cause particle data logging in zgoubi.plt at the downstream end of a drift. 'DRIFT\_0' is *LABEL1* here, '.plt' is *LABEL2*.

```
'DRIFT' DRIFT_0
30.3 split 1 2
```

```
'DRIFT' DRIFT_0 .plt
30.3
```

<b>EBMULT</b> <sup>1</sup>		<b>Electro-magnetic multipole</b>		(page 110 in PART A)	
		dim.	data type	FIT num.	
<i>IL</i>	$IL = 1, 2[\times 10^n]$ , 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2 $[\times 10^n]$ , 7	I	1	
<i>XL, R<sub>0</sub>, <math>\phi_1, \phi_2, \dots, \phi_{10}</math></i>	<b>Electric poles</b> Length of element ; radius at pole tip ; potential at pole tip for dipole, quadrupole, ..., 20-pole electric components	2*cm, 10*V	12*E	2-13	
<i>X<sub>E</sub>, <math>\lambda_E, E_2, \dots, E_{10}</math></i>	<b>Entrance face</b> Integration zone ; fringe field extent : dipole fringe field extent = $\lambda_E$ ; quadrupole fringe field extent = $\lambda_E * E_2$ ; etc. ; 20-pole fringe field extent = $\lambda_E * E_{10}$ (for any component : sharp edge if field extent is zero)	2*cm, 9*no dim	11*E	14-24	
<i>NCE, C<sub>0</sub> – C<sub>5</sub></i>	same as <i>QUADRUPO</i>	0-6, 6*no dim	I,6*E	25-31	
<i>X<sub>S</sub>, <math>\lambda_S, S_2, \dots, S_{10}</math></i>	<b>Exit face</b> Integration zone ; as for entrance	2*cm, 9*no dim	11*E	32-42	
<i>NCS, C<sub>0</sub> – C<sub>5</sub></i>		0-6, 6*no dim	I, 6*E	43-49	
<i>R1, R2, R3, ..., R10</i>	Skew angles of electric field components	10*rad	10*E	50-59	
<i>XL, R<sub>0</sub>, B1, B2, ..., B10</i>	<b>Magnetic poles</b> Length of element ; radius at pole tip ; field at pole tip for dipole, quadrupole, ..., 20-pole magnetic components	2*cm, 10*kG	12*E	60-71	
<i>X<sub>E</sub>, <math>\lambda_E, E_2, \dots, E_{10}</math></i>	<b>Entrance face</b> Integration zone ; fringe field extent : dipole fringe field extent = $\lambda_E$ ; quadrupole fringe field extent = $\lambda_E * E_2$ ; etc. ; 20-pole fringe field extent = $\lambda_E * E_{10}$ (for any component : sharp edge if field extent is zero)	2*cm, 9*no dim	11*E	72-82	
<i>NCE, C<sub>0</sub> – C<sub>5</sub></i>	same as <i>QUADRUPO</i>	0-6, 6*no dim	I,6*E	83-89	

<sup>1</sup> Use *PARTICUL* to declare mass and charge.

		dim.	data type	FIT num.
$X_S, \lambda_S, S_2, \dots, S_{10}$	<b>Exit face</b> Integration zone ; as for entrance	2*cm, 9*no dim	11*E	90-100
$NCS, C_0 - C_5$		0-6, 6*no dim	I, 6*E	101-107
$R1, R2, R3, \dots, R_{10}$	Skew angles of magnetic field components	10*rad	10*E	108-117
$XPAS$	Integration step	cm	E	118-120
$KPOS, XCE, YCE, ALE$	$KPOS=1$ : element aligned, 2 : misaligned ; shifts, tilt.	1-2, 2*cm, rad	I, 3*E	121-125

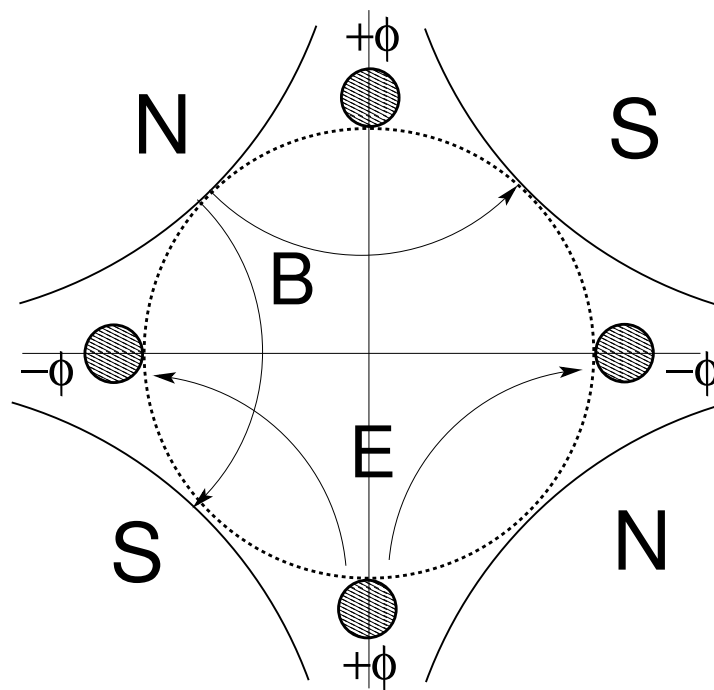
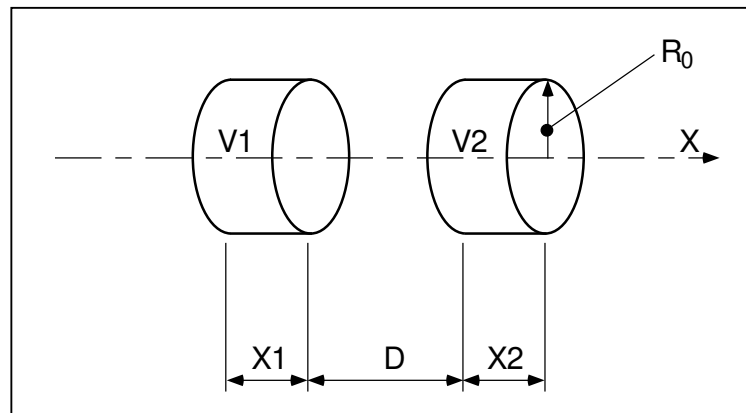


Figure 55: An  $\vec{E}, \vec{B}$  quadrupole.

**EL2TUB**<sup>1</sup>**Two-tube electrostatic lens**

$IL$	$IL = 1, 2[\times 10^n], 7$ : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	$0-2[\times 10^n], 7$	I
$X_1, D, X_2, R_0$	Length of first tube ; distance between tubes ; length of second tube ; inner radius	$3*m$	$4*E$
$V_1, V_2$	Potentials	$2*V$	$2*E$
$XPAS$	Integration step	cm	E
$KPOS, XCE, YCE, ALE$	$KPOS=1$ : element aligned, $2$ : misaligned ; shifts, tilt.	$1-2, 2*cm,$	$I, 3*E$



Two-electrode cylindrical electric lens.

<sup>1</sup> Use *PARTICUL* to declare mass and charge.

ELCYLDEF

Electrostatic cylindrical deflector

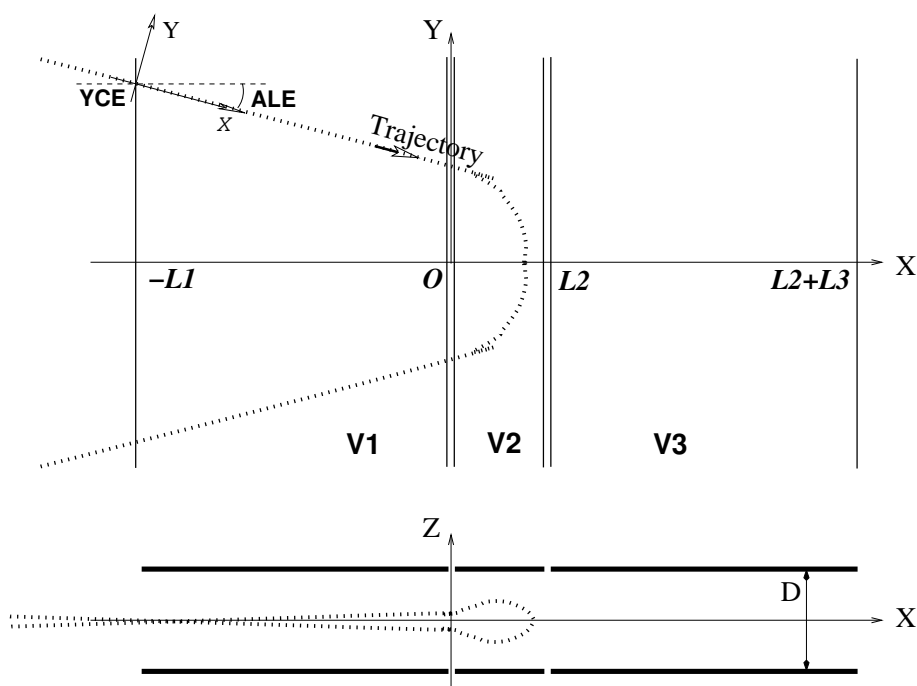
(page 112 in PART A)

## UNDER DEVELOPMENT. To be documented.

$IL$	$IL = 1, 2[\times 10^n]$ , 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2 $[\times 10^n]$ , 7	I
Dev, RM, E, Ind [, MOD [, K2, K3]]	Deviation; reference radius, E-field at RM; index; potential model (1-3: default, Wollnik, Ikegami); 2nd and 3rd order field index (only used if MOD=3).	rad, m, V/m, - [, 1-3 [, -, -]]	4*E [, I [, 2*E]]
$X_E, \lambda_E$	<b>Entrance face :</b> Integration zone extent ; fringe field extent ( $\simeq 2R_0$ , $\lambda_E = 0$ for sharp edge)	2*cm	2*E
$NCE, C_0 - C_5$	$NCE =$ unused $C_0 - C_5 =$ Fringe field coefficients such that $G(s) = G_0/(1 + \exp P(s))$ , with $G_0 = B_0/R_0$ and $P(s) = \sum_{i=0}^5 C_i (s/\lambda)^i$	0-6, 6*no dim	I, 6*E
$X_S, \lambda_S$	<b>Exit face</b> See entrance face	2*cm	2*E
$NCS, C_0 - C_5$		0-6, 6*no dim	I, 6*E
$XPAS$	Integration step	cm	E
$KPOS$	Positioning of the map, normally 2. Two options :	1-2	I
<b>KPOS</b> RE, TE, RS, TS	$KPOS=2$ , positioning as follows : Radius and angle of reference, respectively, at entrance and exit of the deflector (as in DIPOLE, see Fig. B, p. 263).	2 cm, rad, cm, rad	I 4*E

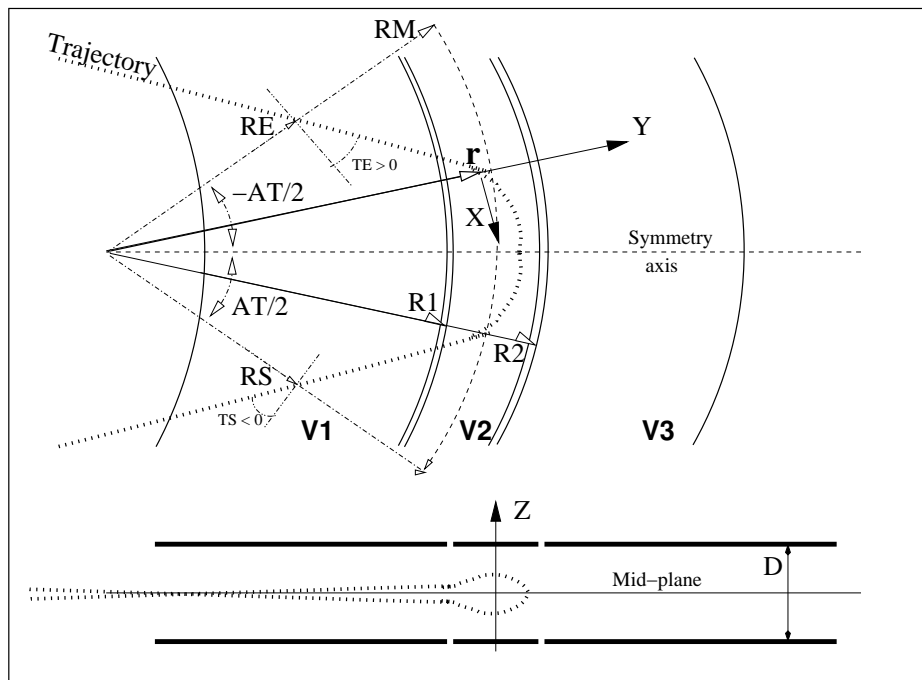


<b>ELMIR</b>	<b>Electrostatic N-electrode mirror/lens, straight slits</b>	(page 113 in PART A)	
<i>IL</i>	$IL = 1, 2[\times 10^n]$ , 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	$0-2[\times 10^n]$ , 7	I
<i>N, L1, ..., LN, D, MT</i>	Number of electrodes ; electrode lengths ; gap ; mode (11/H-mir, 12/V-mir, 21/V-lens, 22/H-lens)	2 – 7, N*m, m	I, N*E, E, I
<i>V1, ..., VN</i>	Electrode potentials (normally $V1 = 0$ )	N*V	N*E
<i>XPAS</i>	Integration step	cm	E
<i>KPOS, XCE, YCE, ALE</i>	<i>KPOS</i> =1 : element aligned ; 2 : misaligned ; shifts, tilt ; 3 : automatic positioning, <i>YCE</i> = pitch, <i>ALE</i> = half-deviation	1-2, 2*cm, rad	I, 3*E



Electrostatic N-electrode mirror/lens, straight slits, in the case  $N = 3$ , in horizontal mirror mode ( $MT = 11$ ). Possible non-zero entrance quantities *YCE*, *ALE* should be specified using *CHANGREF*, or using *KPOS*=3 with *YCE* and *ALE*=half-deviation matched to the reference trajectory.

ELMIRC	Electrostatic N-electrode mirror/lens, circular slits	(page 114 in PART A)	
<i>IL</i>	$IL = 1, 2[\times 10^n]$ , 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2 $[\times 10^n]$ , 7	I
<i>R1, R2, AT, D</i>	Radius of first and second slits ; total deviation angle ; gap	4*m 2*m, rad, m	4*E 4*E
$V - VA, VB - V$	Potential difference	2*V	2*E
<i>XPAS</i>	Integration step	cm	E
<i>KPOS</i>	Normally $KPOS = 2$ for positioning ;	1-2	I
<i>RE, TE, RS, TS</i>	Radius and angle at respectively entrance and exit.	cm, rad, cm, rad	4*E

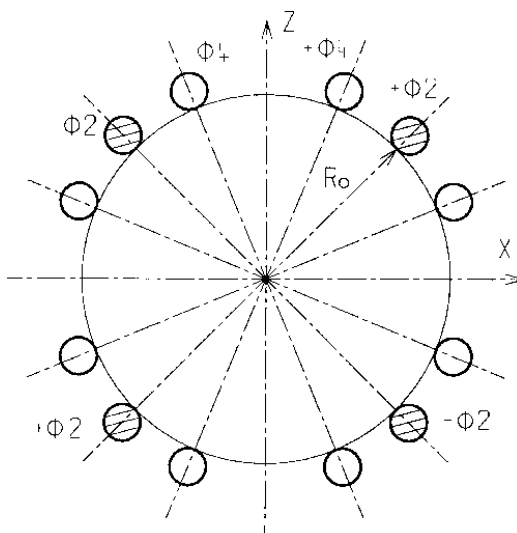


Electrostatic N-electrode mirror/lens, circular slits, in the case  $N = 3$ , in horizontal mirror mode.

**ELMULT**<sup>1</sup>

**Electric multipole**

<i>IL</i>	<i>IL</i> = 1, 2[ $\times 10^n$ ], 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2[ $\times 10^n$ ], 7	I
<i>XL, R<sub>0</sub>, E<sub>1</sub>, E<sub>2</sub>, ..., E<sub>10</sub></i>	Length of element ; radius at pole tip ; field at pole tip for dipole, quadrupole, ..., dodecapole components	2*cm, 10*V/m	12*E
<i>X<sub>E</sub>, λ<sub>E</sub>, E<sub>2</sub>, ..., E<sub>10</sub></i>	<b>Entrance face</b> Integration zone ; fringe field extent : dipole fringe field extent = $\lambda_E$ ; quadrupole fringe field extent = $\lambda_E * E_2$ ; ... 20-pole fringe field extent = $\lambda_E * E_{10}$ (sharp edge if field extent is zero)	2*cm, 9*no dim	11*E
<i>NCE, C<sub>0</sub> – C<sub>5</sub></i>	same as <i>QUADRUPO</i>	0-6, 6*no dim	I, 6*E
<i>X<sub>S</sub>, λ<sub>S</sub>, S<sub>2</sub>, ..., S<sub>10</sub></i>	<b>Exit face</b> Integration zone ; as for entrance	2*cm, 9*no dim	11*E
<i>NCS, C<sub>0</sub> – C<sub>5</sub></i>		0-6, 6*no dim	I, 6*E
<i>R<sub>1</sub>, R<sub>2</sub>, R<sub>3</sub>, ..., R<sub>10</sub></i>	Skew angles of field components	10*rad	10*E
<i>XPAS</i>	Integration step	cm	E
<i>KPOS, XCE, YCE, ALE</i>	<i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt.	1-2, 2*cm, rad	I, 3*E



<sup>1</sup> Use *PARTICUL* to declare mass and charge.

**ELREVOL**<sup>1</sup>**1-D uniform mesh electric field map**

X-axis cylindrical symmetry is assumed

<i>IC, IL</i>	<i>IC</i> = 1, 2 : print the map <i>IL</i> = 1, 2[ $\times 10^n$ ], 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2; 0-2[ $\times 10^n$ ], 7	2*I
<i>ENORM, X-NORM</i>	Field and X-coordinate unit conversion coeff. Convert values as read from map file, to MV/cm and cm units.	2*UnitConv.	2*E
<i>TITL</i>	Title. Start with "FLIP" to get field map X-flipped.		A80
<i>IX</i>	Number of longitudinal nodes of the map	$\leq 400$	I
<i>FNAME</i> <sup>2</sup>	File name		A80
<i>ID, A, B, C</i> [, <i>A', B', C', B''</i> , etc., if <i>ID</i> $\geq 2$ ]	Integration boundary. Ineffective when <i>ID</i> = 0. <i>ID</i> = -1, 1 or $\geq 2$ : as for <i>CARTEMES</i>	$\geq -1$ , 2*no dim, cm [,2*no dim, cm, etc.]	I,3*E [,3*E,etc.]
<i>IODRE</i>	Unused	2, 25 or 4	I
<i>XPAS</i>	Integration step	cm	E
<i>KPOS, XCE, YCE, ALE</i>	<i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt.	1-2, 2*cm, rad	I, 3*E

<sup>1</sup> Use *PARTICUL* to declare mass and charge.<sup>2</sup> *FNAME* (e.g., e-lens.map) contains the field data. These must be formatted according to the following *FORTRAN* sequence :

```

OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD' [,FORM='UNFORMATTED'])
DO 1 I = 1, IX
  IF (BINARY) THEN
    READ(NL) X(I), EX(I)
  ELSE
    READ(NL,*) X(I), EX(I)
  ENDIF
1 CONTINUE

```

where  $X(I)$  and  $EX(I)$  are the longitudinal coordinate and field component at node ( $I$ ) of the mesh. Binary file names *FNAME* must begin with 'B\_' or 'b\_'. 'Binary' will then automatically be set to 'TRUE.'

<b>EMMA</b>	<b>2-D Cartesian or cylindrical mesh field map for EMMA FFAG</b>	(page 118 in PART A)	
<i>IL</i>	<i>IL</i> = 1, 2[ $\times 10^n$ ], 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2[ $\times 10^n$ ], 7	I
<i>BNORM, XN, YN, ZN</i>	Field and X-,Y-,Z-coordinate unit conversion coefficients. Convert values as read from map file, to kG and cm or rad.	4*UnitConv.	4*E
<i>TITL</i>	Title. Start with "FLIP" to get field map X-flipped		A80
<i>IX, IY, IZ, MOD[i]</i>	Number of nodes of the mesh in the <i>X</i> , <i>Y</i> and <i>Z</i> directions, <i>IZ</i> = 1 for single 2-D map ; <i>MOD</i> : operational and map <i>FORMAT</i> reading mode <sup>1</sup> <i>MOD</i> $\leq 19$ : Cartesian mesh ; <i>MOD</i> $\geq 20$ : cylindrical mesh ; .i, optional, tells the reading <i>FORMAT</i> , default is '*'	$\leq 400, \leq 200,$ $1, \geq 0[.1-9]$	3*I
<i>FNAME</i> <sup>1</sup> ( <i>K</i> = 1, <i>NF</i> )	Names of the <i>NF</i> files that contain the 2-D maps, ordered from <i>Z</i> (1) to <i>Z</i> ( <i>NF</i> ). If <i>MOD</i> =0 : a single map, superimposition of QF and QD ones, is built for tracking. If <i>MOD</i> =1 : a single map, <i>interpolated</i> from QF[ <i>x<sub>F</sub></i> ] and QD[ <i>x<sub>D</sub></i> ] ones, is built for tracking. If <i>MOD</i> =22 : a single map, superimposition of QF and QD ones, is built for tracking. If <i>MOD</i> =24 : field at particle is interpolated from a (QF,QD) pair of maps, closest to current ( <i>x<sub>F</sub></i> , <i>x<sub>D</sub></i> ) value, taken from of set of (QF,QD) pairs registered in <i>FNAME</i> ...		A80
<i>ID, A, B, C</i> [, <i>A', B', C', B''</i> , etc., if <i>ID</i> $\geq 2$ ]	Integration boundary. Ineffective when <i>ID</i> = 0. <i>ID</i> = -1, 1 or $\geq 2$ : as for <i>CARTEMES</i>	$\geq -1, 2$ *no dim, cm [,2*no dim, cm, etc.]	I,3*E [,3*E,etc.]
<i>IODRE</i>	If <i>IZ</i> = 1 : as in <i>CARTEMES</i> If <i>IZ</i> $\neq 1$ : unused	2, 25 or 4	I
<i>XPAS</i>	Integration step	cm	E
<i>KPOS, XCE, YCE, ALE</i>	<i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt.	1-2, 2*cm, rad	I, 3*E

<sup>1</sup> *FNAME* normally contains the field map data. If *MOD*=24 *FNAME*(*K*) contains the names of the QF maps and QD maps, as well as the QF-QD distance attached to each one of these pairs.

<b>ERRORS</b>	<b>Injecting errors in optical elements</b> (UNDER DEVELOPMENT)	(page 158 in PART A)
<i>ONF, NBR, SEED</i> [, <i>PRINT</i> ]	On/off switch (1/0) ; number of error sets to be injected ( <i>i.e.</i> , ...,[]) as well, number of lines following this one) ; random seed. Occurrence of <i>PRINT</i> will save error series in zgoubi.ERRORS.out.	I1, I, I [,A5]
<b>The next line depends on the optical element of concern, and is to be one of the following : (only limited possibilities at the moment, under development)</b>		
<i>MULTIPOL</i> {[ <i>LBL1</i> [, <i>LBL2</i> ]}, <i>N, TYP, AR, UG,</i> <i>VC, HW, CUTOFF</i>	Keyword concerned [optionally, first and/or second label ; ,[,],,,,2*kG, wilde cards accepted: *LBL1-2, LBL1-2*] ; N=1-10 : pole concerned if TYP=BP or =XR, otherwise ignored ; TYP= BP : field at pole/kG (dipole to 20-pole), or XR : pole roll-angle/rad (dipole to 20-pole), or ZS : vertical shift of MULTIPOL/cm; AR=A or R : error value is absolute or relative (to current one) ; UG=U or G : uniform or Gaussian random law ; VC= central value ; HW= half-width (case UG=U) or sigma (case UG=G) ; cut-off value in units of sigma (unused if UG=U).	A8 [,A20[,A20]], I, A2, A1, A1, 3*E
<i>TOSCA</i> {[ <i>LBL1</i> [, <i>LBL2</i> ]}, <i>N, TYP, AR, UG,</i> <i>VC, HW, CUTOFF</i>	Keyword concerned [optionally, first and/or second label] ; ,[,],,,,2*kG, N is unused ; TYP= BP : field coefficient <i>BNORM</i> ; AR, UG, etc. : see above.	A8 [,A20[,A20]], I, A2, A1, A1, 3*E

#### • Example : ERRORS

```
' ERRORS'
1 5 123466                !                cut-off
MULTIPOL{KCV}  1  BP R U  0.d0                1e-3        0 ! cutoff unused (U)
MULTIPOL{B}    2  XR A U  0.d0                0.349e-3      0
MULTIPOL{*_QF} 1  XR A U  1.57079632679 0.          0
MULTIPOL{*_QF} 1  BP A U  0.                  0.004         0
TOSCA          1  BP A G -1.d-3              1.e-5         3 ! cutoff at 3-sigma
```

In this example the various attributes of the error keyword take the following values and meanings :

◇ 1st line :

- *ONF* = 1 : error setting is on ; *NBR*=3 : three error setting lines follow ; *seed* = 123456

◇ 2nd line :

- all *MULTIPOL* keywords with first label “KCV” in the optical sequence are concerned ; multipole component ip=1 is affected, *i.e.*, dipole component ; “BP, R, U, 0.d0, 1e-3, 0” : a random relative defect  $dB/B$  (AR=R) concerning the field  $B_{ip}$  (TYP=BP) for that component, will be sorted in a uniform (UG=U) distribution centered on  $V_{ip} = 0$  (VC=0.d0) with half-width  $HW = 10^{-3}$ , CUTOFF at n-sigma (n=0 here) is unused (only used for Gaussian distributions, case UG=G) -  $B_{ip}$  is changed to  $B_{ip} + V_{ip} + B_{ip} \times dB/B$ .

◇ 3rd line :

- all *MULTIPOL* keywords with first label “B” in the optical sequence are concerned ; multipole component ip=2 is affected, *i.e.*, quadrupole component ; “XR, A, U, 0.d0, 0.349e-3, 0” : a random absolute defect  $dR_{ip}$  (AR=A) of the roll-angle  $R_{ip}$  (TYP=XR) for that component, will be sorted in a uniform (UG=U) distribution centered on  $V_{ip} = 0$  (VC=0.d0) with half-width  $HW = 0.349 \times 10^{-3}$  rad, CUTOFF at n-sigma (n=0 here) is unused (only used for Gaussian distributions, case UG=G) -  $R_{ip}$  is changed to  $R_{ip} + V_{ip} + dR_{ip}$ .

◇ 4th and 5th lines :

- all *MULTIPOL* keywords in the optical sequence with first label of the form “\*\_QF” are concerned (\* is a wilde card). Their dipole component is applied a  $\pi/2$  skewing (4th line), and that skewed dipole field is given a value at random, uniform, within [-0.004,0.004] kG (5th line).

◇ 6th line :

- all *TOSCA* keywords in the optical sequence are concerned ; field map field coefficient *BNORM* (see TOSCA keyword) is affected ; BP, A, G, -1.d-3, 1.e-5, 3 : a random absolute defect  $dB_N$  (AR=A) concerning *BNORM* (TYP=BP) will be sorted in a Gaussian (UG=G) distribution centered on  $BNC = -10^{-3}$  (VC=-1.d-3) with rms value  $HW = 10^{-5}$ , CUTOFF 3-sigma - *BNORM* is changed to  $BNC + dB_N$  (not to  $BNORM + BNC + dB_N$  !). (If AR=R instead : random relative defect  $dB_N/B_N$  generated, and *BNORM* changed to  $VC + dB_N/B_N$ ).

**FAISCEAU [, FORCE]    Print particle coordinates**

Printout (in zgoubi.res) particle coordinates at the location where the keyword is introduced in the structure.

If first label is 'FORCE', will force printout when normally inhibited (e.g., by *OPTIONS[WRITE]* or *REBELOTE[KWRIT]*)

**FAISCNL                    Store particle coordinates in file FNAME**

*FNAME*<sup>1</sup>                    Name of storage file                    A80  
(e.g., zgoubi.fai, or b\_zgoubi.fai for binary storage).

**FAISTORE                    Store coordinates every *IP* other pass [, at elements with appropriate label]**

*FNAME*<sup>1</sup>                    Name of storage file (e.g. zgoubi.fai). Optional : a list of up to 10 labels,                    A80,  
[,*LABEL(s)*]                    (the first label of element(s) at the exit of which the store should occur) ;                    [, 0-10\*A20]  
wild card accepted, in the form '\**LABEL*' or '*LABEL*\*' or '\**LABEL*\*'.  
If either *FNAME* or first *LABEL* in the list is 'none' then *FAISTORE* is inhibited. Store occurs at all elements if the list begins with *LABEL*=all or ALL.

*IP*                            Store every *IP* other pass (when using *REBELOTE* with  $NPASS \geq IP - 1$ ).                    I

<sup>1</sup> Stored data can be read back from *FNAME* using *OBJET*, *KOBJ* = 3.

<b>FFAG</b>	<b>FFAG magnet, <math>N</math>-tuple</b> <b>UNDER DEVELOPMENT</b> $B_Z = \sum_{i=1}^N B_{Z0,i} \mathcal{F}_i(R, \theta) (R/R_{M,i})^{K_i}$	(page 119 in PART A)	
<i>IL</i>	$IL = 1, 2[\times 10^n]$ , 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2 $[\times 10^n]$ , 7	I
$N, AT, RM$	Number of dipoles in the FFAG $N$ -tuple ; total angular extent of the dipole ; reference radius	no dim, deg, cm	I, 2*E
<i>Repeat <math>N</math> times the following sequence</i> _____			
$ACN, \delta RM,$ $B_{Z0}, K$	Azimuth for dipole positioning ; $R_{M,i} = RM + \delta RM$ ; field at $R_{M,i}$ ; index	deg, cm, kG, no dim	4*E
<b>ENTRANCE FIELD BOUNDARY</b>			
$g_0, \kappa$ $NC, C_0 - C_5$ , shift $\omega^+, \theta, R_1, U_1, U_2, R_2$	Fringe field extent ( $g = g_0 (RM/R)^\kappa$ ) Unused ; $C_0$ to $C_5$ : fringe field coefficients ; EFB shift Azimuth of entrance EFB with respect to $ACN$ ; wedge angle of EFB ; radii and linear extents of EFB (use $U_{1,2} = \mp\infty$ when $R_{1,2} = \infty$ )	cm, no dim 0-6, 6*no dim, cm 2*deg, 4*cm	2*E I, 7*E 6*E
(Note : $g_0 = 0, \omega^+ = ACN, \theta = 0$ and KIRD=0 for <u>sharp edge</u> )			
<b>EXIT FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)</b>			
$g_0, \kappa$ $NC, C_0 - C_5$ , shift $\omega^-, \theta, R_1, U_1, U_2, R_2$	Fringe field parameters, see above	cm, no dim 0-6, 6*no dim, cm 2*deg, 4*cm	2*E I, 7*E 6*E
(Note : $g_0 = 0, \omega^- = -AT + ACN, \theta = 0$ and KIRD=0 for <u>sharp edge</u> )			
<b>LATERAL FIELD BOUNDARY to be implemented - following data not used</b>			
$g_0, \kappa$ $NC, C_0 - C_5$ , shift $\omega^-, \theta, R_1, U_1, U_2, R_2$		cm, no dim 0-6, 6*no dim, cm 2*deg, 4*cm	2*E I, 7*E 6*E
<i>End of repeat</i> _____			
$KIRD, Resol$	If KIRD=0 : analytical computation of field derivatives ; Resol = 2/4 for 2nd/4th order field derivatives computation If KIRD = 2, 4 or 25 : numerical interpolation of field derivatives ; size of flying interpolation mesh is $XPAS/Resol$ KIRD=2 or 25 : second degree, 9- or 25-point grid KIRD=4 : fourth degree, 25-point grid	0, 2, 25 or 4 ; no dim	I, E
$XPAS$	Integration step	cm	E
$KPOS$	Positioning of the magnet, normally 2. Two options :	1-2	I
<b>If <math>KPOS = 2</math></b> $RE, TE, RS, TS$	Positioning as follows : Radius and angle of reference, respectively, at entrance and exit of the magnet	cm, rad, cm, rad	4*E
<b>If <math>KPOS = 1</math></b> $DP$	Automatic positioning of the magnet, by means of reference relative momentum	no dim	E



<b>FFAG-SPI</b>	<b>Spiral FFAG magnet, <math>N</math>-tuple UNDER DEVELOPMENT</b>		
	$B_Z = \sum_{i=1}^N B_{Z0,i} \mathcal{F}_i(R, \theta) (R/R_{M,i})^{K_i}$		
$IL$	$IL = 1, 2[\times 10^n]$ , 7 : print coordinates along trajectories, fields, etc., into zgoubi.res (1) or zgoubi.plt ( $2[\times 10^n]$ ) or zgoubi.impdev.out (7).	0-2 $[\times 10^n]$ , 7	I
$N, AT, RM$	Number of dipoles in the FFAG $N$ -tuple ; total angular extent of the dipole ; reference radius.	no dim, deg, cm	I, 2*E
<i>Repeat <math>N</math> times the following sequence</i> _____			
$ACN, \delta RM,$ $B_{Z0}, K$	Azimuth for dipole positioning ; $R_{M,i} = RM + \delta RM$ ; field at $R_{M,i}$ ; index.	deg, cm, kG, no dim	4*E
<b>ENTRANCE FIELD BOUNDARY</b>			
$g_0, \kappa$ $NC, C_0 - C_5$ , shift $\omega^+, \xi$ , 4 dummies	Fringe field extent ( $g = g_0 (RM/R)^\kappa$ ) Unused ; $C_0$ to $C_5$ : fringe field coefficients ; EFB shift Azimuth of entrance EFB with respect to $ACN$ ; spiral angle ; 4 $\times$ unused.	cm, no dim 0-6, 6*no dim, cm 2*deg, 4*unused	2*E I, 7*E 6*E
<b>EXIT FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)</b>			
$g_0, \kappa$ $NC, C_0 - C_5$ , shift $\omega^-, \xi$ , 4 dummies	Fringe field parameters, see above	cm, no dim 0-6, 6*no dim, cm 2*deg, 4*unused	2*E I, 7*E 6*E
<b>LATERAL FIELD BOUNDARY to be implemented - following data not used</b>			
$g_0, \kappa$ $NC, C_0 - C_5$ , shift $\omega^-, \theta, R_1, U_1, U_2, R_2$		cm, no dim 0-6, 6*no dim, cm 2*deg, 4*cm	2*E I, 7*E 6*E
<i>End of repeat</i> _____			
<b>Integration boundaries</b> - next line is optional, starting with string IntLim :			
IntLim, $ID, A, B, C$ [, $A', B', C'$ ]	Integration boundary. Line has to start with 'IntLim'. $ID = -1$ : integration in the magnet begins at entrance boundary defined by A, B, C. $ID = 1$ : integration is terminated at exit boundary defined by $A', B', C'$ . $ID = 2$ : both entrance and exit boundaries.	-1, 1, 2; deg; cm; deg [; id.]	I, 3*E [,3*E]
$KIRD, Resol$ [, $DNEWT$ ]	If $KIRD=0$ : analytical computation of field derivatives ; $Resol = 2/4$ for 2nd/4th order field derivatives computation. If $KIRD = 2, 4$ or 25 : numerical interpolation of field derivatives ; size of flying interpolation mesh is $XPAS/Resol$ . $KIRD=2$ or 25 : second degree, 9- or 25-point grid $KIRD=4$ : fourth degree, 25-point grid If $DNEWT$ is added, the distance to the magnet edge is calculated numerically (in the case $KIRD \neq 0$ ). In its absence, and if $\kappa = -1$ , then the distance in terms of generalised azimuthal angle is used instead.	0, 2, 25 or 4 ; no dim	I, E
$XPAS$	Integration step	cm	E
$KPOS,$ $RE, TE, RS, TS$	Positioning of the magnet, has to be 2. As follows : radius and angle of reference, respectively, at entrance and exit of the magnet.	2, 2*(cm, rad)	I, 4*E

**FIN, END****End of input data list**

Any information in zgoubi.dat following these keywords will be ignored

**FINISH**

**End of execution**

Execution of zgoubi problem will stop here.

<b>FIT, FIT2</b>	<b>Fitting procedure</b>	(page 160 in PART A)
<i>NV</i> [, <i>nofinal</i> ] [, <i>noSYSout</i> ] [, <i>save</i> [, <i>FileName</i> ]]	<i>NV</i> : Number of physical parameters to be varied ; 'nofinal' avoids final run (default is : final run performed using fitted values, once fit is done) ; 'noSYSout' inhibits system output of variable and constraint status, except for start and end, and update of penalty value in between ; 'save' saves fit variables when fit is completed, either in 'FileName' if specified, or by default in <i>zgoubi.FITVALS.out</i> .	$\leq 20$ [, <i>nofinal</i> ] <i>save</i> [string] I [, A7] [, A4 [, A80]]
<b>For I = 1, NV</b> indifferently, either : <i>IR, IP, XC, DV</i>	<i>repeat NV times the following sequence</i>  Number of the element in the structure (see p. 161) ; number of the physical parameter in the element ; coupling switch of the form <i>int1.int2</i> , with <i>int1.int2</i> < 0 if coupled variation is to be with opposite signs, or <i>int1.int2</i> > 0 for same sign variation; relative variation range ( $\pm$ ).	$\leq \text{MXL}^1, \leq \text{MXD}^1,$ $\pm \text{MXL.MXD}^2,$ relative or [ <i>v1,v2</i> ] 2*I, 2*E
or : <i>IR, IP, XC, [V<sub>min</sub>, V<sub>max</sub>]</i>	<i>IR, IP, XC</i> as above; <i>V<sub>min</sub>, V<sub>max</sub></i> : lower and upper limits of the variable, absolute <sup>3</sup> .	2*I, 3*E
<i>NC</i> [, <i>Penalty</i> [, <i>ITER</i> ]] <sup>4</sup> <b>For I = 1, NC</b> <i>IC</i> [, <i>JC</i> ], <i>I, J, IR, V</i> <sup>3</sup> , <i>WV, NP</i> [, <i>p<sub>i</sub></i> ( <i>i</i> = 1, <i>NP</i> )]	Number of constraints [, <i>penalty</i> [, max. numb. of iterations]]. <i>repeat NC times the following sequence :</i> <i>IC, I</i> and <i>J</i> define the type of constraint (see table below) ; <i>IR</i> : number of the element after which the constraint applies, or <i>IR</i> = #End to apply at end of the sequence ( <i>i.e.</i> , just before <i>FIT[2]</i> ) ; <i>V</i> : value ; <i>W</i> : weight (the stronger the lower <i>WV</i> ) ; <i>NP</i> : number of parameters, and, <i>iff NP</i> $\geq 1$ then provide <i>N</i> parameter values <i>p<sub>i</sub></i> ( <i>i</i> = 1, <i>NP</i> ).	$\leq 20$ [, $10^{-n}$ [, > 0]] I [, E [, I]]  0-5, 3*(>0), I[,I1], 3*I, current unit, 2*E, I, NP*E 2*no dim., curr. units

<sup>1</sup> The values for the maximum number of elements (*i.e.*, keywords in *zgoubi.dat*) that a sequence in **zgoubi** can contain, *MXL*, and for the maximum number of parameters under a keyword, *MXD*, are set in the include file *MXLD.H*.

<sup>2</sup> The coupling input data *XC* is of the form "integer1.integer2" with integer1  $\leq$  *MXL* and integer2  $\leq$  *MXD*. *MXL* and *MXD* are the maximum number of elements, and data in an optical element, defined in *MXLD.H* include file: can be increased there, if necessary, then requires re-compiling.

<sup>3</sup> *V* is in current **zgoubi** units in the case of particle coordinates (*i.e.*, cm, mrad,  $\mu$ s, momentum relative to BORO) and B field (kG). It is in MKSA units (m, rad) in the case of matrix coefficients.

<sup>4</sup> *FIT[2]* will stop when the sum of the squared residuals gets < *penalty*, or when the maximum allowed number of iterations is reached.

### • Examples : Combining FIT and REBELOTE

#### Combining FIT and REBELOTE, Example 1:

In the example below, *FIT* requests that (i) the particle trajectory with initial coordinates defined by *OBJET* have identical horizontal coordinates at both ends of the snake (this is achieved by varying  $Y_0$ ,  $T_0$  in *OBJET*), and that (ii) the trajectory across the snake - an helix - be  $Y$ -centered along the snake axis.

The way this works :

*FIT* is executed a first time for  $B\rho_{ref} = BORO = 7205.1782956$  kG.cm, namely the execution loops between *OBJET* and *FIT* until the constraints are best fulfilled. Once this is completed, the execution pointer then goes to the next instruction in the data list, namely '*SPNPRT*', this is discussed below, and then points to *REBELOTE*, which will have the effect of sending it back to the beginning of *zgoubi.dat* data list. However, prior to that, the "*IOPT=1*" flag, fourth data in first row in *REBELOTE*, requests a change of, next line, parameter number 35 in element "*OBJET*" (that is, the relative rigidity of the particle,  $D$ ). The list of values follows, namely,  $\{1.3872739973 \ 2.1368296674 \ 4.8261190694 \ 11.015241321\}$ . The change occurs **4** times, according to *NRBLT=4*, the first data in *REBELOTE* data list.

Note that, (i) spin data so computed (spin vector components, precession, etc) are stored/stacked in *zgoubi.SPNPRT.Out* by placing the keyword '*SPNPRT*' with label "*PRINT*", between *FIT* and *REBELOTE* ; (ii) current updated *FIT* variables are saved in *zgoubi.FITVALS.out* using the 'save' command in *FIT*, and further stacked in *zgoubi.FITVALS.out\_cat* for each *REBELOTE* case, using *SYSTEM*.

The AGS warm snake 3-D OPERA field map used here, *table55.map*, can be found here:

<https://sourceforge.net/p/zgoubi/code/HEAD/tree/trunk/examples/AGS/snakeFieldMaps/Wsnk3D/>.

```
Centering 5 helical orbits in the AGS warm helical snake 3-D OPERA map.
'OBJET'          ! This data list may be copy-pasted and run, as is.
7.2051782956D3   ! Reference rigidity of the problem
2
1 1              ! A single particle. Initial coordinates :
-2.2  0.  0.  0.  0.  1.  'o'          ! Yo, To, Zo, Po, So, p/po
1
'PARTICUL'      ! proton data are necessary for spin tracking
938.27203 1.602176487E-19 1.7928474 0 0 ! M, Q, G factor
'SPNTRK'
4.1             ! Initial spin is positionned vertical
0.  0.  1.

'FAISCEAU'
'SPNPRT'

'TOSCA'
0 20
1.e1 100. 100. 100.
HEADER_0 wsnake
801 29 29 12.1 ! The map is a 801x29x29 node 3-D mesh.
table55.tab    ! AGS warm snake 3-D OPERA field map.
0 0 0 0
2
.1
2 0.  .0  0.  0.

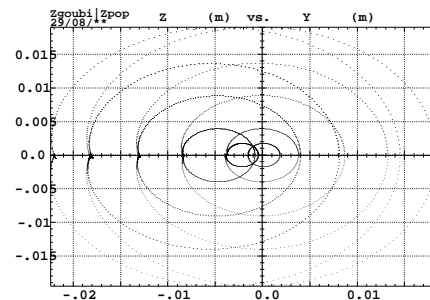
'FAISCEAU'

'FIT'
2 save          ! Two variables. Save FIT variables (in zgoubi.FITVALS.out).
1 30 0 [-3,3]   ! Vary initial coordinate Y_0 (horiz. position)
1 31 0 [-3,3]   ! Vary initial coordinate T_0 (horiz. angle)
3 1E-2         ! Three constraints (penalty 2E-3 requested) :
3.1 1 2 6 0. 1. 0 ! Y_0=Y after at exit of the magnet
3.1 1 3 6 0. 1. 0 ! T_0=T after at exit of the magnet
7.3 1 2 6 0. 1. 0 ! Y_min+Y_max=0 inside the OPERA field map

'SPNPRT' PRINT ! Stack spin data (in zgoubi.SPNPRT.Out).
'SYSTEM'      ! Save zgoubi.FITVALS.out data following from successive REBELOTE.
1
cat zgoubi.FITVALS.out >> zgoubi.FITVALS.out_cat

'REBELOTE'    ! Will loop on re-doing the FIT for 4 additional particle rigidities
4 0.1 0 1
1             ! List of 4 successive values of D in OBJET follows
OBJET 35 1.3872739973 2.1368296674 4.8261190694 11.015241321

'SYSTEM'      ! Save a copy of zgoubi.FITVALS.out_cat and of zgoubi.SPNPRT.Out_cat
2
\cp zgoubi.FITVALS.out_cat zgoubi.FITVALS.out_cat_copy
\cp zgoubi.SPNPRT.Out      zgoubi.SPNPRT.Out_copy
'END'
```



This figure shows how, using *REBELOTE*, the helical trajectories at five different momenta are moved by the *FIT* procedure, one after the other, from initial off-centered position to final centering on the snake axis (at  $Y=Z=0$ ).

Type of constraint	Parameters defining the constraints (see also footnotes)					Additional parameter(s) NP Param. values, $pr_1 - pr_{NP}$	Recommended [MC]OBJET (not an exclusive list) ; comments
	IC	I	J	Constraint (all are defined and managed in ff.f)			
<b>Transported <math>\sigma</math>-matrix</b> ( $\sigma(s) = T\sigma(0)\bar{T}$ )	0	1-6	1-6	$\sigma_{IJ}$ ( $\sigma_{11} = \beta_Y, \sigma_{21, 12} = \alpha_Y$ , etc.)			OBJET[KOBJ=5.1]
<b>Periodic <math>\sigma</math>-matrix</b> ( $\sigma = I \cos \mu + J \sin \mu$ ), <b>tunes</b> ( $N = 1 - 9$ for MATRIX block 1-9)	0.N ( $N \leq 9$ )	1-6 7 8 9 10	1-6 [0, 1] [0, 1] [-1, 1] [-1, 1]	$\sigma_{IJ}$ ( $\sigma_{11} = \cos \mu_Y + \alpha_Y \sin \mu_Y$ , etc.) $\mu_Y / 2\pi$ $\mu_Z / 2\pi$ $\cos(\mu_Y)$ $\cos(\mu_Z)$			OBJET[KOBJ=5] if $N = 1$ , OBJET[KOBJ=5.N] if $N \geq 2$ .
<b>First order transport coeffs. &amp; determinants.</b>	1	1-6 7 8	1-6 j j	Transport coeff. $R_{IJ}$ $j \neq 8$ : YY-determinant ; $j=8$ : YZ-det. $j \neq 7$ : ZZ-determinant ; $j=7$ : ZY-det.			OBJET[KOBJ=5]
<b>Chromas.</b>	1.1	7 8	7 8	$dQ_Y / dp/p$ $dQ_Z / dp/p$			OBJET[KOBJ=5.2] or KOBJ=5.3
<b>Second order transport coeffs.</b>	2	1-6	11-66	Transport coeff. $T_{I,j,k}$ ( $j = [J/10], k = J - 10[J/10]$ )			OBJET[KOBJ=6]
<b>Trajectory coordinates</b>  (I = particle number; J=1-7 for D,Y,T,Z,P,S,time)	3	1-IMAX -1 -2 -3 -4	1-7 1-7 1-7 1-7 1-7	$F(J, I)$ $\langle F(J, i) \rangle_{i=1}^{IMAX}$ if NP=0 $i=1, 12$ if NP=2 $Sup( F(J, i) )_{i=1, IMAX}$ $Dist [F(J, I)_{i=1, 12}]$ $Dist [PU_i, i = 1, N]$	0 [or 2]	$[I_1, I_2]$	[MC]OBJET $[1 \leq I_1 \leq I_2 \leq IMAX]$
	3.1	1-IMAX	1-7	$ F(J, I) - FO(J, I) $	3	$I_1, I_2, \Delta I$	$1 \leq I_1 \leq I_2 \leq IMAX$
	3.2	1-IMAX	1-7	$ F(J, I) + FO(J, I) $	2	NOEL <sub>A</sub> , NOEL <sub>B</sub>	PU range
	3.4	1-IMAX	1-7	$ F(J, I) - F(J, K) $	1	K	$K \leq IMAX$
	3.5	1-IMAX	1-7	$(F(J, I) - F(J, K)) / F(J, K)$	1	K	$K \leq IMAX$
<b>Ellipse parameters</b>	4	1-6	1-6	$\sigma_{IJ}$ ( $\sigma_{11} = \beta_Y$ , $\sigma_{12} = \sigma_{21} = \alpha_Y$ , etc.)			OBJET[KOBJ=8] ; MCOBJET[KOBJ=3]
<b>Number of particles</b>	5	-1 1-3 4-6	$\leq 1$ $\leq 1$ $\leq 1$	$N_{survived} / IMAX$ $N_{in \epsilon_{Y,Z,X}} / N_{survived}$ $N_{in best \epsilon_{Y,Z,X,rms}} / N_{survived}$	1	$\epsilon_{Y,Z,X} / \pi$	OBJET MCOBJET MCOBJET
<b>Coordinates &amp; fields, across optical elements</b> (J=1, 2, 3 for $B_X, Y, Z$ )	7.1	1-IMAX	1-7	min. (param.=1) or max. (2) of $F(J, I)$	1 [or 3]	1 or 2 [, X1, X2]	[MC]OBJET
	7.2	1-IMAX	1-7	$\max(F(J, I)) - \min(F(J, I))$	0 [or 2]	[X1, X2]	
	7.3	1-IMAX	1-7	$\min(F(J, I)) + \max(F(J, I))$	id.	id.	
	7.6	1-IMAX	1-3	min. or max. value of $B_J$	1 [or 3]	1 or 2 [, X1, X2]	
	7.7	1-IMAX	1-3	$\max(B_J) - \min(B_J)$	0 [or 2]	[X1, X2]	
	7.8	1-IMAX	1-3	$\min(B_J) + \max(B_J)$	id.	id.	
	7.9	1-IMAX	1-3	$\int B_J ds$	id.	id.	
	7.10	1-IMAX	1-7	to be documented	id.	id.	
<b>Spin</b>	10	1-IMAX	1-4	$S_{X,Y,Z}(I),  \vec{S}(I) $			SPNTRK+
	10.1	1-IMAX	1-3	$ S_{X,Y,Z}(I) - SO_{X,Y,Z}(I) $			[MC]OBJET
	10.2	1-IDMAX	any	spin precession angle ( $[-\pi, \pi]$ )			OBJET[KOBJ=2]
	10.3	1-IDMAX	1-3	X, Y or Z component of precession axis	0 [or 1]	[1]	OBJET[KOBJ=1,2]
	10.4	1-IDMAX	any	fractional spin tune			OBJET[KOBJ=2]
	10.5	1-IMAX	1-3	$\langle S_{X,Y,Z} \rangle_{N=1, I \text{ particles}}$			OBJET[KOBJ=2]

◇ IMAX: total number of particles tracked.

◇  $F(J, I)$  : particle coordinate array ; J: coordinate (1-7 for respectively D, Y, T, Z, P, S, time, resp. units: none, cm, mrad, cm, mrad, cm,  $\mu s$ ) ; I: particle number.

◇  $FO(J, I)$  : initial particle coordinates.

◇  $S(J, I)$  : spin coordinate array ; J: coordinate (1, 2, 3 for respectively X, Y, Z) ; I: particle number.

◇  $SO(J, I)$  : initial spin coordinates.

◇ Constraints 7.1 - 7.9: X1 and X2, optional, define the longitudinal coordinate interval inside the optical element, over which it applies.

◇ Constraints 10.2 - 10.4: I is the number of the concerned 3-particle momentum group, among the IDMAX momentum groups defined by OBJET[KOBJ=2, IDMAX]; if only one group needed, OBJET[KOBJ=1, IMAX=3] may be used as well.

◇ Constraint 10.3: If NP=1 and  $pr_{NP}=1$ , then the absolute value of the X, Y or Z component is accounted for.

**Combining FIT and REBELOTE, Example 2:**

The goal in this problem is to produce the outer and inner coil current values of RHIC 9 o'clock snake, for a series of values of the snake axis orientation and spin rotation by the snake.

Down the optical sequence, the execution pointer meets *FIT2*; when the latter is completed, the execution pointer continues to *REBELOTE*. The latter will re-launch the problem NRBLT=5 times, but prior to re-launching, it will change the values of 3 of the constraints in *FIT*.

This results in the graph displayed on the right.

The files and scripts in this example, including the two snake OPERA field maps, can be found in Zgoubi repository:

[https://sourceforge.net/p/zgoubi/code/HEAD/tree/trunk/exemples/RHIC/spinTransparency/spinTransparencyExperiment/scanSnake\\_MuPhi/](https://sourceforge.net/p/zgoubi/code/HEAD/tree/trunk/exemples/RHIC/spinTransparency/spinTransparencyExperiment/scanSnake_MuPhi/)

```
Snake 1. 9 o'clock snake, in Yellow going clockwise.
'OBJET'
79.366778931425273 * 1d3
1
3 1 1 1 1 1 ! 3 times the same particle, to compute a spin matrix.
0.0 0.0 0.0 0.0 0.0 0.0000000E+00
0. 0. -1.3206397 0. 0. 1.
'PARTICUL'
PROTON

'SPNIRK'
4
1.000 0.000 0.000000000E+00 ! Respective spins of the 3 particles, in that order, for
0.000 1.000 0.000000000E+00 ! spin matrix computation by SPNPRT[MATRIX].
0.000 0.000 1.000000000E+00
'SCALING'
1 2
TOSCA snkLowB
-1
184.47838 ! low-field coils current (A)
1
TOSCA snkHighB
-1
198.70954 ! high-field coils current (A)
1

'TOSCA' snkLowB
0 0 ! .plt
-1. 1.000 1.000 1.000000000E+00
HEADER_8 RHIC_helix
361 81 81 15.1 0.748502994012e-4 ! = 1/1.3360e4
b_model13a2a-x-4_4_y-4_4_z-180_180-integral.table
0 0 0
2
.3
2 0.000 0.000 0.000000000E+00
'DRIFT' 2.162m_C1toC2
-98.800000
'TOSCA' snkHighB
0 0 ! .plt
+1. 1.000 1.000 1.000000000E+00
HEADER_8 RHIC_helix
361 81 81 15.1 0.246305418719e-4 ! = 1/4.060e4
b_model13a2a322a-x-4_4_y-4_4_z-180_180-integral.table
0 0 0
2
.3
2 0.000 0.000 0.000000000E+00
'DRIFT' YO9_B7.1 VMON ! center of 9 o'clock snake
-37.6
'DRIFT' 2.162m_C2toC3
-37.6

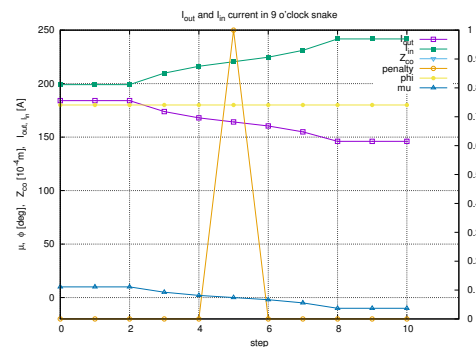
'TOSCA' snkHighB
0 0 ! .plt
-1. 1.000 1.000 1.000000000E+00
HEADER_8 RHIC_helix
361 81 81 15.1 0.246305418719e-4 ! = 1/4.060e4
b_model13a2a322a-x-4_4_y-4_4_z-180_180-integral.table
0 0 0
2
.3
2 0.000 0.000 0.000000000E+00
'DRIFT' 2.162m_C3toC4
-98.800000
'TOSCA' snkLowB
0 0 ! .plt
+1. 1.000 1.000 1.000000000E+00
HEADER_8 RHIC_helix
361 81 81 15.1 0.748502994012e-4 ! = 1/1.3360e4
b_model13a2a-x-4_4_y-4_4_z-180_180-integral.table
0 0 0
2
.3
2 0.000 0.000 0.000000000E+00
'DRIFT'
16.4

'FIT2'
3 save ! Save FIT outcomes (variable values, penalty, ...) to zgoubi.FITVALS.out, for plotting.
1 42 0 1.5 ! Z_0, vertical position of reference orbit at OBJET.
4 4 0 1.5 ! Low B coil current.
4 8 0 1.5 ! High B coil current.
5 1e-8
7.3 1 4 10 0. ! Vertically center traj #1 in 3rd helix;
10.2 1 0 #End 3.14159265359 1. 0 ! D-group 1. Spin rotation angle = 180;
10.3 1 1 #End 0.984807753012 1. 0 ! X component of rotation axis = cos(-10deg);
10.3 1 2 #End -0.173648177667 .01 0 ! Y component of rotation axis = sin(-10deg);
10 1 4 #End 1. .1 0 ! |S|=1.

'FAISCEAU'
'SPNPRT' MATRIX PRINT ! Cumulate spin matrix and data in zgoubi.SPNPRT.Out, at each pass.

'REBELOTE'
5 0.1 0 1
3
FIT2 64 3.0368728 2.9845130 2.9321531 2.8797932 2.8797932 ! mu (deg),
FIT2 74 0.9945218 0.9975640 0.9993908 1. 1. ! cos(phi),
FIT2 84 -0.1045284 -0.0697564 -0.0348994 0. 0. ! sin(phi).

'END'
```



This figure is obtained from `zgoubi.FITVALS.out`, using the `gnuplot` script below:

```
set title "I_(out) and I_(in) current in 9 o'clock snake"
set xtics; set ytics nomirror; set y2tics
set key maxrow 1; set grid
system "grep '4 2 4 ' zgoubi.FITVALS.out | cat > temp1"
system "grep '4 3 8 ' zgoubi.FITVALS.out | cat > temp2"
system "grep 'Fit reached penalty value' zgoubi.FITVALS.out | cat > temp3"
system "grep '10 1 0 19 ' zgoubi.FITVALS.out | cat > temp4"
system "grep '10 1 2 19 ' zgoubi.FITVALS.out | cat > temp5"

r2d = 180. / (4.*atan(1.))

plot \
"temp1" u 0:($6) w lp pt 4 tit "I_(out)" ,\
"temp2" u 0:($6) w lp pt 5 tit "I_(in)" ,\
"temp3" u 0:($5) axes xly2 w lp pt 6 tit "penalty" ,\
"temp4" u 0:($7 *r2d) w lp pt 7 tit "phi" ,\
"temp5" u 0:(-asin($7) *r2d * 10.) w lp pt 8 tit "-10* mu"
```

### • Example : FIT[2] options

In this example, *FIT[2]* options, “nofinal”, “save FileName”, “penalty” and “ITER”, are all specified explicitly. *FIT[2]* finds the reference closed orbit; the outcomes of that FIT are saved in *zgoubiFIT.result*; the final run normally following FIT (with variables updated to their matched values) is skipped; penalty and maximum number of calls to the function are explicit (default values would be, respectively, 1e-10 and 999). When FIT is done, the execution pointer moves on to *FAISCEAU*, etc.

```
An example with some explicit FIT options
'GETFITVAL'
zgoubi.FITVALS.In
'MCOBJET'
79.e3                                     ! reference rigidity.
3.1
1
2 2 2 2 1 1
0. 0. 0. 0. 0. 0. 4.61943880E-01
 2.703371 1.859656 1.48563786487e-9 9
-2.592712 2.421706 1.48563786487e-9 9
0. 1. 0. 9
12345 23456 34567
'SPNTRK'                                 ! It seems spins are being tracked!
4.1
0. 0. 1.
'PICKUPS'
2

'INCLUDE'                                 ! Include some periodic structure concerned.
1
AGS_Booster.inc[ring_S:ring_E]           ! AGS_Booster.inc file includes the optical sequence, it
                                         ! can be a standalone zgoubi input data file.

'FIT2'
4 save zgoubiFIT.result nofinal          ! Save variables in zgoubiFIT.result when FIT done. No final run.
2 40 0 [-1.e-2,1.e-2]                   ! The four variables are Yo, To, Zo, Po in MCOBJET.
2 41 0 [-1.e-2,1.e-2]
2 42 0 [-1.e-2,1.e-2]
2 43 0 [-1.e-2,1.e-2]
4 1e-15 90                               ! Penalty=1e-15 (default would be 1e-10). Maximum allowed calls to the function is 90.
3.1 1 2 7500 0. 1. 0                    ! The four constraints are : initial Y, T, Z, P of
3.1 1 3 7500 0. 1. 0                    ! particle #1 equal to its final Y, T, Z, P.
3.1 1 4 7500 0. 1. 0
3.1 1 5 7500 0. 1. 0

'FAISCEAU'                               ! This will log particle coordinates in zgoubi.res.
'SPNPRT'                                 ! This will log spin vector data - thus on the closed orbit - in zgoubi.res.
'END'
```



**FOCALE                      Particle coordinates and horizontal beam size at distance *XL***

*XL*                      Distance from the location of the keyword                      cm                      E

**FOCALEZ                      Particle coordinates and vertical beam size at distance *XL***

*XL*                      Distance from the location of the keyword                      cm                      E

**GASCAT****Gas scattering***KGA*

Off/On switch

0, 1

I

*AI, DEN*

Atomic number ; density

2\*E

**GETFITVAL**                    **Get values of variables as saved from former FIT[2] run**

*FNAME*                    Name of storage file. Zgoubi will proceed silently if A  
*FNAME*='none' or *FNAME*='NONE', or if the file is not found.

• **Example : GETFITVAL**

Expected content of file *FNAME* is the result of a prior *FIT[2]*, of the form:

```

STATUS OF VARIABLES (Iteration # 95)
LMNT  VAR  PARAM  MINIMUM  INITIAL  FINAL  MAXIMUM  STEP  NAME  LBL1  LBL2
145   1    4    -3.000E+03  762.    761.9484791 3.000E+03  1.254E-05 MULTIPOL  HKIC  DHCB02
182   2    4    -1.000E+03 -231.   -230.9846875 1.000E+03  4.182E-06 MULTIPOL  HKIC  DHCB08
146   3    4    -1.000E+03 -320.   -319.8554171 1.000E+03  4.182E-06 MULTIPOL  VKIC  DVCB02
183   4    4    -1.000E+03  528.    527.7249064 1.000E+03  4.182E-06 MULTIPOL  VKIC  DVCB08
615   5    4    -3.000E+03  308.    307.6860565 3.000E+03  1.254E-05 MULTIPOL  HKIC  DHCF02
651   6    4    -1.000E+03 -114.   -113.8490362 1.000E+03  4.182E-06 MULTIPOL  HKIC  DHCF08
616   7    4    -1.000E+03 -78.9   -78.88730937 1.000E+03  4.182E-06 MULTIPOL  VKIC  DVCF02
652   8    4    -1.000E+03  212.    211.8789183 1.000E+03  4.182E-06 MULTIPOL  VKIC  DVCF08
# STATUS OF CONSTRAINTS
# TYPE  I  J  LMNT#  DESIRED  WEIGHT  REACHED  KI2  *  Parameter(s)
# 3  1  2  127  0.0000000E+00  1.0000E+00  1.0068088E-08  6.0335E-01 * 0 :
# 3  1  3  127  0.0000000E+00  1.0000E+00  7.0101405E-09  2.9250E-01 * 0 :
# 3  1  4  127  0.0000000E+00  1.0000E+00  2.9184383E-10  5.0696E-04 * 0 :
# 3  1  5  127  0.0000000E+00  1.0000E+00  3.1142381E-10  5.7727E-04 * 0 :
# 3  1  2  436  0.0000000E+00  1.0000E+00  3.8438378E-09  8.7944E-02 * 0 :
# 3  1  3  436  0.0000000E+00  1.0000E+00  1.5773011E-09  1.4808E-02 * 0 :
# 3  1  4  436  0.0000000E+00  1.0000E+00  2.2081272E-10  2.9022E-04 * 0 :
# 3  1  5  436  0.0000000E+00  1.0000E+00  5.7930552E-11  1.9975E-05 * 0 :
# Function called 1859 times
# Xi2 = 1.68006E-16 Busy...

```

**HALBACH**

**Halbach style permanent magnet**

UNDER DEVELOPMENT

A

<b>HELIX</b>	<b>Helical dipole</b>	(page 126 in PART A)	
<i>IL</i>	<i>IL</i> = 1, 2[ $\times 10^n$ ], 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).		
<i>XL, PIT B<sub>0</sub>, <math>\phi</math></i>	Length; pitch; field; angle of $\vec{B}$ to vertical at entrance	cm, cm, kG, rad	4*E
<i>KIRD, resol</i>	Field and derivatives analytical: KIRD=0, resol =2 (default); Field derivatives from mesh: KIRD=3, mesh size is XPAS/Resol	-, -	2*I
<i>XPAS</i>	Integration step	cm	E
<i>KPOS, XCE, YCE, ALE</i>	<i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt.	1-2, 2*cm, rad	I, 3*E

<sup>1</sup>For the geek: details in subroutines `rhelix.f` (read input) and `helix.f` (initialize data, computes field)

<b>GOTO</b>	<b>Branching statement</b>		
	<b>Under development</b>		
OPTION	Can be 'PASS#', 'GOBACK', 'NEXT' (skips the next element), it can also be an existing <i>LABEL1</i> (will send execution pointer to closest following occurrence of <i>LABEL1</i> ).	String	A
<b>If OPTION = PASS#</b> <i>LABEL1, ..., LABELN</i>	One additional line needed : List of labels (expected is label #1 that follows a keyword) to go to, pass after pass (passes are incremented, from 1 to N, by <i>REBELOTE</i> for instance).	N strings	N*A20

- **Example : GOTO**

see page 298, case of an energy recovery, multiple-recirculation linac.

<b>HISTO</b>	<b>1-D histogram</b>	(200 in Part A)	
<i>J</i> , $X_{\min}$ , $X_{\max}$ , <i>NBK</i> , <i>NH</i> [, <i>PRINT</i> ]	<i>J</i> = coordinate to be histogrammed, as follows: <ul style="list-style-type: none"> <li>• current coordinates : 1(<i>D</i>), 2(<i>Y</i>), 3(<i>T</i>), 4(<i>Z</i>), 5(<i>P</i>), 6(<i>S</i>),</li> <li>• initial coordinates : 11(<i>D</i><sub>0</sub>), 12(<i>Y</i><sub>0</sub>), 13(<i>T</i><sub>0</sub>), 14(<i>Z</i><sub>0</sub>), 15(<i>P</i><sub>0</sub>), 16(<i>S</i><sub>0</sub>),</li> <li>• spin : 21(<i>S</i><sub>X</sub>), 22(<i>S</i><sub>Y</sub>), 23(<i>S</i><sub>Z</sub>), 24( <i>S</i> ) ;</li> </ul> $X_{\min}$ , $X_{\max}$ = limits of the histogram, in units of the coordinate of concern ; <i>NBK</i> = number of bins ; <i>NH</i> = number of the histogram (for independence of histograms of the same coordinate); <i>PRINT</i> , optional, will cause output to zgoubi.HISTO.out.	1-24, 2* current units, ≤ 120, 1-5 [,A]	I, 2*E, 2*I
<i>NBL</i> , <i>KAR</i> , <i>NORM</i> , <i>TYP</i>	Number of lines (= vertical amplitude) ; alphanumeric character ; normalization if <i>NORM</i> = 1, otherwise <i>NORM</i> = 0 ; <i>TYP</i> = 'P' : primary particles are histogrammed, or 'S' : secondary, or Q : all particles - for use with <i>MCDESINT</i>	normally 10-40, char., 1-2, P-S-Q	I, A1, I, A1

• **Example : HISTO**

The following will cause an histogram of the relative rigidity (D), case of the sole P-primary particles (case of in-flight decay). 'PRINT' causes output to zgoubi.HISTO.out (can be used for gnuplot-ing, for instance).

```
'HISTO'
1 0.3 2.1 100 2 PRINT ! D coordinate; in [0.3,2.1]; 100 bins; this histogram is #2; PRINT option.
100 B 0 P ! 100 lines; tag='B'; do not normalize; 'P' for primary particles only.
```

**IMAGE**                      **Localization and size of horizontal waist**

**IMAGES**                      **Localization and size of horizontal waists**

For each momentum group, as classified by means of *OBJET*, *KOBJ* = 1, 2 or 4

**IMAGESZ**                      **Localization and size of vertical waists**

For each momentum group, as classified by means of *OBJET*, *KOBJ* = 1, 2 or 4

**IMAGEZ**                      **Localization and size of vertical waist**



<b>INCLUDE</b>	<b>File include statement</b>	(page 172 in PART A)	
<i>NBF</i>	Number of files to be included (1 at the moment - under development).	1	I
<b>NBF following lines (one file name per line) :</b>			
<i>FNAME</i> [ <i>LBL_1A</i> ] [, [ <i>LBL_2A</i> ]]: [ <i>LBL_1B</i> ] [, [ <i>LBL_2B</i> ]] ]	Name of input zgoubi.dat-style file to be included. <i>LBL_1A</i> , <i>_2A</i> : entrance branching tags, <i>LBL_1B</i> , <i>_2B</i> : exit branching tags.  (An INCLUDED file must terminate with “ ’END’ “.)	A80	[,[A20], [A20]]

### • Example : INCLUDE

In this example of an energy-recovery electron recirculator [85], the linac, spreader and combiner data lists are given in separate files, declared as *INCLUDE*s. Using *INCLUDE* has a series of merits :

- it shortens the parent input data file, so making the optical structure more apparent,
- it allows picking optical spreader, combiner and ring sequences from files that may actually be set to - for instance - compute optical parameters ; in that case, *LBL\_A* and *LBL\_B* are used for this targetted picking : they define the ends of the sequence to be *INCLUDE*'d, ignoring the upper and lower parts in the *INCLUDE*'d file (which may be specific, for instance, to *MATRIX* computations).

Note the following : (i) juggling between linac, spreaders, combiners and recirculating ring, is handled using *GOTO* statement, (ii) the use of *REBELOTE* in this ERL context is discussed in an example page 327.

```

Cornell CBETA prototype (of an eRHIC) FFAG arc ERL
'OBJET'
19.9411300960454126 ! T.m E_tot=6MeV
5.01
.01 .01 .01 .01 .01 .01
0. 0. 0. 0. 0. 1.
-0.8475 1.5739 -0.8481 1.59 0. 1. 7.49E-03 4.12E-03 0. 0.
'PARTICUL'
0.51099892 1.60217653e-19 1.15965218076e-3 0.0 0.0

'ERRORS'
0 1 123466 dB(kG)
MULTIPOL{} 1 BP A U 0.d0 0.0 9999

'SCALING'
1 2
MULTIPOL
2
1. 1.
1 9999
MULTIPOL SA1.1 SB1.1
2
1. 1.
1 9999

'OPTIONS'
1 1
WRITE ON

!///// Racetrack sequence starts here, at linac entrance
! Linac
'INCLUDE'
1
./LA/LA_obj5.inc[LA.MAR.BEG\1:LA.MAR.END\1]

! Goto splitter line
'GOTO'
PASS#
SA1 SA2 SA3 SA4 SA3 SA2 SA1
'MARKER' RT_SA

! Goto FFAG return loop : arc+(DS+straight+DS)+arc
'GOTO'
PASS#
FA FA FA FA FA FA FA FA
'MARKER' RT_FA
'GOTO'
PASS#
X X X X X X X X
'MARKER' RT_X
'GOTO'
PASS#
FB FB FB FB FB FB FB FB
'MARKER' RT_FB

! Goto combiner line
'GOTO'
PASS#
SB1 SB2 SB3 SB4 SB3 SB2 SB1
'MARKER' RT_SB

'MATRIX'
1 0

'MARKER' rebelote
'REBELOTE' ! send zgoubi execution pointer back to
7 0.1 99 ! the top (and linac entrance), 7 times

'MARKER' stop
'STOP'
!!!!!!!!!!!!!!!!!!!!
!// job done, here //
!!!!!!!!!!!!!!!!!!!!

! Splitter lines
'MARKER' SA1
'INCLUDE'
1
./SA/SA1.inc[LA.MATCH1:S1.MERGE.MAR.END\1]
'GOTO'
GOBACK
'MARKER' SA2
'INCLUDE'
1
./SA/SA2.inc[LA.DEMER.MAR.BEG\2:S1.MERGE.MAR.END\2]
'GOTO'
GOBACK
'MARKER' SA3
'INCLUDE'
1
./SA/SA3.inc[SA3#S:SA3#E]
'GOTO'
GOBACK
'MARKER' SA4
'INCLUDE'
1
./SA/SA4.inc[SA4#S:SA4#E]
'GOTO'
GOBACK

'MARKER' FA
'INCLUDE'
1
./FA/FA.inc[FA#S:FA#E]
'GOTO'
GOBACK

'MARKER' X
'INCLUDE'
1
./X./X.inc[X#S:X#E]
'GOTO'
GOBACK

'MARKER' FB
'INCLUDE'
1
./FB/FB.inc[FB#S:FB#E]
'GOTO'
GOBACK

! Combiner lines
'MARKER' SB1
'INCLUDE'
1
./SB/SB1.inc[SB1#S:SB1#E]
'GOTO'
GOBACK
'MARKER' SB2
'INCLUDE'
1
./SB/SB2.inc[SB2#S:SB2#E]
'GOTO'
GOBACK
'MARKER' SB3
'INCLUDE'
1
./SB/SB3.inc[SB3#S:SB3#E]
'GOTO'
GOBACK
'MARKER' SB4
'INCLUDE'
1
./SB/SB4.inc[SB4#S:SB4#E]
'GOTO'
GOBACK

'MATRIX'
1 0
'END'

```

<b>MAP2D</b>		<b>2-D Cartesian uniform mesh field map - arbitrary magnetic field</b>	
<i>IC, IL</i>	<i>IC</i> = 1, 2 : print the field map <i>IL</i> = 1, 2[ $\times 10^n$ ], 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2; 0-2[ $\times 10^n$ ], 7	2*I
<i>BNORM, XN, YN</i>	Field and X-,Y-coordinate unit conversion coeffs. Convert values as read from map file, to kG and cm units.	3*UnitConv.	3*E
<i>TITL</i>	Title. Start with "FLIP" to get field map X-flipped.		A80
<i>IX, JY</i>	Number of longitudinal and horizontal-transverse nodes of the mesh (the Z elevation is arbitrary)	$\leq 400, \leq 200$	2*I
<i>FNAME</i> <sup>1</sup>	File name		A80
<i>ID, A, B, C</i> [, <i>A', B', C', B''</i> , etc., if <i>ID</i> $\geq 2$ ]	Integration boundary. Ineffective when <i>ID</i> = 0. <i>ID</i> = -1, 1 or $\geq 2$ : as for <i>CARTEMES</i>	$\geq -1, 2$ *no dim, cm [,2*no dim, cm, etc.]	I,3*E [,3*E,etc.]
<i>IODRE</i>	Degree of polynomial interpolation, 2nd or 4th order.	2, 4	I
<i>XPAS</i>	Integration step	cm	E
<i>KPOS, XCE, YCE, ALE</i>	<i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt.	1-2, 2*cm, rad	I, 3*E

<sup>1</sup> *FNAME* (e.g., magnet.map) contains the field map data.

These must be formatted according to the following *FORTRAN* read sequence (normally compatible with *TOSCA* code *OUTPUTS* - details and possible updates are to be found in the source file 'fmapw.f') :

```

OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD')
DO 1 J = 1, JY
  DO 1 I = 1, IX
    IF (BINARY) THEN
      READ(NL) Y(J), Z(1), X(I), BY(I,J), BZ(I,J), BX(I,J)
    ELSE
      READ(NL,100) Y(J), Z(1), X(I), BY(I,J), BZ(I,J), BX(I,J)
100   FORMAT (1X, 6E11.4)
    ENDIF
  1   CONTINUE

```

where  $X(I)$ ,  $Y(J)$  are the longitudinal, horizontal coordinates in the at nodes  $(I, J)$  of the mesh,  $Z(1)$  is the vertical elevation of the map, and  $BX$ ,  $BY$ ,  $BZ$  are the components of the field.

For binary files, *FNAME* must begin with 'B\_' or 'b\_'; a logical flag 'Binary' will then automatically be set to '.TRUE.'

<b>MAP2D-E</b>		<b>2-D Cartesian uniform mesh field map - arbitrary electric field</b>	
<i>IC, IL</i>	<i>IC</i> = 1, 2 : print the field map <i>IL</i> = 1, 2[ $\times 10^n$ ], 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2; 0-2[ $\times 10^n$ ], 7	2*I
<i>ENORM, X-, Y-NORM</i>	Field and X-,Y-coordinate unit conversion coeffs. Convert values as read from map file, to MV/cm and cm units.	3*UnitConv.	3*E
<i>TITL</i>	Title. Start with "FLIP" to get field map X-flipped.		A80
<i>IX, JY</i>	Number of longitudinal and horizontal-transverse nodes of the mesh (the Z elevation is arbitrary)	$\leq 400, \leq 200$	2*I
<i>FNAME</i> <sup>1</sup>	File name		A80
<i>ID, A, B, C</i> [, <i>A', B', C'</i> , <i>B''</i> , etc., if <i>ID</i> $\geq 2$ ]	Integration boundary. Ineffective when <i>ID</i> = 0. <i>ID</i> = -1, 1 or $\geq 2$ : as for <i>CARTEMES</i>	$\geq -1, 2^*$ no dim, cm [,2*no dim, cm, etc.]	I,3*E [,3*E,etc.]
<i>IODRE</i>	Degree of polynomial interpolation, 2nd or 4th order.	2, 4	I
<i>XPAS</i>	Integration step	cm	E
<i>KPOS, XCE, YCE, ALE</i>	<i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt.	1-2, 2*cm, rad	I, 3*E

<sup>1</sup> *FNAME* (e.g., "mirror.map") contains the field map data.

These must be formatted according to the following *FORTRAN* read sequence - details and possible updates are to be found in the source file 'fmapw.f' :

```

OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD')
DO 1 J = 1, JY
  DO 1 I = 1, IX
    IF (BINARY) THEN
      READ(NL) Y(J), Z(1), X(I), EY(I,J), EZ(I,J), EX(I,J)
    ELSE
      READ(NL,100) Y(J), Z(1), X(I), EY(I,J), EZ(I,J), EX(I,J)
100   FORMAT (1X, 6E11.4)
    ENDIF
  1   CONTINUE

```

where  $X(I)$ ,  $Y(J)$  are the longitudinal, horizontal coordinates in the at nodes  $(I, J)$  of the mesh,  $Z(1)$  is the vertical elevation of the map, and  $EX$ ,  $EY$ ,  $EZ$  are the components of the field.

For binary files, *FNAME* must begin with 'B\_' or 'b\_'; a logical flag 'Binary' will then automatically be set to '.TRUE.'

**MARKER**

**Marker**

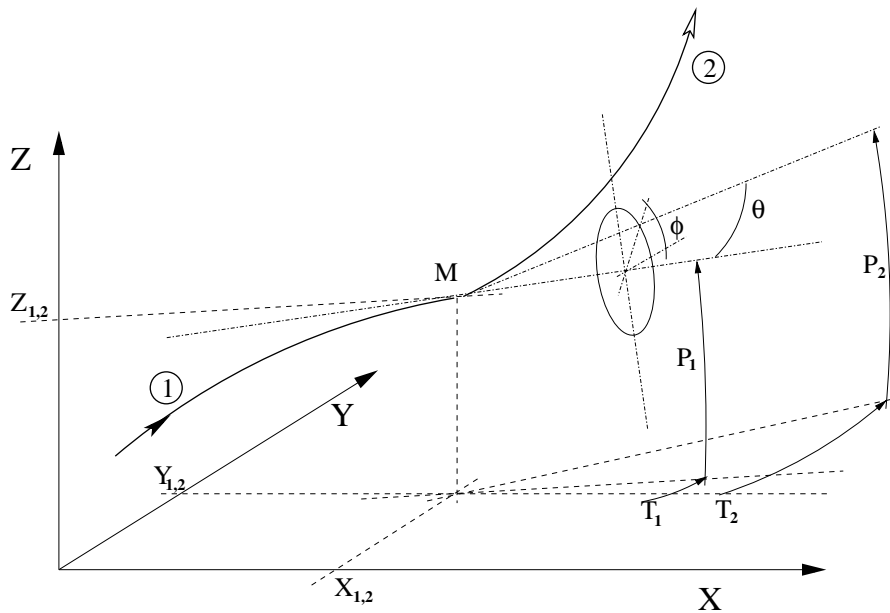
Just a marker. No data

'*.plt*' as a second *LABEL* will cause storage of current coordinates into *zgoubi.plt*

<b>MATRIX</b>	<b>Calculation of transfer coefficients, periodic parameters</b>	(page 202 in PART A)
<i>IOR</i> , <i>IFOC</i> [, <i>PRINT</i> ] [, <i>coupled</i> ]	<p>Options :</p> <p><i>IOR</i> = 0 : Same effect as <i>FAISCEAU</i></p> <p><i>IOR</i> = 1, using <i>OBJET</i>, <i>KOBJ</i> = 5[.N] : 1st order transport parameters,  - if <i>KOBJ</i> = 5.1 : beam matrix, phase advance, otherwise  - <math>N \geq 2</math>: 1st order data (<i>cf.</i> <i>IFOC</i> value) for N different cases of  - optical reference.</p> <p><i>IOR</i> = 2, using <i>OBJET</i>, <i>KOBJ</i> = 6 : 1st order transport matrix [<i>R<sub>ij</sub></i>],  2nd order array [<i>T<sub>ijk</sub></i>] and a few higher order transport coefficients ;  some periodic parameters in addition if <i>IFOC</i> &gt; 10 ;</p> <p><i>IOR</i> = 3, using <i>OBJET</i>, <i>KOBJ</i> = 6.1 : 1st order transport matrix [<i>R<sub>ij</sub></i>],  2nd and 3rd order arrays [<i>T<sub>ijk</sub></i>], [<i>T<sub>ijkl</sub></i>].</p> <p><i>IFOC</i> = 0 : matrix at actual location<sup>1</sup> ;  <i>IFOC</i> = 1 : matrix at the closest first order horizontal focus<sup>1</sup> ;  <i>IFOC</i> = 10 + <i>NPER</i> : same as <i>IFOC</i> = 0, and also calculates 1 or N sets  (case respectively of <i>KOBJ</i> = 5 or <i>KOBJ</i> = 5.N, <math>2 \leq N \leq 99</math>) of Twiss  parameters, tune numbers, etc., assuming <i>zgoubi.dat</i> file describes  one period of a <i>NPER</i>-period structure<sup>1</sup>.</p>	0-2, 0-1 or > 10    2*I [,A]
	Occurrence of “ <i>PRINT</i> ” will cause printout to <i>zgoubi.MATRIX.out</i> file.	
	Occurrence of “ <i>coupled</i> ”, in the periodic case ( <i>IFOC</i> = 10 + <i>NPER</i> , above), will cause use of coupled formalism.	

<sup>1</sup> The reference in computing transport coefficients and other periodic parameters is particle # 1 of the ray-traced set. If N 13-particle sets are ray-traced in order to get (in a single *zgoubi* run) transport coefficients for N different cases of reference orbit (using *OBJET*, *KOBJ* = 5.N,  $2 \leq N \leq 99$ ), the reference particle for each 13-particle set is respectively particle #1, 12, 23, ...  $1+(N-1) \times 11$ .

<b>MCDESINT</b> <sup>1</sup>	<b>Monte-Carlo simulation of in-flight decay</b> <b>M1 → M2 + M3</b>	(page 173 in PART A)
<i>M2, M3, τ2</i> <sup>2</sup> [ <i>, back</i> ] [ <i>, INFO</i> ] <sup>3</sup>	Mass of the two decay products (M1 is defined using <i>PARTICUL</i> ) ; COM lifetime of M2 [ <i>; allow backward: 1/0 default is 0</i> ] [ <i>; Print out info</i> ]	2*MeV/c <sup>2</sup> , s [ <i>,0-1</i> ] 3*E [ <i>, E</i> ] [ <i>, INFO</i> ] [ <i>, A4</i> ]
<i>I1, I2, I3</i>	Seeds for random number generators	3*≈ 10 <sup>6</sup> 3*I



Particle 1 decays into 2 and 3 ; **zgoubi** then calculates trajectory of 2, while 3 is discarded.  $\theta$  and  $\phi$  are the scattering angles of particle 2 relative to the direction of the incoming particle 1. They transform to  $T_2$  and  $P_2$  in Zgoubi frame.

<sup>1</sup> *MCDESINT* must be preceded by *PARTICUL*, for the definition of the mass and lifetime of the incoming particle M1.  
<sup>2</sup>  $\tau_2$  can be left blank, in which case the lifetime of particle 2 is set to zero (it decays immediately, which from a practical point of view means that it is not tracked).  
<sup>3</sup> Presence of 'INFO' will cause more info on decay kinematics parameters to be printed zgoubi.res at each decay.

<b>MCOBJET</b>	<b>Monte-Carlo generation of a 6-D object</b>	(page 60 in PART A)	
<i>BORO</i> [* <i>fac</i> ]	Reference rigidity = <i>BORO</i> [* <i>fac</i> ], with <i>fac</i> an arbitrary numerical value.	kG.cm [-,]	E [E]
<i>KOBJ</i>	Type of support of the random distribution <i>KOBJ</i> = 1 : window <i>KOBJ</i> = 2 : grid <i>KOBJ</i> = 3 : phase-space ellipses	1-3	I
<i>IMAX</i>	Number of particles to be generated	$\leq 10^4$	
<i>KY, KT, KZ, KP, KX, KD</i> <sup>1</sup>	Type of probability density	6*(1-3)	6*I
<i>Y<sub>0</sub>, T<sub>0</sub>, Z<sub>0</sub>, P<sub>0</sub>, X<sub>0</sub>, D<sub>0</sub></i>	Mean value of coordinates ( $D_0 = B\rho/BORO$ )	m, rad, m, rad, m, no dim	6*E
<b>If <i>KOBJ</i> = 1</b>	<b>In a window</b>		
$\delta Y, \delta T, \delta Z, \delta P, \delta X, \delta D$	Distribution widths, depending on <i>KY, KT</i> etc. <sup>1</sup>	m, rad, m, rad, m, no dim	6*E
$N_{\delta Y}, N_{\delta T}, N_{\delta Z}, N_{\delta P}, N_{\delta X}, N_{\delta D}$	Sorting cut-offs (used only for Gaussian density)	units of $\sigma_Y, \sigma_T,$ etc.	6*E
$N_0, C_0, C_1, C_2, C_3$	Parameters involved in calculation of P(D)	no dim.	5*E
<i>IR1, IR2, IR3</i>	Random sequence seeds	$3 \approx 10^6$	3*I
<b>If <i>KOBJ</i> = 2</b>	<b>On a grid</b>		
<i>IY, IT, IZ, IP, IX, ID</i>	Number of bars of the grid		6*I
<i>PY, PT, PZ, PP, PX, PD</i>	Distances between bars	m, rad, m rad, m, no dim	6*E
$\delta Y, \delta T, \delta Z, \delta P, \delta X, \delta D$	Width of the bars ( $\pm$ ) if uniform, Sigma value if Gaussian distribution	<i>ibidem</i>	6*E
$N_{\delta Y}, N_{\delta T}, N_{\delta Z}, N_{\delta P}, N_{\delta X}, N_{\delta D}$	Sorting cut-offs (used only for Gaussian density)	units of $\sigma_Y, \sigma_T,$ etc.	6*E
$N_0, C_0, C_1, C_2, C_3$	Parameters involved in calculation of $P(D)$	no dim.	5*E
<i>IR1, IR2, IR3</i>	Random sequence seeds	$3 \approx 10^6$	3*I

<sup>1</sup> Let  $x = Y, T, Z, P$  or  $X$ . *KY, KT, KZ, KP* and *KX* can take the values

1 : uniform,  $p(x) = 1/2\delta x$  if  $-\delta x \leq x \leq \delta x$

2 : Gaussian,  $p(x) = \exp(-x^2/2\delta x^2)/\delta x\sqrt{2\pi}$

3 : parabolic,  $p(x) = 3(1 - x^2/\delta x^2)/4\delta x$  if  $-\delta x \leq x \leq \delta x$

*KD* can take the values

1 : uniform,  $p(D) = 1/2\delta D$  if  $-\delta D \leq x \leq \delta D$

2 : exponential,  $p(D) = \text{No} \exp(C_0 + C_1 l + C_2 l^2 + C_3 l^3)$  if  $-\delta D \leq x \leq \delta D$

3 : kinematic,  $D = \delta D * T$

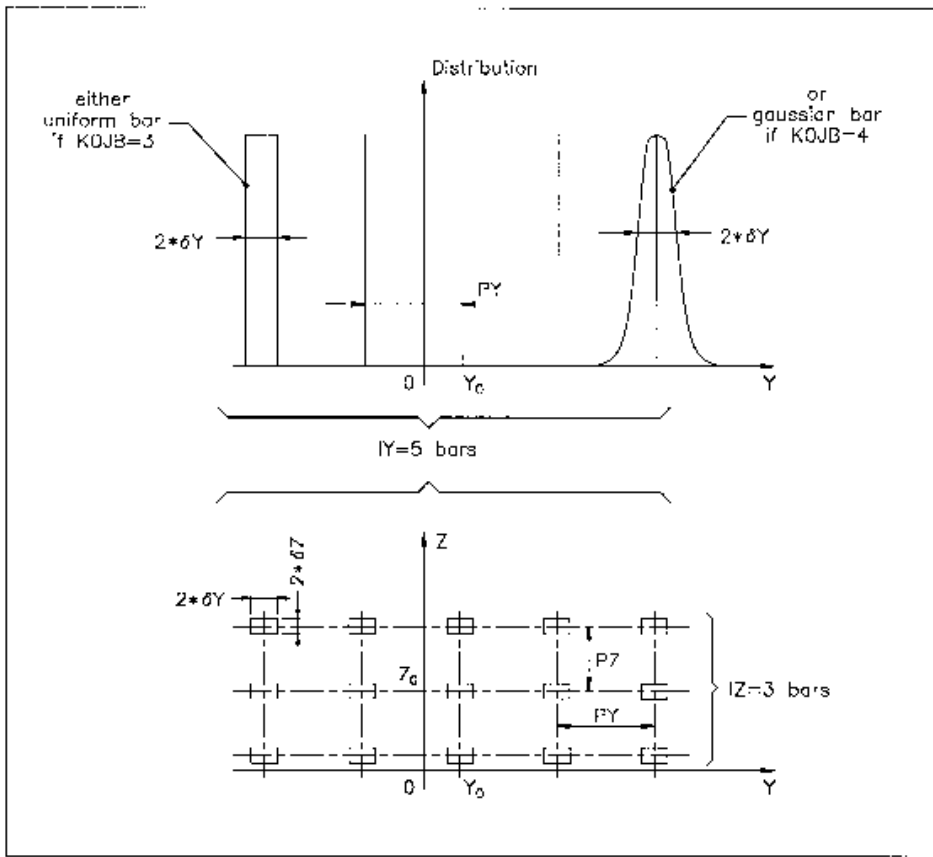


**If KOBJ = 3****Over 2-D phase-space ellipses<sup>1</sup>. Four lines follow :**

$\alpha_Y, \beta_Y, \varepsilon_Y/\pi, N_{\sigma_{\varepsilon_Y}}$ [, $N'_{\sigma_{\varepsilon_Y}}$ if $N_{\sigma_{\varepsilon_Y}} < 0$ ] <sup>2</sup> [, $D_Y, D_T$ ]	Ellipse parameters and emittance, Y-T phase-space ; cut-off ; dispersion and derivative	no dim, m/rad, m, units of $\sigma(\varepsilon_Y)$ ; [, m, rad]	4*E [,E] [,2*E]
$\alpha_Z, \beta_Z, \varepsilon_Z/\pi, N_{\sigma_{\varepsilon_Z}}$ [, $N'_{\sigma_{\varepsilon_Z}}$ if $N_{\sigma_{\varepsilon_Z}} < 0$ ] <sup>2</sup> [, $D_Z, D_P$ ]	Ellipse parameters and emittance, Z-P phase-space ; cut-off ; dispersion and derivative	no dim, m/rad, m, units of $\sigma(\varepsilon_Z)$ [, m, rad]	4*E [,E] [,2*E]
$\alpha_X, \beta_X, \varepsilon_X/\pi, N_{\sigma_{\varepsilon_X}}$ [, $N'_{\sigma_{\varepsilon_X}}$ if $N_{\sigma_{\varepsilon_X}} < 0$ ] <sup>2</sup>	Ellipse parameters and emittance, X-D phase-space ; cut-off	no dim, m/rad, m, units of $\sigma(\varepsilon_X)$	4*E [,E]
<i>IR1, IR2, IR3</i>	Random sequence seeds. In order for different runs to produce different series (meaning, overriding these seeds) launch zgoubi using 'irank' argument: [pathTo]/zgoubi -irank someNonZeroInteger -in mySequence.dat, or [pathTo]/zgoubi -irank someNonZeroInteger (input file is zgoubi.dat) ... Note: <i>SRLOSS</i> will also differ under the effect of irank.	$3*\simeq 10^6$	3*I

<sup>1</sup> Similar possibilities, non-random, are offered with *OBJET*, KOBJ=8 (p. 309)

<sup>2</sup> Works with Gaussian density type only : sorting is within the ellipse frontier ( $\frac{1 + \sigma_Y^2}{\beta_Y^2} Y^2 + 2\alpha_Y Y T + \beta_Y T^2 = \frac{\varepsilon_Y}{\pi}$ ) if  $N_{\sigma_{\varepsilon_Y}} > 0$ , or, if  $N_{\sigma_{\varepsilon_Y}} < 0$  sorting is within the annular region [ $|N_{\sigma_{\varepsilon_Y}}|, N'_{\sigma_{\varepsilon_Y}}$ ] of that ellipse.



A scheme of input parameters to *MCOBJET* when *KOBJ= 2*.

Top : Possible distributions of the *Y* coordinate

Bottom : A 2-D grid in (*Y*, *Z*) space.

MULTIPOL	Magnetic Multipole	dim.	data type	FIT num.
$IL$	$IL = 1, 2[\times 10^n]$ , 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2 $[\times 10^n]$ , 7	I	1
$XL, R_0, B_1, B_2, \dots, B_{10}$ ,	Length of element ; radius at pole tip ; field at pole tip for dipole, quadrupole, ..., dodecapole components	2*cm, 10*kG	12*E	2-13
$X_E, \lambda_E, E_2, \dots, E_{10}$	<b>Entrance face</b> Integration zone ; fringe field extent : dipole fringe field extent = $\lambda_E$ ; quadrupole fringe field extent = $\lambda_E * E_2$ ; ... 20-pole fringe field extent = $\lambda_E * E_{10}$ (sharp edge if field extent is zero)	2*cm, 9*no dim	11*E	14-24
$NCE, C_0 - C_5$	same as <i>QUADRUPO</i>	0-6, 6*no dim	I, 6*E	25-31
$X_S, \lambda_S, S_2, \dots, S_{10}$	<b>Exit face</b> Integration zone ; as for entrance	2*cm, 9*no dim	11*E	32-42
$NCS, C_0 - C_5$		0-6, 6*no dim	I, 6*E	43-49
$R_1, R_2, R_3, \dots, R_{10}$	Skew angles of field components	10*rad	10*E	50-59
$XPAS$	Integration step	cm	E	60-62
<b>If KPOS = 1-3</b> $KPOS, XCE, YCE, ALE$	Positioning as follows : $KPOS=1$ : element aligned ; $KPOS=2$ : misaligned ; shifts, tilt ; $KPOS=3$ : effective only if $B_1 \neq 0$ : entrance and exit frames are tilted wrt. the magnet by an angle of • either $ALE$ if $ALE \neq 0$ • or $2 \text{Arcsin}(B_1 XL / 2BORO)$ if $ALE=0$	1-3, 2*cm, rad	I, 3*E	63-66
<b>If KPOS = 4</b> $KPOS, XS, YS, ZR, ZS, YR$	UNDER DEVELOPMENT X-, Y-shift, Z-rotation, Z-shift, Y-rotation Automatic ZR-positioning (as with $ALE$ in case $KPOS = 3$ ).	4, cm, 2*(cm, rad)	I, 5*E	63-68
<b>If KPOS = 5</b> $KPOS, XS, YS, ZS, XR, YR, ZR$	UNDER DEVELOPMENT X-, Y-, Z-shift, X-, Y-, Z-rotation: Automatic ZR-positioning (as with $ALE$ in case $KPOS = 3$ ).	5, 3*cm, 3*rad	I, 5*E	

<b>OBJET</b>	<b>Generation of an object</b>	(page 63 in PART A)	
<i>BORO</i> [* <i>fac</i> ]	Reference rigidity = <i>BORO</i> [* <i>fac</i> ], with <i>fac</i> an arbitrary numerical value.	kG.cm [-]	E [E]
<i>KOBJ</i> [.K2]	Option index [.More options]	1-6	I
<b>If KOBJ = 1</b> [.N]	By default, tracking assumes mid-plane symmetry; N=1: generate particles with $Z_0 > 0$ only; N=2: same as default, but all particles tracked, no mid-plane symmetry assumed.		
<i>IY, IT, IZ, IP, IS, ID</i>	Ray-Tracing assumes mid-plane symmetry. Generated points : $YR \pm IY * dY, TR \pm IT * dT, ZR \pm IZ * dZ, PR \pm IP * dP$ [ $ZR + IZ * dZ, PR + IP * dP$ if <i>KOBJ</i> = 1.1], $SR \pm IS * dS$ and $DR \pm ID * dD$ coordinates ( $IY \leq 20, \dots, ID \leq 20$ )	$IY*IT*IZ*IP*IS*ID \leq 10^4$	6*I
<i>dY, dT, dZ, dP, dS, dD</i>	Sampling size in <i>Y, T, Z, P, S</i> and in relative momentum ( $dD = \delta B\rho/BORO$ )	2(cm,mrad), cm, no dim	6*E
<i>YR, TR, ZR, PR, SR, DR</i>	Reference trajectory ( $DR = B\rho/BORO$ )	2(cm,mrad), cm, no dim	6*E
<b>If KOBJ = 2</b> [.1]	All the initial coordinates must be entered explicitly [2.1: units are different, see below]		
<i>IMAX, IDMAX</i>	total number of particles ; number of distinct momenta (if <i>IDMAX</i> > 1, group particles of same momentum)	$IMAX \leq 10^4$	2*I
<b>For I = 1, IMAX</b>	Repeat <i>IMAX</i> times the following line		
<i>Y, T, Z, P, S, D, LET</i>	Coordinates and tagging of the <i>IMAX</i> particles ; If <i>KOBJ</i> = 2.1 input units are different : 2(m,rad), m, no dim	2(cm,mrad), cm, no dim	6*E, A1
<i>IEX</i> ( <i>I</i> = 1, <i>IMAX</i> )	<i>IMAX</i> times 1 or -9. If <i>IEX</i> ( <i>I</i> ) = 1 trajectory <i>I</i> is ray-traced, it is not if <i>IEX</i> ( <i>I</i> ) = -9.	1 or -9	<i>IMAX</i> I
<b>If KOBJ=3</b> [.N, N=0 - 3] [, 'HEADER_NH']	Read coordinates from storage file. Optional 'HEADER_NH' tells the number of header lines <i>NH</i> (case $N \neq 0$ : default $NH=0$ ; case $N=0$ : $NH=NHDFL$ hardcoded in <code>robjet.f</code> ). N=0 (default) : [b_]zgoubi.fai style data file FORMAT N=1 : read FORMAT is ``READ (NL, *) Y, T, Z, P, S, DP`` N=2 : read FORMAT is ``READ (NL, *) X, Y, Z, PX, PY, PZ`` N=3 : read FORMAT is ``READ (NL, *) DP, Y, T, Z, P, S, TIME, MASS, CHARGE`` N=4 : read FORMAT is ``READ (NL, *) Y, T, Z, P, S, DP, LET, IEN, IT`` N=5 : read FORMAT is ``READ (NL, *, ERR=97, END=95) Y, T, Z, P, TIME, GAMMA`` N=6 : read FORMAT is ``READ (NL, *) Y, Z, X, PY, PZ, PX``		
<i>IT1, IT2, ITStep</i>	Read particles numbered <i>IT1</i> to <i>IT2</i> , step <i>ITStep</i> (For more than $10^4$ particles stored in <i>FNAME</i> , use 'REBELOTE')	$\geq 1, \geq IT1, \geq 1$	3*I
<i>IP1, IP2, IPStep</i>	Read particles that belong in pass numbered <i>IP1</i> to <i>IP2</i> , step <i>IPStep</i>	$\geq 1, \geq IP1, \geq 1$	3*I
<i>YF, TF, ZF, PF, SF, DF, TF, TAG</i>	Scaling factor. TAG-ing letter : no effect if $TAG='*$ , otherwise only particles with $TAG \equiv LET$ are retained.	7*no.dim, char.	7*E, A1
<i>YR, TR, ZR, PR, SR, DR, TR</i>	Reference. Given the previous line of data, all coordinate <i>C</i> (=Y, T...) is transformed to $C*CF+CR$	2(cm, mrad), cm, no dim, $\mu s$	7*E
<i>InitC</i>	0 : set new $\vec{R}_0 = \vec{R}_0$ as read, new $\vec{R} = \vec{R}$ as read ; 1 : set new $\vec{R}_0 = \vec{R}$ as read, new $\vec{R} = \vec{R}$ as read ; 2 : save $\vec{R}$ as read in new $\vec{R}_0$ , set new $\vec{R} = \vec{R}_0$ as read.	0-1	I
<i>FNAME</i>	File name (e.g., [b_]zgoubi.fai) ( <i>N</i> in <i>KOBJ</i> =3.N determines formatting)		

<b>If KOBJ = 5</b> [ $N, N \geq 1$ ]	<b>Generation of 13 particles, or 13*N if <math>N \geq 2</math></b> (for use with <i>MATRIX</i> [ <i>IORD</i> =1], <i>TWISS</i> )		
$dY, dT, dZ, dP, dS, dD$	Sampling size in $Y, T, Z, P, S$ (unused) and $D$	2(cm,mrad), cm, no dim	6*E
$YR, TR, ZR, PR, SR, DR$	Reference trajectory ( $DR = B\rho/BORO$ ; $SR$ is not used)	2(cm,mrad), cm, no dim	6*E
◇ If $N=1$ ( $KOBJ = 5.1$ )	One additional data line :		
$\alpha_Y, \beta_Y, \alpha_Z, \beta_Z, \alpha_S, \beta_S,$ $D_Y, D'_Y, D_Z, D'_Z$ (NOT the periodic $\eta, \eta'$ )	Initial beam ellipse parameters <sup>1</sup>	2(no dim,m), ?, ?, 2(m,rad)	6*E, 4*E
◇ If $N \geq 2$ ( $KOBJ = 5.N$ )	An additional $N-1$ data lines <sup>2</sup>		
$YR, TR, ZR, PR, SR, DR$	Reference trajectory for each of the $N-1$ additional 13-particle sets # $i=2,N$ ; ( $DR = B\rho/BORO$ ; $SR$ is not used).	2(cm,mrad), cm, no dim	6*E
<b>If KOBJ =</b> $\begin{bmatrix} 6 \\ 6.1 \end{bmatrix}$	<b>Generation of</b> $\begin{bmatrix} 61 \\ 102 \end{bmatrix}$ <b>particles</b> (for use with <i>MATRIX</i> , <i>IORD</i> = $\begin{bmatrix} 2 \\ 3 \end{bmatrix}$ )		
$dY, dT, dZ, dP, dS, dD$	Sampling size in $Y, T, Z, P, S$ (unused) and $D$	2(cm,mrad), cm, no dim	6*E
$YR, TR, ZR, PR, SR, DR$	Reference trajectory ; $DR = B\rho/BORO$	2(cm,mrad), cm, no dim	6*E
<b>If KOBJ = 7</b>	<b>Object with kinematics</b>		
$IY, IT, IZ, IP, IS, ID$	Number of points in $\pm Y, \pm T, \pm Z, \pm P, \pm S$ ; $ID$ is not used	$IY*IT*IZ*IP*IS \leq 10^4$	6*I
$dY, dT, dZ, dP, dS, dD$	Sampling size in $Y, T, Z, P$ and $S$ ; $dD$ = kinematic coefficient, such that $D(T) = DR + dD * T$	2(cm,mrad), cm, mrad <sup>-1</sup>	6*E
$YR, TR, ZR, PR, SR, DR$	Reference ( $DR = B\rho/BORO$ )	2(cm,mrad), cm, no dim	6*E
<b>If KOBJ = 8</b>	<b>Generation of phase-space coordinates on ellipses</b> <sup>3</sup>		
$IY, IZ, IS$	Number of samples in each 2-D phase-space ; if zero the central value (below) is assigned	$0 \leq IY, IZ, IS \leq IMAX,$ $1 \leq IY * IZ * IS \leq IMAX$	3*I
$Y_0, T_0, Z_0, P_0, S_0, D_0$	Central values ( $D_0 = B\rho/BORO$ )	2(m, rad), m, no dim	6*E
$\alpha_Y, \beta_Y, \varepsilon_Y/\pi$	ellipse parameters and emittances	no dim, m, m	3*E
$\alpha_Z, \beta_Z, \varepsilon_Z/\pi$		no dim, m, m	3*E
$\alpha_S, \beta_S, \varepsilon_S/\pi$		no dim, m, m	3*E

<sup>1</sup> They can be transported by using *MATRIX*.

<sup>2</sup> The maximum value for  $N$ , normally *MXREF*=999, is hard-coded in *obj5.f*.

<sup>3</sup> Similar possibilities, random, are offered with *MCOBJET*, *KOBJ*=3 (p. 305).

• **Example : OBJET, KOBJ=3**, recovering from a crash

The job below simulates spin tracking in SATURNE 3 GeV proton synchrotron. Turn-by-turn coordinates at the beginning of the ring sequence are stored in zgoubi.fai during the tracking. *CAVITE/Option=2* is used.

The left column shows the input data file for initial job, turn3000.dat. Note: the 'INCLUDE' command uploads SATURNE.ring sequence, from the very file used in the example in Sec. 5 (page 374, PART B of the guide) - it only loads that part of the file which is comprised between the markers labelled "#StartRing" and "#EndRing").

The right column, job resume.dat, shows how *OBJET, KOBJ=3* is used to allow resuming tracking at start of turn #1000, possibly following a crash during that turn. *OBJET, KOBJ=3* requests reading initial coordinates from turn3000.fai, at turn 1000.

A possible check: the 2,000 tracking by resume.dat (right column below) should yield all three phase spaces, namely, (Y, T), (Z, P), (D, time) superimposing with turns 1,000-3,000 of the original turn3000.dat job (left column). If this does not happen: a quick check is by comparison of turn 1000 in turn3000.fai and the initial value of the coordinates in resume.fai, they should be *identical*.

This 3,000-turn job crashed at turn number 1,000.  
Particle coordinates were stored turn-by-turn at the beginning of the optical sequence, in turn3000.fai.

The input data below recover from the crash, starting with coordinates as read at turn 1,000 in turn3000.fai.

```
SATURNE. CROSSING GammaG=7-NUz, NUz=3.60877(perturbed)
'OBJET'
5018.67
2
1 1
0. 0. .458 0. 0. 1.00 'o'
1

'PARTICUL'
PROTON
'SPNTRK'
3
'FAISCNL'
turn3000.fai

'SCALING'
1 3 PRINT
QUADRUPO
-1
5018.67E-3
1
MULTIPOL
-1
5018.67E-3
1
BEND
-1
5018.67E-3
1

'INCLUDE'
1
SATURNE.ring[#StartRing:#EndRing]

'CAVITE'
2 PRINT
105.5556848673 3.
6000. 0.205303158892

'REBELOTE'
2999 0.2 99
'FAISCEAU'
'SPNPRT'
'END'
```

```
SATURNE. CROSSING GammaG=7-NUz, NUz=3.60877(perturbed)
'OBJET'
5018.67
3
1 1 1
1000 1000 1
1. 1. 1. 1. 1. 1. 1. *
0. 0. 0. 0. 0. 0. 0.
0
turn3000.fai

'PARTICUL'
PROTON
'SPNTRK'
3
'FAISTORE'
resume.fai
1

'SCALING'
1 3 PRINT
QUADRUPO
-1
5018.67E-3
1
MULTIPOL
-1
5018.67E-3
1
BEND
-1
5018.67E-3
1

'INCLUDE'
1
SATURNE.ring[#StartRing:#EndRing]

'CAVITE'
2 PRINT
105.5556848673 3.
6000. 0.205303158892

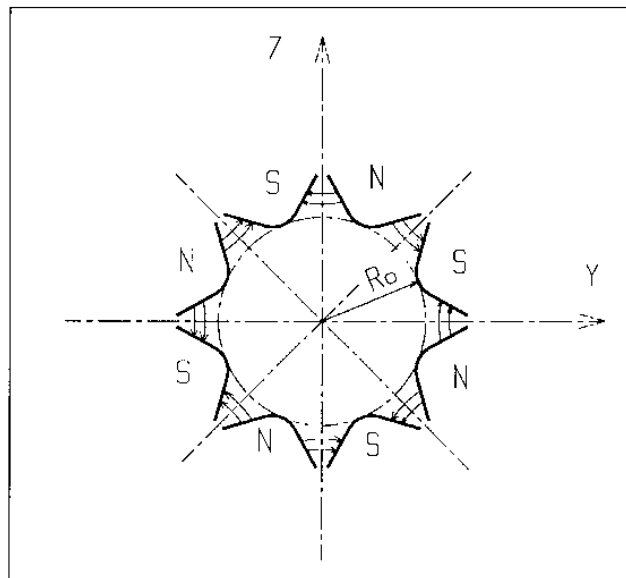
'REBELOTE'
2000 0.2 99
'FAISCEAU'
'SPNPRT'
'END'
```

Particle and spin coordinates at the end of the complete 3,000 turn run and at the end of the 1,000 to 3,000 turn recovery, as from the two files above, are *identical*, they are displayed below.

TRACE DU FAISCEAU (follows element # 90) 1 TRAJECTOIRES													
OBJET							FAISCEAU						
	D	Y(cm)	T(mr)	Z(cm)	P(mr)	S(cm)	D-1	Y(cm)	T(mr)	Z(cm)	P(mr)	S(cm)	
o 1	1.0000	0.000	0.000	0.458	0.000	0.0000	0.0029	0.006	0.007	0.498	-1.263	0.000000E+00	1
---				Time of flight (mus) :				mass (MeV/c2) :					
				1244.3530				938.272					
Spin components of each of the 1 particles, and rotation angle :													
INITIAL							FINAL						
	SX	SY	SZ	S	SX	SY	SZ	S	GAMMA	(Si,Sf) (deg.)	(Z,Sf_X) (deg.)	(Z,Sf) (deg.)	
													(Sf_X : projection of Sf on YZ plane)
o 1	0.000000	0.000000	1.000000	1.000000	-0.926463	0.185319	0.327602	1.000000	1.8937	70.877	29.496	70.877	1

<b>OBJETA</b>	<b>Object from Monte-Carlo simulation of decay reaction</b>		
	$M1 + M2 \longrightarrow M3 + M4$ and $M4 \longrightarrow M5 + M6$		
<i>BORO</i>	Reference rigidity	kG.cm	E
<i>IBODY, KOBJ</i>	Body to be tracked : $M3$ ( <i>IBODY</i> =1), $M5$ ( <i>IBODY</i> =2) $M6$ ( <i>IBODY</i> =3) ; type of distribution for $Y_0$ and $Z_0$ : uniform ( <i>KOBJ</i> = 1) or Gaussian ( <i>KOBJ</i> = 2)	1-3,1-2	2*I
<i>IMAX</i>	Number of particles to be generated (use ' <i>REBELOTE</i> ' for more)	$\leq 10^4$	I
$M_1 - M_6$	Rest masses of the bodies	6*GeV/c <sup>2</sup>	6*E
$T_1$	Kinetic energy of incident body	GeV	E
$Y_0, T_0, Z_0, P_0, D_0$	Only those particles in the range $Y_0 - \delta Y \leq Y \leq Y_0 + \delta Y$ ..... $D_0 - \delta D \leq D \leq D_0 + \delta D$ will be retained	2(cm,mrad), no dim	5*E
$\delta Y, \delta T, \delta Z, \delta P, \delta D$		2(cm,mrad), no dim	5*E
<i>XL</i>	Half length of object : $-XL \leq X_0 \leq XL$ (uniform random distribution)	cm	E
<i>IR1, IR2</i>	Random sequence seeds	$2^* \simeq 0^6$	2*I

OCTUPOLE	Octupole magnet	(page 131 in PART A)
$IL$	$IL = 1, 2[\times 10^n], 7$ : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2 $[\times 10^n]$ , 7 I
$XL, R_0, B_0$	Length ; radius and field at pole tip of the element	2*cm, kG 3*E
$X_E, \lambda_E$	<b>Entrance face :</b> Integration zone ; Fringe field extent ( $\lambda_E = 0$ for sharp edge)	2*cm 2*E
$NCE, C_0 - C_5$	$NCE = \text{unused}$ $C_0 - C_5 = \text{fringe field coefficients}$ such that : $G(s) = G_0/(1 + \exp P(s))$ , with $G_0 = B_0/R_0^3$ and $P(s) = \sum_{i=0}^5 C_i (s/\lambda)^i$	0-6, 6*no dim I, 6*E
$X_S, \lambda_S$	<b>Exit face :</b> Parameters for the exit fringe field ; see entrance	2*cm 2*E
$NCS, C_0 - C_5$		0-6, 6*no dim I, 6*E
$XPAS$	Integration step	cm E
$KPOS, XCE, YCE, ALE$	$KPOS=1$ : element aligned, $2$ : misaligned ; shifts, tilt.	1-2, 2*cm, rad I, 3*E



Octupole magnet



**OPTICS****Write out optical functions. Log to zgoubi.OPTICS.out**

*IOPT*[*NLBL*], label(s)      *IOPT* = 0/1 : Off/On ; *NLBL* = number of labels (default is 1) ;      0-1[.1-10], - ,      I[.I],  
 [,*PRINT*]                      'label' : can be 'all', 'ALL', or a list of *NLBL* first label(s) as      - , -                      1-10\*A20,  
 [, *coupled*]                      appearing at one or more elements in zgoubi.dat sequence ;                      I1 or A5, A7  
     wild card accepted, in the form '\*label' or 'label\*' or '\*label\*'.  
     Presence of 'PRINT' will cause print out of optical  
     functions in zgoubi.OPTICS.out ; optical function computation is  
     in coupled hypothesis if specified, uncoupled otherwise.

**• Example : OPTICS**

The following will cause a print of transported optical functions in zgoubi.res, at all optical elements of the zgoubi.dat sequence, as well as their print out in zgoubi.OPTICS.out :

```
'OPTICS'
1 all PRINT
```

<b>OPTIONS</b>	<b>Global or special options</b>	(page 176 in PART A)	
<i>IOPT, NBOP</i>	<i>IOPT</i> = 0/1 : Off/On ; 0 (off) will inhibit all <i>NBOP</i> subsequent requests. <i>NBOP</i> : total number of options.	0-1, $\geq 0$	2*I
<b>NBOP lines follow. Possible choices :</b>			
<i>'plt', IL</i>	IL=-1, 0-2, 7: option inactive (IL=-1); all .plt inhibited (IL=0); log stepwise particle data to zgoubi.res (IL=1), or to zgoubi.plt (IL=2), or to zgoubi.impdev.out (IL=7).	<i>'plt', -1,0-2,7</i>	A, I
<i>'AGSQUADS', OPT</i>	OPT = ON : modify AGS quads winding currents <sup>21</sup> 1.	<i>'AGSQUADS', 'ON'</i>	2*A
<i>'CONSTY', OPT</i>	OPT = ON : forces constant <i>Y</i> and <i>Z</i> across optical elements.	<i>'CONSTY', 'ON' or 'OFF'</i>	2*A
<i>'MONITOR', OPT</i>	OPT = IDLUNI : monitor IDLUNI counter.	<i>'AGSQUADS', 'IDLUNI'...</i>	2*A
<i>'RAND_SEED', OPT</i>	OPT = MCOBJ3 or SRLOSS. Will use date-and-time, and/or will freeze initialization of the random series.	<i>'MCOBJET3', 'SRLOSS'</i>	2*A
<i>'WRITE', OPT</i>	OPT = ON/OFF : allows/inhibites (most) write statements to zgoubi.res.	<i>'WRITE', 'ON' or 'OFF'</i>	2*A

<sup>1</sup>For the geek: this is managed in agsqf.f.

• **Example : OPTIONS, CONSTY**

The example below yields (data saved to zgoubi.plt due to  $IL = 2$ ) 3-D field components (trivially uniform in that particular case,  $B_R = B_\theta = 0$ ,  $B_Z = B_0$  at all  $(R, \theta, Z)$ ) on a  $n_R \times n_\theta \times n_Z = (41-1)/2 \times (R*AT/XPAS) \times (11-1)/2$  node, uniform 3-D mesh, with  $\Delta R = \delta Y$  in *OBJET* = 1 cm,  $\Delta\theta \equiv XPAS/RM = 10.471975512$  mrd,  $\Delta Z = \delta Z$  in *OBJET* = 0.2 cm, across a 60 degree hard-edged sector dipole (6 of these make up a uniform field cyclotron).

```
Cyclotron, classical.
'OBJET'
1000.e0
1
41 1 11 1 1 1
1. 0. 0.2 0. 0. 1.
0. 0. 0. 0. 0. 1.

'OPTIONS'
1 1
CONSTY ON

'DIPOLES'
2
1 60. 50.
30. 0. 5. 1 0.
0. 0.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.      ! EFB 1  hard-edge
30. 0. 1.E6 -1.E6 1.E6 1.E6
0. 0.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.      ! EFB 2
-30. 0. 1.E6 -1.E6 1.E6 1.E6
0. 0.
0 0. 0. 0. 0. 0. 0. 0. 0.      ! EFB 3
0. 0. 0. 0. 0. 0. 0. 0. 0.
0 2 KIRD, Resol=IDB
0.523598775598
2 00. 0. 00. 0.
'FAISCEAU'

'END'
```

**ORDRE****Taylor expansions order***IO*Taylor expansions of  $\vec{R}$  and  $\vec{u}$  (Eq. 1.2.4) up to  $\vec{u}^{(IO)}$   
(default is  $IO = 4$ )

2-5

I

**PARTICUL**      **Particle characteristics**      (page 179 in PART A)

**Either :**

$M, Q, G, \tau, X$       Mass (or coded, see Note below) ; charge ; gyromagnetic factor ;  
COM life-time ; unused      MeV/c<sup>2</sup>, C, no dim, s, NA      5\*E

**or :**

Name      ELECTRON, POSITRON, MUON+, MUON-, PION+, PION-,  
PROTON, DEUTERON, HELION      A

Note : If  $M$  is of the form {M1 M2}, then when masses are assigned to particles from a previously defined object (by [MC]OBJET), then the first half of the particles are given the mass M1, and the second half are given the mass M2.

If  $Q$  is zero, the reference charge is left unchanged.

Only the parameters to be used need their value be specified (for instance  $M, Q$  only when electric lenses are used,  $M, Q, G$  only when tracking spin, etc.) ; unused parameters can be set to zero.

**PICKUPS****Bunch centroid path; orbit**

Particle coordinates at pickups may be printed out in `zgoubi.PICKUP.out`;  
 they are anyway listed next to the sequence '`END`' keyword down `zgoubi.res`.  
 In multi-turn mode, pickup contents are zeroed at start of each pass.

`N` [`,PRINT`]                    0 : inactive  
                                    $\geq 1$  : number of *LABELs* at which beam centroid is computed                     $\geq 0$  [, string]                    I [,A]  
`[,PRINT]` is optional: will cause print of pickup contents to `zgoubi.PICKUP.out`.

**A list of N keywords' labels follows**

`LABEL1` [`,LABEL2`, [...]]    The N label(s) at which beam data are to be computed/recorded.                    N string(s)                    N\*A20  
 If some "`LABELi`" in this list actually does not appear  
 in `zgoubi.dat` optical sequence, then it is peacefully ignored ;  
 wild card accepted, in the form '`*LABEL`' or '`LABEL*`' or '`*LABEL*`'.

**• Example : PICKUPS**

A trick :

```
'PICKUPS'
1
none labelA labelB ...
```

This is a possible way to inhibit an earlier use of *PICKUPS* with "labelA, labelB, ..." keyword list. It is sufficient (and necessary) for that, that no keyword in `zgoubi.dat` data list have "none" as its first *LABEL*.

**PLOTDATA**

**Intermediate output for the PLOTDATA graphic software**

*To be documented.*

<b>POISSON</b>		<b>Read magnetic field data from <i>POISSON</i> output</b>	
<i>IC, IL</i>	<i>IC</i> = 1, 2 : print the field map <i>IL</i> = 1, 2[ $\times 10^n$ ], 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2; 0-2[ $\times 10^n$ ], 7	2*I
<i>BNORM, XN, YN</i>	Field and X-,Y-coordinate unit conversion coeffs.	3*no dim	3*E
<i>TITL</i>	Title. Start with "FLIP" to get field map X-flipped		A80
<i>IX, IY</i>	Number of longitudinal and transverse nodes of the uniform mesh	$\leq 400, \leq 200$	2*I
<i>FNAME</i> <sup>1</sup>	File name		A80
<i>ID, A, B, C</i> [, <i>A', B', C', B''</i> , etc., if <i>ID</i> $\geq 2$ ]	Integration boundary. Ineffective when <i>ID</i> = 0. <i>ID</i> = -1, 1 or $\geq 2$ : as for <i>CARTEMES</i>	$\geq -1, 2$ *no dim, cm [,2*no dim, cm, etc.]	I,3*E [,3*E,etc.]
<i>IORDRE</i>	Degree of interpolation polynomial as for <i>DIPOLE-M</i>	2, 25 or 4	I
<i>XPAS</i>	Integration step	cm	E
<i>KPOS, XCE, YCE, ALE</i> <i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt.		1-2, 2*cm, rad	I, 3*E

<sup>1</sup> *FNAME* (e.g., "outpoi.lis") contains the field map data.

These must be formatted according to the following *FORTRAN* read sequence - details and possible updates are to be found in the source file 'fmapw.f' :

```

I = 0
11 CONTINUE
I = I+1
READ(LUN,101,ERR=99,END=10) K, K, K, R, X(I), R, R, B(I)
101 FORMAT(I1, I3, I4, E15.6, 2F11.5, 2F12.3)
GOTO II
10 CONTINUE

```

where *X(I)* is the longitudinal coordinate, and *B(I)* is the *Z* component of the field at a node (*I*) of the mesh. *K*'s and *R*'s are variables appearing in the *POISSON* output file outpoi.lis, not used here.



<b>POLARMES</b>		<b>2-D polar mesh magnetic field map</b>	
	mid-plane symmetry is assumed		
<i>IC, IL</i>	<i>IC</i> = 1, 2 : print the map <i>IL</i> = 1, 2[ $\times 10^n$ ], 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2; 0-2[ $\times 10^n$ ], 7	2*I
<i>BNORM, AN, RN</i>	Field and A-,R-coordinate unit conversion coeffs.	3*no dim	3*E
<i>TITL</i>	Title. Start with "FLIP" to get field map X-flipped		A80
<i>IA, JR</i>	Number of angular and radial nodes of the mesh	$\leq 400, \leq 200$	2*I
<i>FNAME</i> <sup>1</sup>	File name		A80
<i>ID, A, B, C</i> [, <i>A', B', C', B''</i> , etc., if <i>ID</i> $\geq 2$ ]	Integration boundary. Ineffective when <i>ID</i> = 0. <i>ID</i> = -1, 1 or $\geq 2$ : as for <i>CARTEMES</i>	$\geq -1, 2$ *no dim, cm [,2*no dim, cm, etc.]	I,3*E [,3*E,etc.]
<i>IODRE</i>	Degree of interpolation polynomial (see <i>DIPOLE-M</i> )	2, 25 or 4	I
<i>XPAS</i>	Integration step	cm	E
<i>KPOS</i> <b>If <i>KPOS</i> = 2</b>	as for <i>DIPOLE-M</i> . Normally 2.	1-2	I
<i>RE, TE, RS, TS</i> <b>If <i>KPOS</i> = 1</b>		cm, rad, cm, rad	4*E
<i>DP</i>		no dim	E

<sup>1</sup> *FNAME* (e.g., spes2.map) contains the field data.

These must be formatted according to the following *FORTRAN* read sequence - details and possible updates are to be found in the source file 'fmapw.f' :

```

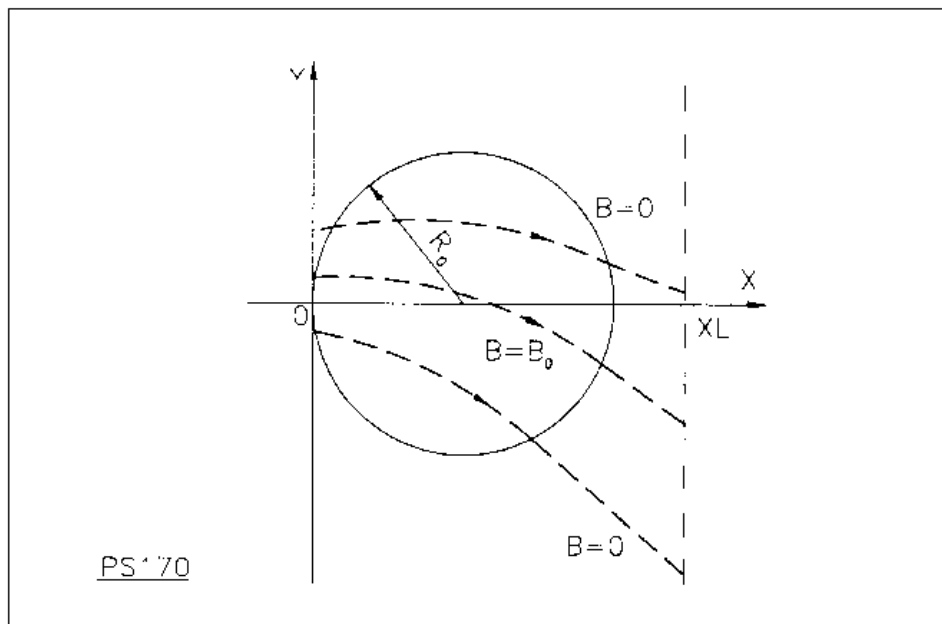
OPEN (UNIT = NL, FILE = FNAME, STATUS = 'OLD' [,FORM='UNFORMATTED'])
IF (BINARY) THEN
  READ(NL) (Y(J), J=1, JY)
ELSE
  READ(NL,100) (Y(J), J=1, JY)
ENDIF
100  FORMAT(10 F8.2)
  DO 1 I = 1, IX
  IF (BINARY) THEN
    READ (NL) X(I), (BMES(I,J), J=1, JY)
  ELSE
    READ(NL,101) X(I), (BMES(I,J), J=1, JY)
  101  FORMAT(10 F8.1)
  ENDIF
  1  CONTINUE

```

where *X(I)* and *Y(J)* are the longitudinal and transverse coordinates and *BMES* is the *Z* field component at a node (*I, J*) of the mesh. For binary files, *FNAME* must begin with 'B\_' or 'b\_'. 'Binary' will then automatically be set to 'TRUE.'

**PS170****Simulation of a round shape dipole magnet**

<i>IL</i>	$IL = 1, 2[\times 10^n], 7$ : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	$0-2[\times 10^n], 7$	I
<i>XL, R<sub>0</sub>, B<sub>0</sub></i>	Length of the element, radius of the circular dipole, field	$2^*cm, kG$	$3^*E$
<i>XPAS</i>	Integration step	cm	E
<i>KPOS, XCE, YCE, ALE</i>	<i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt.	1-2, $2^*cm, rad$	I, $3^*E$



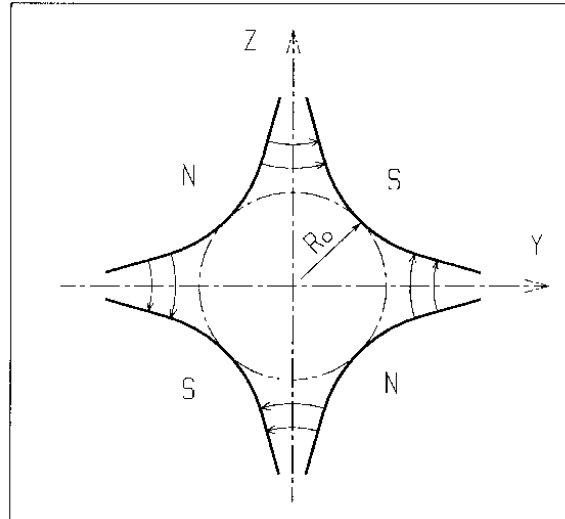
Scheme of the PS170 magnet simulation.

**QUADISEX****Sharp edge magnetic multipoles**

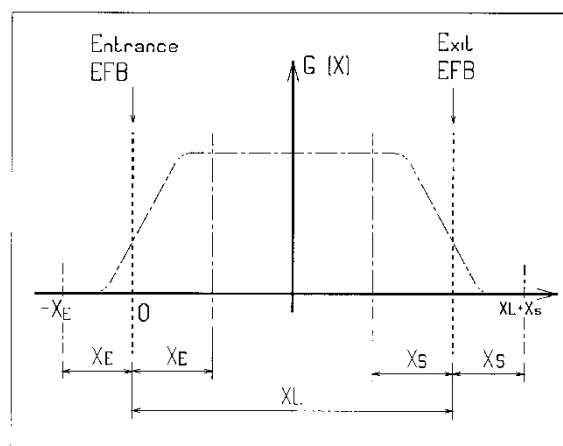
$$B_Z |_{Z=0} = B_0 \left( 1 + \frac{N}{R_0} Y + \frac{B}{R_0^2} Y^2 + \frac{G}{R_0^3} Y^3 \right)$$

<i>IL</i>	<i>IL</i> = 1, 2[×10 <sup>n</sup> ], 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2[×10 <sup>n</sup> ], 7	I
<i>XL, R<sub>0</sub>, B<sub>0</sub></i>	Length of the element ; normalization distance ; field	2*cm, kG	3*E
<i>N, EB1, EB2, EG1, EG2</i>	Coefficients for the calculation of B. if <i>Y</i> > 0 : <i>B</i> = <i>EB1</i> and <i>G</i> = <i>EG1</i> ; if <i>Y</i> < 0 : <i>B</i> = <i>EB2</i> and <i>G</i> = <i>EG2</i> .	5*no dim	5*E
<i>XPAS</i>	Integration step	cm	E
<i>KPOS, XCE, YCE, ALE</i>	<i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt.	1-2, 2*cm, rad	I, 3*E

<b>QUADRUPO</b>	<b>Quadrupole magnet</b>	(page 136 in PART A)	
<i>IL</i>	<i>IL</i> = 1, 2[ $\times 10^n$ ], 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2[ $\times 10^n$ ], 7	I
<i>XL, R<sub>0</sub>, B<sub>0</sub></i>	Length ; radius and field at pole tip	2*cm, kG	3*E
<i>X<sub>E</sub>, <math>\lambda_E</math></i>	<b>Entrance face :</b> Integration zone extent ; fringe field extent ( $\simeq 2R_0$ , $\lambda_E = 0$ for sharp edge)	2*cm	2*E
<i>NCE, C<sub>0</sub> - C<sub>5</sub></i>	<i>NCE</i> = unused <i>C<sub>0</sub> - C<sub>5</sub></i> = Fringe field coefficients such that $G(s) = G_0/(1 + \exp P(s))$ , with $G_0 = B_0/R_0$ and $P(s) = \sum_{i=0}^5 C_i (s/\lambda)^i$	0-6, 6*no dim	I, 6*E
<i>X<sub>S</sub>, <math>\lambda_S</math></i>	<b>Exit face</b> See entrance face	2*cm	2*E
<i>NCS, C<sub>0</sub> - C<sub>5</sub></i>		0-6, 6*no dim	I, 6*E
<i>XPAS</i>	Integration step	cm	E
<i>KPOS, XCE, YCE, ALE</i>	<i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt.	1-2, 2*cm, rad	I, 3*E



Quadrupole magnet



Scheme of the elements *QUADRUPO*, *SEXTUPOL*, *OCTUPOLE*, *DECAPOLE*, *DODECAPO* and *MULTIPOL*

( $OX$ ) is the longitudinal axis of the reference frame  $(0, X, Y, Z)$  of **zgoubi**.

The length of the element is  $X_L$ , but trajectories are calculated from  $-X_E$  to  $X_L + X_S$ , by means of automatic prior and further  $X_E$  and  $X_S$  translations.

**REBELOTE****'Do it again'**

*NPASS*, *KWRIT*, *K*[.n],  
 [*IOPT*],  
 [, *LABEL1* [, *LABEL2*']]

*NPASS* : Number of runs ; *KWRIT*= *i.j*: *i*=1/0 to switch arbitrary ; 0-1 ; 3\*I  
 on/off FORTRAN WRITES to .res, *j*=integer ≥ 1 for logging 0, 99 ; [1[.any]] 2A10  
 turn number to screen every  $10^{j-1}$  turns; 0-1 ; 0, 22, 99 2A10

*K* option :

*K* = 0 : initial conditions (coordinates and spins) are generated following the regular functioning of [*MC*]*OBJET*. If random generators are used (e.g. in *MCOBJET*) their seeds will not be reset.

*K* = 99 or 98 : coordinates at end of previous pass are used as initial coordinates for the next pass ; *idem* for spin components. *K*=99 causes a reset of particle path-length in *CAVITE*<sup>1</sup>, *K*=98 does not.

*K*=98 IS UNDER DEVELOPMENT, IN *cavite.f*, IN THIS VERSION OF THE CODE.

*K* = 99.1 : a "LABEL1" is expected in the optical sequence, subsequent passes will start from that element with *LABEL1*, down to *REBELOTE* and so forth ;

*K* = 99.2 : "LABEL1" and "LABEL2" are expected ; the last pass (# *NPASS*+1) will terminate at element with *LABEL2* whereupon *zgoubi* execution will jump to the keyword next to *REBELOTE* and will be carried out down to 'END'.

*IOPT* is optional :

*IOPT* = 1 : will change the value of *NPRM* different parameters in one or more of optical elements, using the data list in the next line.

**If IOPT = 1<sup>2</sup>**

*NPRM* Number of parameters to be changed at each of the *NPASS* passes. I

***NPRM* lines with the following sequence (tells parameters concerned, and for each, its successive values) :**

*LMNT*, *KPRM*, values *LMNT* : two possible forms : (i) *LMNT*=element number in *zgoubi.dat* sequence ; (ii) *LMNT*=KEYWORD{*LABEL1* [, *LABEL2*]}, i.e., keyword concerned followed optionnally by *LABEL1* (and possibly *LABEL2*) concerned. -, -, NV\*dim<sup>3</sup> I or A, I, NV\*E

*KPRM* : parameter number under that keyword (similar nomenclature to *FIT*[2], see page 284) ; 'values' defines the new parameter values. 'values' can take two forms :

- either an explicit list of *NPASS* numerical values
- or, first\_value:last\_value - the interval between consecutive values will be (last\_value - first\_value)/(npass-1).

<sup>1</sup> Usefull if the tracking distance is orders of magnitudes greater than particle distances within a bunch: the computation of the synchronous phase of a particle (from its total path length and its distance to the virtual synchronous particle) may be jeopardized if the difference is within computer accuracy.

<sup>2</sup> *IOPT*=1 is compatible with use of the *FIT*[2] procedure : e.g., allows successive *FIT*s in a single **zgoubi** run, with successive sets of optical parameter values.

<sup>3</sup> *V* is in current **zgoubi** units in the case of particle coordinates (cm, mrad). It is in MKSA units (m, rad) in the case of matrix coefficients.

• **Example : REBELOTE[IOPT=1]**

In this example of an energy-recovery electron recirculator based on FFAG arcs, the arguments in the keyword *AUTOREF* and *CAVITE* are changed 20 times by *REBELOTE*, over the 21-pass recirculation process [86]. The role of *AUTOREF* is to mimic a spreader-recombiner, *i.e.*, re-centering the beam on the design FFAG orbit at the start of each one of the 11 accelerated and 10 decelerated return loop passes. The RF voltage in *CAVITE* is set positive (accelerating, from 7.944 GeV to 21.16 GeV) during the first 11 passes through the optical structure, and negative (decelerating, from 21.16 to 7.944 GeV) over the remaining 10 passes. The voltage value at each pass is adjusted (for an energy of 1.322 GeV per turn on average) so to compensate the energy lost by synchrotron radiation in the arcs.

```

eRHIC ENERGY RECOVERY LINAC RECIRCULATOR WITH FFAG ARCS.
'MCOBJET'                                     1
57.36635309d3      reference rigidity (kG.cm)
3
2000
2 2 2 2 1 1
-5.360667E-03      5.059706E-3      0. 0. 0.      4.619439E-01 'o'
0. 1 0. 3
0. 1 0. 3
0. 1. 0. 3
123456 234567 345678
'PARTICUL'                                       2
0.51099892 1.60217653e-19 1.15965218076e-3 0.0 0.0
'SPNTRK'                                       3
'FAISCEAU'                                       4
'SRLOSS'                                        5
1 srLoss
MULTIPOL
1 123456
'SCALING'                                       6
1 1
MULTIPOL
-1
57.36635309 57.36635309
1 11
'MARKER' ARC#S_1                                7
'OPTIONS'                                       8
1 1
WRITE OFF
'MARKER' MARK CELLSTART                        9
'DRIFT' DRIF HD                               10
14.547181
'MULTIPOL' RBEN BD2                           11
0 .Dip
90. 10. 0. -0.87159105 0. 0. 0. 0. 0. 0. 0.
0. 0. 10.00 4.0 0.800 0.00 0.00 0.00 0.00 0. 0. 0. 0.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.
0. 0. 10.00 4.0 0.800 0.00 0.00 0.00 0.00 0. 0. 0. 0.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
#30|90|30 Dip BD2
3 0.0E+00 4.0704703E-01 -1.5071892929E-03
'DRIFT' DRIF D                                 12

'MULTIPOL' RBEN QF2                            13
0 .Dip
110. 10. 0. 0.86286642 0. 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0. 0. 10.00 4.0 0.800 0.00 0.00 0.00 0.00 0. 0. 0. 0.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.
0. 0. 10.00 4.0 0.800 0.00 0.00 0.00 0.00 0. 0. 0. 0.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
#30|110|30 Dip QF2
3 0. -3.6008008661E-01 -1.8710983347E-03
'DRIFT' DRIF HD                               14
14.547181
'MARKER' MARK CELLEND                          15
-----
6*138-1 additional such FD cells, simulating a 6 arc
energy recovery ring, 138 cells per arc.
-----
'MARKER' ARC#E_6                               5820
'OPTIONS'                                       5821
1 1
WRITE ON
'FAISTORE'                                       5822
zgoubi.fai
1
'FAISCEAU'                                       5823
'CAVITE'                                        5824
3
0 0
1.322e9 1.57079632679
'AUTOREF'                                       5825
4.1
0. -5.168354E-01 4.169759E+00 5.38813280E-01
'FAISCEAU'                                       5826
'REBELOTE'                                       5827
20 0.1 99 1
4
AUTOREF 11 -4.7493E-01 -4.1290E-01 -3.3292E-01 -2.3684E-01
-1.26268E-01 -2.59705E-03 ... (a list of 20 values)
AUTOREF 12 3.34619E+00 2.58215E+00 1.87157E+00 1.20916
5.90197E-01 1.05897E-02 ... (a list of 20 values)
AUTOREF 13 .615682680 .692552080 .769421480 .84629088
.923160280 1.00002970 ... (a list of 20 values)
CAVITE 20 1.337399e9 1.33934e9 1.34012e9 1.34018e9
... (15 more voltage values) ... -1.30660e9 -1.309577e9
'SRPRNT'                                       5828
'END'                                           5829

```

• **Example : Combining REBELOTE and FIT**

See page 285.

**RESET****Reset counters and flags**

Resets counters involved in *CHAMBR*, *COLLIMA*, *HISTO* and *INTEG* procedures.

Switches off *CHAMBR*, *MCDESINT*, *SCALING* and *SPNTRK* options.



<b>SCALING</b>	<b>Power supplies and R.F. function generator</b>	(page 183 in PART A)	
<i>IOPT, NFAM</i> [, <i>PRINT</i> ]	<i>IOPT</i> = 0 (inactive) or 1 (active) ; <i>NFAM</i> = number of families to be scaled. Occurrence of <i>PRINT</i> will cause printing of scaling infos in zgoubi.SCALING.out.	0-1, 1-45 [-]	I1, I2 [,A]
<b>For NF=1, NFAM :</b> repeat <i>NFAM</i> times the following sequence :			
<i>NAMEF</i> [, <i>Lbl</i> ...]	Name of family ( <i>i.e.</i> , keyword of concern) ; up to 9 labels, wild card accepted, in the form '*myLABEL' or 'myLABEL*'.	A10 [,1-9*A20]	
<i>NT</i> [ <i>Int</i> ]	<i>NT</i> ≥ 1 : normal usage, number of timings (see <i>NT</i> = -1 for a convenient, easy short-cut) ; 1. <i>Int</i> : Int=1 or 10-13, optional. For special scaling laws and/or reading from external files ; <i>NT</i> = -1 : field scaling is updated to follow $B\rho$ change by <i>CAVITE</i> ; <i>NT</i> = -2 : to read RF law for <i>CAVITE</i> from an external file (default name zgoubi.freqLaw.In) <sup>1</sup> ; <i>NT</i> = -60 : field scaling follows $B\rho$ change by <i>CAVITE</i> - to be documented <sup>2</sup> ; <i>NT</i> = -77 : field scaling law for FNAL proton driver - to be documented <sup>2</sup> ; <i>NT</i> = -87 : AGS tune-jump quadrupoles - to be documented <sup>2</sup> ; <i>NT</i> = -88 : AC dipole function generator <sup>3</sup> .	-88[.1, .10-.13], ..., 1[.1, .10-.13]	I[,I]
<i>SCL(I)</i> , <i>I</i> = 1, <i>NT</i>	Scaling values. if <i>NT</i> = 1 : <i>SCL(I)</i> = <i>SCL(1)</i> whatever the pass number. Special cases : if <i>NT</i> = -1 : a single value, <i>SCL</i> [* <i>fac</i> ] (scaling value is <i>SCL</i> [* <i>fac</i> ], with <i>fac</i> an arbitrary numerical value) ; if <i>NT</i> = -2 : unused ; if <i>NT</i> = -88, four vlues expected : phase of the ac dipole; Q1, tune at start of the sweep; Q2, tune at end of the sweep; scale factor.	- - - [, -] - rad, -, -, -	NT*E E E [, E] E 4*E
<i>TIM(I)</i> , <i>I</i> = 1, <i>NT</i>	Corresponding timings, in units of turns Special cases : if <i>NT</i> = -1 : <i>TIM(I≡1)</i> =1 ; if <i>NT</i> = -2 : unused ; if <i>NT</i> = -88, four turn numbers expected : hold duration (at zero); ramp up duration; sweep duration; ramp down duration.	NT turn#  1 - 4 turn#	NT*I  I I 4*E

1

The formatting of these ascii data file is given page~185.

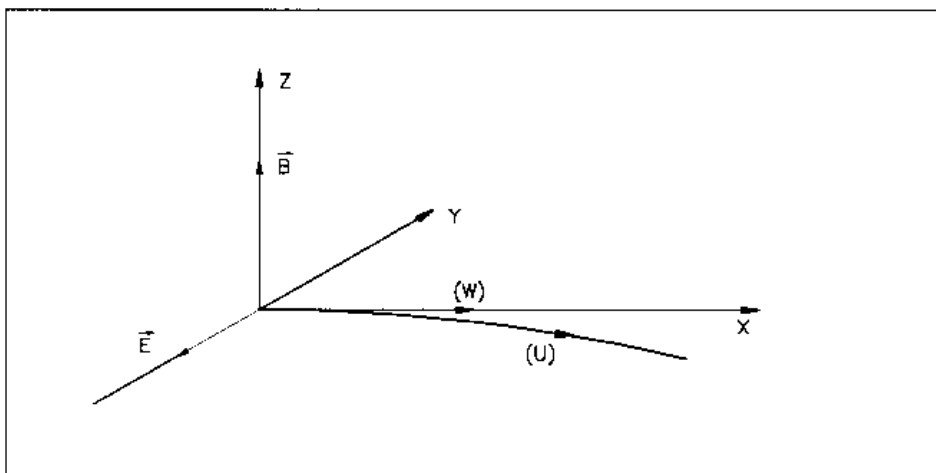
<sup>2</sup>For the geek: details in subroutines *rscal.f* (read input), *scal0.f*, *scaler.f*, *scalin.f*.<sup>3</sup>Was developed in 2009 to simulate RHIC spin flipper, and AGS Booster AC dipole later.

**SEPARA**<sup>1</sup>**Wien Filter - analytical simulation** $IA, XL, E, B,$  $IA = 0$  : element inactive $IA = 1$  : horizontal separation $IA = 2$  : vertical separation ;

Length of the separator ; electric field ; magnetic field.

0-2, m,  
V/m, T

I, 3\*E



Horizontal separation between a wanted particle, (W), and an unwanted particle, (U).  
 (W) undergoes a linear motion while (U) undergoes a cycloidal motion.

<sup>1</sup> SEPARA must be preceded by PARTICUL for the definition of mass and charge of the particles.

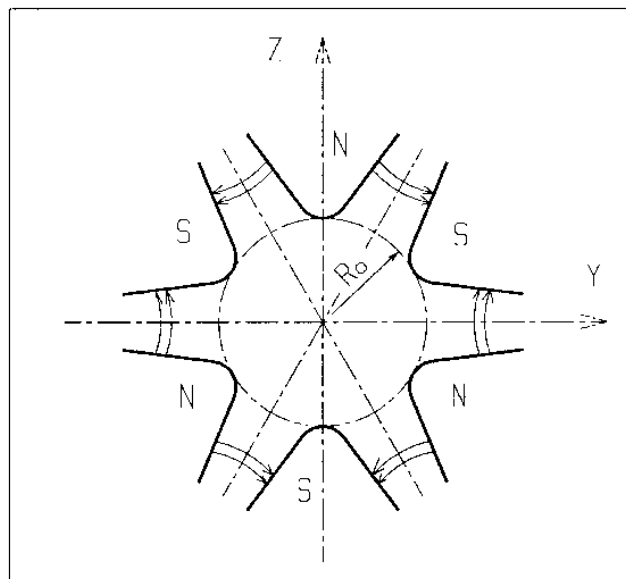
**SEXQUAD****Sharp edge magnetic multipole**

$$B_Z |_{Z=0} = B_0 \left( \frac{N}{R_0} Y + \frac{B}{R_0^2} Y^2 + \frac{G}{R_0^3} Y^3 \right)$$

<i>IL</i>	<i>IL</i> = 1, 2[×10 <sup>n</sup> ], 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2[×10 <sup>n</sup> ], 7	I
<i>XL, R<sub>0</sub>, B<sub>0</sub></i>	Length of the element ; normalization distance ; field	2*cm, kG	3*E
<i>N, EB1, EB2, EG1, EG2</i>	Coefficients for the calculation of B. if <i>Y</i> > 0 : <i>B</i> = <i>EB1</i> and <i>G</i> = <i>EG1</i> ; if <i>Y</i> < 0 : <i>B</i> = <i>EB2</i> and <i>G</i> = <i>EG2</i> .	5*no dim	5*E
<i>XPAS</i>	Integration step	cm	E
<i>KPOS, XCE, YCE, ALE</i>	<i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt.	1-2, 2*cm, rad	I, 3*E

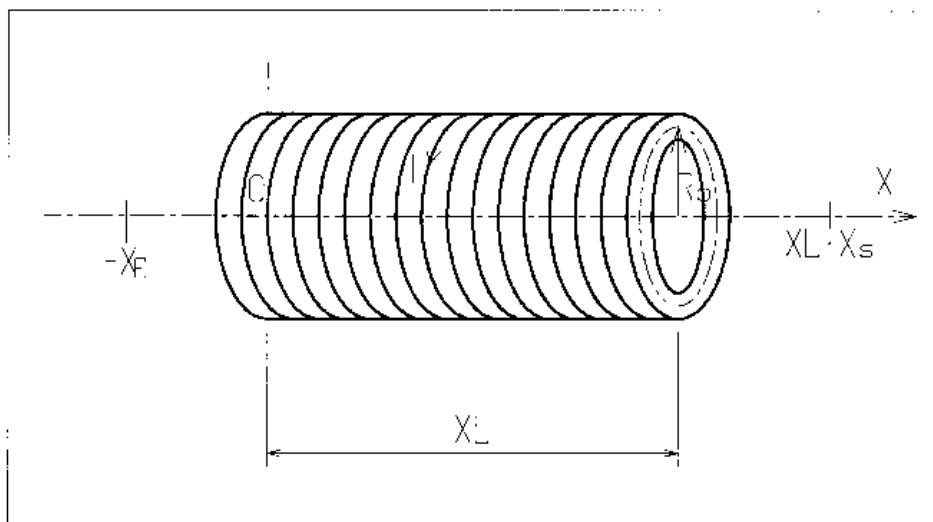
**SEXTUPOL****Sextupole Magnet**

$IL$	$IL = 1, 2[\times 10^n], 7$ : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	$0-2[\times 10^n], 7$	I
$XL, R_0, B_0$	Length ; radius and field at pole tip of the element	$2^*cm, kG$	$3^*E$
$X_E, \lambda_E$	<b>Entrance face :</b> Integration zone ; fringe field extent ( $\lambda_E = 0$ for sharp edge)	$2^*cm$	$2^*E$
$NCE, C_0 - C_5$	$NCE = \text{unused}$ $C_0 - C_5 = \text{Fringe field coefficients such that}$ $G(s) = G_0/(1 + \exp P(s))$ , with $G_0 = B_0/R_0^2$ and $P(s) = \sum_{i=0}^5 C_i (s/\lambda)^i$	$0-6, 6^*$ no dim	I, $6^*E$
$X_S, \lambda_S$	<b>Exit face :</b> Parameters for the exit fringe field ; see entrance	$2^*cm$	$2^*E$
$NCS, C_0 - C_5$		$0-6, 6^*$ no dim	I, $6^*E$
$XPAS$	Integration step	cm	E
$KPOS, XCE, YCE, ALE$	$KPOS=1$ : element aligned, $2$ : misaligned ; shifts, tilt.	$1-2, 2^*cm, rad$	I, $3^*E$



Sextupole magnet

SOLENOID	Solenoid	(page 140 in PART A)	
$IL$	$IL = 1, 2[\times 10^n], 7$ : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	$0-2[\times 10^n], 7$	I
$XL, R_0, B_0 [, MODL]$	Length ; radius ; asymptotic field ( $=\mu_0 NI/XL$ ) ; $MODL=1$ : default, $r$ -extrapolation from on-axis field model, $MODL=2$ : elliptic integrals method	$2^*cm, kG [, no dim]$	$3^*E$
$X_E, X_S$	Entrance and exit integration zones	$2^*cm$	$2^*E$
$XPAS$	Integration step	cm	E
$KPOS, XCE, YCE, ALE$	$KPOS=1$ : element aligned, $2$ : misaligned ; shifts, tilt.	$1-2, 2^*cm, rad$	I, $3^*E$



<b>SPACECHARG</b>	<b>Space charge correction kick</b>		(page 187 in PART A)
UNDER DEVELOPMENT			
<i>LMNT, model [, PRINT]</i>	<i>LMNT</i> = 'all', 'none', or a list of up to 10 keywords ; <i>model</i> = 'KV' or 'Gaussian' ; <i>PRINT</i> : optional, log space charge computation data in zgoubi.SPACECHARG.out.	-, - [-,-]	(1-10)*A, A [,A]
$\lambda$	Linear charge density	<i>C/m</i>	E

<b>SPINR</b>	<b>Spin rotation</b> <i>SCALING</i> can be used to scale $\mu$ .	(page 142 in PART A)	
<i>IOPT</i>	Option	0-2	I
<b>If IOPT=0</b>	Element inactive		
<i>X, X</i>	Unused	2*unused	2*E
<b>If IOPT=1</b>	<b>Axis and spin precession angle values.</b>		
$\phi, \mu$	Angle (in $(X, Y)$ plane) between X-axis and spin precession axis, $\phi \in [-180, 180]$ ; spin precession angle, $\mu \in [0, 180]$ .	deg, deg	2*E
<b>If IOPT=2</b>	<b>Given the spin precession axis direction, <math>\phi</math>, in the horizontal plane, the spin precession angle follows a function of the Lorentz factor : <math>\mu(\gamma) = \left(\frac{B}{B_0}\right)^2 \times \left(C_0 + \frac{C_1}{\gamma} + \frac{C_2}{\gamma^2} + \frac{C_3}{\gamma^3}\right)</math></b>		
$\phi, B, B_0,$ $C_0, C_1, C_2, C_3$	Angle (in $(X, Y)$ plane) between the X-axis and the spin precession axis ; six coefficients that define the spin precession around that axis.	deg.; 6*no dim	7*E

**SPNPRNL**                      **Store spin coordinates in file *FNAME***

*FNAME*<sup>1</sup>                      Name of storage file (*e.g.*, zgoubi.spn)                      A80

**SPNSTORE**                      **Store spin coordinates every *IP* other pass**

*FNAME*<sup>1</sup>                      Name of storage file (*e.g.*, zgoubi.spn) [ ; label(s) of the element(s)                      A80  
[,*LABEL*(s)]<sup>2</sup>                      at the exit of which the store occurs (10 labels maximum)].                      [, 1-10\*A20]

*IP*                      Store every *IP* other pass (when using *REBELOTE*                      I  
with  $NPASS \geq IP - 1$ ).

<sup>1</sup> *FNAME* = 'none' will inhibit printing.

<sup>2</sup> If first *LABEL* = 'none' then printing will be inhibited.



**SPNPRT** [,PRINT]  
[,MATRIX]

**Print spin coordinates, compute spin matrix**

(page 208 in PART A)

Print spin coordinates in zgoubi.res, at the location where this keyword is introduced in the structure.

A label 'PRINT' is optional, if it appears, then local data are stored in the file zgoubi.SPNPRT.Out. That file is opened at the first occurrence of SPNPRT and left open until **zgoubi** execution is completed.

A label 'MATRIX' is optional, if it appears, then computation of spin rotation matrix is performed. This however assumes appropriate trajectory and spin sampling (see Sec. 6.5, p. 208).

### • Example : Spin matrix

In the example below, SPNPRT followed by LABEL=MATRIX is a request for the computation of spin transport matrix of the AGS warm Siberian (an oxymoron?) snake (using its 3D OPERA field map, here). Three matrices will be computed in this example (this number is arbitrary), as OBJET defines 3 successive groups of 3 particles each, with each group its specific relative rigidity  $D = \frac{B\rho}{BORO} = 1, 0.98855, 0.090783$  (typically in the AGS: extraction rigidity at  $G\gamma = 45.5$ , rigidity at the strongest intrinsic resonance  $G\gamma = 36 + \nu_y \approx 44.98$ , and injection rigidity at  $G\gamma = 4.5$ ).

The AGS warm snake 3-D OPERA field map used here, table55.map, can be found here:

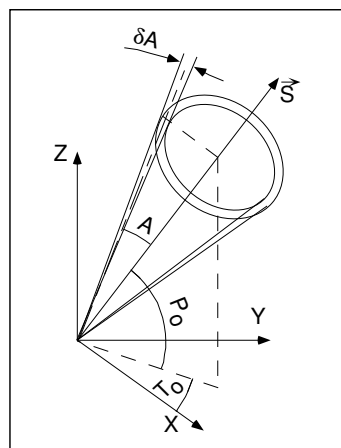
<https://sourceforge.net/p/zgoubi/code/HEAD/tree/trunk/examples/AGS/snakeFieldMaps/Wsnk3D/>.

```
Spin transport matrices of AGS warm Siberian snake, at 3 different energies.
'OBJET'          ! This data list may be copy-pasted and run, as is.
79.366778931425273 *1D3          ! Reference rigidity of the problem.
2
9 3          ! 9 particles, 3 different momenta.
0. 0. 0. 0. 0. 1. 'o'          ! First 3-partile group; D=1, rigidity is BORO above.
0. 0. 0. 0. 0. 1. 'o'
0. 0. 0. 0. 0. 1. 'o'
0. 0. 0. 0. 0. 0.98855355391551 'o'          ! 2nd 3-particle group, rigidity = D*BORO.
0. 0. 0. 0. 0. 0.98855355391551 'o'
0. 0. 0. 0. 0. 0.98855355391551 'o'
0. 0. 0. 0. 0. 9.078330315E-002 'o'          ! 2nd 3-particle group, rigidity = D*BORO.
0. 0. 0. 0. 0. 9.078330315E-002 'o'
0. 0. 0. 0. 0. 9.078330315E-002 'o'
1 1 1 1 1 1 1 1 1
'PARTICUL'          ! proton data are necessary for spin tracking.
PROTON
'SPNTRK'
4          ! Initial spins of the 9 particles, in that order for matrix computation.
1. 0. 0.
0. 1. 0.
0. 0. 1.
1. 0. 0.
0. 1. 0.
0. 0. 1.
1. 0. 0.
0. 1. 0.
0. 0. 1.

'TOSCA'          ! The AGS
0 20
1.e1 100. 100. 100.
HEADER_0 wsnake
801 29 29 12.1          ! The map is a 801x29x29 node 3D mesh.
table55.tab          ! AGS warm snake 3-D OPERA field map.
4          ! Initial spins of the 9 particles, in that order for matrix computation.
0 0 0 0
2
.1
2 0. .0 0. 0.

'SPNPRT' MATRIX
'END'
```

<b>SPNTRK</b> <sup>1</sup>	<b>Spin tracking switch, and spin coordinate initialization</b> ( <i>Spin coordinate initializations are ignored if using OBJET[KOBJ=3]</i> )	(page 188 in PART A)	
$KSO$ [. $KSO2$ ], [ <i>startAt N</i> ]	$KSO=0$ : spin tracking [switched] off <sup>2</sup> ; $KSO=-1$ : spin tracking resumes. Otherwise : as stated below; “startAt N” is a request to only start spin tracking at turn number N	-1 or 0 or 1-3 Or 4[.1] or 5	I[.J] [A I]
<b>If KSO = 1 – 3</b>	$KSO = 1$ (respectively 2, 3) : all particles have their spin automatically set to (1,0,0), longitudinal (respectively (0,1,0), horizontal and (0,0,1), vertical)		
<b>If KSO = 4</b>	Repeat $IMAX$ times (for $IMAX$ particles in ‘OBJET’) the following sequence :		
$S_X, S_Y, S_Z$	$X, Y$ and $Z$ initial components of the initial spin.	3*no dim	3*E
<b>If KSO = 4.1</b>			
$S_X, S_Y, S_Z$	$X, Y$ and $Z$ components of the initial spins. These will be assigned to all particles.	3*no dim	3*E
<b>If KSO = 4.2</b>	Read spins from ancillary file - <i>UNDER DEVELOPMENT</i>		
$FNAME$ <sup>3</sup>	File name		A80
<b>If KSO = 5</b>	Random distribution in a cone (see figure). Enter the following two sequences : <i>UNDER DEVELOPMENT - TO BE COMPLETED IN spn.f.</i>		
$TO, PO$ $A, \delta A$	Angles of average polarization direction. $A$ = angle of the cone around ( $TO, PO$ ) ; $\delta A$ = standard deviation of distribution around $A$	2*rad 2*rad	2*E 2*E
$IR$	Random sequence seed	$\lesssim 10^6$	I



Option  $KSO = 5$  : spins are distributed at random within an annular strip with standard deviation  $\delta A$ , centered at an angle  $A$  with respect to the axis of mean polarization ( $\vec{S}$ ) defined by  $T_0$  and  $P_0$ .

<sup>1</sup> *SPNTRK* must be preceded by *PARTICUL* for the definition of  $G$  and mass.

<sup>2</sup> Spin tracking can be switched off at any location in *zgoubi.dat* data list using  $KSO=0$ , and further away resumed using  $KSO=-1$ .

<sup>3</sup> *FNAME* contains the spin components. Formatting should be has follows :

*TXT, SXI, SYI, SZI, SMI, SX, SY, SZ, SM*

with *TXT* a (arbitrary) 3-character string, *SXI, SYI, SZI, SMI* the initial spin components and their modulus (unused), *SX, SY, SZ, SM* the current spin components (these will be tracked further) and their modulus(unused).

<b>SRLOSS</b>	<b>Synchrotron radiation energy loss</b>	(page 190 in PART A)
<i>KSR</i> [.i] [, <i>PRINT</i> ] [, <i>KSOK</i> ]	On/off switch ; <i>i</i> = 1 for info output into <code>zgoubi.SRLOSS.Out</code> ; optional, <i>PRINT</i> , anywhere beyond <i>KSR</i> : has the same effect as <i>i</i> = 1; under development: Sokolov-Ternov switch, optional (1/0 for on/off, default is off).	0 or 1[.1], -, 0 or 1 I[.I] [,A] [,I]
<hr/>		
Next line, two possibilities :		
<b>either 1 KEYWORD[{<i>LABEL1</i> list}] subject to SR (possibly, all KEYWORDS) :</b>		
<i>KWRD</i> [{ <i>lblst</i> }] [, <i>STR</i> [, <i>List</i> ]]	Apply SR loss to a particular KEYWORD, or to <i>KWRD</i> = 'ALL' or 'all', and in the former case: { <i>lblst</i> }={ <i>lbl1</i> , <i>lbl2</i> , ...} : a list of up to 5 <i>LABEL1</i> s to which this applies ; <i>STR</i> , optional : <i>STR</i> = 'scale' will scale fields following energy loss, <i>List</i> : a list of up to 10 keywords to which this scaling applies.	A[1-5A] [,A [,1-10A]]
<b>or, up to 10 keywords (no <i>LABEL1</i> list in that case) subject to SR :</b>		
<i>KWRD1</i> [, <i>KWRD2</i> [, <i>KWRD3</i> ...]] [, <i>STR</i> [, <i>List</i> ]]	A list of up to 10 KEYWORDS to apply SR loss to ; <i>STR</i> , optional : <i>STR</i> = 'scale' will scale fields following energy loss, <i>List</i> : a list of up to 10 keywords to which this scaling applies.	1-10A [,A [,1-10A]]
<hr/>		
<i>Option</i> , seed	Option = 0 or 1 : SR causes dp only ; Option = 2 : SR causes both dp and angle kick (under development).	1 - 3, > 10 <sup>5</sup> I1, I6

**SRPRNT****Print SR loss statistics** into zgoubi.res

<b>SVDOC</b>	<b>SVD orbit correction UNDER DEVELOPEMENT - Feb. 2019</b>	(page 191 in PART A)	
INF, KWRT [, NAMFI]	INF: 0/1 (on/off); KWRT: 0/1 (verbatim); NAMFI: name of $A^{-1}$ and $A$ matrix log file (optional). default is zgoubi.SVD_Amatrix.out).	0/1, 0/1, any	2*I, A
<i>LABEL1</i> [, <i>LABEL2</i> , [...]]	PU label(s), at which beam data are to be computed/recorded. If some " <i>LABELi</i> " in this list actually does not appear in zgoubi.dat optical sequence, then it is peacefully ignored ; wild card accepted: '* <i>LABEL</i> ' or ' <i>LABEL</i> *' or '* <i>LABEL</i> *'.	string, string, ...	N*A20
<i>LABEL1</i> [, <i>LABEL2</i> , [...]]	Corrector label(s).	string, string, ...	N*A20
HKIC, VKIC	Maximum corrector kick angle values.		
JNF, NDAT, ITER	JNF: 0/1, additional trials on/off; number of data to be changed to set defect; ITER: number of additional trials to be performed.	0/1, any, any	3*I
<b>If <i>NDAT</i> <math>\neq</math> 0 repeat NDAT times:</b>			
Keyword, param	Keyword to be changed, its parameter number	string, any	A20, I

### • Example : SVD OC

In this example the orbit correction algorithm is applied to a periodic structure. FIT first find the closed orbit coordinates. Following that the SVD OC procedure is launched to compute the  $A$  matrix, as described in PART A of the guide. Once  $A$  is known, in a first step  $A^{-1}$  is applied to the correctors; in a second step, FIT will find the coordinates of the corrected periodic orbit. The latter is expected much damped if computation of  $A$  went on properly. In this example the process of generating a new closed orbit, and finding the corrected closed orbit, is repeated 5 times (it could be more, for the sake of accuracy on statistics, for instance); at each iteration of the process ERRORS generates the necessary lattice errors.

Corrected orbits are logged in zgoubi.SVDOrbits.out.

```
Title. The 'OPTICAL SEQUENCE', below, is assumed periodic.
'OBJET'
1000.00                                !   reference rigidity (kG.cm).
2
1 1
0.E+00   0.E+00   0.E+00   0.E+00   0. 1. 'o' ! This does not have to be the closed orbit, as FIT
1                                               ! will first find it.
...
'ERRORS'                                ! Uncomment PRINT to get ERRORS data logged in zgoubi.ERRORS.out.
0 2 123456 ! PRINT                        ! ERRORS is OFF: SVD OC will set it on for subsequent trials.
MULTIPOL{HDEF} 1 BP A U 0. 1.5e3  3
MULTIPOL{VDEF} 1 BP A U 0. 1.5e3  3
...
! Start or INCLUDE optical sequence here:
...
! Some vertical corrector belonging in 'VKIC' family
'MULTIPOL' VKIC
[MULTIPOL data list]
...
! Some zero-length horizontal BPM belonging in 'HMON' family
'DRIFT'   HMON
0.0000
...
! End optical sequence here

'FIT'                                ! This finds the periodic orbit:
4                                     ! 4 variables:
1 30 0 [-.1,.1]                       ! Vary Y0,
1 31 0 [-.1,.1]                       ! Vary T0,
1 32 0 [-.1,.1]                       ! Vary Z0,
1 33 0 [-.1,.1]                       ! Vary P0.
4                                     ! 4 constraints:   (default penalty=1e-10):
3.1 1 2 #End 0. 1. 0                  ! Yfinal = Y0,
3.1 1 2 #End 0. 1. 0                  ! Tfinal = T0,
3.1 1 2 #End 0. 1. 0                  ! Zfinal = Z0,
3.1 1 2 #End 0. 1. 0                  ! Pfinal = P0.

'SVD OC'
1 0 zgoubi.SVDOrbits.out
PUH{HMON} PUV{VMON}                   ! Names of PU-H, -V, -HV families (max each =MPULAB/3=1).
CRH{HKIC} CRV{VKIC}                   ! Names of corr-H, -V families; kick (rad).
1.e-3   1e-3                           ! Kick values for computation of svd matrix (rad).
1 2 5 ! 1: triggers ERRORS; 2: 2 data are changed prior to that, following 2 lines; 5 trials.
ERRORS 12 ! parameter 12 in ERRORS: value of HDEF dipole field: will set H orbit,
ERRORS 22 ! parameter 22 in ERRORS: value of VDEF dipole field: will set V orbit.
!                                     step 1: A^-1 is applied to correctors.
!                                     Step 2: when back down to FIT (from here), FIT will get the corrected periodic orbit.

'END'
```

### • More examples can be found in

<https://sourceforge.net/p/zgoubi/code/HEAD/tree/branches/exemples/KEYWORDS/SVD OC/>

<b>SYNRAD</b>	<b>Synchrotron radiation spectral-angular densities</b>		
<b>KSR</b>	Switch 0 : inhibit SR calculations 1 : start 2 : stop	0-2	I
<b>If KSR = 0</b>			
<i>D1, D2, D3</i>	Dummies		3*E
<b>If KSR = 1</b>			
<i>X0, Y0, Z0</i>	Observer position in frame of magnet next to <i>SYNRAD</i>	3*m	3*E
<b>If KSR = 2</b>			
$\nu_1, \nu_2, N$	Frequency range and sampling	2*eV, no dim	2*E, I

**SYSTEM****System call***NCMD*

The number of calls to follow.

 $\geq 0$ 

I

*NCMD lines follow, one command per line.*



### • Example : SYSTEM

The first occurrence of the command (top region in the data list below) allows establishing links from remote folders to the current one (the one in which **zgoubi** is presently run). These folders happen to contain files appearing in the *SCALING* command, as well as OPERA field maps of the siberian snakes used in subsequent *TOSCA* commands.

The second occurrence of the command (bottom region in the data list below) allows saving **zgoubi.res** output file as resulting from the **zgoubi** run, under a different name.

```

AGS, polarized protons. 2 snakes. t = 145 ms.
'GETFITVAL'
fitVals.data
'OBJET'      LBL_OBjfit
7069.3668040036146
5
.01 .01 .01 .01 0. .0001
 0. 0. 0. 0. 0. 1.  'o'
'FAISCEAU'
'SYSTEM'
2
ln -sf /rap/lattice_tools/zgoubi/AgsZgoubiModel/snakeFieldMaps/TOSC3D/Csnk3D .
ln -sf /rap/lattice_tools/zgoubi/AgsZgoubiModel/snakeFieldMaps/TOSC3D/Wsnk3D .
'SCALING'    LBL_SCLfit
1 17
AGSMM *AF *BF *CF  !# of params 2Bchanged. dB0 (FIT#3) dB1 (FIT#4) dB2
-1 3 13 3E-20 14 -1.473595E-03 15 8.01942
1.00000000 ! (FIT #6)
1
AGSMM *AD *BD *CD  !# of params 2Bchanged. dB0 (FIT#9) dB1 (FIT#10) dB2
-1 3 13 9E-20 14 -1.011857E-03 15 -1.70597
1.000000 ! (FIT#12)
1
AGSQVAD QH_*      !# of params 2Bchanged. (FIT#15)
-1 1 15 0.0
1.000000
1
AGSQVAD QV_* QP_* !# of params 2Bchanged. (FIT#15)
-1 1 15 0.0
1.000000
1
MULTIPOL QJUMP_*
-1
7.06936680E+00
1
MULTIPOL COH1
1,10
./Csnk3D/Hlx68.2_Sol142.3/CHREF+_dipolCORR.scal
1 4
MULTIPOL COV1
1,10
./Csnk3D/Hlx68.2_Sol142.3/CHREF+_dipolCORR.scal
1 5
MULTIPOL COH2
1,10
./Csnk3D/Hlx68.2_Sol142.3/CHREF+_dipolCORR.scal
1 6
MULTIPOL COV2
1,10
./Csnk3D/Hlx68.2_Sol142.3/CHREF+_dipolCORR.scal
1 7
MULTIPOL WOH1
1,10
./Csnk3D/Hlx68.2_Sol142.3/CHREF+_dipolCORR.scal_51.8_0_0
1 4
MULTIPOL WOV1
1,10
./Csnk3D/Hlx68.2_Sol142.3/CHREF+_dipolCORR.scal_51.8_0_0
1 5
MULTIPOL WOH2
1,10
./Csnk3D/Hlx68.2_Sol142.3/CHREF+_dipolCORR.scal_51.8_0_0
1 6
MULTIPOL WOV2
1,10
./Csnk3D/Hlx68.2_Sol142.3/CHREF+_dipolCORR.scal_51.8_0_0
1 7
CHANGREF WSNKE
1,12
./Csnk3D/Hlx68.2_Sol142.3/CHREF+_dipolCORR.scal_51.8_0_0
1 1 1 3
CHANGREF WSNKO
1,12
./Csnk3D/Hlx68.2_Sol142.3/CHREF+_dipolCORR.scal_51.8_0_0
1 1 1 2
CHANGREF CSNKE
1,12
./Csnk3D/Hlx68.2_Sol142.3/CHREF+_dipolCORR.scal
1 1 1 3
CHANGREF CSNKO
1,12
./Csnk3D/Hlx68.2_Sol142.3/CHREF+_dipolCORR.scal
1 1 1 2
'MARKER' #Start
'OPTIONS'
1 1 ! options
WRITE OFF
'AGSMM' MM_A01BF
0
3 0 0 0.00000E+00 1.00000E+00 1.00000E+00 1.0
2.1 1 0. 1 0.
0. 0. 10.00 4.0 0.800 0.00 0.00 0.00 0.00 0.0 0. 0. 0.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.
0. 0. 10.00 4.0 0.800 0.00 0.00 0.00 0.00 0.0 0. 0. 0.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
3.0 Dip MM_A01BF
4 0. 0. 0. 0. 0.
[.....]
'TOSCA'
0 0
1.e-3 100. 100. 100.
HEADER_4 csnake
281 29 29 15.2 .682 .423
./Csnk3D/Hlx68.2_Sol142.3/b_table_for_Helix_3T.tab
./Csnk3D/Hlx68.2_Sol142.3/b_table_for_Solen.tab
0 0 0 0
2
.1
2 0. .00 0. 0.
[.....]
'TOSCA'
0 0
1.e-3 100. 100. 100.
HEADER_4 csnake
281 29 29 15.2 .518 1.e-30
./Csnk3D/Hlx68.2_Sol142.3/b_table_for_Helix_3T.tab_2
./Csnk3D/Hlx68.2_Sol142.3/b_table_for_Solen.tab_2
0 0 0 0
2
.1
2 0. .00 0. 0.
[.....]
'MARKER' #End
'FAISCEAU'
'OPTIONS'
1 1 ! options
WRITE ON
'FAISCEAU'
'MATRIX'
1 11
'SYSTEM'
1
cp zgoubi.res zgoubi.res_save
'END'

```

### • Example : SYSTEM

An additional example can be found page 285

<b>TOSCA</b>		<b>2-D and 3-D Cartesian or cylindrical mesh field map</b>		(page 143 in PART A)	
		dim.	data type	FIT num.	
<i>IC, IL</i>	<i>IC</i> = 1, 2 : print the map <i>IL</i> = 1, 2[ $\times 10^n$ ], 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2; 0-2[ $\times 10^n$ ], 7	2*I	1, 2	
<i>BNORM, XN, YN, ZN</i>	Field and X-,Y-,Z-coordinate unit conversion coefficients. Convert values as read from map file, to kG and cm or rad.	4*UnitConv.	4*E	10-13	
<i>TITL</i>	Title. Include "FLIP" to get field map X-flipped. Include "HEADER n" if <i>FNAME</i> starts with $n \geq 1$ header lines. Include "ZroBXY" to force $B_X = B_Y = 0$ at all $Z=0$ (only applies for $MOD=15$ and $MOD=24$ ). Include "RHIC_helix" to normalize <i>BNORM</i> to measured value <sup>1</sup> ; in that case, if $SCALING[TOSCA]$ is used, $sign(BNORM*SCAL) == sign(BNORM)$ [field sign is independent of sign(SCAL)].		A80		
<i>IX, IY, IZ, 2</i> <i>MOD[.MOD2 [ , a(i), i=1, MOD2]]</i>	Number of nodes of the mesh in the <i>X, Y</i> and <i>Z</i> directions, <i>IZ</i> = 1 for a 2-D map ; <i>MOD</i> and <i>MOD2</i> : field map style, see table next page ; $a(i = 1, MOD2)$ are the field normalisation factors.	$\leq MXX, \leq MXY,$ $\leq IZ, [0-22.1-9], -$	3*I, I[,I [, MOD2 *E]]	20-24 [, 25-27]	
Next NF lines : <i>FNAME</i> <sup>3</sup>	Map file name(s), one line per name. If $MOD=0$ : $NF = 1 + [IZ/2]$ , the <i>NF</i> 2-D maps are for $0 \leq Z \leq Z_{max}$ , they are symmetrized with respect to the $Z(1) = 0$ plane. If $MOD=1$ : $NF = IZ$ , no symmetry assumed ; $Z(1) = Z_{max}$ , $Z(1 + [IZ/2]) = 0$ and $Z(NF) = -Z_{max}$ . If $MOD=12$ : a single <i>FNAME</i> file contains the all 3-D volume. <i>MOD=15, 20-22, etc.</i> : see table page 348		A80		
<i>ID, A, B, C [ , A', B', C', A'', etc., if ID ≥ 2]</i>	Integration boundary. Ineffective when $ID = 0$ . $ID = -1, 1$ or $\geq 2$ : as for <i>CARTEMES</i>	$\geq -1, \text{cm},$ $2*\text{no dim } [,idem]$	I,3*E [ 3*E,etc.]	30-33 [ 34-36, ...]	
<i>IODRE</i>	If $IZ = 1 : 2, 25$ or 4 as in <i>CARTEMES</i> ; unused if $IZ \neq 1$ .	2, 25 or 4	I	40	
<i>XPAS</i>	Integration step	cm	E	50	
<b>If Cartesian mesh (see MOD) :</b> <i>KPOS, XCE, YCE, ALE</i>	<i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt	1-2, 2*cm, rad	I, 3*E	60-63	
<b>If polar mesh :</b> <i>KPOS</i>	As for <i>POLARMES</i> . Normally 2.	1-2	I	60	
<b>If KPOS = 2 :</b> <i>RE, TE, RS, TS</i>		cm, rad, cm, rad	4*E	70-73	

<sup>1</sup>B field along RHIC Snake helix versus helix current, normalized to field measurements or not, is managed in program *toscac.f*.

<sup>1</sup>*MXX, MXY, IZ* may be changed, they are stated in the include file *PARIZ.H*.

<sup>2</sup>Case of 2-D field maps : Each file *FNAME(K)* contains the field specific to elevation  $Z(K)$  and must be formatted according to the following *FORTRAN* read sequence (that usually fits *TOSCA* code *OUTPUTS* - details and possible updates are to be found in the source file '*fmapw.f*') :

```
DO JF = 1, NF
OPEN (UNIT = NL, FILE = FNAME(JF), STATUS = 'OLD' [,FORM='UNFORMATTED'])
DO J = 1, JY ; DO I = 1, IX
READ(NL,*) Y(I), Z(JF), X(I), BY(J,I), BZ(J,I), BX(J,I)
ENDDO ; ENDDO
NL = NL + 1
ENDDO
```

node coordinates, field components at node

*BX* and *BY* are assumed zero at all nodes of the 2-D mesh, regardless of *BX(J,1,I)*, *BY(J,1,I)* values.

Case of 3-D field maps :

```
DO JF = 1, NF
OPEN (UNIT = NL, FILE = FNAME(JF), STATUS = 'OLD' [,FORM='UNFORMATTED'])
DO J = 1, JY ; DO K = 1, KX ; DO I = 1, IX
READ(NL,*) Y(I), Z(K), X(I), BY(J,K,I), BZ(J,K,I), BX(J,K,I)
ENDDO ; ENDDO ; ENDDO
NL = NL + 1
ENDDO
```

node coordinates, field components at node

• **Example** : RHIC Siberian snake module, using, or not, “RHIC\_helix” flag.

The two *TOSCA* input data lists below are equivalent, they provide quasi-identical fields. The left one features “RHIC\_helix” as part of the field map “*TITL*” (which normalizes the OPERA field to the input current BNORM (Amps) using measured field versus current data); the right one does not - this is the origin of a very small difference in the field along the helix [72].

Incidentally: these two jobs perform a centering of the helical motion along the helix (that’s the *FIT* constraint), finding the optimal initial vertical trajectory coordinate (*FIT* variable). They end up with a computation of the spin matrix transport along that trajectory. The field map used can be found in zgoubi repository, at

<https://sourceforge.net/p/zgoubi/code/HEAD/tree/trunk/exemples/RHIC/3DSnakes/fieldMaps/>

```

RHIC Siberian snake low field module.
'OBJET'
853.71244451829796e3          ! E=255 GeV.
1
3 1 1 1 1 1          ! Spin matrix computation requires sets (only 1 here)
0. 0. 0. 0. 0. 0.          ! of 3 identical particles.
0. 0. -0.21589188 0. 0.00 1.

'PARTICUL'
PROTON
'SPNTRK'
4
1. 0. 0.          ! Spin vectors for each particle of a 3-set.
0. 1. 0.
0. 0. 1.

'TOSCA'      high_normalized
0 2
-322. 1. 1. 1.          ! BNORM is a currnt, is Amps, here.
HEADER_8      RHIC_helix
361 81 81 15.1 0.246305418719e-4  ! = 1/4.060e4
b_model3a2a322a-x-4_4_y-4_4_z-180_180-integral.table
0 0 0 0
2
.1
2 0.000000000E+00 0.000000000E+00 0.000000000E+00

'FIT'
1
1 42 0 [-3,3]          ! Vary Z_0.
1 1D-10
7.3 1 4 4 0. 1. 0  ! |min(Z)+max(Z)|=0 in element 4=helix -> vertical centering.

'SPNPRT' MATRIX PRINT
'END'

RHIC Siberian snake low field module.
'OBJET'
853.71244451829796e3          ! E=255 GeV.
1
3 1 1 1 1 1          ! Spin matrix computation requires sets (only 1 here)
0. 0. 0. 0. 0. 0.          ! of 3 identical particles.
0. 0. -0.21589188 0. 0.00 1.

'PARTICUL'
PROTON
'SPNTRK'
4
1. 0. 0.          ! Spin vectors for each particle of a 3-set.
0. 1. 0.
0. 0. 1.

'TOSCA'      high_B_notNormalized
0 2
-1.E-03 1. 1. 1.          ! Normalize data read from map: B to kG, and X, Y, Z to cm.
HEADER_8          ! No "RHIC_helix" here!!
361 81 81 15.1 1.
b_model3a2a322a-x-4_4_y-4_4_z-180_180-integral.table
0 0 0 0
2
.1
2 0.000000000E+00 0.000000000E+00 0.000000000E+00

'FIT'
1 nofinal
1 42 0 [-3,3]          ! Vary Z_0.
1 1D-10
3.2 1 4 #End 0. 1. 0  ! |Z_final + Z_0| = 0: this centers the helical trajectory.

'SPNPRT' MATRIX PRINT
'END'

```

<sup>3</sup>For binary files, *FNAME* must begin with 'B\_' or 'b\_'.

• **The various *IZ*, *MOD* and *MOD2* options, when using *TOSCA*.**

**IZ** : number of nodes of the *complete* field map along the *Z* direction (IZ=1 for 2-D)

**MOD, MOD2** : determine the coordinate system, symmetries, reading format and column sequence, etc.

**NF** : number of field map input data files to be declared. Always include mid-plane map.

**Expected columns** : formatting of the coordinates and field data columns in the field map data file(s)

**'Example' example folder** : examples of *zgoubi* runs using field maps can be found in the subfolders of

*zgoubi-code/exemples/KEYWORDS/TOSCA/cartesian* (case  $MOD \leq 19$ ) or

*zgoubi-code/exemples/KEYWORDS/TOSCA/cylindrical* (case  $MOD \geq 20$ ).

The rightmost column below indicates the subfolder of concern, following (*IZ*, *MOD*, *MOD2*) options of interest.

**MOD  $\leq$  19 : Cartesian mesh**

IZ	MOD	.MOD2		NF	Expected columns	Example folder
1	0, 1	none or .1, .2, .3	<b>2-D</b> map. File contains $B_Z(X, Y) _{Z=0}$ , mid-plane antisymmetry assumed. Several different reading formats (see <i>fmapw.f/fmapr3</i> ).	1	Y, Z, X, BY, BZ, BX	IZ-MOD-.MOD2_1-0-none/ (GSI KAOS spectrometer)
1	3	none or .1	<b>2-D</b> map. Used for AGS main magnet. If MOD2=1, $B_Z(X, Y, Z = 0)$ field is perturbed by $(1 + n_1 Y + n_2 Y^2 + n_3 Y^3)$ factor.	1	Special - see example	AGS/usingMainMagnetsMaps (AGS with main magnet maps)
1	15	.1 - .4	<b>2-D</b> map. Up to 4 files to be combined linearly into a new map : field at all node of new map is $\vec{B} = \sum_{i=1}^{MOD2} a_i \vec{B}_i$ . Mid-plane antisymmetry is assumed : each file has to contain $B_Z(X, Y, Z = 0)$ .	1 to 4	Y, Z, X, BY, BZ, BX	IZ-MOD-.MOD2_1-15-.1-.4 (EMMA FFAG cell)
>1	0	none	<b>3-D</b> map. Files span upper half of magnet, one per $(X, Y) _{0 \leq Z \leq Z_{max}}$ plane including median plane, mid-plane antisymmetry assumed.	1+IZ/2	Y, Z, X, BY, BZ, BX	IZ-MOD-.MOD2_gt1-0-none/ (GSI KAOS spectrometer)
odd, $\geq 3$	1	none	<b>3-D</b> map. Files span full magnet volume, one file per $(X, Y)$ plane, no symmetry assumed.	IZ	Y, Z, X, BY, BZ, BX	IZ-MOD-.MOD2_gt1-1-none/ (AGS warm helix snake)
>1	12	none	<b>3-D</b> map. Single file, upper half of magnet, mid-plane antisymmetry assumed.	1		
>1	12	.1	<b>3-D</b> map. Single file, whole magnet volume, no symmetry assumed.	1	Y, Z, X, BY, BZ, BX	IZ-MOD-.MOD2_gt1-12-.1/ (AGS warm helix snake)
>1	12	.2	<b>3-D</b> map. Single file, 1/8th of the magnet, symmetry wrt. $(X, Y)_{Z=0}$ , $(X, Z)_{Y=0}$ , $(Y, Z)_{X=0}$ planes.	1		
odd, $\geq 3$	15	.1 - .4	<b>3-D</b> map. Up to 4 files to be combined linearly into a new map, field at all node of new map is $\vec{B} = \sum_{i=1}^{MOD2} a_i \vec{B}_i$ . Each file has to contain $B_{X, Y, Z}(X, Y, Z)$ data over <i>IZ</i> equally <i>Z</i> -spaced $(X, Y)$ planes (no symmetry assumed).	1 to 4		
odd, $\geq 3$	16	.1 - .4	<b>3-D</b> map. Fields from up to 4 maps to be combined linearly into a new field value at particle location, $\vec{B} = \sum_{i=1}^{MOD2} a_i \vec{B}_i$ . Each file has to contain $B_{X, Y, Z}(X, Y, Z)$ data over <i>IZ</i> equally <i>Z</i> -spaced $(X, Y)$ planes.	1 to 4	UNDER DEVELOPMENT	

**MOD  $\geq$  20 : Cylindrical mesh**

IZ	MOD	.MOD2		NF	Expected columns	Example folder
1	25	.1 - .4	<b>2-D</b> map. Up to 4 files to be combined linearly into a new map : at all node of new map $\vec{B} = \sum_{i=1}^{MOD2} a_i \vec{B}_i$ . Each file contains mid-plane $B_Z(X, Y, Z = 0)$ data, mid-plane antisymmetry is assumed.	1 to 4	Y, Z, X, BY, BZ, BX	IZ-MOD-.MOD2_1-15-.1-.4 (EMMA FFAG cell)
>1	20, 21		<b>3-D</b> map. Single file. MOD=20 : 1/4 magnet, cylindrical symmetry with respect to $(Y, Z)$ plane and antisymmetry wrt $(X, Y)$ plane. MOD=21 : another type of symmetry (to be documented - see <i>fmapw.f</i> ).	1	$Y(r, \theta)$ , $Z$ , $X(r, \theta)$ , BY, BZ, BX	IZ-MOD-.MOD2_gt1-20 (KEK 150 MeV FFAG)
odd, $\geq 1$	22, 23	.1 - .4	<b>2-D</b> or <b>3-D</b> map. Mid-plane antisymmetry assumed. Up to 4 files can be combined linearly into a new one; each file is $Z \geq 0$ half-magnet. Field at all node of new map is $\vec{B} = \sum_{i=1}^{MOD2} a_i \vec{B}_i$ . Each file has to contain $B_{X, Y, Z}(X, Y, Z)$ data over $(IZ+1)/2$ equally <i>Z</i> -spaced $(X, Y)$ planes. MOD=23 : special, test code (see <i>fmapw.f</i> ).	1 to 4	$Y(r, \theta) = r \cos \theta$ , $Z$ , $X(r, \theta) = r \sin \theta$ , BY, BZ, BX	IZ-MOD-.MOD2_gt1-22-.1-.4 (ERIT FFAG)
odd, $\geq 3$	24		<b>3-D</b> map, full volume. No symmetry assumed. The file contains $B_{X, Y, Z}(X, Y, Z)$ data over <i>IZ</i> equally <i>Z</i> -spaced $(X, Y)$ planes.	1	$\theta$ , $R$ , $Z$ , $B_\theta$ , $BR$ , $BZ$	

**TRANSMAT****Matrix transfer**

<i>IORDRE</i>	Transport matrix order	1-2	I
<i>XL</i>	Length (ineffective, for updating)	m	E
For $IA = 1, 6$ :			
$R(IA, IB), IB = 1, 6$	First order matrix	m, rad	6 lines 6*E each
<b>If IORDRE = 2</b>	Following records <i>only</i> if $IORDRE = 2$		
$T(IA, IB, IC),$	Second order matrix, six 6*6 blocks	m, rad	36 lines 6*E each

**TRAROT****Translation-Rotation***TX, TY, TZ,  
RX, RY, RZ*

Translations, rotations

3\*m, 3\*rad

6\*E

**TWISS**<sup>1</sup>**Compute periodic optical parameters. Log to zgoubi.TWISS.out** (page 210 in PART A)*Hint: use FIT preceding TWISS for closed orbit search in case the latter is not known.*

KTW,  
 FacD, FacA  
 [, SRINT] [, DKINT]

KTW = 0/1/2/3 : Off / as *MATRIX* / add computation of chromaticities / add computation of anharmonicities. Chromatic orbits are at  $\text{FacD} \times D = B\rho/\text{BORO}$ , with  $D$  the momentum sampling in *OBJET* ; *FacA* : unused ;  
 presence of “SRINT” will cause computation of SR integrals<sup>2</sup> ;  
 presence of “DKINT” will cause computation of Derbenev-Kondratenko spin integrals<sup>3</sup>.

0-3, 2\*any I1, 2\*E [,A]  
 [,coupled] [,A] [,A]  
 [, SRINT]  
 [, DKINT]

<sup>1</sup> *TWISS* may be preceded by *FIT[2]*, this allows finding the on- and off-momentum closed orbits prior to computation of respectively on- and off-momentum transport matrices, see example page 352. Prior *FIT[2]* may also find the stable spin precession direction, useful for instance if *DKINT=1* as computation of Derbenev-Kondratenko integrals requires  $\vec{n}_0$  along closed orbit.

<sup>2</sup> Synchrotron radiation integrals are computed along particle number 1 trajectory, it is the responsibility of the user to ensure that particle number 1 is on the closed orbit, for instance using *FIT[2]*.

<sup>3</sup> Derbenev-Kondratenko spin integrals are computed along particle number 1 trajectory, it is the responsibility of the user to ensure that (i) particle number 1 is on the closed orbit, (ii) its initial spin coordinates in *SPNTRK* are on the spin closed orbit.

• **Example** : orbit finding using *FIT*, followed by a *TWISS* procedure

“*KTW=2*” under *TWISS*, in the data sequence below, will cause 4 successive passes through the optical sequence, with the following purposes:

- pass 1: compute the on-momentum  $4 \times 4$  transport matrix  $[T_{ij}]$  and, if the sequence is optically stable, deduce the periodic beam matrix  $\sigma = I \cos \mu + J \sin \mu \equiv [T_{ij}]$
- pass 2 and 3: repeat for off-momentum matrices  $[T_{ij}](\pm \frac{dp}{p})$ ,
- pass 4: transport the on-momentum matrix through the sequence using  $\sigma(s) = [T_{ij}] \sigma(s=0) [\tilde{T}_{ij}]$ , compute chromaticities, etc., print out outcomes to *zgoubi.TWISS.out* and *zgoubi.res*.

These 4 passes will result in

(i) the delivery of *zgoubi.TWISS.out* file, formatted in a similar form to *MADX*'s *twiss* output, namely,

- a header with global parameters (sequence length, tunes, chromaticities, max- $\beta$ 's, etc.)
- a listing of periodic orbit, optical functions and other data, along the structure ;

(ii) a print out to *zgoubi.res* of

- the on-momentum matrix of the optical structure,
- off-momentum matrices at  $\frac{dp}{p} = \pm FacD * dD$ , with *dD* the momentum sampling defined by *OBJET[KOBJ=5]*

(page 308),

- *Twiss* parameters at the end of the structure, in the case of a stable periodic structure,
- momentum compaction, chromaticities, etc.

In the following example, the *FIT* instruction which precedes *TWISS* first finds the orbit (on-momentum at the first pass,  $\pm dp/p$  chromatic orbits at passes 2 and 3 respectively) which will then be the reference trajectory in *OBJET[KOBJ=5]*, in generating the 13 sample rays necessary for computation of the transport matrix.

```
'OBJET'
20015.55          ! Rigidity (kG.cm) for 6 GeV electrons.
5                ! Generate an 13 particle set. Normally used for MATRIX or TWISS computation.
.001 .01 .001 .01 0. .0001 ! Coordinate sampling for MATRIX computation : dY, dT, dZ, dP, ds (unused) and dD.
0. 0. 0. 0. 0. 1.      ! Coordinates of the reference orbit: Y-, T-, Z-, P-, s- (unused) and D-ref,
                        ! these are the coordinates of particle number 1 of the 13-particle set.

'INCLUDE'
1
./myOpticalSequenceFile[fromHere:toThere]

'FIT'
4                ! 4 variables.
1 30 0 [-1.,1.]  ! vary Y0 of reference particle (i.e., particle number 1 of the 13-set)
1 31 0 [-1.,1.]  ! vary T0 of reference particle (i.e., particle number 1 of the 13-set)
1 32 0 [-1.,1.]  ! vary Z0 of reference particle (i.e., particle number 1 of the 13-set)
1 33 0 [-1.,1.]  ! vary P0 of reference particle (i.e., particle number 1 of the 13-set)
4                ! 4 constraints; penalty value is default 1e-10.
3.1 1 2 #End 0. 1. 0 ! particle 1: Y0 initial = Y final
3.1 1 3 #End 0. 1. 0 ! particle 1: T0 initial = T final
3.1 1 4 #End 0. 1. 0 ! particle 1: Z0 initial = Z final
3.1 1 5 #End 0. 1. 0 ! particle 1: P0 initial = P final
'FAISCEAU'
'TWISS'
2 1. 1.          ! KTW = 2, FacD = 1
'END'
```

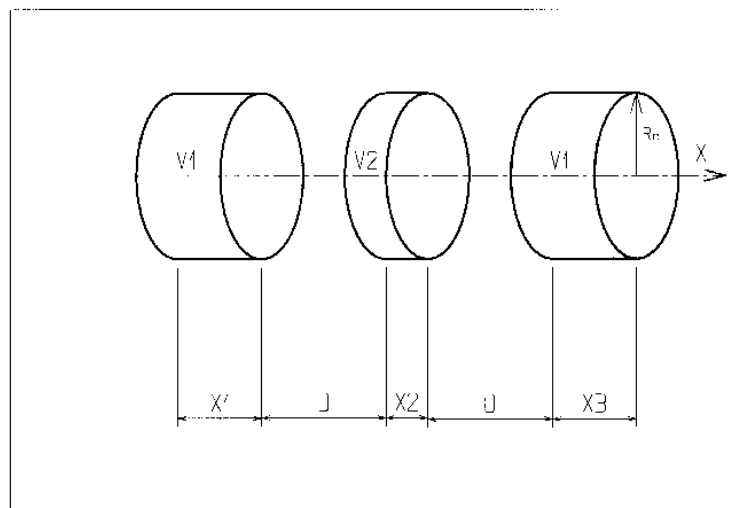


**UNDULATOR**                    **Undulator magnet**

*Under development, to be documented*

**UNIPOT****Unipotential electrostatic lens**

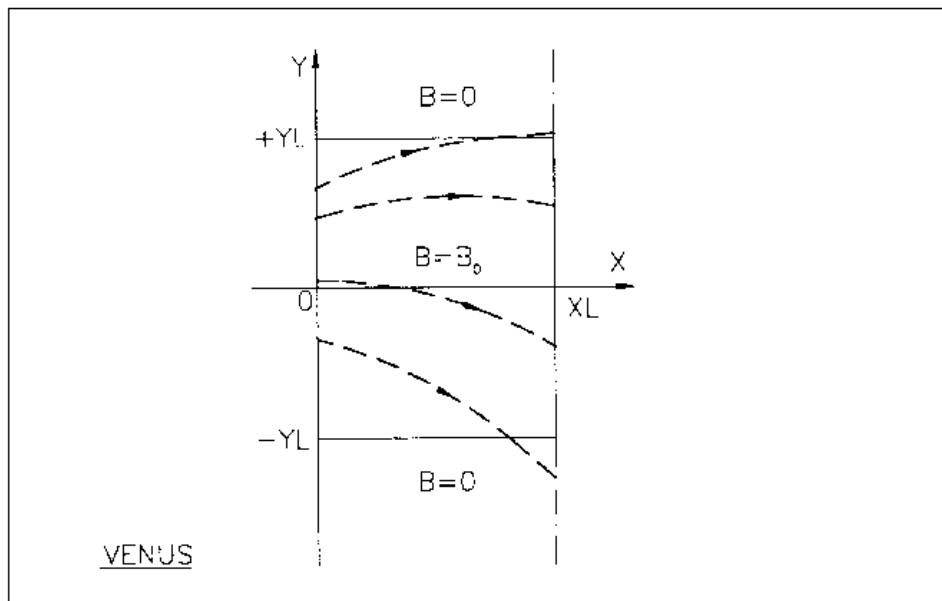
$IL$	$IL = 1, 2[\times 10^n], 7$ : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	$0-2[\times 10^n], 7$	I
$X_1, D, X_2, X_3, R_0$	Length of first tube ; distance between tubes ; length of second and third tubes ; radius	$5*m$	$5*E$
$V_1, V_2$	Potentials	$2*V$	$2*E$
$XPAS$	Integration step	cm	E
$KPOS, XCE, YCE, ALE$	$KPOS=1$ : element aligned, $2$ : misaligned ; shifts, tilt.	$1-2, 2*cm, rad$	$I, 3*E$



**VENUS**

**Simulation of a rectangular dipole magnet**

<i>IL</i>	<i>IL</i> = 1, 2[ $\times 10^n$ ], 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2[ $\times 10^n$ ], 7	I
<i>XL, YL, B<sub>0</sub></i>	Length ; width = $\pm YL$ ; field	2*cm, kG	3*E
<i>XPAS</i>	Integration step	cm	E
<i>KPOS, XCE, YCE, ALE</i>	<i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt.	1-2, 2*cm, rad	I, 3*E



A scheme of a few trajectories through *VENUS* rectangular dipole.

**VFFA****Simulation of a VFFA dipole, or series of**

(page 152 in PART A)

To be documented.

Examples, including input data files, are available at

<https://sourceforge.net/p/zgoubi/code/HEAD/tree/trunk/exemples/FFAG/VFFA/>

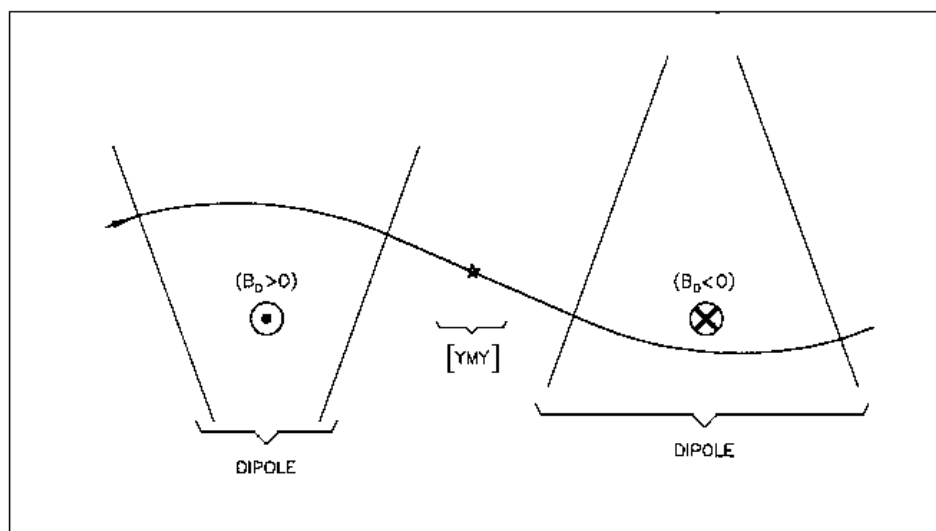
<b>WIENFILT</b> <sup>1</sup>	<b>Wien filter</b>	(page 153 in PART A)	
<i>IL</i>	<i>IL</i> = 1, 2[ $\times 10^n$ ], 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	0-2[ $\times 10^n$ ], 7	I
<i>XL, E, B, HV</i>	Length ; electric field ; magnetic field ; option : element inactive ( <i>HV</i> = 0) horizontal ( <i>HV</i> = 1) or vertical ( <i>HV</i> = 2) separation	m, V/m, T, 0-2	3*E, I
<i>X<sub>E</sub>, <math>\lambda_{E_E}</math>, <math>\lambda_{B_E}</math></i>	<b>Entrance face :</b> Integration zone extent ; fringe field extents, E and B respectively ( $\simeq$ gap height)	3*cm	3*E
<i>C<sub>E0</sub>-C<sub>E5</sub></i> <i>C<sub>B0</sub>-C<sub>B5</sub></i>	Fringe field coefficients for <i>E</i> Fringe field coefficients for <i>B</i>	6*no dim 6*no dim	6*E 6*E
<i>X<sub>S</sub>, <math>\lambda_{E_S}</math>, <math>\lambda_{B_S}</math></i> <i>C<sub>E0</sub>-C<sub>E5</sub></i> <i>C<sub>B0</sub>-C<sub>B5</sub></i>	<b>Exit face :</b> See entrance face	3*cm 6*no dim 6*no dim	3*E 6*E 6*E
<i>XPAS</i>	Integration step	cm	E
<i>KPOS, XCE, YCE, ALE</i>	<i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt.	1-2, 2*cm, rad	I, 3*E

<sup>1</sup> Use *PARTICUL* to declare mass and charge.

YMY

Reverse signs of  $Y$  and  $Z$  reference axes

(page 154 in PART A)

Equivalent to a  $180^\circ$  rotation with respect to  $X$ -axisThe use of  $YMY$  in a sequence of two dipoles of opposite signs.

# **PART C**

## **Examples of input data files and output result files**





## INTRODUCTION

Several examples of the use of **zgoubi** are given here. They show the contents of the input and output data files, and are also intended to help understanding some subtleties of the data definition.

**Example 1:** checks the resolution of the QDD spectrometer SPES 2 of SATURNE Laboratory [78], by means of a *Monte Carlo initial object* and an *analysis of images* at the focal plane with histograms. The *measured field maps* of the spectrometer are used for that purpose. The layout of SPES 2 is given in Fig. 56.

**Example 2:** calculates the *first and second order transfer matrices* of an 800 MeV/c kaon beam line [79] at each of its four foci: at the end of the first separation stage (vertical focus), at the intermediate momentum slit (horizontal focus), at the end of the second separation stage (vertical focus), and at the end of the line (double focusing). The first bending is represented by its *3-D map* previously calculated with the TOSCA magnet code. The second bending is simulated with *DIPOLE*. The layout of the line is given in Fig. 57.

**Example 3:** illustrates *the use of MCDESINT and REBELOTE* with a simulation of the *in-flight decay*

$$K \longrightarrow \mu + \nu$$

in the SATURNE Laboratory spectrometer SPES 3 [45]. The angular acceptance of SPES 3 is  $\pm 50$  mrd horizontally and  $\pm 50$  mrd vertically; its momentum acceptance is  $\pm 40\%$ . The bending magnet is simulated with *DIPOLE*. The layout of SPES 3 is given in Fig. 58.

**Example 4:** illustrates the operation of *the fitting procedure*: a quadrupole triplet is tuned from -0.7/0.3 T to field values leading to transfer coefficients  $R_{12}=16.6$  and  $R_{34}=-.88$  at the end of the beam line. Other example can be found in [80].

**Example 5:** shows the use of the *spin and multi-turn tracking procedures*, applied to the case of the SATURNE 3 GeV synchrotron [10, 13, 81]. Protons with initial vertical spin ( $\vec{S} \equiv \vec{S}_Z$ ) are accelerated through the  $\gamma G = 7 - \nu_Z$  depolarizing resonance. For easier understanding, some results are summarized in Figs. 60, 61 (obtained with the graphic post-processor **zpop**, see Part D).

**Example 6:** shows *ray-tracing through a micro-beam line* that involves *magneto-electrostatic quadrupoles* for the suppression of second order (chromatic) aberrations [9]. The extremely small beam spot sizes involved (less than 1 micrometer) reveal the high accuracy of the ray-tracing (Figs. 62).

Many more examples can be found on the **zgoubi** development web site [8].



## 1 MONTE CARLO IMAGES IN SPES 2

Former Saclay/SATURNE and CERN QDD mass spectrometer [78].

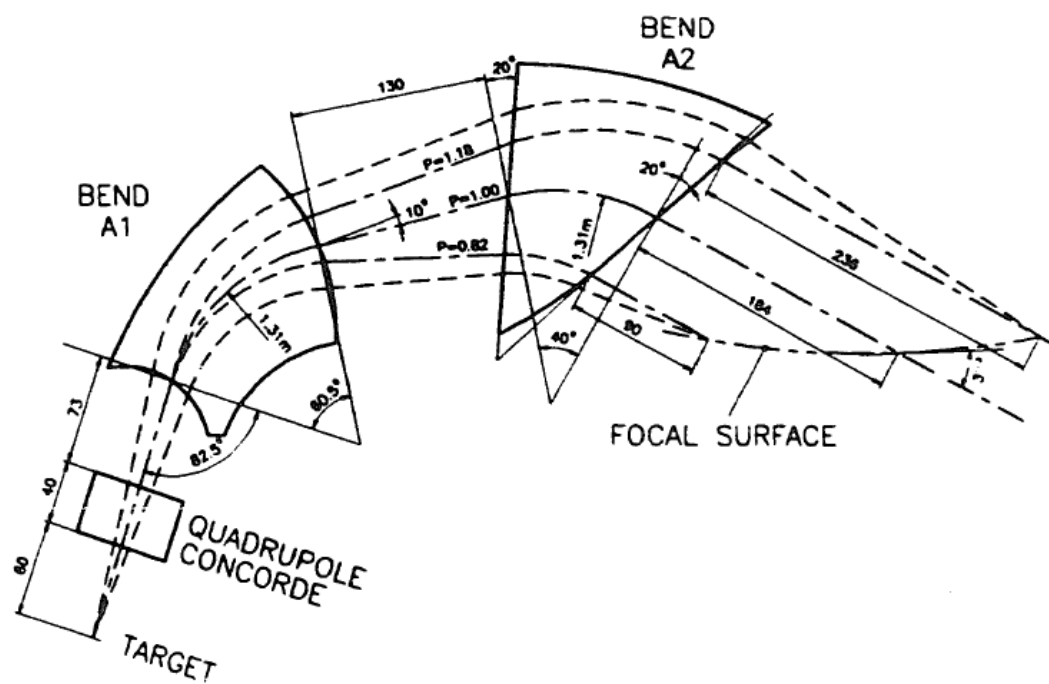


Figure 56: SPES 2 Layout.



4 HISTO

HISTOGRAMME DE LA COORDONNEE PHI
PARTICULES PRIMAIRES ET SECONDAIRES
DANS LA FENETRE : -60.00 / 60.00 (MRD)
NORMALISE

20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1

1234567890123456789012345678901234567890123456789012345678901
2 3 4 5 6 7 8 9

TOTAL COMPTAGE : 10000 SUR 10000
NUMERO DU CANAL MOYEN : 51
COMPTAGE AU " " : 163
VAL. PHYS. AU " " : 3.331E-15 (MRD)
RESOLUTION PAR CANAL : 1.50

PARAMETRES PHYSIQUES DE LA DISTRIBUTION :
COMPTAGE = 10000 PARTICULES
MIN = -50.00 , MAX = 49.99 , MAX-MIN = 99.99 (MRD)
MOYENNE = 0.2838 (MRD)
SIGMA = 28.75 (MRD)

TRAJ l IEX,D,Y,T,Z,P,S,time : 1 0.9980 0.000 -30.24 0.000 44.63 0.0000 0.0000

Excerpt from zgoubi.res : the final momentum resolution histogram at the spectrometer focal surface.

16 HISTO

HISTO OF

HISTOGRAMME DE LA COORDONNEE Y
PARTICULES PRIMAIRES ET SECONDAIRES
DANS LA FENETRE : -0.5000 / 2.000 (CM)
NORMALISE

20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1

1234567890123456789012345678901234567890123456789012345678901
2 3 4 5 6 7 8 9

TOTAL COMPTAGE : 10000 SUR 10000
NUMERO DU CANAL MOYEN : 51
COMPTAGE AU " " : 246
VAL. PHYS. AU " " : 0.750 (CM)
RESOLUTION PAR CANAL : 3.125E-02 (CM)

PARAMETRES PHYSIQUES DE LA DISTRIBUTION :
COMPTAGE = 10000 PARTICULES
MIN = -0.1486 , MAX = 1.652 , MAX-MIN = 1.800 (CM)
MOYENNE = 0.7576 (CM)
SIGMA = 0.4621 (CM)

TRAJ l IEX,D,Y,T,Z,P,S,time : 1 0.9980 0.2475 74.43 -6.2488E-03 -6.929 697.41 0.0000

2 TRANSFER MATRICES ALONG A TWO-STAGE SEPARATION KAON BEAM LINE

800 MeV/c kaon beam line at BNL Alternating Gradient Synchrotron [79]. The line includes two separation Wien filters.

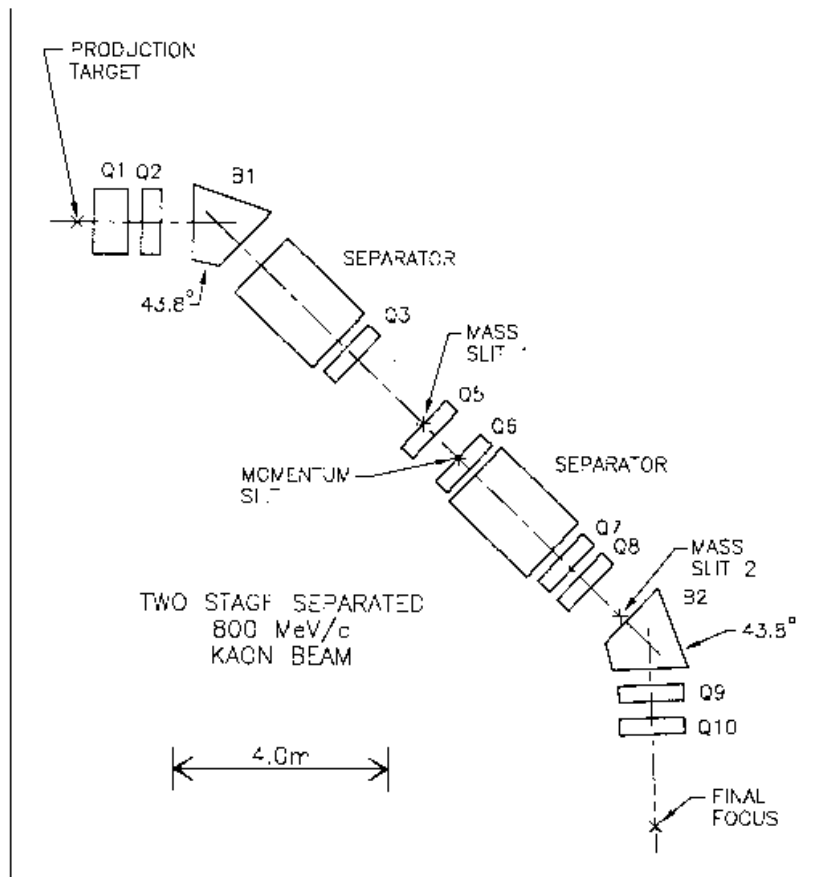
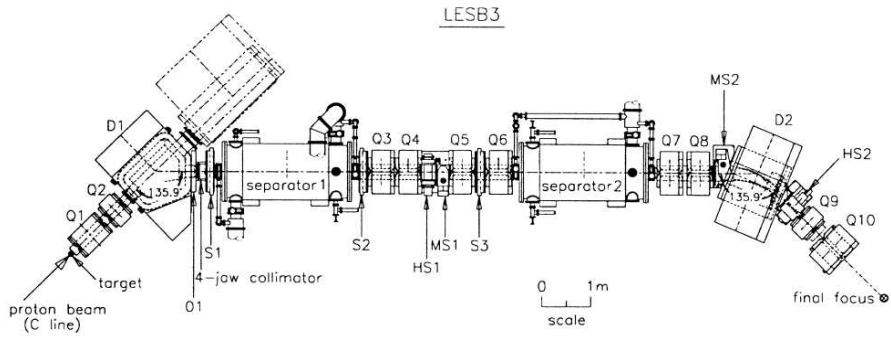


Figure 57: 800 MeV/c kaon beam line layout.

## zgoubi data file.

```

800 MeV/c KAON BEAM LINE. CALCULATION OF TRANSFER COEFFICIENTS.
'OBJET' 1
2668.5100 AUTOMATIC GENERATION OF
6 AN OBJECT FOR CALCULATION
OF THE FIRST ORDER TRANSFER
COEFFICIENTS WITH 'MATRIX'
.1 .1 .1 .1 0. .001
0. 0. 0. 0. 0. 1.
'PARTICUL' 2
493.646 1.60217733E-19 0. 0. 0. KAON M & Q, FOR USE IN WIEN FILTER
'DRIFT' 3
35.00000
'QUADRUPO' Q1 4
0
76.2 15.24 13.6
30. 30.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
30. 30.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
1.1
1 0. 0. 0.
'DRIFT' 5
25.00000
'QUADRUPO' Q2 6
0
45.72 15.24 -11.357
30. 30.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
30. 30.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
1.1
1 0. 0. 0.
'DRIFT' 7
-1.898
'TOSCA' 3-D MAP THE OF FIRST
0 0 BENDING MAGNET
1.0313E-3 1. 1. 1. B, X, Y, Z NORMALIZATION COEFFCNTS
1D map at z=0, from TOSCA
59 39 1 0
bw6_0.map
0 0. 0. 0.
2
1.1
1 0 0 0
'CHANGREF' 9
0. -70.78 -43.8
'FAISCEAU' 10
'DRIFT' 11
-49.38
'OCTUPOLE' 12
0
10. 15.24 .6
0. 0.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
0. 0.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
.4
1 0. 0. 0.
'SEXTUPOL' SX1, COMPENSATION
0 OF THE Theta.Phi ABERRATION
AT VF1
10. 15.24 2.4
0. 0. 0. 0.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
0. 0. 0. 0.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
.4
1 0. 0. 0.
'DRIFT' 14
50.0
'WIENFILT' FIRST VERTICAL WIEN FILTER 15
0
2.16 55.E5 -.0215576 2
20. 10. 10.
0.2401 1.8639 -0.5572 0.3904 0. 0.
0.2401 1.8639 -0.5572 0.3904 0. 0.
20. 10. 10.
0.2401 1.8639 -0.5572 0.3904 0. 0.
0.2401 1.8639 -0.5572 0.3904 0. 0.
1.
1. 0. 0. 0.
'DRIFT' 16
30.
'QUADRUPO' Q3 17
0
45.72 15.24 -6.34
30. 30.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
30. 30.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
1.1
1 0. 0. 0.
'DRIFT' 18
10.0
'MULTIPOL' SX2 + OCTU, COMPENSATION
0 OF THE D.Phi AND D2.Phi
ABERRATIONS AT VF1
10. 15.24 0. 0. -8. 1.2 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
.4
1 0. 0. 0.
'DRIFT' 20
90.0
'MATRIX' TRANSFER COEFFICIENTS 21
2 0
'COLLIMA' FIRST VERTICAL FOCUS,
2 MASS SLIT 22
2 14.6 .15E10 0. 0.
'DRIFT' 23
20.0
'QUADRUPO' Q5 24
0
45.72 15.24 10.93
30. 30.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
30. 30.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
1.1
1 0. 0. 0.
'DRIFT' 25
10.0
'MULTIPOL' COMPENSATION OF
0 ABERRATIONS AT VF2
10. 15.24 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
.4
1 0. 0. 0.
'DRIFT' 27
10.0
'QUADRUPO' Q6 28
0
45.72 15.24 -11.18
30. 30.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
30. 30.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
1.1
1 0. 0. 0.
'DRIFT' 29
50.0
'WIENFILT' SECOND VERTICAL WIEN FILTER 30
0
2.16 -55.E5 .0215576 2
20. 10. 10.
0.2401 1.8639 -0.5572 0.3904 0. 0.
0.2401 1.8639 -0.5572 0.3904 0. 0.
20. 10. 10.
0.2401 1.8639 -0.5572 0.3904 0. 0.
0.2401 1.8639 -0.5572 0.3904 0. 0.
1.
1. 0. 0. 0.
'DRIFT' 31
30.0
'QUADRUPO' Q7 32
0
45.72 15.24 -6.44
30. 30.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
30. 30.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
1.1
1 0. 0. 0.
'DRIFT' 33
25.00000
'QUADRUPO' Q8 34
0
45.72 15.24 8.085
30. 30.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
30. 30.
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
1.1
1 0. 0. 0.
'DRIFT' 35
40.0
'COLLIMA' SECOND VERTICAL FOCUS,
2 MASS SLIT 36
1 17. .2E10 0. 0.
'MATRIX' TRANSFER COEFFICIENTS 37
2 0
'DRIFT' 38
-25.0
'DIPOLE-M' SIMULATION OF THE MAP
3 OF THE SECOND BENDING MAGNET
(UPGRADED VERSION OF 'AIMANT') 39
150 60
18.999 0. 0. 0.
79.3329 17.7656 140.4480 110. 170.
15. -1.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.
0.00 21.90 1.E6 -1.E6 1.E6 1.E6
15. -1.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.
-43.80 -21.90 -1.E6 -1.E6 1.E6 -1.E6
0. 0.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.
-43.80 -21.90 -1.E6 -1.E6 1.E6 -1.E6 1E6
0
NO SHIMMING
2
SCD ORDER INTERPOLATION
2.5
2
147.48099 -0.31007 147.48099 0.31007
'DRIFT' 40
-15.00000
'QUADRUPO' Q9 41
0
35.56 12.7 -13.69 -13.91
30. 25.4
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
30. 25.4
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
.5
1 0. 0. 0.

```

```

'DRIFT'                                42
25.00000
'QUADRUPO'                               Q10 43
0
35.56 12.7 11.97
30. 25.4
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
30. 25.4
4 0.2490 5.3630 -2.4100 0.9870 0. 0.
1.1
1 0. 0. 0.
'DRIFT'                                44
200.0
'MATRIX'                               TRANSFER COEFFICIENTS 45
2 0 AT THE FINAL FOCUS
'END'                                    46

```

### Excerpt of zgoubi.res : first and second order transfer matrices and higher order coefficients at the end of the line.

```

FIRST ORDER COEFFICIENTS ( MKSA ) :
3.60453 -4.453265E-02 -3.049728E-04 -1.165832E-04 0.00000 -5.229783E-02
-2.05368 0.270335 4.700517E-05 1.763910E-05 0.00000 -9.561918E-02
2.240965E-05 -8.687757E-07 -3.60817 -1.731805E-02 0.00000 -7.815367E-02
1.185290E-05 -4.356398E-07 -2.05043 -0.286991 0.00000 -3.983392E-02
-0.387557 2.313953E-02 -2.264218E-05 -8.015244E-06 1.00000 0.374917
0.00000 0.00000 0.00000 0.00000 0.00000 1.00000
DetY-1 = -0.1170246601, DetZ-1 = 0.0000034613
R12=0 at 0.1647 m, R34=0 at -0.6034E-01 m
First order symplectic conditions (expected values = 0) :
-0.1170 3.4614E-06 -1.8207E-04 3.0973E-05 4.6007E-04 -8.0561E-05
SECOND ORDER COEFFICIENTS ( MKSA ) :
1 11 7.34 1 21 -1.78 1 31 1.399E-02 1 41 1.456E-02 1 51 0.00 1 61 36.3
1 12 -1.78 1 22 -530. 1 32 -1.308E-03 1 42 -1.743E-03 1 52 0.00 1 62 12.3
1 13 1.399E-02 1 23 -1.308E-03 1 33 -0.611 1 43 -0.522 1 53 0.00 1 63 -2.771E-02
1 14 1.456E-02 1 24 -1.743E-03 1 34 -0.522 1 44 0.163 1 54 0.00 1 64 -2.211E-02
1 15 0.00 1 25 0.00 1 35 0.00 1 45 0.00 1 55 0.00 1 65 0.00
1 16 36.3 1 26 12.3 1 36 -2.771E-02 1 46 -2.211E-02 1 56 0.00 1 66 2.88
2 11 -303. 2 21 3.81 2 31 3.684E-02 2 41 3.581E-02 2 51 0.00 2 61 144.
2 12 3.81 2 22 -62.9 2 32 -5.821E-04 2 42 -1.638E-04 2 52 0.00 2 62 -0.759
2 13 3.684E-02 2 23 -5.821E-04 2 33 1.05 2 43 1.94 2 53 0.00 2 63 -1.031E-02
2 14 3.581E-02 2 24 -1.638E-04 2 34 1.94 2 44 6.70 2 54 0.00 2 64 -4.285E-02
2 15 0.00 2 25 0.00 2 35 0.00 2 45 0.00 2 55 0.00 2 65 0.00
2 16 144. 2 26 -0.759 2 36 -1.031E-02 2 46 -4.285E-02 2 56 0.00 2 66 -65.3
3 11 -0.145 3 21 2.158E-02 3 31 20.6 3 41 86.0 3 51 0.00 3 61 -0.201
3 12 2.158E-02 3 22 64.6 3 32 1.61 3 42 0.496 3 52 0.00 3 62 8.793E-02
3 13 20.6 3 23 1.61 3 33 0.710 3 43 0.128 3 53 0.00 3 63 39.1
3 14 86.0 3 24 0.496 3 34 0.128 3 44 64.8 3 54 0.00 3 64 7.17
3 15 0.00 3 25 0.00 3 35 0.00 3 45 0.00 3 55 0.00 3 65 0.00
3 16 -0.201 3 26 8.793E-02 3 36 39.1 3 46 7.17 3 56 0.00 3 66 1.46
4 11 -8.254E-02 4 21 1.146E-02 4 31 10.7 4 41 47.3 4 51 0.00 4 61 -0.127
4 12 1.146E-02 4 22 33.0 4 32 0.787 4 42 0.157 4 52 0.00 4 62 3.566E-02
4 13 10.7 4 23 0.787 4 33 0.365 4 43 6.774E-02 4 53 0.00 4 63 17.5
4 14 47.3 4 24 0.157 4 34 6.774E-02 4 44 33.1 4 54 0.00 4 64 1.05
4 15 0.00 4 25 0.00 4 35 0.00 4 45 0.00 4 55 0.00 4 65 0.00
4 16 -0.127 4 26 3.566E-02 4 36 17.5 4 46 1.05 4 56 0.00 4 66 0.715
5 11 568. 5 21 -7.67 5 31 -5.970E-02 5 41 -5.682E-02 5 51 0.00 5 61 -251.
5 12 -7.67 5 22 225. 5 32 1.283E-03 5 42 6.947E-04 5 52 0.00 5 62 2.77
5 13 -5.970E-02 5 23 1.283E-03 5 33 19.2 5 43 10.2 5 53 0.00 5 63 0.215
5 14 -5.682E-02 5 24 6.947E-04 5 34 10.2 5 44 1.59 5 54 0.00 5 64 0.129
5 15 0.00 5 25 0.00 5 35 0.00 5 45 0.00 5 55 0.00 5 65 0.00
5 16 -251. 5 26 2.77 5 36 0.215 5 46 0.129 5 56 0.00 5 66 112.
HIGHER ORDER COEFFICIENTS ( MKSA ) :
Y/Y3 5784.8
Y/T3 9.40037E+05
Y/Z3 0.70673
Y/P3 0.42104
T/Y3 -18607.
T/T3 1.04607E+05
T/Z3 -0.10234
T/P3 5.25793E-02
Z/Y3 32.161
Z/T3 18.425
Z/Z3 -872.50
Z/P3 -785.20
P/Y3 15.460
P/T3 7.5264
P/Z3 -409.98
P/P3 -389.15

```



3 IN-FLIGHT DECAY IN SPES 3

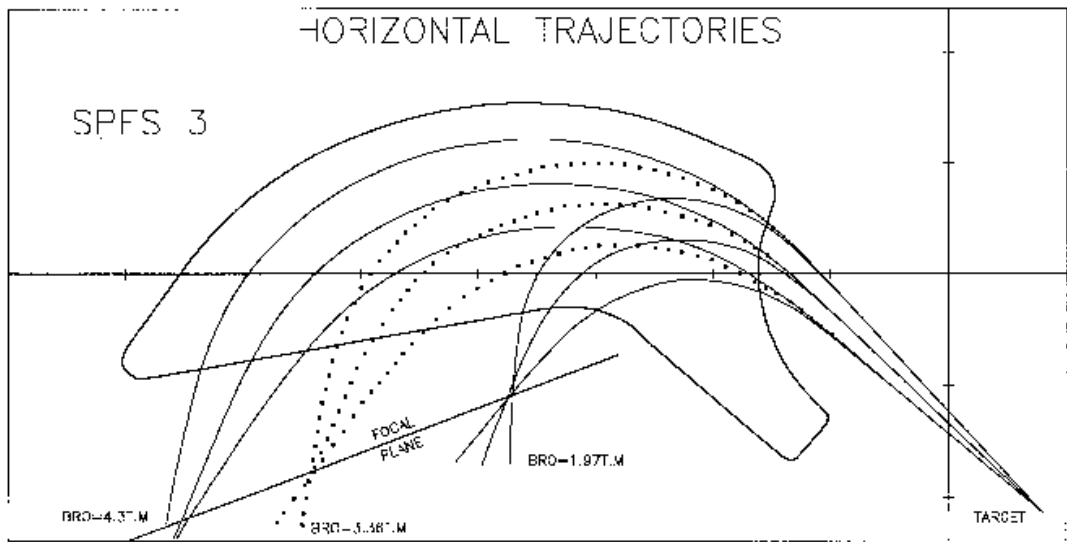


Figure 58: Layout of SPES 3 spectrometer.

## zgoubi data file

```

SIMULATION OF PION IN-FLIGHT DECAY IN SPES3 SPECTROMETER
'MCOBJET' 1
3360. REFERENCE RIGIDITY (PION).
1 DISTRIBUTION IN WINDOW.
200 BUNCHES OF 200 PARTICLES.
1 1 1 1 1 1 UNIFORM DISTRIBUTION
0. 0. 0. 0. 0. 1. CENTRAL VALUES OF BARS.
.5e-2 50.e-3 .5e-2 50.e-3 0. 0.4 WIDTH OF BARS.
1 1 1 1 1 1 CUT-OFFS (UNUSED)
9 9. 9. 9. 9. UNUSED.
186387 548728 472874 SEEDS.
'PARTICUL' 2
139.6000 0. 0. 26.03E-9 0. PION MASS AND LIFE TIME
'MCDESINT' 3
105.66 0. PION -> MUON + NEUTRINODECAY
136928 768370 548375
'ESL' 4
77.3627
'CHAMBR' 5
1 STOPS ABERRANT MUONS.
1 100. 10. 245. 0.
'DIPOLE' 6
2
80. 208.5 TOTAL ANGLE AT, CENTRAL RADIUS RM
33. 30. 0. 0. 0. CENTRAL ANGLE, FIELD, INDICES
46. -1. ENTRANCE EFB
4. .14552 5.21405 -3.38307 14.0629 0. 0. 0.
15. 0. -65. 0. 0. -65.
46. -1. EXIT EFB
4. .14552 5.21405 -3.38307 14.0629 0. 0. 0.
-15. 69. 85. 0. 1.E6 1.E6
0. 0. LATERAL EFB (INHIBITED)
4. .14552 5.21405 -3.38307 14.0629 0. 0. 0.
-15. 69. 85. 0. 1.E6 1.E6 1E6
2 10.0 2ND DEGREE INTERP., MESH 0.4 CM
4. STEP SIZE
2
164.755 .479966 233.554 -.057963
'CHAMBR' 7
2
1 100. 10. 245. 0.
'CHANGREF' 8
0. 0. -49. TILT ANGLE OF
FOCAL PLANE.
'HISTO' 9
TOTAL SPECTRUM (PION + MUON).
2 -170. 130. 60 1
20 'Y' 1 'Q'
'HISTO' 10
PION SPATIAL SPECTRUM
AT FOCAL PLANE.
2 -170. 130. 60 2
20 'P' 1 'P'
'HISTO' 11
MUON SPATIAL SPECTRUM
AT FOCAL PLANE.
2 -170. 130. 60 3
20 'y' 1 'S'
'HISTO' 12
MUON MOMENTUM SPECTRUM
AT FOCAL PLANE.
1 .2 1.7 60 3
20 'd' 1 'S'
'REBELOTE' 13
(49+1) RUNS = CALCULATION OF
49 0.1 0 (49+1)*200 TRAJECTORIES.
'END' 14

```

Excerpt of zgoubi.res : histograms of primary and secondary particles at focal surface of SPES3.

```
*****
9 HISTO TOTAL SPECTRUM
*****
HISTOGRAMME DE LA COORDONNEE Y
PARTICULES PRIMAIRES ET SECONDAIRES
DANS LA FENETRE : -1.7000E+02 / 1.3000E+02 (CM)
NORMALISE
*****
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
1234567890123456789012345678901234567890123456789012345678901
3 4 5 6 7 8
TOTAL COMPTAGE : 9887 SUR 10000
NUMERO DU CANAL MOYEN : 55
COMPTAGE AU " " : 281
VAL. PHYS. AU " " : 0.000E+00 (CM)
RESOLUTION PAR CANAL : 5.000E+00 (CM)
PARAMETRES PHYSIQUES DE LA DISTRIBUTION :
COMPTAGE = 9887 PARTICULES
MIN = -1.6687E+02, MAX = 9.4131E+01, MAX-MIN = 2.6100E+02 (CM)
MOYENNE = -9.2496E-01 (CM)
SIGMA = 5.3583E+01 (CM)
*****
```

```
*****
10 HISTO PION SPATIAL
*****
HISTOGRAMME DE LA COORDONNEE Y
PARTICULES PRIMAIRES
DANS LA FENETRE : -1.7000E+02 / 1.3000E+02 (CM)
NORMALISE
*****
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
1234567890123456789012345678901234567890123456789012345678901
3 4 5 6 7 8
TOTAL COMPTAGE : 9282 SUR 10000
NUMERO DU CANAL MOYEN : 55
COMPTAGE AU " " : 264
VAL. PHYS. AU " " : 0.000E+00 (CM)
RESOLUTION PAR CANAL : 5.000E+00 (CM)
PARAMETRES PHYSIQUES DE LA DISTRIBUTION :
COMPTAGE = 9282 PARTICULES
MIN = -9.5838E+01, MAX = 9.3504E+01, MAX-MIN = 1.8934E+02 (CM)
MOYENNE = 4.9971E-01 (CM)
SIGMA = 5.3215E+01 (CM)
*****
```

```
*****
11 HISTO MUON SPATIAL
*****
HISTOGRAMME DE LA COORDONNEE Y
PARTICULES SECONDAIRES
DANS LA FENETRE : -1.7000E+02 / 1.3000E+02 (CM)
NORMALISE
*****
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
1234567890123456789012345678901234567890123456789012345678901
3 4 5 6 7 8
TOTAL COMPTAGE : 605 SUR 10000
NUMERO DU CANAL MOYEN : 50
COMPTAGE AU " " : 14
VAL. PHYS. AU " " : -2.500E+01 (CM)
RESOLUTION PAR CANAL : 5.000E+00 (CM)
PARAMETRES PHYSIQUES DE LA DISTRIBUTION :
COMPTAGE = 605 PARTICULES
MIN = -1.6687E+02, MAX = 9.4131E+01, MAX-MIN = 2.6100E+02 (CM)
MOYENNE = -2.2782E+01 (CM)
SIGMA = 5.4452E+01 (CM)
*****
```

```
*****
12 HISTO MUON MOMENTUM
*****
HISTOGRAMME DE LA COORDONNEE D
PARTICULES SECONDAIRES
DANS LA FENETRE : 2.0000E-01 / 1.7000E+00
NORMALISE
*****
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
1234567890123456789012345678901234567890123456789012345678901
3 4 5 6 7
TOTAL COMPTAGE : 605 SUR 10000
NUMERO DU CANAL MOYEN : 46
COMPTAGE AU " " : 16
VAL. PHYS. AU " " : 8.250E-01
RESOLUTION PAR CANAL : 2.500E-02
PARAMETRES PHYSIQUES DE LA DISTRIBUTION :
COMPTAGE = 605 PARTICULES
MIN = 3.7184E-01, MAX = 1.3837E+00, MAX-MIN = 1.0119E+00
MOYENNE = 8.1693E-01
SIGMA = 2.2849E-01
*****
```

4 USE OF THE FITTING PROCEDURE

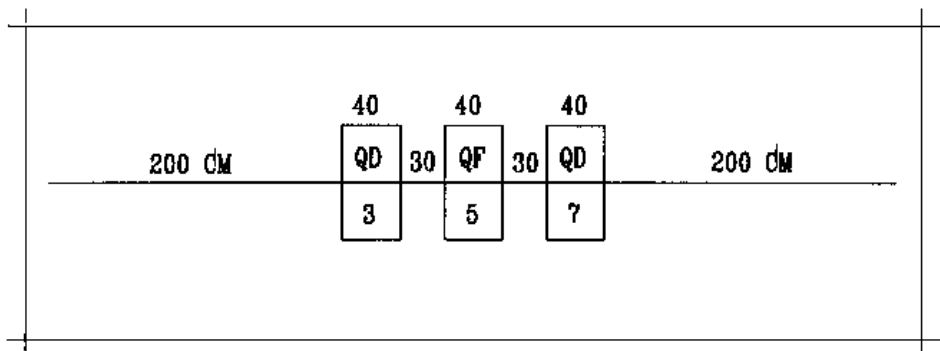


Figure 59: Vary B in all quadrupoles, for fitting of the transfer coefficients  $R_{12}$  and  $R_{34}$  at the end of the line. The first and last quadrupoles are coupled so as to present the same value of B.

zgoubi data file.

```

MATCHING A SYMMETRIC QUADRUPOLE TRIPLET
'OBJET'
2501.73          RIGIDITY (kG.cm)
5                11 PARTICLES GENERATED FOR USE OF MATRIX
2.  2.  2.  2.  0.  .001
0.  0.  0.  0.  0.  1.
'ESL'
200.
'QUADRUPO'      3
0
40.  15.  -7.
0.  0.
6  .1122  6.2671  -1.4982  3.5882  -2.1209  1.723
0.  0.
6  .1122  6.2671  -1.4982  3.5882  -2.1209  1.723
5.
1 0.  0.  0.
'ESL'
30.
'QUADRUPO'      5
0
40.  15.  3.
0.  0.
6  .1122  6.2671  -1.4982  3.5882  -2.1209  1.723
0.  0.
6  .1122  6.2671  -1.4982  3.5882  -2.1209  1.723
5.
1 0.  0.  0.
'ESL'
30.
'QUADRUPO'      7
0
40.  15.  -7.
0.  0.
6  .1122  6.2671  -1.4982  3.5882  -2.1209  1.723
0.  0.
6  .1122  6.2671  -1.4982  3.5882  -2.1209  1.723
5.
1 0.  0.  0.
'ESL'
200.
'MATRIX'
1 0
'FIT2'          VARY B IN QUADS FOR FIT OF R12 AND R34
2              # OF VARIABLES
3  12  7.12  2.  SYMMETRIC TRIPLET => QUADS #1 AND #3 ARE COUPLED
5  12  0.  2.  PRMTR #12 OF ELEMENTS #3, 5 AND 7 IS FIELD VALUE
2  1.E-10      # OF CONSTRAINTS, PENALTY
1  1  2  8  16.6  1.  CNSTRNT #1 : R12=16.6 AFTER LAST DRIFT (LMNT #8)
1  3  4  8  -.88  1.  CNSTRNT #2 : R34=-.88 AFTER LAST DRIFT
'END'
    
```

**Excerpt of zgoubi.res : first order transfer matrices prior  
to and after fitting.**

\*\*\*\*\*  
TRANSFER MATRIX WITH STARTING CONDITIONS :

MATRICE DE TRANSFERT ORDRE 1 ( MKSA )					
5.43427	17.0254	0.00000	0.00000	0.00000	0.00000
1.67580	5.43425	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	-1.27003	-0.974288	0.00000	0.00000
0.00000	0.00000	-0.629171	-1.27004	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	1.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	1.00000

\*\*\*\*\*

STATE OF VARIABLES AFTER MATCHING :

LMNT	VAR	PARAM	MINIMUM	INITIAL	FINAL	MAXIMUM	STEP	NAME	LBL1	LBL2
3	1	12	-21.0	-7.00	-6.972765137	7.00	1.707E-04	QUADRUPO	3	*
7	1	120	-6.97	-7.00	-6.972765137	7.00	1.707E-04			
5	2	12	-3.00	3.00	3.229344585	9.00	1.266E-04	QUADRUPO	5	*

STATUS OF CONSTRAINTS (Target penalty = 1.0000E-10)

TYPE	I	J	LMNT#	DESIRED	WEIGHT	REACHED	KI2	Parameter(s)
1	1	2	8	1.6600000E+01	1.0000E+00	1.6599981E+01	*	0 :
1	3	4	8	-8.8000000E-01	1.0000E+00	-8.8000964E-01	*	0 :

Fit reached penalty value 8.4374E-11

\*\*\*\*\*

MATRIX, WITH NEW VARIABLES :

5.27056	16.6000	0.00000	0.00000	0.00000	0.00000
1.61443	5.27450	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	-1.24205	-0.880010	0.00000	0.00000
0.00000	0.00000	-0.622553	-1.24620	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	1.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	1.00000

\*\*\*\*\*

## 5 MULTITURN SPIN TRACKING IN SATURNE 3 GeV SYNCHROTRON

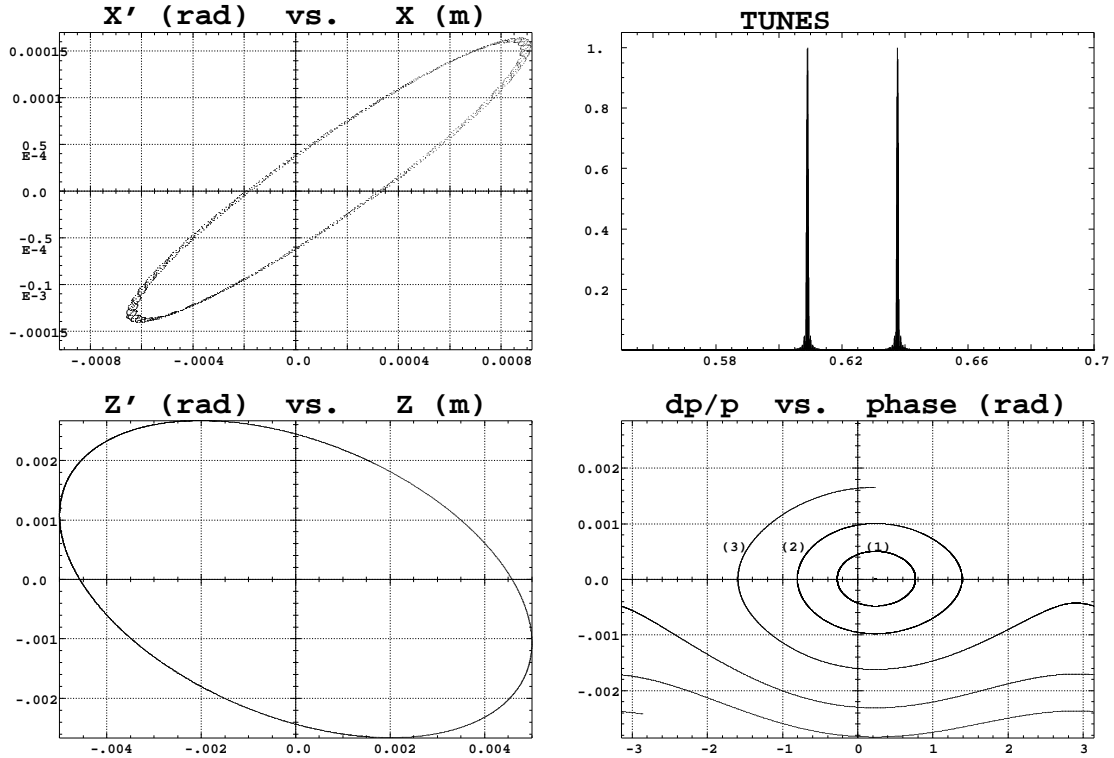


Figure 60: Tracking over 3000 turns. These simulations exhibit the first order parameters and motions as produced by the multi-turn ray-tracing.

- (A) Horizontal phase-space: the particle has been launched near to the closed orbit (the fine structure is due to  $Y - Z$  coupling induced by bends fringe fields, also responsible of the off-centering of the local closed orbit - at ellipse center).
- (B) Vertical phase-space: the particle has been launched with  $Z_0 = 4.58 \cdot 10^{-3}$  m,  $Z'_0 = 0$ . A least-square fit by  $\gamma_Z Z^2 + 2\alpha_Z Z Z' + \beta_Z Z'^2 = \varepsilon_Z / \pi$  yields  $\beta_Z = 2.055$  m,  $\alpha_Z = 0.444$ ,  $\gamma_Z = 0.582 \text{ m}^{-1}$ ,  $\varepsilon_Z / \pi = 12 \cdot 10^{-6}$  m.rad in agreement with matrix calculations.
- (C) Fractional tune numbers obtained by Fourier analysis for  $\varepsilon_Y / \pi = \varepsilon_Z / \pi \simeq 12 \cdot 10^{-6}$  m.rad:  $\nu_Y = 0.63795$ ,  $\nu_Z = 0.60912$  (the integer part is 3 for both).
- (D) Longitudinal phase-space (“(DP, phase)” in Zgoubi notations): particles with initial momentum dispersion of  $5 \cdot 10^{-4}$  (1),  $10^{-3}$  (2),  $1.65 \cdot 10^{-3}$  (3) (out of acceptance), are accelerated at 1405 eV/turn ( $\dot{B} = 2.1$  T/s); analytical calculations give accordingly momentum acceptance of  $1.65 \cdot 10^{-3}$ .

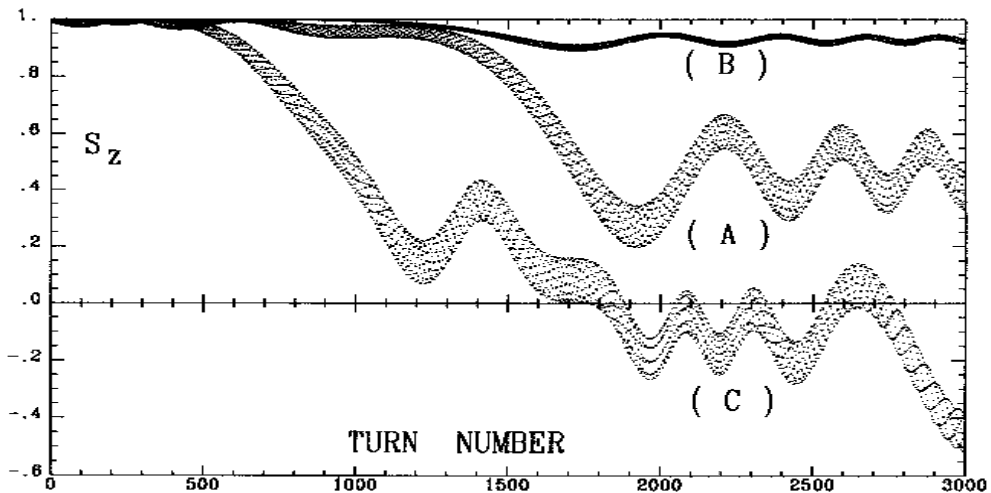


Figure 61: Crossing of  $\gamma G = 7 - \nu_Z$ , at  $\dot{B} = 2.1$  T/s.

- (A)  $\varepsilon_Z / \pi = 12.2 \cdot 10^{-6}$  m.rad. The strength of the resonance is  $|\varepsilon| = 3.3 \cdot 10^{-4}$ . As expected from the Froissart-Stora formula [82] the asymptotic polarization is about 0.44.
- (B) The emittance is now  $\varepsilon_Z / \pi = 1.2 \cdot 10^{-6}$  m.rad; comparison with (A) shows that  $|\varepsilon|$  is proportional to  $\sqrt{\varepsilon_Z}$ .
- (C) Crossing of this resonance for a particle having a momentum dispersion of  $10^{-3}$ .

zgoubi data file (beginning and end).

```

SATURNE. CROSSING GammaG=7-NUZ, NUZ=3.60877(perturbed)
'OBJET'
5015.388                      834.04 MeV, proton
2
  4 1
6.2E-02    6.5E-02    .458 0. 0. 1.00 'o'  EpsilonY/pi ~ 0. (Closed orbit)
0.356      0.379    .458 0. 0. 1.0005 '1'
0.647      0.689    .458 0. 0. 1.001   '2'
1.024      1.09     .458 0. 0. 1.0016 '3'
1 1 1 1
'SCALING'
1 4
MULTIPOL
2
5015.388E-3 5034.391E-3    CROSSING GammaG=7-Nuz+/-14E, E=3.3E-4
1 3442                      AT 2.1 T/s, IN 3442 MACHINE TURNS,
QUADRUPO                      FROM 834.041 TO 838.877 MeV
2
5015.388E-3 5034.391E-3
1 3442
BEND
2
5015.388E-3 5034.391E-3
1 3442
CAVITE
2
1. 1.00378894    RELATIVE CHANGE OF SYNCHRONOUS RIGIDITY
1 3442
'PARTICUL'
938.2723 1.6021892E-19 1.7928474 0. 0.
'SPNTRK'
3
'MARKER' #StartRing
'QUADRUPO'          QP 1          5
0
46.723 10. .763695          .763695 = FIELD FOR BOR0=1 T.m
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
#30|50|30 Quad
1 0. 0. 0.
'ESL'          SD 2          6
71.6256
'BEND'          DIP 3 4 3          7
0
247.30039 0. 1.57776
20. 8. .04276056667
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
20. 8. .04276056667 20. 8.
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
#30|120|30 bend 3 0. 0. 0. -.1963495408
'ESL'          SD 2          8
71.6256
'MULTIPOL'          QP 5          9
0
48.6273 10. 0. -.765533 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
#30|50|30 Quad
1 0. 0. 0.
'ESL'          SD 2          10
71.6256
'BEND'          DIP 3 4 3          11
0
247.30039 0. 1.57776
20. 8. .04276056667
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
20. 8. .04276056667 20. 8.
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
#30|120|30 bend 3 0. 0. 0. -.1963495408
'ESL'          SD 2          12
71.6256
'QUADRUPO'          QP 1          13
0
46.723 10. .763695
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
#30|50|30 Quad
1 0. 0. 0.
'ESL'          SD 2          14
71.6256
'BEND'          DIP 3 4 3          15
0
247.30039 0. 1.57776
20. 8. .04276056667
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
20. 8. .04276056667 20. 8.
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
#30|120|30 bend 3 0. 0. 0. -.1963495408
'ESL'          SD 2          16
71.6256
'MULTIPOL'          QP 5          17
0
48.6273 10. 0. -.765533 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
#30|50|30 Quad
1 0. 0. 0.

```

```

'ESL'          SD 2          18
71.6256
'BEND'          DIP 3 4 3          19
0
247.30039 0. 1.57776
20. 8. .04276056667
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
20. 8. .04276056667 20. 8.
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
#30|120|30 bend 3 0. 0. 0. -.1963495408
'ESL'          SD 2          20
71.6256
'QUADRUPO'          QP 1          21
0
46.723 10. .763695
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
#30|50|30 Quad
1 0. 0. 0.
'ESL'          SD 2          22
392.148
'MULTIPOL'          QP 5          23
0
48.6273 10. 0. -.765533 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
#30|50|30 Quad
1 0. 0. 0.
'ESL'          SD 2          24
392.148
'QUADRUPO'          QP 1          25
0
46.723 10. .763695
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
#30|50|30 Quad
1 0. 0. 0.
'ESL'          SD 2          26
71.6256
'BEND'          DIP 3 4 3          27
0
247.30039 0. 1.57776
20. 8. .04276056667
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
20. 8. .04276056667 20. 8.
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
#30|120|30 bend 3 0. 0. 0. -.1963495408
'ESL'          SD 2          28
71.6256
'MULTIPOL'          QP 5          29
0
48.6273 10. 0. -.765533 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
#30|50|30 Quad
1 0. 0. 0.
'ESL'          SD 2          30
71.6256
'BEND'          DIP 3 4 3          31
0
247.30039 0. 1.57776
20. 8. .04276056667
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
20. 8. .04276056667 20. 8.
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
#30|120|30 bend 3 0. 0. 0. -.1963495408
'ESL'          SD 2          32
392.148
'MARKER' #EndRing          86
'CAVITE'          87
1
105.5556848673 3.
6000. 0.          SIN(phis) = .234162, dE=1.40497 keV/Turn.
'FAISCNL'          88
b_zgoubi.fai
'SPNFRNL'          89
zgoubi.spn
'SPNFR'          90
'REBELOTE'          91
2999 0.1 99          TOTAL NUMBER OF TURNS = 3000
'END'          92

```

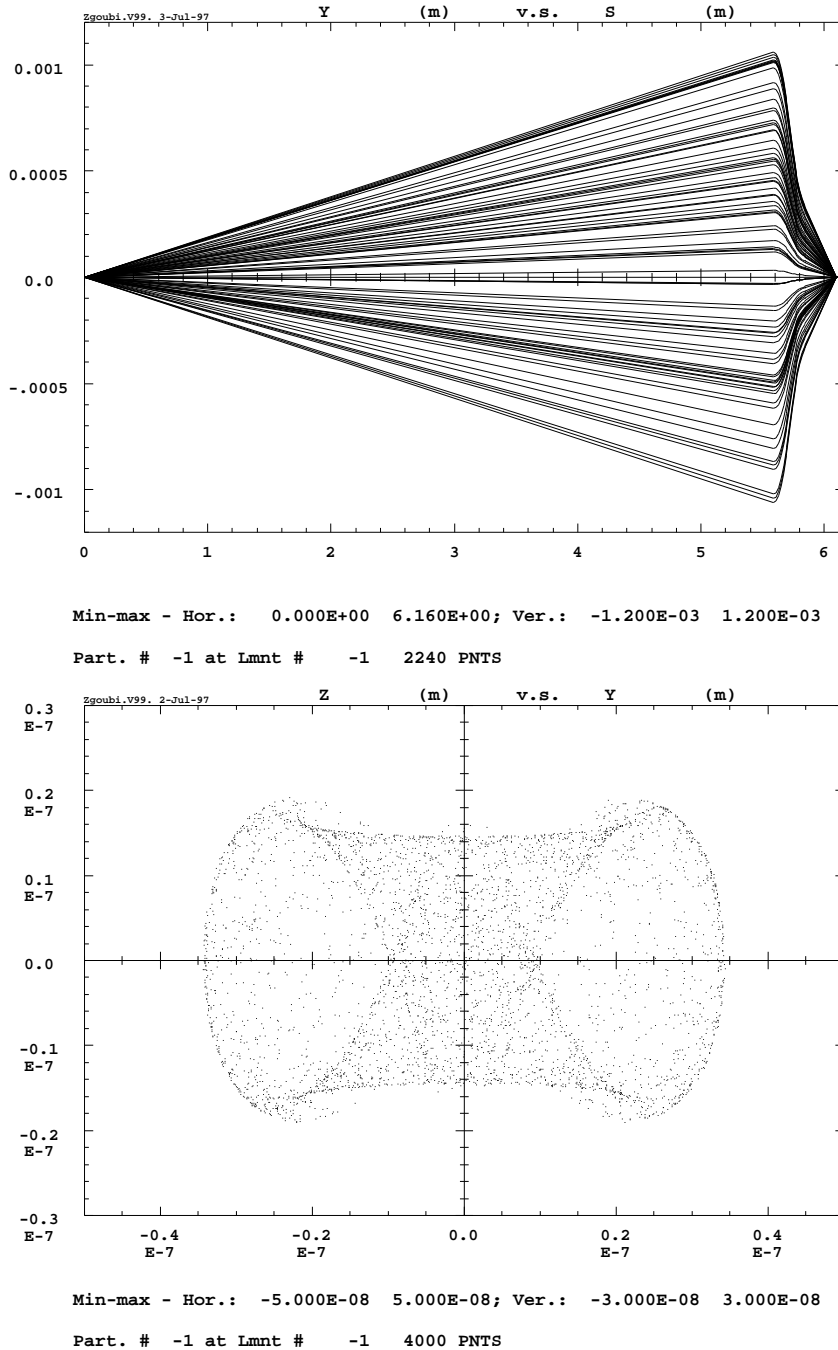
6 MICRO-BEAM FOCUSING WITH  $\vec{E} \times \vec{B}$  QUADRUPOLES

Figure 62: *Upper plot*: 50-particle beam tube ray-traced through a double focusing quadrupole doublet typical of the front end design of micro-beam lines. Initial conditions are :  $Y_0 = Z_0 = 0$ , angles  $T_0$  and  $P_0$  random uniform within  $\pm 0.2$  mrad, and momentum dispersion  $\delta p/p$  uniform in  $\pm 3 \cdot 10^{-4}$ .

*Lower plot*: (D) sub-micronic cross-section at the image plane of a 4000-particle beam with initial conditions as above, obtained thanks to the second-order achromatic magneto-electrostatic quadrupole doublet (the image size would be  $\Delta Y \approx \Delta Z \approx \pm 50 \mu m$  with regular magnetic quadrupoles, due to the momentum dispersion). Note the high resolution of the ray-tracing which still reveals image structure of nanometric size.



## zgoubi data file.

```

MICROBEAM LINE, WITH A MAGNETO-ELECTROSTATIC QUADRUPOLE DOUBLET.
'MCOBJET'          RANDOM OBJECT DEFINITION          1
20.435             RIGIDITY (20keV PROTONS).
1                 DISTRIBUTION IN WINDOW.
200               NUMBER OF PARTICLES.
1 1 1 1 1 1      UNIFORM DISTRIBUTION.
0. 0. 0. 0. 0. 1. CENTRAL VALUE, AND
0. .2e-3 0. .2e-3 0. 0.0003 HALF WIDTH OF DISTRIBUTION.
10. 10. 10. 10. 10. 10. CUT-OFFS (UNUSED).
9 9. 9. 9. 9.    FOR P(D) - UNUSED.
186387 548728 472874 SEEDS.
'PARTICUL'        PARTICLE MASS AND CHARGE          2
938.2723 1.60217733E-19 0. 0. 0. FOR INTEGRATION IN E-FIELD.
'DRIFT'           DRIFT.                             3
500.
'DRIFT'           DRIFT.                             4
59.
'EBMULT'          FIRST MAGNETO-ELECTROSTATIC        5
0                 QUADRUPOLE.
10.2 10. 0. -9272.986 0. 0. 0. 0. 0. 0. 0. 0. ELECTRIC Q-POLE COMPONENT.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. ENTRANCE EFB, SHARP EDGE.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. EXIT EFB, SHARP EDGE.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
10.2 10. 0. 1.89493 0. 0. 0. 0. 0. 0. 0. 0. MAGNETIC Q-POLE COMPONENT.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. ENTRANCE EFB, SHARP EDGE.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. EXIT EFB, SHARP EDGE.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
.8
1 0. 0. 0.
'DRIFT'           DRIFT.                             6
4.9
'EBMULT'          SECOND MAGNETO-ELECTROSTATIC        7
0                 QUADRUPOLE.
10.2 10. 0. 13779.90 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
10.2 10. 0. -2.81592 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
.8
1 0. 0. 0.
'DRIFT'           DRIFT.                             8
25.
'HISTO'           HISTOGRAM                          9
2 -5E-6 5E-6 60 2 OF THE Y COORDINATE.
20 'Y' 1 'Q'
'HISTO'           HISTOGRAM                          10
4 -5E-6 5E-6 60 2 OF THE Z COORDINATE.
20 'Z' 1 'Q'
'FAISCNL'        RAYS ARE STORED IN rays.out        11
rays.out         FOR FURTHER PLOTTING.
'REBELOTE'       RUN AGAIN, FOR RAY-TRACING          12
19 0.1 0         A TOTAL OF 200*(19+1) PARTICLES.
'END'            13

```

## zgoubi.res file.

```

*****
12 REBELOTE RUN AGAIN
*****
Multiple pass,
from element # 1 : MCOBJET /label1=RANDOM /label2=OBJECT to REBELOTE /label1=RANDOM /label2=OBJECT
ending at pass # 20 at element # 12 : REBELOTE /label1=RUN /label2=AGAIN

End of pass # 19 through the optical structure

Total of 3800 particles have been launched

*****
Next pass is # 20 and last pass through the optical structure
*****
1 MCOBJET RANDOM OBJECT
Reference magnetic rigidity = 20.435 KG*CM
Object built up of 200 particles
Distribution in a Window
Central values (MKGA units):
Yo, To, Zo, Fo, Xo, BR/BORO : 0.000 0.000 0.000 0.000 0.000 1.000
Width ( +/-, MKGA units ) :
DY, DT, DZ, DP, DX, DBR/BORO : 0.00 2.000E-04 0.00 2.000E-04 0.00 3.000E-04
Cut-offs ( * +/-Width ) :
NY, NT, NZ, NP, NX, NBR/BORO : 1.00 1.00 1.00 1.00 1.00 1.00
Type of sorting :
Y, T, Z, P, X, D : Uniform Uniform Uniform Uniform Uniform Uniform
*****
2 PARTICUL PARTICLE MASS
Particle properties :
Mass = 938.272 MeV/c2
Charge = 1.602177E-19 C
Reference data :
mag. rigidity (KG.cm) : 20.435000 =p/q, such that dev.=B*L/rigidity
mass (MeV/c2) : 938.27230
momentum (MeV/c) : 6.1262621
energy, total (MeV) : 938.29230
energy, kinetic (MeV) : 1.99998909E-02
beta = v/c : 6.5291616518E-03
gamma : 1.000021316
beta*gamma : 6.5293008252E-03
electric rigidity (MeV) : 3.9999376635E-02 =T[eV]*(gamma+1)/gamma, such that dev.=E*L/rigidity
*****

```

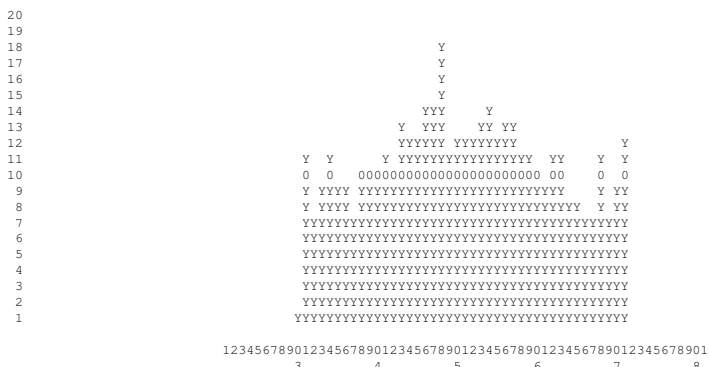
```

*****
3 DRIFT      DRIFT.      3
      Drift, length = 500.00000 cm
TRAJ #1 IEX,D,Y,T,Z,P,S,time : 1 2.999865E-04 -2.027052E-02 -4.054104E-02 -9.359114E-02 -1.871823E-01 5.0000001E+02 2.55365E+00
Cumulative length of optical axis = 5.000000000 m at ; Time (for ref. rigidity & particle) = 2.554417E-06 s
*****
4 DRIFT      DRIFT.      4
      Drift, length = 59.00000 cm
TRAJ #1 IEX,D,Y,T,Z,P,S,time : 1 2.999865E-04 -2.266244E-02 -4.054104E-02 -1.046349E-01 -1.871823E-01 5.5900001E+02 2.85498E+00
Cumulative length of optical axis = 5.590000000 m at ; Time (for ref. rigidity & particle) = 2.855839E-06 s
*****
5 EBMULT    FIRST      ELECTRO-MA
----- MULTIPOLE :
      Length of element = 10.200000 cm
      Bore radius      RO = 10.000 cm
      E-DIPOLE        = 0.0000000E+00 V/m
      E-QUADRUPOLE   = -9.2729860E+03 V/m
      E-SEXTUPOLE    = 0.0000000E+00 V/m
      E-OCTUPOLE     = 0.0000000E+00 V/m
      E-DECAPOLE     = 0.0000000E+00 V/m
      E-DODECAPOLE  = 0.0000000E+00 V/m
      E-14-POLE      = 0.0000000E+00 V/m
      E-16-POLE      = 0.0000000E+00 V/m
      E-18-POLE      = 0.0000000E+00 V/m
      E-20-POLE      = 0.0000000E+00 V/m
      Entrance/exit field models are sharp edge
      FINTE, FINTS, gap : 0.0000E+00 0.0000E+00 5.0000E+00
----- MULTIPOLE :
      Length of element = 10.200000 cm
      Bore radius      RO = 10.000 cm
      B-DIPOLE        = 0.0000000E+00 kG
      B-QUADRUPOLE   = 1.8949300E+00 kG
      B-SEXTUPOLE    = 0.0000000E+00 kG
      B-OCTUPOLE     = 0.0000000E+00 kG
      B-DECAPOLE     = 0.0000000E+00 kG
      B-DODECAPOLE  = 0.0000000E+00 kG
      B-14-POLE      = 0.0000000E+00 kG
      B-16-POLE      = 0.0000000E+00 kG
      B-18-POLE      = 0.0000000E+00 kG
      B-20-POLE      = 0.0000000E+00 kG
      Entrance/exit field models are sharp edge
      FINTE, FINTS, gap : 0.0000E+00 0.0000E+00 5.0000E+00
*** Warning : sharp edge model, vertical wedge focusing approximated with first order kick. FINT at entrance = 0.000
*** Warning : sharp edge model, vertical wedge focusing approximated with first order kick. FINT at exit = 0.000
      Integration step : 0.8000 cm
Cumulative length of optical axis = 5.692000000 m ; Time (for ref. rigidity & particle) = 2.907949E-06 s
*****
6 DRIFT      DRIFT.      6
      Drift, length = 4.90000 cm
TRAJ #1 IEX,D,Y,T,Z,P,S,time : 1 2.839285E-04 -1.310701E-02 9.564949E-01 -1.603604E-01 -5.590035E+00 5.7410014E+02 2.93210E+00
Cumulative length of optical axis = 5.741000000 m at ; Time (for ref. rigidity & particle) = 2.932982E-06 s
*****
7 EBMULT    SECOND     ELECTRO-MA
----- MULTIPOLE :
      Length of element = 10.200000 cm
      Bore radius      RO = 10.000 cm
      E-DIPOLE        = 0.0000000E+00 V/m
      E-QUADRUPOLE   = 1.3779900E+04 V/m
      E-SEXTUPOLE    = 0.0000000E+00 V/m
      E-OCTUPOLE     = 0.0000000E+00 V/m
      E-DECAPOLE     = 0.0000000E+00 V/m
      E-DODECAPOLE  = 0.0000000E+00 V/m
      E-14-POLE      = 0.0000000E+00 V/m
      E-16-POLE      = 0.0000000E+00 V/m
      E-18-POLE      = 0.0000000E+00 V/m
      E-20-POLE      = 0.0000000E+00 V/m
      Entrance/exit field models are sharp edge
      FINTE, FINTS, gap : 0.0000E+00 0.0000E+00 5.0000E+00
----- MULTIPOLE :
      Length of element = 10.200000 cm
      Bore radius      RO = 10.000 cm
      B-DIPOLE        = 0.0000000E+00 kG
      B-QUADRUPOLE   = -2.8159200E+00 kG
      B-SEXTUPOLE    = 0.0000000E+00 kG
      B-OCTUPOLE     = 0.0000000E+00 kG
      B-DECAPOLE     = 0.0000000E+00 kG
      B-DODECAPOLE  = 0.0000000E+00 kG
      B-14-POLE      = 0.0000000E+00 kG
      B-16-POLE      = 0.0000000E+00 kG
      B-18-POLE      = 0.0000000E+00 kG
      B-20-POLE      = 0.0000000E+00 kG
      Entrance/exit field models are sharp edge
      FINTE, FINTS, gap : 0.0000E+00 0.0000E+00 5.0000E+00
*** Warning : sharp edge model, vertical wedge focusing approximated with first order kick. FINT at entrance = 0.000
*** Warning : sharp edge model, vertical wedge focusing approximated with first order kick. FINT at exit = 0.000
      Integration step : 0.8000 cm
Cumulative length of optical axis = 5.843000000 m ; Time (for ref. rigidity & particle) = 2.985092E-06 s
*****

```

9 HISTO HISTOGRAM

HISTOGRAMME DE LA COORDONNEE Y  
 PARTICULES PRIMAIRES ET SECONDAIRES  
 DANS LA FENETRE : -5.0000E-06 / 5.0000E-06 (CM)  
 NORMALISE



TOTAL COMPTAGE : 4000 SUR 4000  
 NUMERO DU CANAL MOYEN : 51  
 COMPTAGE AU " " : 109  
 VAL. PHYS. AU " " : 0.000E+00 (CM)  
 RESOLUTION PAR CANAL : 1.667E-07 (CM)

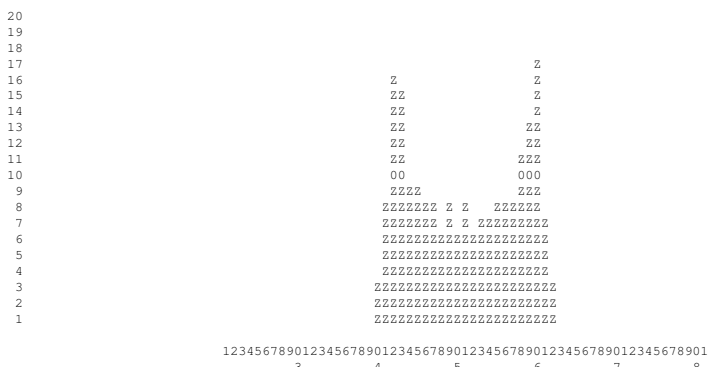
PARAMETRES PHYSIQUES DE LA DISTRIBUTION :  
 COMPTAGE = 4000 PARTICULES  
 MIN = -3.4326E-06, MAX = 3.4347E-06, MAX-MIN = 6.8674E-06 (CM)  
 MOYENNE = -2.8531E-08 (CM)  
 SIGMA = 1.8619E-06 (CM)

TRAJ #1 D,Y,T,Z,P,S,IEX : 1.0002E+00 9.0257E-07 -2.3996E-01 -1.0770E-06 1.7947E+00 6.09300E+02 1

\*\*\*\*\*

10 HISTO HISTOGRAM

HISTOGRAMME DE LA COORDONNEE Z  
 PARTICULES PRIMAIRES ET SECONDAIRES  
 DANS LA FENETRE : -5.0000E-06 / 5.0000E-06 (CM)  
 NORMALISE



TOTAL COMPTAGE : 4000 SUR 4000  
 NUMERO DU CANAL MOYEN : 51  
 COMPTAGE AU " " : 169  
 VAL. PHYS. AU " " : 0.000E+00 (CM)  
 RESOLUTION PAR CANAL : 1.667E-07 (CM)

PARAMETRES PHYSIQUES DE LA DISTRIBUTION :  
 COMPTAGE = 4000 PARTICULES  
 MIN = -1.9150E-06, MAX = 1.9110E-06, MAX-MIN = 3.8260E-06 (CM)  
 MOYENNE = -3.8539E-09 (CM)  
 SIGMA = 1.1232E-06 (CM)

TRAJ #1 D,Y,T,Z,P,S,IEX : 1.0002E+00 9.0257E-07 -2.3996E-01 -1.0770E-06 1.7947E+00 6.09300E+02 1

\*\*\*\*\*

11 FAISCNL RAYS ARE

Print[s] occur at

\*\*\*\*\*

12 REBELOTE RUN AGAIN,

\*\*\*\* FIN D'EFFET DE 'REBELOTE' \*\*\*\*  
 IL Y A EU 20 PASSAGES DANS LA STRUCTURE  
 # PARTICULES ENVOYEEES : 4000

\*\*\*\*\*

PGM PRINCIPAL : ARRET SUR CLE REBELOTE

\*\*\*\*\*



## **PART D**

**Running zgoubi and  
its post-processor/graphic interface zpop**



## INTRODUCTION

The **zgoubi** package, including this guide, examples, the **zpop** graphic/analysis post-processor, is available on web [8]. The **zgoubi FORTRAN** package is transportable; it has been compiled, linked and executed over the years on several types of systems (e.g. CDC, CRAY, IBM, DEC, HP, SUN, VAX, UNIX, LINUX).

An additional *FORTRAN* code, **zpop**, allows the post-processing and graphic treatment of **zgoubi** output files. **zpop** has been routinely used on DEC, HP, SUN stations, and is now on UNIX and LINUX systems.

### 1 GETTING TO RUN zgoubi AND zpop

#### 1.1 Making the Executable Files zgoubi and zpop

##### 1.1.1 The transportable package zgoubi

Compile and link the *FORTRAN* source files (normally, just run the Makefile), to create the executable **zgoubi**.

**zgoubi** is written in standard *FORTRAN*, mostly 77, therefore it does not require linking to any library.

##### 1.1.2 The post-processor and graphic interface package zpop

Compile the *FORTRAN* source files (normally, just run the Makefile).

**zpop** links to the graphic library, libminigraf.a [83]. This creates the executable **zpop**.

**zpop** is to be run on an xterm window.

#### 1.2 Running zgoubi

The principles are the following:

- Fill up **zgoubi.dat** with the input data that describe the problem (see examples, Part C).
- Run **zgoubi** - no argument needed, by default the input file is assumed to be **zgoubi.dat**. Specific arguments allow specifying a different input data file name, see below.
- Results of the execution will be printed by default into **zgoubi.res** and, upon options as appearing in **zgoubi.dat**, into various additional output files, for instance **zgoubi.fai**, **zgoubi.plt** (see section 2 below).
- The command **zgoubi** accepts the following arguments (see **zgoubi\_main.f** source file) :
  - in or -fileIn, followed by an the input file name,
  - out or -fileOut, followed by an output file name,
  - fileLog, followed by a log file name,
  - rank, followed by an integer number, this causes the seed in *MCOBJET* to be overridden, replaced by reading the current time instead,
  - saveExec, this will cause saving a copy of **zgoubi** exec in the local directory,
  - saveZpop, this will cause saving a copy of **zpop** exec in the local directory.

#### 1.3 Running zpop

Run **zpop** on an xterm window. This will open a graphic window. And then (Fig. ??):

- To access the data plotting sub-menu, select option 7.
- To access the data treatment sub-menu, select option 8.
- An on-line Help provides some information regarding the available post-processing procedures (Fourier transform, elliptical fit, synchrotron radiation spectral brightness, field map contours, tune diagram, etc.).

## 2 zgoubi OUTPUT FILES

When explicitly requested by means of the appropriate keywords, options, or dedicated *LABEL*'s, extra storage files are opened by **zgoubi** (*FORTRAN* "OPEN" statement) and filled up during the execution.

**Main menu:**

```

*****
MENU :

Working directory now :
[...]                               0000/home/fmeot

 7 Menu for direct plots from
   .plt, .fai, .spn type zgoubi output files

 8 Menu for other Analysis/Plot on
   .plt, .fai, .spn type zgoubi output files

10 Exit zpop

*****

* Option number :

```

**Sub-menu 7:**

```

*****
MENU - Graphic processing :

Working directory now :
[...]                               0000/home/fmeot

 1 OPEN FILE - current is none
 2 VARIABLES TO PLOT
   PRESENT: H-AXIS : Y (KX= 2)
            V-AXIS : Y' (KY= 3)

 3 PLOT OPTIONS
 4 MANUAL SCALES
 5 AUTOMATIC SCALES
 7 ** PLOT Y V.S. X **
 8 Save graphic screen
 9 EXIT THIS MENU
12 ERASE DISPLAY
15 KILL/CREATE GRAPHIC SUB-PROCESS
20 Superpose a curve
77 ** PLOT Y V.S. X, turn by turn **
88 HELP

*****

** No input data file opened, yet :
   go through Menu, option      1

* Option number :

```

**Sub-menu 8:**

```

*****
MENU - Analysis/Graphic :

Working directory now :
[...]                               0000/home/fmeot

 1 Open file - current is none
 3 Plot options
 4 Manual scales
 6 Synoptic
 8 Save graphic screen
 9 Exit this menu
10 SR loss parameters
11 Isofield lines from a map
12 Erase display
13 Fourier spectrum - multiturn
14 Ellipse fit - multiturn or multiparticle
15 Kill/create graphic sub-process
16 Synchrotron radiation distributions
17 Pick-up signals
18 Fringe field matching
19 Transverse beam density matching
20 Superpose a curve
21 Tune diagram
88 HELP

*****

* Option number :

```

Figure 63: Zpop main menu and sub-menus



Their content can be post-processed using the interactive program **zpop** and its dedicated graphic and analysis procedures.

Most of these files are formatted in columns, plotting their content using 'gnuplot' [87] is straightforward. A library of 'gnuplot' command files can be found in **zgoubi** sourceForge repository [8].

**zgoubi** procedures that create and fill these extra output files are the following (refer to Part A and Part B of the guide):

- Keywords *FAISCNL*, *FAISTORE*: fill a '.fai' type file (normally named [b\_]zgoubi.fai) with particle and spin coordinates and a lot of other informations.
- Keywords *SPNPRNL*, *SPNSTORE*: fill a '.spn' type file (normally named [b\_]zgoubi.spn) with spin coordinates and other informations.
- Option  $IC = 2$ , with field map keywords (e.g. *CARTEMES*, *TOSCA*) : fill zgoubi.map with 2-D field map.
- Option  $IL = 2$ , with magnetic and electric element keywords: fill zgoubi.plt with the particle coordinates and experienced fields, step after step, all along the optical element.
- Using the keyword *MARKER* with '.plt' as a second *LABEL* will cause storage of current coordinates into zgoubi.plt.
- A "PRINT" instruction may appear as an argument in many keywords, for instance *CAVITE*, *MATRIX*, ... This results in an output file with name of the form "zgoubi.thisKeyword.out", sometimes "zgoubi.thisKeyword.Out", for instance:  
*zgoubi.CAVITE.Out* (keyword *CAVITE*), *zgoubi.MATRIX.out* (keyword *MATRIX*), *zgoubi.OPTICS.out* (keyword *OPTICS*), *zgoubi.PICKUP.out* (keyword *PICKUPS*), *zgoubi.TWISS.out* (keyword *TWISS*), etc.

The purpose of this is mostly twofold: allow easy plotting as the data are listed under the form of columns (using gnuplot for instance), and a help in debugging. It is in addition easy to add variables to the Fortran write statement of concern, for more computation information output to *zgoubi.thisKeyword.out*. Lookup "PRINT" in the Index, for details.

Typical examples of graphics that one can expect from the post-processing of these files by **zpop** are the following (see examples, Part C):

- '.fai' type files:  
 possible plots include phase-space plots (transverse and longitudinal), aberration curves, at the position where *FAISTORE* or *FAISCNL* appears in the optical structure. Histograms of coordinates. Fourier analysis (e.g. , spin motion, tunes, in case of multi-turn tracking), calculation of Twiss parameters from phase-space ellipse matching.
- zgoubi.map  
 possible plots include isomagnetic field lines of a 2-D map. Superimposing trajectories read from zgoubi.plt is possible.
- zgoubi.plt  
 possible plots include trajectories inside magnets and other lenses (these can be superimposed over field lines obtained from zgoubi.map). Fields experienced by the particles at the traversal of optical elements. Spectral-angular distribution of synchrotron radiation.
- zgoubi.spn  
 possible plots include spin coordinates and histograms, at the position where *SPNPRNL* appears in the structure. Resonance crossing when performing multi-turn tracking.



## References

- [1] . Suadinos, J.: Spectrométrie magnétique. Ecole d'été des Houches, Méthodes expérimentales en physique nucléaire, 2-14 septembre 1974.
- [2] Jacques Thirion, Preface, in "The 20 years of the synchrotron Saturne-2". A. Boudard, P.-A. Chamouard Eds. World Scientific, May 1998
- [3] F. Méot, , *Spin Dynamics*, Summer 2021 USPAS Spin Class. <https://uspas.fnal.gov/materials/21onlineSBU/onlineSBU-Spin.shtml>.
- [4] F. Méot et S. Valéro, *Manuel d'utilisation de Zgoubi*, Rapport IRF/LNS/88-13, CEA Saclay, 1988.
- [5] F. Méot and S. Valéro, *Zgoubi users' guide*, Int. Rep. CEA/DSM/LNS/GT/90-05, CEA Saclay (1990) & TRIUMF report TRI/CD/90-02 (1990).
- [6] F. Méot and S. Valéro, *Zgoubi users' guide - Version 3*, Int. Rep. DSM/LNS/GT/93-12, CEA Saclay (1993).
- [7] F. Méot and S. Valero, *Zgoubi users' guide - Version 4*, FNAL Tech. Rep. FERMILAB-TM-2010 (Aug. 1997), & Int. Rep. CEA DSM DAPNIA/SEA-97-13, Saclay (Oct. 1997).
- [8] <http://zgoubi.sourceforge.net/>
- [9] F. Méot, *The electrification of Zgoubi*, SATURNE report DSM/LNS/GT/93-09, CEA Saclay (1993) ;  
F. Méot, *Generalization of the Zgoubi method for ray-tracing to include electric fields*, NIM A 340 (1994) 594-604.
- [10] D. Carvounas, *Suivi numérique de particules chargées dans un solénoïde*, rapport de stage, CEA/LNS/GT, 1991.
- [11] F. Méot, *Raytracing in 3-D field maps with Zgoubi*, report DSM/LNS/GT/90-01, CEA Saclay, 1990.
- [12] G. Leleux, *Compléments sur la physique des accélérateurs*, cours du DEA Grands-Instruments, Univ. Paris-VI, report IRF/LNS/86-101, CEA Saclay, March 1986.
- [13] F. Méot, *A numerical method for combined spin tracking and raytracing of charged particles*, NIM A313 (1992) 492, and proc. EPAC (1992) p.747.
- [14] D. J. Kelliher et al., *Muon decay ring study*, Procs. EPAC08 Conf., Genoa, Italy (2008).
- [15] F. Méot, N. Monseu, *Lattice Design and Study Tools Regarding the Super-B Project*, Procs. IPAC10 Conf., Kyoto, Japan (2010).
- [16] F. Méot, *Spin tracking simulations in AGS based on ray-tracing methods*, Tech. Note C-AD/AP/452, BNL C-AD (2009) ;  
F. Méot, *Zgoubi-ing AGS : spin motion with snakes and jump-quads*, Tech. Note C-AD/AP/453, BNL C-AD (2009).
- [17] F. Méot, M. Bai, V. Ptitsyn, V. Ranjbar, *Spin Code Benchmarking at RHIC*, Procs. PAC11 Conf., New York (2011).
- [18] F. Méot, *Raytracing Based Spin Tracking*, EDM Searches at Storage Rings Workshop, ECT - Center for Studies in Nuclear Physics and related Areas, Trento, Italy (Oct. 1-5, 2012).
- [19] Hock, K.: Imperfection resonance crossing in the AGS Booster. BNL-219848-2020-TECH, C-A/AP/633, BNL (2020).  
<https://technotes.bnl.gov/PDF?publicationId=219848>
- [20] Méot, F.: Polarized e-bunch acceleration at Cornell RCS: Tentative trackingsimulations. BNL-114452-2017-TECH, EIC/57 (2017).  
<https://technotes.bnl.gov/PDF?publicationId=42654>
- [21] Méot, F., Ranjbar, V.: Simulation of Polarized Electron Bunch Acceleration in EIC RCS. Bunch and Spin Densities at Extraction. EIC-ADD-TN-007, BNL (Sept. 2020)
- [22] Méot, F., Wang, E.: Spin simulations in eRHIC Wien filter. BNL-212123-2019-TECH, EIC/68, BNL (2019).  
<https://technotes.bnl.gov/PDF?publicationId=212123>
- [23] F. Méot, *An ergodic approach to polarization in eRHIC electron storage ring*, EIC Tech Note 59, BNL, Oct. 2018;  
<https://technotes.bnl.gov/PDF?publicationId=209351>
- [24] V. Bargmann, L. Michel, V.L. Telegdi, *Precession of the polarization of particles moving in a homogeneous electromagnetic field*, Phys. Rev. Lett. 2 (1959) 435.
- [25] S.Y. Lee, *Spin dynamics and snakes in synchrotrons*, World Scientific, 1997.
- [26] S.Y. Lee, private communication, April 2017.
- [27] F. Méot and J. Payet, *Numerical tools for the simulation of synchrotron radiation loss and induced dynamical effects in high energy transport lines*, Report DSM/DAPNIA/SEA-00-01, CEA Saclay (2000).

- [28] G. Leleux et al., *SR perturbations in long transport lines*, IEEE 1991 Part Acc. Conf., San Francisco (May 1991).
- [29] P. Lapostolle, F. Méot, S. Valero, *A new dynamics code DYNAC for electrons, protons and heavy ions in LINACS with long accelerating elements*, 1990 LINAC Conf., Albuquerque, NM, USA.
- [30] *Electron Lab for Europe*, Blue Book, CNRS-IN2P3 (1994).
- [31] F. Méot, *Benchmarking stepwise ray-tracing in rings in presence of radiation damping*, Procs. PAC11 Conf., New York (2011).
- [32] F. Méot, *Simulation of radiation damping in rings, using stepwise ray-tracing methods*, 2015 JINST 10 T06006.
- [33] V. O. Kostroun, *Simple numerical evaluation of modified Bessel functions and integrals [...]*, NIM 172 (1980) 371-374.
- [34] F. Méot, *Synchrotron radiation interferences at the LEP miniwiggler*, Report CERN SL/94-22 (AP), 1994.
- [35] L. Ponce, R. Jung, F. Méot, *LHC proton beam diagnostics using synchrotron radiation*, Yellow Report CERN-2004-007.
- [36] F. Méot, *A theory of low frequency far-field synchrotron radiation*, Particle Accelerators Vol 62, pp. 215-239 (1999).
- [37] Albert Hofmann, *The Physics of Synchrotron Radiation*, Cambridge University Press, May 13, 2004.
- [38] Malek Haj Tahar, *Fixed Field Ring Accelerators and ADS-Reactor Application*, PhD dissertation, Brookhaven National Laboratory and Grenoble-Alpes University (2016).  
<https://www.osti.gov/biblio/1351800>
- [39] M. Haj Tahar, "FFAG14 workshop tutorial", September 2014, <https://www.bnl.gov/ffag14/tutorial/>
- [40] Space charge studies in FFAG using the tracking code Zgoubi; Tahar, M. Haj ; Méot, F. ; Tsoupas, N., 6th International Particle Accelerator Conference (IPAC15), Richmond, VA (United States), 3-8 May 2015,  
<https://www.osti.gov/biblio/1183868-space-charge-studies-ffag-using-tracking-code-zgoubi>.
- [41] B. Mayer, personal communication, CEA Saclay, Laboratoire National SATURNE, 1990.
- [42] L. Farvacque et al., *Beta user's guide*, Note ESRF-COMP-87-01, 1987 ;  
J. Payet, IRF/LNS, CEA Saclay, private communication ;  
J.M. Lagniel, *Recherche d'un optimum*, Note IRF/LNS/SM 87/48, CEA Saclay 1987.
- [43] Installed by J. S. Berg, BNL (2007). Cf. *Detection and remediation of stagnation in the Nelder-Mead algorithm using a sufficient decrease condition*, C. T. Kelley, Siam J. Optim., Vol. 10, No. 1, pp. 43-55.
- [44] F. Desforges, *Implementation of a coupled treatment of the one-turn mapping in the ray-tracing code zgoubi*, C-AD Note C-A/AP/461, Brookhaven National Laboratory, Sept 2012.
- [45] F. Méot and N. Willis, *Raytrace computation with Monte Carlo simulation of particle decay*, internal report CEA/LNS/88-18 CEA Saclay, 1988.
- [46] E.J. Bleser, *The parameters of the bare AGS*, Tech. Note AGS/AD?Tech. Note 430, March 15, 1996.
- [47] These transfer functions have been copied from the MADX model of the AGS.
- [48] H.A. Enge, *Deflecting magnets*, in *Focusing of Charged Particles*, ed. A. Septier, Vol. II, pp 203-264, Academic Press Inc., 1967.
- [49] P. Birien et S. Valéro, *Projet de spectromètre magnétique à haute résolution pour ions lourds*, Section IV p.62, Note CEA-N-2215, CEA Saclay, mai 1981.
- [50] Files developed by Simon White, January 2012. Including beam-beam spin kick after Ref. [51].
- [51] Y. K. Batygin, *Spin depolarization due to beam-beam collisions*, Phys. Rev. E, Vol. 58, 1, July 1998.
- [52] A. Hofmann, S. Myers, *Beam Dynamics in a Double RF System*, CERN ISR-TH-RF/80-26 (July 1980).

- [53] F. Lemuët, F. Méot, *Developements in the ray-tracing code Zgoubi for 6-D multiturn tracking in FFAG rings*, NIM A 547 (2005) 638-651.
- [54] F. Méot, *6-D beam dynamics simulations in FFAGs using the ray-tracing code Zgoubi*, ICFA Beam Dyn. Newslett. 43:44-50 (2007).
- [55] Franck Lemuët, *Collection and muon acceleration in the neutrino factory project*, PhD thesis, Paris KI University, April 2007.
- [56] V. M. Kel'man and S. Ya. Yavor, *Achromatic quadrupole electron lenses*, Soviet Physics - Technical Physics, vol. 6, No 12, June 1962 ;  
S. Ya. Yavor et al., *Achromatic quadrupole lenses*, NIM 26 (1964) 13-17.
- [57] A. Septier, *Cours du DEA de physique des particules, optique corpusculaire*, Université d'Orsay, 1966-67, pp. 38-39.
- [58] S. P. Karetskaya et al., *Mirror-bank energy analyzers*, in *Advances in electronics and electron physics*, Vol. 89, Acad. Press (1994) 391-491.
- [59] A. Septier et J. van Acker, *Les lentilles quadrupolaires électriques*, NIM 13 (1961) 335-355 ;  
Y. Fujita and H. Matsuda, *Third order transfer matrices for an electrostatic quadrupole lens*, NIM 123 (1975) 495-504.
- [60] J. Fourier, F. Martinache, F. Méot, J. Pasternak, *Spiral FFAG lattice design tools, application to 6-D tracking in a proton-therapy class lattice*, NIM A 589 (2008).
- [61] F. Méot, *Tracking studies regarding EMMA FFAG project*, Internal report CEA DAPNIA-06-04 (2006).
- [62] J. S. Berg et al., *Recent developments on the EMMA on-line commissioning software*, Procs. IPAC10 Conf., Kyoto, Japan (2010).
- [63] S. Machida et al., *Acceleration in the linear non-scaling fixed-field alternating-gradient accelerator EMMA*, Nature Physics, vol. 8, March 2012.
- [64] Installed by Pavel Akishin, JINR, Dubna, 1992.
- [65] K. Halbach, "Design of permanent multipole magnets with oriented rare earth cobalt material", Nucl. Instrum. Meth. 169 (1980) pp. 1-10.
- [66] John P. Blewett and R. Chasman: Orbits and fields in the helical wiggler. Journal of Applied Physics 48, 2692 (1977); <https://doi.org/10.1063/1.324119>. Published Online: 26 August 2008. <https://aip.scitation.org/doi/pdf/10.1063/1.324119>
- [67] N. Tsoupas et als., "The beam optics of the FFAG cell of the CBETA ERL accelerator", MOPIK122, Proceedings of IPAC2017, Copenhagen, Denmark.
- [68] S.M. Lund and K. Halbach, "Iron free permanent magnet systems for charged particle beam optics", UCRL-JC-121454 (1995)
- [69] M.W. Garrett, *Calculation of fields [...] by elliptic integrals*, J. Appl. Phys., 34, 9, sept. 1963.
- [70] P.F. Byrd and M.D. Friedman, *Handbook of elliptic integrals for engineers and scientists*, pp. 282-283, Springer-Verlag, Berlin, 1954.
- [71] Installation by M. Bai, BNL, 2009.
- [72] F. Méot, et als.: Re-visiting RHIC snakes: OPERA fields,  $\vec{n}_0$  dance. Int. Rep. C-A/AP/590 (2017). <https://technotes.bnl.gov/PDF?publicationId=42159>
- [73] A. Tkatchenko, *Computer program UNIPOT*, SATURNE, CEA Saclay, 1982.
- [74] S.B. Kowalski, H.A. Enge, The ion-optical program raytrace, NIM A258 Vol. 3 (1987) 407.
- [75] N. Tsoupas, private communication, Brookhaven National Laboratory, Oct 2015.
- [76] J.L. Chuma, *PLOTDATA*, TRIUMF Design Note TRI-CO-87-03a.
- [77] F. Méot et al., A model of the AGS in the ray-tracing code Zgoubi, Tech. Note C-A/AP/\*\*\*\*, 2013.
- [78] J. Thirion et P. Birien, *Le spectromètre II*, Internal Report DPh-N/ME, CEA Saclay, 23 Déc. 1975 ;  
H. Catz, *Le spectromètre SPES II*, Internal Report DPh-N/ME, CEA Saclay, 1980 ;  
A. Moalem, F. Méot, G. Leleux, J.P. Penicaud, A. Tkatchenko, P. Birien, *A modified QDD spectrometer for  $\eta$  meson decay measurements*, NIM A289 (1990) 168-175

- [79] P. Pile, I-H. Chiang, K. K. Li, C. J. Kost, J. Doornbos, F. Méot et al., *A two-stage separated 800-MeV/c Kaon beamline*, TRIUMF and BNL Preprint (1997).
- [80] F. Méot, *The raytracing code Zgoubi*, CERN SL/94-82 (AP) (1994), 3rd Intern. Workshop on Optimization and Inverse Problems in Electromagnetism, CERN, Geneva, Switzerland, 19-21 Sept. 1994.
- [81] E. Grorud, J.L. Laclare, G. Leleux, *Résonances de dépolarisation dans SATURNE 2*, Int. report GOC-GERMA 75-48/TP-28, CEA Saclay (1975), and, Home Computer Codes POLAR and POPOL, IRF/LNS/GT, CEA Saclay (1975).
- [82] M. Froissart et R. Stora, *Dépolarisation d'un faisceau de protons polarisés dans un synchrotron*, NIM 7 (1960) 297-305.
- [83] J.L. Hamel, *mini graphic library LIBMINIGRAF*, CEA-DSM, Saclay, 1996.
- [84] F. Méot and A. Paris, *Concerning effects of fringe fields and longitudinal distribution of b10 in low- $\beta$  regions on dynamics in LHC*, report FERMILAB-TM-2017, August 23, 1997.
- [85] A White Paper: The Cornell-BNL FFAG-ERL Test Accelerator, I. Ben-Zvi et al., Dec. 16, 2014.
- [86] F. Méot, *End-to-end 9-D polarized bunch transport in FFAG eRHIC*, EIC14 workshop, <http://www.jlab.org/conferences/eic2014/> (2014).
- [87] <http://www.gnuplot.info/>
- [88] F. Méot et al., *Polarized e-bunch acceleration at Cornell RCS. Tentative tracking simulations*, EIC Tech Note 57, BNL, Oct. 2018; <https://technotes.bnl.gov/PDF?publicationId=42654>

# **INDEX**

## Index

- .plt, 17, 93, 129, 176, **213**, 214, 215
- .plt: log stepwise particle data - see **OPTIONS**, 314
- ALE*, 216
  
- acceleration, 84, 180, 252, 326, 329
- AGSMM, **69**, 216, **237**
- AGSQUAD, **70**, **238**
- AGSQUADS: modify winding currents - see **OPTIONS**, 314
- AIMANT, **71**, 102, **239**
  - parameter numbering, 161
- alignment (mis-) of optical elements, **216**
- AUTOREF, **76**, **243**
  
- backward ray-tracing, **213**
- BEAMBEAM, **77**, **244**
  - spin kick, 77, 244
- BEND, 30, **78**, **245**
- BETATRON, **80**, **247**
- BINARY, **156**, **248**
- BORO, **60**, **63**, **67**, 85, 256, **304**, **308**, **311**
- $BORO \times D_{ref}$ , 69, **84**, **88**, 186, **213**, 285
- $B\rho_{ref}$ , 69, **84**, 88, 186, **213**, 213, 285
- BREVOL, 29, **81**, **249**
  
- CARTEMES, 29, 81, **82**, 117, 127, 128, 132, 133, 144, 215, **250**, 277, 346, 385
- cartesian coordinates, **216**
- CAVITE, **84**, 86, 183, 216, **252**
  - double RF system, 86, 252
  - particle path length reset, 181, 215
  - recovering from a crash, 89
- CHAMBR, **91**, 91, 181, 182, **254**, 328
- CHANGREF, 76, 91, **92**, 94, 174, **255**
- checking field, **213**
- checking trajectories, **213**
- chromaticities
  - FITting, 166
  - from **MATRIX**, 204
  - from **TWISS**, 210
    - example, 352
  - how to compute them, 215
- CIBLE, **94**, 173, **256**
- circular accelerators, 78, 180, 215
- closed orbit, 17, 84, 164, 210
  - computation, 89
  - reference trajectory, 213
- COLLIMA, **95**, 181, 182, **257**, 328
- CONSTY: constant Y and Z coordinates - see **OPTIONS**, 314
- coupling, **203**, **210**
- CYCLOTRON, **96**, **258**
- cyclotron, 84, 87, 105
  
- DECAPOLE, **98**, 178, **261**, 325
- DIPOLE, 71, 91, **99**, 104, 160, 216, **262**, 361
  - parameter numbering, 161
- DIPOLE-M, 71, **102**, 133, 219, **264**, 320
  - parameter numbering, 161
- DIPOLES, 71, **104**, **266**
- do-loop
  - REBELOTE, 180, 326
  - SVDOC, 191, 341
  - TWISS, 210, 351
- DODECAPO, 31, **108**, 178, **267**, 325
- double RF system, 86
- DRIFT, **109**, 174, **268**
  - split, 109, 214, 268
- droite de coupure: see integration boundary, 82
  
- EBMULT, 30, 31, **110**, 178, **269**
  - parameter numbering, 161
- EL2TUB, 29, **111**, **271**
- ELCYLDEF, **112**, **272**
- ELMIR, **113**, **273**
- ELMIRC, **114**, **274**
- ELMULT, 31, 110, **115**, 178, **275**
  - parameter numbering, 161
- ELREVOL, 29, **117**, 215, **276**
- EMMA, **118**, **277**
- END, 155, **157**, 205, **282**
- Enge fringe field model, 71, 96, 105, 106, 136
- ERRORS, **158**, **278**
- ESL, **109**, 174, **268**
  
- FAISCEAU, **196**, **279**
- FAISCNL, **196**, 215, **279**, 385
- FAISTORE, **196**, 215, **279**, 385
  - binary storage, 197
  - in presence of *FIT[2]*, 168, 197
- FFAG, 84, 87, 105, 118, **119**, 120, 123, **280**
- FFAG magnet
  - radial, 105, **119**, **280**
  - spiral, **122**, **258**, **281**
- FFAG-SPI, **122**, **281**
- Field maps, **220**
  - 1-D, 29
  - 2-D, arbitrary (X,Y) plane, 30, 35
  - 2-D, mid-plane, 29, 32
  - 3-D, 30, 37
  - binarizing a file, 156
  - binary file, 156
  - HEADER, 144, 220
  - interpolation, **31**
    - 1-D, **31**
    - 2-D, arbitrary plane, **35**
    - 2-D, mid-plane symmetry, **32**
    - 2-D, polar frame, **34**
    - 3-D, **37**
  - making a -, 176
  - positioning, 216
- Field maps, electric, 31
  - 1-D, 117
  - 2-D, arbitrary (X,Y) plane, 128
- Field maps, magnetic, 31
  - 1-D, 81
  - 2-D, arbitrary plane, 127
  - 2-D, mid-plane, 71, 82, 102, 118, 133, 143
  - 3-D, 118, 143
- FIN, **157**, **282**



**FINISH, 159, 283**  
**FIT[2], 155, 160, 160, 163, 284, 326**  
 combined with REBELOTE, 167, 181  
 constraint, **163, 165, 286**  
 chromaticities, using SCALING, 166  
 tunes, using SCALING, 166  
 coupled variables, **163**  
 using SCALING, 163, **186**  
 in presence of FAISTORE, 168  
 max. number of iterations, **168, 284**  
 no final run, 168  
 numbering keywords, 160  
 numbering parameters, 160  
 options, **168, 170**  
 penalty, **168, 284**  
 updated input data file, **168**  
 variable values, 284  
 get, **170, 291**  
 range, **163**  
 save, **168**  
 zgoubi.FIT.out.dat, **168**  
**FLIP-ped field map, 81, 82, 117, 127, 128, 133, 134, 144**  
**FOCALE, 198, 289**  
**FOCALEZ, 199, 289**  
 fringe field, 74, 78  
 Enge coefficients, 71, 96, 105, 106, 136  
 flutter, 96  
 overlapping, 104  
**GASCAT, 169, 290**  
**GETFITVAL, 155, 170, 291**  
 gnuplot, 385  
**GOTO, 171, 294**  
**HALBACH, 125, 292**  
**HELIX, 126, 293**  
**HISTO, 61, 95, 174, 180, 182, 200, 200, 220, 295, 328**  
*IC*, 215  
*ID*, **82, 250, 258, 281**  
*IDMAX*, **63, 63**  
*IEX*, 35, **64, 82, 91, 95, 193, 197, 207, 220**  
*IL*, 91, 161, 162, 176, **213, 215**  
**IMAGE, 76, 198, 198, 296**  
**IMAGES, 63, 198, 198, 296**  
**IMAGESZ, 199, 296**  
**IMAGEZ, 199, 296**  
*IMAX*, **60, 63, 67, 84, 85, 174, 180, 188, 196, 208, 215, 219**  
**INCLUDE, 172, 297**  
**INTEG, 182, 328**  
 integration boundary, 81, 82, 117, 127, 128, 144, 249–251, 346  
 integration step size, 220  
 coded, **219**  
 negative, **174, 213**  
**IODRE, 30, 32, 75, 82, 103, 127, 128, 144, 178**  
 keyword numbering (in FIT[2], REBELOTE[IOPT=1], etc.),  
 160  
**KPOS, 216**  
**LABEL, 129, 183, 197, 205, 207,**  
 LABEL1, LABEL2 215, **215,**  
 LABEL1, LABEL2 268, 279, 301, 318, 336, 385  
**MAP2D, 30, 127, 144, 299**  
**MAP2D-E, 30, 128, 300**  
 maps, summing, 249  
**MARKER, 129, 301, 385**  
**MATRIX, 65, 66, 76, 147, 160, 164, 202, 203, 243, 302, 309**  
 1st, 2nd, 3rd order transport coefficients, 202, 302  
 beam matrix, 175, 202  
 periodic beam matrix, 210  
 spin rotation matrix, 208  
**MCDESINT, 61, 63, 155, 173, 174, 180–182, 200, 303, 328,**  
 361  
**MCOBJET, 60, 61, 164, 173, 180, 220, 304, 326**  
 misalignment, **158, 216, 216**  
 momentum compaction, 352  
**MONITOR: monitoring zgoubi execution - see OPTIONS, 314**  
 Monte Carlo, 60, 67, 173, 180, 219, 304, 311  
 multi-particle, 180, **219, 326**  
 multi-turn tracking, 60, 84, 155, **180, 180, 183, 213, 215, 326,**  
 361, 374, 385  
**MULTIPOL, 30, 31, 110, 115, 130, 178, 183, 307, 325**  
 parameter numbering, 161  
 negative charge, 60, 63, 220, 304, 308  
 negative momentum, 60, 63, 220, 304, 308  
 negative rigidity, **220**  
**NPASS, 84, 180, 197, 200, 207, 215, 219, 279, 326, 336**  
 numbering  
 of keywords (in FIT[2], REBELOTE[IOPT=1], etc.), 160  
 of parameters (in FIT[2], REBELOTE[IOPT=1], etc.), 160  
**OBJET, 63, 76, 85, 91, 163, 164, 173, 198, 210, 216, 220, 308**  
 for 1st, 2nd, 3rd order *MATRIX* computation, 66, 308  
 recovering from a crash, 65, 310  
**OBJETA, 67, 67, 180, 311**  
**OCTUPOLE, 131, 178, 312, 325**  
 offsetting optical elements - X, Y or Z, 158, **216, 278**  
**OPTICS, 175, 313**  
**OPTIONS, 176, 214, 314**  
 .plt, 314  
*IL*, 176  
 AGSQUADS, 314  
 CONSTY, 176, 314  
 MONITOR, 314  
 RAND\_SEED, 314  
 WRITE, 314  
 orbit  
 correction, **55**  
 SVDOK, 55, 191  
 finding, using FIT[2], 352  
 orbit correction  
 using *FIT[2]*, 55  
**ORDRE, 178, 316**  
 outpoi.lis, 132  
 parameter numbering (in FIT[2], REBELOTE[IOPT=1], etc.),  
 160  
 particle path length reset by CAVITE, 181, 215  
**PARTICUL, 67, 95, 110, 111, 113–115, 117, 138, 150, 153,**  
 155, 173, **179, 187, 188, 190, 193, 303, 317**  
**PICKUPS, 205, 215, 318**  
**PLOTDATA, 206, 319**  
**POISSON, 132, 320**

polar coordinates, **219**  
 POLARMES, **133, 219, 321**  
 positioning of optical elements, **216**  
 PRINT, 385  
     CAVITE, 90  
     HISTO, 200  
     MATRIX, 204  
     OPTICS, 175  
     SPNPRT, 208  
 PS170, **134, 322**  
  
 QUADISEX, 29, 30, **135, 323**  
 QUADRUPO, 30, 31, 91, 98, 115, 130, 131, **136, 139, 160,**  
     161, 178, 216, **324, 325**  
     parameter numbering, 161  
  
 RAND\_SEED: random seed initialization - see OPTIONS, 314  
 REBELOTE, 60, 65, 84, 155, 160, **180, 180, 183, 197, 200,**  
     205, 207, 215, 216, 219, 279, 308, **326, 336, 361**  
     circular accelerators, 180, 215  
     combined with FIT[2], 167, 181  
     recovering from a crash, 181  
 recovering from a crash, 65, 89, 181, 186, 310  
 reference rigidity, 60, 63, **84, 186, 213**  
 reference trajectory, closed orbit, **213**  
 RESET, **182, 328**  
 rotating optical elements - X, Y or Z, 158, 216, 278  
  
 SCALING, 85, 86, 182, **183, 183, 215, 216, 252, 328, 329**  
     & FIT  
     coupling variables, 163, 186  
     parameter numbering, 162  
     recovering from a crash, 186  
     scaling an arbitrary parameter, 183  
     scaling fields in optical elements, 183  
 SEPARA, **138, 330**  
 SEXQUAD, 29, **135, 331**  
 SEXTUPOL, 30, **139, 178, 325, 332**  
 SOLENOID, **140, 333**  
 SPACECHARG, 53, **187, 334**  
 SPES2, 361, 363  
 SPES3, 361, 369  
 SPIN  
      $\vec{n}_0$  vector, 166  
     beam-beam kick, 77, 244  
     diffusion integrals, **44, 210**  
     rotation matrix, 208  
     example, 337  
     tracking, **39, 92, 148, 181, 188, 195, 200, 207, 220, 295,**  
         326, **338, 361**  
 SPINR, **142, 335**  
 SPNPRNL, **207, 336, 385**  
 SPNPRT, 190, **208, 337**  
 SPNSTORE, **207, 336, 385**  
 SPNTRK, 155, 180–182, **188, 220, 328, 338**  
 SRLOSS, **190, 339, 340**  
 SRPRNT, **209, 340**  
 stopped particles, 91, 95, 181, 197, **220, 254, 257**  
 storage files, 383  
 SVD, **191, 341**  
     iterate orbit defect & correction, 191  
     multiple reference orbits, 191  
     SVD matrix, 191  
     updated zgoubi.SVD.out.dat, 191  
     zgoubi.SVD\_Amatrix.out, 191  
 synchrotron motion, 84, 180, 183, 252, 326, 329  
 synchrotron radiation, **45**  
     energy loss, **45, 86, 190, 209, 327, 339**  
     energy loss treatment in CAVITE, 86, 88  
     integrals, **47, 210**  
     spectral angular brightness, **48, 193, 343**  
 SYNRAD, **193, 343**  
 SYSTEM, **194, 344**  
 system call, 194, 344  
  
 tag, symmetry index, 63  
 tag, tag character, 61, 63, 64, 174, 193, 196, 197, 200, 207  
 TARGET, **94, 256**  
 time varying fields, **213**  
 TOSCA, 29, **143, 161, 178, 215, 216, 346, 361, 385**  
     parameter numbering, 161  
 TRANSMAT, **147, 349**  
 transport coefficients, **202, 302**  
     1st, 2nd or 3rd order, 202, 302  
 TRAROT, 79, **148, 350**  
 Tunes  
     FITting, 166  
 TWISS, 65, **210, 351**  
     find orbits with FIT[2], 210, 351  
     example, 352  
  
 UNDULATOR, **149, 353**  
 UNIPOT, 29, **150, 354**  
 Units in zgoubi, **212**  
  
 variable (FIT, FIT2), **160, 160, 284**  
 VENUS, 29, 30, 151, 152, **355**  
 VFFA, **152, 356**  
  
 wedge angle, 78  
 WIENFILT, 30, **153, 357**  
 WRITE: allow/inhibit - see OPTIONS, 314  
  
 XCE, 216  
 XPAS  
     negative, 174  
 XPAS, coded, 219  
 XPAS, negative, 213  
  
 YCE, 216  
 YMY, 79, **154, 358**  
  
 zgoubi, 383  
     - running the code, 383  
     zgoubi output files, 221, 383  
     zgoubi.CAVITE.out, **90**  
     zgoubi.dat, 197, 207, 383  
     zgoubi.f, 383  
     zgoubi.fai, 196, 207, **223, 279, 308, 385**  
     zgoubi.FIT.out.dat, **168**  
     zgoubi.HISTO.out, **200**  
     zgoubi.impdev.out, **214, 214**  
     zgoubi.map, 215, 385  
     zgoubi.MATRIX.out, **204**  
     zgoubi.OPTICS.out, **175**

zgoubi.PICKUP.out, 205  
zgoubi.plt, 129, 214, 215, **224**, 301, 385  
zgoubi.res, 213, 215, **222**, 383  
zgoubi.spn, 207, 336, 385  
zgoubi.SPNPRT.Out, 285, 298, 327, 337  
zgoubi.sre, 193  
zgoubi.TWISS.out, 210, **226**  
zpop, 48, 196, 207, 214, **382**  
- running the code, 383