

# ZGOUBI USERS' GUIDE

ZGOUBI ON WEB :

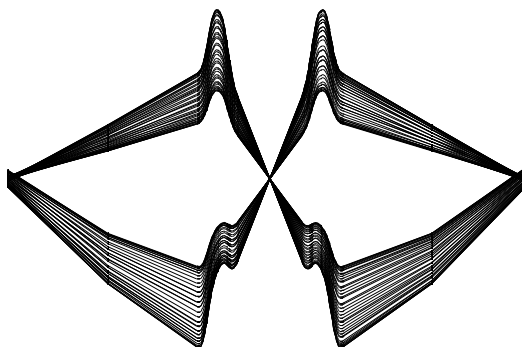
<http://sourceforge.net/projects/zgoubi/>

**François Méot**

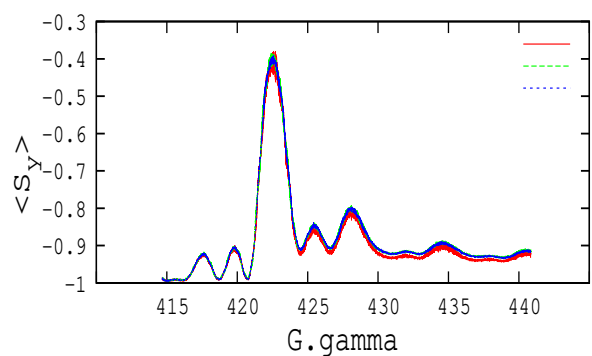
*Brookhaven National Laboratory  
Collider-Accelerator Department  
Upton, NY 11973*

29 April 2019  
(r1221)

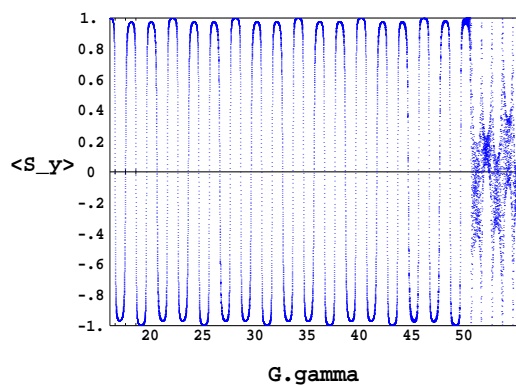
ATLAS & CMS IRs, LHC



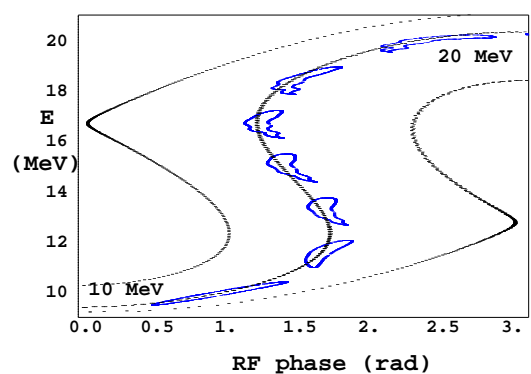
Snake Resonance Crossing in RHIC



Spin flip during ramp, AGS with snakes



Serpentine acceleration in EMMA



**Cover figures :**

*upper left* : collision optics at ATLAS and CMS [70],

*upper right* : evolution of the polarization upon crossing of the  $393+Q_y$  spin resonance in RHIC [14],

*lower left* : spin-flipping over an acceleration cycle in the AGS, in presence of the helical snakes [13],

*lower right* : serpentine acceleration in the prototype linear FFAG EMMA [52, 54].

## 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>11</b>
<b>PREFACE TO THE BNL EDITION (2015+)</b>	<b>13</b>
<b>PREFACE TO THE BNL EDITION (2012)</b>	<b>14</b>
<b>INTRODUCTION TO THE 4th EDITION (1997)</b>	<b>15</b>
<b>INTRODUCTION</b>	<b>17</b>
<b>1 NUMERICAL CALCULATION OF MOTION AND FIELDS</b>	<b>19</b>
1.1 <b>zgoubi</b> Frame . . . . .	19
1.2 Integration of the Lorentz Equation . . . . .	19
1.2.1 Integration in Magnetic Fields . . . . .	21
1.2.2 Integration in Electric Fields . . . . .	21
1.2.3 Integration in Combined Electric and Magnetic Fields . . . . .	24
1.2.4 Calculation of the Time of Flight . . . . .	25
1.3 Calculation of $\vec{E}$ and $\vec{B}$ Fields and their Derivatives . . . . .	25
1.3.1 1-D (Axial) Analytical Field Models and Extrapolation . . . . .	26
1.3.2 Extrapolation from 1-D axial field map . . . . .	26
1.3.3 Extrapolation From Median Plane Field Models . . . . .	26
1.3.4 Extrapolation from Arbitrary 2-D Field Maps . . . . .	27
1.3.5 Interpolation in 3-D Field Maps . . . . .	27
1.3.6 2-D Analytical Field Models and Extrapolation . . . . .	27
1.3.7 3-D Analytical Models of Fields . . . . .	27
1.4 Calculation of $\vec{E}$ and $\vec{B}$ from Field Maps . . . . .	28
1.4.1 1-D Axial Map, with Cylindrical Symmetry . . . . .	28
1.4.2 2-D Median Plane Map, with Median Plane Antisymmetry . . . . .	29
1.4.3 Arbitrary 2-D Map, no Symmetry . . . . .	31
1.4.4 3-D Field Map . . . . .	32
<b>2 SPIN TRACKING</b>	<b>35</b>
2.1 Introduction . . . . .	35
2.2 Integration in Magnetic Fields . . . . .	35
2.3 Integration in Electric Fields . . . . .	36
2.4 Integration in Combined Electric and Magnetic Fields . . . . .	37
2.5 Radiative polarization . . . . .	38
2.6 Spin Diffusion . . . . .	38
<b>3 SYNCHROTRON RADIATION</b>	<b>39</b>
3.1 Energy Loss and Dynamical Effects . . . . .	39
3.2 Spectral-Angular Radiated Densities . . . . .	41
3.2.1 Calculation of the Radiated Electric Field . . . . .	41
3.2.2 Calculation of the Fourier Transform of the Electric Field . . . . .	43
<b>4 SPACE CHARGE</b>	<b>45</b>
<b>5 ORBIT CORRECTION</b>	<b>47</b>
<b>6 DESCRIPTION OF THE AVAILABLE PROCEDURES</b>	<b>49</b>
6.1 Introduction . . . . .	49
6.2 Definition of an Object . . . . .	51
6.2.1 MCOBJET : Monte-Carlo generation of a 6-D object . . . . .	52
6.2.2 OBJET : Generation of an object . . . . .	55
6.2.3 OBJETA : Object from Monte-Carlo simulation of decay reaction [32] . . . . .	59
6.3 Declaring Options . . . . .	60

6.3.1	BINARY : <i>BINARY</i> ↔ <i>FORMATTED</i> data converter . . . . .	61
6.3.2	END or FIN : End of input data list . . . . .	62
6.3.3	ERRORS : Injecting errors in optical elements . . . . .	63
6.3.4	FIT, FIT2 : Fitting procedure . . . . .	64
6.3.5	GASCAT : Gas scattering . . . . .	71
6.3.6	GETFITVAL : Get values of <i>variables</i> as saved from former FIT[2] run . . . . .	72
6.3.7	MCDESINT : Monte-Carlo simulation of in-flight decay[36] . . . . .	73
6.3.8	OPTICS : Write out optical functions. Log to zgoubi.OPTICS.out . . . . .	75
6.3.9	OPTIONS : Global or special options . . . . .	76
6.3.10	ORDRE : Taylor expansions order . . . . .	77
6.3.11	PARTICUL : Particle characteristics . . . . .	78
6.3.12	REBELOTE : 'Do it again' . . . . .	79
6.3.13	RESET : Reset counters and flags . . . . .	81
6.3.14	SCALING : Power supplies and R.F. function generator . . . . .	82
6.3.15	SPACECHARG : Space charge . . . . .	86
6.3.16	SPNTRK : Spin tracking . . . . .	87
6.3.17	SRLOSS : Synchrotron radiation energy loss [20] . . . . .	89
6.3.18	SVDOC : SVD orbit correction . . . . .	90
6.3.19	SYNRAD : Synchrotron radiation spectral-angular densities . . . . .	92
6.3.20	SYSTEM : System call . . . . .	93
6.4	Optical Elements and Related Numerical Procedures . . . . .	95
	From AGSMM, AGSQUAD,... to EBMULT, EL2TUB,... to MAP2D-E, MULTIPOL,... to TOSCA, YMY . . . . .	95
6.5	Output Procedures . . . . .	168
6.5.1	FAISCEAU, FAISCNL, FAISTORE : Print/Store particle coordinates . . . . .	169
6.5.2	FOCALE, IMAGE[S] : Particle coordinates and beam size ; localization and size of horizontal waist . . . . .	171
6.5.3	FOCALEZ, IMAGE[S]Z : Particle coordinates and beam size ; localization and size of vertical waist . . . . .	172
6.5.4	HISTO : 1-D histogram . . . . .	173
6.5.5	IMAGE[S][Z] : Localization and size of vertical waists . . . . .	174
6.5.6	MATRIX : Calculation of transfer coefficients, periodic parameters . . . . .	175
6.5.7	PICKUPS : Bunch centroid path; orbit . . . . .	177
6.5.8	PLOTDATA : Intermediate output for the PLOTDATA graphic software . . . . .	178
6.5.9	SPNPRNL, SPNSTORE : Print/Store spin coordinates . . . . .	179
6.5.10	SPNPRT : Print spin coordinates . . . . .	180
6.5.11	SRPRNT : Print SR loss statistics . . . . .	181
6.5.12	TWISS : Calculation of periodic optical parameters. Log to zgoubi.TWISS.out . . . . .	182
6.6	Complements Regarding Various Functionalities . . . . .	183
6.6.1	Units in <b>zgoubi</b> . . . . .	183
6.6.2	Reference rigidity . . . . .	184
6.6.3	Time Varying Fields . . . . .	184
6.6.4	Backward Ray-Tracing . . . . .	184
6.6.5	Checking Trajectories and Fields Across Optical Elements - the <i>IL</i> , <i>.plt</i> and <i>IC</i> flags . . . . .	184
6.6.6	Labeling Keywords . . . . .	185
6.6.7	Multi-turn Tracking in Circular Machines . . . . .	185
6.6.8	Positioning, (Mis-)Alignment, of Optical Elements and Field Maps . . . . .	186
6.6.9	Coded Integration Step . . . . .	188
6.6.10	Ray-tracing of an Arbitrarily Large Number of Particles . . . . .	188
6.6.11	Stopped Particles : The <i>IEX</i> Flag . . . . .	188
6.6.12	Negative Rigidity . . . . .	188

## **PART B Keywords and input data formatting 189**

### **GLOSSARY OF KEYWORDS, PART B 191**

### **OPTICAL ELEMENTS VERSUS KEYWORDS 193**

### **INTRODUCTION 195**

## **PART C Examples of input data files and output result files 305**

### **INTRODUCTION 307**

<b>1</b>	<b>MONTE CARLO IMAGES IN SPES 2</b>	<b>309</b>
<b>2</b>	<b>TRANSFER MATRICES ALONG A TWO-STAGE SEPARATION KAON BEAM LINE</b>	<b>312</b>
<b>3</b>	<b>IN-FLIGHT DECAY IN SPES 3</b>	<b>315</b>
<b>4</b>	<b>USE OF THE FITTING PROCEDURE</b>	<b>318</b>
<b>5</b>	<b>MULTITURN SPIN TRACKING IN SATURNE 3 GeV SYNCHROTRON</b>	<b>320</b>
<b>6</b>	<b>MICRO-BEAM FOCUSING WITH <math>\vec{E} \times \vec{B}</math> QUADRUPOLES</b>	<b>322</b>
<b>PART D</b>	<b>Running zgoubi and its post-processor/graphic interface zpop</b>	<b>327</b>
	<b>INTRODUCTION</b>	<b>329</b>
<b>1</b>	<b>GETTING TO RUN zgoubi AND zpop</b>	<b>329</b>
1.1	Making the Executable Files <b>zgoubi</b> and <b>zpop</b> . . . . .	329
1.1.1	The transportable package zgoubi . . . . .	329
1.1.2	The post-processor and graphic interface package zpop . . . . .	329
1.2	Running <b>zgoubi</b> . . . . .	329
1.3	Running <b>zpop</b> . . . . .	329
<b>2</b>	<b>STORAGE FILES</b>	<b>329</b>
	<b>REFERENCES</b>	<b>332</b>
	<b>INDEX</b>	<b>335</b>



# **PART A**

## **Description of software contents**





## Glossary of Keywords, Part A

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

<b>AGSMM</b>	AGS main magnet .....	95
<b>AGSQUAD</b>	AGS quadrupole .....	96
<b>AIMANT</b>	Generation of dipole mid-plane 2-D map, polar frame .....	97
<b>AUTOREF</b>	Transport beam into a new reference frame .....	102
<b>BEAMBEAM</b>	Beam-beam lens .....	103
<b>BEND</b>	Bending magnet, Cartesian frame .....	104
<b>BETATRON</b>	Betatron core (momentum kick) .....	106
<b>BINARY</b>	<i>BINARY</i> ↔ <i>FORMATTED</i> data converter .....	61
<b>BREVOL</b>	1-D uniform mesh magnetic field map .....	107
<b>CARTEMES</b>	2-D Cartesian uniform mesh magnetic field map .....	108
<b>CAVITE</b>	Accelerating cavity .....	110
<b>CHAMBR</b>	Long transverse aperture limitation .....	114
<b>CHANGREF</b>	Transformation to a new reference frame .....	115
<b>CIBLE</b>	Generate a secondary beam following target interaction .....	117
<b>COLLIMA</b>	Collimator .....	118
<b>DECAPOLE</b>	Decapole magnet .....	119
<b>DIPOLE</b>	Dipole magnet, polar frame .....	120
<b>DIPOLE-M</b>	Generation of dipole mid-plane 2-D map, polar frame .....	122
<b>DIPOLÉS</b>	Dipole magnet <i>N</i> -tuple, polar frame .....	124
<b>DODECAPO</b>	Dodecapole magnet .....	127
<b>DRIFT</b>	Field free drift space .....	128
<b>EBMULT</b>	Electro-magnetic multipole .....	129
<b>EL2TUB</b>	Two-tube electrostatic lens .....	130
<b>ELMIR</b>	Electrostatic N-electrode mirror/lens, straight slits .....	131
<b>ELMIRC</b>	Electrostatic N-electrode mirror/lens, circular slits .....	132
<b>ELMULT</b>	Electric multipole .....	133
<b>ELREVOL</b>	1-D uniform mesh electric field map .....	135
<b>EMMA</b>	2-D Cartesian or cylindrical mesh field map for EMMA FFAG .....	136
<b>END</b>	End of input data list .....	62
<b>ERRORS</b>	Injecting errors in optical elements .....	63
<b>ESL</b>	Field free drift space .....	128
<b>FAISCEAU</b>	Print particle coordinates .....	169
<b>FAISCNL</b>	Store particle coordinates in file FNAME .....	169
<b>FAISTORE</b>	Store coordinates every <i>IP</i> other pass at labeled elements .....	169
<b>FFAG</b>	FFAG magnet, <i>N</i> -tuple .....	137
<b>FFAG-SPI</b>	Spiral FFAG magnet, <i>N</i> -tuple .....	139
<b>FIN</b>	End of input data list .....	62
<b>FIT</b>	Fitting procedure .....	64
<b>FOCALE</b>	Particle coordinates and horizontal beam size at distance <i>XL</i> .....	171
<b>FOCALEZ</b>	Particle coordinates and vertical beam size at distance <i>XL</i> .....	172
<b>GASCAT</b>	Gas scattering .....	71
<b>GETFITVAL</b>	Get values of <i>variables</i> as saved from former FIT[2] run .....	72
<b>GOTO</b>	Branching statement .....	141
<b>HISTO</b>	1-D histogram .....	173
<b>IMAGE</b>	Localization and size of horizontal waist .....	171
<b>IMAGES</b>	Localization and size of horizontal waists .....	171
<b>IMAGESZ</b>	Localization and size of vertical waists .....	172
<b>IMAGEZ</b>	Localization and size of vertical waist .....	172
<b>INCLUDE</b>	File include statement .....	142

<b>MAP2D</b>	2-D Cartesian uniform mesh field map - arbitrary magnetic field .....	143
<b>MAP2D-E</b>	2-D Cartesian uniform mesh field map - arbitrary electric field .....	144
<b>MARKER</b>	Marker .....	145
<b>MATRIX</b>	Calculation of transfer coefficients, periodic parameters .....	175
<b>MCDESINT</b>	Monte-Carlo simulation of in-flight decay .....	73
<b>MCOBJET</b>	Monte-Carlo generation of a 6-D object .....	52
<b>MULTIPOL</b>	Magnetic multipole .....	146
<b>OBJET</b>	Generation of an object .....	55
<b>OBJETA</b>	Object from Monte-Carlo simulation of decay reaction .....	59
<b>OCTUPOLE</b>	Octupole magnet .....	147
<b>OPTICS</b>	Write out optical functions. Log to zgoubi.OPTICS.out .....	75
<b>OPTIONS</b>	Global or special options .....	76
<b>ORDRE</b>	Taylor expansions order .....	77
<b>PARTICUL</b>	Particle characteristics .....	78
<b>PICKUPS</b>	Bunch centroid path; orbit .....	177
<b>PLOTDATA</b>	Intermediate output for the PLOTDATA graphic software .....	178
<b>POISSON</b>	Read magnetic field data from <i>POISSON</i> output .....	148
<b>POLARMES</b>	2-D polar mesh magnetic field map .....	149
<b>PS170</b>	Simulation of a round shape dipole magnet .....	150
<b>QUADISEX</b>	Sharp edge magnetic multipoles .....	151
<b>QUADRUPO</b>	Quadrupole magnet .....	152
<b>REBELOTE</b>	'Do it again' .....	79
<b>RESET</b>	Reset counters and flags .....	81
<b>SCALING</b>	Power supplies and R.F. function generator .....	82
<b>SEPARA</b>	Wien Filter - analytical simulation .....	154
<b>SEXQUAD</b>	Sharp edge magnetic multipole .....	151
<b>SEXTUPOL</b>	Sextupole magnet .....	155
<b>SOLENOID</b>	Solenoid .....	156
<b>SPACECHARG</b>	Space charge .....	86
<b>SPINR</b>	Spin rotation .....	157
<b>SPNPRNL</b>	Store spin coordinates into file FNAME .....	179
<b>SPNPRT</b>	Print spin coordinates .....	180
<b>SPNSTORE</b>	Store spin coordinates every <i>IP</i> other pass at labeled elements .....	179
<b>SPNTRK</b>	Spin tracking .....	87
<b>SRLOSS</b>	Synchrotron radiation energy loss .....	89
<b>SRPRNT</b>	Print SR loss statistics .....	181
<b>SVDOC</b>	SVD orbit correction .....	??
<b>SYNRAD</b>	Synchrotron radiation spectral-angular densities .....	92
<b>SYSTEM</b>	System call .....	93
<b>TARGET</b>	Generate a secondary beam following target interaction .....	117
<b>TOSCA</b>	2-D and 3-D Cartesian or cylindrical mesh field map .....	158
<b>TRANSMAT</b>	Matrix transfer .....	161
<b>TRAROT</b>	Translation-Rotation of the reference frame .....	162
<b>TWISS</b>	Calculation of periodic optical parameters. Log to zgoubi.TWISS.out ..	182
<b>UNDULATOR</b>	Undulator magnet .....	163
<b>UNIPOT</b>	Unipotential cylindrical electrostatic lens .....	164
<b>VENUS</b>	Simulation of a rectangular shape dipole magnet .....	165
<b>WIENFILT</b>	Wien filter .....	166
<b>YMY</b>	Reverse signs of <i>Y</i> and <i>Z</i> reference axes .....	167

## Optical elements versus keywords

---

### What can be simulated What keyword(s) can be used for that

---

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 or sector	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	DIPOL, 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, DIPOL
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

**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 “include”, more or less equivalent to “call file” as found in other optics codes, useful for long or repetitive sequences.

A consequence of the latter is the possibility of requesting a “zgoubi.SEQUENCE.Out” output file which contains the whole developed optical sequence in a single file.

## 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 [1]-[4].

**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 191, 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 functioning checks, 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*" (**zgoubi**'s 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 (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 [5]. 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 [1]. 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 [2]

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 [3].

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 SR 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 Jülich, 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 191, has stretched with new simulation and computing procedures, ranging from fitting (*FIT2* procedure ; additional constraints) to overlapping magnetic field capabilities (*DIPOL*, *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 is in the median plane on a reference curve which generally (but not always) coincides with the optical axis of optical elements.

### 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  $M_1$  is obtained in the same way from

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

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.

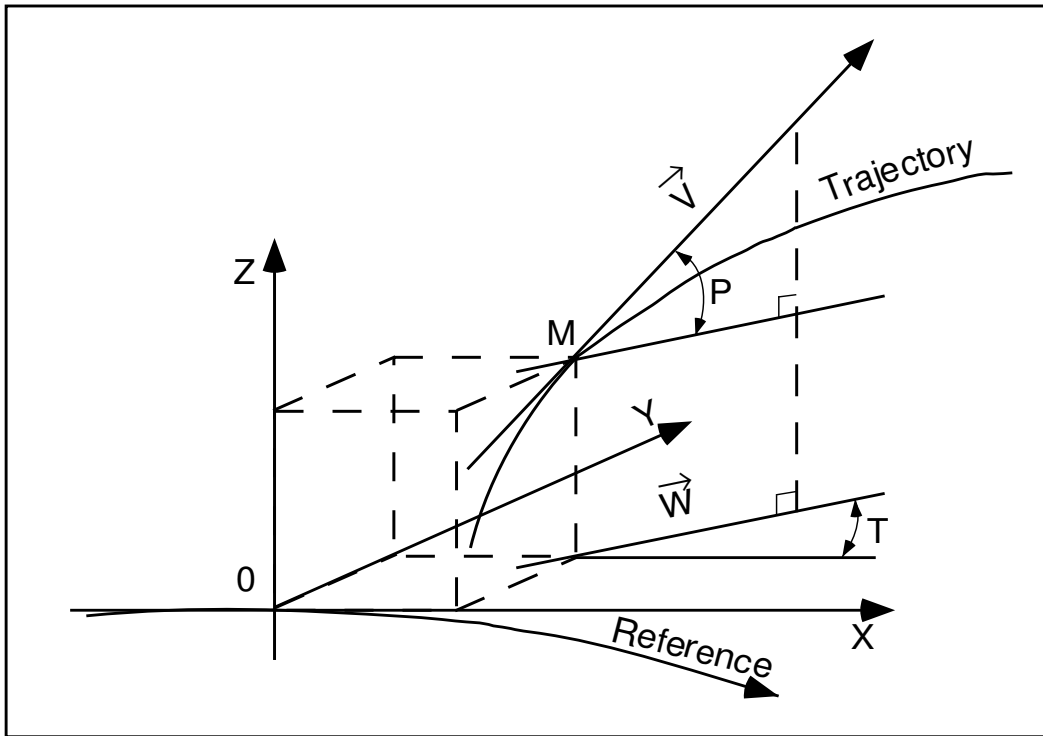


Figure 1: Reference frame and coordinates  $(Y, T, Z, P)$  in **zgoubi**.

$OX$  : in the direction of motion,

$OY$  : normal to  $OX$ ,

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

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

$T$  = angle between  $\vec{W}$  and the  $X$ -axis,

$P$  = angle between  $\vec{W}$  and  $\vec{v}$ .

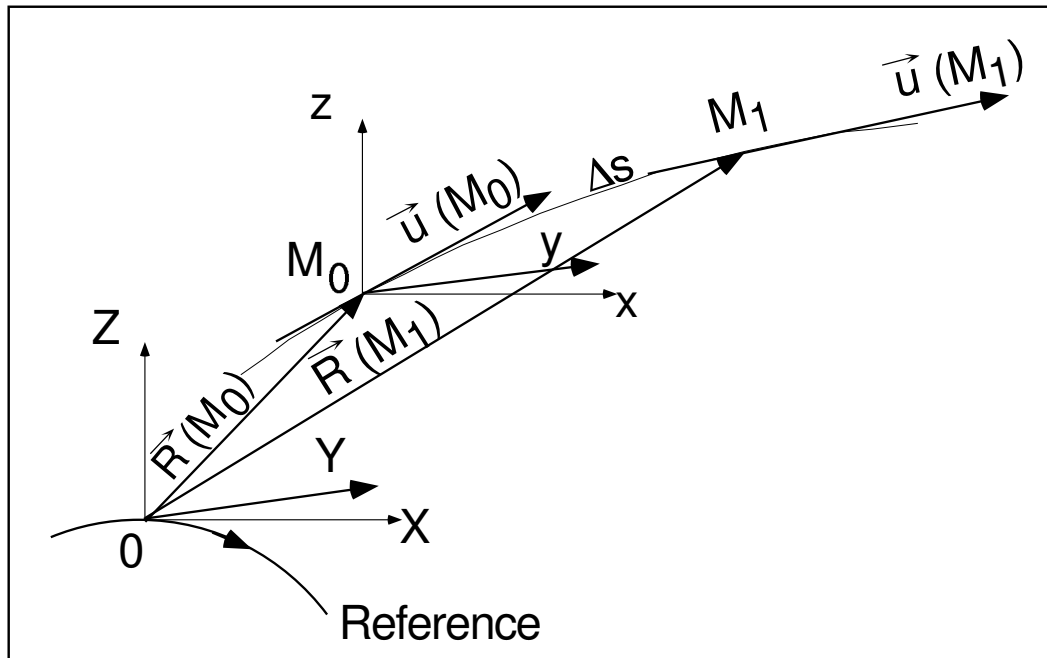


Figure 2: Position and velocity of a particle in the reference frame.

### 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 [6]

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

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

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

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

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

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

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  $(B\rho)^{(n)}$

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

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

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

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

$$p = \frac{v}{c} \frac{W + m_0 c^2}{c}$$

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

$$\begin{aligned} \left(\frac{1}{v}\right) &= \frac{1}{c^2} \frac{W + m_0 c^2}{q B \rho} \\ \left(\frac{1}{v}\right)' &= \frac{1}{c^2} \frac{(\vec{e} \cdot \vec{u})}{B \rho} - \frac{1}{v} \frac{(B \rho)'}{B \rho} \\ \left(\frac{1}{v}\right)'' &= \frac{1}{c^2} \frac{(\vec{e} \cdot \vec{u})'}{B \rho} - 2 \left(\frac{1}{v}\right)' \frac{(B \rho)'}{B \rho} - \frac{1}{v} \frac{(B \rho)''}{B \rho} \\ \left(\frac{1}{v}\right)''' &= \frac{1}{c^2} \frac{(\vec{e} \cdot \vec{u})''}{B \rho} - 3 \left(\frac{1}{v}\right)'' \frac{(B \rho)'}{B \rho} - 3 \left(\frac{1}{v}\right)' \frac{(B \rho)''}{B \rho} - \frac{1}{v} \frac{(B \rho)'''}{B \rho} \\ \left(\frac{1}{v}\right)'''' &= \text{etc.} \end{aligned} \quad (1.2.21)$$

### 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

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

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



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

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

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

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

$$T(M_1) = T(M_0) + \Delta s/v \quad (1.2.25)$$

### 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 [7]

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

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

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 *IORORE* 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 [8]

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 *QUADRUPO*, *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 [9]

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

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

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

#### 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).

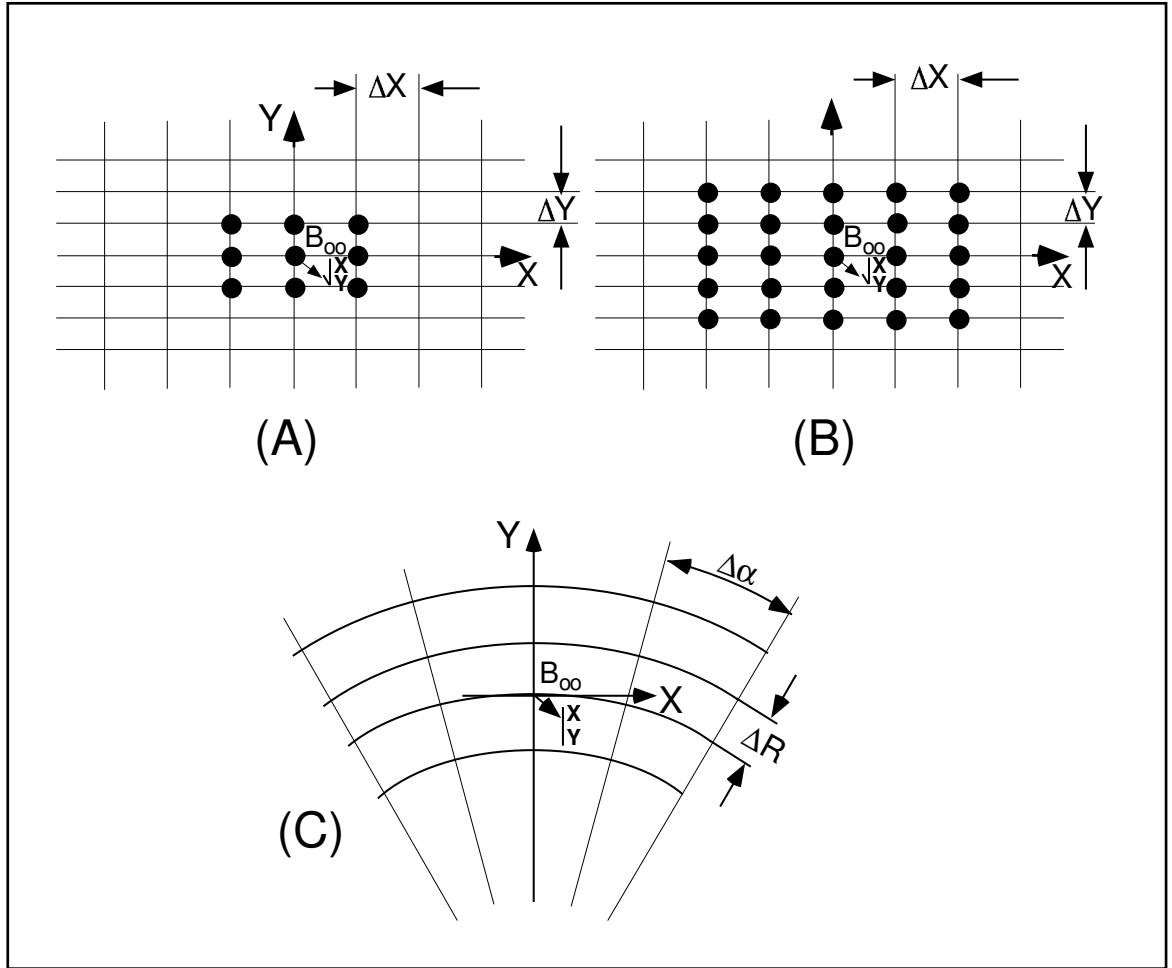


Figure 3: Mesh in the  $(X, Y)$  plane. The grid is centered on the node which is closest to the actual position of the particle.

A : Cartesian mesh, 9-point interpolation grid.

B : Cartesian mesh, 25-point interpolation grid.

C : Polar mesh and moving Cartesian frame.

**zgoubi** provides three types of polynomial interpolation from the mesh (option *IORDRE*) ; 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

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

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  $\partial S / \partial A_{ij} = 0$ .

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 Special 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  $B \equiv B_Z(Z = 0)$ )

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

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^3 B}{\partial X^2 \partial Y} &= \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^3 B}{\partial X \partial Y^2} &= \frac{1}{R} \frac{\partial^3 B}{\partial \alpha \partial R^2} + \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} &= \frac{\partial^3 B}{\partial R^3} \\
\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^4 B}{\partial X^3 \partial Y} &= \frac{1}{R^3} \frac{\partial^4 B}{\partial \alpha^3 \partial R} - \frac{4}{R^4} \frac{\partial^3 B}{\partial \alpha^3} + \frac{3}{R^2} \frac{\partial^3 B}{\partial \alpha \partial R^2} - \frac{3}{R^3} \frac{\partial^2 B}{\partial \alpha \partial R} + \frac{6}{R^4} \frac{\partial B}{\partial \alpha} \\
\frac{\partial^4 B}{\partial X^2 \partial Y^2} &= \frac{1}{R^4} \frac{\partial^4 B}{\partial \alpha^2} - \frac{4}{R^3} \frac{\partial^3 B}{\partial \alpha^2 \partial R} - \frac{2}{R^2} \frac{\partial^2 B}{\partial R^2} + \frac{2}{R^3} \frac{\partial B}{\partial R} + \frac{1}{R^2} \frac{\partial^4 B}{\partial \alpha^2 \partial R^2} + \frac{1}{R} \frac{\partial^3 B}{\partial R^3} \\
\frac{\partial^4 B}{\partial X \partial Y^3} &= \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^4 B}{\partial Y^4} &= \frac{\partial^4 B}{\partial R^4}
\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 6.6.11, p. 188) 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.

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 *IORDRE* in keyword data list - see for instance *MAP2D*, *MAP2D-E*) either a second degree polynomial interpolation from a  $3 \times 3$  points grid (*IORDRE*=2), or a fourth degree polynomial interpolation from a  $5 \times 5$  points grid (*IORDRE*=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).

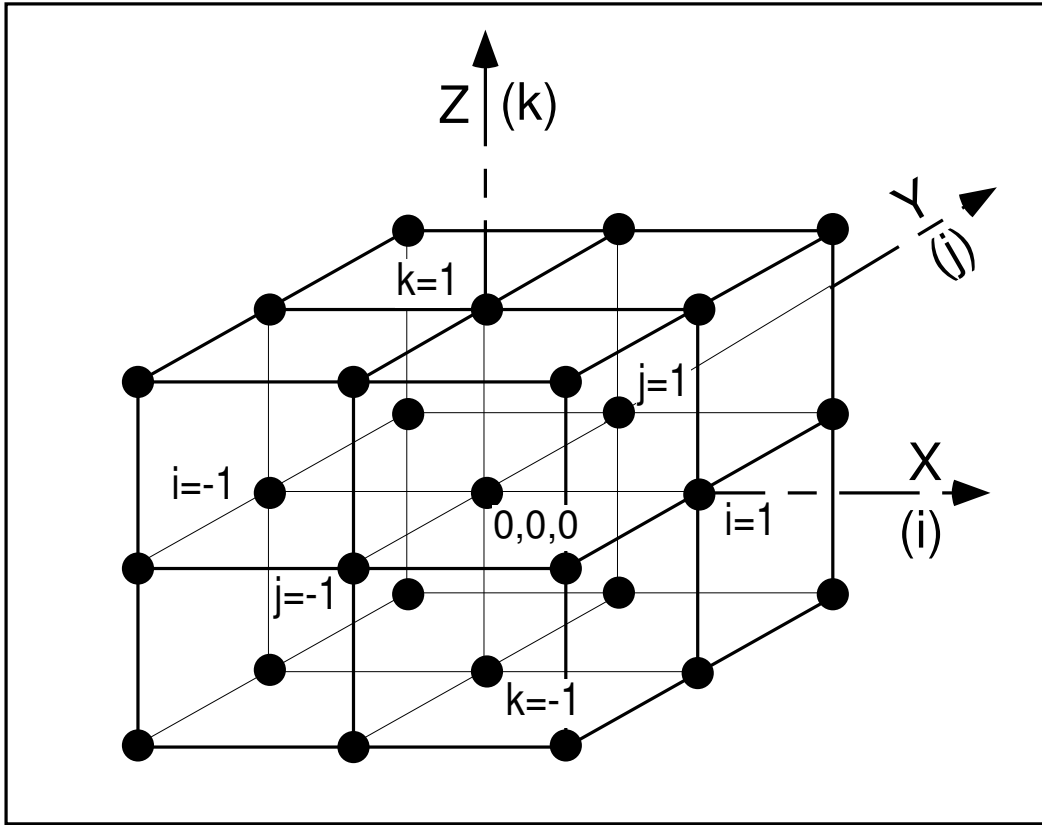


Figure 4: A 3-D 27-point grid is used for interpolation of magnetic or electric fields and derivatives up to second order. The central node of the grid ( $i = j = k = 0$ ) is the closest to the actual position of the particle.

#### 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



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

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

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

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 tracking was implemented in **zgoubi** in the SATURNE years [10], it has been since used in a number of studies, including the transport of polarized muons in the Neutrino Factory [11], polarization survival in the superB project [12], studies on proton polarization with helical snakes at AGS [13] and RHIC [14], in the magic-energy electrostatic ring of the proton-EDM project [15], spin diffusion in eRHIC collider lattice - which includes spin rotators [16].

### 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 [17]

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

with, in the laboratory frame,

$$\vec{\omega} = (1 + \gamma G)\vec{b} + G(1 - \gamma)\vec{b}_{\parallel} + \gamma(G + \frac{1}{1 + \gamma})\frac{\vec{e} \times \vec{v}}{c^2} \quad (2.1.2)$$

$t$ ,  $q$ ,  $m$ ,  $c$ ,  $\gamma$  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.

Equation (2.1.1) is normalized by introducing the same notation as for the Lorentz equation, page 19, 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  $\vec{B} = \vec{b}/B\rho$ ,  $\vec{E} = \vec{e}/B\rho$ ,

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

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. 21.

### 2.3 Integration in Electric Fields

In purely electric fields  $\vec{b} = 0$ , thus eq. (2.1.2) 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 easily 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

$$\begin{aligned}
B\rho \vec{S}' &= \vec{S} \times \vec{\omega}_b + \vec{S} \times \vec{\omega}_e \\
B\rho \vec{S}'' &= \vec{S}' \times \vec{\omega}_b + \vec{S} \times \vec{\omega}_b' + \vec{S}' \times \vec{\omega}_e + \vec{S} \times \vec{\omega}_e' - (B\rho)' \vec{S}' \\
B\rho \vec{S}''' &= \underbrace{\vec{S}'' \times \vec{\omega}_b + 2\vec{S}' \times \vec{\omega}_b' + \vec{S} \times \vec{\omega}_b''}_{\text{Magnetic field component, eq. (2.2.2)}} + \underbrace{\vec{S}''' \times \vec{\omega}_e + 2\vec{S}' \times \vec{\omega}_e' + \vec{S} \times \vec{\omega}_e'' - \left( (B\rho)'' \vec{S}' + 2(B\rho)' \vec{S}'' \right)}_{\text{Electric field component, eq. (2.3.2)}} \\
&\text{etc.}
\end{aligned} \tag{2.4.1}$$

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 [18], namely, see Eqs. 2.1.1, 2.1.2 (in the absence of electric fields),

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

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,

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

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 [19], that event being determined by the Monte Carlo photon emission simulation process described in Sec. 3.1. 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 solving the differential eq. 2.5.1, the technique follows the **zgoubi** method, namely, introducing derivatives with respect to the path variable  $s$ , and fields normalized to  $B\rho$ ,

$$\frac{d\vec{S}}{ds} = \left( \vec{S} \times (\vec{1} + a\gamma)\vec{B} + a(1 - \gamma)\vec{B}_{\parallel} \right) - \frac{1}{\tau_{ST}} \left( \vec{S} - \frac{2}{9}(\vec{S} \cdot \vec{\beta})\vec{\beta} + \frac{8}{5\sqrt{3}} \frac{\vec{\beta} \times \dot{\vec{\beta}}}{|\dot{\vec{\beta}}|} \right) \tag{2.5.3}$$

The derivatives of the unit velocity vector  $\vec{u}$  are outcomes of the solution of particle motion as detailed in Sec. 1.2. The derivatives of the polarization vector are computed as detailed in ec. 2.1.

## 2.6 Spin Diffusion

To be documented.

### 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 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 [20], based on a method developed in the frame of the ELFE project (“EU Lab. For Electrons”) [21, 22, 23]. It was later used, and benchmarked including coupling, to account for damping effects in beam studies for various electron-ring and recirculator projects [12, 24, 25], and to account for spin diffusion studies in eRHIC collider lattice - which includes spin rotators [16].

**SR electromagnetic impulse and spectrum** computation was first installed in **zgoubi** for the study of interference effects at the LEP beam diagnostics miniwiggler [27]. These simulation tools were next used to design the LHC SR beam diagnostics systems, located in the IR4 RF section [28].

#### 3.1 Energy Loss and Dynamical Effects

Most of the contents of the present section are drawn from Ref. [20].

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>1</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>2</sup>

<sup>1</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>2</sup>This leads for instance, in the case of electrons, to the classical formula  $\lambda/\Delta\theta \approx 129.5\text{E(GeV)}/2\pi \approx \gamma/94.9$ .

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

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.

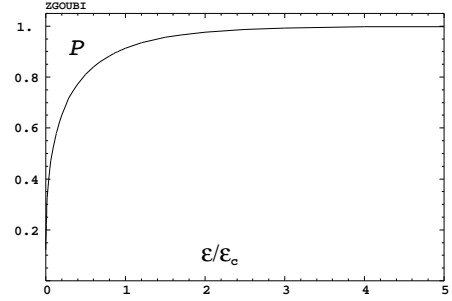
### Energy of the Photons

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>3</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

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



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

About 40 values of  $\mathcal{P}(\epsilon/\epsilon_c)$  computed from eq. 3.1.3 [26], 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  $\epsilon/\epsilon_c = \left(\frac{5 \cdot 2^{1/3} \Gamma(\frac{1}{3})}{12\sqrt{3}\mathcal{P}}\right)^3$  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., *[MCJOBJET]*)
- critical energy  $\epsilon_c(eV) = \frac{3\gamma^3 c \hbar}{2\rho} \frac{1}{e}$ , 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*.

<sup>3</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.



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.

### 3.2 Spectral-Angular Radiated Densities

Most of the content in the present section is drawn from Refs. [27, 28, 29].

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. 5, this is developed in section 3.2.1. These ingredients further allow calculating the spectral-angular density of the radiation<sup>4</sup>, this is developed in section 3.2.2.

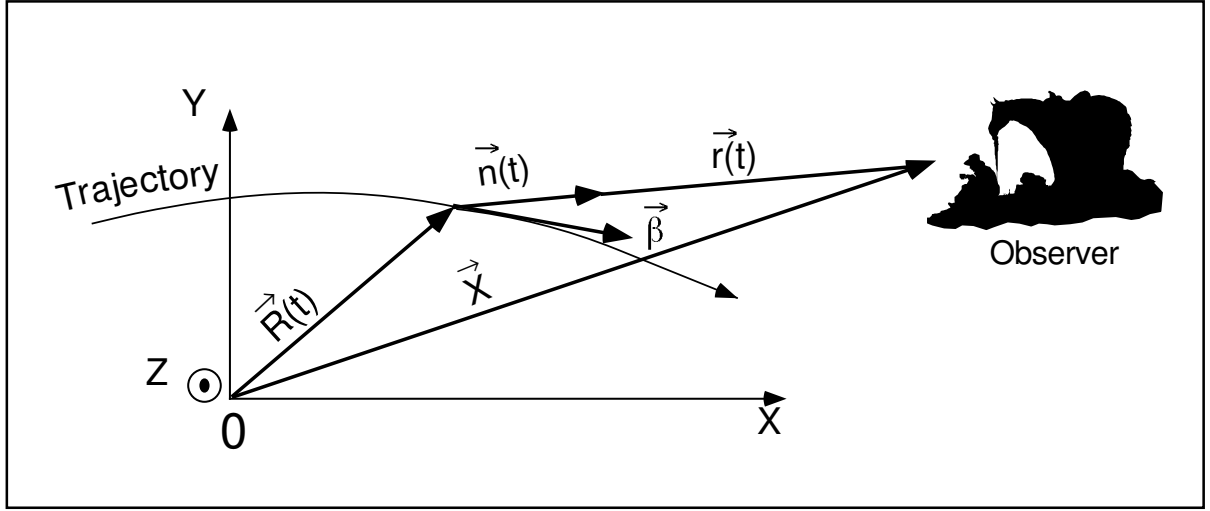


Figure 5: A scheme of the reference frame in **zgoubi** together with the vectors entering in the definition of the electric field radiated by the accelerated particle :

$(x, y)$  : 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  $\vec{v}/c = (1/c)d\vec{R}/dt$ .

#### 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 [30]

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

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

<sup>4</sup>These calculations have been installed in the post-processor **zpop**.

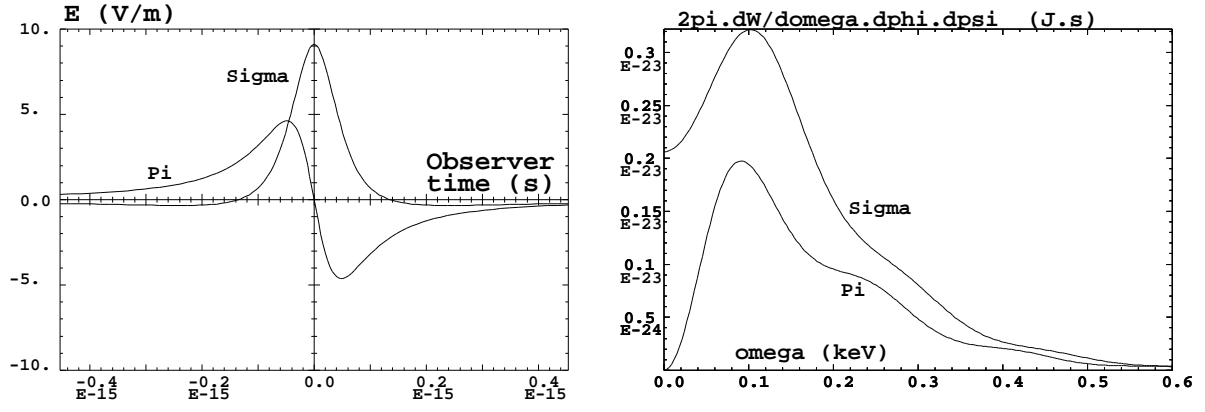


Figure 6: *Left* : typical shapes of the  $\mathcal{E}_\sigma(t)$  and  $\mathcal{E}_\pi(t)$  components of the electric field impulse (eq. 3.2.1) emitted by a 2.5 GeV electron on a  $\rho = 53.6$  m circular trajectory in a  $l = 20$  cm-long dipole, as observed in the direction of the centre of the 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. [29].

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

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  $\vec{e} = 0$ )

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

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. 6 has been produced using **zpop**, it shows the typical shape of the electric field impulse at the observer, with central peak width [29]

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

#### 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

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

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. 6-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 models for the simulation of space charge effects. This capability has been first developed in the context of a PhD work regarding fixed field ring proton driver design studies and accelerator-driven subcritical reactor applications. This section of the guide is essentially drawn from that PhD dissertation [31], which can be referred to for additional details - the document provides benchmarking proofs in both KV and Gaussian density models (see below), 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.

TO BE DOCUMENTED FURTHER



## 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*. Deatils can be found in Sec. 6.3.18, p. 90.

*TO BE DOCUMENTED FURTHER*





## 6 DESCRIPTION OF THE 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 6.6 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**).

### 6.2.1 MCOBJET : Monte-Carlo generation of a 6-D object

*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* identified with 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,  $D = \delta D * T$ .

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 6.6.12, page 188).

**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  $IMAX = IY * IT * IZ * IP * IX * ID$  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. 7.

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)

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

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  $N_\varepsilon > 0$ ).

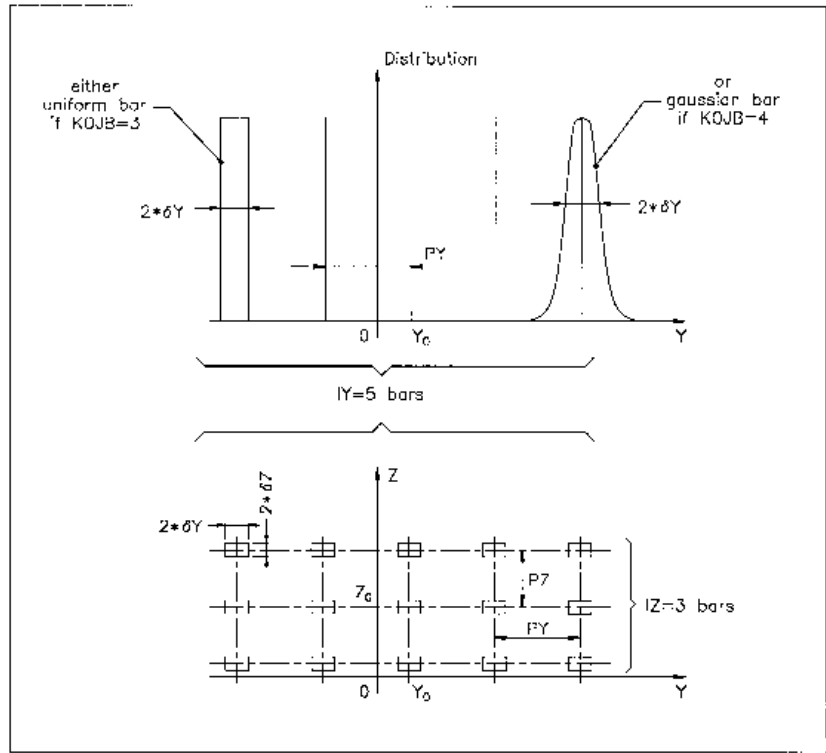


Figure 7: A scheme of input parameters to *MCOBJET* when  $KOBJ= 2$ .

Top : Possible distributions of the  $Y$  coordinate.

Bottom : A 2-D grid in  $(Y, Z)$  space.

### 6.2.2 OBJET : Generation of an object

*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  $IMAX = IY * IT * IZ * IP * IX * ID$  particles with initial coordinates

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

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  $DR * BORO$ , resulting in the rigidity of a particle defined by  $I * dD$  to be  $(DR + I * dD) * BORO$ .

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

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

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  $D * BORO$  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 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  $C = Y, T, Z, \dots$  is changed into  $CF \cdot C + CR$ . 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 appropriate manner. The following *FORTRAN* sequence is an instance, 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) LET (I), IEX(I), (FO(J,I),J=1,6), (F(J,I),J=1,6), I, IREP(I),
          > LET(I), IEX(I), -1.D0+FO(1,I), (FO(J,I),J=2,MXJ),
          > -1.D0+F(1,I), F(2,I), F(3,I),
          > (F(J,I),J=4,MXJ), ENEKI,
          > ID,I, IREP(I), SORT(I), D,D,D,D, RET(I), DPR(I),
          > D, D, D, BORO, IPASS, KLEY, LBL1, LBL2, NOEL
      100  FORMAT(1X,
C1  LET(IT), KEX, 1.D0-FO(1,IT), (FO(J,IT),J=2,MXJ),
      1      A1,1X,I2,1P,7E16.8,
C2  1.D0-F(1,IT), (FO(J,IT),J=2,MXJ),
      2      /,3E24.16,
C3  Z,P*1.D3,SAR, TAR, DS,
      3      /,4E24.16,E16.8,
C4  KART, IT,IREP(IT),SORT(IT),X, BX,BY,BZ, RET(IT), DPR(IT),
      4      /,I1,2I6,7E16.8,
C5  EX,EY,EZ, BORO, IPASS, KLEY, (LABEL(NOEL,I),I=1,2),NOEL
      5      /,4E16.8, I6,1X, A8,1X, 2A10, I5)
      ENDDO

```

where the meaning of the parameters (apart from *D*=dummy real, *ID*=dummy integer) is the following

*LET(I)* : one-character string (for tagging)  
*IEX(I)* : flag, see *KOBJ*= 2 and page 188  
*FO(1-6,I)* : coordinates *D, Y, T, Z, P* and path length of particle number *I*, at the origin.  $D * BORO$  = rigidity  
*F(1-6,I)* : id, at the current position.



*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*, *FAISTORE*.  
If more than *IMAX* 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 *FORTRAN* 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** : As for **KOBJ=3.1**, except for the different format

```

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

```

where *PX*, *PY*, and *PZ*, are the momenta in MeV/*c*. Note that *DPR* will be ignored in this case.

**KOBJ = 3.3** : As for **KOBJ=3.1**, except for the different format :

```

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

```

where *MASS* is the mass in MeV/*c* and *CHARGE* is the charge in units of the elementary charge.

**Note** : For details and possible updates in the formatted read of concern in the *FORTRAN*, regarding options 3.1-3.3, 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 82, an example of job resuming is given page 260.

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

$$dY, \quad dT, \quad dZ, \quad dP, \quad dD$$

These values should be small enough, so that the paraxial ray approximation be valid. The code generates 11 particles, with initial coordinates

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

given the reference trajectory

$$YR, \quad TR, \quad ZR, \quad PR, \quad DR$$

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

**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 11 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)

$$dY, \quad dT, \quad dZ, \quad dP, \quad dD$$

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

$$YR, \quad TR, \quad ZR, \quad PR, \quad DR$$

(with  $DR * BORO$  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 11 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

$$D_I = DR + PD * T_I$$

while  $T_I$  is in the range

$$0, \quad \pm PT, \quad \pm 2 * PT, \quad \dots, \quad \pm IT/2 * PT$$

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)

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

where  $\alpha, \beta$  are the ellipse parameters and  $\varepsilon/\pi$  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.

### 6.2.3 OBJETA : Object from Monte-Carlo simulation of decay reaction [32]

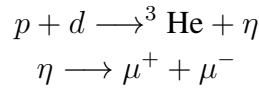
This generator simulates the reactions

$$M_1 + M_2 \longrightarrow M_3 + M_4$$

and then

$$M_4 \longrightarrow M_5 + M_6$$

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 Declaring 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 structure (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 in **zgoubi** input data list). 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* data list.

*SYSTEM* as well, may appear anywhere in the data list.

### 6.3.1 BINARY : *BINARY* ↔ *FORMATTED* data converter

This procedure translates field map data files from “BINARY” to “FORMATTED” – in the *FORTRAN* 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 *FORTRAN READ* statement :

```
READ (unit=ln, *) (X7(I), I=1, NCOL)
```

If *J* = 1, *NCOL* should be consistent with the following *FORTRAN 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 *FORTRAN* file ‘binary.f’.

### 6.3.2 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.

In some cases, these keywords may cause some information to be printed in *zgoubi.res*, for instance when the keyword *PICKUPS* is used.

### 6.3.3 ERRORS : Injecting errors in optical elements

#### *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, functionalities at the moment are the following :

*MULTIPOL* can be injected multipole field defects on any one of its dipole to 20-pole component. The defect can be relative (to the current field value) or absolute (adds to the current field value).

*TOSCA* 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 same defect series is applied at each pass, by default. However one may want to simulate one turn by repeating a cell subset, using *REBELOTE*, in which case defects of random type in the subset should be changed at each pass. This requires modification in `errors.f` program.

### 6.3.4 FIT, FIT2 : Fitting procedure

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 [33]. *FIT2* is a simplex method (Nelder-Mead method), it has been implemented in 2007 [34]. One or the other may converge faster, or may have some advantages/disadvantages, depending on the problem.

Any physical parameter of any element in the *zgoubi.dat* data list 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 transfer 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 [35].

A full list of the constraints available is given in the table page 238.

*FIT*, *FIT2* are compatible with the use of (*i.e.*, can be encompassed in) *REBELOTE* for successive fitting trials using various sets of parameters (option  $K = 22$  in *REBELOTE*).

### 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 structure

*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)

followed by, either

*DV* = allowed relative range of variation of the physical parameter *IP*

or

$[Vmin, Vmax]$  = 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 :

Input data	Numbering for FIT
'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.

Input data	Numbering for FIT
'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
Input data	Numbering for FIT
'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*.

Input data	Numbering for FIT
'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	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	Numbering for FIT
'DIPOLE-M'	
NFACE, IC, IL	1, 2, 3
IAMAX, IRMAX	4, 5
B <sub>0</sub> , N, B, G	6, 7, 8, 9
AT, ACENT, RM, RMIN, RMAX	10, 11, 12, 13, 14
$\lambda$ , $\xi$	15, 16
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> , shift	17, 18, 19, 20, 21, 22, 23, 24
$\omega$ , $\theta$ , R <sub>1</sub> , U <sub>1</sub> , U <sub>2</sub> , R <sub>2</sub>	25, 26, 27, 28, 29, 30
etc.	

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="text" value="1"/> <input type="text" value="9"/>	1 2	Non relevant
AGSMM *AF *BF		Keywords concerned, their labels
-1 3 12 <input type="text" value="1."/> 13 <input type="text" value="1."/> 14 <input type="text" value="1."/>	3 4 5	dB1, dB2, dB3 parameters in AGSMM
<input type="text" value="7.2135"/>	6	Field factor
<input type="text" value="1"/>	7	Timing
AGSMM *AD *BD		
-1 3 12 <input type="text" value="1."/> 13 <input type="text" value="1."/> 14 <input type="text" value="1."/>	8 9 10	
<input type="text" value="7.2135"/>	11	
<input type="text" value="1"/>	12	
AGSMM *CF		
-1 3 12 <input type="text" value="1."/> 13 <input type="text" value="1."/> 14 <input type="text" value="1."/>	13 14 15	
<input type="text" value="7.2135"/>	16	
<input type="text" value="1"/>	17	
AGSQUAD QH_*		
3		
<input type="text" value="0.605"/> <input type="text" value="0.77"/> <input type="text" value="0.879"/>	18 19 20	Field factor
<input type="text" value="1"/> <input type="text" value="2000"/> <input type="text" value="10000"/>	21 22 23	Timing
AGSQUAD QV_*		
3		
<input type="text" value="0.587"/> <input type="text" value="0.83"/> <input type="text" value="0.83"/>	24 25 26	
<input type="text" value="1"/> <input type="text" value="2000"/> <input type="text" value="10000"/>	27 28 29	
MULTIPOL		General rule. Precedes labeled cases if any. (meaningless)
<input type="text" value="-1"/>	30	
<input type="text" value="0.72135154291"/>	31	
<input type="text" value="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="text" value="0.1"/> <input type="text" value="0.3"/>	33 34	Field factor
<input type="text" value="1"/> <input type="text" value="10"/>	35 36	Timing

### Coupled Variables ( $XC$ )

Coupling a variable parameter to any other parameter in the structure is possible. This is done by giving  $XC$  a value of the form  $int.ijk$  where the integer part “int” is the number of the coupled element in the structure (equivalent to  $IR$ , see above), and the decimal part “ijk” is the number of its parameter of concern (equivalent to  $IP$ , see above) (if the parameter number is in the range 1, 2, ..., 9 (resp. 10, 11, ... 19, and 100, ...), then  $ijk$  must take the form  $00k$  (resp.  $0jk$ ,  $ijk$ )). For example,  $XC = 20.010$  is a request for coupling with the parameter number 10 of element number 20 of the structure, while  $XC = 20.100$  is a request for coupling with the parameter number 100 of element 20.

An element of the structure 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 be redundant information).

$XC$  can be either positive or negative. If  $XC > 0$ , then the coupled parameter will be given the same value as the variable parameter (for example, symmetric quadrupoles in a lens triplet will be given the same field). If  $XC < 0$ , then the coupled parameter will be given a variation opposite to that of the variable, so that the sum of the two parameters stays constant (for example, an optical element can be shifted while preserving the length of the structure, by coupling together its upstream and downstream drift spaces).

### 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)  $[v_{min}, v_{max}]$  : 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 238 ; adding or changing a constraint resorts to the *FORTRAN* file `ff.f`.

Each constraint is defined by the following list of data :

$IC[.IC2]$	=	type of constraint (see table p. 238).
$I, J$	=	constraint ( <i>i.e.</i> , $R_{ij}$ , determinant, tune ; $T_{ijk}$ ; $\sigma_{ij}$ ; trajectory # $I$ and coordinate # $J$ )
$IR$	=	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 preceeding ' $FIT[2]$ '
$V$	=	desired value of the constraint
$W$	=	weight of the constraint (smaller $W$ for higher weight)
$NP$	=	$NP$ additional parameters, values $PR_{1-NP}$ , 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 structure and identifying the first order transfer 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.

Type of constraint	Parameters defining the constraints						Recommended [MC]OBJET ; comments
	IC	I	J	Constraint	Additional parameter(s) NP	Param. values, $pr_1 - pr_{NP}$	
Transported $\sigma$ -matrix ( $\sigma(s) = T\sigma(0)\bar{T}$ )	0	1-6	1-6	$\sigma_{IJ}$ ( $\sigma_{11} = \beta_Y, \sigma_{21}, 12 = \alpha_Y$ , etc.)			OBJET/KOBJ=5.1
Periodic $\sigma$ -matrix ( $\sigma = I \cos \mu + J \sin \mu$ ), tunes ( $N = 1 - 9$ for MATRIX block 1-9)	0.N ( $N \leq 9$ )	1-6 7 8 9 10	1-6 any any any any	$\sigma_{IJ}$ ( $\sigma_{11} = \cos \mu_Y + \alpha_Y \sin \mu_Y$ , 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 $N \geq 2$
First order transport coeffs. & determinants. Chromas.	1  1.1	1-6 7 8 7 8	1-6 j j 7 8	Transport coeff. $R_{IJ}$ $j \neq 8$ : YY-determinant ; j=8: YZ-det. $j \neq 7$ : ZZ-determinant ; j=7: ZY-det. $dQ_Y / dp/p$ $dQ_Z / dp/p$			OBJET/KOBJ=5  OBJET/KOBJ=5.2
Second order transport coeffs.	2	1-6	11-66	Transport coeff. $T_{I,j,k}$ ( $j = [J/10], k = J - 10[J/10]$ )			OBJET/KOBJ=6
Trajectory coordinates (I = particle number; J=1-7 for D,Y,T,Z,P,S,time)	3  3.1 3.2 3.4 3.5	1-IMAX -1 -2 -3 -4 1-IMAX 1-IMAX 1-IMAX 1-IMAX	1-7 1-7 1-7 1-7 1-7 1-7 1-7 1-7	$F(J, I)$ $< F(J, i) >_{i=I1, I2}$ $Sup( F(J, i) )_{i=1, IMAX}$ $Dist [F(J, I)_{i=I1, I2}]$ $Dist [PU_i, i = 1, N]$ $ F(J, I) - FO(J, I) $ $ F(J, I) + FO(J, I) $ $ F(J, I) - F(J, K) $ $(F(J, I) - F(J, K))/F(J, K)$	$\begin{Bmatrix} 0 \\ 2 \end{Bmatrix}$  3 2  1 1	$I_1, I_2$  $I_1, I_2, \Delta I$ $NOEL_A, NOEL_B$  K K	[MC]OBJET $1 \rightarrow IMAX$ $1 \leq I_1 \leq I_2 \leq IMAX$  $1 \leq I_1 \leq I_2 \leq IMAX$ PU range  $K \leq IMAX$ $K \leq IMAX$
Ellipse parameters	4	1-6	1-6	$\sigma_{IJ}$ ( $\sigma_{11} = \beta_Y$ , $\sigma_{12} = \sigma_{21} = \alpha_Y$ , etc.)			OBJET/KOBJ=8 ; MCOBJET/KOBJ=3
Number of particles	5	-1 1-3 4-6	any any any	$N_{survived}/IMAX$ $N_{in \epsilon_{Y,Z,X}}/N_{survived}$ $N_{in \text{ best } \epsilon_{Y,Z,X,rms}}/N_{survived}$	1	$\epsilon_{Y,Z,X}/\pi$	OBJET MCOBJET MCOBJET
Coordinates & fields, across optical elements (J=1, 2, 3 for $B_X, Y, Z$ )	7.1 7.2 7.3 7.6 7.7 7.8 7.9	1-IMAX 1-IMAX 1-IMAX 1-IMAX 1-IMAX 1-IMAX 1-IMAX	1-7 1-7 1-7 1-3 1-3 1-3 1-3	min. or max. of $F(J, I)$ $\max(F(J, I)) - \min(F(J, I))$ $\min F(J, I) + \max(F(J, I))$ min. or max. value of $B_J$ $\max(B_J) - \min(B_J)$ $\min(B_J) + \max(B_J)$ $\int B_J ds$	1   1	1 or 2   1 or 2	[MC]OBJET
Spin	10 10.1 10.2 10.3	1-IMAX 1-IMAX 1-IDMAX 1-IDMAX	1-4 1-3 any 1-3	$S_{X,Y,Z}(I),  \vec{S}(I) $ $ S_{X,Y,Z}(I) - SO_{X,Y,Z}(I) $ spin rotation angle (rad) rotation axis, X-, Y- or Z-component			SPNTRK+ [MC]OBJET OBJET/KOBJ=2 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.

**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,time) over  $(I2 - I1 + 1)$  particles in the range  $i = I1 - I2$ , get it at  $N$  different pickup locations  $PU_1, PU_2, \dots, PU_N$ . The constraint is, minimizing the distance between these averages,  $Dist[<F(J, i)>_{i=I1, I2}]_{PU_1-N}$  (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 data list 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 given  $I$ -type (for respectively  $Y, Z, D$ ) phase-space emittance value. Then,  $NP=1$ , followed by the emittance value. The center and shape of the ellipse are determined by a matching to the position and shape of the particle distribution.

$I = 4, 5, 6$ , same as previous case, except for the ellipse, taken to be the *rms* matched ellipse to the distribution. Thus  $NP=0$ .

$IC=7$  *series* : The quantity to be constrained is taken *inside* the optical element with number *IR* (see page 67).

$IC=7.1$  : 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 ( $PR = 1$ ) or maximum ( $PR = 2$ ) 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  $I$ .

$IC=7.8$  : 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.

$IC=10$  : 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,  $X, Y, Z$ .

## OBJECT DEFINITION

Depending on the type of constraint (see table p. 238), 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 [33]. 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).

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. [34].

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) in *zgoubi.dat* data list. More on that can be found under *REBELOTE*, Sec. 6.3.12, page 80.

An example is given page 239.

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).

## 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]* data list, then that final run does occur. This is necessary in particular when it is desired to store resulting particle data, using *FAISTORE*, at the sole very final pass.
- '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.

An example using all five options, "nofinal", "save [FileName]", "penalty" and "ITER", is given in page 240.

**6.3.5 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.*

### 6.3.6 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 70). For that final run *GETFITVAL* will be inhibited so avoid overriding the updated variable values.



### 6.3.7 MCDESINT : Monte-Carlo simulation of in-flight decay[36]

When *MCDESINT* is met in a structure (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 average distance  $s_1$  at which a particle will decay is related to its center of mass lifetime  $\tau_1$  by

$$s_1 = c\tau_1 \sqrt{\gamma_1^2 - 1}$$

The actual path length  $s$  up to the decay point is calculated, for each one of the particles defined by *[MC]OBJET* and prior ray-tracing, from a random number  $0 < R_1 \leq 1$  by using the exponential decay formula

$$s = -s_1 \ln R_1$$

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  $0 < R_3 \leq 1$  by

$$\begin{aligned} \theta^* &= \arccos(1 - 2R_2) & (0 < \theta^* \leq \pi) \\ \phi &= 2\pi R_3 & (0 < \phi \leq 2\pi) \end{aligned}$$

$\phi$  is a relativistic invariant, and  $\theta$  in the laboratory frame (Fig. 8) 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 actual path length of the particle,  $F(6, I)$ , is compared to its limit path length  $s$ . If  $s$  is passed, then the particle is considered as having decayed at  $F(6, I) - \frac{XPAS}{2}$ , 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.

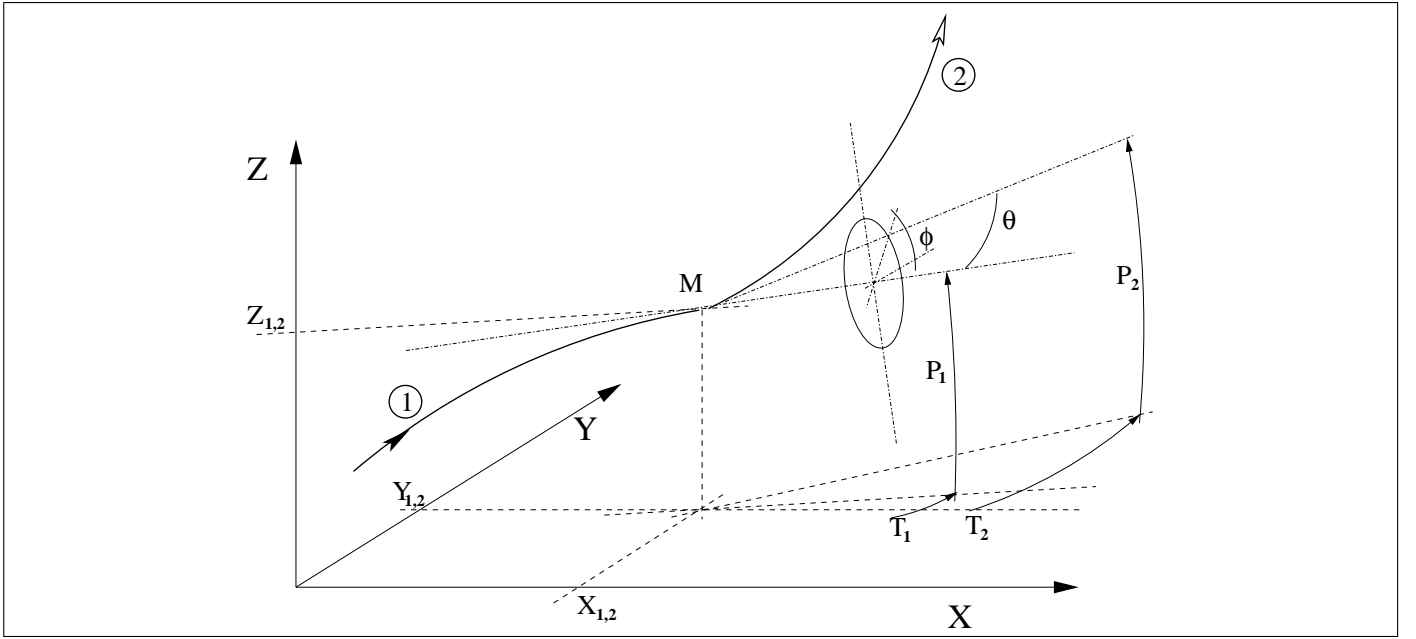


Figure 8: At position  $M(X_1, Y_1, Z_1)$ , particle 1 decays into 2 and 3 ; **zgoubi** then proceeds with the computation of the trajectory of 2, while 3 is discarded.  
 $\theta$  and  $\phi$  are the scattering angles of particle 2 relative to the direction of the incoming particle 1 ; they transform to  $T_2$  and  $P_2$  in **zgoubi** frame.

In *ESL* and *CHANGREF*,  $F(6, I)$  is compared to  $s$  at the end of the element. If the decay occurs inside the element, the particle is considered as having decayed at its actual limit path length  $s$ , thus its coordinates at  $s$  are recalculated by translation.

The limit path length of all particles ( $I = 1, I_{MAX}$ ) is 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 and correct. For instance, negative drifts may occur in a structure for some of the particles when using *CHANGREF* (due 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, after the same path length. 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*.

### 6.3.8 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 11 as defined by *OBJET*, to be the reference particle for computation of the transport coefficients.

Note : In the case of a periodic structure, 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 structure,
- otherwise, printing out at all keyword featuring *LABEL*  $\equiv$  *label* as a first label (see section 6.6.6, page 185, 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*.

### 6.3.9 OPTIONS : Global or special options

*OPTIONS* allows on/off switching of various options.

Available, for now :

- “WRITE ON or OFF”

Role : inhibit (most of) write statements to *zgoubi.res*.

Interest : “WRITE OFF” may allow substantial savings on CPU time. “WRITE ON” sets back to normal (the ‘normal’ state of write outs may depend on the status of other commands, such as *REBELOTE*, on-going *FIT[2]*, etc.).

- “CONSTY ON or 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, as follows.

Concurrent use of with  $IL = 1, 2, \text{ or } 7$  (cf. Sec. 6.6.5, p. 184) 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 of 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 *DIPOLLES*, *FFAG*.

Initial coordinates are set in a very regular manner using *OBJET*. The *Y* coordinate will stand for the radius value in the case of optical elements defined in polar coordinates (e.g., *DIPOLLES*, polar field maps, etc.).

### 6.3.10 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. 21, eq. 1.2.15 p. 23,) 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 *IORDR* in field map declarations (*DIPOLE-M*, *TOSCA*, etc.).

### 6.3.11 PARTICUL : Particle characteristics

Since **zgoubi** works using the rigidity, (*BORO*, as declared in *[MC]OBJET*), *PARTICUL* only needs be introduced (normally, following *[MC]OBJET* in the input data file *zgoubi.dat*) when the definition of some characteristics of the particles (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**, since it only knows the rigidity, will skip the computation of such quantities as time of flight.

### 6.3.12 REBELOTE : 'Do it again'

When *REBELOTE* is encountered in the input data file, the code execution 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 *LABEL*.

Then *NPASS-1* passes (from *LABEL* to *REBELOTE*) follow.

As to the last pass, number *NPASS+1*, there are two possibilities :

- either it also encompasses the whole *LABEL* 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 *LABEL* placed between the first above mentioned *LABEL*, 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.

The two functionalities of *REBELOTE* are the following :

- *REBELOTE* can be used for Monte Carlo simulations when more than  $\text{Max}(IMAX)$  particles are to be tracked. Thus, when the following random procedures are used : *MCOBJET*, *OBJETA*, *MCDESINT*, *SPNTRK* ( $KSO = 5$ ), their random seeds are not reset and independent statistics will add up.

This includes **Monte Carlo simulations**, in beam lines : normally  $K = 0$ . *NPASS* runs through the same structure, from *MCOBJET* to *REBELOTE* will follow, resulting in the calculation of  $(1 + NPASS) * IMAX$  trajectories, with as many random initial coordinates.

- *REBELOTE* can be used for multi-turn ray-tracing in circular machines : normally  $K = 99$  in that case (or  $K = 99$ , see below). *NPASS* turns in the same structure 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-*LABEL* 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 *LABEL*), and an extraction line (its description will then follow *REBELOTE*), whereas the ring description extends from the first *LABEL* to *REBELOTE*, with possible extraction, at the last pass, at the location of the second *LABEL*, located between the first one and *REBELOTE*,

Note : Some *CAVITE* options cause a *reset-to-zero of individual particle path length*<sup>5</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.

In addition to what precedes, *REBELOTE* can change the value of arbitrary parameters in *zgoubi.dat* data list, using the forth argument *IOPT=1* (see page 275). *NPRM* tells the number of parameters to be changed. A series of *NPRM* lists of values, one list per parameter to be changed and each list with *NRBLT* data, tells the values to be taken by each one of these parameters, over the *NRBLT* passes of the *REBELOTE* process.

Output prints 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 commands printing pass number and some other information onto the video output, every  $10^{j-1}$  turns if  $j > 0$  ; output is switched off if  $j = 0$ .

*REBELOTE* also provides information : statistical calculations and related data regarding particle decay (*MCDESINT*), spin tracking (*SPNTRK*), stopped particles (*CHAMBR*, *COLLIMA*), etc.

<sup>5</sup>For the geek: the “f(6,I)=0.d0” statements in *cavite.f*.

**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.

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 the *REBELOTE* procedure is given page 276.

An example of *FIT* preceding *REBELOTE* is given page 239.

**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. 82, 85, an example of job resuming is given page 260.



**6.3.13 RESET : Reset counters and flags**

Resets counters involved in *CHAMBR*, *COLLIMA*, *HISTO* and *INTEG* procedures.

Switches off *CHAMBR*, *MCDESINT*, *SCALING* and *SPNTRK* options.

### 6.3.14 SCALING : Power supplies and R.F. function generator

*SCALING* acts as a function generator dedicated to varying fields in optical elements, potentials in electrostatic devices, possibly in correlation with RF parameters in *CAVITE*. It is normally intended to be declared right after the object definition.

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') can as well 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*.

The latter (*i.e.*, fields being correlated to *CAVITE*) holds in general, allowing as well the simulation of a recirculating linac (thus optical elements between cavities are scaled, following the rigidity increase), 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 *MULTIPOL*s in *zgoubi.dat* optical structure data file. Subsets can be selected by labeling keywords in the data file (section 6.6.6, page 185) and adding the corresponding *LABEL(s)* in the *SCALING* declarations (9 *LABEL*s maximum - 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. A linear interpolation will determine the value of the scaling factor between successive timings.

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) instance 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.

An example of data formatting for the simulation of an acceleration cycle in a circular machine is given in the following.

*SCALING*

1	5	Active. $NF = 5$ families of elements are concerned, as listed below :
<i>QUADRUPO QFA QFB</i>		- <i>QUADRUPO</i> 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
<i>MULTIPOL</i>		- 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
<i>MULTIPOL QDA QDB</i>		- <i>MULTIPOL</i> 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
<i>BEND</i>		- All <i>BEND</i> keywords, regardless of any <i>LABEL</i>
2		$NT = 2$ timings
18131.E-3	24176.E-3	As above
1	6379	
<i>CAVITE</i>		- Accelerating cavity
3		$NT = 3$ timings
1	1.22	The synchronous rigidity $(B\rho)_s$ increases,
1	1200	from $(B\rho)_{s_o}$ to $1.22 * (B\rho)_{s_o}$ from turn 1 to 1200, and
		from $1.22 * (B\rho)_{s_o}$ to $1.33352 (B\rho)_{s_o}$ 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

$$\begin{aligned} B(N) &= SCALE(N) B_0 \\ B_i(N) &= SCALE(N) B_{i0} \end{aligned}$$

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  $(B\rho)_{s_o}$  be the initial rigidity (namely,  $(B\rho)_{s_o} = BORO$  as defined in the keyword *OBJET* for instance). Then, at turn number  $N$ ,

$$\begin{aligned} \text{if } 1 \leq N \leq 1200 \text{ then, } SCALE(N) &= 1 + \frac{1.22 - 1}{1 + 1200 - 1} (N - 1) \\ \text{if } 1200 \leq N \leq 6379 \text{ then, } SCALE(N) &= 1.22 + \frac{1.33352 - 1.22}{1 + 6379 - 1200} (N - 1200) \end{aligned}$$

and then,

$$(B\rho)_s(N) = SCALE(N) \cdot (B\rho)_{s_o}$$

from which value the calculations of  $f_{RF}(N)$  follow.

$NT$  can take negative values, then acting as an option switch (rather than giving number of timings), 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,

$$\text{SCALE}(N) = \text{SCALE}(N - 1) * \frac{B\rho(N)}{B\rho(N - 1)}$$

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	

•  $NT = -2$  : this is convenient for reading an RF law for *CAVITE* from an external data file, including usage for acceleration in fixed field accelerators.

•  $NT = 1.10$  : 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 ' <i>COH1</i> ' as first label.
1.10 [ <i>FAC</i> ]	Scaling is read from a file, file name e.g. ' <i>scalingFile</i> '; <i>FAC</i> is an additional scaling factor.
<i>scalingFile</i>	File name (file is expected to contain at least $N2$ columns).
1 N	Column number $N2$ gives the scaling factor at rigidity values given 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.

*KEYWORD LABEL1*  
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 ' <i>COH1</i> ' as first label.
1.12	Scaling is read from a file, file name e.g. ' <i>scalingFile</i> '.
<i>scalingFile</i>	File name (file is expected to contain at least $2+2*N2$ columns).
$N1, N2, p11, p12,$	$N1$ =number of pairs, see next; $N2$ = ;
$p21, p22, ...$	Scaling is obtained from interpolation in column $pi1$ and applied to parameter $pi2$ of <i>AGSQUAD</i> .

•  $NT = 1.13$  : TO BE DOCUMENTED.

*KEYWORD LABEL1*  
1.13                      To be documented

## Notes :

1. In causing, via *CAVITE*, a change of the synchronous rigidity, *SCALING* causes a change of the reference rigidity, following (see *CAVITE*)

$$B\rho_{ref} = BORO \longrightarrow B\rho_{ref} = BORO + \delta B\rho_s$$

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.

### **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).

### 6.3.15 SPACECHARG : Space charge

The keyword *SPACECHARG* enables (or disables) the calculation of space charge effects. It must be preceded by *PARTICUL* for definition of mass and charge values, as they enter in the definition of space charge parameters.

*SPACECHARG* is supposed 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 a keyword.

*SPACECHARG* is supposed to appear a second time at the location where calculations should stop, with *LMNT* set to 'none'. It results in summary outputs in *zgoubi.res*.

Occurrence of *PRINT* amongst *SPACECHARG* arguments results in storage of computational data and other relevant informations in *zgoubi.SPACECHARG.out*, as space charge computation proceeds.

### 6.3.16 SPNTRK : Spin tracking

The keyword *SPNTRK* allows switching spin tracking on (index  $KSO=1$ ) or off ( $KSO=0$ ), or resuming (index  $KSO=-1$ , following an occurrence  $KSO=0$ ). It also permits the attribution of an initial spin to each one of the *IMAX* particles of the beam, following a distribution that depends on the option index  $KSO$ . 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*).

**$KSO = 5$  :** random generation of *IMAX* initial spin conditions as described in Fig. 9. Given a mean polarization axis ( $\vec{S}$ ) 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 ( $\vec{S}$ ) axis. Examples of simple distributions available by this mean are given in Fig. 10.

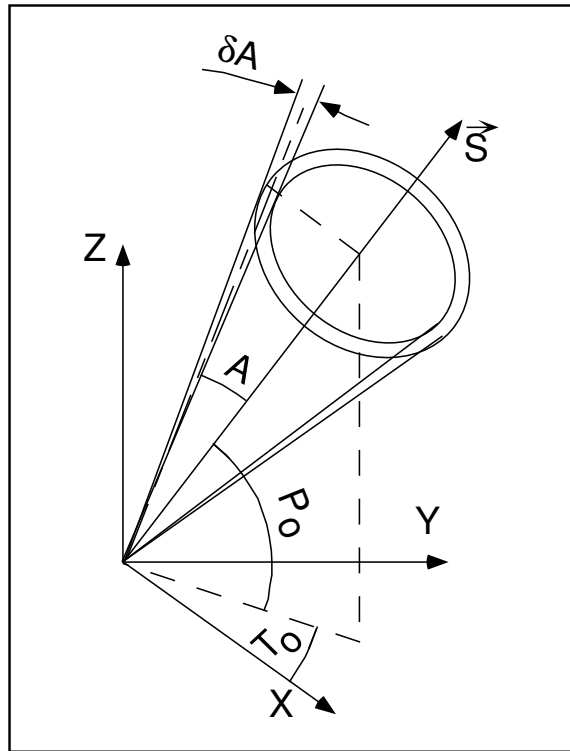


Figure 9: Spin distribution as obtained with option  $KSO = 5$ .

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  $P_0$ .

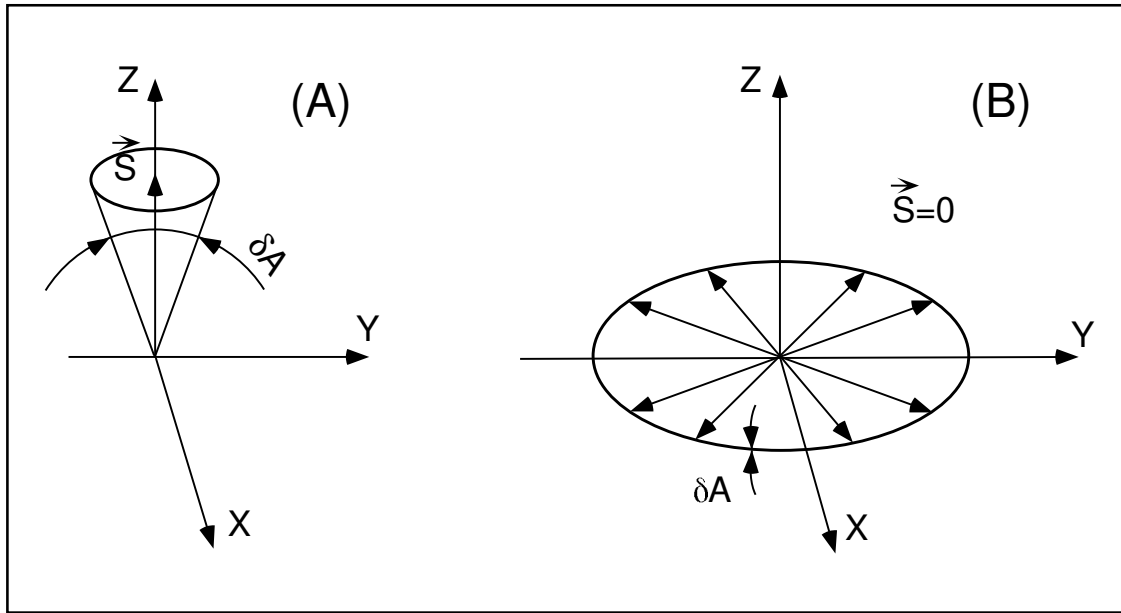


Figure 10: Examples of the use of  $KSO = 5$ .

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  $\delta A = 0$ .



**6.3.17 SRLOSS : Synchrotron radiation energy loss [20]**

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.

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.

*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*.

### 6.3.18 SVDOC : SVD orbit correction

#### UNDER DEVELOPMENT - Feb. 2019

The method is the following :

1. Reference trajectory (case of beam line) or orbit (case of periodic structure- requires *FIT*[2], see below) : 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 beam line, *FIT* would allow to determine the necessary initial coordinates (varied by *FIT*) for a particle with a given momentum to acquire some requested radial excursion as some requested location down the line (the constraint(s) in *FIT*); (ii) if the sequence is assumed periodic, *FIT* 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 in PART B, page 298.

2. The algorithm computes the  $A$  matrix (in  $A\vec{x} = \vec{y} - \vec{y}_0$ , with  $x$  the correctors signal,  $y$  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, or multiple-pass ERL, or FFAG lattice).

3. The inverse matrix  $A^{-1}$  is computed and stored (logged to *zgoubi.SVD.out*).

4. Optionally, *SVDOC* can carry on with error trials, in the following way:

- (i) apply the compensation signal (the SVD-inverted  $A^{-1}$  matrix) on the correctors,
- (ii) apply orbit defects,
- (iii) find the corrected [multiple-]orbit (logged to *zgoubi.SVDOrbit.out*),
- (iv) repeat (ii)-(iv).

The method is the following for instance in the case of a periodic sequence:

- use *ERRORS* to activate orbit defects. Note that *ERRORS* must be set to OFF in the *zgoubi* sequence, as (i) the first pass through the sequence is used to determine the  $\vec{y}_0$  reference orbit mentioned earlier (possibly using *FIT*[2], placed at the end of the sequence), (ii) the ON/OFF switch in *ERRORS* is managed by *SVDOC* in this procedure;

- *SVDOC* the strength in the correctors that the SVD used is reset to zero;

- *FIT*[2] finds the orbit resulting from these defects, this orbit is recorded at the PUs that the SVD used;

- *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.SVDOrbits.out*, together with the uncorrected orbit).

A typical way to use the orbit correction algorithm is the following:

```
Title. The 'OPTICAL SEQUENCE', below, is assumed periodic.
'OBJET'
1000.000000      !      reference rigidity (kG.cm)
2
1 1
0.E+00      0.E+00      0.E+00      0.E+00      0. 1. 'o'
1
! Start 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      ! 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      ! 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

'SVDOC'
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}      ! Name 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" data are changed prior to that, following 2 lines; 5 trials.
ERRORS 12      ! parameter 12 in ERRORS: value of HKIC dipole field: will set H orbit,
ERRORS 22      ! parameter 22 in ERRORS: value of VKIC 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'
```

*TO BE DOCUMENTED FURTHER*

### 6.3.19 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  $\vec{E}$  (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 6.6.11)

*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*.

### 6.3.20 **SYSTEM** : System call

The keyword *SYSTEM* allows one or a series of system calls. It can appear anywhere, an arbitrary number of times, in the *zgoubi.dat* data list. 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 of the use of *SYSTEM* are given in pp. 239, 292.



## 6.4 Optical Elements and Related Numerical Procedures

### AGSMM : AGS main magnet

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 27), 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. [37], and following in that the methods found in the MAD model of the AGS [38].
- 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 input data 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 is that no field value is required in defining the AGS main magnets in the *zgoubi.dat* input data list.

**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 27. 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 [38].



**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. 11A and 11B,

*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

$$B_Z = \mathcal{F}(R, \theta) * B_0 * \left( 1 + N * \left( \frac{R - RM}{RM} \right) + B * \left( \frac{R - RM}{RM} \right)^2 + G * \left( \frac{R - RM}{RM} \right)^3 \right) \quad (6.4.1)$$

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. 11A and 11B), 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. 12) given by

$$F = \frac{1}{2} \frac{(\lambda - s)^2}{\lambda^2 - \xi^2} \quad \text{if } \xi \leq s \leq \lambda \quad (6.4.2)$$

$$F = 1 - \frac{1}{2} \frac{(\lambda - s)^2}{\lambda^2 - \xi^2} \quad \text{if } -\lambda \leq s \leq -\xi \quad (6.4.3)$$

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

$$F = \frac{1}{2} + \frac{s}{\lambda + \xi} \quad \text{if } 0 \leq s \leq \xi \quad (6.4.4)$$

$$F = \frac{1}{2} - \frac{s}{\lambda + \xi} \quad \text{if } -\xi \leq s \leq 0 \quad (6.4.5)$$

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.



If  $\xi = -1$ ,  $F$  is an exponential type fringe field (Fig. 12) given by [39]

$$F = \frac{1}{1 + \exp P(s)} \quad (6.4.6)$$

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

$$P(s) = C_0 + C_1 \left(\frac{s}{\lambda}\right) + C_2 \left(\frac{s}{\lambda}\right)^2 + C_3 \left(\frac{s}{\lambda}\right)^3 + C_4 \left(\frac{s}{\lambda}\right)^4 + C_5 \left(\frac{s}{\lambda}\right)^5 \quad (6.4.7)$$

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  $s = \pm\lambda$ , so as not to perturb the 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. 6.4.1) is (Fig. 13)

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

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

$$\delta R = \frac{R_{MAX} - R_{MIN}}{IRMAX - 1}$$

and the angular step by

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

where,  $R_{MIN}$  and  $R_{MAX}$  are the lower and upper radial limits of the field map, and  $AT$  is its total angular aperture (Fig. 11B).  $IRMAX$  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

$$B_Z * \left( 1 + \frac{\Delta B_Z}{B_0} \frac{R - R_0}{R_{MAX} - R_{MIN}} \right)$$

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

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

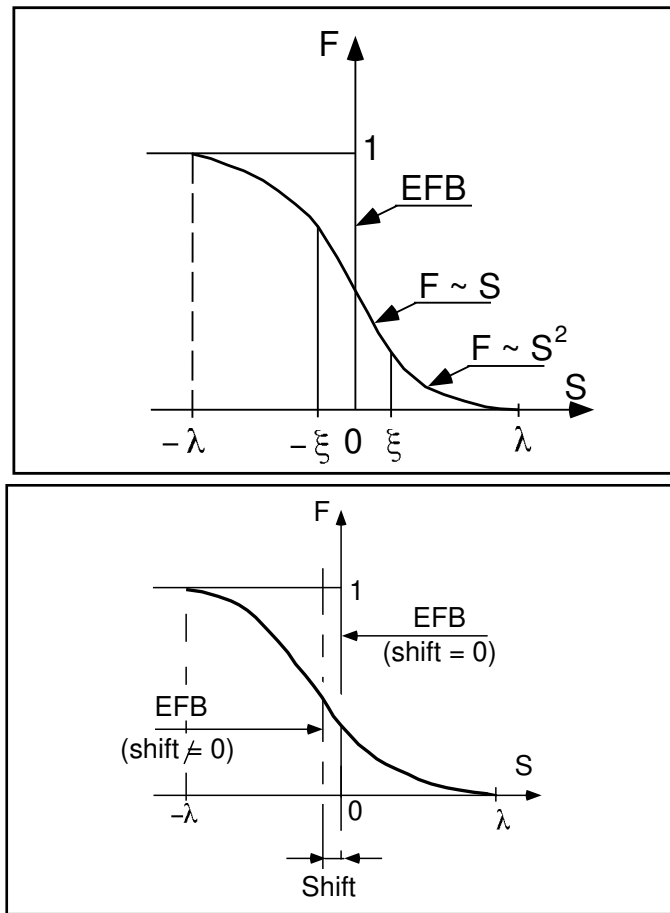


Figure 12: Second order type fringe field (upper plot) and exponential type fringe field (lower plot).

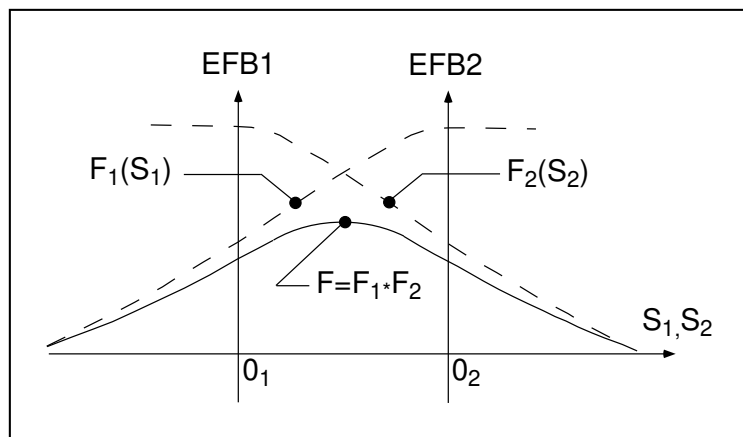


Figure 13: Effective value of  $\mathcal{F}(R, \theta)$  for overlapping fringe fields  $F_1$  and  $F_2$  centered at  $O_1$  and  $O_2$ .

If  $NBS \geq 1$ , then  $NBS$  shims are introduced at positions  $\frac{R_1 + R_2}{2}, \frac{\theta_1 + \theta_2}{2}$  (Fig. 14) [40]  
 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. 14,  $\rho = \frac{R_1 + R_2}{2}$  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

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

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  $IODRE$  ( $IODRE=2, 25$  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.

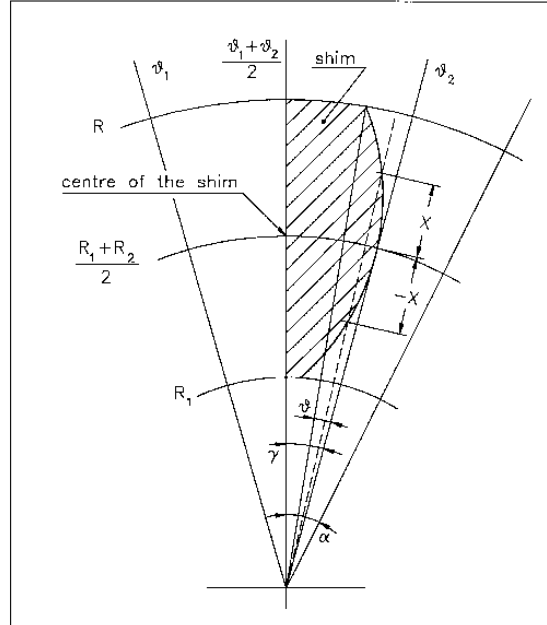


Figure 14: A second order profile shim. The shim is centered at  $\frac{(R_1 + R_2)}{2}$  and  $\frac{(\theta_1 + \theta_2)}{2}$ .

**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 [41], in the weak-strong approximation.

Upon option using *SPNTRK*, *BEAMBEAM* will include spin kicks, after modelling as described in Ref. [42].

**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

*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 6.6.8, p. 186).

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. 15. 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. 153) ; 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 6.6.8).

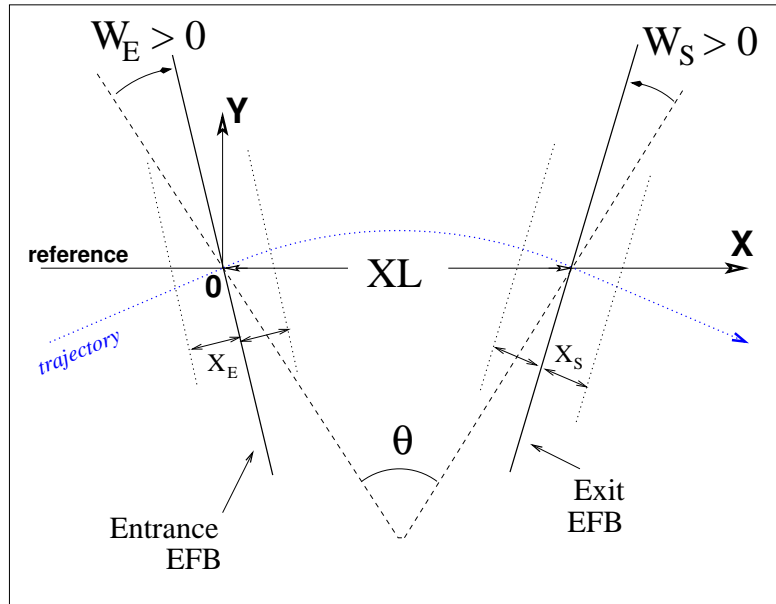


Figure 15: Geometry and parameters of *BEND* :  $XL$  = length,  $\theta$  = deviation,  $W_E, W_S$  are the entrance and exit wedge angles. The motion is computed in the Cartesian frame  $(O, X, Y, Z)$ .

**From matrice-style code modeling, to zgoubi :** Fig. 16 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).



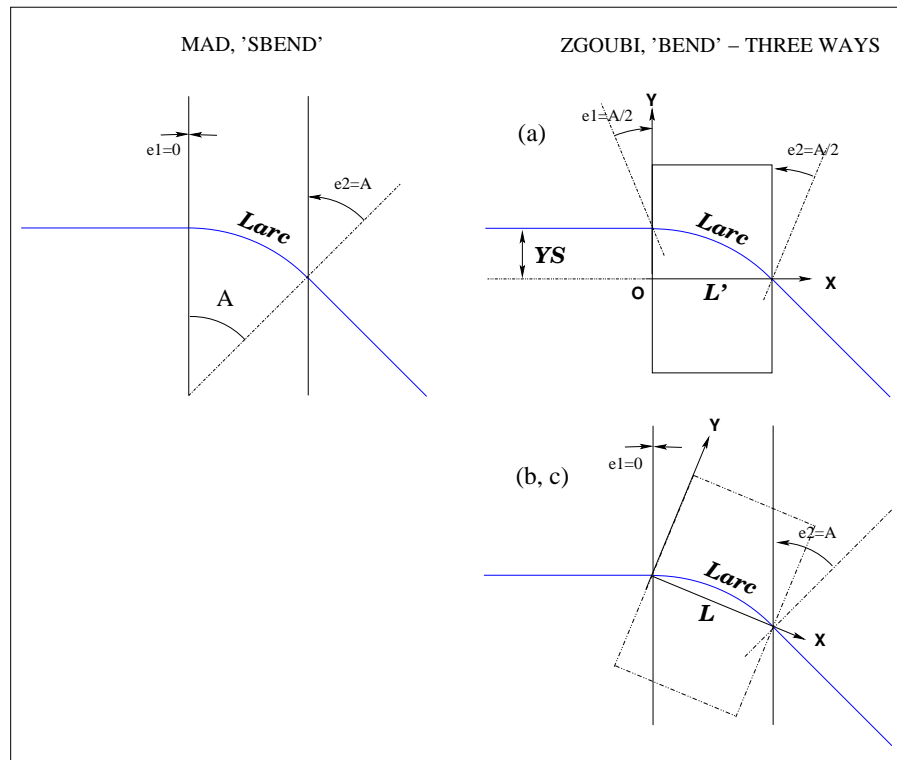


Figure 16: Simulating a matrix method style of rectangular magnet (e.g., MAD's 'SBEND') by means of *BEND*, in three different ways, see text. In case (a), the straight length in *zgoubi*'s *BEND* is  $L' = \rho \sin(A)$ , whereas in cases (b, c) it is  $L = 2\rho \sin(A/2)$ .

• 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)  $YS = -\rho \sin(A)$  translated, non-rotated *BEND*, followed by A Z-rotation of frame

```
'CHANGREF'
ZR -4.997916927068
'BEND'
1
99.8750260394 0. 1.
0. 0. 0.05 ! e1=A/2
0 0. 0. 0. 0. 0.
0. 0. 0.05 ! e2=A/2
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. 0.1 ! e2=A
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. 0.1 ! e2=A
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. 16) 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)
        ENDIF
      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.

### 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  $IORDRE$  ( $IORDRE = 2, 25$  or  $4$ , see section 1.4.2).

In case a particle exits the mesh, its  $IEX$  flag is set to  $-1$  (see section 6.6.11, p. 188), 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. 17).

**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.

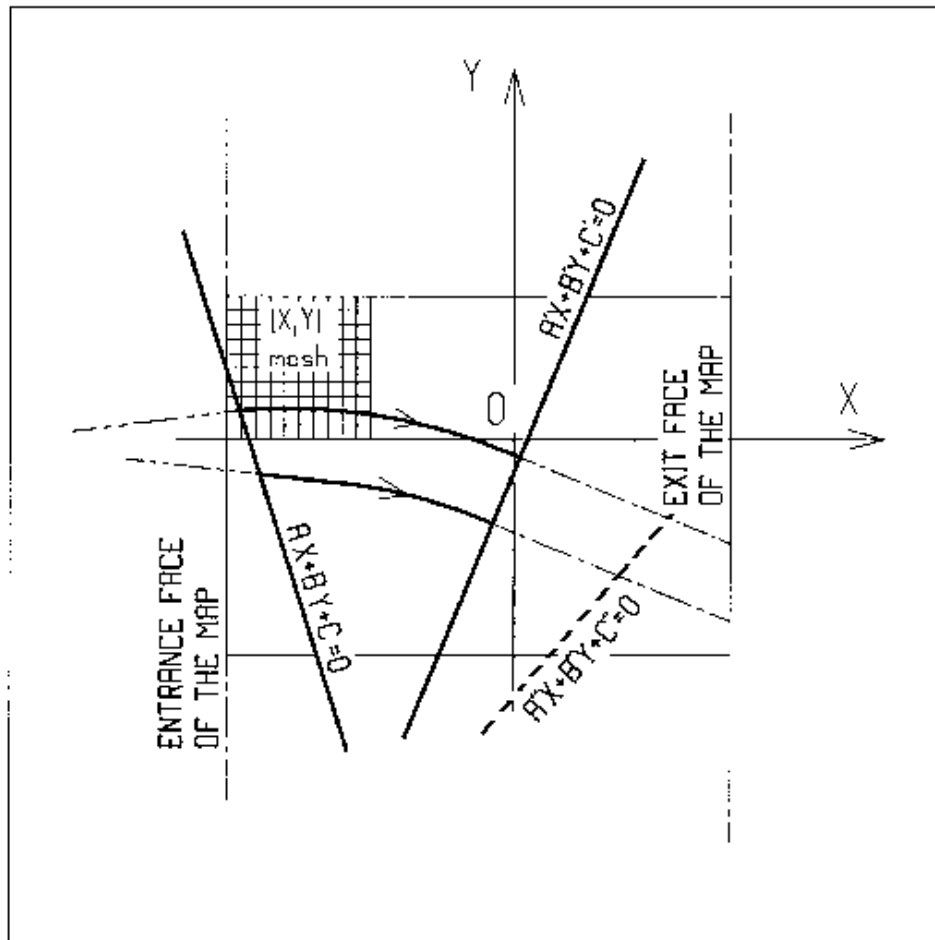


Figure 17:  $OXY$  is the coordinate system of the mesh. Integration boundaries may be defined, using  $ID \neq 0$ : particle coordinates are extrapolated linearly from the entrance face of the map, onto the boundary  $A'X + B'Y + C' = 0$ ; after ray-tracing inside the map and terminating on the boundary  $AX + BY + C = 0$ , 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

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 6.6.10). 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  $q$ .

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,

$$B\rho_{ref} = BORO \longrightarrow B\rho_{ref} = BORO + \delta B\rho_s \quad (6.4.8)$$

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 structure) 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 6.6.10. *CAVITE* may conveniently be located *at the end* of the optical structure, 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 R.F  
 $B\rho_s$  = 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

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

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

$$\begin{aligned}p_{XI} &= [p_I^2 - (p_{I_0}^2 - p_{XI_0}^2)]^{1/2} \\ p_{YI} &= p_{YI_0}, \quad \text{and} \quad p_{ZI} = p_{ZI_0} \quad (\text{longitudinal kick}) \\ X_I &= X_{I_0}, \quad Y_I = Y_{I_0} \quad \text{and} \quad Z_I = Z_{I_0} \quad (\text{zero length cavity})\end{aligned}$$

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. In that case  $\phi_s$  should be given the appropriate value for compensation for SR induced 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 [43]. 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 file (with name normally “zgoubi.freqLaw.In”). See routines `cavite.f` and `scaln.f` for details. Was first used for acceleration in scaling FFAG [51].
- **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 [52, 53].  
 $IOPT = 7$  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. 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,

$$B\rho_{ref} = BORO \rightarrow B\rho_{ref} = BORO + \delta B\rho_s \quad \text{with} \quad \delta B\rho_s = \frac{\Delta(\beta_s(W_s + M))}{qc}, \quad \Delta W_s = q\hat{V} \cos(\phi_s) \quad (6.4.9)$$

The cavity length  $L$  as specified in the input data is also used to update the path length of the particles (linearly). An option (‘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 = 1 : Chambers cavity with  $\Delta W/W$  approximation, namely, for both planes,

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{out} = \begin{pmatrix} \sqrt{(W_i/W_o)} & L_{cav} \times \sqrt{(W_i/W_o)} \\ 0 & \sqrt{(W_i/W_o)} \end{pmatrix} \begin{pmatrix} x \\ x' \end{pmatrix}_{in} \quad (6.4.10)$$

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 cavity without approximation on  $\Delta W/W$ , namely, for both planes,

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{out} = \begin{pmatrix} \cos(u) - \sqrt{2} \sin(u) \cos(\phi) & vW_i \sin(u) \cos(\phi) \\ -\frac{\sin(u)}{vW_o} (2 \cos(\phi) + \frac{1}{\cos(\phi)}) & \frac{W_i}{W_o} (\cos(u) + \sqrt{2} \sin(u) \cos(\phi)) \end{pmatrix} \begin{pmatrix} x \\ x' \end{pmatrix}_{in} \quad (6.4.11)$$

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 1, with matrix in both planes re-normalized to determinant=1.
- IOP = 0 : Transverse matrix is that of a drift with length  $L$ .
- **If  $IOPT = 11$  :** under development.



The same simulation of a synchrotron RF cavity as for **IOPT** = 2 is performed, but in addition **IOPT** = 10 handles synchrotron radiation energy loss, both stationary or accelerated bucket.

### **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 (hence it will be almost everywhere non-zero). One way to calculate it at the beginning of the structure (*i.e.*, where the initial particle coordinates are defined) is to ray-trace 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 entire closed orbit, and as a sub-product its length  $\mathcal{L}$  (the  $F(6, 1)$  coordinate in the *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 85. 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 82) and the examples there, as well as the example page 260.

**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 structure 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<sub>S</sub>*, 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. 29, and 11, p. 98). 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 6.6.11) 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 anywhere in a structure.

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

$$\begin{aligned} T_2 &= T_1 - ALE \\ Y_2 &= \frac{(Y_1 - YCE) \cos T_1 + XCE \sin T_1}{\cos T_2} \\ DL^2 &= (XCE - Y_2 \sin ALE)^2 + (YCE - Y_1 + Y_2 \cos ALE)^2 \\ Z_2 &= Z_1 + DL \tan P_1 \\ S_2 &= S_1 + \frac{DL}{\cos P_1} \\ P_2 &= P_1 \end{aligned}$$

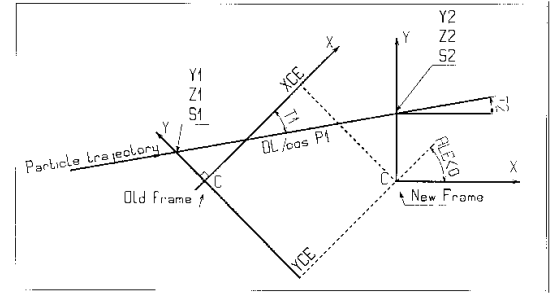


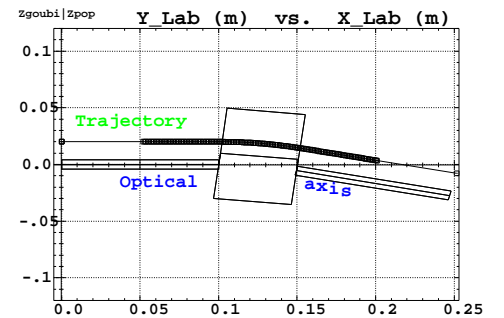
Figure 18: Scheme of the *CHANGREF* procedure.

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  $XCE - Y_2 \sin(ALE)$ .

The example below shows the use of *CHANGREF* for the 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
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
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 step-wise 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 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'

```

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

The reaction is  $1 + 2 \longrightarrow 3 + 4$  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. 19.

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

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

followed by

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

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

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

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

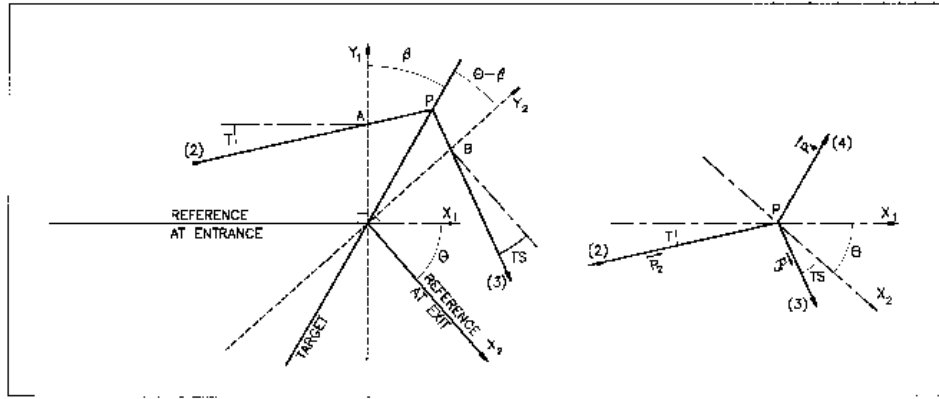


Figure 19: Scheme of the principles of *CIBLE (TARGET)*

$A, T$  = 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

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

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

#### Longitudinal Phase-space 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*).

#### Transverse Phase-space Collimation

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

$$\gamma_Y Y^2 + 2\alpha_Y Y T + \beta_Y T^2 \geq \epsilon_Y / \pi \quad \text{if } IFORM = 11 \text{ or } 14$$

$$\gamma_Z Z^2 + 2\alpha_Z Z P + \beta_Z P^2 \geq \epsilon_Z / \pi \quad \text{if } IFORM = 12 \text{ or } 15$$

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

When a particle is stopped, its index *IEX* (see *OBJET* and section 6.6.11) is set to the value -4, and its actual path length is stored in the array *SORT* for possible further use with *HISTO*).

**DECAPOLE : Decapole magnet (Fig. 20)**

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. 152.

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

$$B_X = 0$$

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

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

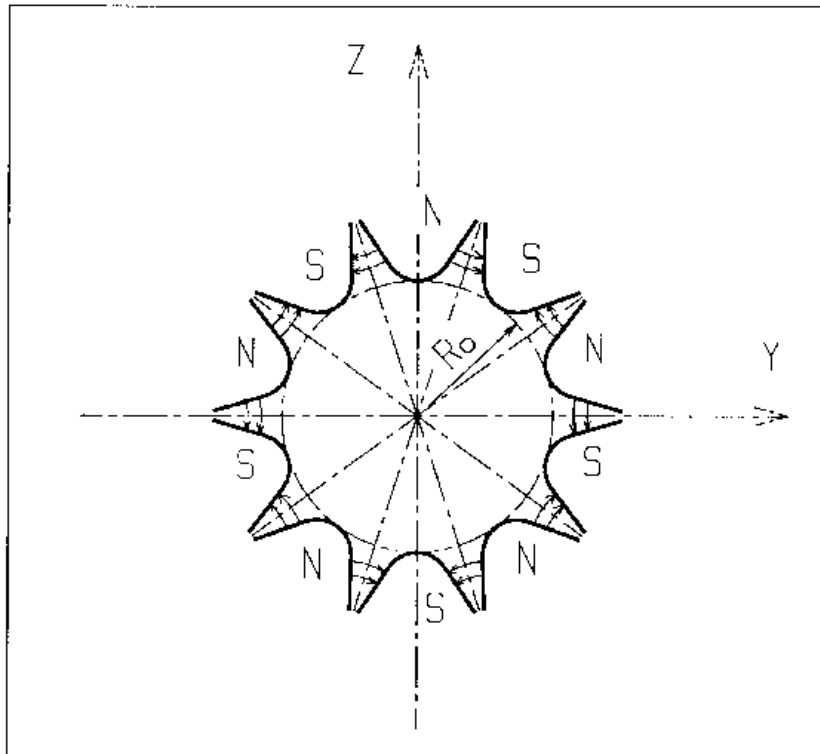


Figure 20: Decapole magnet

### DIPOLE : Dipole magnet, polar frame

*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 ; the essential difference in *DIPOLE* is in its skipping that intermediate stage of field map generation found in *DIPOLE-M* and *AIMANT*.

*DIPOLE* is available in a version, *DIPOLÉS*, that allows overlapping of fringe fields in a configuration of neighboring magnets.

The dimensioning of the magnet is defined by (Fig. 11, p. 98)

$AT$  : 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

$$B_Z(R, \theta) = \mathcal{F}(R, \theta) * B_0 * \left( 1 + N * \left( \frac{R - RM}{RM} \right) + B * \left( \frac{R - RM}{RM} \right)^2 + G * \left( \frac{R - RM}{RM} \right)^3 \right) \quad (6.4.12)$$

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,  $\lambda$  (normally equal to the gap size), is associated 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).

$F$  is an exponential type fringe field (Fig. 12, p. 100) given by [39]

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

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 coefficient attached to the entrance (respectively exit, lateral) EFB. At any position on a trajectory the resulting value of the fringe field coefficient (eq. 6.4.12) is

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

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

### Calculation of the Mid-plane Field and Derivatives

$B_Z(R, \theta)$  in Eq. 6.4.12 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.

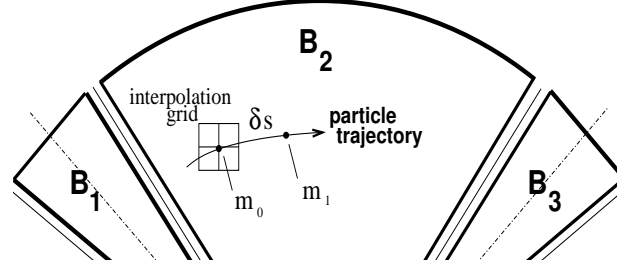


Figure 21: Interpolation method.  $m_0$  and  $m_1$  are the projections in the median plane of particle positions  $M_0$  and  $M_1$  and separated by  $\delta s$ , projection of the integration step.

### 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

*DIPOLE-M* is a more recent, 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. 11A and 11B.

*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

$$B_Z(R, \theta) = \mathcal{F}(R, \theta) * B_0 * \left( 1 + N * \left( \frac{R - RM}{RM} \right) + B * \left( \frac{R - RM}{RM} \right)^2 + G * \left( \frac{R - RM}{RM} \right)^3 \right) \quad (6.4.13)$$

where  $N$ ,  $B$  and  $G$  are respectively the first, second and third order field indices and  $\mathcal{F}$  is the fringe field coefficient.

### Calculation of the Fringe Field Coefficient

With each EFB a realistic extent of the fringe field,  $\lambda$  (normally equal to the gap size), is associated 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).

$F$  is an exponential type fringe field (Fig. 12) given by [39]

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

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 coefficient attached to the entrance (respectively exit, lateral) EFB. At any node of the map mesh, the resulting value of the fringe field coefficient (eq. 6.4.13) is

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

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. 11B).  $IRMAX$  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

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

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

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

If  $NBS \geq 1$ , then  $NBS$  shims are introduced at positions  $\frac{R_1 + R_2}{2}, \frac{\theta_1 + \theta_2}{2}$  (Fig. 14) [40]  
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

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

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 [44, 45]

*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>6</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. 6.4.14, 6.4.16) 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. 11-A, p. 98, 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,

$$(i) \quad B_{Zi}(R, \theta) = B_{Z0,i} \mathcal{F}_i(R, \theta) \left( 1 + b_{1i}(R - RM_i)/RM_i + b_{2i}(R - RM_i)^2/RM_i^2 + \dots \right) \quad (6.4.14)$$

$$(ii) \quad B_Z(R, \theta) = B_{Z0,i} + \sum_{i=1}^N \mathcal{F}_i(R, \theta) \left( b_{1i}(R - RM_i) + b_{2i}(R - RM_i)^2 + \dots \right) \quad (6.4.15)$$

$$(iii) \quad \text{Sandro Ruggiero's FFAG style of field, experimental, see routine dipsi.f}$$

wherein  $B_{Z0,i}$  is a reference field, at reference radius  $RM_i$ , and  $\mathcal{F}(R, \theta)$  is the fringe field coefficient, see below. This field model is proper to simulate for instance chicane dipoles, cyclotron or FFAG magnets, etc.

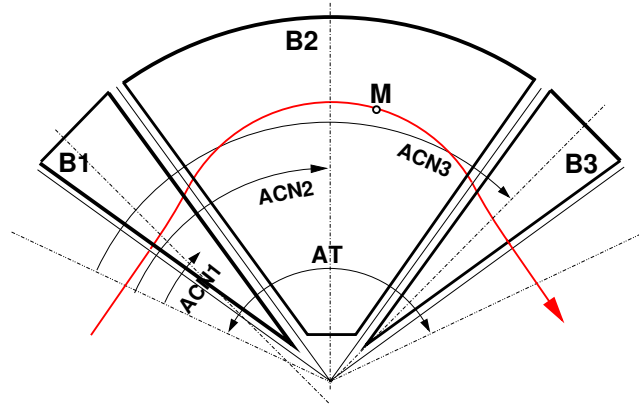


Figure 22: Definition of a dipole triplet using the *DIPOLLES* or *FFAG* procedures.

### Calculation of the Fringe Field Coefficient

In a dipole, a realistic extent of the fringe field,  $g$ , is associated with each EFB, and a fringe field coefficient  $F$  is calculated.

$F$  is an exponential type fringe field (Fig. 12, page 100) given by [39]

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

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

<sup>6</sup>FFAG can be referred to as another instance of a procedure based on such method.

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

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 coefficient attached to the entrance (respectively exit) EFB ; at any position on a trajectory the resulting value of the fringe field coefficient is taken to be

$$\mathcal{F}_i(R, \theta) = F_E * F_S \quad (6.4.16)$$

### 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

$$B_Z(R, \theta) = \sum_{i=1, N} B_{Zi}(R, \theta) \quad (6.4.17)$$

$$\frac{\partial^{k+l} \vec{B}_Z(R, \theta)}{\partial \theta^k \partial r^l} = \sum_{i=1, N} \frac{\partial^{k+l} \vec{B}_{Zi}(R, \theta)}{\partial \theta^k \partial r^l} \quad (6.4.18)$$

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. 6.4.17), 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 [46] :*

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. 6.4.17 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

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

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.

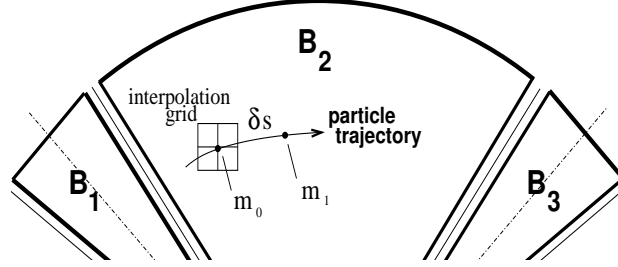


Figure 23: Interpolation method.  $m_0$  and  $m_1$  are the projections in the median plane of particle positions  $M_0$  and  $M_1$  and separated by  $\delta s$ , projection of the integration step.

### 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. 11,  $\omega^+ = ACENT$  (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  $Z$

$$V(X, Y, Z) = G(X) \left( Y^4 - \frac{10}{3} Y^2 Z^2 + Z^4 \right) Y Z$$

with  $G_0 = \frac{B_0}{R_0^5}$

The modelling of the fringe field form factor  $G(X)$  is described under *QUADRUPO*, p. 152.

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

$$\begin{aligned} B_X &= 0 \\ B_Y &= G_0(5Y^4 - 10Y^2Z^2 + Z^4)Z \\ B_Z &= G_0(Y^4 - 10Y^2Z^2 + 5Z^4)Y \end{aligned}$$

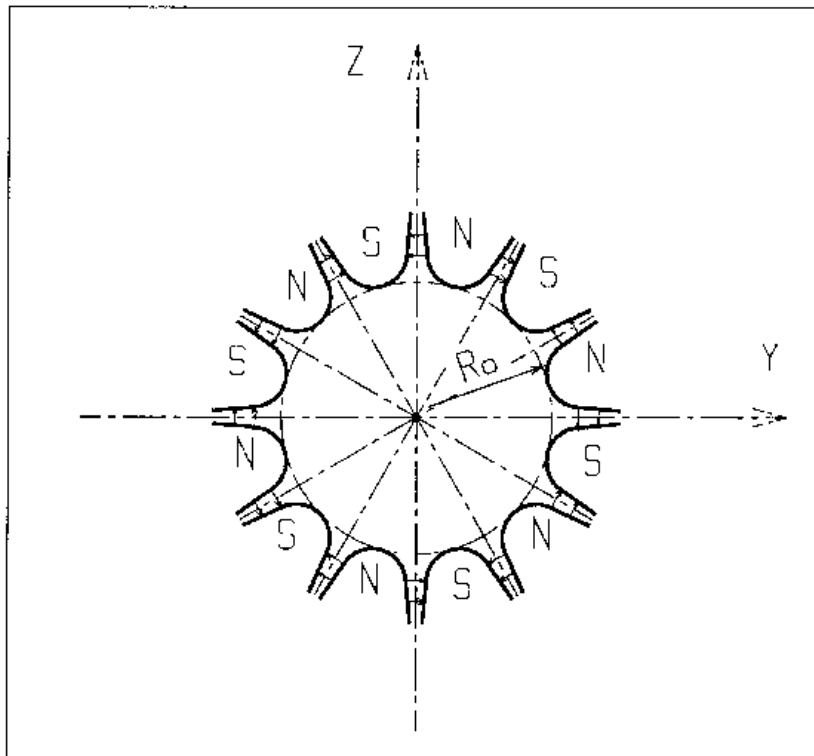


Figure 24: Dodecapole magnet

**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 a structure. This is an exact transport: the associated equations of motion are (Fig. 25)

$$Y_2 = Y_1 + XL * \operatorname{tg} T$$

$$Z_2 = Z_1 + \frac{XL}{\cos T} \operatorname{tg} P$$

$$SAR_2 = SAR_1 + \frac{XL}{\cos T * \cos P}$$

whereas the velocity vector is left unchanged.

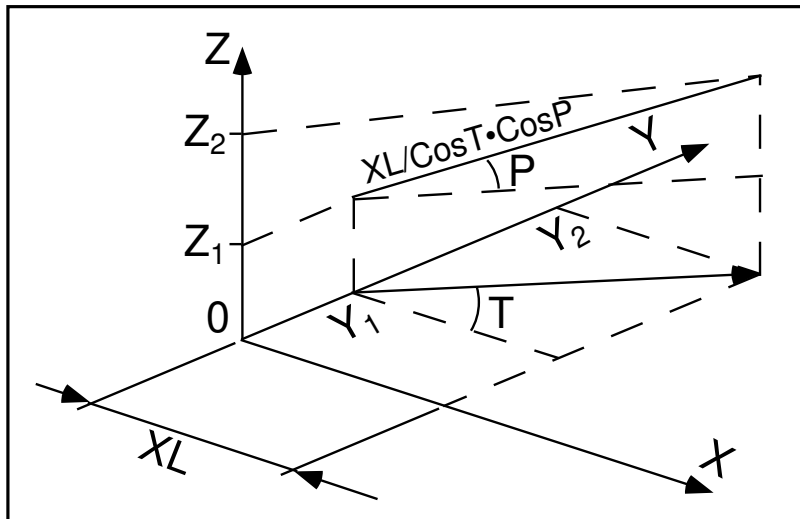


Figure 25: Transfer of particles in a drift space.

The flag '.plt' can be added as a 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 useful 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 after step.



**EBMULT : Electro-magnetic multipole**

*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.

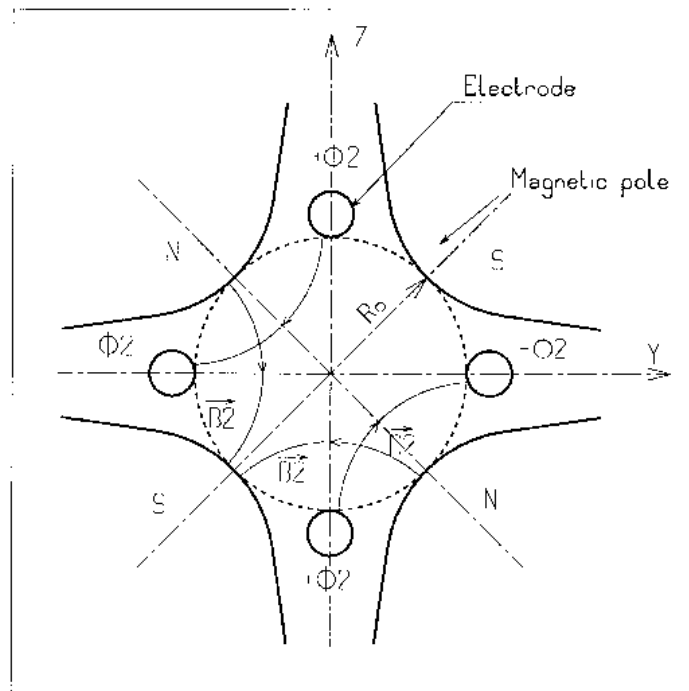


Figure 26: An example of  $\vec{E}$ ,  $\vec{B}$  multipole : the achromatic quadrupole (known for its allowing null second order chromatic aberrations [47]).

**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 [48]

$$V(X) = \frac{V_2 - V_1}{2} \operatorname{th} \frac{\omega x}{R_0} \left[ + \frac{V_1 + V_2}{2} \right] \quad \text{if } D = 0$$

$$V(X) = \frac{V_2 - V_1}{2} \frac{1}{2\omega D/R_0} \ell n \frac{\operatorname{ch} \omega \frac{x+D}{R_0}}{\operatorname{ch} \omega \frac{x-D}{R_0}} \left[ + \frac{V_1 + V_2}{2} \right] \quad \text{if } D \neq 0$$

( $x$  = distance from half-way between the electrodes ;  $\omega = 1.318$  ;  $\operatorname{th}$  = hyperbolic tangent ;  $\operatorname{ch}$  = hyperbolic cosine) from which 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.

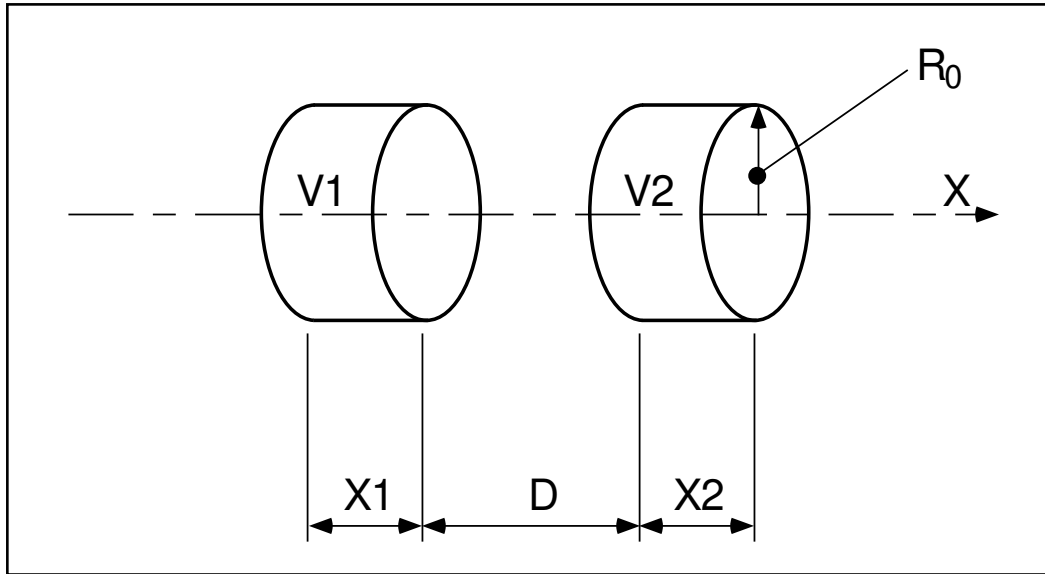


Figure 27: Two-electrode cylindrical electric lens.

**ELMIR : Electrostatic N-electrode mirror/lens, straight slits**

The device works as mirror or lens, horizontal or vertical. It is made of  $N$  2-plate electrodes and has mid-plane symmetry.

Electrode lengths are  $L1, L2, \dots, LN$ .  $D$  is the mirror/lens gap. The model for the  $Y$ -independent electrostatic potential is (after Ref. [49, p.412])

$$V(X, Z) = \sum_{i=2}^N \frac{Vi - Vi-1}{\pi} \arctan \frac{\sinh(\pi(X - Xi-1)/D)}{\cos(\pi Z/D)}$$

where  $Vi$  are the potential at the  $N$  electrodes (and normally  $V1 = 0$  refers to the incident beam energy),  $Xi$  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 27).

The total  $X$ -extent of the mirror/lens is  $L = \sum_{i=1}^N Li$ .

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 6.6.11 on page 188). 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.

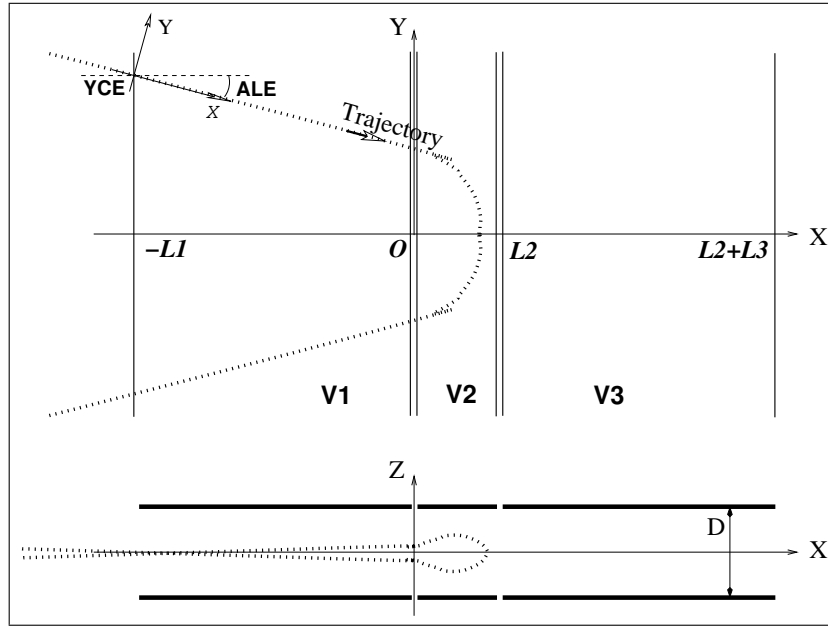


Figure 28: Electrostatic N-electrode mirror/lens, straight slits, in the case  $N = 3$ , in horizontal mirror mode ( $MT = 11$ ).

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 [49]**

The device works as mirror or lens, horizontal or vertical. It is made of  $N$  2-plate electrodes and has mid-plane symmetry<sup>7</sup>.

Electrode slits are circular, concentric with radii  $R1, R2, \dots, R_{N-1}$ ,  $D$  is the mirror/lens gap. The model for the mid-plane ( $Z = 0$ ) radial electrostatic potential is (after Ref. [49, 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,  $R1 - RE$  and  $R1 - RS$  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 6.6.8).

Use *PARTICUL* prior to *ELMIRC*, for the definition of particle mass and charge.

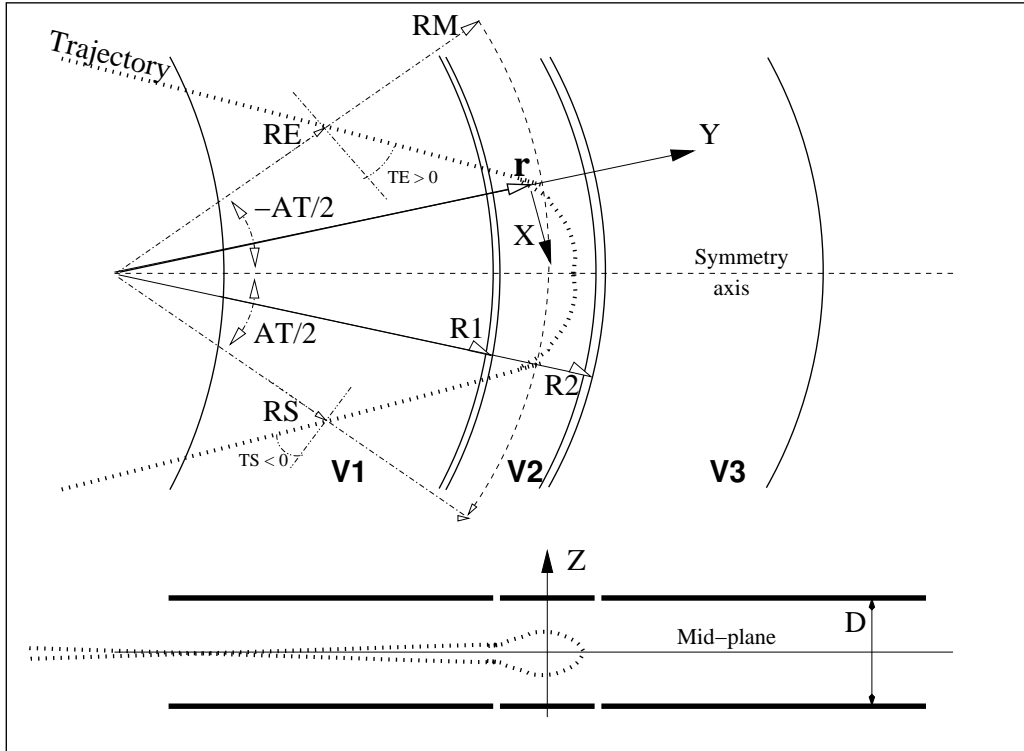


Figure 29: Electrostatic N-electrode mirror/lens, circular slits, in the case  $N = 3$ , in horizontal mirror mode.

<sup>7</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

$$\begin{aligned}\vec{M}_E &= \vec{E}1 + \vec{E}2 + \vec{E}3 + \dots + \vec{E}10 \\ \frac{\partial \vec{M}_E}{\partial X} &= \frac{\partial \vec{E}1}{\partial X} + \frac{\partial \vec{E}2}{\partial X} + \frac{\partial \vec{E}3}{\partial X} + \dots + \frac{\partial \vec{E}10}{\partial X} \\ \frac{\partial^2 \vec{M}_E}{\partial X \partial Z} &= \frac{\partial^2 \vec{E}1}{\partial X \partial Z} + \frac{\partial^2 \vec{E}2}{\partial X \partial Z} + \frac{\partial^2 \vec{E}3}{\partial X \partial Z} + \dots + \frac{\partial^2 \vec{E}10}{\partial X \partial Z} \\ &\text{etc.}\end{aligned}$$

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 28), 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 [47, 50]

$$K_n = \frac{1}{2} \frac{\gamma}{\gamma^2 - 1} \frac{V_n}{R_0^n}$$

( $V_n$  = 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  $\lambda_E * E_2$  ( $\lambda_E * E_3$ , ...,  $\lambda_E * E_{10}$ ) at entrance and  $\lambda_S * S_2$  ( $\lambda_S * S_3$ , ...,  $\lambda_S * S_{10}$ ) 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.

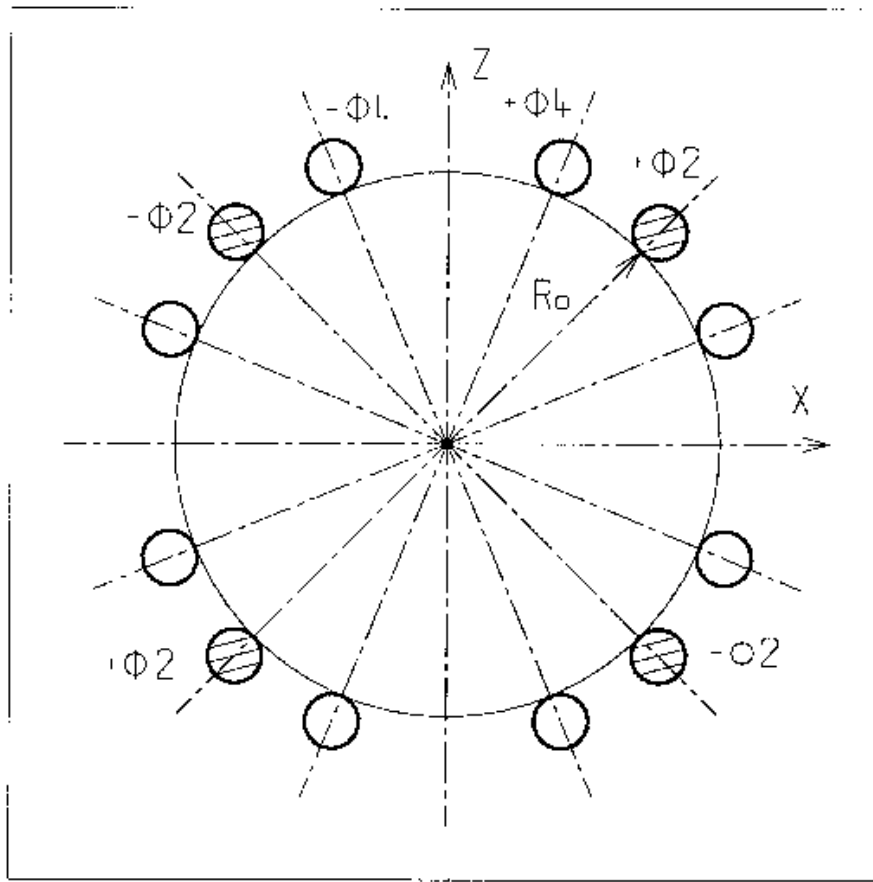


Figure 30: An electric multipole combining skew-quadrupole ( $\vec{E}2 \neq \vec{0}$ ,  $\vec{R}2 = \pi/4$ ) and skew-octupole ( $\vec{E}4 \neq \vec{0}$ ,  $\vec{R}4 = \pi/8$ ) components ( $\vec{E}1 = \vec{E}3 = \vec{E}5 = \dots = \vec{E}10 = \vec{0}$ ) [50].

**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.

**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>8</sup> [53, 54].

*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 *FORTTRAN* file `emmac.f`, for details.

<sup>8</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 [44, 45]**

*FFAG* works much like *DIPOL*ES 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 *DIPOL*ES does similar job by using a Taylor  $r$ -expansion of  $B_0(r/r_0)^k$ .

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. [44]. This is done in the way described below.

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 (in the same manner as in *DIPOL*E, *DIPOL*E- $M$ ) (see Fig. 11-A page 98, and Fig. 31)

- $ACN_i$  : 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

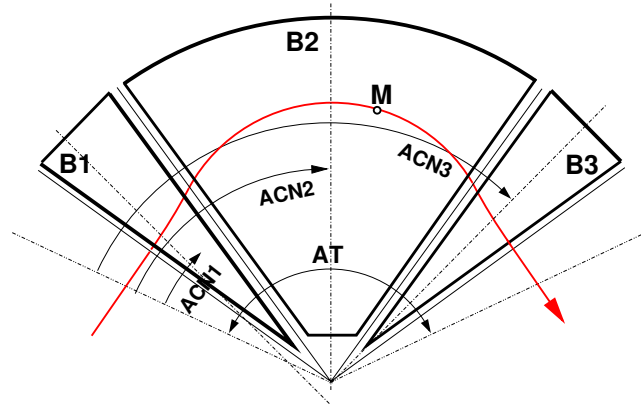


Figure 31: Definition of a dipole  $N$ -tuple ( $N = 3$ , a triplet here) using the *DIPOL*ES or *FFAG* procedures.

**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

$$B_{Zi}(R, \theta) = B_{Z0,i} \mathcal{F}_i(R, \theta) (R/R_M)^{K_i}$$

wherein  $B_{Z0,i}$  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 coefficient  $\mathcal{F}_i(R, \theta)$  associated with a dipole is computed as in the procedure *DIPOL*ES (eq. 6.4.16), including (rigorously if the interpolation method is used, see page 125, or to order zero if the analytic method is used, see page 125) radial dependence of the gap size

$$g(R) = g_0 (RM/R)^\kappa \quad (6.4.19)$$

so to simulate the effect of gap shaping on  $B_{Zi}(R, \theta)|_R$  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 *DIPOL*ES procedure (see page 125).

### 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. 11,  $\omega^+ = ACENT$  (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 [45, 51]**

*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 137). 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 11-A).

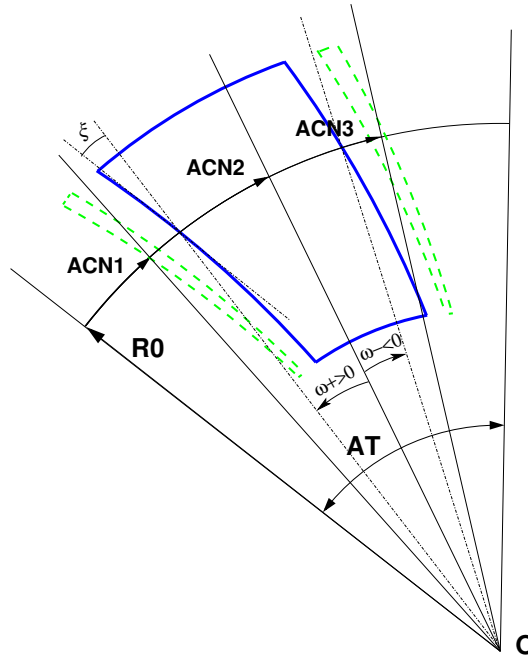


Figure 32: A  $N$ -tuple spiral sector FFAG magnet ( $N = 3$  here, simulating active field clamps at entrance and exit side of a central dipole).

**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

$$B_{Zi}(R, \theta) = B_{Z0,i} \mathcal{F}_i(R, \theta) (R/R_M)^{K_i}$$

wherein  $B_{Z0,i}$  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 coefficient  $\mathcal{F}_i(R, \theta)$  associated with a dipole is computed as in the procedure *DIPOLLES* (eq. 6.4.16), including radial dependence of the gap size

$$g(R) = g_0 (RM/R)^\kappa \quad (6.4.20)$$

so to simulate the effect of gap shaping on  $B_{Zi}(R, \theta)|_R$  field fall-off, over the all radial extent of the dipole (with normally - yet not necessarily in practice -  $\kappa \approx K_i$ ).

**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*ES procedure (see page 125).

**GOTO : Branching statement [55]**

\*\*\*\*\* To be completed \*\*\*\*\*

*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 structures, to direct the **zgoubi** amongst the series of spreaders of mergers. A example is given in \*\*\*\*\*.

An example of an input data file of a recirculator ring, structured with *GOTO* statements, can be found in page 248.

**INCLUDE : File include statement**

*INCLUDE* works much like the fortran “include” statement : it allows a fragment of an input data list to be taken from a separate file ; when *INCLUDE* is met in the input data file, the content of the separate file, or part of it, is inserted in the data sequence.

The *INCLUDE* statement takes the following arguments :

- the number of input files to be included,
- their names, in as many lines, one name per line.

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 some keyword 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 329).

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*.

An example of an input data file of a recirculating ring, structured with *INCLUDE* statements, can be found in page 248.

**MAP2D : 2-D Cartesian uniform mesh field map - arbitrary magnetic field [55]**

*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 *FORTRAN* 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
            READ(NL,100) Y(J), Z, X(I), BY(I,J), BZ(I,J), BX(I,J)
100      FORMAT (1X, 6E11.4)
          ENDIF
        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 27, 1.4.3 page 31). 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*.

**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 *FORTRAN* 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
      READ(NL,100) Y(J), Z, X(I), EY(I,J), EZ(I,J), EX(I,J)
100    FORMAT (1X, 6E11.4)
    ENDIF
  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 27, 1.4.3 page 31). 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*.



**MARKER : Marker**

*MARKER* does nothing. Just a marker. No data.

As any other keyword, *MARKER* is allowed two *LABEL*s. 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  $\lambda_E * E_2$  ( $\lambda_E * E_3, \dots, \lambda_E * E_{10}$ ) at entrance and  $\lambda_S * S_2$  ( $\lambda_S * S_3, \dots, \lambda_S * S_{10}$ ) 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 6.6.8).

**OCTUPOLE : Octupole magnet (Fig. 33)**

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. 152.

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

$$\begin{aligned} B_X &= 0 \\ B_Y &= G_0(3Y^2 - Z^2) Z \\ B_Z &= G_0(Y^2 - 3Z^2) Y \end{aligned}$$

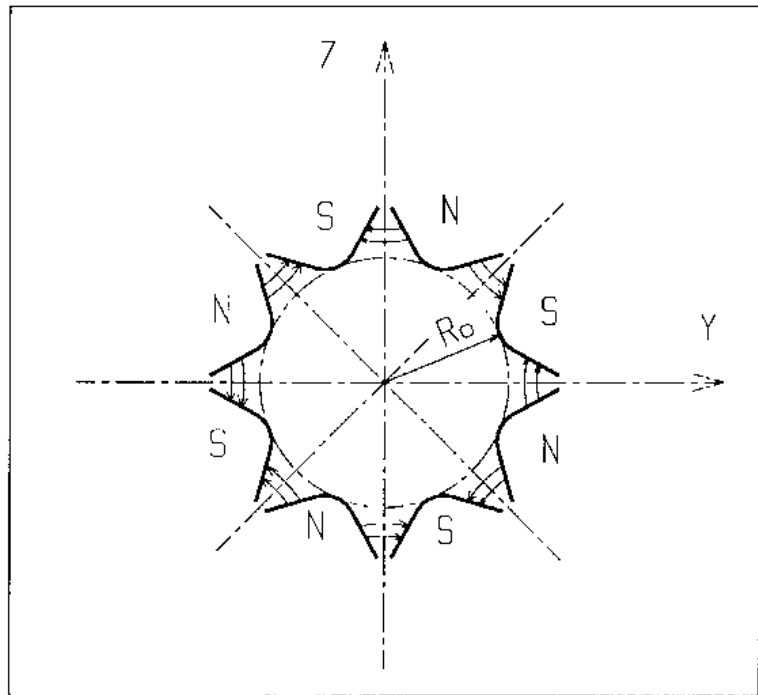


Figure 33: Octupole magnet

**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 *FORTTRAN* 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.  $Ks$  and  $Rs$  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**

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*).

**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).

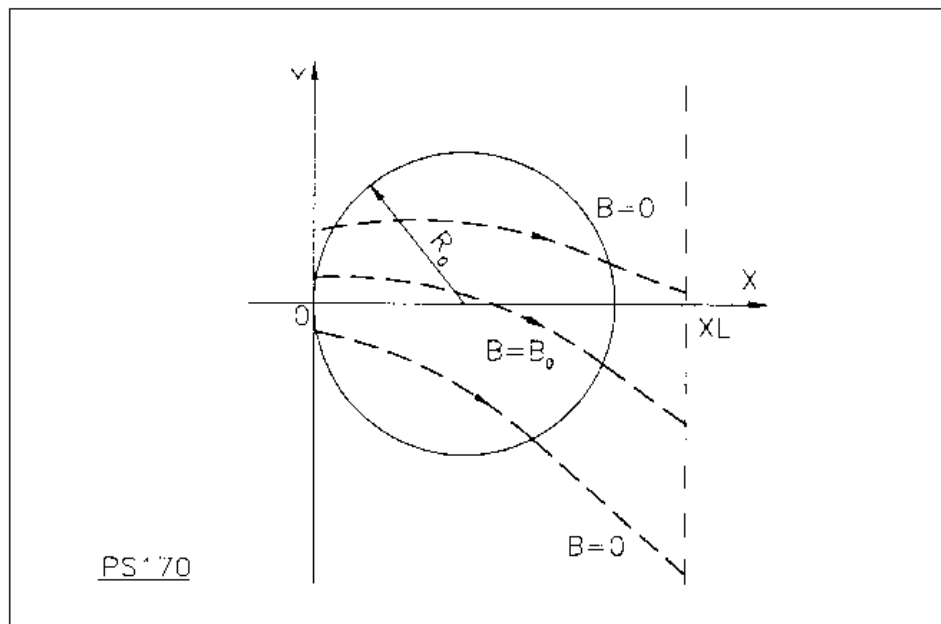


Figure 34: Scheme of the PS170 magnet simulation.

**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)**

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  $Z$ .

$$V(X, Y, Z) = \left( G(X) - \frac{G''(X)}{12} (Y^2 + Z^2) + \frac{G''''(X)}{384} (Y^2 + Z^2)^2 - \frac{G''''''(X)}{23040} (Y^2 + Z^2)^3 \right) YZ$$

$$( \quad G^{(n)}(X) = d^n G(X) / dX^n, \text{etc.} )$$

where  $G(X)$  is the gradient on axis [39] :

$$G(X) = \frac{G_0}{1 + \exp P(d(X))} \quad \text{with} \quad G_0 = \frac{B_0}{R_0}$$

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,  $\lambda \simeq 2 * R_0$ ).

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

$$\begin{aligned} B_X &= 0 \\ B_Y &= G_0 Z \\ B_Z &= G_0 Y \end{aligned}$$



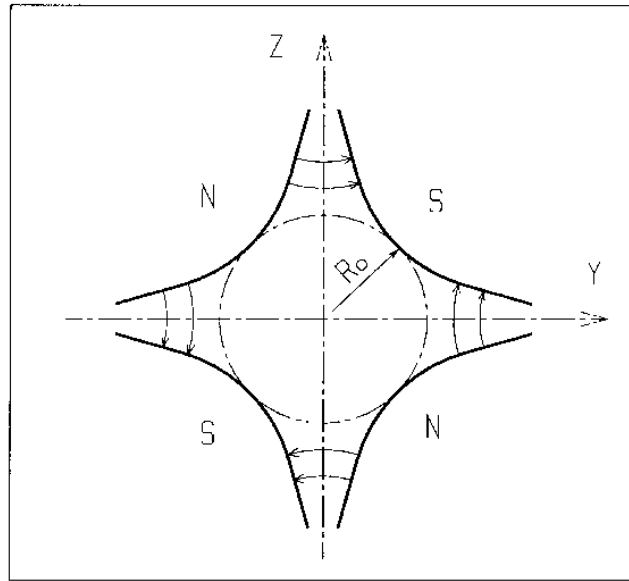
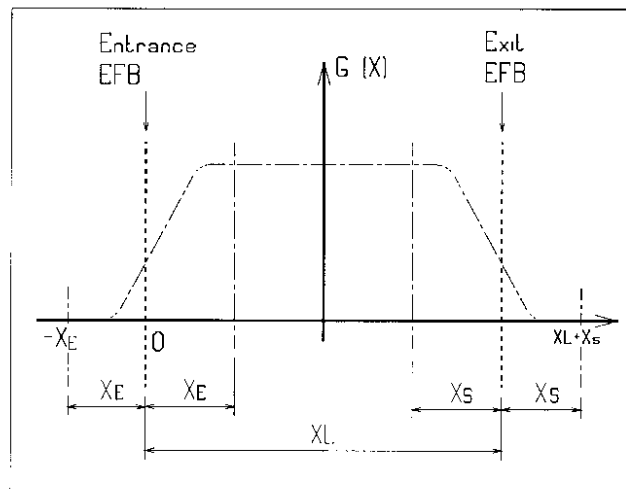


Figure 35: Quadrupole magnet

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 : simulation by stepwise integration can be found in *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

$$B(T) = -\frac{E \left( \frac{V}{m} \right)}{\beta_W \cdot c \left( \frac{m}{s} \right)}$$

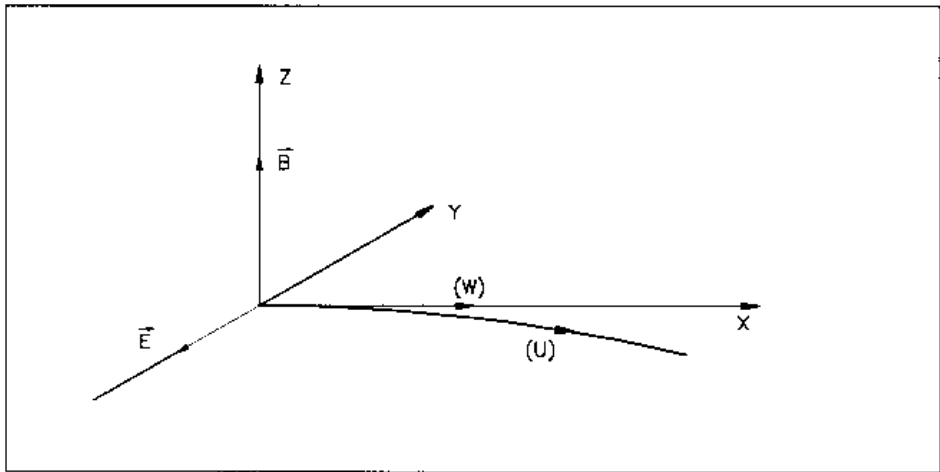


Figure 37: Horizontal separation between a wanted particle, (W), and an unwanted particle, (U). (W) undergoes a linear motion while (U) undergoes a cycloidal motion.

**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. 152.

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

$$\begin{aligned} B_X &= 0 \\ B_Y &= 2G_0 Y Z \\ B_Z &= G_0 (Y^2 - Z^2) \end{aligned}$$

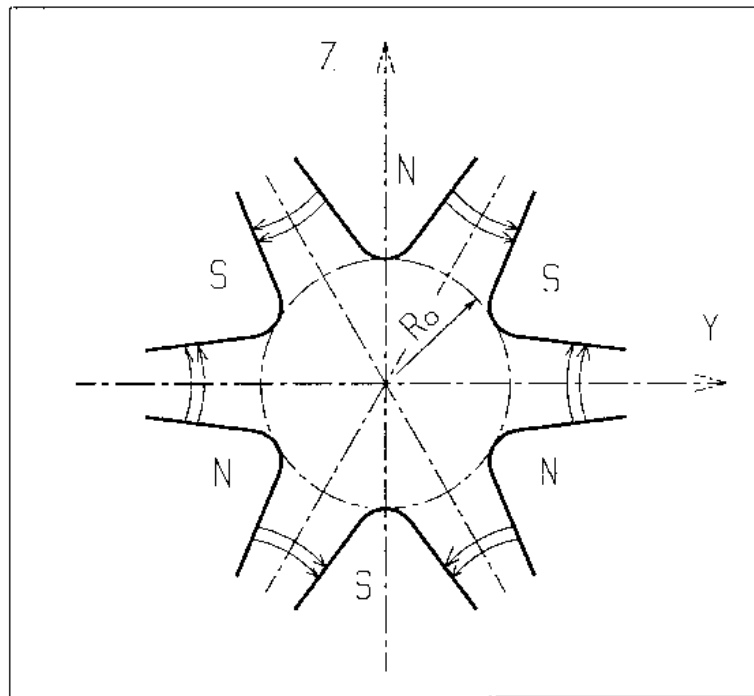


Figure 38: Sextupole magnet

**SOLENOID : Solenoid (Fig. 39)**

The solenoidal magnet has an effective length  $XL$ , a mean radius  $R_0$  and an asymptotic field  $B_0 = \mu_0 NI / XL$  (i.e.,  $\int_{-\infty}^{\infty} B_X(X, r) dX = \mu_0 NI$ ,  $\forall r < R_0$ ), wherein  $B_X$ =longitudinal field component,  $NI$  = number of ampere-Turns,  $\mu_0 = 4\pi 10^{-7}$ .

The distance of ray-tracing beyond the effective length  $XL$ , is  $X_E$  at the entrance, and  $X_S$  at the exit (Fig. 39).

Two methods are available for the computation of the field  $\vec{B}(X, r)$  and its derivatives.

**Method 1 :** uses the on-axis field model

$$B_X(X, r = 0) = \frac{B_0}{2} \left[ \frac{XL/2 - X}{\sqrt{(XL/2 - X)^2 + R_0^2}} + \frac{XL/2 + X}{\sqrt{(XL/2 + X)^2 + R_0^2}} \right] \quad (6.4.21)$$

with  $X = r = 0$  taken at the center of the solenoid. This model assumes that the coil thickness is small compared to the mean radius  $R_0$ . In this model, the magnetic length is

$$L_{mag} \equiv \frac{\int_{-\infty}^{\infty} B_X(X, r < R_0) dX}{B_X(X = r = 0)} = XL \sqrt{1 + \frac{4R_0^2}{XL^2}} > XL \quad (6.4.22)$$

with in addition

$$B_X(\text{center}) \equiv B_X(X = r = 0) = \frac{\mu_0 NI}{XL \sqrt{1 + \frac{4R_0^2}{XL^2}}} \xrightarrow{R_0 \ll XL} \frac{\mu_0 NI}{XL}$$

From eq. 6.4.21, the field and its derivatives at all  $(X, Y, Z)$  are extrapolated, following the method described in section 1.3.1.

**Method 2 :** computes  $\vec{B}(X, r)$  (with  $r = (Y^2 + Z^2)^{1/2}$ ) and its derivatives up to second order at all  $(X, Y, Z)$  following the technique in Ref. [56], based on the three complete elliptic integrals  $K$ ,  $E$  and  $\Pi$ . The latter are calculated with the algorithm proposed in the same reference, their derivatives are calculated by means of recursive relations [57].

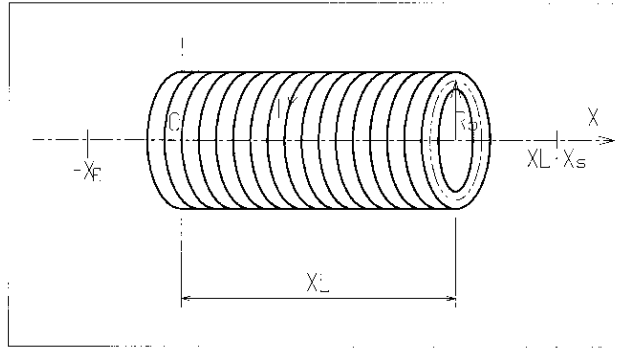


Figure 39: Solenoidal magnet.

**For software developers**

Solenoid parameters are read from `zgoubi.dat` by `rsolen.f`. Initializations prior to tracking done in `solenof.f`. Field computation during tracking is in `solenf.f`.

**SPINR : Spin rotation [58]**

*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 6.6.2, page 184).

**TOSCA : 2-D and 3-D Cartesian or cylindrical mesh field map** (page 293 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 159, lists all possibilities as a function of the values given to *IZ* and *MOD.MOD2*.

A double-flag, *MOD[.MOD2]*, determines whether Cartesian or Z-axis cylindrical mesh is used, and the nature and/or form of the field map data set.

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>9</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
        READ(NL,100) Y(J), Z(K), X(I), BY(J,K,I), BZ(J,K,I), BX(J,K,I)
100      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  $X$ -,  $Y$ -,  $Z$ -*JNORM* 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 *IORDRE* (*IORDRE* = 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 options, for instance :

- ◊ including 'HEADER n' allows specifying the number of header lines ('n' non-data lines) at the top of the field map file,
- ◊ 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>9</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  $MOD2$  possibilities, 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.

**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 fmapw.f/fmapr3).	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 - 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 IZ equally Z-spaced $(X, Y)$ planes (no symmetry assumed).	1 - 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 IZ equally Z-spaced $(X, Y)$ planes.	1 - 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 - 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 fmapw.f).	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 Z-spaced $(X, Y)$ planes. MOD=23 : special, test code (see fmapw.f).	1 - 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 IZ equally Z-spaced $(X, Y)$ planes.	1	$\theta, R, Z, B_\theta, BR, BZ$	

◇ including ‘RHIC\_helix’ will normalize  $BNORM$  to measured B field value versus helix current (measured B(I) data are hard-coded in program `toscac.f`). In that case,  $BNORM$  may be given the current value, in Amps, while  $a(1)$  in mode  $MOD.MOD2=15.1$  is used to normalize the field map (namely,  $a(1) = 1/B_{max}$  with  $B_{max}$  the maximum field value on helix axis).



**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) transfer matrix as usually involved in second order beam optics [33]. Second order transfer 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  $X1$  (resp.  $X2$ ,  $X3$ ). The distance between the electrodes is  $D$ . The potentials are  $V1$  and  $V2$ . The inner radius is  $R_0$  (Fig. 40). The model for the electrostatic potential along the axis is [59]

$$V(x) = \frac{V2 - V1}{2\omega D} \left[ \ln \frac{\cosh \frac{\omega \left( x + \frac{X2}{2} + D \right)}{R_0}}{\cosh \frac{\omega \left( x + \frac{X2}{2} \right)}{R_0}} + \ln \frac{\cosh \frac{\omega \left( x - \frac{X2}{2} - D \right)}{R_0}}{\cosh \frac{\omega \left( x - \frac{X2}{2} \right)}{R_0}} \right]$$

( $x$  = 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  $X1 + X2 + X3 + 2D$  ; stepwise integration starts at entrance of the first electrode and terminates at exit of the third one.

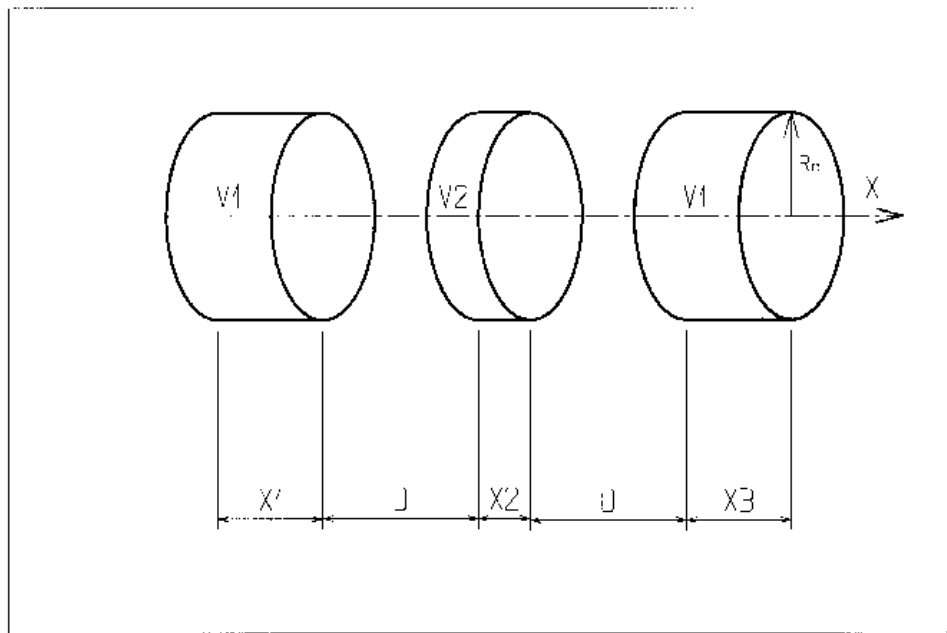


Figure 40: Three-electrode cylindrical unipotential lens.

**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. 41).

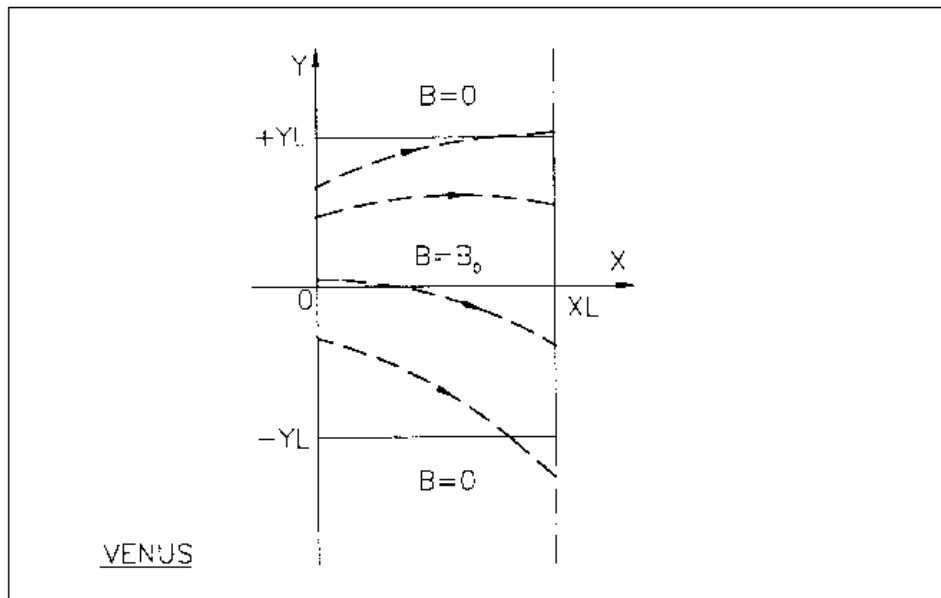


Figure 41: Scheme of *VENUS* rectangular dipole.

**WIENFILT : Wien filter**

*WIENFILT* simulates a Wien filter, with transverse and orthogonal electric and magnetic fields  $\vec{E}_Y, \vec{B}_Z$  or  $\vec{E}_Z, \vec{B}_Y$  (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 [39]

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

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 value of the coefficient respective to 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**

YMY performs a  $180^\circ$  rotation of particle coordinates with respect to the  $X$ -axis, as shown in Fig. 42. This is done by means of a change of sign of  $Y$  and  $Z$  axes, and therefore coordinates, as follows

$$Y2 = -Y1, \quad T2 = -T1, \quad Z2 = -Z1 \quad \text{and} \quad P2 = -P1$$

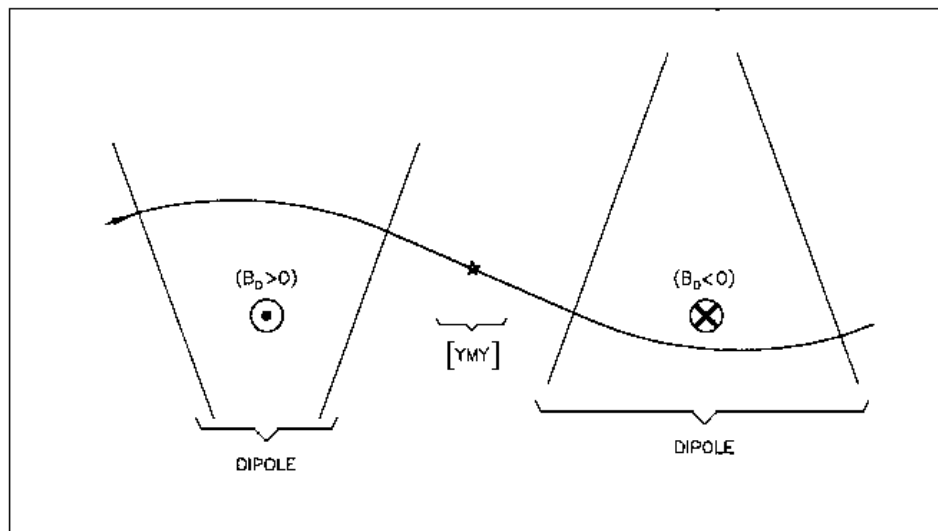


Figure 42: The use of YMY in a sequence of two identical dipoles of opposite signs.

### **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.



### 6.5.1 FAISCEAU, FAISCNL, FAISTORE : Print/Store particle coordinates

• **FAISCEAU** can be introduced anywhere in a structure data list (zgoubi.dat). 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 **FORTTRAN** 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),
2    -1.D0+F(1,I),F(2,I),F(3,I), (F(J,I),J=4,MXJ),
>    (SI(J,I),J=1,4), (SF(J,I),J=1,4),
>    ENEKI,ENERG,
4    I,IREP(I), SORT(I), (AMQ(J,I),J=1,5),RET(I),DPR(I),PS,
5    BORO, IPASS, NOEL, KLEY,LBL1,LBL2,LET(I)
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,
8    TX1,KLEY,TX1,TX1,LBL1,TX1,TX1,LBL2,TX1,TX1,LET(I),TX1
    INCLUDE "FRMFai.H"
1  CONTINUE
ENDIF

110  FORMAT(1X,
C1    KEX,      XXXO, (FO(J,IT),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**)

<i>LET(I)</i>	: one-character string, for tagging particle number <i>I</i>
<i>IEX, I, IREP(I)</i>	: flag, particle number, index
<i>FO(1 – 6, I)</i>	: coordinates <i>D, Y, T, Z, P</i> and path length at the origin of the structure
<i>F(1 – 6, I)</i>	: <i>idem</i> at the current position
<i>SORT(I)</i>	: path length at which the particle has possibly been stopped (see <b>CHAMBR</b> or <b>COLLIMA</b> )
<i>RET(I), DPR(I)</i>	: synchrotron phase space coordinates ; <i>RET</i> =phase (radian), <i>DPR</i> = momentum dispersion (MeV/c) (see <b>CAVITE</b> )
<i>IPASS</i>	: turn number (see <b>REBELOTE</b> )
<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      HPCKUP      VPCKUP
12
```

will result in output prints into *zgoubi.fai*, at pass 1 and then at every 12 other pass, each time elements of the *zgoubi.dat* data list labeled either *HPCKUP* 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. 70)).

### 6.5.2 FOCAL, IMAGE[S] : Particle coordinates and beam size ; localization and size of horizontal waist

*FOCAL* calculates the dimensions of the beam and its mean transverse position, at a longitudinal distance  $XL$  from the position corresponding to the keyword *FOCAL*.

*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*=1, 2 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*, *FOCAL*) or of each group of momenta (*IMAGES*)

- Longitudinal position :

$$FOCAL : X = XL$$

$$IMAGE[S] : X = - \frac{\sum_{i=1}^N Y_i * \text{tg}T_i - \left( \sum_{i=1}^N Y_i * \sum_{i=1}^N \text{tg}T_i \right) / N}{\sum_{i=1}^N \text{tg}^2 T_i - \left( \sum_{i=1}^N \text{tg}T_i \right)^2 / N}$$

$$Y = Y_1 + X * \text{tg}T_1$$

where  $Y_1$  and  $T_1$  are the coordinates of the first particle of the beam (*IMAGE*, *FOCAL*) 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 (*FOCAL*), 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 (*FOCAL*), and total width, respectively,  $W$  and  $WT$

$$W = 2.35 \left( \frac{1}{N} \sum_{i=1}^N Y M_i^2 - Y^2 M^2 \right)^{\frac{1}{2}}$$

$$WT = \max(YM_i) - \min(YM_i)$$

**6.5.3 FOCALZ, 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  $T_i$ .

#### 6.5.4 HISTO : 1-D histogram

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 in a structure. 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 in the structure will add up.

**6.5.5 IMAGE[S][Z] : Localization and size of vertical waists**

See FOCAL[Z].

### 6.5.6 MATRIX : Calculation of transfer coefficients, periodic parameters

*MATRIX* causes the calculation of the transfer coefficients through the optical structure, from the *OBJET* down to the location where *MATRIX* is introduced in the structure, or, upon option, down to the horizontal focus closest to that location. In this last case the position of the focus is calculated automatically in the same way as the position of the waist in *IMAGE*. Depending on *OBJET* and on option *IFOC*, *MATRIX* also delivers the beam matrix and betatron phase advances or (case of a periodic structure) periodic beam matrix and tunes, chromaticities and other global parameters.

Depending on the value of option *IORD*, different procedures follow

- If *IORD* = 0, *MATRIX* is inhibited (equivalent to *FAISCEAU*, whatever *IFOC*).
- If *IORD* = 1, using *OBJET*, *KOBJ* = 5[N], the first order transfer matrix  $[R_{ij}]$  is calculated, from a third order approximation of the coordinates. For instance

$$Y^+ = \left(\frac{Y}{T_0}\right) T_0 + \left(\frac{Y}{T_0^2}\right) T_0^2 + \left(\frac{Y}{T_0^3}\right) T_0^3, \quad Y^- = -\left(\frac{Y}{T_0}\right) T_0 + \left(\frac{Y}{T_0^2}\right) T_0^2 - \left(\frac{Y}{T_0^3}\right) T_0^3$$

will yield, neglecting third order terms,

$$R_{11} = \left(\frac{Y}{T_0}\right) = \frac{Y^+ - Y^-}{2T_0}$$

This is repeated  $N$  times if *OBJET*, *KOBJ* = 5.N ( $2 \leq N \leq 99$ ) is used, so delivering first order data for each 11-particle set, with for each set the reference trajectory being trajectory number  $1+(N-1) \times 11$  in the  $N \times 11$ -particle list (keyword *FAISCEAU* will print out that list in *zgoubi.res*, if desired).

If *OBJET*, *KOBJ* = 5.1 is used instead (hence introducing initial optical function values,  $\alpha_{Y,Z}$ ,  $\alpha'_{Y,Z}$ ,  $D_{Y,Z}$ ,  $D'_{Y,Z}$ ), then, using the  $R_{ij}$  above, *MATRIX* will transport the optical functions and phase advances  $\phi_Y$ ,  $\phi_Z$ , following

$$\begin{pmatrix} \beta \\ \alpha \\ \gamma \end{pmatrix}_{at \text{ MATRIX}} = \begin{pmatrix} R_{11}^2 & -2R_{11}R_{12} & R_{12}^2 \\ -R_{11}R_{21} & R_{12}R_{21} & R_{11}R_{12} \\ R_{21}^2 & -2R_{21}R_{22} & R_{22}^2 \end{pmatrix} \begin{pmatrix} \beta \\ \alpha \\ \gamma \end{pmatrix}_{at \text{ OBJET}}$$

$$\Delta\phi_Y = \text{Atan} \frac{R_{12}}{(R_{11}\beta_{Y,objet} - R_{12}\alpha_{Y,objet})}, \quad \Delta\phi_Z = \text{Atan} \frac{R_{34}}{(R_{33}\beta_{Z,objet} - R_{34}\alpha_{Z,objet})}, \quad (6.5.1)$$

$$\phi_{Y,Z} \rightarrow \phi_{Y,Z} + 2\pi \quad \text{if } \phi_{Y,Z} < 0, \text{ given } [0, \pi] \text{ Atan determination}$$

and print these out.

- If *IORD* = 2, using *OBJET*, *KOBJ* = 6, fifth order Taylor expansions are used for the calculation of the first order transfer matrix  $[R_{ij}]$  and of the second order matrix  $[T_{ijk}]$ . Other higher order coefficients are also calculated.
- If *IORD* = 3, using *OBJET*, *KOBJ* = 6.1, transport coefficients up to 3rd order are computed using 102 rays (after routines developed for *RAYTRACE* [60, 61]).

An automatic generation of an appropriate object for the use of *MATRIX* can be obtained using the procedure *OBJET* (pages 55, 258), as follows

- if *IORD* = 1, use *OBJET*(*KOBJ* = 5[N,  $2 \leq N \leq 99$ ]), that generates up to  $99 \times 11$  initial coordinates. In this case, up to 99 matrices may be calculated, each one wrt. to the reference trajectory of concern (trajectory number 1, 12, 23, ...  $1+(N-1) \times 11$  respectively).
- if *IORD* = 2, use *OBJET*(*KOBJ* = 6) that generates 61 initial coordinates.
- if *IORD* = 3, use *OBJET*(*KOBJ* = 6.1) that generates 102 initial coordinates.

The next option, *IFOC*, acts as follows

- If *IFOC* = 0, the transfer 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  $Y^+ = Y^+(i) - Y(Ref)$ , and  $T^+ = T^+(i) - T(ref.)$ .
- If *IFOC* = 1, the transfer 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  $Y^+ = Y^+(i) - Y_{focus}$ , while  $T^+$  is defined as  $T^+ = T^+(i) - T(ref.)$ .
- 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 Structures

- If *IFOC* = 10 + *NPeriod*, then, from the 1-turn transport matrix as obtained in the way described above, *MATRIX* calculates periodic parameters characteristic of the structure such as optical functions and tune numbers, assuming that it is *NPeriod*-periodic, and in the coupled hypothesis, based on the Edwards-Teng method [35]. This only makes sense when met at the end of the zgoubi.dat sequence.

This is repeated  $N$  times if *OBJET*, *KOBJ* = 5. $N$  ( $2 \leq N \leq 99$ ) is used, so delivering first order data for each 11-particle set.

If *IORD* = 2 (using *OBJET*, *KOBJ* = 6) additional periodic parameters are computed such as chromaticities, beta-function momentum dependence, etc.

**PRINT** : Addition of “*PRINT*” following *IORD*, *IFOC* [, *coupled*] 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 “*coupled*” next to *IORD*, *IFOC* [, *PRINT*], in the case of periodic beam matrix request (i.e., *IFOC* = 10 + *NPeriod*) will cause use of coupled 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.



### 6.5.7 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 structure, with *REBELOTE* thus being used to cause multi-turn tracking, *PICKUPS* are zeroed 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.

### 6.5.8 PLOTDATA : Intermediate output for the PLOTDATA graphic software

*PLOTDATA* was at the origin implemented for the purpose of plotting particle coordinates using the TRIUMF *PLOTDATA* package [62]. 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*.

### 6.5.9 SPNPRNL, SPNSTORE : Print/Store spin coordinates

• *SPNPRNL* has similar effect to *SPNPRT* (page 180), 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    HPCKUP    VPCKUP
12
```

will result in output prints into *zgoubi.spn*, every 12 other pass, each time elements of the *zgoubi.dat* data list labeled either *HPCKUP* or *VPCKUP* 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.

### 6.5.10 SPNPRT : Print spin coordinates

*SPNPRT* can be introduced anywhere in a structure. It produces a print out (to zgoubi.res) 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 data list.

*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 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, the local stable precession axis coordinates) and rigidity, and their spins are on a direct trihedra, 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* with *KOBJ=1* or *KOBJ=2* can be used for instance to generate these particles by groups of like rigidity. *SPNPRT* will thus compute as many matrices.

**6.5.11 SRPRNT : Print SR loss statistics**

*SRPRNT* may be introduced anywhere in a structure. 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*.

### 6.5.12 TWISS : Calculation of periodic optical parameters. Log to zgoubi.TWISS.out

*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* is normally placed at the end of the structure ; it causes a series of up to 5 successive passes in the structure (at the manner of *REBELOTE*).

The object necessary for these calculations will be generated automatically if one uses *OBJET* with option *KOBJ= 5*.

*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 11, 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 297.

- A first pass (the only one if *KTW=1*) through the structure allows computing the periodic optical functions at the end of the structure, from the rays.

- The periodic dispersions are used to define chromatic closed orbits at  $\pm\delta p/p$ . 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.

- 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).

#### 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.

## 6.6 Complements Regarding Various Functionalities

### 6.6.1 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 multipol 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 multipol 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.

### 6.6.2 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

$$B\rho_{ref} = BORO \longrightarrow B\rho_{ref} = BORO + \delta B\rho_s$$

with  $\delta B\rho_s$  the synchronous rigidity increase (or decrease). A typical configuration where this would occur is that of multi-turn tracking in a pulsed synchrotron, where in general strengths have to follow the acceleration (see section 6.6.7).

### 6.6.3 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 6.6.7), 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).

### 6.6.4 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 structure, and then reversing the integration step sign in all the optical elements.

An illustration of this feature is given in the following Figure 43.

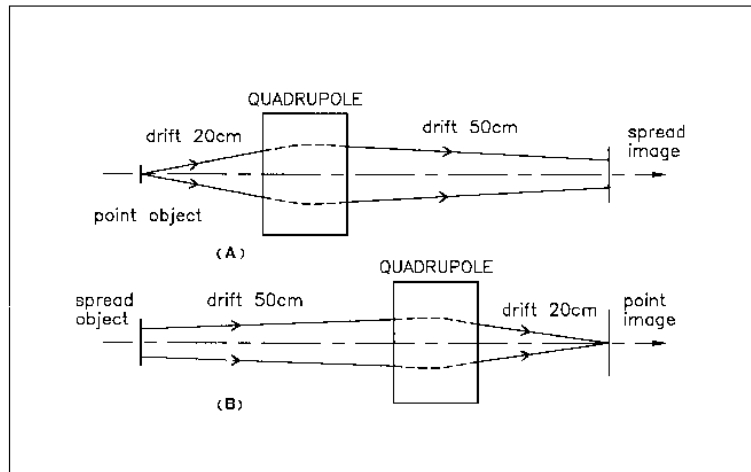


Figure 43: A. Regular forward ray-tracing, from object to image.

B. Same structure, with backward ray-tracing from image to object : negative integration step XPAS is used in the quadrupole.

### 6.6.5 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).

- *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>10</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).

<sup>10</sup>See Part D of the Guide.



-  $IL = 7$  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. 44, a plot of the quadrupole and sextupole field indices along the reference orbit in a combined function main dipole pertaining to the AGS lattice [63].

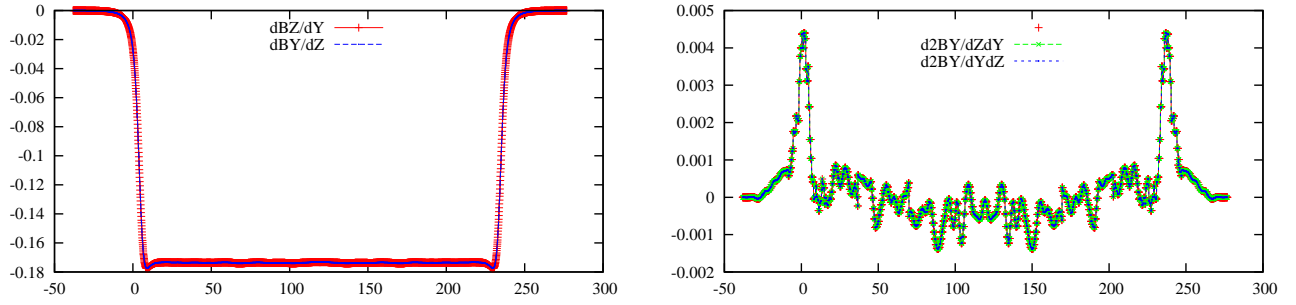


Figure 44: Typical field and derivatives - here, along the 10 GeV proton orbit across an AGS A-type main magnet. Derivatives are obtained from second degree polynomial interpolation from the magnet measured field map. Left :  $dB_Z(s)/dY$  and  $dB_Y(s)/dZ$ , right :  $d^2B_Y(s)/dZdY$ ,  $d^2B_Y(s)/dYdZ$ . These data were stored in `zgoubi.impdev.out` during execution, by stating  $IL = 7$  under *TOSCA* keyword.

- 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 128), to log local coordinates to `zgoubi.plt` in the same manner as  $IL=2$  does.
- *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.

### 6.6.6 Labeling Keywords

Keywords in `zgoubi` input data file `zgoubi.dat` can be *LABEL*'ed, for the purpose of the execution of such procedures as *REBELOTE*, *PICKUPS*, *FAISCNL*, *FAISTORE*, *SCALING*,.. And also for the purpose of particle coordinate storage into `zgoubi.plt` (see section 6.6.5,  $IL = 2$  option).

A keyword in `zgoubi.dat` accepts two *LABEL*s. 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`).

### 6.6.7 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` data list with its argument  $NPASS+1$  being the number of turns to be performed. It will cause a jump of the "multi-turn pointer" back to (details on page 79), either the beginning of the data list (default case), or to a particular *LABEL* in that list. From then on, tracking resumes down to *REBELOTE* again, and so forth until the requested number of passes has been reached.

Note : Some *CAVITE* options cause a *reset-to-zero of individual particle path length*<sup>11</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.

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>11</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* data list (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).

### 6.6.8 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, *e.g.*, *XCE*, *YCE* for translation and *ALE* for rotation. The positioning works in two different ways, depending whether the element is defined in Cartesian ( $X, Y, Z$ ) coordinates (*e.g.*, *QUADRUPO*, *TOSCA*), or polar ( $R, \theta, Z$ ) coordinates (*DIPOLE*).

#### 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. 45. 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. 45.

*KPOS* = 3 option is available for some optical elements (*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 structures, as follows (Fig 46).

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 *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).

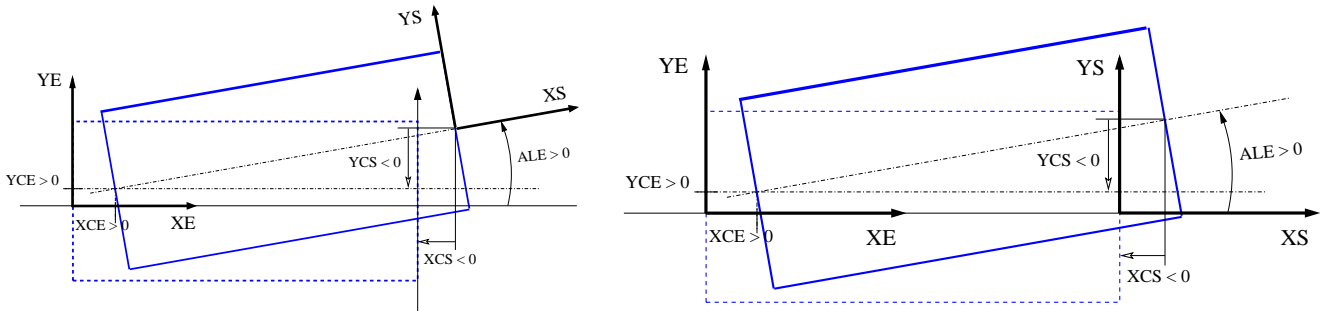


Figure 45: Case of Cartesian frame optical element. Left : positioning an optical element using *KPOS* = 1. Right : Mis-aligning an optical element using *KPOS* = 2. ( $X_E, Y_E$ ) and ( $X_S, Y_S$ ) are respectively the incoming and outgoing reference frames.

*KPOS* = 4 applies to *AGSMM* (AGS main magnet) and *MULTIPOL*. It orients the magnet in a way similar to *KPOS* = 3, with reference frame Z-rotated by  $\alpha_Z/2$  as drawn from  $L = 2 \frac{BORO}{B1} \sin(\alpha_Z/2)$ .

However additional magnet mis-alignments (alignment errors) are available (Fig. 47) : Z-rotation  $\theta_Z$ , Z-shift  $\delta Z$ , Y-rotation  $\theta_Y$ . Rotations apply at the center of the magnet.

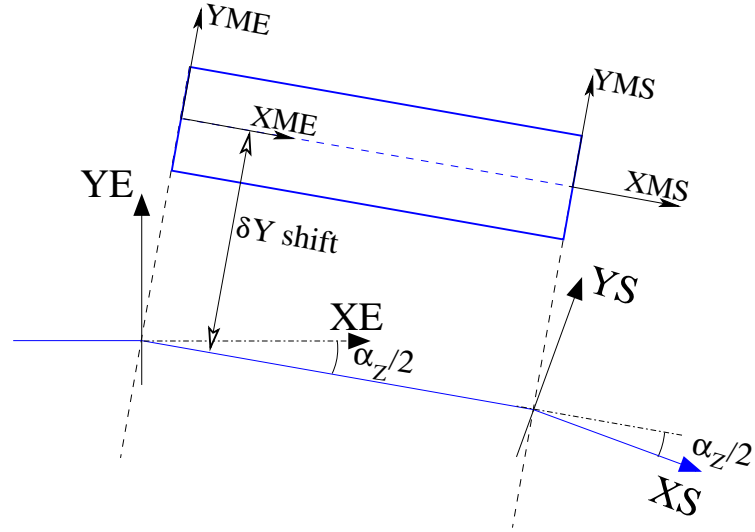


Figure 46: Alignment in *BEND* by half the deviation,  $\alpha_Z/2$ , using  $KPOS=3$ .  $(X_E, Y_E)$  and  $(X_S, Y_S)$  are respectively the incoming and outgoing reference frames. Trajectory coordinates inside the magnet are defined in the magnet frame,  $(X_{ME}, Y_{ME})$ .

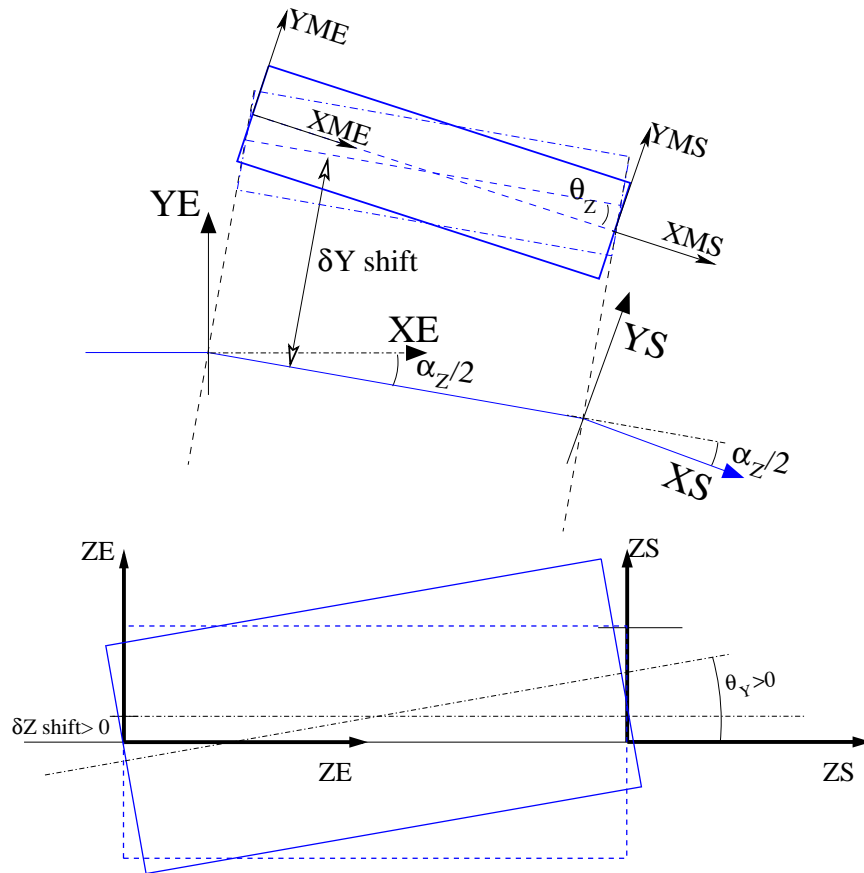


Figure 47: Additional magnet mis-alignment with  $KPOS=4$ .

### 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 (Fig. ??; 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. 11A page 98A).

#### 6.6.9 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.

#### 6.6.10 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 structure, 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.

#### 6.6.11 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 `depla.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

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 29). In all other cases the particle will be stopped.

#### 6.6.12 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*.

## **PART B**

### **Keywords and input data formatting**



## Glossary of Keywords, Part B

---

### Available keywords : how to format their input data list.

<b>AGSMM</b>	AGS main magnet .....	197
<b>AGSQUAD</b>	AGS quadrupole .....	198
<b>AIMANT</b>	Generation of dipole mid-plane 2-D map, polar frame .....	199
<b>AUTOREF</b>	Transport beam into a new reference frame .....	203
<b>BEAMBEAM</b>	Beam-beam lens .....	204
<b>BEND</b>	Bending magnet, Cartesian frame .....	205
<b>BETATRON</b>	Betatron core (momentum kick) .....	207
<b>BINARY</b>	<i>BINARY</i> ↔ <i>FORMATTED</i> data converter .....	208
<b>BREVOL</b>	1-D uniform mesh magnetic field map .....	209
<b>CARTEMES</b>	2-D Cartesian uniform mesh magnetic field map .....	210
<b>CAVITE</b>	Accelerating cavity .....	212
<b>CHAMBR</b>	Long transverse aperture limitation .....	213
<b>CHANGREF</b>	Transformation to a new reference frame .....	214
<b>CIBLE</b>	Generate a secondary beam following target interaction .....	215
<b>COLLIMA</b>	Collimator .....	216
<b>DECAPOLE</b>	Decapole magnet .....	217
<b>DIPOLE</b>	Dipole magnet, polar frame .....	218
<b>DIPOLE-M</b>	Generation of dipole mid-plane 2-D map, polar frame .....	219
<b>DIPOLES</b>	Dipole magnet <i>N</i> -tuple, polar frame .....	221
<b>DODECAPO</b>	Dodecapole magnet .....	222
<b>DRIFT</b>	Field free drift space .....	223
<b>EBMULT</b>	Electro-magnetic multipole .....	224
<b>EL2TUB</b>	Two-tube electrostatic lens .....	226
<b>ELMIR</b>	Electrostatic N-electrode mirror/lens, straight slits .....	227
<b>ELMIRC</b>	Electrostatic N-electrode mirror/lens, circular slits .....	228
<b>ELMULT</b>	Electric multipole .....	229
<b>ELREVOL</b>	1-D uniform mesh electric field map .....	230
<b>EMMA</b>	2-D Cartesian or cylindrical mesh field map for EMMA FFAG .....	231
<b>END</b>	End of input data list .....	236
<b>ESL</b>	Field free drift space .....	223
<b>ERRORS</b>	Injecting errors in optical elements .....	232
<b>FAISCEAU</b>	Print particle coordinates .....	233
<b>FAISCNL</b>	Store particle coordinates in file FNAME .....	233
<b>FAISTORE</b>	Store coordinates every <i>IP</i> other pass at labeled elements .....	233
<b>FFAG</b>	FFAG magnet, <i>N</i> -tuple .....	234
<b>FFAG-SPI</b>	Spiral FFAG magnet, <i>N</i> -tuple .....	235
<b>FIN</b>	End of input data list .....	236
<b>FIT,FIT2</b>	Fitting procedure .....	237
<b>FOCALE</b>	Particle coordinates and horizontal beam size at distance <i>XL</i> .....	241
<b>FOCALEZ</b>	Particle coordinates and vertical beam size at distance <i>XL</i> .....	241
<b>GASCAT</b>	Gas scattering .....	242
<b>GETFITVAL</b>	Get values of <i>variables</i> as saved from former FIT[2] run .....	243
<b>GOTO</b>	Branching statement .....	244
<b>HISTO</b>	1-D histogram .....	245
<b>IMAGE</b>	Localization and size of horizontal waist .....	246
<b>IMAGES</b>	Localization and size of horizontal waists .....	246
<b>IMAGESZ</b>	Localization and size of vertical waists .....	246
<b>IMAGEZ</b>	Localization and size of vertical waist .....	246
<b>INCLUDE</b>	File include statement .....	247
<b>MAP2D</b>	2-D Cartesian uniform mesh field map - arbitrary magnetic field .....	249
<b>MAP2D-E</b>	2-D Cartesian uniform mesh field map - arbitrary electric field .....	250

<b>MARKER</b>	Marker .....	251
<b>MATRIX</b>	Calculation of transfer coefficients, periodic parameters .....	252
<b>MCDESINT</b>	Monte-Carlo simulation of in-flight decay .....	253
<b>MCOBJET</b>	Monte-Carlo generation of a 6-D object .....	254
<b>MULTIPOL</b>	Magnetic multipole .....	257
<b>OBJET</b>	Generation of an object .....	258
<b>OBJETA</b>	Object from Monte-Carlo simulation of decay reaction .....	261
<b>OCTUPOLE</b>	Octupole magnet .....	262
<b>OPTICS</b>	Write out optical functions. Log to zgoubi.OPTICS.out .....	263
<b>OPTIONS</b>	Global or special options .....	264
<b>ORDRE</b>	Taylor expansions order .....	265
<b>PARTICUL</b>	Particle characteristics .....	266
<b>PICKUPS</b>	Bunch centroid path; orbit .....	267
<b>PLOTDATA</b>	Intermediate output for the PLOTDATA graphic software .....	268
<b>POISSON</b>	Read magnetic field data from <i>POISSON</i> output .....	269
<b>POLARMES</b>	2-D polar mesh magnetic field map .....	270
<b>PS170</b>	Simulation of a round shape dipole magnet .....	271
<b>QUADISEX</b>	Sharp edge magnetic multipoles .....	272
<b>QUADRUPO</b>	Quadrupole magnet .....	273
<b>REBELOTE</b>	'Do it again' .....	275
<b>RESET</b>	Reset counters and flags .....	277
<b>SCALING</b>	Power supplies and R.F. function generator .....	278
<b>SEPARA</b>	Wien Filter - analytical simulation .....	279
<b>SEXQUAD</b>	Sharp edge magnetic multipole .....	280
<b>SEXTUPOL</b>	Sextupole magnet .....	281
<b>SOLENOID</b>	Solenoid .....	282
<b>SPACECHARG</b>	Space charge .....	283
<b>SPINR</b>	Spin rotation .....	284
<b>SPNPRNL</b>	Store spin coordinates into file FNAME .....	285
<b>SPNSTORE</b>	Store spin coordinates every <i>IP</i> other pass at labeled elements .....	285
<b>SPNPRT</b>	Print spin coordinates .....	285
<b>SPNTRK</b>	Spin tracking .....	286
<b>SRLOSS</b>	Synchrotron radiation energy loss .....	287
<b>SRPRNT</b>	Print SR loss statistics .....	288
<b>SVDOC</b>	SVD orbit correction .....	289
<b>SYNRAD</b>	Synchrotron radiation spectral-angular densities .....	290
<b>SYSTEM</b>	System call .....	291
<b>TARGET</b>	Generate a secondary beam following target interaction .....	215
<b>TOSCA</b>	2-D and 3-D Cartesian or cylindrical mesh field map .....	293
<b>TRANSMAT</b>	Matrix transfer .....	295
<b>TRAROT</b>	Translation-Rotation of the reference frame .....	296
<b>TWISS</b>	Calculation of periodic optical parameters. Log to zgoubi.TWISS.out .....	297
<b>UNDULATOR</b>	Undulator magnet .....	299
<b>UNIPOT</b>	Unipotential cylindrical electrostatic lens .....	300
<b>VENUS</b>	Simulation of a rectangular shape dipole magnet .....	301
<b>WIENFILT</b>	Wien filter .....	302
<b>YMY</b>	Reverse signs of <i>Y</i> and <i>Z</i> reference axes .....	303



## Optical elements versus keywords

### What can be simulated What keyword(s) can be used for that

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 or sector	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	DIPOL, 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	BREVL
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 *FORTRAN 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.



AGSMM	AGS main 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
$MOD, dL, R_0, dB1, dB2, dB3$	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
$NBLW[.MOD], NBLW \text{ times} : NW, I$	Number of back-leg windings [back-leg winding model <sup>2</sup> ] ; for each back-leg winding : number of windings, current.	$\leq 2, NBLW \times$ (any, Amp.)	I, $NBLW \times$ (I, E)
$X_E, \lambda_E, E_2, E_3$	<b>Entrance face</b> Integration zone ; fringe field extent : dipole fringe field extent = $\lambda_E$ ; quadrupole fringe field extent = $\lambda_E * E_2$ ; sextupole fringe field extent = $\lambda_E * E_3$ (sharp edge if field extent is zero)	2*cm, 2*no dim.	4*E
$NCE, C_0 - C_5$	same as <i>QUADRUPO</i>	0-6, 6*no dim.	I, 6*E
$X_S, \lambda_S, S_2, S_3$	<b>Exit face</b> Integration zone ; as for entrance	2*cm, 2*no dim.	4*E
$NCS, C_0 - C_5$		0-6, 6*no dim.	I, 6*E
$R1, R2, R3$	Skew angles of field components	3*rad	10*E
$XPAS$	Integration step	cm	E
<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 $B1 \neq 0$ : entrance and exit frames are tilted wrt. the magnet by an angle of • either ALE if $ALE \neq 0$ • or $2 \arcsin(B1 XL / 2BORO)$ if $ALE=0$	1-3, 2*cm, rad	I, 3*E
<b>If KPOS = 4</b> $KPOS, XS, YS, ZR, ZS, YR$	X-, Y-shift, Z-rotation, Z-shift, Y-rotation Automatic ALE positioning as $KPOS = 3$ , and in addition possible X- or Y- or Z-misalignment or Z-, Y-rotation. (under development).	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.

**AGSQUAD****AGS quadrupole**

		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	
$XL, R_0, I_{W1}, I_{W2}, I_{W3}, dI_{W1}, dI_{W2}, dI_{W3}$	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
$X_E, \lambda_E$	<b>Entrance face</b> Integration zone ; fringe field extent. (sharp edge if field extent is zero)	2*cm	11*E	20, 21
$NCE, C_0 - C_5$	Same as <i>QUADRUPO</i>	0-6, 6*no dim.	I, 6*E	30 - 36
$X_S, \lambda_S$	<b>Exit face</b> Integration zone ; as for entrance	2*cm	11*E	40, 41
$NCS, C_0 - C_5$		0-6, 6*no dim.	I, 6*E	50 - 56
$R1$	Roll angle	rad	E	60
$XPAS$	Integration step	cm	E	70
$KPOS, XCE, YCE, ALE$	$KPOS=1$ : element aligned, $2$ : misaligned ; shifts, tilt.	1-2, 2*cm, rad	I, 3*E	80 - 83

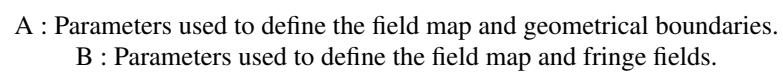
**AIMANT****Generation of dipole mid-plane 2-D map, polar frame**

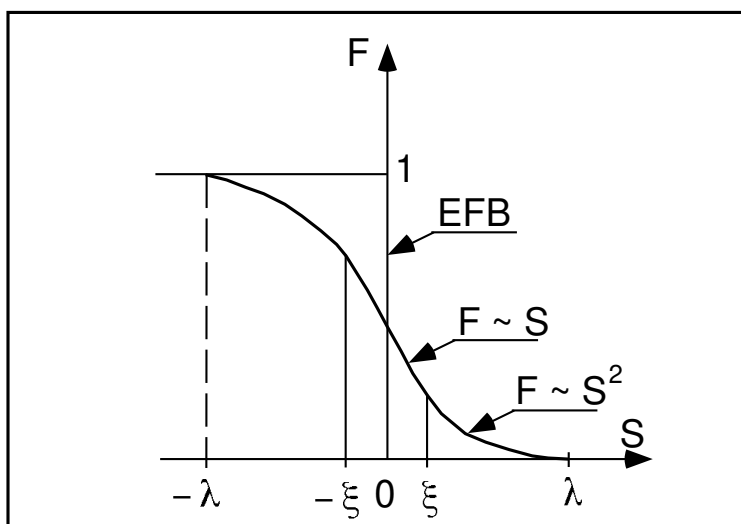
$$B_Z = \mathcal{F}B_0 \left( 1 - N \left( \frac{R - RM}{RM} \right) + B \left( \frac{R - RM}{RM} \right)^2 + G \left( \frac{R - RM}{RM} \right)^3 \right)$$

<i>NFACE, IC, IL</i>	Number of field boundaries <i>IC</i> = 1, 2 : print field map <i>IL</i> = 1, 2[×10 <sup>n</sup> ], 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[×10 <sup>n</sup> ], 7 3*I	
<i>IAMAX, IRMAX</i>	Azimuthal and radial number of nodes of the mesh	≤ 400, ≤ 10 <sup>4</sup>	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
<b>ENTRANCE FIELD BOUNDARY</b>			
<i>λ, ξ</i>	Fringe field extent (normally ≃ 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 : $F = (1 + \exp(P(s)))^{-1}$ $P(s) = C_0 + C_1(\frac{s}{\lambda}) + C_2(\frac{s}{\lambda})^2 + \dots + C_5(\frac{s}{\lambda})^5$	cm, (cm)	2*E
<i>NC, C<sub>0</sub> – C<sub>5</sub>, shift</i>	NC = 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
<i>ω<sup>+</sup>, θ, R<sub>1</sub>, U<sub>1</sub>, U<sub>2</sub>, R<sub>2</sub></i>	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> )			
<b>EXIT FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)</b>			
<i>λ, ξ</i>	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
<i>ω<sup>-</sup>, θ, R<sub>1</sub>, U<sub>1</sub>, U<sub>2</sub>, R<sub>2</sub></i>	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> )			

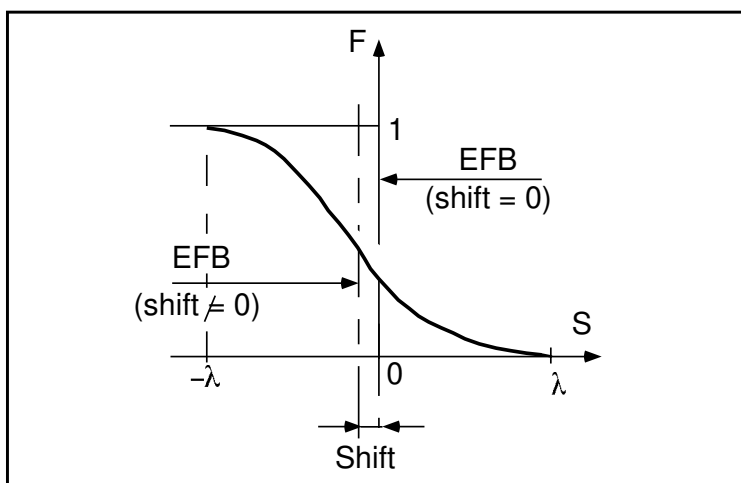
<b>If NFACE = 3</b>	LATERAL FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY) Next 3 records <i>only</i> if NFACE = 3		
$\lambda, \xi$	Fringe field parameters	cm, (cm)	2*E
$NC, C_0 - C_5, shift$	See above	0-6, 6*no dim., cm	I, 7*E
$\omega^-, \theta, R_1, U_1, U_2, R_2, RM3$	Positioning and shape of the lateral 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 $B * \left( 1 + \frac{\Delta B}{B_0} \frac{R - R_0}{RM_{AX} - RM_{IN}} \right)$	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>IORDRE</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







Second order type fringe field.



Exponential type fringe field.

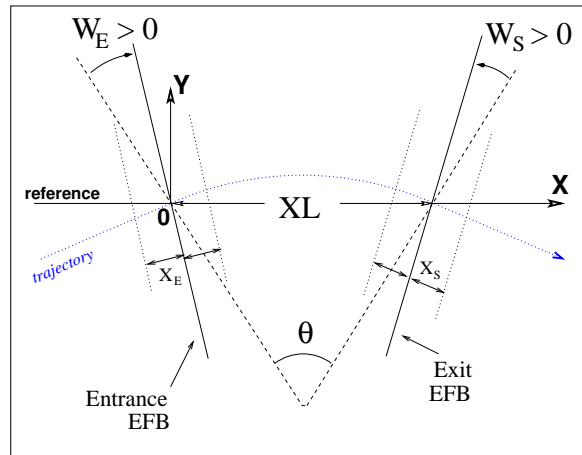
AUTOREF	Transport beam into a new reference frame			
<i>I</i>	1 : Equivalent to <i>CHANGREF</i> ( <i>XCE</i> = 0, <i>YCE</i> = <i>Y</i> (1), <i>ALE</i> = <i>T</i> (1)), <i>i.e.</i> , recentering of the beam on particle #1.	1-2 1-2	I I	
	2 : Equivalent to <i>CHANGREF</i> ( <i>XW</i> , <i>YW</i> , <i>T</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).			
	3 : Equivalent to <i>CHANGREF</i> ( <i>XW</i> , <i>YW</i> , <i>T</i> ( <i>I</i> 1)), with ( <i>XW</i> , <i>YW</i> ) the location of the intersection (waist) of particles <i>I</i> 1, <i>I</i> 2 and <i>I</i> 3 (for instance : <i>I</i> 1 = central trajectory, <i>I</i> 2 and <i>I</i> 3 = paraxial trajectories that intersect at the first order focus).			
	4 : Equivalent to <i>CHANGREF</i> ( <i>XCE</i> , <i>YCE</i> , <i>ALE</i> ).			
	4.1 : Equivalent to <i>CHANGREF</i> ( <i>XCE</i> , <i>YCE</i> , <i>ALE</i> ) with in addition centering of the beam on a new relative momentum <i>DCE</i> .			
	5 : The beam is centered vertically on <i>ZCE</i> , <i>PLE</i> .			
<b>If</b> <i>I</i> = 3 <i>I</i> 1, <i>I</i> 2, <i>I</i> 3	Provide next record if <i>I</i> = 3 Three particle numbers	3*(1- <i>IMAX</i> )	3*I	
<b>If</b> <i>I</i> = 4 <i>XCE</i> , <i>YCE</i> , <i>ALE</i>	Provide next record if <i>I</i> = 4 <i>XCE</i> and beam centroid new coordinates <i>YCE</i> , <i>ALE</i>	2*cm, mrad	3*E	
<b>If</b> <i>I</i> = 4.1 <i>XCE</i> , <i>YCE</i> , <i>ALE</i> , <i>DCE</i> , <i>TIME</i>	Provide next record if <i>I</i> = 4.1 <i>XCE</i> and beam centroid new coordinates <i>YCE</i> , <i>ALE</i> , <i>DCE</i> , <i>TIME</i>	2*cm, mrad, -, $\mu$ S	6*E	
<b>If</b> <i>I</i> = 4.2 <i>XCE</i> , <i>YCE</i> , <i>ALE</i> , <i>DCE</i> , <i>TIME</i>	Provide next record if <i>I</i> = 4.2 <i>XCE</i> and beam centroid new coordinates <i>YCE</i> , <i>ALE</i> , <i>DCE</i> , time setting <i>TIME</i> (same for all particles).	2*cm, mrad, -, $\mu$ S	6*E	
<b>If</b> <i>I</i> = 5 <i>ZCE</i> , <i>PLE</i>	Provide next record if <i>I</i> = 5 New vertical beam centroid coordinates <i>ZCE</i> , <i>PLE</i>	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

**BEND****Bending magnet, Cartesian frame**

$IL$	$IL = 1, 2[\times 10^n]$ , 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 \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

4 .2401 1.8639 -.5572 .3904 0. 0. 0.

20. 8. 0.04276056667

4 .2401 1.8639 -.5572 .3904 0. 0. 0.

#30|10|30

3 0. 0. -.1963495408

'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

4 .2401 1.8639 -.5572 .3904 0. 0. 0.

20. 8. -0.04276056667

4 .2401 1.8639 -.5572 .3904 0. 0. 0.

#30|10|30

3 0. 0. .1963495408

'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

4 .2401 1.8639 -.5572 .3904 0. 0. 0.

20. 8. 0.04276056667

4 .2401 1.8639 -.5572 .3904 0. 0. 0.

#30|10|30

3 0. 0. -.1963495408

'YMY'

'FAISCEAU'

'MATRIX'

1 0

'END'

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	

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	

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	

<b>BETATRON</b>	<b>Betatron core (momentum kick)</b>		
dp-kick	Momentum kick, relative to particle momentum.	-	E

<b>BINARY</b>	<b><i>BINARY</i>↔<i>FORMATTED</i> data converter</b>	
<i>NF</i> [ <i>.J</i> ], <i>NCol</i> , <i>NHDR</i>	Number of files to convert [ <i>READ</i> format type, see below], of data columns, of header lines.	$\leq 20, \leq 7, 0 - 9 \quad 3*I1$
<b>The next <i>NF</i> lines :</b>		
<i>FNAME</i>	Name of the file to be converted. File content is assumed binary <i>iff</i> name begins with “B_” or “b_”, assumed formatted otherwise.	A80

*READ* format, case of formatted input file :

If <i>FRM</i> not given	Format is '*'
If <i>FRM</i> =1	Format is '1X, 7E11.*'

*READ* format, case of binary input file :

Expected format is 7 column rows.



**BREVOL****1-D uniform mesh magnetic 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>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</i> [, <i>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>IORDRE</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)
  ENDIF
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).

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

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''</i> , etc., 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''</i> , etc.) boundaries	$\geq -1, 2^*$ no dim., cm [, $2^*$ no dim., cm, etc.]	I, 3*E [3*E, etc.]
<i>IORDRE</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

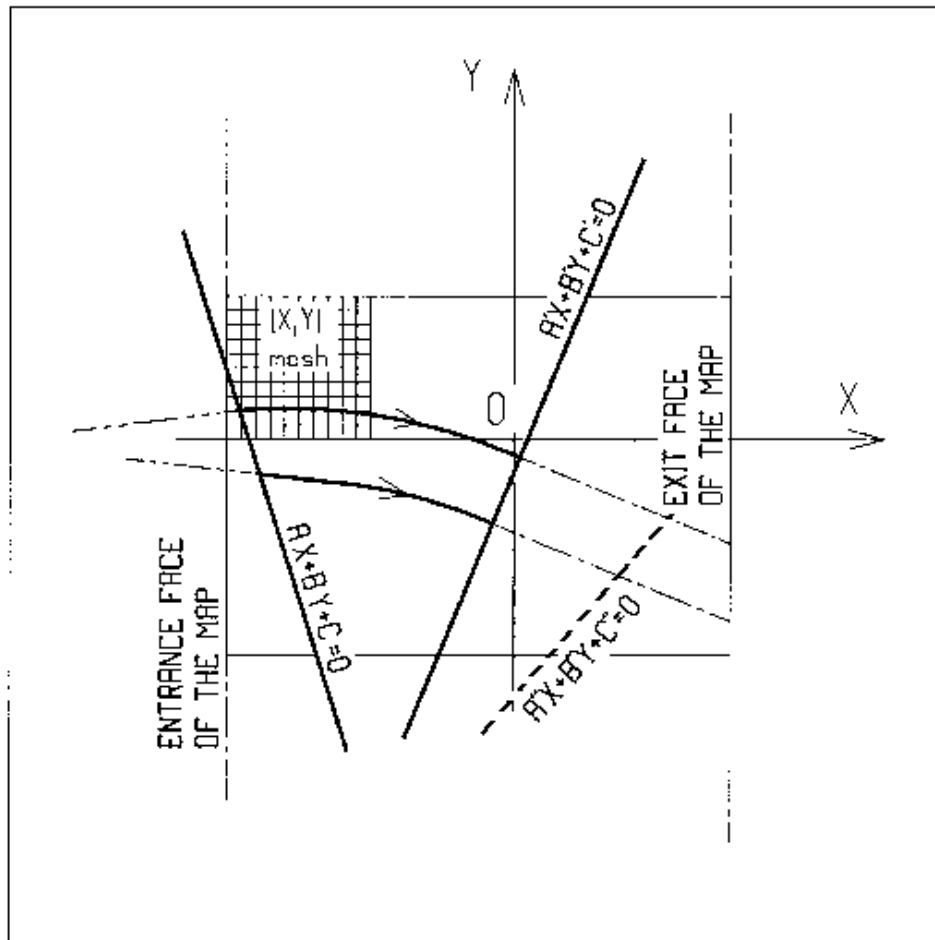
<sup>2</sup> *FNAME* (e.g., spes2.map) contains the field data. These must be formatted according to the following *FORTRAN* sequence :

```

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  $X(I)$  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.'



$OXY$  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</sup>	<b>Accelerating cavity</b> $\Delta W = qV \sin(2\pi h f \Delta t + \varphi_s)$ and other voltage and frequency laws.		
<i>IOPT</i> [ <i>i</i> ] [ <i>,PRINT</i> ] [ <i>,Facility</i> ]	Option, see below. <i>i</i> = 1 or 'PRINT' causes info printout to <code>zgoubi.CAVITE.Out</code> ; 'Facility' : 'CornellSynch' (only used with <i>IOPT=11</i> ) or 'eRHIC_RCS' (with <i>IOPT=11</i> ).	0-11	I[.I] [A, A]
<b>If IOPT=0</b>	Element inactive		
<i>X</i> , <i>X</i>	Unused.		
<i>X</i> , <i>X</i>	Unused.		
<b>If IOPT=1</b> <sup>2</sup>	$f_{RF}$ follows the timing law given by <i>SCALING</i>		
$\mathcal{L}$ , <i>h</i>	Reference closed orbit length ; harmonic number.	m, no dim.	2*E
$\hat{V}$ , <i>X</i>	R.F. peak voltage ; unused.	V, unused	2*E
<b>If IOPT=2</b>	$f_{RF}$ follows $\Delta W_s = q\hat{V} \sin \phi_s$ . Handles double RF systems		
$\mathcal{L}$ , <i>h</i> or $\{h_1, h_2\}$	Reference closed orbit length ; harmonic number(s).	m, no dim.	2*E
$\hat{V}$ or $\{\hat{V}_1, \hat{V}_2\}$ , $\phi_s$	R.F. peak voltage(s) ; synchronous phase.	V, rad	2*E
<b>If IOPT=3</b>	No synchrotron motion : $\Delta W = q\hat{V} \sin \phi_s$		
<i>X</i> , <i>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".		
$\mathcal{L}$ , $E_k$	Orbit length and kinetic energy at start of acceleration.	m, MeV	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</i> , $f_{RF}$	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.		
<i>L</i> , $f_{RF}$ , [ <i>, ID</i> ]	Cavity length ; RF frequency ; damp option, ID=0 (default : just a drift), or $ID = \pm 1$ ( $dE/E \ll 1$ hyp. and, resp <sup>ly</sup> damped or determinant=1), or $ID = \pm 2$ (no hyp. on $dE/E$ and, resp <sup>ly</sup> damped or determinant=1).	m, Hz	2*E [,I]
$\hat{V}$ , $\Phi_s$ , <i>IOP</i>	R.F. peak voltage ; synchronous phase ; matrix options.	V, rad, -2-+2	2*E, I
<b>If IOPT=11</b>	Acceleration in presence of SR loss.		
$\mathcal{L}$ , $E_k$	Orbit length and kinetic energy at start of acceleration.	m, MeV	2*E
$\hat{V}$ , $\Phi_s$ , $U_0$	R.F. peak voltage ; synchronous phase ; SR loss at first pass <sup>3</sup> .	V, rad	2*E

Routines concerned, mostly : `rcavit.f` (read input), `cavite.f` (cavity models).

<sup>1</sup> Use *PARTICUL* to declare mass and charge.

<sup>2</sup> For ramping the R.F. frequency following  $B\rho(t)$ , use *SCALING*, with family *CAVITE*.

<sup>3</sup> SR loss at subsequent turns is actualized using  $U \propto E^4$ .

<b>CHAMBR</b>	<b>Long transverse aperture limitation <sup>1</sup></b>		
<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[J], C1, C2, C3, 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 : $[C1, C2]$ , in Z : $[C3, C4]$	1-2[.0-1]	I[.I], 4*E

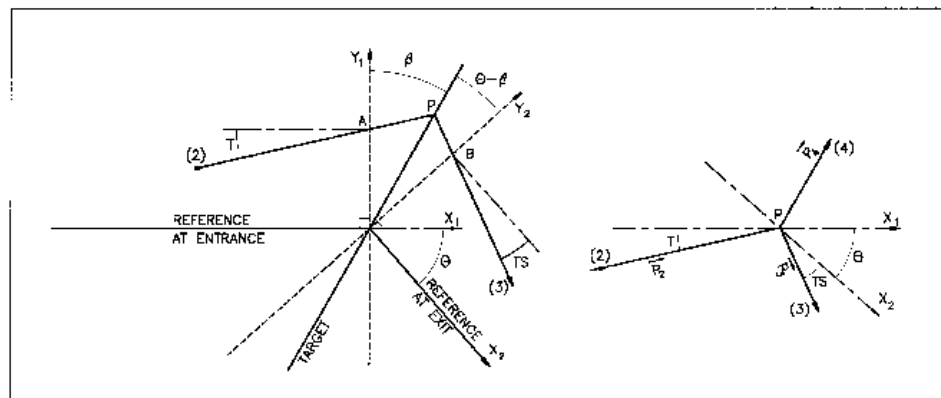
<sup>1</sup> Any particle out of limits is stopped.

<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,  $YC \simeq RM$ ).



**CIBLE, TARGET****Generate a secondary beam following target interaction**

$M_1, M_2, M_3, Q$ $T_2, \theta, \beta$	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 * mrad	3 * E
$BORO$	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

**COLLIMA****Collimator** <sup>1</sup>

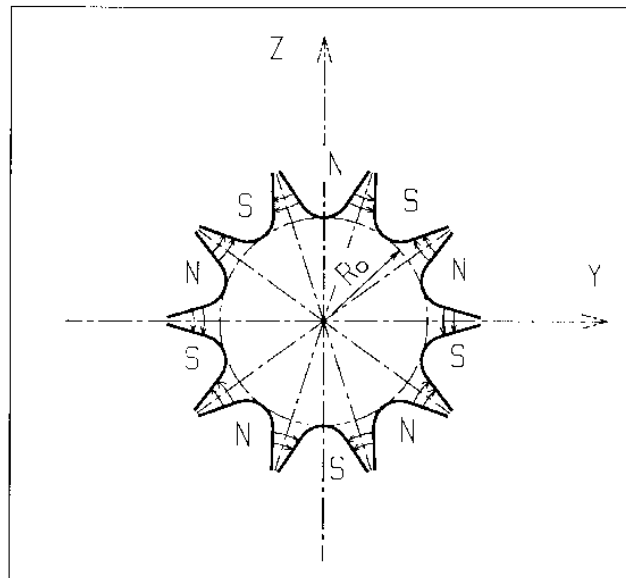
<i>IA</i>	0 : element inactive 1 : element active 2 : element active and print information on stopped particles	0-2	I
<b>Physical-space collimation</b>			
<i>IFORM</i> [.J], <i>C1</i> , <i>C2</i> , <i>C3</i> , <i>C4</i>	<i>IFORM</i> = 1 or 2 : rectangular or elliptical aperture ; <i>J</i> = 0, default : 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> .J, <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 collimation</b>			
<i>IFORM</i> , $\alpha$ , $\beta$ , $\epsilon/\pi$ , <i>N<sub><math>\sigma</math></sub></i>	<i>IFORM</i> = 11, 14 : horizontal collimation ; horizontal ellipse parameters (unused if 14) <sup>2</sup> , emittance, cut-off <i>IFORM</i> = 12, 15 : vertical collimation ; vertical ellipse parameters (unused if 15) <sup>2</sup> , emittance, cut-off <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> The rejection boundary is the *rms* ellipse matched to the particle distribution.



**DECAPOLE****Decapole 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	$2^*\text{cm}$ , kG	$3^*\text{E}$
$X_E, \lambda_E$	Entrance face : Integration zone extent ; fringe field extent ( $\lesssim 2R_0$ , $\lambda_E = 0$ for sharp edge)	$2^*\text{cm}$	$2^*\text{E}$
$NCE, C_0 - C_5$	$NCE = \text{unused}$ $C_0 - C_5 = \text{Fringe field coefficients such that}$ $G(s) = G_0/(1 + \exp P(s))$ , with $G_0 = B_0/R_0^4$ and $P(s) = \sum_{i=0}^5 C_i (s/\lambda)^i$	unused, $6^*\text{no dim.}$	I, $6^*\text{E}$
$X_S, \lambda_S$ $NCS, C_0 - C_5$	Exit face : see entrance face	$2^*\text{cm}$ $0-6, 6^*\text{no dim.}$	$2^*\text{E}$ I, $6^*\text{E}$
$XPAS$	Integration step	cm	E
$KPOS, XCE, YCE, ALE$	$KPOS=1$ : element aligned, $2$ : misaligned ; shifts, tilt.	$1-2, 2^*\text{cm}$ , rad	I, $3^*\text{E}$



**DIPOLE****Dipole magnet, polar frame**

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

dim.

data  
typeFIT  
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 $U_1 = \mp\infty$ 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	19-24
<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  (Note : $\lambda = 0$ , $\omega^- = -AT + \text{ACENT}$ and $\theta = 0$ for <u>sharp edge</u> )	2*deg, 4*cm	6*E	35-40
<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>IORDRE, 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 map, normally 2. Two options :	1-2	I	
<b>If KPOS = 2</b> <i>RE, TE, RS, TS</i>	Positioning as follows : Radius and angle of reference, respectively, at entrance and exit of the map.	cm, rad, cm, rad	4*E	64-67
<b>If KPOS = 1</b> <i>DP</i>	Automatic positioning of the map, by means of reference relative momentum	no dim.	E	

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

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

$NFACE, IC, IL$	Number of field boundaries $IC = 1, 2$ : print field map $IL = 1, 2[\times 10^n]$ , 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	2-3; 0-2; 0-2 $[\times 10^n]$ , 7 3*I	
$IAMAX, IRMAX$	Azimuthal and radial number of nodes of the mesh	$\leq 400, \leq 200$	2*I
$B_0, N, B, G$	Field and field indices	kG, 3*no dim.	4*E
$AT, ACENT, RM, RMIN, RMAX$	Mesh parameters : total angle of the map ; azimuth for positioning of EFBs ; reference radius ; minimum and maximum radii	2*deg, 3*cm	5*E
<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
$NC, C_0 - C_5, \text{shift}$	Unused ; $C_0$ to $C_5$ : see above ; EFB shift	0-6, 6*no dim., cm	I, 7*E
$\omega^+, \theta, R_1, U_1, U_2, R_2$	Azimuth of entrance EFB with respect to $ACENT$ ; wedge angle of EFB ; radii and linear extents of EFB (use $U_{1,2} = \mp\infty$ when $R_{1,2} = \infty$ )  (Note : $\lambda = 0, \omega^+ = ACENT$ and $\theta = 0$ for <u>sharp edge</u> )	2*deg, 4*cm	6*E
<b>EXIT FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)</b>			
$\lambda, \xi$ $NC, C_0 - C_5, \text{shift}$	Fringe field parameters	cm, unused 0-6, 6*nodim., cm	2*E 1, 7*E
$\omega^-, \theta, R_1, U_1, U_2, R_2$	Positioning and shape of the exit EFB  (Note : $\lambda = 0, \omega^- = -AT + ACENT$ and $\theta = 0$ for <u>sharp edge</u> )	2*deg, 4*cm	6*E
<b>If <math>NFACE = 3</math></b>			
<b>LATERAL FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)</b>			
Next 3 records <i>only</i> if $NFACE = 3$			
$\lambda, \xi$ $NC, C_0 - C_5, \text{shift}$	Fringe field parameters	cm, (cm) 0-6, 6*no dim., cm	2*E I, 7*E
$\omega^-, \theta, R_1, U_1, U_2, R_2, RM3$	Positioning and shape of the lateral EFB ; RM3 is the radial position on azimuth $ACENT$	2*deg, 5*cm	7*E
$NBS$	Option index for perturbations to the field map	normally 0	I
<b>If <math>NBS = 0</math></b>			
Normal value. No other record required			
<b>If <math>NBS = -2</math></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 <math>NBS = -1</math></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>IORDRE</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> <i>DP</i>	Automatic positioning of the map, by means of reference relative momentum	no dim.	E

**DIPOLES****Dipole magnet  $N$ -tuple, polar frame**

$$(i) B_Z = \sum_{i=1}^N B_{Z0,i} \mathcal{F}_i(R, \theta) (1 + b_{1i}(R - RM_i)/RM_i + b_{2i}(R - RM_i)^2/RM_i^2 + \dots)$$

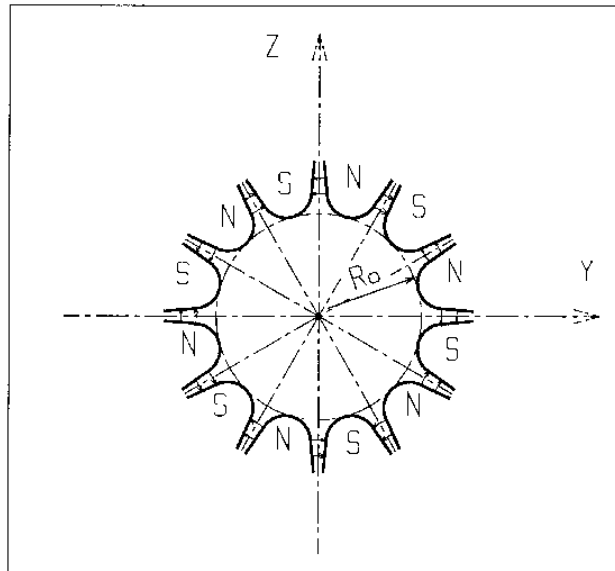
$$(ii) B_Z = B_{Z0,i} + \sum_{i=1}^N \mathcal{F}_i(R, \theta) (b_{1i}(R - RM_i) + b_{2i}(R - RM_i)^2 + \dots)$$

		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)
<b>ENTRANCE FIELD BOUNDARY</b>				
$g_0, \kappa$	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 $U_1 = \mp \infty$ 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> _____				
$KIRD[n], Resol$	If $KIRD=0$ : analytical computation of field derivatives ; $n=0$ : default, $B_Z$ formula (i) above, $n=1$ : $B_Z$ formula (ii), $n=7-9$ : 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 ; 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	

<sup>1</sup> Non-zero  $\delta RM$  requires  $KIRD = 2, 4$  or 25.

**DODECAPO****Dodecapole 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	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
$NCE, C_0 - C_5$	$NCE = \text{unused}$ $C_0 - C_5 = \text{Fringe field coefficients such that}$ $G(s) = G_0/(1 + \exp P(s))$ , with $G_0 = B_0/R_0^5$ 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

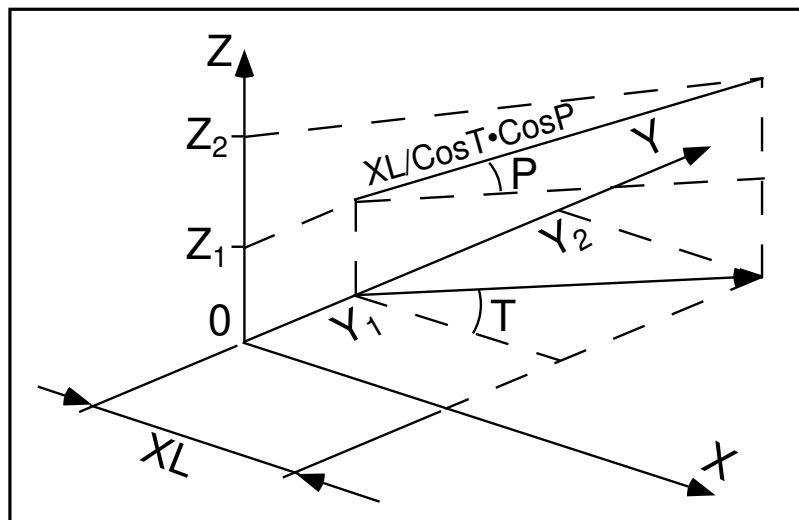


**DRIFT, ESL****Field free drift space** $XL$  [, 'split', N [,  $IL$ ]]

length; 'split', optional, will split into N segments ;  
 $IL=1$  or  $2$  (0 is default), optional, causes log of particle data in  
 zgoubi.res or zgoubi.plt.

cm

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 log to zgoubi.plt at the downstream end of a drift. 'DRIFT\_0' is *LABEL 1* here, '.plt' is *LABEL 2*.

```
'DRIFT'      DRIFT_0
30.3      split 1 2
```

```
'DRIFT'      DRIFT_0      .plt
30.3
```

**EBMULT<sup>1</sup>****Electro-magnetic multipole**

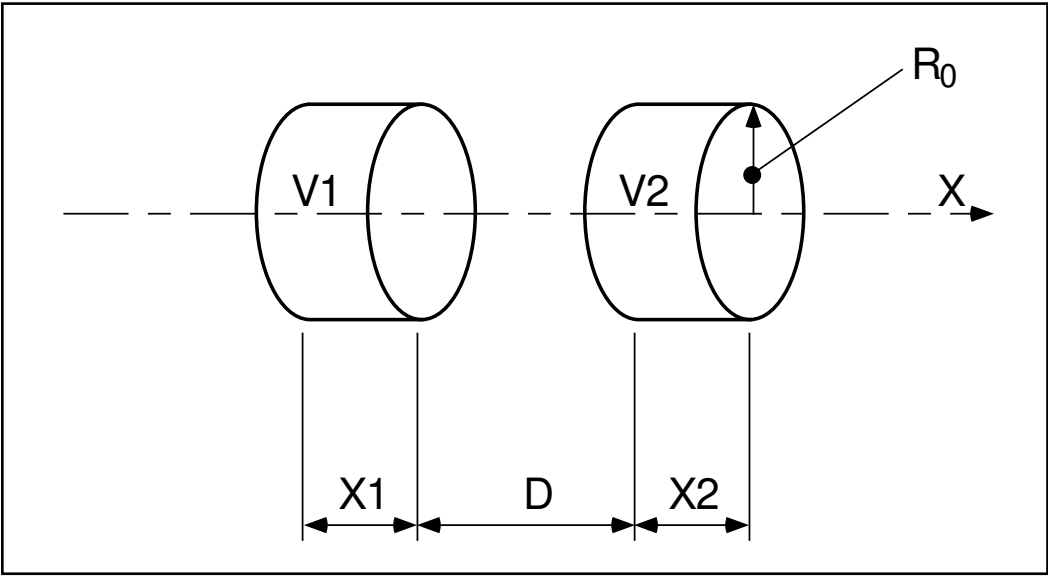
$IL$	$IL = 1, 2[\times 10^n]$ , 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	$0-2[\times 10^n]$ , 7	I
$XL, R_0, E_1, E_2, \dots, E_{10}$	<b>Electric poles</b> Length of element ; radius at pole tip ; field at pole tip for dipole, quadrupole, ..., 20-pole electric components	$2^*cm, 10^*V/m$	$12^*E$
$X_E, \lambda_E, E_2, \dots, E_{10}$	<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 = $\lambda_E * E_{10}$ (for any component : sharp edge if field extent is zero)	$2^*cm, 9^*no\ dim.$	$11^*E$
$NCE, C_0 - C_5$	same as <i>QUADRUPO</i>	$0-6, 6^*no\ dim.$	$I, 6^*E$
$X_S, \lambda_S, S_2, \dots, S_{10}$	<b>Exit face</b> Integration zone ; as for entrance	$2^*cm, 9^*no\ dim.$	$11^*E$
$NCS, C_0 - C_5$		$0-6, 6^*no\ dim.$	$I, 6^*E$
$R1, R2, R3, \dots, R_{10}$	Skew angles of electric field components	$10^*rad$	$10^*E$
$XL, R_0, B1, B2, \dots, B_{10}$	<b>Magnetic poles</b> Length of element ; radius at pole tip ; field at pole tip for dipole, quadrupole, ..., 20-pole magnetic components	$2^*cm, 10^*kG$	$12^*E$
$X_E, \lambda_E, E_2, \dots, E_{10}$	<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 = $\lambda_E * E_{10}$ (for any component : sharp edge if field extent is zero)	$2^*cm, 9^*no\ dim.$	$11^*E$
$NCE, C_0 - C_5$	same as <i>QUADRUPO</i>	$0-6, 6^*no\ dim.$	$I, 6^*E$

<sup>1</sup> Use *PARTICUL* to declare mass and charge.





<b>EL2TUB</b> <sup>1</sup>	<b>Two-tube electrostatic lens</b>		
<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>X</i> <sub>1</sub> , <i>D</i> , <i>X</i> <sub>2</sub> , <i>R</i> <sub>0</sub>	Length of first tube ; distance between tubes ; length of second tube ; inner radius	3*m	4*E
<i>V</i> <sub>1</sub> , <i>V</i> <sub>2</sub>	Potentials	2*V	2*E
<i>XPAS</i>	Integration step	cm	E
<i>KPOS</i> , <i>XCE</i> , <i>YCE</i> , <i>ALE</i>	<i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt.	1-2, 2*cm,	I, 3*E

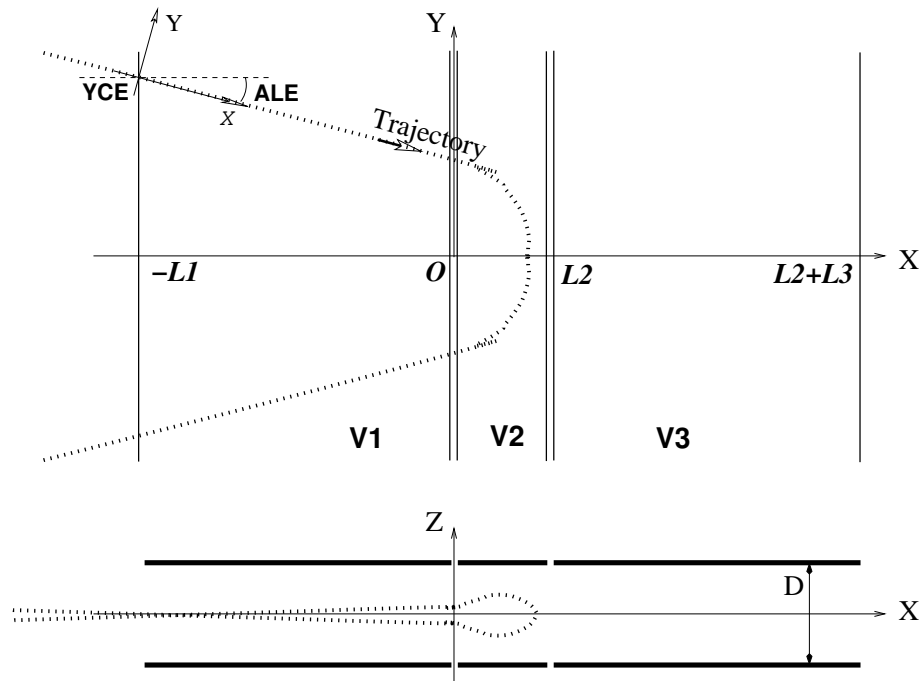


Two-electrode cylindrical electric lens.

<sup>1</sup> Use *PARTICUL* to declare mass and charge.

**ELMIR****Electrostatic N-electrode mirror/lens, straight slits**

$IL$	$IL = 1, 2[\times 10^n]$ , 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
$N, L1, \dots, LN, D, MT$	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
$V1, \dots, VN$	Electrode potentials (normally $V1 = 0$ )	$N^*V$	$N^*E$
$XPAS$	Integration step	cm	E
$KPOS, XCE, YCE, ALE$	$KPOS=1$ : element aligned ; $2$ : misaligned ; shifts, tilt ; $3$ : automatic positioning, $YCE$ = pitch, $ALE$ = half-deviation	$1-2, 2^*cm, rad$	I, $3^*E$

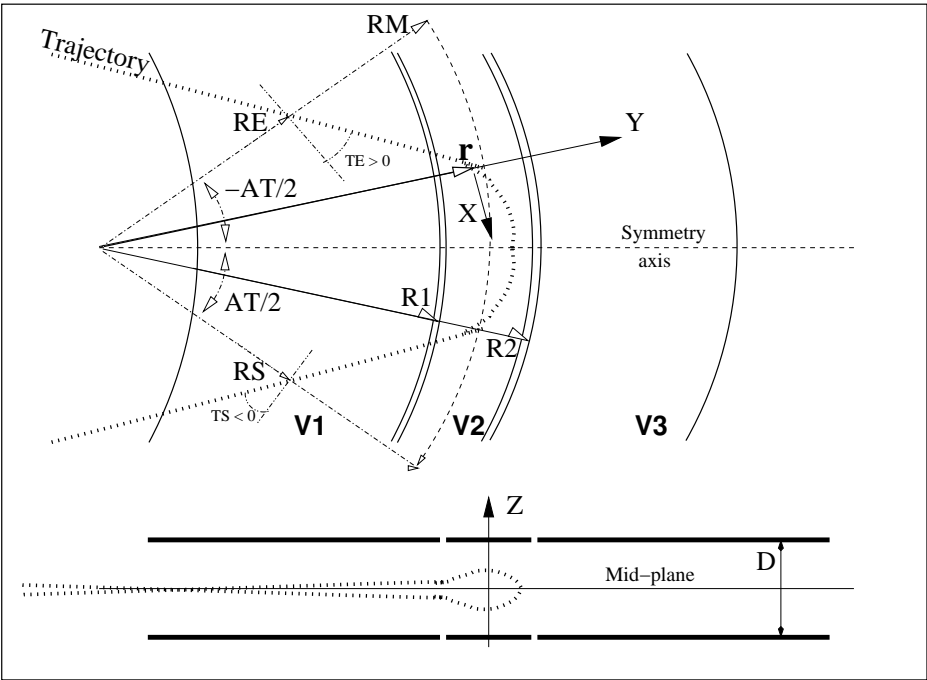


Electrostatic N-electrode mirror/lens, straight slits, in the case  $N = 3$ , in horizontal mirror mode ( $MT = 11$ ). Possible non-zero entrance quantities  $YCE$ ,  $ALE$  should be specified using *CHANGREF*, or using  $KPOS=3$  with  $YCE$  and  $ALE$ =half-deviation matched to the reference trajectory.

ELMIRC

Electrostatic N-electrode mirror/lens, circular slits

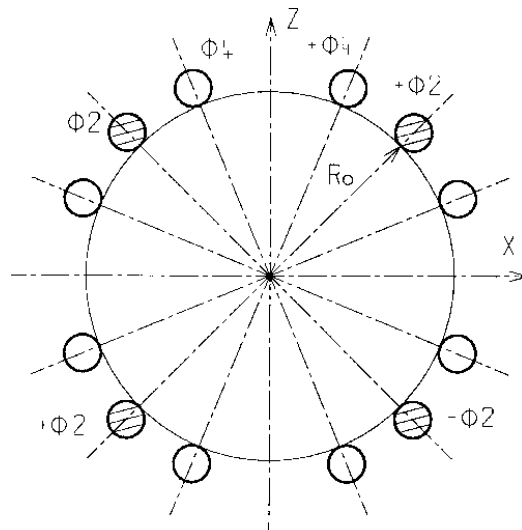
$IL$	$IL = 1, 2[\times 10^n]$ , 7 : print coordinates, fields, etc., step-by-step, in zgoubi.res (1), zgoubi.plt (2), zgoubi.impdev.out (7).	$0-2[\times 10^n]$ , 7	I
$R1, R2, AT, D$	Radius of first and second slits ; total deviation angle ; gap	$4*m$ $2*m, rad, m$	$4*E$ $4*E$
$V - VA, VB - V$	Potential difference	$2*V$	$2*E$
$XPAS$	Integration step	cm	E
$KPOS$ RE, TE, RS, TS	Normally $KPOS = 2$ for positioning ; Radius and angle at respectively entrance and exit.	1-2 cm, rad, cm, rad	I $4*E$



Electrostatic N-electrode mirror/lens, circular slits, in the case  $N = 3$ , in horizontal mirror mode.

**ELMULT**<sup>1</sup>**Electric multipole**

$IL$	$IL = 1, 2[\times 10^n]$ , 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
$XL, R_0, E1, E2, \dots, E10$	Length of element ; radius at pole tip ; field at pole tip for dipole, quadrupole, ..., dodecapole components	$2*\text{cm}$ , $10*\text{V/m}$	$12*\text{E}$
$X_E, \lambda_E, E_2, \dots, E_{10}$	<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)	$2*\text{cm}$ , 9*no dim.	$11*\text{E}$
$NCE, C_0 - C_5$	same as <i>QUADRUPO</i>	0-6, 6*no dim.	I, 6*E
$X_S, \lambda_S, S_2, \dots, S_{10}$	<b>Exit face</b> Integration zone ; as for entrance	$2*\text{cm}$ , 9*no dim.	$11*\text{E}$
$NCS, C_0 - C_5$		0-6, 6*no dim.	I, 6*E
$R1, R2, R3, \dots, R10$	Skew angles of field components	$10*\text{rad}$	$10*\text{E}$
$XPAS$	Integration step	cm	E
$KPOS, XCE, YCE, ALE$	$KPOS=1$ : element aligned, 2 : misaligned ; shifts, tilt.	1-2, $2*\text{cm}$ , 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>IORDRE</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>		
<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>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 $X, Y$ and $Z$ directions, $IZ = 1$ for single 2-D map ; <i>MOD</i> : operational and map <i>FORMAT</i> reading mode <sup>1</sup> $MOD \leq 19$ : Cartesian mesh ; $MOD \geq 20$ : 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> ( $K = 1, NF$ )	Names of the <i>NF</i> files that contain the 2-D maps, ordered from $Z(1)$ to $Z(NF)$ . If $MOD=0$ : a single map, superimposition of QF and QD ones, is built for tracking. If $MOD=1$ : a single map, <i>interpolated</i> from QF[ $x_F$ ] and QD[ $x_D$ ] ones, is built for tracking. If $MOD=22$ : a single map, superimposition of QF and QD ones, is built for tracking. If $MOD=24$ : field at particle is interpolated from a (QF,QD) pair of maps, closest to current ( $x_F, x_D$ ) value, taken from of set of (QF,QD) pairs registered in FNAME...			A80
<i>ID, A, B, C</i> [, $A', B', C'$ , $B''$ , etc., if $ID \geq 2$ ]	Integration boundary. Ineffective when $ID = 0$ . $ID = -1, 1$ 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>	If $IZ = 1$ : as in <i>CARTEMES</i> If $IZ \neq 1$ : unused	2, 25 or 4	I	
<i>XPAS</i>	Integration step	cm	E	
<i>KPOS, XCE, YCE, ALE</i>	$KPOS=1$ : 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.





**FAISCEAU****Print particle coordinates**

Print 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., during FIT, or REBELOTE).

**FAISCNL****Store particle coordinates in file FNAME***FNAME*<sup>1</sup>

Name of storage file  
(e.g., zgoubi.fai, or b\_zgoubi.fai for binary storage).

A80

**FAISTORE****Store coordinates every *IP* other pass [, at elements with appropriate label]**

*FNAME* <sup>1</sup>  
[,*LABEL*(s)]

Name of storage file (e.g. zgoubi.fai). Optional : a series of up to 10 label(s), (the first label of element(s) at the exit of which the store will occur) ; wild card accepted, in the form '\**LABEL*' or '*LABEL*\*' or '\**LABEL*\*'.  
If either *FNAME* or first *LABEL* is 'none' then *FAISTORE* is inhibited.  
Store occurs at all elements if first *LABEL* is 'all' or 'ALL'.

A80,  
[, 0-10\*A20]

*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.

**FFAG****FFAG magnet,  $N$ -tuple****UNDER DEVELOPMENT**

$$B_Z = \sum_{i=1}^N B_{Z0,i} \mathcal{F}_i(R, \theta) (R/R_{M,i})^{K_i}$$

$IL$	$IL = 1, 2[\times 10^n]$ , 7 : print coordinates, 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

Repeat  $N$  times the following sequence \_\_\_\_\_

$ACN, \delta RM,$ $B_{Z0}, K$	Azimuth for dipole positioning ; $R_{M,i} = RM + \delta RM$ ; field at $R_{M,i}$ ; index	deg, cm, kG, no dim.	4*E
----------------------------------	---	-------------------------	-----

**ENTRANCE FIELD BOUNDARY**

$g_0, \kappa$	Fringe field extent ( $g = g_0 (RM/R)^\kappa$ )	cm, no dim.	2*E
$NC, C_0 - C_5$ , shift	Unused ; $C_0$ to $C_5$ : fringe field coefficients ; 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 $ACN$ ; wedge angle of EFB ; radii and linear extents of EFB (use $U_1 = \mp\infty$ when $R_{1,2} = \infty$ )	2*deg, 4*cm	6*E

(Note :  $g_0 = 0$ ,  $\omega^+ = ACN$ ,  $\theta = 0$  and KIRD=0 for sharp edge)

**EXIT FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)**

$g_0, \kappa$	Fringe field parameters, see above	cm, no dim	2*E
$NC, C_0 - C_5$ , 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 :  $g_0 = 0$ ,  $\omega^- = -AT + ACN$ ,  $\theta = 0$  and KIRD=0 for sharp edge)

**LATERAL FIELD BOUNDARY to be implemented - following data not used**

$g_0, \kappa$		cm, no dim	2*E
$NC, C_0 - C_5$ , 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

End of repeat \_\_\_\_\_

$KIRD, Resol$	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 KPOS = 2</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 KPOS = 1</b> $DP$	Automatic positioning of the magnet, by means of reference relative momentum	no dim.	E
----------------------------	---	---------	---

**FFAG-SPI****Spiral FFAG magnet,  $N$ -tuple****UNDER DEVELOPMENT**

$$B_Z = \sum_{i=1}^N B_{Z0,i} \mathcal{F}_i(R, \theta) (R/R_{M,i})^{K_i}$$

$IL$	$IL = 1, 2[\times 10^n]$ , 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
Repeat $N$ times the following sequence _____			
$ACN, \delta RM, B_{Z0}, K$	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>			
$g_0, \kappa$	Fringe field extent ( $g = g_0 (RM/R)^\kappa$ )	cm, no dim.	2*E
$NC, C_0 - C_5$ , shift	Unused ; $C_0$ to $C_5$ : fringe field coefficients ; EFB shift	0-6, 6*no dim, cm	I, 7*E
$\omega^+, \xi$ , 4 dummies	Azimuth of entrance EFB with respect to $ACN$ ; spiral angle ; 4 $\times$ unused.	2*deg, 4*unused	6*E
<b>EXIT FIELD BOUNDARY (See ENTRANCE FIELD BOUNDARY)</b>			
$g_0, \kappa$	Fringe field parameters, see above	cm, no dim	2*E
$NC, C_0 - C_5$ , shift		0-6, 6*no dim, cm	1, 7*E
$\omega^-, \xi$ , 4 dummies		2*deg, 4*unused	6*E
<b>LATERAL FIELD BOUNDARY to be implemented - following data not used</b>			
$g_0, \kappa$		cm, no dim	2*E
$NC, C_0 - C_5$ , 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

End of repeat \_\_\_\_\_

**Integration boundaries** - next line is optional, starting with string `IntLim` :

<code>IntLim, ID, A, B, C</code> [ <code>A', B', C'</code> ]	Integration boundary. Line has to start with ' <code>IntLim</code> '. $ID = -1$ : 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 [ <code>id.</code> ]	I, 3*E [,3*E]
$KIRD, Resol$	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
$KPOS, RE, TE, RS, TS$	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

**FIT, FIT2**

*NV* [, *nofinal*]  
[, *noSYSout*]  
[, *save* [, *FileName*]]

**Fitting procedure**

*NV* : Number of physical parameters to be varied ; 'nofinal'  $\leq 20$  [, *nofinal*] I [, A7]  
avoids final run (default is : final run performed using *save* [string] [, A4 [, A80]]  
'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 zgoubi.FITVALS.out.

**For I = 1, NV**

*repeat NV times the following sequence*

either :

*IR, IP, XC, DV*

Number of the element in the structure ;  $\leq \text{MXL}^1$ ,  $\leq \text{MXD}^1$ , 2\*I, 2\*E  
number of the physical parameter in the element ;  $\pm \text{MXL.MXD}^2$ ,  
coupling switch (off = 0) ; variation range ( $\pm$ ). relative

or :

*IR, IP, XC, [V<sub>min</sub>, V<sub>max</sub>]*

*V<sub>min</sub>, V<sub>max</sub>* : lower and upper limits of the variable. see footnote<sup>3</sup> 2\*I, 3\*E

*NC* [, *Penalty* [, *ITER*]]<sup>4</sup>

Number of constraints [, *penalty* [, max. numb. of iterations]].  $\leq 20$  [,  $10^{-n}$  [, > 0]] I [, E [, I]]

**For I = 1, NC**

*repeat NC times the following sequence :*

*IC* [, *JC*], *I, J, IR, V*<sup>3</sup>,  
*WV, NP* [, *p<sub>i</sub>* (*i* = 1, *NP*)]

*IC, I* and *J* define the type of constraint (see table below) ; 0-5, 3\*(>0), I[,I1], 3\*I, 2\*E,  
*IR* : number of the element after which the constraint applies ; current unit, I, *NP*\*E  
*V* : value ; *W* : weight (the stronger the lower *WV*) 2\*no dim.,  
*NP* : number of parameters ; if *NP*  $\geq 1$ , *p<sub>i</sub>* (*i* = 1, *NP*) : curr. units  
parameter values.

<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 "integer.ijk" with integer  $\leq \text{MXD}$  and i, j, k 1-digit integers and such that  $ijk \leq \text{MXD}$ .

<sup>3</sup> *V* is in current **zgoubi** units in the case of particle coordinates (*i.e.*, cm, mrad,  $\mu\text{s}$ , 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.

Type of constraint	Parameters defining the constraints						Recommended [MC]OBJET ; comments
	IC	I	J	Constraint	Additional parameter(s) NP	Param. values, $pr_1 - pr_{NP}$	
Transported $\sigma$ -matrix ( $\sigma(s) = T\sigma(0)\tilde{T}$ )	0	1-6	1-6	$\sigma_{IJ}$ ( $\sigma_{11} = \beta_Y, \sigma_{21, 12} = \alpha_Y$ , etc.)			OBJET/KOBJ=5.1
Periodic $\sigma$ -matrix ( $\sigma = I \cos\mu + J \sin\mu$ ), tunes ( $N = 1 - 9$ for MATRIX block 1-9)	0.N ( $N \leq 9$ )	1-6	1-6	$\sigma_{IJ}$ ( $\sigma_{11} = \cos\mu_Y + \alpha_Y \sin\mu_Y$ , etc.)			OBJET/KOBJ=5 if $N = 1$ , OBJET/KOBJ=5.N if $N \geq 2$
		7	any	$\mu_Y/2\pi$			
		8	any	$\mu_Z/2\pi$			
		9	any	$\cos(\mu_Y)$			
First order transport coeffs. & determinants. Chromas.	1	1-6	1-6	Transport coeff. $R_{IJ}$			OBJET/KOBJ=5
		7	j	$j \neq 8$ : YY-determinant ; j=8: YZ-det.			
	1.1	7	7	$j \neq 7$ : ZZ-determinant ; j=7: ZY-det.			OBJET/KOBJ=5.2
		8	8	$dQ_Y / dp/p$ $dQ_Z / dp/p$			
Second order transport coeffs.	2	1-6	11-66	Transport coeff. $T_{I,j,k}$ ( $j = [J/10], k = J - 10[J/10]$ )			OBJET/KOBJ=6
Trajectory	3	1-IMAX	1-7	$F(J, I)$	{0 2	$I_1, I_2$	[MC]OBJET
coordinates (I = particle number; J=1-7 for D,Y,T,Z,P,S,time)	-1	1-7	$\langle F(J, i) \rangle_{i=I1, I2}$				1 $\rightarrow$ IMAX 1 $\leq I_1 \leq I_2 \leq$ IMAX
	-2	1-7	$Sup( F(J, i) )_{i=1, IMAX}$				
	-3	1-7	$Dist[F(J, I)_{i=I1, I2}]$	3			
	-4	1-7	$Dist[PU_i, i = 1, N]$	2			
	3.1	1-IMAX	1-7	$ F(J, I) - FO(J, I) $			
	3.2	1-IMAX	1-7	$ F(J, I) + FO(J, I) $			
	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	
Ellipse parameters	4	1-6	1-6	$\sigma_{IJ}$ ( $\sigma_{11} = \beta_Y$ , $\sigma_{12} = \sigma_{21} = \alpha_Y$ , etc.)			OBJET/KOBJ=8 ; MCOBJET/KOBJ=3
Number of particles	5	-1	any	$N_{survived}/IMAX$	1	$\epsilon_{Y,Z,X}/\pi$	OBJET
		1-3	any	$N_{in \epsilon_{Y,Z,X}}/N_{survived}$			MCOBJET
		4-6	any	$N_{in \text{ best } \epsilon_{Y,Z,X,rms}}/N_{survived}$			MCOBJET
Coordinates & fields, across optical elements (J=1, 2, 3 for $B_X, Y, Z$ )	7.1	1-IMAX	1-7	min. or max. of $F(J, I)$	1	1 or 2	[MC]OBJET
	7.2	1-IMAX	1-7	$\max(F(J, I)) - \min(F(J, I))$			
	7.3	1-IMAX	1-7	$\min F(J, I) + \max(F(J, I))$			
	7.6	1-IMAX	1-3	min. or max. value of $B_J$	1	1 or 2	
	7.7	1-IMAX	1-3	$\max(B_J) - \min(B_J)$			
	7.8	1-IMAX	1-3	$\min(B_J) + \max(B_J)$			
	7.9	1-IMAX	1-3	$\int B_J ds$			
Spin	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 rotation angle (rad)			OBJET/KOBJ=2
	10.3	1-IDMAX	1-3	rotation axis, X-, Y- or Z-component			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,  $\mu\text{s}$ ) ; 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.

### • Combining FIT and REBELOTE : An example

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 "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*.

```
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
warmSnake.map     ! AGS warm snake 3-D OPERA map
0 0 0 0
2
.1
2 0. .0 0. 0.

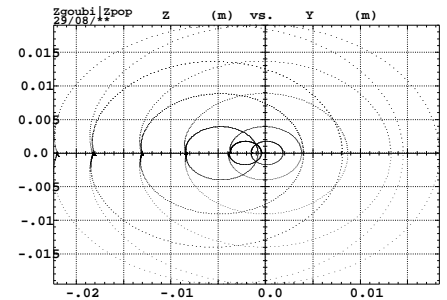
'FAISCEAU'

'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  $Y=Z=0$ ).

### • FIT[2] options : An example

*FIT[2]* options, “nofinal”, “save FileName”, “penalty” and “ITER”, are all specified in the example below.

```
An example with FIT options
'GETFITVAL'
zgoubi.FITVALS.In
'MCOBJET'1
57.36635309d3      ! reference rigidity
3.1
1
2 2 2 2 1 1
0. 0. 0. 0. 0.    4.61943880E-01
 2.703371  1.859656  1.48563786487e-99   9
-2.592712  2.421706  1.48563786487e-99   9
0. 1. 0.      9
12345 23456 34567
'PARTICUL'
0.51099892 1.60217653e-19 1.15965218076e-3 0.0 0.0
'PICKUPS'
 2
#StartRing MULT BD2S #EndRing
...
'DRIFT'      DRIF      HD
14.38218115
'MARKER'     #SSMid
'MARKER'     #EndRing
'FIT2'
4 save zgoubiFIT.result nofinal ! Variables saved in zgoubiFIT.result when FIT2 done. No final run.
!!! 4 save nofinal ! Would cause variables to be saved in default file zgoubi.FITVALS.out instead.
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-10 90                    ! penalty=1e-10. Maximum allowed calls to zgoubi 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
```



**FOCALE                      Particle coordinates and horizontal beam size at distance *XL***

<i>XL</i>	Distance from the location of the keyword	cm	E
-----------	---	----	---

**FOCALEZ                      Particle coordinates and vertical beam size at distance *XL***

<i>XL</i>	Distance from the location of the keyword	cm	E
-----------	---	----	---

<b>GASCAT</b>	<b>Gas scattering</b>		
<i>KGA</i>	Off/On switch	0, 1	I
<i>AI, DEN</i>	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  
FNAME='none' or FNAME='NONE', or if the file is not found.

A

<b>GOTO</b>	<b>Branching statement</b>		
	<b>Under development</b>		
OPTION	Can be 'PASS#' or 'GOBACK'	String	A
<b>If OPTION = PASS#</b> <i>LABEL_1, ..., LABEL_N</i>	One 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> ..	N strings	N*A20

- 
- Using **GOTO** : An example is given page 248

**HISTO****1-D histogram**

$J$ ,  $X_{\min}$ ,  $X_{\max}$ ,  
 $NBK$ ,  $NH$

$J$  = type of coordinate to be histogrammed ;  
the following are available :  
• current coordinates :  
1( $D$ ), 2( $Y$ ), 3( $T$ ), 4( $Z$ ), 5( $P$ ), 6( $S$ ),  
• initial coordinates :  
11( $D_0$ ), 12( $Y_0$ ), 13( $T_0$ ), 14( $Z_0$ ), 15( $P_0$ ), 16( $S_0$ ),  
• spin :  
21( $S_X$ ), 22( $S_Y$ ), 23( $S_Z$ ), 24( $< S >$ ) ;  
 $X_{\min}$ ,  $X_{\max}$  = limits of the histogram, in units  
of the coordinate of concern ;  $NBK$  = number of  
channels ;  $NH$  = number of the histogram (for  
independence of histograms of the same coordinate)

1-24, 2\*  
current units,  
< 120, 1-5

I, 2\*E, 2\*I

$NBL$ ,  $KAR$ ,  
 $NORM$ ,  $TYP$

Number of lines (= vertical amplitude) ;  
alphanumeric character ; normalization if  
 $NORM = 1$ , otherwise  $NORM = 0$  ;  $TYP = 'P'$  :  
primary particles are histogrammed, or 'S' :  
secondary, or Q : all particles - for use  
with *MCDESINT*

normally 10-40,  
char., 1-2, P-S-Q

I, A1, I, A1

**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**

**INCLUDE**

**File include statement**

*NBF*

Number of files to be included.

-

I

**NBF following lines (one file name per line) :**

*FNAME* [, [*LBL\_1A* [, *LBL\_2A*]]: Name of input zgoubi.dat-style file to be included. *LBL\_A* : A80 [, [A20], [A20]]  
[*LBL\_1B* [, *LBL\_2B*]] entrance branching tag, *LBL\_B* : exit branching tag.

● Using **INCLUDE** : An example.

In this example of an energy-recovery electron recirculator [71], the linac, spreader and combiner data lists are given in separate files, declared as *INCLUDEs*. 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 276.

```
Cornell CBETA prototype (of an eRHIC) FFAG arc ERL
'OBJECT'
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 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'</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>IORDRE</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.'

**MAP2D-E****2-D Cartesian uniform mesh field map - arbitrary electric field**

<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>IORDRE</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

**MATRIX****Calculation of transfer coefficients, periodic parameters***IORD, IFOC*[, *PRINT*] [, *coupled*]

Options :

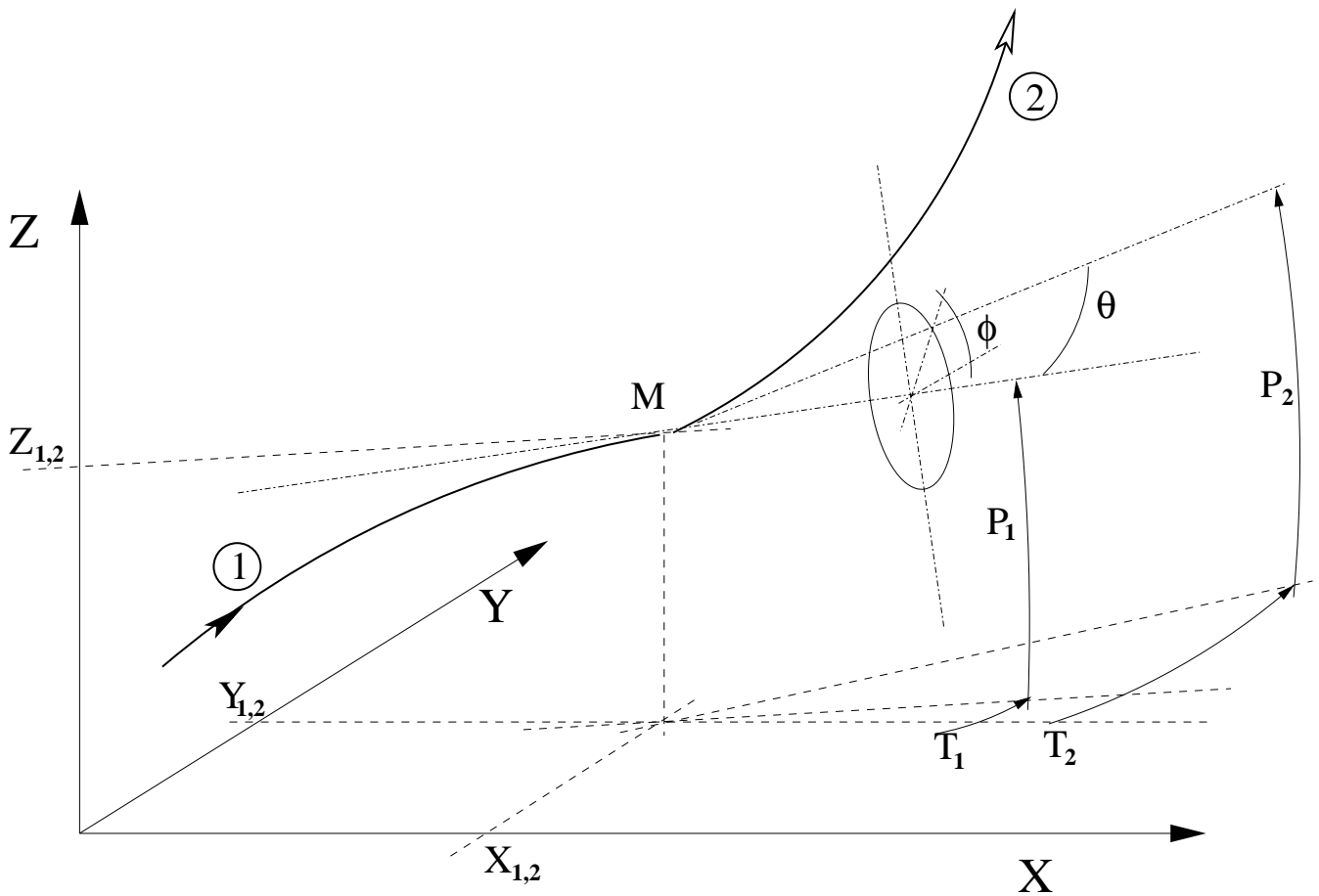
0-2, 0-1 or &gt; 10    2\*I [,A]

*IORD* = 0 : Same effect as *FAISCEAU**IORD* = 1, using *OBJET*, *KOBJ* = 5[.N] : 1st order transfer parameters,- if *KOBJ* = 5.1 : beam matrix, phase advance, otherwise- 1st order data (*cf.* *IFOC* value) for N different cases of optical reference.*IORD* = 2, using *OBJET*, *KOBJ* = 6 : 1st order transfer matrix [*R<sub>ij</sub>*],2nd order array [*T<sub>ijk</sub>*] and a few higher order transfer coefficients ;some periodic parameters in addition if *IFOC* > 10 ;*IORD* = 3, using *OBJET*, *KOBJ* = 6.1 : 1st order transfer matrix [*R<sub>ij</sub>*],2nd and 3rd order arrays [*T<sub>ijk</sub>*], [*T<sub>ijkl</sub>*].*IFOC* = 0 : matrix at actual location <sup>1</sup> ;*IFOC* = 1 : matrix at the closest first order horizontal focus <sup>1</sup> ;*IFOC* = 10 + *NPER* : same as *IFOC* = 0, and also calculates 1 or N sets (case respectively of *KOBJ* = 5 or *KOBJ* = 5.N,  $2 \leq N \leq 99$ ) of Twiss parameters, tune numbers, etc., assuming *zgoubi.dat* file describes one period of a *NPER*-period structure <sup>1</sup>.Occurrence of “*PRINT*” will cause printout to *zgoubi.MATRIX.out* file.Occurrence of “*coupled*”, in the periodic case (*IFOC* = 10 + *NPER*, 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 11-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 11-particle set is respectively particle #1, 12, 23, ... 1+(N-1)×11.

**MCDESINT**<sup>1</sup>**Monte-Carlo simulation of in-flight decay** $M1 \rightarrow M2 + M3$ [INFO,]<sup>2</sup>  $M2, M3, \tau_2$ <sup>3</sup>[Switch,] ; masses of the two decay products;  
COM lifetime of particle 2[-,]  $2 \cdot \text{MeV}/c^2, \text{s}$  [A4,]  $3 \cdot E$  $I1, I2, I3$ 

Seeds for random number generators

 $3 \cdot 10^6$  $3 \cdot I$ 

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> Presence of 'INFO' will cause more info on decay kinematics parameters to be printed zgoubi.res at each decay.

<sup>3</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).

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

<i>BORO</i>	Reference rigidity	kG.cm	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>		
<i><math>\delta Y, \delta T, \delta Z, \delta P, \delta X, \delta D</math></i>	Distribution widths, depending on <i>KY, KT</i> etc. <sup>1</sup>	m, rad, m, rad, m, no dim.	6*E
<i>N<sub><math>\delta Y</math></sub>, N<sub><math>\delta T</math></sub>, N<sub><math>\delta Z</math></sub>, N<sub><math>\delta P</math></sub>, N<sub><math>\delta X</math></sub>, N<sub><math>\delta D</math></sub></i>	Sorting cut-offs (used only for Gaussian density)	units of $\sigma_Y, \sigma_T$ , etc.	6*E
<i>N<sub>0</sub>, C<sub>0</sub>, C<sub>1</sub>, C<sub>2</sub>, C<sub>3</sub></i>	Parameters involved in calculation of P(D)	no dim.	5*E
<i>IR1, IR2, IR3</i>	Random sequence seeds	$3^* \simeq 10^6$	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
<i><math>\delta Y, \delta T, \delta Z, \delta P, \delta X, \delta D</math></i>	Width of the bars ( $\pm$ ) if uniform, Sigma value if Gaussian distribution	<i>ibidem</i>	6*E
<i>N<sub><math>\delta Y</math></sub>, N<sub><math>\delta T</math></sub>, N<sub><math>\delta Z</math></sub>, N<sub><math>\delta P</math></sub>, N<sub><math>\delta X</math></sub>, N<sub><math>\delta D</math></sub></i>	Sorting cut-offs (used only for Gaussian density)	units of $\sigma_Y, \sigma_T$ , etc.	6*E
<i>N<sub>0</sub>, C<sub>0</sub>, C<sub>1</sub>, C<sub>2</sub>, C<sub>3</sub></i>	Parameters involved in calculation of $P(D)$	no dim.	5*E
<i>IR1, IR2, IR3</i>	Random sequence seeds	$3^* \simeq 10^6$	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****On a phase-space ellipse <sup>1</sup>. Four lines follow :**
 $\alpha_Y, \beta_Y, \varepsilon_Y/\pi, N_{\sigma_{\varepsilon_Y}}$   
 $[, N'_{\sigma_{\varepsilon_Y}} \text{ if } N_{\sigma_{\varepsilon_Y}} < 0]^2$   
 $[, D_Y, D_T]$ 

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]

 $\alpha_Z, \beta_Z, \varepsilon_Z/\pi, N_{\sigma_{\varepsilon_Z}}$   
 $[, N'_{\sigma_{\varepsilon_Z}} \text{ if } N_{\sigma_{\varepsilon_Z}} < 0]^2$   
 $[, D_Z, D_P]$ 

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]

 $\alpha_X, \beta_X, \varepsilon_X/\pi, N_{\sigma_{\varepsilon_X}}$   
 $[, N'_{\sigma_{\varepsilon_X}} \text{ if } N_{\sigma_{\varepsilon_X}} < 0]^2$ 

Ellipse parameters and  
 emittance, X-D phase-space ; cut-off

no dim., m/rad,  
 m, units of  $\sigma(\varepsilon_X)$

4\*E [,E]

IR1, IR2, IR3

Random sequence seeds.

3\* $\simeq 10^6$ 

3\*I

In order for different runs to produce different series (meaning, overriding these  
 seeds) launch zgoubi using 'irank' argument: [pathTo]/zgoubi -irank 'someNonZeroInteger'.

<sup>1</sup> Similar possibilities, non-random, are offered with *OBJET*, KOBJ=8 (p. 259)

<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.





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
$XL, R_0, B1, B2, \dots, B10$ ,	Length of element ; radius at pole tip ; field at pole tip for dipole, quadrupole, ..., dodecapole components	$2*\text{cm}, 10*\text{kG}$	$12*\text{E}$	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)	$2*\text{cm}, 9*\text{no dim.}$	$11*\text{E}$	14-24
$NCE, C_0 - C_5$	same as <i>QUADRUPO</i>	$0-6, 6*\text{no dim.}$	$I, 6*\text{E}$	25-31
$X_S, \lambda_S, S_2, \dots, S_{10}$	<b>Exit face</b> Integration zone ; as for entrance	$2*\text{cm}, 9*\text{no dim.}$	$11*\text{E}$	32-42
$NCS, C_0 - C_5$		$0-6, 6*\text{no dim.}$	$I, 6*\text{E}$	43-49
$R1, R2, R3, \dots, R10$	Skew angles of field components	$10*\text{rad}$	$10*\text{E}$	50-59
$XPAS$	Integration step	cm	E	60
<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 $B1 \neq 0$ : entrance and exit frames are tilted wrt. the magnet by an angle of • either ALE if $ALE \neq 0$ • or $2 \text{Arcsin}(B1 XL / 2BORO)$ if $ALE=0$	$1-3, 2*\text{cm}, \text{rad}$	$I, 3*\text{E}$	61-64
<b>If KPOS = 4</b> $KPOS, XS, YS, ZR, ZS, YR$	X-, Y-shift, Z-rotation, Z-shift, Y-rotation Automatic ALE positioning as $KPOS = 3$ , and in addition possible X- or Y- or Z-misalignment or Z-, Y-rotation. (under development).	$4, \text{cm}, 2*(\text{cm}, \text{rad})$	$I, 5*\text{E}$	61-66

OBJET	Generation of an object		
<i>BORO</i>	Reference rigidity	kG.cm	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 : $YR \pm IY * dY, TR \pm IT * dT, ZR \pm IZ * dZ, PR \pm IP * dP$ $[ZR + IZ * dZ, PR + IP * dP \text{ if } KOBJ = 1.1], SR \pm IS * dS$ 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 $Y, T, Z, P, S$ 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		
<i>IMAX, IDMAX</i>	total number of particles ; number of distinct momenta (if $IDMAX > 1$ , 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(cm,mrad), cm, no dim., 2(m,rad), m, no dim.,	6*E, A1
<i>IEX</i> ( $I = 1, IMAX$ )	<i>IMAX</i> times 1 or -9. If $IEX(I) = 1$ trajectory <i>I</i> is ray-traced, it is not if $IEX(I) = -9$ .	1 or -9	<i>IMAXI</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 <i>robjet.f</i> ). 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``		
<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 $C (=Y, T, \dots)$ 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) (N in <i>KOBJ</i> =3.N determines storage FORMAT)		A80

<b>If KOBJ = 5[.N, <math>N \geq 1</math>]</b>	<b>Generation of 11 particles, or 11*N if <math>N \geq 2</math> (for use with MATRIX, IORD = 1)</b>		
$dY, dT, dZ, dP, dS, dD$	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 sused)	2(cm,mrad), cm, no dim.	6*E
$\diamond$ If $N=1$ ( $KOBJ = 5.1$ ) $\alpha_Y, \beta_Y, \alpha_Z, \beta_Z, \alpha_S, \beta_S,$ $D_Y, D'_Y, D_Z, D'_Z$	One additional data line : Initial beam ellipse parameters <sup>1</sup>	2(no dim.,m), ?, ?, 2(m,rad)	6*E, 4*E
$\diamond$ If $N \geq 2$ ( $KOBJ = 5.N$ ) $YR, TR, ZR, PR, SR, DR$	An additional N-1 data lines <sup>2</sup> Reference trajectory for block # i=2,N (block 1 is similar line above) ( $DR = B\rho/BORO$ ; $SR$ is not sused).	2(cm,mrad), cm, no dim.	6*E
<b>If KOBJ = 6</b>	<b>Generation of 61 particles (for use with MATRIX, IORD = 2)</b>		
$dY, dT, dZ, dP, dS, dD$	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 <sup>3</sup></b>		
$IY, IZ, IS$	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. 255).

### • OBJET, KOBJ=3, recovering from a crash : An example

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 320, 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.

```
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'
```

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
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) : 1244.3530 mass (MeV/c2) : 938.272
Spin components of each of the 1 particles, and rotation angle :

INITIAL                                FINAL

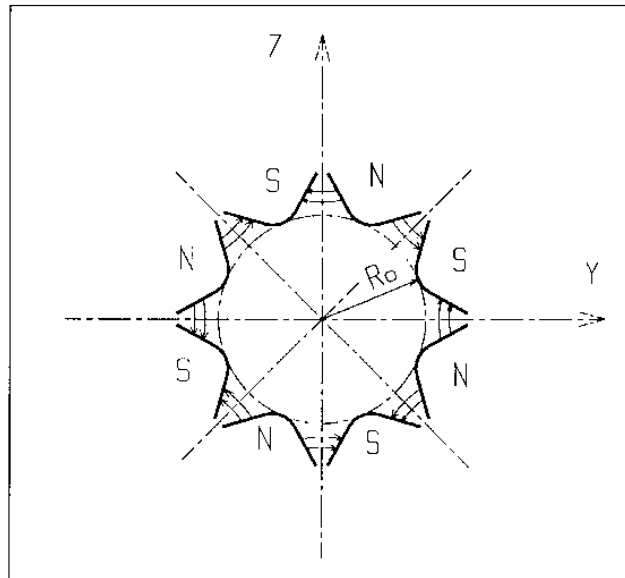
SX      SY      SZ      |S|      SX      SY      SZ      |S|      GAMMA (Si,Sf) (Z,Sf_X) (Z,Sf)
(deg.) (deg.) (deg.)
(Sf_X : projection of Sf on YZ plane)

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 'REBELOTE' for more)	$\leq 10^4$	I
$M_1 - M_6$	Rest masses of the bodies	$6 \cdot \text{GeV}/c^2$	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**

$IL$	$IL = 1, 2[\times 10^n]$ , 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)	$2^*cm$	$2^*E$
$NCE, C_0 - C_5$	$NCE$ = unused $C_0 - C_5$ = fringe field coefficients such that : $G(s) = G_0/(1 + \exp P(s))$ , with $G_0 = B_0/R_0^3$ and $P(s) = \sum_{i=0}^5 C_i (s/\lambda)^i$	any, $6^*no\ dim.$	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)  
 [, *IMP* or 'PRINT']  
 [, *coupled*]

*IOPT* = 0/1 : Off/On ; *NLBL* = number of labels (default is 1) ; 0-1[.1-10], -, I[.I],  
 'label' : can be 'all', 'ALL', or a list of *NLBL* first label(s) as -, - 1-10\*A20,  
 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\*'.  
*IMP* = 1, or presence of 'PRINT', will cause storage 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
```

**OPTIONS      Global or special options**

<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
-------------------	---	---------------	-----

**NBOP lines follow. Possible choices :**

'WRITE', <i>OPT</i>	<i>OPT</i> = ON/OFF : allows/inhibites (most) write statements to zgoubi.res	'WRITE', 'ON' or 'OFF'	2*A
---------------------	--	------------------------	-----

'CONSTY', <i>OPT</i>	<i>OPT</i> = ON : forces constant <i>Y</i> and <i>Z</i> across optical elements	'CONSTY', 'ON'/'OFF'	2*A
----------------------	---	----------------------	-----

● **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.
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 2 KIRD, Resol=IDB
0.523598775598
2 00. 0. 00. 0.
'FAISCEAU'

'END'

```



ORDRE	Taylor expansions order		
<i>IO</i>	Taylor expansions of $\vec{R}$ and $\vec{u}$ up to $\vec{u}^{(IO)}$ (default is $IO = 4$ )	2-5	I

**PARTICUL      Particle characteristics**

**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 :**

Type	ELECTRON, POSITRON, MUON+, MUON-, PROTON, DEUTERON, or HELION	A
------	---	---

Note : If  $M$  is of the form {M1 M2}, then when masses are assigned to particles from a previously defined object, 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) ; 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 “*LABEL<sub>i</sub>*” 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'</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>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.

**POLARMES****2-D polar mesh magnetic field map**

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'</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>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</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' :

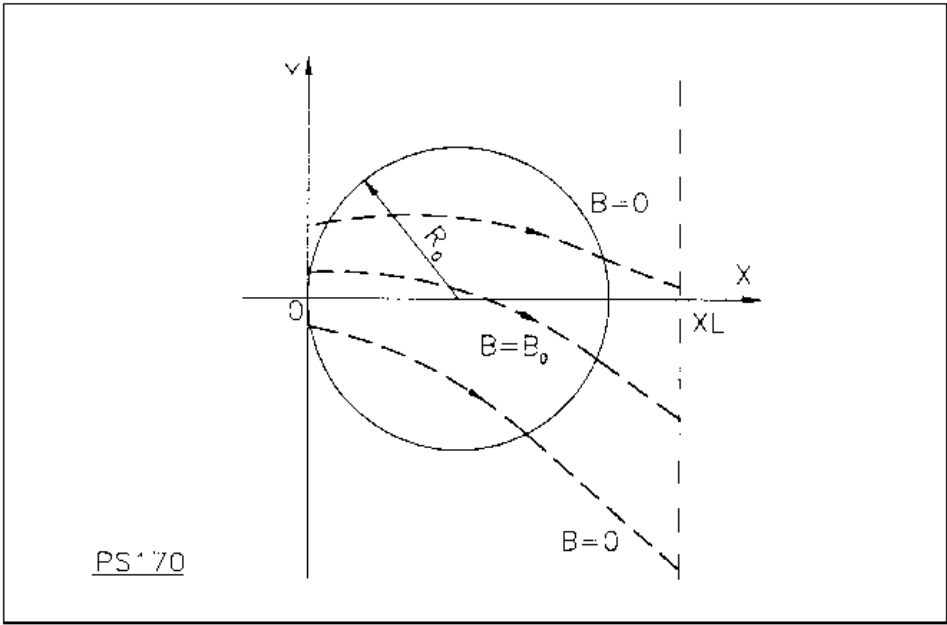
```

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			
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 of the element, radius of the circular dipole, field	$2^*\text{cm}$ , kG	$3^*\text{E}$	
XPAS	Integration step	cm	E	
KPOS, XCE, YCE, ALE	KPOS=1 : element aligned, 2 : misaligned ; shifts, tilt.	1-2, $2^*\text{cm}$ , rad	I, $3^*\text{E}$	



Scheme of the PS170 magnet simulation.

**QUADISEX****Sharp edge magnetic multipoles**

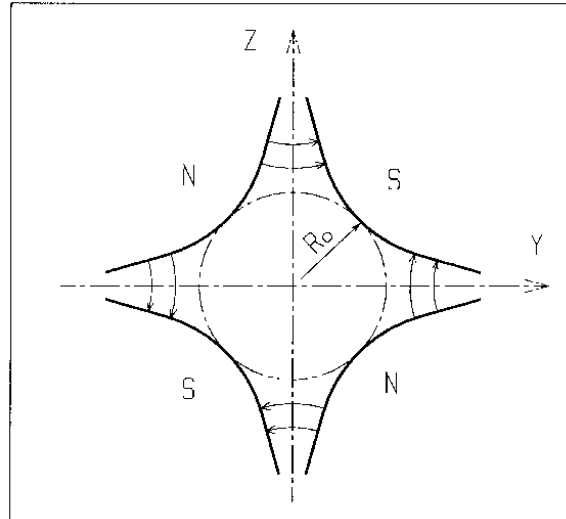
$$B_Z|_{Z=0} = B_0 \left( 1 + \frac{N}{R_0} Y + \frac{B}{R_0^2} Y^2 + \frac{G}{R_0^3} Y^3 \right)$$

<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</i> , <i>R</i> <sub>0</sub> , <i>B</i> <sub>0</sub>	Length of the element ; normalization distance ; field	2*cm, kG	3*E
<i>N</i> , <i>EB</i> 1, <i>EB</i> 2, <i>EG</i> 1, <i>EG</i> 2	Coefficients for the calculation of B. if <i>Y</i> > 0 : <i>B</i> = <i>EB</i> 1 and <i>G</i> = <i>EG</i> 1 ; if <i>Y</i> < 0 : <i>B</i> = <i>EB</i> 2 and <i>G</i> = <i>EG</i> 2.	5*no dim.	5*E
<i>XPAS</i>	Integration step	cm	E
<i>KPOS</i> , <i>XCE</i> , <i>YCE</i> , <i>ALE</i>	<i>KPOS</i> =1 : element aligned, 2 : misaligned ; shifts, tilt.	1-2, 2*cm, rad	I, 3*E

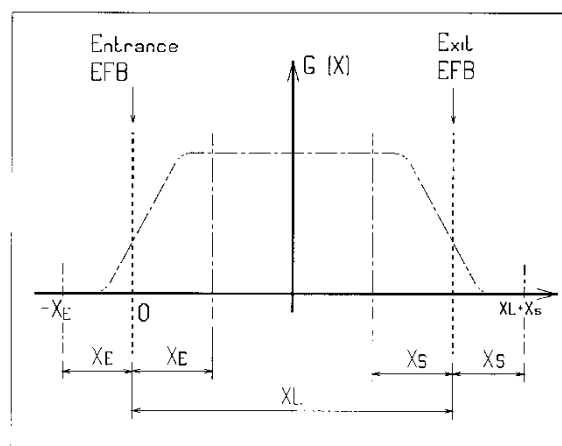


**QUADRUPO****Quadrupole magnet**

$IL$	$IL = 1, 2[\times 10^n]$ , 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	2*cm, kG	3*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 $P(s) = \sum_{i=0}^5 C_i (s/\lambda)^i$	any, 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
$KPOS, XCE, YCE, ALE$	$KPOS=1$ : 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  $XL$ , but trajectories are calculated from  $-X_E$  to  $XL + X_S$ , by means of automatic prior and further  $X_E$  and  $X_S$  translations.

**REBELOTE****'Do it again'**

*NPASS*, *KWRIT*, *K* [.n],  
*[IOPT]*,  
 [, *Label1* [, *Label1'*]]

*NPASS* : Number of runs ; *KWRIT* = 1.1 (resp. 0.0) switches (inhibits) *FORTRAN WRITE*s to .res and to screen ; arbitrary ; 0-1 ; 3\*I  
*K* option : 0, 99 ; [1[.any]] 2A10  
*K* = 0 : initial conditions (coordinates and spins) 0-1 ; 0, 22, 99 2A10  
 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 "Label1'" are expected ; the last pass (# *NPASS*+1) will terminate at element with Label1' 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>

<i>NPRM</i>	Number of parameters to be changed at each of the <i>NPASS</i> passes.	I
-------------	--	---

***NPRM* lines with the following sequence (tells parameters concerned, and for each, its successive values) :**

<i>LMNT</i> , <i>KPRM</i> , values	<i>LMNT</i> : two possibilities, (i) keyword number in <i>zgoubi.dat</i> sequence ; (ii) <i>KEYWORD</i> [, <i>LABEL</i> ], <i>i.e.</i> , keyword concerned followed optionnally by is label ; <i>KPRM</i> : parameter number under that keyword (similar nomenclature to <i>FIT</i> [2], see page 237) ; 'values' are the new parameter values. It can take two forms : - either an explicit list of <i>NPASS</i> values - or, first_value:last_value (interval will be (last_value - first_value)( <i>npass</i> -1))	-, -, <i>NV</i> *dim <sup>3</sup> 2*I, <i>NV</i> *E
------------------------------------	---	--

<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.

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 [72]. 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 ring turns. 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.

- **Combining REBELOTE and FIT** : An example is given page 239.

**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**

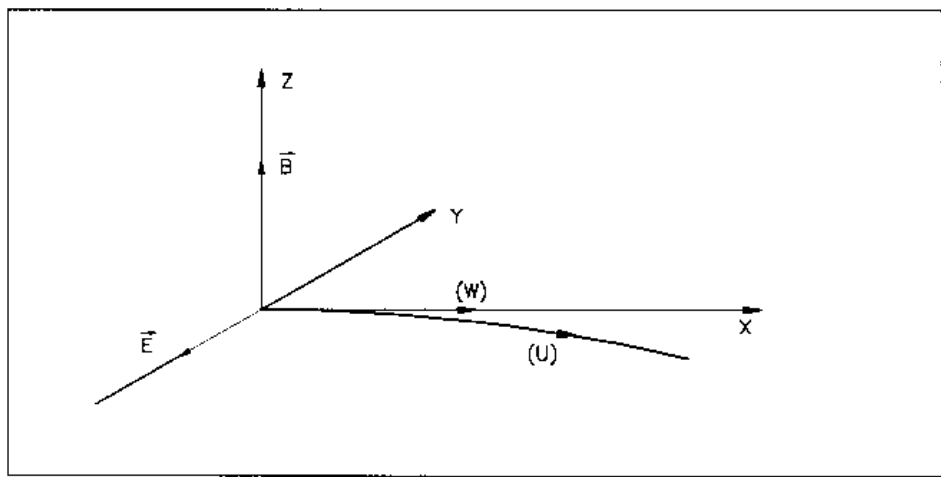
<i>IOPT</i> , <i>NFAM</i> [, <i>PRINT</i> ]	<i>IOPT</i> = 0 (inactive) or 1 (active) ; <i>NFAM</i> = number of families to be scaled. Occurrence of <i>PRINT</i> will cause printing of scaling infos in zgoubi.SCALING.out.	0-1, 1-45 [-]	I1, I2 [,A]
<b>For NF=1, NFAM :</b>	repeat <i>NFAM</i> times the following sequence :		
<i>NAMEF</i> [, <i>Lbl</i> [, <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*'.  <i>NT</i> > 0 : number of timings ; <i>NT</i> = -1 : field scaling factor updated by <i>CAVITE</i> ; <i>NT</i> = -2 : RF law in <i>CAVITE</i> is read from external data file.		A10 [,1-9*A20],A20]]
<i>NT</i>		-2, -1 or 1-10	I
<i>SCL</i> ( <i>I</i> ), <i>I</i> = 1, <i>NT</i>	Scaling values, and in particular : a single value if <i>NT</i> = -1 ; unused if <i>NT</i> = -2 ; if <i>NT</i> = 1 then <i>SCL</i> ( <i>I</i> ) = <i>SCL</i> (1) whatever the pass number.	relative	NT*E
<i>TIM</i> ( <i>I</i> ), <i>I</i> = 1, <i>NT</i>	Corresponding timings, in units of turns (1 if <i>NT</i> = -1 ; unused if <i>NT</i> = -2).	turn number	NT*I

**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**

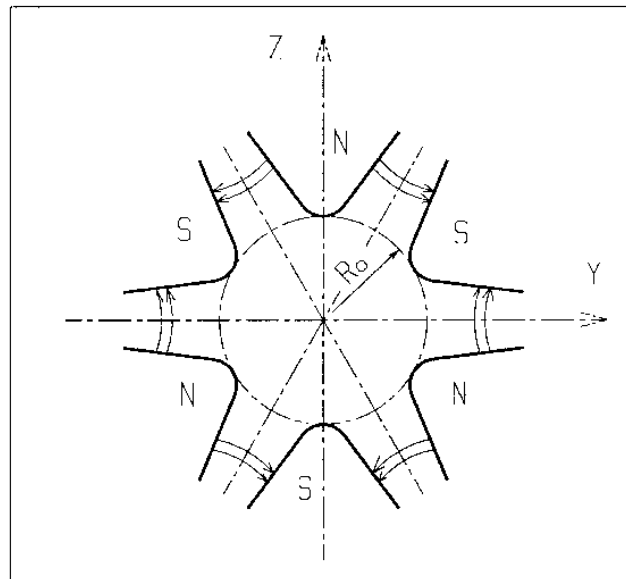
$$B_Z|_{Z=0} = B_0 \left( \frac{N}{R_0} Y + \frac{B}{R_0^2} Y^2 + \frac{G}{R_0^3} Y^3 \right)$$

<i>IL</i>	<i>IL</i> = 1, 2[ $\times 10^n$ ], 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</i> , <i>R</i> <sub>0</sub> , <i>B</i> <sub>0</sub>	Length of the element ; normalization distance ; field	2*cm, kG	3*E
<i>N</i> , <i>EB1</i> , <i>EB2</i> , <i>EG1</i> , <i>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</i> , <i>XCE</i> , <i>YCE</i> , <i>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	$2^*\text{cm}$ , kG	$3^*\text{E}$
$X_E, \lambda_E$	<b>Entrance face :</b> Integration zone ; fringe field extent ( $\lambda_E = 0$ for sharp edge)	$2^*\text{cm}$	$2^*\text{E}$
$NCE, C_0 - C_5$	$NCE = \text{unused}$ $C_0 - C_5 = \text{Fringe field coefficients such that}$ $G(s) = G_0/(1 + \exp P(s))$ , with $G_0 = B_0/R_0^2$ and $P(s) = \sum_{i=0}^5 C_i (s/\lambda)^i$	any, $6^*$ no dim.	I, $6^*\text{E}$
$X_S, \lambda_S$	<b>Exit face :</b> Parameters for the exit fringe field ; see entrance	$2^*\text{cm}$	$2^*\text{E}$
$NCS, C_0 - C_5$		$0-6, 6^*\text{no dim.}$	I, $6^*\text{E}$
$XPAS$	Integration step	cm	E
$KPOS, XCE, YCE, ALE$	$KPOS=1$ : element aligned, $2$ : misaligned ; shifts, tilt.	$1-2, 2^*\text{cm}$ , rad	I, $3^*\text{E}$

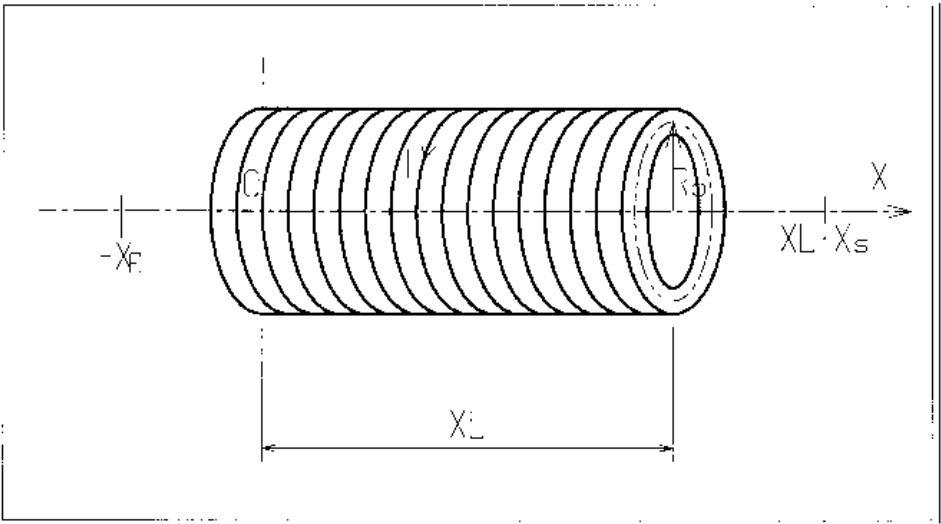


Sextupole magnet

SOLENOID

Solenoid

$IL$	$IL = 1, 2[\times 10^n]$ , 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



SPACECHARG	Space charge		
<i>LMNT, model [, PRINT]</i>	<i>LMNT</i> = 'all', 'none', or keyword ; model : 'KV', 'Gaussian' ; [optional : print out to zgoubi.SPACECHARG.out].	-, - [-,-]	2*A [,A]
$\lambda$	Linear charge density	$C/m$	E

**SPINR****Spin rotation**

<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 the X-axis and the spin precession axis ; spin precession angle around that axis.	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***

<i>FNAME</i> <sup>1</sup>	Name of storage file ( <i>e.g.</i> , zgoubi.spn)	A80
---------------------------	--	-----

<b>SPNSTORE</b>	Store spin coordinates every $IP$ other pass
-----------------	--

<i>FN</i> AME <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 <i>REBELOTE</i> with $NPASS \geq IP - 1$ ).	I
------	---	---

<b>SPNPRT [,PRINT]</b>	<b>Print spin coordinates, compute spin matrix</b>
<b>[,MATRIX]</b>	

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. 6.5.10, p. 180).

<sup>1</sup> *FNAME* = 'none' will inhibit printing.

<sup>2</sup> If first *LABEL* = 'none' then printing will be inhibited.

**SPNTRK**<sup>1</sup>**Spin tracking****KSO** [.KSO2]**KSO**=0 : spin tracking [switched] off<sup>2</sup> ; **KSO**= -1 : spin tracking resumes. Otherwise : as stated below.-1 or 0 or 1  
1-3 Or 4[.1] or 5**If KSO = 1 – 3****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)**If KSO = 4**Repeat **IMAX** times (corresponding to the **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

**If KSO = 4.1** $S_X, S_Y, S_Z$  $X, Y$  and  $Z$  components of the initial spins. These will be assigned to all particles.

3\*no dim.

3\*E

**If KSO = 4.2****FNAME**<sup>3</sup>File name (*under development*)

A80

**If KSO = 5**

Random distribution in a cone (see figure)

Enter the following two sequences :

 $TO, PO$ 

Angles of average polarization direction.

2\*rad

2\*E

 $A, \delta A$  $A$  = angle of the cone around ( $TO, PO$ ) ;  $\delta A$  = standard deviation of distribution around  $A$ 

2\*rad

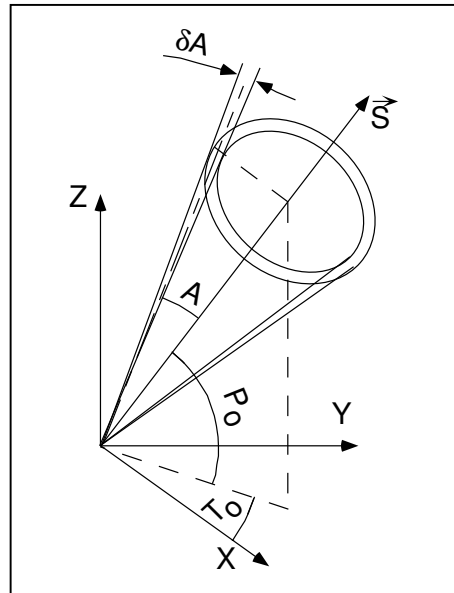
2\*E

**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 as 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).

SRLOSS	Synchrotron radiation energy loss		
<i>KSR</i> [ <i>i</i> ] [, <i>PRINT</i> ] [, <i>KSOK</i> ]	On/off switch ; $i = 1$ for info output into <code>zgoubi.SRLOSS.Out</code> ; optional, <i>PRINT</i> , anywhere beyond <i>KSR</i> : has the same effect as $i = 1$ ; 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]
<i>STR1</i> [{ <i>lblst</i> }] [, <i>STR2</i> [, <i>List</i> ]]	Apply SR loss to <i>STR1</i> = ' <i>ALL</i> ' or ' <i>all</i> ' or a particular KEYWORD ; { <i>lblst</i> }={ <i>lbl1</i> , <i>lbl2</i> , ..., <i>lbl5</i> } : a list of up to 5 labels to which this applies. Optional : <i>STR2</i> = ' <i>scale</i> ' 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]
Option, seed	Option = 0 or 1 : SR causes dp only ; option = 2 : SR causes dp and angle kick (under development).	1 – 3, $> 10^5$	I, I6

**SRPRNT****Print SR loss statistics** into zgoubi.res



**SVDOC****SVD orbit correction  
UNDER DEVELOPEMENT - Feb. 2019**

<i>LABEL1</i> [, <i>LABEL2</i> , [...]]	The N 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, in the form ‘ <i>*LABEL</i> ’ or ‘ <i>LABEL*</i> ’ or ‘ <i>*LABEL*</i> ’.	N string(s)	N*A20
---	---	-------------	-------

<b>SYNRAD</b>	<b>Synchrotron radiation spectral-angular densities</b>		
<i>KSR</i>	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.

● **SYSTEM** : An example (an additional example can be found page 239)

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 21
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
AGSQUD QH_*      !# of params 2Bchanged.      (FIT#15)
-1          1          15 0.0
1.000000
1
AGSQUD QV_* QP_*      !# of params 2Bchanged.      (FIT#15)
-1          1          15 0.0
1.000000
1
MULTIPOL QJUMP_*
-1
7.06936680E+00
1
MULTIPOL SXH_*
-1
7.06936680E+00
1
MULTIPOL SXV_*
-1
7.06936680E+00
1
AGSMM MM_F08CD MM_F09BF MM_G02BF MM_G03CD MM_G16AD MM_G17CF ! blw1 / G9 bump
-1          1          22 -.0762e-99          ! Amp. (FIT #31)
1.0
1
AGSMM MM_H04CD MM_H05AF MM_H18CF MM_H19BD MM_I12BD MM_I13CF ! blw1 / H11 bump
-1          1          22 -.06093e-99          ! Amp. (FIT #35)
1.0
1
MULTIPOL COH1
1.10
./Csnk3D/Hlx68.2_Sol42.3/CHREF+_dipolCORR.scal
1 4
MULTIPOL COV1
1.10
./Csnk3D/Hlx68.2_Sol42.3/CHREF+_dipolCORR.scal
1 5
MULTIPOL COH2
1.10
./Csnk3D/Hlx68.2_Sol42.3/CHREF+_dipolCORR.scal
1 6
MULTIPOL COV2
1.10
./Csnk3D/Hlx68.2_Sol42.3/CHREF+_dipolCORR.scal
1 7
MULTIPOL WOH1
1.10
./Csnk3D/Hlx68.2_Sol42.3/CHREF+_dipolCORR.scal_51.8_0.0
1 4
MULTIPOL WOV1
1.10
./Csnk3D/Hlx68.2_Sol42.3/CHREF+_dipolCORR.scal_51.8_0.0
1 5
MULTIPOL WOH2
1.10
./Csnk3D/Hlx68.2_Sol42.3/CHREF+_dipolCORR.scal_51.8_0.0
1 6
MULTIPOL WOV2
1.10
./Csnk3D/Hlx68.2_Sol42.3/CHREF+_dipolCORR.scal_51.8_0.0
1 7
CHANGREF WSNKE
1.12
./Csnk3D/Hlx68.2_Sol42.3/CHREF+_dipolCORR.scal_51.8_0.0
1 1 1 3
CHANGREF WSNKO
1.12
./Csnk3D/Hlx68.2_Sol42.3/CHREF+_dipolCORR.scal_51.8_0.0
1 1 1 2
CHANGREF CSNKE
1.12
./Csnk3D/Hlx68.2_Sol42.3/CHREF+_dipolCORR.scal
1 1 1 3
CHANGREF CSNKO
1.12
./Csnk3D/Hlx68.2_Sol42.3/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.
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.
3.0 Dip MM_A01BF
4 0. 0. 0. 0. 0.
[.....]
'TOSCA'
0 0
1.e-3 100. 100. 100.
HEADER_4 csna
281 29 29 15.2 .682 .423
./Csnk3D/Hlx68.2_Sol42.3/b_table_for_Helix_3T.tab
./Csnk3D/Hlx68.2_Sol42.3/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 csna
281 29 29 15.2 .518 1.e-30
./Csnk3D/Hlx68.2_Sol42.3/b_table_for_Helix_3T.tab_2
./Csnk3D/Hlx68.2_Sol42.3/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'

```

**TOSCA****2-D and 3-D Cartesian or cylindrical mesh field map** (page 158 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 <i>MOD</i> =15 and <i>MOD</i> =24). Include “RHIC_helix” to normalize <i>BNORM</i> to measured value (B field versus helix current, in program <code>toscac.f</code> ).		A80	
<i>IX, IY, IZ</i> , <sup>1</sup> <i>MOD</i> [ <i>MOD2</i> [, <i>a(i)</i> , <i>i</i> = 1, <i>MOD2</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 a 2-D map ; <i>MOD</i> and <i>MOD2</i> : field map style, see table next page ; <i>a(i)</i> = 1, <i>MOD2</i> ) are the field normalisation factors.	$\leq MXX, \leq MXY, \leq IZ$ , [0-22.1-9], -	3*I, I[, I [, <i>MOD2</i> *E]]	20-24 [, 25-27
Next NF lines : <i>FNAME</i> <sup>3</sup>	Map file name(s), one line per name. If <i>MOD</i> =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 <i>MOD</i> =1 : $NF = IZ$ , no symmetry assumed ; $Z(1) = Z_{max}$ , $Z(1 + [IZ/2]) = 0$ and $Z(NF) = -Z_{max}$ . If <i>MOD</i> =12 : a single <i>FNAME</i> file contains the all 3-D volume. <i>MOD</i> =15, 20-22, etc. : see table next page.		A80	
<i>ID, A, B, C</i> [, <i>A', B', C'</i> , <i>A''</i> , etc., if $ID \geq 2$ ]	Integration boundary. Ineffective when $ID = 0$ . $ID = -1, 1$ or $\geq 2$ : as for <i>CARTEMES</i>	$\geq -1$ , cm, 2*no dim. [, <i>idem</i> ]	I, 3*E [, 3*E, etc.]	30-33 [, 34-36, .
<i>IORDRE</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> *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(J), 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,I,I)*, *BY(J,I,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(J), 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

<sup>3</sup> For binary files, *FNAME* must begin with 'B\_' or 'b\_'.

• The various **IZ**, **MOD** and **MOD2** possibilities, 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)

**'Exemple' 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.

**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 fmapw.f/fmapr3).	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 - 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 IZ equally Z-spaced $(X, Y)$ planes (no symmetry assumed).	1 - 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 IZ equally Z-spaced $(X, Y)$ planes.	1 - 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 - 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 fmapw.f).	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 Z-spaced $(X, Y)$ planes. MOD=23 : special, test code (see fmapw.f).	1 - 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 IZ equally Z-spaced $(X, Y)$ planes.	1	$\theta$ , $R$ , $Z$ , $B_\theta$ , $BR$ , $BZ$	

TRANSMAT	Matrix transfer		
<i>IORDRE</i>	Transfer 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 <math>IORDRE = 2</math></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

<b>TRAROT</b>	<b>Translation-Rotation</b>		
$TX, TY, TZ,$ $RX, RY, RZ$	Translations, rotations	3*m, 3*rad	6*E



**TWISS**<sup>1</sup>**Calculation of periodic optical parameters. Log to zgoubi.TWISS.out**

*KTW*[*KTW2*],  
*FacD*, *FacA* [, *coupled*]

*KTW* = 0/1/2/3 : Off / as *MATRIX* / add computation of  
 chromaticities / add computation of anharmonicities.

0-3[.1], 2\*any  
 [,coupled]

I1[.I1],  
 2\*E [,A]

*KTW2* : default is NO initial computation of closed orbit, whereas  
*KTW2* = 1 will cause computation of initial closed orbit, prior to TWISS-ing ;  
*FacD* × *D* = *Bρ*/*BORO* value applied, with *D* the momentum sampling  
 in *OBJET* ; *FacA* : unused ;  
 “*coupled*” will cause use of coupled formalism for matrix computation from rays.

<sup>1</sup> *TWISS* may be preceded by *FIT*[2]. This allows finding the closed orbit prior to doing *TWISS*. This is illustrated in the example page 298.

• **Example** : orbit finding using *FIT*, followed by a *TWISS* procedure

- “*KTW=2*” under *TWISS* will cause 4 successive runs of the optical sequence and will result in the delivery of a *zgoubi.TWISS.out* file, and in the print out to *zgoubi.res* of
- the on-momentum matrix of the optical structure,
  - off-momentum matrices at  $\frac{dp}{p} = \pm FacD * \delta D$ ,
  - the Twiss parameters in the hypothesis of a stable periodic structure,
  - the momentum compaction, chromaticities, etc.

Un-comment the *FIT* instruction below, and it will first find the orbit which will then be the reference trajectory in *OBJET[KOBJ=5]*, in generating the 11 sample rays necessary for *TWISS*.

```
'OBJET'
20015.55      ! Rigidity (kG.cm) for 6 GeV electrons.
5             ! Generate an 11 particle set. Normally used for MATRIX or TWISS computation.
.001 .01 .001 .01 0. .0001 ! Coordinate sampling for MATRIX computation : delta-Y, -T, -Z etc.
0. 0. 0. 0. 0. 1.      ! Coordinates of the reference orbit: Y- ,T- ,Z- ,P-, s-ref (unused), D-ref
                        ! this is the coordinates of particle numb 1 of the 11-particle set.
.....
Optical sequence here
.....
! 'FIT'
! 2
! 1 30 0 [-1.,1.]      ! vary Y-ref
! 1 31 0 [-1.,1.]      ! vary T-ref
! 4 1e-10
! 3.1 1 2 #End 0. 1. 0   ! particle 1: Y initial = Y final
! 3.1 1 3 #End 0. 1. 0   ! particle 1: T initial = T final
! 3.1 1 4 #End 0. 1. 0   ! particle 1: Z initial = Z final
! 3.1 1 5 #End 0. 1. 0   ! particle 1: P initial = P final
'FAISCEAU'
'TWISS'
2 1. 1.          ! KTW = 3, FacD = 1
'END'
```

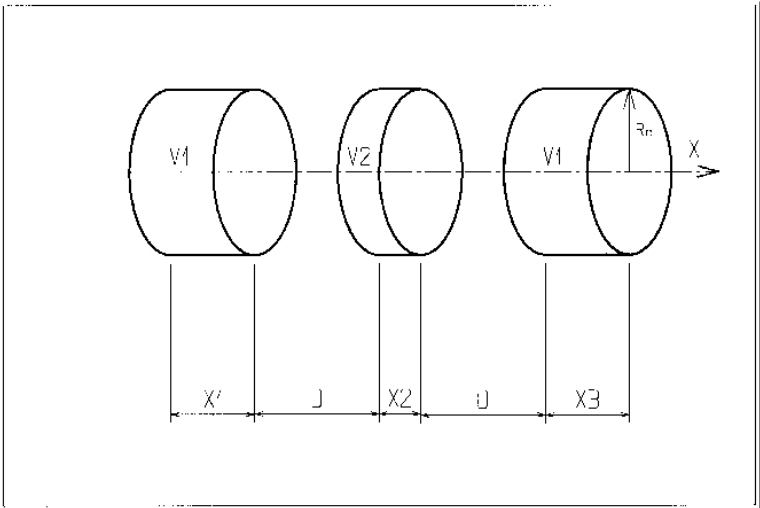
**UNDULATOR**                      **Undulator magnet**

*Under development, to be documented*

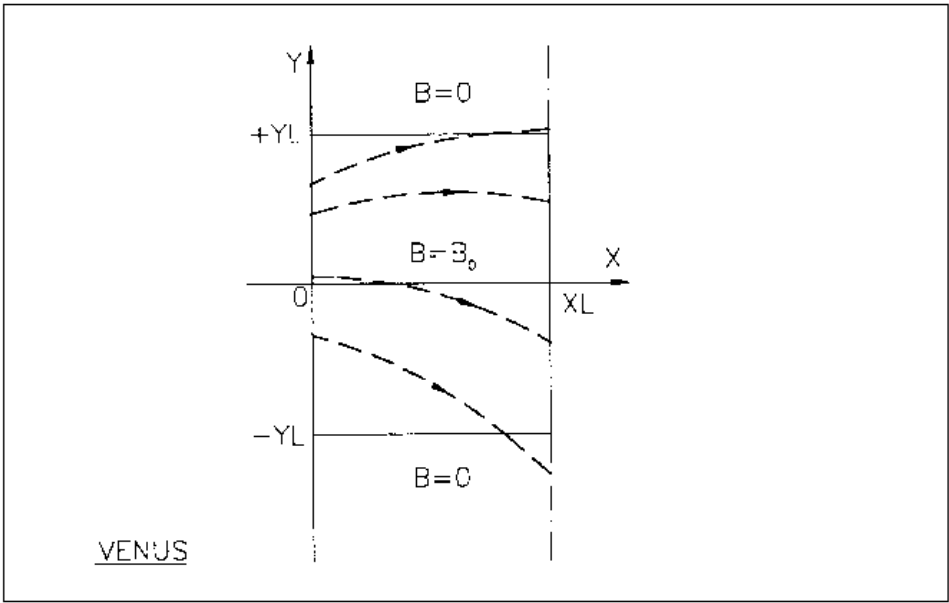
UNIPOT

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			
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, YL, B <sub>0</sub>	Length ; width = $\pm YL$ ; field	2*cm, kG	3*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	

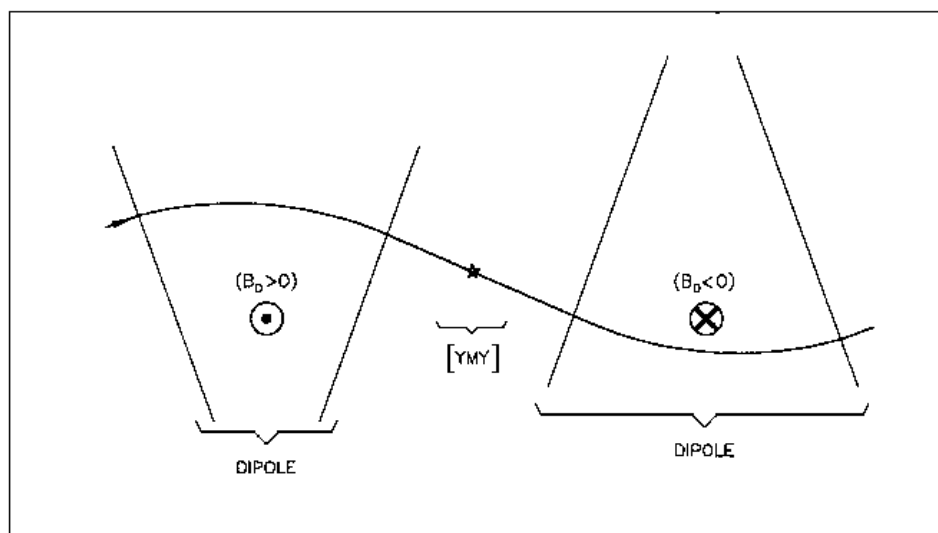


Scheme of VENUS rectangular dipole.

**WIENFILT**<sup>1</sup>**Wien filter**

$IL$	$IL = 1, 2[\times 10^n]$ , 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
$XL, E, B, HV$	Length ; electric field ; magnetic field ; option : element inactive ( $HV = 0$ ) horizontal ( $HV = 1$ ) or vertical ( $HV = 2$ ) separation	m, V/m, T, 0-2	3*E, I
$X_E, \lambda_{E_E}, \lambda_{B_E}$	<b>Entrance face :</b> Integration zone extent ; fringe field extents, E and B respectively ( $\simeq$ gap height)	3*cm	3*E
$C_{E0}-C_{E5}$ $C_{B0}-C_{B5}$	Fringe field coefficients for $E$ Fringe field coefficients for $B$	6*no dim. 6*no dim.	6*E 6*E
$X_S, \lambda_{E_S}, \lambda_{B_S}$ $C_{E0}-C_{E5}$ $C_{B0}-C_{B5}$	<b>Exit face :</b> See entrance face	3*cm 6*no dim. 6*no dim.	3*E 6*E 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

<sup>1</sup> Use *PARTICUL* to declare mass and charge.

**YMY****Reverse signs of  $Y$  and  $Z$  axes**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 [64], 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. 48.

**Example 2:** calculates the *first and second order transfer matrices* of an 800 MeV/c kaon beam line [65] 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. 49.

**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 [36]. 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. 50.

**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 R12=16.6 and R34=-.88 at the end of the beam line. Other example can be found in [66].

**Example 5:** shows the use of the *spin and multi-turn tracking procedures*, applied to the case of the SATURNE 3 GeV synchrotron [7, 10, 67]. 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. 52, 53 (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 [6]. The extremely small beam spot sizes involved (less than 1 micrometer) reveal the high accuracy of the ray-tracing (Figs. 54).

Many more examples can be found on the **zgoubi** development web site [5].



## 1 MONTE CARLO IMAGES IN SPES 2

Former Saclay/SATURNE and CERN QDD mass spectrometer [64].

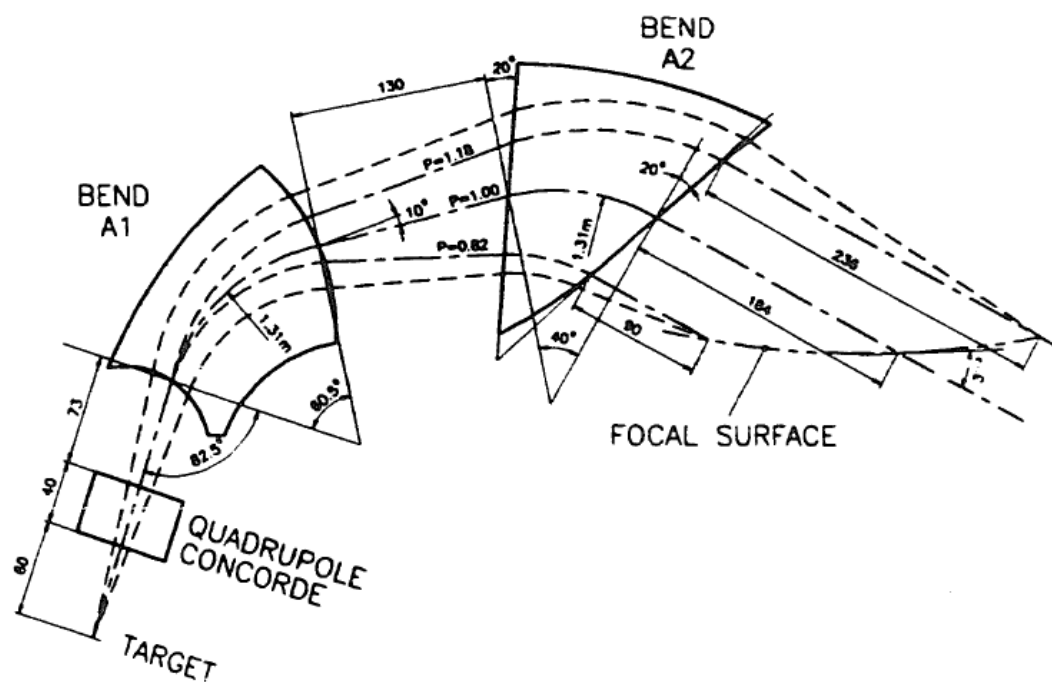


Figure 48: SPES 2 Layout.

**Excerpt from zgoubi.res : histograms of initial beam coordinates.**

```

2 HISTO

HISTOGRAMME DE LA COORDONNEE D
PARTICULES PRIMAIRES ET SECONDAIRES
DANS LA FENETRE : 0.9970 / 1.003
NORMALISE

20
19
18
17      D           D           D           D           D
16      D           D           D           D           D
15      D           D           D           D           D
14      D           D           D           D           D
13      D           D           D           D           D
12      D           D           D           D           D
11      D           D           D           D           D
10      O           O           O           O           O
9       D           D           D           D           D
8       D           D           D           D           D
7       D           D           D           D           D
6       D           D           D           D           D
5       D           D           D           D           D
4       D           D           D           D           D
3       D           D           D           D           D
2       D           D           D           D           D
1       D           D           D           D           D

    1234567890123456789012345678901234567890123456789012345678901
        2             3             4             5             6             7             8             9


TOTAL COMPTAGE          :   10000 SUR 10000
NUMERO DU CANAL MOYEN   :     51
COMPTAGE AU " "         :   2038
VAL. PHYS. AU " "       :   1.000
RESOLUTION PAR CANAL    :   7.500E-0
PARAMETRES PHYSIQUES DE LA DISTRIBUTION -
COMPTAGE = 10000 PARTICULES
MIN = 0.9980 , MAX = 1.002 , MAX-MIN = 4.0000E-03
MOYENNE = 1.000
SIGMA = 1.4108E-03

TRAJ 1 IEX,D,Y,T,Z,P,S,time : 1 0.9980 0.000 -30.24 0.000 44.63 0.0000 0.0000
*****
3 HISTO

HISTOGRAMME DE LA COORDDONEE THETA
PARTICULES PRIMAIRES ET SECONDAIRES
DANS LA FENETRE : -60.00 / 60.00 (MRD)
NORMALISE

20
19
18
17                                     T                               T
16                                  T T                       T T               T
15                                T T T                    T T T              TT TT
14                             TT T TT T TT T            TTTTTT TTTT TTT TTT
13                         TTTTTT TTTTTT TTTT TTT TTT T  TTTTTTTTTTTTTT TTTT TTTTTT
12                     TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
11                 TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
10                00000000000000000000000000000000000000000000000000000000000000
9                  TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
8                  TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
7                  TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
6                  TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
5                  TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
4                  TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
3                  TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
2                  TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
1                  TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT

    1234567890123456789012345678901234567890123456789012345678901
        2             3             4             5             6             7             8             9


TOTAL COMPTAGE          :   10000 SUR 10000
NUMERO DU CANAL MOYEN   :     51
COMPTAGE AU " "         :   128
VAL. PHYS. AU " "       :  3.331E-15 (MRD)
RESOLUTION PAR CANAL    :   1.50 (MRD)

PARAMETRES PHYSIQUES DE LA DISTRIBUTION :
COMPTAGE = 10000 PARTICULES
MIN = -49.99 , MAX = 50.00 , MAX-MIN = 99.98 (MRD)
MOYENNE = 0.3320 (MRD)
SIGMA = 29.04 (MRD)

TRAJ 1 IEX,D,Y,T,Z,P,S,time : 1 0.9980 0.000 -30.24 0.000 44.63 0.0000 0.0000
```



## 2 TRANSFER MATRICES ALONG A TWO-STAGE SEPARATION KAON BEAM LINE

800 MeV/c kaon beam line at BNL Alternating Gradient Synchrotron [65]. The line includes two separation Wien filters.

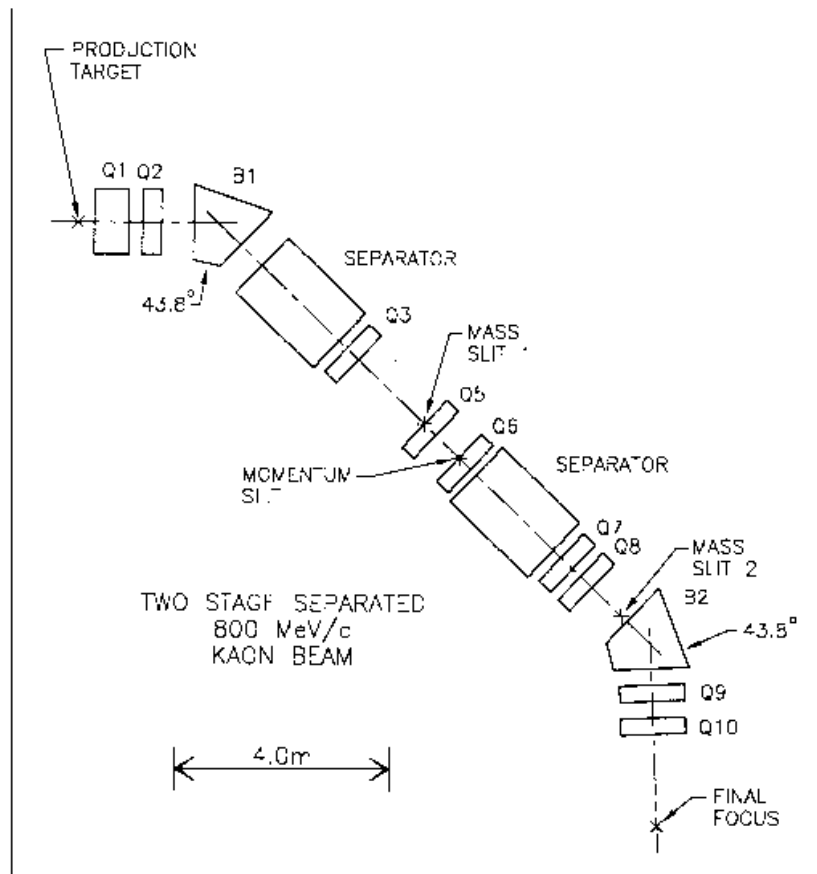
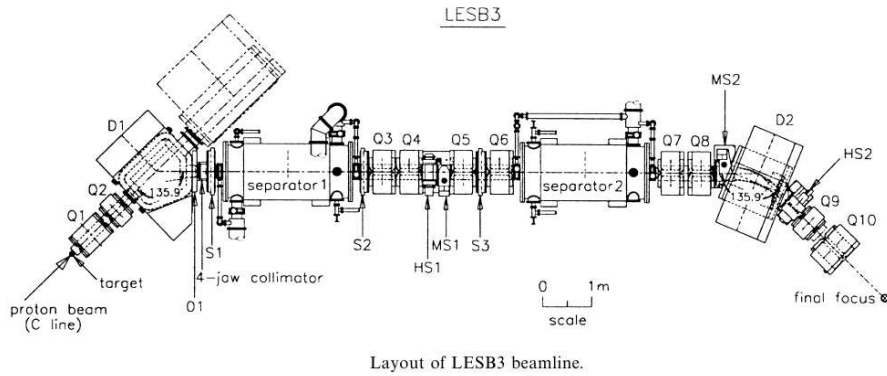


Figure 49: 800 MeV/c kaon beam line layout.





```

'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
2 0 AT THE FINAL FOCUS                     45
'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

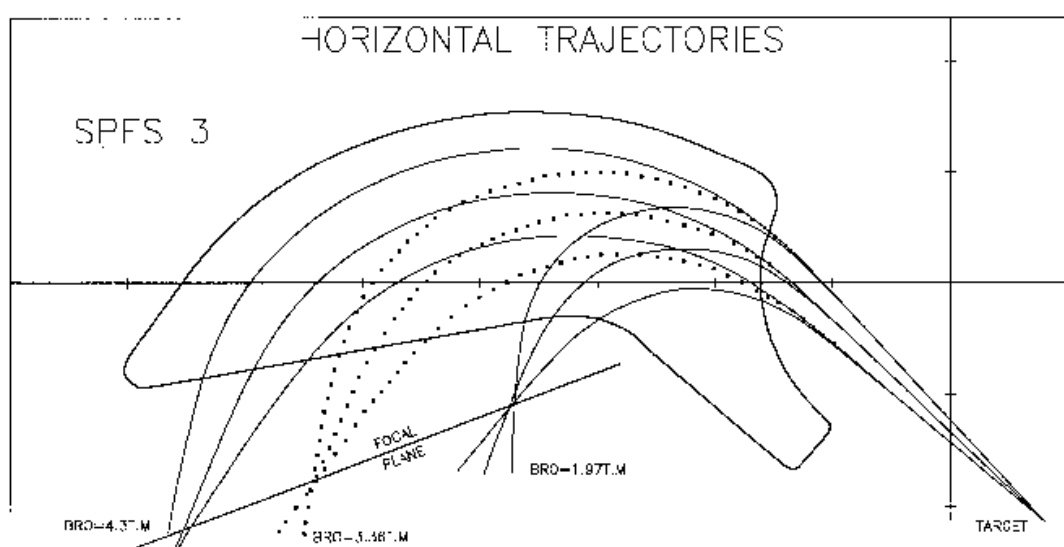


Figure 50: Layout of SPES 3 spectrometer.

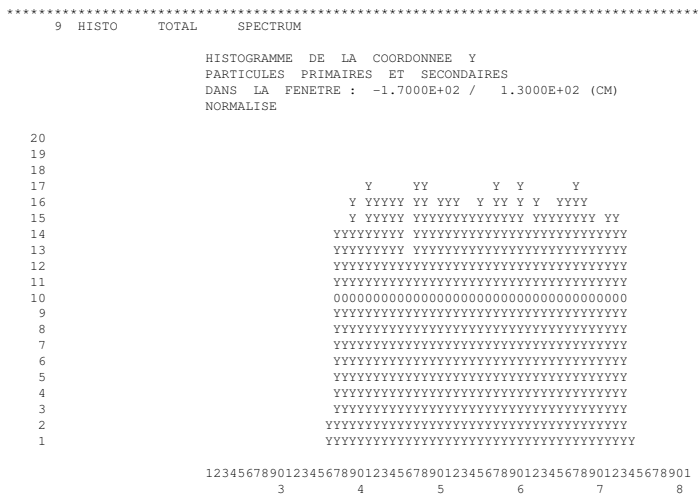
**zgoubi data file**

```

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
2 -170. 130. 60 2 AT FOCAL PLANE.
20 'P' 1 'P'
'HISTO' 11
MUON SPATIAL SPECTRUM
2 -170. 130. 60 3 AT FOCAL PLANE.
20 'y' 1 'S'
'HISTO' 12
MUON MOMENTUM SPECTRUM
1 .2 1.7 60 3 AT FOCAL PLANE.
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.**

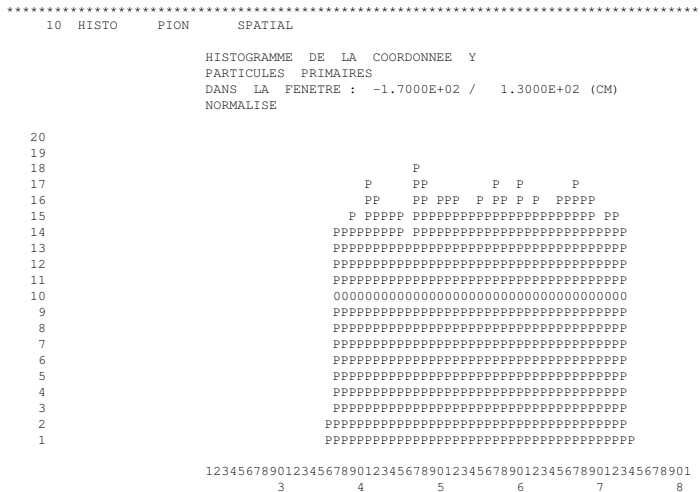


```

TOTAL COMPTAGE          : 9887 SUR 10000
NUMERO DU CANAL MOYEN   : 55
COMPTAGE AU " "         : 281
VAL. PHYS. AU " "      : 0.00E+00 (CM)
RESOLUTION PAR CANAL    : 5.000E+00 (CM)

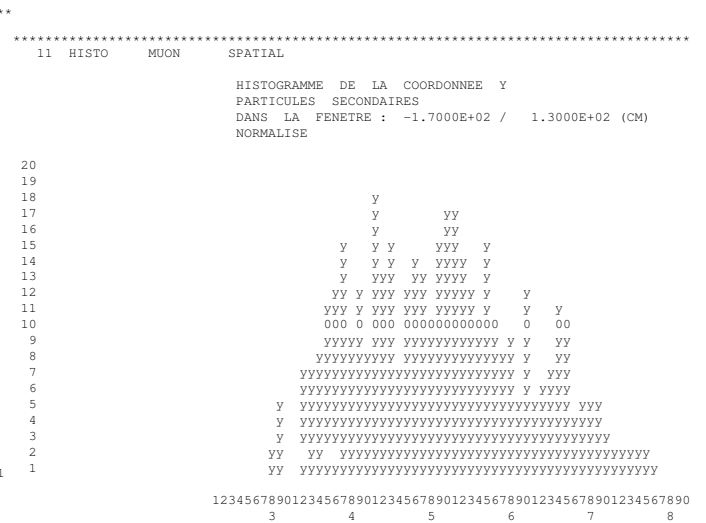
```

```
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)
```



TOTAL COMPTAGE	:	9282	SUR	10000
NUMERO DU CANAL MOYEN	:	55		
COMPTAGE AU " "	:	264		
VAL. PHYS. AU " "	:	0.00E+00	(CM)	
RESOLUTION PAR CANAL	:	5.00E+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)
```

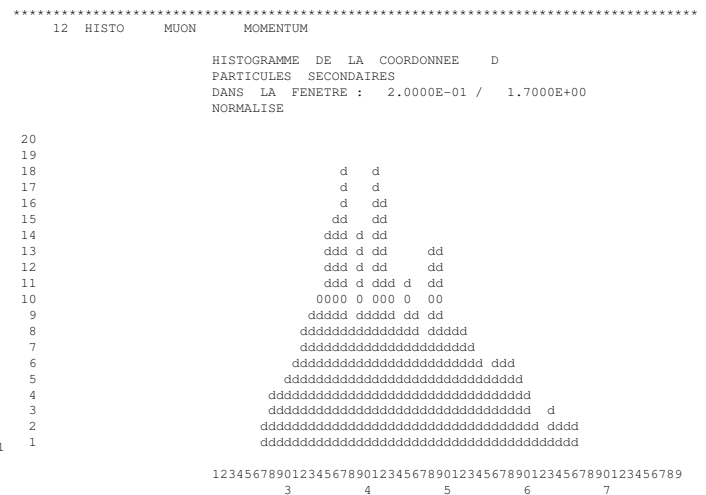


```

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)
```



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

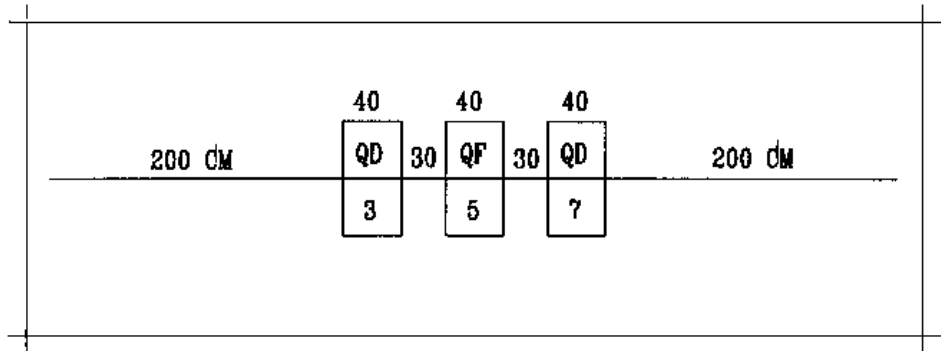


Figure 51: Vary B in all quadrupoles, for fitting of the transfer coefficients  $R_{12}$  and  $R_{34}$  at the end of the line. The first and last quadrupoles are coupled so as to present the same value of B.

## zgoubi data file.

```

MATCHING A SYMMETRIC QUADRUPOLE TRIPLET
'OBJET'                                     1
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' '                                     2
200.
'QUADRUPO'      3                           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'                                     4
30.
'QUADRUPO'      5                           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'                                     6
30.
'QUADRUPO'      7                           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'                                     8
200.
'MATRIX'                                             9
1 0
'FIT2'          VARY B IN QUADS FOR FIT OF R12 AND R34      10
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'                                             11

```

**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

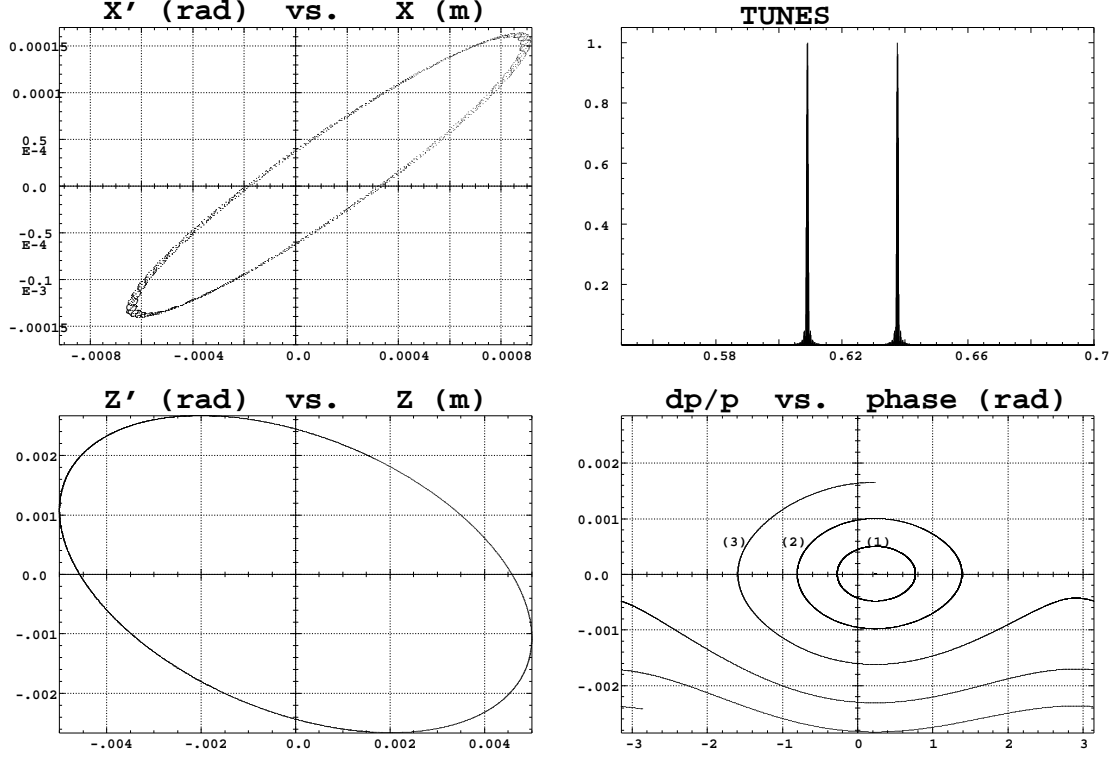


Figure 52: Tracking over 3000 turns. These simulations exhibit the first order parameters and motions as produced by the multi-turn ray-tracing.

(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  $1.65 \cdot 10^{-3}$ .

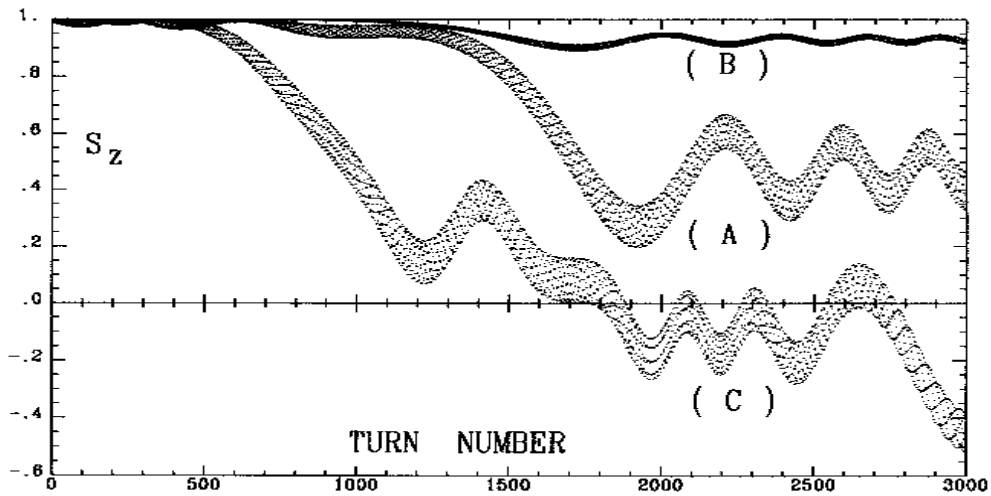


Figure 53: Crossing of  $\gamma G = 7 - \nu_Z$ , at  $\dot{B} = 2.1$  T/s.

(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 [68] 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  $10^{-3}$ .





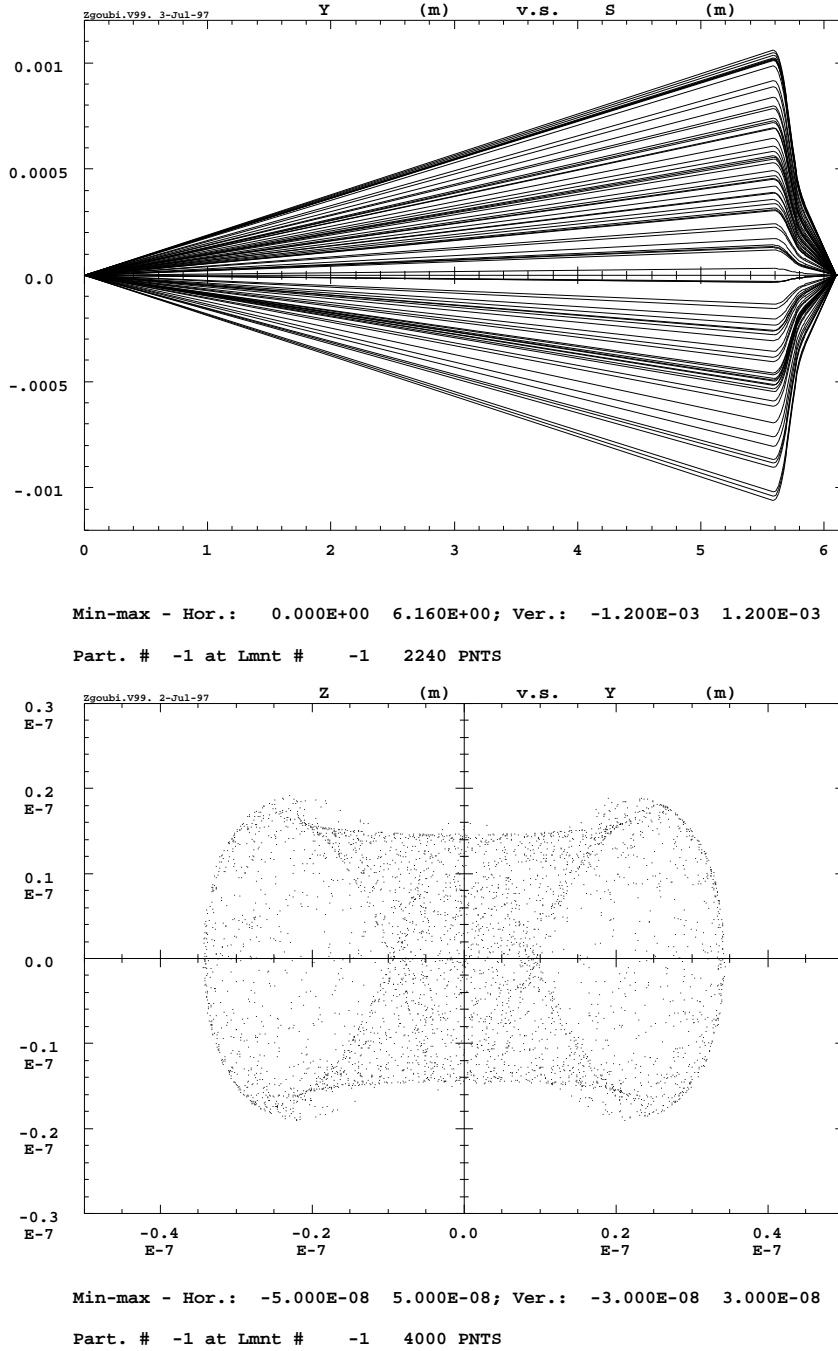
6 MICRO-BEAM FOCUSING WITH  $\vec{E} \times \vec{B}$  QUADRUPOLES

Figure 54: *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. 0. 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. ELECTRIC Q-POLE COMPONENT.
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. 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.
10.2 10. 0. 1.89493 0. 0. 0. 0. 0. 0. 0. MAGNETIC Q-POLE COMPONENT.
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. 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.
.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.
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.
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.
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.
.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 (MKSA units):
Yo, To, Zo, Po, Xo, BR/BORO : 0.000 0.000 0.000 0.000 0.000 1.000
Width ( +/- , MKSA 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 ) :
NV, 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 = 5.00000000 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 = 5.59000000 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 = 5.69200000 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 = 5.74100000 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 = 5.84300000 m ; Time (for ref. rigidity & particle) = 2.985092E-06 s

*****

```

9 HISTO HISTOGRAMME

HISTOGRAMME DE LA COORDONNEE Y  
PARTICULES PRIMAIRES ET SECONDAIRES  
DANS LA FENETRE : -5.0000E-06 / 5.0000E-06 (CM)  
NORMALISE

```

20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1

```

123456789012345678901234567890123456789012345678901

3 4 5 6 7 8

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,IE : 1.0002E+00 9.0257E-07 -2.3996E-01 -1.0770E-06 1.7947E+00 6.09300E+02 1

\*\*\*\*\*  
10 HISTO HISTOGRAMME

HISTOGRAMME DE LA COORDONNEE Z  
PARTICULES PRIMAIRES ET SECONDAIRES  
DANS LA FENETRE : -5.0000E-06 / 5.0000E-06 (CM)  
NORMALISE

```

20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1

```

123456789012345678901234567890123456789012345678901

3 4 5 6 7 8

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,IE : 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' \*\*\*\*  
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 [5]. 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).

Link **zpop** with the graphic library, libminigraf.a [69]. This will create the executable **zpop**, that is to be run on an xterm type 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, default input is **zgoubi.dat**.
- Results of the execution will be printed into **zgoubi.res** and, upon options appearing in **zgoubi.dat**, into several other outputs files (see section 2 below).
- The command **zgoubi** accepts the following arguments (see **zgoubi\_main.f** source file) :
  - in or -fileIn, followed by an 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.
- 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, field map contours, tune diagram, etc.).

## 2 STORAGE FILES

When explicitly requested by means of the adequate keywords, options, or dedicated *LABEL*'s, extra storage files are opened by **zgoubi** (*FORTRAN* "OPEN" statement) and filled.

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' [73] is straightforward. A library of 'gnuplot' command files can be found in **zgoubi** sourceForge repository [5].

The **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.

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  
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  
Isomagnetic field lines of 2-D map. Superimposing trajectories read from zgoubi.plt is possible.
- zgoubi.plt  
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 and angular distributions of synchrotron radiation.
- zgoubi.spn  
Spin coordinates and histograms, at the position where *SPNPRNL* appears in the structure. Resonance crossing when performing multi-turn tracking.

In addition, many routines will deliver execution outcomes in individual output files if requested, usually by introducing a '*PRINT*' command under the keyword of concern. Instances are zgoubi.MATRIX.out (keyword *MATRIX*), zgoubi.TWISS.out (keyword *OPTICS*), zgoubi.PICKUP.out (keyword *PICKUPS*), zgoubi.TWISS.out (keyword *TWISS*), etc.



## References

- [1] F. Méot et S. Valéro, *Manuel d'utilisation de Zgoubi*, Rapport IRF/LNS/88-13, CEA Saclay, 1988.
- [2] 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).
- [3] F. Méot and S. Valéro, *Zgoubi users' guide - Version 3*, Int. Rep. DSM/LNS/GT/93-12, CEA Saclay (1993).
- [4] F. Méot and S. Valéro, *Zgoubi users' guide - Version 4*, FNAL Tech. Rep. FERMILAB-TM-2010 (Aug. 1997), & Int. Rep. CEA DSM DAPNIA/SEA-97-13, Saclay (Oct. 1997).
- [5] <http://zgoubi.sourceforge.net/>
- [6] 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.
- [7] D. Carvounas, *Suivi numérique de particules chargées dans un solénoïde*, rapport de stage, CEA/LNS/GT, 1991.
- [8] F. Méot, *Raytracing in 3-D field maps with Zgoubi*, report DSM/LNS/GT/90-01, CEA Saclay, 1990.
- [9] 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.
- [10] F. Méot, *A numerical method for combined spin tracking and raytracing of charged particles*, NIM **A313** (1992) 492, and proc. EPAC (1992) p.747.
- [11] D. J. Kelliher et al., *Muon decay ring study*, Procs. EPAC08 Conf., Genoa, Italy (2008).
- [12] F. Méot, N. Monseu, *Lattice Design and Study Tools Regarding the Super-B Project*, Procs. IPAC10 Conf., Kyoto, Japan (2010).
- [13] 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).
- [14] F. Méot, M. Bai, V. Ptitsyn, V. Ranjbar, *Spin Code Benchmarking at RHIC*, Procs. PAC11 Conf., New York (2011).
- [15] 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).
- [16] F. éot, *An ergodic approach to polarization in eRHIC electron storage ring*, eRHIC Tech Note 59, BNL, 2018; <https://public.bnl.gov/docs/cad/Documents/An%20ergodic%20approach%20to%20polarization%20in%20eRHIC%20electron%20storage%20ring.pdf>
- [17] 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.
- [18] S.Y. Lee, *Spin dynamics and snakes in synchrotrons*, World Scientific, 1997.
- [19] S.Y. Lee, private communication, April 2017.
- [20] 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).
- [21] G. Leleux et al., *SR perturbations in long transport lines*, IEEE 1991 Part Acc. Conf., San Francisco (May 1991).
- [22] 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.
- [23] *Electron Lab for Europe*, Blue Book, CNRS-IN2P3 (1994).
- [24] F. Méot, *Benchmarking stepwise ray-tracing in rings in presence of radiation damping*, Procs. PAC11 Conf., New York (2011).
- [25] F. Méot, *Simulation of radiation damping in rings, using stepwise ray-tracing methods*, 2015 JINST 10 T06006.
- [26] V. O. Kostroun, *Simple numerical evaluation of modified Bessel functions and integrals [...]*, NIM 172 (1980) 371-374.
- [27] F. Méot, *Synchrotron radiation interferences at the LEP miniwiggler*, Report CERN SL/94-22 (AP), 1994.
- [28] L. Ponce, R. Jung, F. Méot, *LHC proton beam diagnostics using synchrotron radiation*, Yellow Report CERN-2004-007.
- [29] F. Méot, *A theory of low frequency far-field synchrotron radiation*, Particle Accelerators Vol 62, pp. 215-239 (1999).
- [30] Albert Hofmann, *The Physics of Synchrotron Radiation*, Cambridge University Press, May 13, 2004.
- [31] Malek Haj Tahar, *Fixed Field Ring Accelerators and ADS-Reactor Application*, PhD dissertation, Brookhaven National Laboratory and Grenoble-Alpes University (2016).
- [32] B. Mayer, personal communication, CEA Saclay, Laboratoire National SATURNE, 1990.

- [33] 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.
- [34] 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.
- [35] 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.
- [36] F. Méot and N. Willis, *Raytrace computation with Monte Carlo simulation of particle decay*, internal report CEA/LNS/88-18 CEA Saclay, 1988.
- [37] E.J. Bleser, *The parameters of the bare AGS*, Tech. Note AGS/AD?Tech. Note 430, March 15, 1996.
- [38] These transfer functions have been copied from the MADX model of the AGS.
- [39] H.A. Enge, *Deflecting magnets*, in **Focusing of Charged Particles**, ed. A. Septier, **Vol. II**, pp 203-264, Academic Press Inc., 1967.
- [40] 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.
- [41] Files developed by Simon White, January 2012. Including beam-beam spin kick after Ref. [42].
- [42] Y. K. Batygin, *Spin depolarization due to beam-beam collisions*, Phys. Rev. E, Vol. 58, 1, July 1998.
- [43] A. Hofmann, S. Myers, *Beam Dynamics in a Double RF System*, CERN ISR-TH-RF/80-26 (July 1980).
- [44] 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.
- [45] F. Méot, *6-D beam dynamics simulations in FFAGs using the ray-tracing code Zgoubi*, ICFA Beam Dyn.Newslett.43:44-50 (2007).
- [46] Franck Lemuët, *Collection and muon acceleration in the neutrino factory project*, PhD thesis, Paris KI University, April 2007.
- [47] 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.
- [48] A. Septier, *Cours du DEA de physique des particules, optique corpusculaire*, Université d'Orsay, 1966-67, pp. 38-39.
- [49] S. P. Karetskaya *et al.*, *Mirror-bank energy analyzers*, in *Advances in electronics and electron physics*, Vol. 89, Acad. Press (1994) 391-491.
- [50] 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.
- [51] 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).
- [52] F. Méot, *Tracking studies regarding EMMA FFAG project*, Internal report CEA DAPNIA-06-04 (2006).
- [53] J. S. Berg *et al.*, *Recent developments on the EMMA on-line commissioning software*, Procs. IPAC10 Conf., Kyoto, Japan (2010).
- [54] S. Machida *et al.*, *Acceleration in the linear non-scaling fixed-field alternating-gradient accelerator EMMA*, Nature Physics, vol. 8, March 2012.
- [55] Installed by Pavel Akishin, JINR, Dubna, 1992.
- [56] M.W. Garrett, *Calculation of fields [...] by elliptic integrals*, J. Appl. Phys., **34**, 9, sept. 1963.
- [57] P.F. Byrd and M.D. Friedman, *Handbook of elliptic integrals for engineers and scientists*, pp. 282-283, Springer-Verlag, Berlin, 1954.
- [58] Installation by M. Bai, BNL, 2009.
- [59] A. Tkatchenko, *Computer program UNIPOT, SATURNE*, CEA Saclay, 1982.
- [60] S.B. Kowalski, H.A. Enge, *The ion-optical program raytrace*, NIM A258 Vol. 3 (1987) 407.
- [61] N. Tsoupas, private communication, Brookhaven National Laboratory, Oct 2015.
- [62] J.L. Chuma, *PLOTDATA*, TRIUMF Design Note TRI-CO-87-03a.

- [63] F. Méot et al., A model of the AGS in the ray-tracing code Zgoubi, Tech. Note C-A/AP/\*\*\*\*, 2013.
- [64] 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
- [65] 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).
- [66] 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.
- [67] 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).
- [68] M. Froissart et R. Stora, *Dépolarisation d'un faisceau de protons polarisés dans un synchrotron*, NIM 7 (1960) 297-305.
- [69] J.L. Hamel, *mini graphic library LIBMINIGRAF*, CEA-DSM, Saclay, 1996.
- [70] F. Méot and A. París, *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.
- [71] A white paper: The Cornell-BNL FFAG-ERL Test Accelerator, I. Ben-Zvi et al., Dec. 16, 2014.
- [72] F. Méot, *End-to-end 9-D polarized bunch transport in FFAG eRHIC*, EIC14 workshop, <http://www.jlab.org/conferences/eic2014/> (2014).
- [73] <http://www.gnuplot.info/>

## Index

- .plt, 14, 76, 115, 116, 145, 184, **184**, 185
- ALE, 186
- acceleration, 79, 110, 212, 275, 278
- AGSMM, **95**, 186, **197**
- AGSQUAD, **96**, **198**
- AIMANT, **97**, 122, **199**
  - parameter numbering, 65
- alignment (mis-) of optical elements, **186**
- AUTOREF, **102**, **203**
- backward ray-tracing, **184**
- BEAMBEAM, **103**, **204**
  - spin kick, 103, 204
- BEND, 27, **104**, **205**
- BETATRON, **106**, **207**
- BINARY, **61**, **208**
- BORO, **52**, **55**, **59**, 110, 215, **254**, **258**, **261**
- $BORO \times D_{ref}$ , 85, 95, **110**, **112**, **184**, 239
- $B\rho_{ref}$ , 85, 95, **110**, 112, 184, 239
- BREVOL, 26, **107**, **209**
- CARTEMES, 26, 107, **108**, 135, 143, 144, 148, 149, 158, 185, **210**, 231, 293, 330
- cartesian coordinates, **186**
- CAVITE, 82, **110**, 111, 186, **212**
  - double RF system, 111, 212
  - particle path length reset, 79, 185
  - recovering from a crash, 113
- CHAMBR, 79, 81, 114, **114**, **213**, 277
- CHANGREF, 74, 102, 114, 115, **115**, 117, **214**
- checking field, **184**
- checking trajectories, **184**
- chromaticity, **175**, 176, 298
- CIBLE, 73, **117**, **215**
- circular accelerators, 79, 104, 185
- closed orbit, 14, 69, 110, 182
  - computation, 113
- COLLIMA, 79, 81, **118**, **216**, 277
- CONSTY: constant radius - see OPTIONS, 264
- coupling, **176**, **182**
- cyclotron, 110, 112, 124
- DECAPOLE, 77, **119**, **217**, 274
- DIPOLE, 64, 97, 114, **120**, 124, 186, **218**, 307
  - parameter numbering, 65
- DIPOLE-M, 97, **122**, 149, 188, **219**, 269
  - parameter numbering, 65
- DIPOLES, 97, **124**, **221**
- DODECAPO, 27, 77, **127**, **222**, 274
- double RF system, 111
- DRIFT, **128**, **223**
  - split, 128, 185, 223
- droite de coupure: see integration boundary, 108
- EBMULT, 27, 28, 77, **129**, 224
  - parameter numbering, 65
- EL2TUB, 26, **130**, **226**
- ELMIR, **131**, **227**
- ELMIRC, **132**, **228**
- ELMULT, 28, 77, 129, **133**, **229**
  - parameter numbering, 65
- ELREVOL, 26, **135**, 185, **230**
- EMMA, **136**, **231**
- END, 60, **62**, 177, **236**
- ERRORS, **63**, **232**
- ESL, 74, **128**, **223**
- FAISCEAU, **169**, **233**
- FAISCNL, **169**, 185, **233**, 330
- FAISTORE, **169**, 185, **233**, 330
  - binary storage, 170
  - in presence of *FIT[2]*, 70, 170
- FFAG, 110, 112, 124, 136, 137, **137**, 139, **234**
- FFAG magnet
  - radial, 124, **137**, **234**
  - spiral, **139**, **235**
- FFAG-SPI, **139**, **235**
- FIN, **62**, **236**
- FIT[2], 60, 64, **64**, 67, **237**, 275
  - combined with REBELOTE, 70, 80
  - constraint, **67**, 68, **238**
  - coupled variables, **67**
  - in presence of FAISTORE, 70
  - max. number of iterations, **70**, 237
  - no final run, 70
  - numbering, keywords, 64
  - numbering, parameters, 64
  - options, **70**, 72
  - penalty, **70**, 237
  - variable values
    - get, **72**, **243**
    - range, **67**, **237**
    - save, **70**
- FOCALE, **171**, **241**
- FOCALEZ, **172**, **241**
- fringe fields overlapping, 124
- GASCAT, **71**, **242**
- GETFITVAL, 60, **72**, **243**
- gnuplot, 330
- GOTO, **141**, **244**
- HISTO, 53, 74, 81, 118, 173, **173**, 188, **245**, 277
- IC*, 185
- ID*, **108**, **210**, **235**
- IDMAX*, 55, **55**
- IEX*, 31, 56, **56**, 92, 108, 114, 118, 169, 179, **188**
- IL*, 65, 66, 76, 114, 184, **184**, 185
- IMAGE, 102, 171, **171**, **246**
- IMAGES, 55, 171, **171**, **246**
- IMAGESZ, **172**, **246**
- IMAGEZ, **172**, **246**
- IMAX, **52**, **55**, 59, 74, 79, 87, 110, 111, 169, 180, 185, 188
- INCLUDE, **142**, **247**
- INTEG, 81, 277
- integration boundary, 107, 108, 135, 143, 144, 158, 209–211, 293

- integration step size, 188
  - coded, **188**
  - negative, **74**, 184
- IODRE, 27, 29, 77, 101, 108, 123, 143, 144, 158
- KPOS, **186**
- LABEL, 267
- LABEL*, 82, 145, 170, 177, 179, **185**, 233, 251, 267, 285, 330
- MAP2D, 27, **143**, 158, **249**
- MAP2D-E, 27, **144**, **250**
- maps, summing, 209
- MARKER, **145**, **251**, 330
- MATRIX, 57, 58, 64, 67, 102, 161, **175**, 176, 203, **252**, 259
  - 1st, 2nd, 3rd order transport coefficients, 175, 252
  - beam matrix, 75, 175
  - periodic beam matrix, 182
  - spin rotation matrix, 180
- MCDESINT, 53, 55, 60, **73**, 74, 79, 81, 173, **253**, 277, 307
- MCOBJET, **52**, 53, 69, 73, 79, 188, **254**, 275
- misalignment, **186**
- momentum compaction, 298
- Monte Carlo, 52, 59, 73, 79, 188, 254, 261
- multi-particle, 79, **188**, 275
- multi-turn tracking, 52, 60, 79, **79**, 82, 110, 184, **185**, **275**, 307, 320, 330
- MULTIPOL, 27, 77, 82, 129, 133, **146**, **257**, 274
  - parameter numbering, 65
- negative charge, 52, 55, 188, 254, 258
- negative momentum, 52, 55, 188, 254, 258
- negative rigidity, **188**
- NPASS, 79, 110, 170, 173, 179, 185, 188, 233, 275, 285
- OBJET, **55**, 67, 73, 102, 110, 114, 171, 182, 186, 188, **258**
  - for 1st, 2nd, 3rd order *MATRIX* computation, 57, 58, 258
  - recovering from a crash, 57, 260
- OBJETA, 59, **59**, 79, **261**
- OCTUPOLE, 77, **147**, **262**, 274
- OPTICS, **75**, **263**
- OPTIONS, **76**, **264**
  - CONSTY, 76, 264
  - WRITE, 264
- Orbit correction, **47**
  - *FIT*[2], 47
  - *SVDOC*, 47
  - *SVDOC*, 90
- ORDRE, **77**, **265**
- outpoi.lis, 148
- particle path length reset by CAVITE, 79, 185
- PARTICUL, 59, 60, 73, **78**, 86, 87, 89, 92, 118, 129–133, 135, 154, 164, 166, 253, **266**
- PICKUPS, **177**, 185, **267**
- PLOTDATA, **178**, **268**
- POISSON, **148**, **269**
- polar coordinates, **188**
- POLARMES, **149**, 188, **270**
- positioning of optical elements, **186**
- PRINT
  - CAVITE, 113
  - *MATRIX*, 176
  - *OPTICS*, 75
  - *SPNPRT*, 180
- PS170, **150**, **271**
- QUADISEX, 26, 27, **151**, **272**
- QUADRUPO, 27, 64, 65, 77, 114, 119, 133, 146, 147, **152**, 155, 186, **273**, 274
  - parameter numbering, 65
- REBELOTE, 52, 57, 60, 64, 79, **79**, 82, 110, 169, 170, 173, 177, 179, 185, 186, 188, 233, 258, **275**, 285, 307
  - circular accelerators, 79, 185
  - combined with *FIT*[2], 70, 80
  - recovering from a crash, 80
- recovering from a crash, 57, 80, 85, 113, 260
- reference rigidity, 52, 55, 84, 110, **184**
- RESET, **81**, **277**
- SCALING, 81, 82, **82**, 110, 111, 185, 186, 212, 277, **278**
  - parameter numbering, 66
  - recovering from a crash, 85
- SEPARA, **154**, **279**
- SEXQUAD, 26, **151**, **280**
- SEXTUPOL, 27, 77, **155**, 274, **281**
- SOLENOID, **156**, **282**
- SPACECHARG, 45, **86**, **283**
- SPIN
  - $\vec{n}_0$  vector, 69
  - beam-beam kick, 103, 204
  - rotation matrix, 180
  - tracking, **35**, 79, **87**, 115, 162, 168, 173, 179, 188, 245, 275, **286**, 307
- SPINR, **157**, **284**
- SPNPRNL, **179**, **285**, 330
- SPNPRT, 89, **180**, **285**
- SPNSTORE, **179**, **285**, 330
- SPNTRK, 60, 79, 81, **87**, 188, 277, **286**
- SRLOSS, **89**, **287**, 288
- SRPRNT, **181**, **288**
- stopped particles, 79, 114, 118, 169, **188**, 213, 216
- storage files, 329
- SVDOC, **90**, **289**
  - SVD matrix, 90
  - iterate orbit defect & correction, 90
  - multiple reference orbits, 90
  - zgoubi.SVD.out, 90
- synchrotron motion, 79, 82, 110, 212, 275, 278
- synchrotron radiation, 39, 330
- synchrotron radiation energy loss, 89, 111, 181
  - treatment in CAVITE, 112, 113
- synchrotron radiation loss, 287
- synchrotron radiation spectra, 92, 290
- SYNRAD, **92**, **290**
- SYSTEM, **93**, **291**
- system call, 93, 291
- tag, symmetry index, 55
- tag, tag character, 53, 55, 56, 74, 92, 169, 173, 179
- TARGET, **117**, **215**
- time varying fields, **184**
- TOSCA, 26, 65, 77, **158**, 185, 186, **293**, 307, 330



- parameter numbering, 65
- TRANSMAT, **161, 295**
- transport coefficients, **175, 252**
  - 1st, 2nd or 3rd order, 175, 252
- TRAROT, 105, **162, 296**
- TWISS, 57, **182, 297**
  - find orbit, 182, 297
- UNDULATOR, **163, 299**
- UNIPOT, 26, **164, 300**
- Units in **zgoubi**, **183**
- variable (FIT, FIT2), 64, **64, 237**
- VENUS, 26, 27, 165, **301**
- WIENFILT, 27, **166, 302**
- WRITE: allow/inhibit - see OPTIONS, 264
- XCE, 186
- XPAS
  - negative, 74
- XPAS, coded, 188
- XPAS, negative, 184
- YCE, 186
- YMY, 105, **167, 303**
- zgoubi, 329
  - running the code, 329
- zgoubi.CAVITE.out, **113**
- zgoubi.dat, 170, 179, 329
- zgoubi.f, 329
- zgoubi.fai, 169, 179, 233, 258, 330
- zgoubi.impdev.out, 185, **185**
- zgoubi.map, 185, 330
- zgoubi.MATRIX.out, **176**
- zgoubi.OPTICS.out, **75**
- zgoubi.PICKUP.out, 177
- zgoubi.plt, 145, 184, 185, 251, 330
- zgoubi.res, 184, 185, 329
- zgoubi.spn, 179, 285, 330
- zgoubi.SPNPRT.Out, 239, 248, 276
- zgoubi.sre, 92
- zgoubi.TWISS.out, 182
- zpop, 41, 169, 179, 184, 329
  - running the code, 329