ZGOUBI USERS’ GUIDE

ZGOUBI ON WEB:
http://sourceforge.net/projects/zgoubi/

François Méot
Brookhaven National Laboratory
Collider-Accelerator Department
Upton, NY, 11973

October 25, 2012

ATLAS & CMS IRs, LHC

Serpentine acceleration, EMMA

Snake Resonance Crossing, RHIC

Spin flip during ramp, AGS with snakes
Cover figures:

upper left: collision optics at ATLAS and CMS [59],
upper right: polarization upon crossing of 393+Q_y resonance in RHIC [14],
lower left: spin-flip cycle in the AGS with helical snakes [13],
lower right: serpentine acceleration in the prototype linear FFAG EMMA [46, 47].
# Table of contents

PART A Description of software contents 5

GLOSSARY OF KEYWORDS 7

OPTICAL ELEMENTS VERSUS KEYWORDS 9

PREFACE TO THE BNL EDITION (2012) 11

INTRODUCTION TO THE 4th EDITION (1997) 13

INTRODUCTION 15

1 NUMERICAL CALCULATION OF MOTION AND FIELDS 17

1.1 zgoubi Frame .......................................................... 17

1.2 Integration of the Lorentz Equation ................................ 17

1.2.1 Integration in Magnetic Fields .................................. 19

1.2.2 Integration in Electric Fields .................................. 19

1.2.3 Integration in Combined Electric and Magnetic Fields .... 22

1.2.4 Calculation of the Time of Flight .............................. 23

1.3 Calculation of \( \vec{E} \) and \( \vec{B} \) Fields and their Derivatives .... 23

1.3.1 1-D (Axial) Analytical Field Models and Extrapolation .... 24

1.3.2 Extrapolation from 1-D axial field map ....................... 24

1.3.3 Extrapolation From Median Plane Field Models .............. 24

1.3.4 Extrapolation from Arbitrary 2-D Field Maps .............. 25

1.3.5 Interpolation in 3-D Field Maps .................................. 25

1.3.6 2-D Analytical Field Models and Extrapolation ............ 25

1.3.7 3-D Analytical Models of Fields .............................. 25

1.4 Calculation of \( \vec{E} \) and \( \vec{B} \) from Field Maps .............. 26

1.4.1 1-D Axial Map, with Cylindrical Symmetry ................... 26

1.4.2 2-D Median Plane Map, with Median Plane Antisymmetry ... 27

1.4.3 Arbitrary 2-D Map, no Symmetry .............................. 29

1.4.4 3-D Field Map .................................................. 30

2 SPIN TRACKING 33

2.1 Introduction .......................................................... 33

2.2 Integration in Magnetic Fields ...................................... 33

2.3 Integration in Electric Fields ...................................... 34

2.4 Integration in Combined Electric and Magnetic Fields ...... 35

3 SYNCHROTRON RADIATION 37

3.1 Energy Loss and Dynamical Effects ................................ 37

3.2 Spectral-Angular Radiated Densities .............................. 39

3.2.1 Calculation of the Radiated Electric Field .................. 39

3.2.2 Calculation of the Fourier Transform of the Electric Field . 41

4 DESCRIPTION OF THE AVAILABLE PROCEDURES 43

4.1 Introduction .......................................................... 43

4.2 Definition of an Object .............................................. 45

4.3 Declaring Options .................................................. 54

4.4 Optical Elements and Related Numerical Procedures ......... 79

4.5 Output Procedures .................................................. 147

4.6 Complements Regarding Various Functionalities ............. 161

4.6.1 Reference rigidity ................................................. 161

4.6.2 Time Varying Fields ............................................. 161

4.6.3 Backward Ray-Tracing .......................................... 161

4.6.4 Checking Fields and Trajectories Inside Optical Elements . 161

4.6.5 Labeling Keywords ............................................... 162

4.6.6 Multi-turn Tracking in Circular Machines .................. 162

4.6.7 Positioning, (Mis-)Alignment, of Optical Elements and Field Maps 163

4.6.8 Coded Integration Step ......................................... 166
PART A
Description of software contents
Glossary of Keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGSMM</td>
<td>AGS main magnet</td>
<td>79</td>
</tr>
<tr>
<td>AGSQUAD</td>
<td>AGS quadrupole</td>
<td>80</td>
</tr>
<tr>
<td>AIMANT</td>
<td>Generation of dipole mid-plane 2-D map, polar frame</td>
<td>81</td>
</tr>
<tr>
<td>AUTOREF</td>
<td>Automatic transformation to a new reference frame</td>
<td>86</td>
</tr>
<tr>
<td>BEAMBEAM</td>
<td>Beam-beam lens</td>
<td>87</td>
</tr>
<tr>
<td>BEND</td>
<td>Bending magnet, Cartesian frame</td>
<td>88</td>
</tr>
<tr>
<td>BINARY</td>
<td>BINARY/FORMATTED data converter</td>
<td>55</td>
</tr>
<tr>
<td>BREVOL</td>
<td>1-D uniform mesh magnetic field map</td>
<td>89</td>
</tr>
<tr>
<td>CARTEMES</td>
<td>2-D Cartesian uniform mesh magnetic field map</td>
<td>90</td>
</tr>
<tr>
<td>CAVITE</td>
<td>Accelerating cavity</td>
<td>92</td>
</tr>
<tr>
<td>CHAMBR</td>
<td>Long transverse aperture limitation</td>
<td>95</td>
</tr>
<tr>
<td>CHANGREF</td>
<td>Transformation to a new reference frame</td>
<td>96</td>
</tr>
<tr>
<td>CIBLE</td>
<td>Generate a secondary beam following target interaction</td>
<td>98</td>
</tr>
<tr>
<td>COLLIMA</td>
<td>Collimator</td>
<td>99</td>
</tr>
<tr>
<td>DECAPOLE</td>
<td>Decapole magnet</td>
<td>100</td>
</tr>
<tr>
<td>DIPOLE</td>
<td>Dipole magnet, polar frame</td>
<td>101</td>
</tr>
<tr>
<td>DIPOLE-M</td>
<td>Generation of dipole mid-plane 2-D map, polar frame</td>
<td>103</td>
</tr>
<tr>
<td>DIPOLES</td>
<td>Dipole magnet, polar frame</td>
<td>105</td>
</tr>
<tr>
<td>DODECAPO</td>
<td>Dodecapole magnet</td>
<td>109</td>
</tr>
<tr>
<td>DRIFT</td>
<td>Field free drift space</td>
<td>110</td>
</tr>
<tr>
<td>EBMULT</td>
<td>Electro-magnetic multipole</td>
<td>111</td>
</tr>
<tr>
<td>EL2TUB</td>
<td>Two-tube electrostatic lens</td>
<td>112</td>
</tr>
<tr>
<td>ELMIR</td>
<td>Electrostatic N-electrode mirror/lens, straight slits</td>
<td>113</td>
</tr>
<tr>
<td>ELMIRC</td>
<td>Electrostatic N-electrode mirror/lens, circular slits</td>
<td>114</td>
</tr>
<tr>
<td>ELMULT</td>
<td>Electric multipole</td>
<td>115</td>
</tr>
<tr>
<td>ELREVOL</td>
<td>1-D uniform mesh electric field map</td>
<td>117</td>
</tr>
<tr>
<td>EMMA</td>
<td>2-D Cartesian or cylindrical mesh field map for EMMA FFAG</td>
<td>118</td>
</tr>
<tr>
<td>END</td>
<td>End of input data list</td>
<td>56</td>
</tr>
<tr>
<td>ESL</td>
<td>Field free drift space</td>
<td>110</td>
</tr>
<tr>
<td>FAISCEAU</td>
<td>Print particle coordinates</td>
<td>148</td>
</tr>
<tr>
<td>FAISCNL</td>
<td>Store particle coordinates in file FNAME</td>
<td>148</td>
</tr>
<tr>
<td>FAISTORE</td>
<td>Store coordinates every IP other pass at labeled elements</td>
<td>148</td>
</tr>
<tr>
<td>FFAG</td>
<td>FFAG magnet, N-tuple</td>
<td>119</td>
</tr>
<tr>
<td>FFAG-SPI</td>
<td>Spiral FFAG magnet, N-tuple</td>
<td>121</td>
</tr>
<tr>
<td>FIN</td>
<td>End of input data list</td>
<td>56</td>
</tr>
<tr>
<td>FIT,FIT2</td>
<td>Fitting procedure</td>
<td>??</td>
</tr>
<tr>
<td>FOCALE</td>
<td>Particle coordinates and horizontal beam size at distance XL</td>
<td>150</td>
</tr>
<tr>
<td>FOCALEZ</td>
<td>Particle coordinates and vertical beam size at distance XL</td>
<td>151</td>
</tr>
<tr>
<td>GASCAT</td>
<td>Gas scattering</td>
<td>63</td>
</tr>
<tr>
<td>GETFITVAL</td>
<td>Get values of variables as saved from former FIT[2] run</td>
<td>64</td>
</tr>
<tr>
<td>HISTO</td>
<td>1-D histogram</td>
<td>152</td>
</tr>
<tr>
<td>IMAGE</td>
<td>Localization and size of horizontal waist</td>
<td>150</td>
</tr>
<tr>
<td>IMAGES</td>
<td>Localization and size of horizontal waists</td>
<td>150</td>
</tr>
<tr>
<td>IMAGESZ</td>
<td>Localization and size of vertical waists</td>
<td>151</td>
</tr>
<tr>
<td>IMAGEZ</td>
<td>Localization and size of vertical waist</td>
<td>151</td>
</tr>
<tr>
<td>MAP2D</td>
<td>2-D Cartesian uniform mesh field map - arbitrary magnetic field</td>
<td>123</td>
</tr>
<tr>
<td>MAP2D-E</td>
<td>2-D Cartesian uniform mesh field map - arbitrary electric field</td>
<td>124</td>
</tr>
<tr>
<td>MARKER</td>
<td>Marker</td>
<td>125</td>
</tr>
</tbody>
</table>
MATRIX  Calculation of transfer coefficients, periodic parameters .......... 154
MCDESINT  Monte-Carlo simulation of in-flight decay ..................  65
MCOBJET  Monte-Carlo generation of a 6-D object ....................  46
MULTIPOL  Magnetic multipole ........................................... 126
OBJET  Generation of an object .........................................  49
OBJETA  Object from Monte-Carlo simulation of decay reaction .........  53
OCTUPOLE  Octupole magnet ............................................. 127
OPTICS  Write out optical functions ....................................  67
ORDRE  Taylor expansions order ..........................................  68
PARTICUL  Particle characteristics .....................................  69
PICKUPS  Beam centroid path; closed orbit ............................. 156
PLOTDATA  Intermediate output for the PLOTDATA graphic software ... 157
POISSON  Read magnetic field data from POISSON output .............. 128
POLARMES  2-D polar mesh magnetic field map .......................... 129
PS170  Simulation of a round shape dipole magnet ..................... 130
QUADISEX  Sharp edge magnetic multipoles ............................. 131
QUADRUPO  Quadrupole magnet ......................................... 132
REBELOTE  ‘Do it again’ ....................................................  70
RESET  Reset counters and flags .........................................  71
SCALING  Power supplies and R.F. function generator ................  72
SEPARA  Wien Filter - analytical simulation ........................... 134
SEXQUAD  Sharp edge magnetic multipole ............................... 131
SEXTUPOLE  Sextupole magnet .......................................... 135
SOLENOID  Solenoid ....................................................... 136
SPINR  Spin rotation ...................................................... 137
SPNPRNL  Store spin coordinates into file FNAME .................... 158
SPNPRT  Print spin coordinates .......................................... 158
SPNSTORE  Store spin coordinates every IP other pass at labeled elements 158
SPNTRK  Spin tracking .....................................................  74
SRLOSS  Synchrotron radiation loss ....................................  76
SRPRNT  Print SR loss statistics ........................................ 159
SYNRAD  Synchrotron radiation spectral-angular densities ..........  77
TARGET  Generate a secondary beam following target interaction ....  98
TOSCA  2-D and 3-D Cartesian or cylindrical mesh field map .......... 138
TRANSMAT  Matrix transfer ............................................... 140
TRAROT  Translation-Rotation of the reference frame .................. 141
TWISS  Calculation of periodic optical parameters ................... 160
UNDULATOR  Undulator magnet ............................................ 142
UNIPOT  Unipotential cylindrical electrostatic lens .................... 143
VENUS  Simulation of a rectangular shape dipole magnet ............. 144
WIENFILT  Wien filter ..................................................... 145
YMY  Reverse signs of Y and Z reference axes ......................... 146
**Optical elements versus keywords**

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclotron magnet</td>
<td>DIPOLE[S], DIPOLE-M, FFAG, FFAG-SPI</td>
</tr>
<tr>
<td>AGS main magnet</td>
<td>AGSMM</td>
</tr>
<tr>
<td>Decapole</td>
<td>DECAPOLE, MULTIPOLE</td>
</tr>
<tr>
<td>Dipole[s]</td>
<td>AIMANT, BEND, DIPOLE[S], DIPOLE-M, MULTIPOLE, QUADISEX</td>
</tr>
<tr>
<td>Dodecapole</td>
<td>DODECAPO, MULTIPOLE</td>
</tr>
<tr>
<td>FFAG magnet</td>
<td>DIPOLES, FFAG, FFAG-SPI, MULTIPOLE</td>
</tr>
<tr>
<td>Helical dipole</td>
<td>HELIX</td>
</tr>
<tr>
<td>Multipole</td>
<td>MULTIPOLE, QUADISEX, SEXQUAD</td>
</tr>
<tr>
<td>Octupole</td>
<td>OCTRUPOLE, MULTIPOLE, QUADISEX, SEXQUAD</td>
</tr>
<tr>
<td>Quadrupole</td>
<td>QUADRUPOLE, MULTIPOLE, SEXQUAD, AGSQUAD</td>
</tr>
<tr>
<td>Sextupole</td>
<td>SEXTUPOLE, MULTIPOLE, QUADISEX, SEXQUAD</td>
</tr>
<tr>
<td>Skew multipoles</td>
<td>MULTIPOLE</td>
</tr>
<tr>
<td>Solenoid</td>
<td>SOLENOID</td>
</tr>
<tr>
<td>Undulator</td>
<td>UNDULATOR</td>
</tr>
</tbody>
</table>

**Using field maps**

<table>
<thead>
<tr>
<th>Symmetry Type</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-D, cylindrical symmetry</td>
<td>BREVOL</td>
</tr>
<tr>
<td>2-D, mid-plane symmetry</td>
<td>CARTEMES, POISSON, TOSCA</td>
</tr>
<tr>
<td>2-D, no symmetry</td>
<td>MAP2D</td>
</tr>
<tr>
<td>2-D, polar mesh, mid-plane symmetry</td>
<td>POLARMES</td>
</tr>
<tr>
<td>3-D, no symmetry</td>
<td>TOSCA</td>
</tr>
<tr>
<td>EMMA FFAG quadrupole doublet</td>
<td>EMMA</td>
</tr>
<tr>
<td>linear composition of field maps</td>
<td>TOSCA</td>
</tr>
</tbody>
</table>

### ELECTRIC ELEMENTS

<table>
<thead>
<tr>
<th>Element</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-tube (bipotential) lens</td>
<td>EL2TUB</td>
</tr>
<tr>
<td>3-tube (unipotential) lens</td>
<td>UNIPOT</td>
</tr>
<tr>
<td>Decapole</td>
<td>ELMULT</td>
</tr>
<tr>
<td>Dipole</td>
<td>ELMULT</td>
</tr>
<tr>
<td>Dodecapole</td>
<td>ELMULT</td>
</tr>
<tr>
<td>Multipole</td>
<td>ELMULT</td>
</tr>
<tr>
<td>N-electrode mirror/lens, straight slits</td>
<td>ELMIR</td>
</tr>
<tr>
<td>N-electrode mirror/lens, circular slits</td>
<td>ELMIRC</td>
</tr>
<tr>
<td>Octupole</td>
<td>ELMULT</td>
</tr>
<tr>
<td>Quadrupole</td>
<td>ELMULT</td>
</tr>
<tr>
<td>R.F. (kick) cavity</td>
<td>CAVITE</td>
</tr>
<tr>
<td>Sextupole</td>
<td>ELMULT</td>
</tr>
<tr>
<td>Skew multipoles</td>
<td>ELMULT</td>
</tr>
</tbody>
</table>

**Using field maps**

<table>
<thead>
<tr>
<th>Symmetry Type</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-D, cylindrical symmetry</td>
<td>ELREVOL</td>
</tr>
</tbody>
</table>
2-D, no symmetry  MAP2D-E

**ELECTRO-MAGNETIC ELEMENTS**

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decapole</td>
<td>EBMULT</td>
</tr>
<tr>
<td>Dipole</td>
<td>EBMULT</td>
</tr>
<tr>
<td>Dodecapole</td>
<td>EBMULT</td>
</tr>
<tr>
<td>Multipole</td>
<td>EBMULT</td>
</tr>
<tr>
<td>Octupole</td>
<td>EBMULT</td>
</tr>
<tr>
<td>Quadrupole</td>
<td>EBMULT</td>
</tr>
<tr>
<td>Sextupole</td>
<td>EBMULT</td>
</tr>
<tr>
<td>Skew multipoles</td>
<td>EBMULT</td>
</tr>
<tr>
<td>Wien filter</td>
<td>SEPARA, WIENFILT</td>
</tr>
</tbody>
</table>
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 last 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. 7 and 169, 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 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”).

A series of auxiliary computing tools have been developed in addition, 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 losses and their effects. In addition “python” interfaces are being developed by several users, possibly made available on web by their authors.

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 1970’s. 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 Julich, LEP and LHC at CERN), *zgoubi* has also undergone extensive developments regarding storage ring related features.

Many developments have been accomplished since the early 2000s in the frame of a number of project design and beam dynamics studies, as the neutrino factory, lepton and hadron colliders, spin studies at AGS and RHIC, etc. As a consequence the list of optical elements and the compendium of numerical methods, so-called “Glossary of Keywords” list, pp. 7 and 169, has stretched with new simulation and computing procedures, ranging from fitting (*FIT2* procedure; additional constraints) to overlapping magnetic field capabilities (*DIPOLES*, *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 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

$$\frac{d(m\vec{v})}{dt} = q(\vec{e} + \vec{v} \times \vec{b})$$  \hspace{1cm} (1.2.1)

![Figure 1: Reference frame and coordinates (Y, T, Z, P) in zgoubi.](image)

- $OX$: in the direction of motion,
- $OY$: normal to $OX$,
- $OZ$: orthogonal to the $(X,Y)$ plane,
- $\vec{\hat{W}}$: projection of the velocity, $\vec{v}$, in the $(X,Y)$ plane,
- $T$: angle between $\vec{\hat{W}}$ and the $X$-axis,
- $P$: angle between $\vec{\hat{W}}$ and $\vec{v}$. 

Noting \( (\prime) = \frac{d(\prime)}{ds} \) and taking
\[
\vec{u} = \frac{\vec{v}}{v}, \quad ds = v \, dt, \quad \vec{u}' = \frac{d\vec{u}}{ds}, \quad m\vec{v} = mv\vec{u} = q \, B\rho \, \vec{u}
\]
where \( B\rho \) is the rigidity of the particle, this equation can be rewritten
\[
(B\rho)' \, \vec{u} + B\rho \, \vec{u}' = \frac{\vec{e}}{v} + \vec{u} \times \vec{b}
\]
(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)
\[
\vec{R}(M_1) \approx \vec{R}(M_0) + \vec{u}(M_0) \, \Delta s + \vec{u}'(M_0) \frac{\Delta s^2}{2!} + \ldots + \vec{u}''''(M_0) \frac{\Delta s^6}{6!}
\]
(1.2.4)

The rigidity at \( M_1 \) is obtained in the same way from
\[
(B\rho)(M_1) \approx (B\rho)(M_0) + (B\rho)'(M_0) \Delta s + \ldots + (B\rho)'''(M_0) \frac{\Delta s^5}{5!}
\]
(1.2.5)

The equation of time of flight is written in a similar manner
\[
T(M_1) \approx T(M_0) + T'(M_0) \Delta s + \ldots + T'''(M_0) \frac{\Delta s^5}{5!}
\]
(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.
1.2 Integration of the Lorentz Equation

1.2.1 Integration in Magnetic Fields

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

\[
\ddot{\vec{u}} = \vec{u} \times \vec{B}
\]

(1.2.7)

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

\[
\begin{align*}
\vec{u}'' &= \vec{u}' \times \vec{B} + \vec{u} \times \vec{B}' \\
\vec{u}''' &= \vec{u}'' \times \vec{B} + 2\vec{u}' \times \vec{B}' + \vec{u} \times \vec{B}'' \\
\vec{u}'''' &= \vec{u}''' \times \vec{B} + 3\vec{u}'' \times \vec{B}' + 3\vec{u}' \times \vec{B}'' + \vec{u} \times \vec{B}''' \\
\vec{u}''''' &= \vec{u}'''' \times \vec{B} + 4\vec{u}''' \times \vec{B}' + 6\vec{u}'' \times \vec{B}'' + 4\vec{u}' \times \vec{B}''' + \vec{u} \times \vec{B}''''
\end{align*}
\]

(1.2.8)

where \( \vec{B}^{(n)} = \frac{d^n \vec{B}}{d s^n} \).

From

\[
d\vec{B} = \frac{\partial \vec{B}}{\partial X} \, dX + \frac{\partial \vec{B}}{\partial Y} \, dY + \frac{\partial \vec{B}}{\partial Z} \, dZ = \sum_{i=1,3} \frac{\partial \vec{B}}{\partial X_i} \, dX_i
\]

(1.2.9)

and by successive differentiation, we get

\[
\begin{align*}
\vec{B}' &= \sum_i \frac{\partial \vec{B}}{\partial X_i} u_i \\
\vec{B}'' &= \sum_{ij} \frac{\partial^2 \vec{B}}{\partial X_i \partial X_j} u_i u_j + \sum_i \frac{\partial \vec{B}}{\partial X_i} u_i' \\
\vec{B}''' &= \sum_{ijk} \frac{\partial^3 \vec{B}}{\partial X_i \partial X_j \partial X_k} u_i u_j u_k + 3 \sum_{ij} \frac{\partial^2 \vec{B}}{\partial X_i \partial X_j} u_i' u_j + \sum_i \frac{\partial \vec{B}}{\partial X_i} u_i'' \\
\vec{B}'''' &= \sum_{ijkl} \frac{\partial^4 \vec{B}}{\partial X_i \partial X_j \partial X_k \partial X_l} u_i u_j u_k u_l + 6 \sum_{ijk} \frac{\partial^3 \vec{B}}{\partial X_i \partial X_j \partial X_k} u_i' u_j u_k \\
&\quad + 4 \sum_{ij} \frac{\partial^2 \vec{B}}{\partial X_i \partial X_j} u_i'' u_j + 3 \sum_{ij} \frac{\partial^2 \vec{B}}{\partial X_i \partial X_j} u_i' u_j' + \sum_i \frac{\partial \vec{B}}{\partial X_i} u_i''''
\end{align*}
\]

(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

\[
(B_\rho)' \vec{u} + B_\rho \vec{u}' = \frac{\vec{e}}{v}
\]

(1.2.11)

which, by successive differentiations, gives the recursive relations
\[
\begin{align*}
(B\rho)''\dddot{u} + B\rho\dddot{u}' &= \frac{e}{v} \\
(B\rho)'''\dddot{u} + 2(B\rho)'\dddot{u}' + B\rho\dddot{u}'' &= \left(\frac{1}{v}\right)\dddot{e} + \frac{e''}{v} \\
(B\rho)''''\dddot{u} + 3(B\rho)''\dddot{u}' + 3(B\rho)'\dddot{u}'' + B\rho\dddot{u}''' &= \left(\frac{1}{v}\right)^2 \dddot{e} + 2 \left(\frac{1}{v}\right)\dddot{e}' + \left(\frac{1}{v}\right)\dddot{e}''
\end{align*}
\]

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

\[
\begin{align*}
\dddot{u}' &= \left(\frac{1}{v}\right)\dddot{E} - \frac{(B\rho)'}{B\rho} \dddot{u} \\
\dddot{u}'' &= \left(\frac{1}{v}\right)\ddot{E} + \left(\frac{1}{v}\right)\dddot{E}' |_{B\rho} - 2\frac{(B\rho)'}{B\rho} \dddot{u}' - \frac{(B\rho)''}{B\rho} \dddot{u} \\
\dddot{u}''' &= \left(\frac{1}{v}\right)\dddot{E} + 3 \left(\frac{1}{v}\right)\dddot{E}' |_{B\rho} + 6 \left(\frac{1}{v}\right)\dddot{E}'' |_{B\rho} - 3 \left(\frac{B\rho}{B\rho}\right) \dddot{u}''' - 3 \left(\frac{B\rho}{B\rho}\right) \dddot{u}' - \frac{(B\rho)'''}{B\rho} \dddot{u} \\
\dddot{u}'''' &= \left(\frac{1}{v}\right)\dddot{E} + 4 \left(\frac{1}{v}\right)\dddot{E}' |_{B\rho} + 10 \left(\frac{1}{v}\right)\dddot{E}'' |_{B\rho} + 6 \left(\frac{1}{v}\right)\dddot{E}''' |_{B\rho} + 10 \left(\frac{B\rho}{B\rho}\right) \dddot{u}'''' - 5 \left(\frac{B\rho}{B\rho}\right) \dddot{u}''' - \frac{(B\rho)''''}{B\rho} \dddot{u}
\end{align*}
\]

where \( \dddot{E} = \frac{e}{B\rho} \), and \( \left(\frac{\cdot}{B\rho}\right) |_{B\rho} \) denotes differentiation at constant \( B\rho \). These derivatives of the electric field are obtained from the total derivative

\[
d\dddot{E} = \frac{\partial\dddot{E}}{\partial X} dX + \frac{\partial\dddot{E}}{\partial Y} dY + \frac{\partial\dddot{E}}{\partial Z} dZ
\]
\[ E' = \sum_i \frac{\partial E}{\partial X_i} u_i \]

\[ E'' = \sum_{ij} \frac{\partial^2 E}{\partial X_i \partial X_j} u_i u_j + \sum_i \frac{\partial E}{\partial X_i} u'_i \]

\[ E''' = \sum_{ijk} \frac{\partial^3 E}{\partial X_i \partial X_j \partial X_k} u_i u_j u_k + 3 \sum_{ij} \frac{\partial^2 E}{\partial X_i \partial X_j} u'_i u_j + \sum_i \frac{\partial E}{\partial X_i} u''_i \]  

(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

\[ \frac{dp^2}{dt} = \frac{dp^2}{dt} \quad \text{i.e.,} \quad \frac{dp}{dt} p = \frac{dp}{dt} \bar{p} \]  

(1.2.16)

with \(\frac{dp}{dt} = q(\bar{e} + \bar{v} \times \bar{b})\) (eq. 1.2.1), we obtain

\[ \frac{dp}{dt} p = q(\bar{e} + v \times \bar{b}) \cdot \bar{p} = q\bar{e} \cdot \bar{p} \]  

(1.2.17)

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

\[ (B\rho)' = \frac{1}{v} (\bar{e} \cdot \bar{u}) \]

\[ (B\rho)'' = \left( \frac{1}{v} \right)' (\bar{e} \cdot \bar{u}) + \frac{1}{v} (\bar{e} \cdot \bar{u})' \]

\[ (B\rho)''' = \left( \frac{1}{v} \right)'' (\bar{e} \cdot \bar{u}) + 2 \left( \frac{1}{v} \right)' (\bar{e} \cdot \bar{u})' + \frac{1}{v} (\bar{e} \cdot \bar{u})'' \]  

(1.2.18)

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

\[ (\bar{e} \cdot \bar{u})' = \bar{e}' \cdot \bar{u} + \bar{e} \cdot \bar{u}' \]

\[ (\bar{e} \cdot \bar{u})'' = \bar{e}'' \cdot \bar{u} + 2 \bar{e}' \cdot \bar{u}' + \bar{e} \cdot \bar{u}'' \]  

(1.2.19)

etc.

Note that they can be related to the derivatives of the kinetic energy \(W\) by \(dW = \frac{dp}{dt} \cdot \bar{v} dt = q\bar{e} \cdot \bar{v} dt\) which leads to
\[
\frac{d^{n+1}W}{ds^{n+1}} = q \frac{d^n(\vec{e} \cdot \vec{u})}{ds^n}
\]  

(1.2.20)

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

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

\((m_0 \text{ is the rest mass})\) by successive differentiations, that give the recursive relations

\[
\left( \frac{1}{v} \right) = \frac{1}{c^2} \frac{W + m_0 c^2}{qB\rho}
\]

\[
\left( \frac{1}{v} \right)' = \frac{1}{c^2} \frac{(\vec{e} \cdot \vec{u})'}{B\rho} - \frac{1}{v} \frac{(B\rho)'}{B\rho}
\]

\[
\left( \frac{1}{v} \right)'' = \frac{1}{c^2} \frac{(\vec{e} \cdot \vec{u})''}{B\rho} - 2 \left( \frac{1}{v} \right) \frac{(B\rho)'}{B\rho} - \frac{1}{v} \frac{(B\rho)''}{B\rho}
\]

\[
\left( \frac{1}{v} \right)''' = \frac{1}{c^2} \frac{(\vec{e} \cdot \vec{u})'''}{B\rho} - 3 \left( \frac{1}{v} \right) \frac{(B\rho)'}{B\rho} - 3 \left( \frac{1}{v} \right) \frac{(B\rho)''}{B\rho} - \frac{1}{v} \frac{(B\rho)'''}{B\rho}
\]

\[
\left( \frac{1}{v} \right)'''' = \text{etc.}
\]

### 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

\[
(B\rho)'\vec{u} + B\rho\vec{u}' = \frac{\vec{e}}{v} + \vec{u} \times \vec{b}
\]

\[
(B\rho)''\vec{u} + 2(B\rho)'\vec{u}' + B\rho\vec{u}'' = \left( \frac{1}{v} \right)' \vec{e} + \left( \frac{1}{v} \right) \vec{e}' + \frac{1}{v} \vec{e}'' + \left( (\vec{u} \times \vec{b})' \right)
\]

\[
(B\rho)'''\vec{u} + 3(B\rho)''\vec{u}'' + 3(B\rho)'\vec{u}''' + B\rho\vec{u}''' = \left( \frac{1}{v} \right)'' \vec{e} + 2 \left( \frac{1}{v} \right) \vec{e}' + \left( \frac{1}{v} \right) \vec{e}'' + \left( \vec{u} \times \vec{b} \right)''
\]

\[
(B\rho)''''\vec{u} + 4(B\rho)'''\vec{u}'' + 6(B\rho)''\vec{u}''' + 4(B\rho)'\vec{u}'''' + B\rho\vec{u}'''' = \\
\left( \frac{1}{v} \right)''' \vec{e} + 3 \left( \frac{1}{v} \right) \vec{e}'' + 3 \left( \frac{1}{v} \right) \vec{e}''' + \frac{1}{v} \vec{e}'''' + \left( \vec{u} \times \vec{b} \right)'''
\]

(1.2.22)

that provide the derivatives \( \frac{d^n\vec{a}}{ds^n} \) needed in the Taylor expansions (1.2.4)
\[ \vec{u}' = \left( \frac{1}{v} \right) \vec{E} + (\vec{u} \times \vec{B}) - \frac{(B\rho)'}{B\rho} \vec{u} \]

\[ \vec{u}'' = \left( \frac{1}{v} \right) \vec{E} + \left( \frac{1}{v} \right) \vec{E}' |_{B\rho} + (\vec{u} \times \vec{B})' |_{B\rho} - 2 \frac{(B\rho)'}{B\rho} \vec{u}' - \frac{(B\rho)''}{B\rho} \vec{u} \]

\[ \vec{u}''' = \left( \frac{1}{v} \right) \vec{E} + 2 \left( \frac{1}{v} \right) \vec{E}' |_{B\rho} + \frac{1}{v} \vec{E}'' |_{B\rho} + (\vec{u} \times \vec{B})'' |_{B\rho} - 3 \frac{(B\rho)'}{B\rho} \vec{u}'' - 3 \frac{(B\rho)''}{B\rho} \vec{u}' - \frac{(B\rho)'''}{B\rho} \vec{u} \]

\[ \vec{u}'''' = \left( \frac{1}{v} \right) \vec{E} + 3 \left( \frac{1}{v} \right) \vec{E}' |_{B\rho} + 3 \left( \frac{1}{v} \right) \vec{E}'' |_{B\rho} + \left( \frac{1}{v} \right) \vec{E}''' |_{B\rho} + (\vec{u} \times \vec{B})''' |_{B\rho} - 4 \frac{(B\rho)'}{B\rho} \vec{u}''' - 6 \frac{(B\rho)''}{B\rho} \vec{u}'' - 4 \frac{(B\rho)'''}{B\rho} \vec{u}' - \frac{(B\rho)''''}{B\rho} \vec{u} \]

(1.2.23)

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

\[ \vec{E}^{(n)} \mid_{B\rho} = \frac{1}{B\rho} \frac{d^n \vec{e}}{ds^n} \quad \text{and} \quad (\vec{u} \times \vec{B})^{(n)} \mid_{B\rho} = \frac{1}{B\rho} (\vec{u} \times \vec{b})^{(n)}. \]

(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 \( dT / ds = 1 / v, \) \( d^2 T / ds^2 = d(1 / v) / ds, \) etc. that are obtained from eq. (1.2.21). In the absence of electric field eq. (1.2.8) however reduces to the simple form

\[ T(M_1) = T(M_0) + \Delta s / v \]

(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

\[
B_X(X, r) = B_X(X, 0) - \frac{r^2}{4} \frac{\partial^2 B_X}{\partial X^2}(X, 0) + \frac{r^4}{64} \frac{\partial^4 B_X}{\partial X^4}(X, 0) \tag{1.3.1}
\]

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

\[
\begin{align*}
B_X(X, Y, 0) &= 0 \\
B_Y(X, Y, 0) &= 0 \\
B_X(X, Y, Z) &= -B_X(X, Y, -Z) \\
B_Y(X, Y, Z) &= -B_Y(X, Y, -Z) \\
B_Z(X, Y, Z) &= B_Z(X, Y, -Z)
\end{align*} \tag{1.3.2}
\]

Accommodated with Maxwell’s equations, this results in Taylor expansions below, for the three components of $\vec{B}$ (here, $B$ stands for $B_Z(X, Y, 0)$)
1.3 Calculation of \( \mathbf{E} \) and \( \mathbf{B} \) Fields and their Derivatives

\[
B_X(X, Y, Z) = Z \frac{\partial B}{\partial X} - \frac{Z^3}{6} \left( \frac{\partial^3 B}{\partial X^3} + \frac{\partial^3 B}{\partial X \partial Y^2} \right) \\
B_Y(X, Y, Z) = Z \frac{\partial B}{\partial Y} - \frac{Z^3}{6} \left( \frac{\partial^3 B}{\partial X^2 \partial Y} + \frac{\partial^3 B}{\partial Y^3} \right) \\
B_Z(X, Y, Z) = B - \frac{Z^2}{2} \left( \frac{\partial^2 B}{\partial X^2} + \frac{\partial^2 B}{\partial Y^2} \right) + \frac{Z^4}{24} \left( \frac{\partial^4 B}{\partial X^4} + 2 \frac{\partial^4 B}{\partial X^2 \partial Y^2} + \frac{\partial^4 B}{\partial Y^4} \right)
\]  

(1.3.3)

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

1.3.4 Extrapolation from Arbitrary 2-D Field Maps

2-D field maps that give the three components \( B_X(X, Y, Z_0) \), \( B_Y(X, Y, Z_0) \) and \( B_Z(X, Y, Z_0) \) at each node \((X, Y)\) of a \( Z_0 \)-elevation map may be used. \( \mathbf{B} \) 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 \( \mathbf{B} \) 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 \textit{BEND}, \textit{WIENFILT} (that uses the \textit{BEND} procedures), \textit{QUADISEX}, \textit{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 \textit{QUADRUPO}, \textit{SEXTUPO}, \textit{MULTIPOL}, \textit{EBMULT}, etc., the three components of \( \mathbf{B} \) 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

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

(1.3.4)

and similarly for \( \mathbf{E} \) with opposite sign for the gradients.

**Multipoles**

The scalar potential used for the calculation of \( \frac{\partial^{i+j+k} \mathbf{B}_n(X, Y, Z)}{\partial X^i \partial Y^j \partial Z^k} \) \((i + j + k = 0 \text{ to } 4)\) in the case of magnetic and electro-magnetic multipoles with \( 2n \) poles (namely, \textit{QUADRUPO} \((n = 2)\) to \textit{DODECAPO} \((n = 6)\), \textit{MULTIPOL} \((n = 1 \text{ to } 10)\), \textit{EBMULT} \((n = 1 \text{ to } 10)\)) is [9]

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

(1.3.5)

where \( G(X) \) is the longitudinal gradient, defined at the entrance or exit of the optical element by
\[ G(s) = \frac{G_0}{1 + \exp(P(s))}, \quad G_0 = \frac{B_0}{R_0^n} \]  

(1.3.6)

wherein

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

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

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

(1.4.1)

and the coefficients \( A_i \) are calculated by expressions that minimize the quadratic sum

\[ S = \sum_i (B(X, 0) - B_i)^2 \]  

(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
\[ \frac{\partial B}{\partial X} (X, 0) = A_1 + 2A_2X + 3A_3X^2 + 4A_4X^3 + 5A_5X^4 \]
\[ \frac{\partial^2 B}{\partial X^2} (X, 0) = 2A_2 + 6A_3X + 12A_4X^2 + 20A_5X^3 \]
\[ \ldots \]
\[ \frac{\partial^5 B}{\partial X^5} (X, 0) = 120A_5 \]

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.

\texttt{zgoubi} provides three types of polynomial interpolation from the mesh (option \textit{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

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

(1.4.4)
to the second order, or

\[
B(X, Y, 0) = A_{00} + A_{10}X + A_{01}Y + A_{20}X^2 + A_{11}XY + A_{02}Y^2 \\
+ A_{30}X^3 + A_{21}X^2Y + A_{12}XY^2 + A_{03}Y^3
\]

(1.4.5)

\[+ A_{40}X^4 + A_{31}X^3Y + A_{22}X^2Y^2 + A_{13}XY^3 + A_{04}Y^4\]

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

\[
S = \sum_{ij} (B(X, Y, 0) - B_{ij})^2
\]

(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

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

(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)

\[
\frac{\partial B}{\partial X} (X, Y, 0) = A_{10} + 2A_{20}X + A_{11}Y
\]

\[
\frac{\partial B}{\partial Y} (X, Y, 0) = A_{01} + A_{11}X + 2A_{02}Y
\]

(1.4.8)

etc.

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, \((\vec{R}, \alpha, Z)\) to the Cartesian moving frame \((X, Y, Z)\), Fig. 3C. This is done as follows.

In second order calculations the correspondence is (we note \(B \equiv B_Z(Z = 0)\)
The value of \( \mathbf{E} \) is obtained by means of (parameter 1.4)

\[
\begin{align*}
\frac{\partial B}{\partial X} &= 1 - \frac{B}{R} \frac{\partial \alpha}{\partial \alpha} \\
\frac{\partial B}{\partial Y} &= \frac{\partial B}{\partial \alpha} \\
\frac{\partial^2 B}{\partial Y^2} &= \frac{\partial B}{\partial \alpha} \\
\frac{\partial^2 B}{\partial^2 B} &= 1 - \frac{1}{R} \frac{\partial B}{\partial \alpha} \\
\frac{\partial^2 B}{\partial X^2} &= \frac{B}{R^2} \frac{\partial^2 \alpha}{\partial \alpha^2} - \frac{1}{R^2} \frac{\partial^2 B}{\partial \alpha} \\
\frac{\partial^2 B}{\partial X \partial Y} &= \frac{B}{R^2} \frac{\partial \alpha}{\partial \alpha} \frac{\partial R}{\partial \alpha} - \frac{B}{R^2} \frac{\partial^2 \alpha}{\partial \alpha^2} \\
\frac{\partial Y^2}{\partial^2 B} &= \frac{B}{R^2} \frac{\partial \alpha}{\partial \alpha} \\
\frac{\partial^2 B}{\partial X^3} &= \frac{B}{R^3} \frac{\partial^3 \alpha}{\partial \alpha^3} + \frac{3}{R^2} \frac{\partial^2 B}{\partial \alpha^2} - \frac{2}{R^3} \frac{\partial B}{\partial \alpha} \\
\frac{\partial^2 B}{\partial X \partial Y^2} &= \frac{B}{R^2} \frac{\partial \alpha}{\partial \alpha} \frac{\partial R}{\partial \alpha} + \frac{B}{R^2} \frac{\partial^2 \alpha}{\partial \alpha^2} \\
\frac{\partial^2 B}{\partial X^2 \partial Y} &= \frac{B}{R^2} \frac{\partial \alpha}{\partial \alpha} \\
\frac{\partial^2 B}{\partial X \partial Y^3} &= \frac{B}{R^3} \frac{\partial \alpha}{\partial \alpha} \\
\frac{\partial^2 B}{\partial Y^4} &= \frac{B}{R^4}
\end{align*}
\]  

(1.4.9)

In fourth order calculations the relations above are pushed to fourth order in \( X, Y \) whereas

\[
\begin{align*}
\frac{\partial^3 B}{\partial X^3} &= \frac{B}{R^3} \frac{\partial^3 \alpha}{\partial \alpha^3} + \frac{3}{R^2} \frac{\partial^2 B}{\partial \alpha^2} - \frac{2}{R^3} \frac{\partial B}{\partial \alpha} \\
\frac{\partial^3 B}{\partial^2 X \partial Y} &= \frac{B}{R^2} \frac{\partial \alpha}{\partial \alpha} \frac{\partial R}{\partial \alpha} + \frac{B}{R^2} \frac{\partial^2 \alpha}{\partial \alpha^2} \\
\frac{\partial^3 B}{\partial X \partial Y^3} &= \frac{B}{R^3} \frac{\partial \alpha}{\partial \alpha} \\
\frac{\partial^3 B}{\partial Y^4} &= \frac{B}{R^4}
\end{align*}
\]  

(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 4.6.10, p. 166) is then given the value -1.

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

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

The value of \( \mathbf{B} \) 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

\[ B_\ell(X, Y, Z_0) = A_{00} + A_{10}X + A_{01}Y + A_{20}X^2 + A_{11}XY + A_{02}Y^2 \]  \hspace{1cm} (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

\[ \frac{\partial B_\ell}{\partial X} (X, Y, Z_0) = A_{10} + 2A_{20}X + A_{11}Y \]

\[ \frac{\partial^2 B_\ell}{\partial X \partial Y} (X, Y, Z_0) = A_{11} \]

etc. \hspace{1cm} (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,

\[ B_\ell(X, Y, Z) = A_{000} + A_{100}X + A_{010}Y + A_{001}Z + A_{200}X^2 + A_{020}Y^2 + A_{002}Z^2 + A_{110}XY + A_{101}XZ + A_{011}YZ \]  \hspace{1cm} (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
\[ \frac{\partial B_\ell}{\partial X} = A_{100} + 2A_{200}X + A_{110}Y + A_{101}Z \]
\[ \frac{\partial^2 B_\ell}{\partial X^2} = 2A_{200} \] (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 (\( B_\ell^\ell \) 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

\[ S = \sum_{ijk} \left( B_\ell(X, Y, Z) - B_{ijk}^\ell \right)^2 \] (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], and in the magic-energy electrostatic ring of the proton-EDM project [15].

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 [16]

\[
\frac{d\vec{S}}{dt} = \frac{q}{m} \vec{S} \times \vec{\omega} \tag{2.1.1}
\]

with, in the laboratory frame,

\[
\vec{\omega} = (1 + \gamma G) \vec{b} + G(1 - \gamma) \vec{b}_f + \gamma(G + \frac{1}{1 + \gamma}) \vec{e} \times \vec{v} \tag{2.1.2}
\]

\( t, q, m, c, \gamma \) 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, \( \vec{b}_f \) 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 18, 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

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

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

\[
\vec{\Omega} = \frac{\vec{\omega}}{B\rho} = (1 + \gamma G) \vec{B} + G(1 - \gamma) \vec{B}_f + \frac{\beta \gamma}{c} (G + \frac{1}{1 + \gamma}) \vec{E} \times \vec{u} \tag{2.1.4}
\]

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

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

2.2 Integration in Magnetic Fields

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

\[
\vec{\Omega}_b = (1 + \gamma G) \vec{B} + G(1 - \gamma) \vec{B}_f \tag{2.2.1}
\]

\( \vec{S} \) and its derivatives \( \vec{S}^{(n)} = d^n\vec{S}/ds^n \) satisfy the recursive differentiation relations
\[ \tilde{S}' = \tilde{S} \times \tilde{\Omega}_b \]
\[ \tilde{S}'' = \tilde{S}' \times \tilde{\Omega}_b + \tilde{S} \times \tilde{\Omega}'_b \]
\[ \tilde{S}''' = \tilde{S}'' \times \tilde{\Omega}_b + 2\tilde{S}' \times \tilde{\Omega}'_b + \tilde{S} \times \tilde{\Omega}''_b \] (2.2.2)

etc.

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

\[ \tilde{B}_I = (\tilde{B} \cdot \tilde{u}) \tilde{u} \]
\[ \tilde{B}_I' = (\tilde{B}' \cdot \tilde{u} + \tilde{B} \cdot \tilde{u}') \tilde{u} + (\tilde{B} \cdot \tilde{u}) \tilde{u}' \]
\[ \tilde{B}_I'' = (\tilde{B}'' \cdot \tilde{u} + 2\tilde{B}' \cdot \tilde{u}' + \tilde{B} \cdot \tilde{u}'') \tilde{u} + 2(\tilde{B}' \cdot \tilde{u} + \tilde{B} \cdot \tilde{u}'') \tilde{u}' + (\tilde{B} \cdot \tilde{u}) \tilde{u}'' \] (2.2.3)

etc.

The quantities \( \tilde{u}, \tilde{B} \) 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. 19.

2.3 Integration in Electric Fields

In purely electric fields \( \tilde{b} = 0 \), thus eq. (2.1.2) reduces to

\[ \omega_e = \frac{\beta \gamma}{c} (G + \frac{1}{1 + \gamma}) \tilde{e} \times \tilde{u} \] (2.3.1)

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

\[ B_{\rho} \tilde{S}' = \tilde{S} \times \tilde{\omega}_e \]
\[ (B_{\rho})' \tilde{S}' + B_{\rho} \tilde{S}'' = \tilde{S}' \times \tilde{\omega}_e + \tilde{S} \times \tilde{\omega}'_e \]
\[ (B_{\rho})'' \tilde{S}' + 2(B_{\rho})' \tilde{S}'' + B_{\rho} \tilde{S}''' = \tilde{S}'' \times \tilde{\omega}_e + 2\tilde{S}' \times \tilde{\omega}'_e + \tilde{S} \times \tilde{\omega}''_e \] (2.3.2)

that provide the derivatives \( d^n \tilde{S} / ds^n \) 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,

\[ \tilde{\omega}'_e = \left( \frac{\beta \gamma}{c} (G + \frac{1}{1 + \gamma}) \right)' \tilde{e} \times \tilde{u} + \frac{\beta \gamma}{c} (G + \frac{1}{1 + \gamma}) (\tilde{e} \times \tilde{u})' \]
\[ \tilde{\omega}''_e = \left( \frac{\beta \gamma}{c} (G + \frac{1}{1 + \gamma}) \right)'' \tilde{e} \times \tilde{u} + 2 \left( \frac{\beta \gamma}{c} (G + \frac{1}{1 + \gamma}) \right)' (\tilde{e} \times \tilde{u})' + \frac{\beta \gamma}{c} (G + \frac{1}{1 + \gamma}) (\tilde{e} \times \tilde{u})'' \] (2.3.3)

etc.

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,

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

$$\left(\frac{1}{1+\gamma}\right)'' = \left(\frac{1}{\beta}\right)''\left(\frac{1}{\beta} + 1\right)(\gamma - 1) + 2\left(\frac{1}{\beta}\right)'\left(\frac{1}{\beta}\right)'(\gamma - 1) + 2\left(\frac{1}{\beta} - 1\right)\left(\frac{1}{\beta}\right)'\gamma' +$$

etc.  

(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

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

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

$$B_\rho \vec{S}' = \vec{S} \times \vec{\omega}_b$$

$$B_\rho \vec{S}'' = \vec{S}' \times \vec{\omega}_b + \vec{S} \times \vec{\omega}_b'$$

$$B_\rho \vec{S}''' = \vec{S}'' \times \vec{\omega}_b + 2\vec{S}' \times \vec{\omega}_b' + \vec{S} \times \vec{\omega}_b'' - (B_\rho)' \vec{S}'$$

Magnetic field component, eq. (2.2.2)  

$$\text{Electric field component, eq. (2.3.2)}$$

etc.

(2.4.1)

The process is then completed by applying the $\Delta s$ push, eq. (2.1.5).
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 [17], based on a method developed in the frame of the ELFE project (“EU Lab. For Electrons”) [18, 19, 20]. It was next used for including damping effects in beam studies regarding various electron-ring and recirculator projects [12, 21].

**SR electromagnetic impulse and spectrum** computation was first installed in *zgoubi* for the study of interference effects at the LEP beam diagnostics miniwigglers [23]. These simulation tools were next used to design the LHC SR beam diagnostics systems, located in the IR4 RF section [24].

3.1 Energy Loss and Dynamical Effects

Most of the contents of the present section are drawn from Ref. [17].

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)$^1$ a Poisson probability law

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

is considered. $k$ is the number of photons emitted over a $\Delta\theta$ (circular) arc of trajectory such that, the mean number of photons per radian expresses as$^2$

---

$^1$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.

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

where $r_0 = e^2/4\pi \varepsilon_0 m_0 c^2$ 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, $\beta = v/c$, $B_0$ is the particle stiffness. $\lambda$ is evaluated at each integration step from the current values $\beta$, $B_0$ 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 3

$$
P(\epsilon/\epsilon_c) = \frac{3}{5\pi} \int_{0}^{\epsilon/\epsilon_c} K_{5/3}(x)dx$$

(3.1.3)

where $K_{5/3}$ is a modified Bessel function and, $\epsilon_c = \hbar \omega_c$, $\omega_c = 2\pi\gamma^3 c/2\rho$ 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

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

(3.1.4)

About 40 values of $P(\epsilon/\epsilon_c)$ computed from eq. 3.1.3 [22], 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 $0 < P < 1$ is generated uniformly, then $\epsilon/\epsilon_c$ is drawn either by simple inverse linear interpolation of the tabulated values if $P > 0.26$ (corresponding to $\epsilon/\epsilon_c > 10^{-2}$), or, if $P < 0.26$ from eq. 3.1.4 that directly gives $\epsilon/\epsilon_c = \left(\frac{5 \Gamma(1/3)}{12\sqrt{3}P}\right)^3$ with precision no less than 1% at $P \rightarrow 0.26$.

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

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

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

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

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

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

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.

3From 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. [23, 24, 25].

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\(^4\), this is developed in section 3.2.2.

\[ \vec{E}(\vec{n}, \tau) = \frac{q}{4\pi\varepsilon_0 c} \frac{\vec{n}(t) \times \left[ \left( \vec{n}(t) - \vec{\beta}(t) \right) \times \frac{d\vec{\beta}}{dt} \right]}{r(t) \left( 1 - \vec{n}(t) \cdot \vec{\beta}(t) \right)^3} \]  

(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

\(^4\)These calculations have been installed in the post-processor \( zpop \).
Figure 6: Left: typical shapes of the $E_{\sigma}(t)$ and $E_{\pi}(t)$ 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. [25].

$$\frac{d\tau}{dt} = 1 - \vec{n}(t) \cdot \vec{\beta}(t)$$  \hspace{1cm} (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 $\vec{e} = 0$)

$$\frac{d\vec{\beta}}{dt} = \frac{q}{m} \vec{\beta}(t) \times \vec{b}(t)$$  \hspace{1cm} (3.2.3)

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

$$\vec{r}(t) = \vec{X} - \vec{R}(t) \quad \text{and} \quad \vec{n}(t) = \frac{\vec{r}(t)}{|\vec{r}(t)|}$$  \hspace{1cm} (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 [25]

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

**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

$$\vec{n} = (n_x, n_y, n_z) = (\cos \psi \cos \phi, \cos \psi \sin \phi, \sin \psi)$$

$$= \left[1 - 2(\sin^2 \phi/2 + \sin^2 \psi/2) + 4 \sin^2 \phi/2 \sin^2 \psi/2, \sin \phi(1 - 2 \sin^2 \psi/2), \sin \psi \right]$$  \hspace{1cm} (3.2.5)

where $\phi$ and $\psi$ are the observation angles, given by
\[ \phi = \text{Atg}\left(\frac{r_y}{r_z}\right) \quad \text{and} \quad \psi = \text{Atg}\left(\frac{r_z}{\sqrt{r_x^2 + r_y^2}}\right) \] (3.2.6)

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

\[ \vec{\beta} = (\beta_x, \beta_y, \beta_z) = \left[ \sqrt{(\beta_x^2 - \beta_y^2 - \beta_z^2)}, \beta_y, \beta_z \right] = \left[ \sqrt{(1 - 1/\gamma^2 - \beta_y^2 - \beta_z^2)}, \beta_y, \beta_z \right] = (1 - a/2 + a^2/8 - a^3/16 + \ldots, \beta_y, \beta_z) \] (3.2.7)

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

\[ n_x = 1 - \varepsilon_x \quad \text{and} \quad \beta_x = 1 - \xi_x \]

with

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

and

\[ \xi_x = a/2 - a^2/8 + a^3/16 + \ldots \]

All this provides, on the one hand,

\[ \vec{n} - \vec{\beta} = (-\varepsilon_x + \xi_x, n_y - \beta_y, n_z - \beta_z) \] (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,

\[ 1 - \vec{n} \cdot \vec{\beta} = \varepsilon_x + \xi_x - n_y \beta_y - n_z \beta_z - \varepsilon_x \xi_x \] (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

\[ \xi_x = a/2 - a^2/8 + a^3/16 + \ldots \quad (a = 1/\gamma^2 + \beta_y^2 + \beta_z^2) \]

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

\[ \text{FT}_{\omega}[\vec{E}(\tau)] = \int \vec{E}(\tau)e^{-i\omega\tau} d\tau \]

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

\[ \frac{\partial^3 W}{\partial \phi \partial \psi \partial \omega} = \frac{r^2}{\mu_0 c} \left| \text{FT}_{\omega}\left(\vec{E}(\tau)\right) \right|^2 \] (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

\[ \text{FT}_{\omega}\left[\vec{E}(\tau)\right] \approx \sum \vec{E}(\tau_k) e^{-i\omega \tau_k} \Delta \tau_k \] (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 DESCRIPTION OF THE AVAILABLE PROCEDURES

4.1 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 4.2 to 4.5 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 4.6 addresses further a series of functionalities that may be accessed by means of special input data or flags.
4.2 Definition of an Object

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

Several types of automatically generated objects are available, they are described in the following pages and include,
- non-random object, with various distributions: individual particles, grids, object for *MATRIX*, etc.
- Monte Carlo distribution, with various distributions as well: 6-D window, ellipsoids, etc.

A recurrent quantity appearing in these procedures is *IMAX*, the number of particles to be ray-traced. The maximum value allowed for *IMAX* can be changed at leisure in the include file ‘*MAXTRA.H*’ where it is defined (that requires re-compiling *zgoubi*).
MCOBJET : Monte-Carlo generation of a 6-D object

*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, \( D = \delta D \ast T \).

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 \( x - x_0 \) (\( \equiv Y - Y_0, T - T_0, \ldots \)) and \( D - D_0 \) respectively. Negative value for \( D_0 \) is allowed (see section 4.6.11, page 166).

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

\( \delta Y, \delta T, \delta Z, \delta P, \delta X, \delta D \)

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

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

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,
4.2 Definition of an Object

\[ IR_1, \ IR_2, \ IR_3 \ (all \ \simeq 10^6) \]

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 \text{IMAX} = IY * IT * IZ * IP * IX * ID \text{ particles on a hyper-grid}. The input data are the number of bars in each coordinate

\[ IY, \ IT, \ IZ, \ IP, \ IX, \ ID \]

the spacing of the bars

\[ PY, \ PT, \ PZ, \ PP, \ PX, \ PD \]

the width of each bar

\[ \delta Y, \ \delta T, \ \delta Z, \ \delta P, \ \delta X, \ \delta D \]

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

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

This is illustrated in Fig. 7.

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

\[ N_0, \ C_0, \ C_1, \ C_2, \ C_3 \]

needed for generation of the \text{\textit{D}} coordinate upon option KD= 2 (unused if KD= 1, 3) and a set of three integer seeds for initialization of random sequences, \text{IR1}, \text{IR2}, and \text{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 \text{IMAX} particles inside a 6-D ellipsoid defined by the three sets of data (one set per 2-D phase-space)

\[ \alpha_Y, \ \beta_Y, \ \epsilon_Y, \ N_{\epsilon_Y}, \ N'_{\epsilon_Y}, \text{if} \ N_{\epsilon_Y} < 0 \]
\[ \alpha_Z, \ \beta_Z, \ \epsilon_Z, \ N_{\epsilon_Z}, \ N'_{\epsilon_Z}, \text{if} \ N_{\epsilon_Z} < 0 \]
\[ \alpha_X, \ \beta_X, \ \epsilon_X, \ N_{\epsilon_X}, \ N'_{\epsilon_X}, \text{if} \ N_{\epsilon_X} < 0 \]

where \(\alpha, \beta\) are the ellipse parameters and \(\epsilon/\pi\) the \textit{rms} emittance, corresponding to an elliptical frontier

\[
\frac{1 + \alpha_Y^2}{\beta_Y^2} Y^2 + 2\alpha_Y Y T + \beta_Y T^2 = \epsilon_Y/\pi \quad (\text{idem for the (Z, P) or (X, D) planes}). \]

\(N_{\epsilon_Y}, N_{\epsilon_Z}\) and \(N_{\epsilon_X}\) are the sorting cut-offs (used only for Gaussian distributions, \(K_Y, K_T, ..., = 2\)).

The sorting is uniform in surface (for \(K_Y = 1\) or \(K_Z = 1\) or \(K_X = 1\)) or Gaussian (\(K_Y = 2\) or \(K Z = 2\)), and so on, as described above. A uniform sorting has the ellipse above for support. A Gaussian sorting

has the ellipse above for r.m.s. frontier, leading to \(\sigma_Y = \sqrt{\beta_Y \epsilon_Y/\pi}, \sigma_T = \sqrt{(1 + \alpha_Y^2) \beta_Y \epsilon_Y/\pi}\), and similar relations for \(\sigma_Z, \sigma_P, \sigma_X, \sigma_D\).

If \(N_{\epsilon}\) is negative, thus the sorting fills the elliptical ring that extends from \(|N_{\epsilon}|\) to \(N'_{\epsilon}\) (rather than the inner region determined by the \(N_{\epsilon}\) cut-off as discussed above).
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.
4.2 Definition of an Object

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 eight different ways:

KOBJ = 1: Defines a grid in the \(Y, T, Z, P, X, D\) space. One gives the number of points desired

\[ IY, IT, IZ, IP, IX, ID \]

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

\[ PY, PT, PZ, PP, PX, PD \]

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

\[
\begin{align*}
0, & \pm PY, \pm 2 \times PY, \ldots, \pm IY/2 \times PY, \\
0, & \pm PT, \pm 2 \times PT, \ldots, \pm IT/2 \times PT, \\
0, & \pm PZ, \pm 2 \times PZ, \ldots, \pm IZ/2 \times PZ, \\
0, & \pm PP, \pm 2 \times PP, \ldots, \pm IP/2 \times PP, \\
0, & \pm PX, \pm 2 \times PX, \ldots, \pm IX/2 \times PX, \\
0, & \pm PD, \pm 2 \times PD, \ldots, \pm ID/2 \times PD,
\end{align*}
\]

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 incompatible with the use of some procedures (e.g. MCDESINT, which involves random processes).

The last datum is a reference in each coordinate, \(YR, TR, ZR, PR, XR, DR\). For instance the reference rigidity is \(DR \ast BORO\), resulting in the rigidity of a particle of initial condition \(I \ast PD\) to be \((DR + I \ast PD) \ast BORO\).

KOBJ = 1.01: Same as KOBJ= 1 except for the Z symmetry. The initial Z and P conditions are the following

\[
\begin{align*}
0, & PZ, 2 \times PZ, \ldots, (IZ - 1) \ast PZ, \\
0, & PP, 2 \times PP, \ldots, (IP - 1) \ast PP,
\end{align*}
\]

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

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 \(D \ast BORO\) is the rigidity (negative value allowed) and ‘A’ is a (arbitrary) tagging character.
The last record \textit{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 \textit{KOBJ} = 2 may be be useful for the definition of objects including kinematic effects.

\textbf{KOBJ = 2.01}: Same as \textit{KOBJ} = 2 except for the units, meter and radian in that case.

\textbf{KOBJ = 3} : This option allows the reading of initial conditions from an external input file \textit{FNAME}. The next three data lines are:

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

followed by the storage file name \textit{FNAME}.

\textit{IT1, IT2, ITStep} cause the code to read coordinates of particles number \textit{IT1} through \textit{IT2} by step \textit{ITStep}.

\textit{IP1, IP2, IPStep} cause the code to read coordinates belonging in the passes range \textit{IP1} through \textit{IP2}, step \textit{IPStep}.

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

If \textit{InitC} = 1 ray-tracing starts from the current coordinates \(F(J,I)\), if \textit{InitC} = 0 ray-tracing starts from the initial coordinates \(FO(J,I)\), as read from file \textit{FNAME}.

The file \textit{FNAME} must be formatted in the appropriate manner. The following \texttt{FORTRAN} sequence is an instance, details and possible updates are to be found in the source file ‘\texttt{obj3.f}’:

```fortran
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),
  D, D, BORO, IPASS, KLEY, LBL1,LBL2,NOEL
100 FORMAT (1X,
  C1 LET(IT),KEX, 1.D0-FO(1,IT), (FO(J,IT),J=1,6), (F(J,IT),J=1,6), I, IREP (IT),
  > LET (IT),IEX (IT),-1.D0+FO (1,IT), (FO (J,IT),J=2,MXJ),
  > -1.D0+F (1,IT),F (2,IT),F (3,IT),
  > (F (J,IT),J=4,MXJ),ENEKI,
  > ID, IT,IREP (IT), SORT (IT),D,D,D,D,RET (IT),
  D, D, BORO, IPASS, KLEY, LBL1,LBL2,NOEL
C1 KART, IT,IREP (IT),SORT (IT),X,BX,BY,BZ, RET (IT), DPR (IT),
C2 1.,3E24.16,
C3 2.,ZP*1.D3,SAR, TAR, DS,
C4 3.,4.E24.16,6.E16.8,
C5 4.,11,216,7.E16.8,
C5 5.,4.E16.8, 16,1X, A8,1X, 2A10,15)
ENDDO
```

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

\texttt{LET(I)} : one-character string (for tagging)
\texttt{IEX(I)} : flag, see \textit{KOBJ} = 2 and page 166
\texttt{FO(1-6,I)} : coordinates \(D, Y, T, Z, P\) and path length of particle number \(I\), at the origin. \(D * \text{BORO} = \text{rigidity}\)
\texttt{F(1-6,I)} : id, at the current position.

\textit{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.
4.2 Definition of an Object

$KOBJ = 3$ can be used directly 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.

$KOBJ = 3.01$: Same as $KOBJ = 3$, except for the formatting of trajectory coordinate data in $FNAME$, namely, according to the following FORTRAN sequence:

```fortran
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
```

$KOBJ = 3.02$: As for $KOBJ=3.01$, except for the different format:

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

where $PX$, $PY$, and $PZ$, are the momenta in $\text{MeV}/c$. Note that DPR will be ignored in this case.

$KOBJ = 3.03$: As for $KOBJ=3.01$, except for the different format:

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

where $MASS$ is the mass in $\text{MeV}/c$ 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 3.01-3.03, see the source file ‘$obj3.f$’.

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

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

The code generates 11 particles, with initial coordinates:

$$0, \pm PY, \pm PT, \pm PZ, \pm PP, \pm PX, \pm PD$$

These values should be small enough, so that the paraxial ray approximation be valid. The last data line gives the reference:

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

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

$KOBJ = 5.01$: 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 for possible use by the FIT[2] procedure.

$KOBJ = 5.NN$: Like $KOBJ = 5$, except for $NN = 02 - 99$ references needed in this case (thus $NN-1$ additional input data lines), rather than just one. Zgoubi will generate $NN$ sets of 11 particles with initial coordinates in each set taken wrt. one of the $NN$ references.

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

$KOBJ = 6$: Mostly dedicated to the calculation of first, second and other higher order transfer coefficients and various other optical parameters, using for instance $MATRIX$. The input data are the coordinate sampling (normally taken paraxial)
PY, PT, PZ, PP, PX, PD
to allow the building up of an object containing 61 particles (note: their coordinates can be checked by
printing out into zgoubi.res using FAISCEAU), whereas a last data line gives the reference
YR, TR, ZR, PR, XR, DR
(with $DR \ast BORO$ the reference rigidity - negative value allowed), which adds to the previous coordinate
values.

**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

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

while $T_I$ is in the range

$$0, \pm PT, \pm 2 \ast PT, \ldots, \pm IT/2 \ast PT$$
as stated under 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)

$$\alpha_Y, \beta_Y, \varepsilon_Y/\pi$$
$$\alpha_Z, \beta_Z, \varepsilon_Z/\pi$$
$$\alpha_X, \beta_X, \varepsilon_X/\pi$$

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

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

(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.
4.2 Definition of an Object

OBJETA : Object from Monte-Carlo simulation of decay reaction [27]

This generator simulates the reactions

\[ M_1 + M_2 \rightarrow M_3 + M_4 \]

and then

\[ M_4 \rightarrow M_5 + M_6 \]

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 :

\[ p + d \rightarrow ^3\text{He} + \eta \]
\[ \eta \rightarrow \mu^+ + \mu^- \]

The first input data are the reference rigidity

\[ BORO = p_0/q \]

an index \( IBODY \) which specifies the particle to be ray-traced, namely \( M_3 (IBODY = 1) \), \( M_5 (IBODY = 2) \) or \( M_6 (IBODY = 3) \). In this last case, initial conditions for \( M_6 \) 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.

\[ M_1, M_2, M_3, M_4, M_5, M_6 \]

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

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

\[ Y_0, T_0, Z_0, P_0, D_0 \]

and the width of the distribution around the central value

\[ \delta Y, \delta T, \delta Z, \delta P, \delta D \]

so that 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. 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.
4.3 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,
- **GETFITVAL**: get parameter values as saved by earlier run of FIT[2],

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.
- **REBELOTE**: for tracking more than \( IMAX \) particles (the maximum value for \( IMAX \) is defined in the include file ‘MAXTRA.H’).
4.3 Declaring Options

**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.

If \(J\) is not given, the \(NCOL\) arrangement should be consistent with the following *FORTRAN* READ statement:

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

If \(J = 1\), \(NCOL\) should be consistent with the following *FORTRAN* READ statement:

```fortran
READ (unit=ln, fmt='(1x, ncol * E11.2)') (X7(I), I=1, NCOL)
```

Then follow, line by line, the NF names of the files to be translated.

*If* 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’. 
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.
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 drawn from the matrix transport code BETA [28]. FIT2 is a simplex method (Nelder-Mead method), it has been implemented next [29]. 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 [30].

A full list of the constraints available is given in the table page 61.

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

\[
IR = \text{number of the varied element in the structure} \\
IP = \text{number of the physical parameter to be varied in this element} \\
XC = \text{coupling parameter. Normally } XC = 0. \text{ If } XC \neq 0, \text{ coupling will occur (see below)}
\]

followed by, either

\[
DV = \text{allowed relative range of variation of the physical parameter } IP
\]

or

\[
[V_{min}, V_{max}] = \text{allowed interval of variation of the physical parameter } IP
\]

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:

```
<table>
<thead>
<tr>
<th>Input data</th>
<th>Numbering for FIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>'KEYWORD'</td>
<td>1, 2, 3, ..., 9</td>
</tr>
<tr>
<td>first line</td>
<td>10, 11, 12, 13, ..., 19</td>
</tr>
<tr>
<td>second line</td>
<td>this is a comment</td>
</tr>
<tr>
<td>next line</td>
<td>a line of comments is skipped</td>
</tr>
<tr>
<td></td>
<td>20, 21, 22, ..., 29</td>
</tr>
</tbody>
</table>
```

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 **DIPOLE-M**:

```
<table>
<thead>
<tr>
<th>Input data</th>
<th>Numbering for FIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>'DIPOLE-M'</td>
<td>1, 2, 3</td>
</tr>
<tr>
<td>NFACE, IC, IL</td>
<td>4, 5</td>
</tr>
<tr>
<td>IAMAX, IRMAX</td>
<td>6, 7, 8, 9</td>
</tr>
<tr>
<td>B0, N, B, G</td>
<td>10, 11, 12, 13, 14</td>
</tr>
<tr>
<td>AT, ACENT, RM, RMIN, RMAX</td>
<td>15,16</td>
</tr>
<tr>
<td>( \lambda ), ( \xi )</td>
<td>17, 18, 19, 20, 21, 22, 23, 24</td>
</tr>
<tr>
<td>NC, C0, C1, C2, C3, C4, C5, shift</td>
<td>25, 26, 27, 28, 29, 30</td>
</tr>
<tr>
<td>( \omega ), ( \theta ), R1, U1, U2, R2</td>
<td>etc.</td>
</tr>
</tbody>
</table>
```

The examples of **QUADRUPO** (quadrupole) and **TOSCA** (Cartesian or cylindrical mesh field map) are as follows.

```
<table>
<thead>
<tr>
<th>Input data</th>
<th>Numbering for FIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>'QUADRUPO'</td>
<td>1</td>
</tr>
<tr>
<td>IL</td>
<td>10, 11, 12</td>
</tr>
<tr>
<td>XL, R0, B</td>
<td>20, 21</td>
</tr>
<tr>
<td>( X_E, \lambda_E )</td>
<td>30, 31, 32, 33, 34, 35, 36</td>
</tr>
<tr>
<td>NC0, C1, C2, C3, C4, C5</td>
<td>40, 41</td>
</tr>
<tr>
<td>Xs, ( \lambda_s )</td>
<td>50, 51, 52, 53, 54, 55, 56</td>
</tr>
<tr>
<td>NCS, C0, C1, C2, C3, C4, C5</td>
<td>60</td>
</tr>
<tr>
<td>XPAS</td>
<td>70, 71, 72, 73</td>
</tr>
</tbody>
</table>
```

```
| TOSCA            | 1, 2 |
| IC, IL           | 10, 11 [, 12, 13] |
| BNORM, X- [Y, Z]-NORM | 20, 21, 22, 23 |
| TIT              | This is text |
| IX, IY, IZ, MOD  | 30, 31, 32, 33 [34, 35, 36 [, 37, 38, 39] if ID \( \geq 2 \)] |
| FNAME            | 40 |
| ID, A, B, C [A', B', C', etc. if ID \( \geq 2 \)] | 50 |
| IORDRE           | 60, 61, 62, 63    |
```
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):

<table>
<thead>
<tr>
<th>Input data</th>
<th>Numbering for FIT</th>
<th>Quantities to be varied</th>
</tr>
</thead>
<tbody>
<tr>
<td>'SCALING'</td>
<td>1 2</td>
<td>Non relevant</td>
</tr>
<tr>
<td>AGSMM *AF *BF</td>
<td>3 4 5</td>
<td>Keywords concerned, their labels</td>
</tr>
<tr>
<td>-1 3 12</td>
<td>6</td>
<td>dB1, dB2, dB3 parameters in AGSMM</td>
</tr>
<tr>
<td>7.2135</td>
<td>7</td>
<td>Field factor</td>
</tr>
<tr>
<td>AGSMM *AD *BD</td>
<td>8 9 10</td>
<td>Timing</td>
</tr>
<tr>
<td>-1 3 12</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>7.2135</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>AGSMM *CF</td>
<td>13 14 15</td>
<td>Field factor</td>
</tr>
<tr>
<td>-1 3 12</td>
<td>16</td>
<td>Timing</td>
</tr>
<tr>
<td>7.2135</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>AGSQUAD QH_*</td>
<td>18 19 20</td>
<td>Field factor</td>
</tr>
<tr>
<td>3</td>
<td>21 22 23</td>
<td>Timing</td>
</tr>
<tr>
<td>AGSQUAD QV_*</td>
<td>24 25 26</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>27 28 29</td>
<td></td>
</tr>
<tr>
<td>MULTIPOL COH1</td>
<td>30 31</td>
<td>Field factor</td>
</tr>
<tr>
<td>1.10</td>
<td>32 33</td>
<td>Timing</td>
</tr>
<tr>
<td>./Csnk3D/bump_centered.scal</td>
<td>No numbering if 1.10 type of option</td>
<td></td>
</tr>
</tbody>
</table>
**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 $r \cdot ppp$ where the integer part $r$ is the number of the coupled element in the structure (equivalent to $IR$, see above), and the decimal part $ppp$ 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 or 100, ...), then $ppp$ must take the form 00p (resp. 0pp, ppp)). For example, $XC = 20 \cdot 010$ is a request for coupling with the parameter number 10 of element number 20 of the structure, while $XC = 20 \cdot 100$ 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).

**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 $v \times (1 \pm DV)$.

(ii) $[v_{\text{min}}, v_{\text{max}}]$ : 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 61; adding or changing a constraint resorts to the FORTRAN file ff.f.

Each constraint is defined by the following list of data:

- $IC$ = type of constraint (see table p. 61).
- $I$, $J$ = constraint (i.e., $R_{ij}$, determinant, tune; $T_{ijk}$; $\sigma_{ij}$; trajectory #I and coordinate #J)
- $IR$ = number of the keyword at the exit of which the constraint applies
- $V$ = desired value of the constraint
- $W$ = weight of the constraint (smaller $W$ for higher weight)
- $NP$ = NP values follow

$IC=0$ : 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: $\sigma_{11} = \beta_Y, \sigma_{12} = -\alpha_Y, \sigma_{22} = -\alpha_Y, \sigma_{33} = \beta_Z, \sigma_{34} = -\alpha_Z, \sigma_{44} = \gamma_Z$; 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 $Icos\mu + Jsin\mu$.

$IC=1$, 2 : The coefficients $R_{ij}$ and $T_{ijk}$ are calculated following the procedures described in MATRIX, option $IFO\text{C}=0$. 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.
### 4.3 Declaring Options

<table>
<thead>
<tr>
<th>Type of constraint</th>
<th>Parameters defining the constraints</th>
<th>Object definition (recommended)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma$-matrix</td>
<td>$\sigma_{1,1}$ ($\sigma_{11} = 3\beta_Y, \sigma_{12} = \sigma_{21} = 3\alpha_Y$, etc.)</td>
<td>OBJET/KOBJ=$5.6$</td>
</tr>
<tr>
<td>Periodic parameters</td>
<td>$\sigma_{1,1}$ ($\sigma_{11} = \cos/\mu_Y + 3\alpha_Y \sin/\mu_Y$, etc.)</td>
<td>OBJET/KOBJ=$5.0N$</td>
</tr>
<tr>
<td>(N=1-9 for MATRIX block 1-9)</td>
<td>$\sigma_{1,7}$ ($\mu_Y = \mu_Z/2\pi$)</td>
<td>OBJET/KOBJ=$5$</td>
</tr>
<tr>
<td>First order transport coeffs.</td>
<td>$\sigma_{1,7}$ ($\mu_Y = \mu_Z/2\pi$)</td>
<td>OBJET/KOBJ=$5$</td>
</tr>
<tr>
<td>Second order transport coeffs.</td>
<td>$\sigma_{1,7}$ ($\mu_Y = \mu_Z/2\pi$)</td>
<td>OBJET/KOBJ=$6$</td>
</tr>
<tr>
<td>Trajectory coordinates</td>
<td>$F(J, I)$</td>
<td>[MC]OBJET</td>
</tr>
<tr>
<td>Ellipse parameters</td>
<td>$\sigma_{1,1}$ ($\sigma_{11} = 3\beta_Y, \sigma_{12} = \sigma_{21} = 3\alpha_Y$, etc.)</td>
<td>OBJET/KOBJ=$8$; MCOBJET/KOBJ=$3$</td>
</tr>
<tr>
<td>Number of particles</td>
<td>$N_{\text{survived}}/\text{IMAX}$</td>
<td>OBJET</td>
</tr>
<tr>
<td>Spin</td>
<td>$S_{X,Y,Z}(I)$, $</td>
<td>\vec{S}(I)</td>
</tr>
</tbody>
</table>

**IC=3:** If $1 \leq I \leq \text{IMAX}$ then the value of coordinate type $J$ ($J = 1, 6$ for respectively $D, Y, T, Z, P, S$) of particle number $I$ ($1 \leq I \leq \text{IMAX}$) is constrained. However $I$ can take special meaning, as follows.
- $I = -1$ : the constraint is the mean value of coordinate of type $J$,
- $I = -2$ : the constraint is the maximum value of coordinate of type $J$,
- $I = -3$ : the constraint is the distance between two different particles.

**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 $I$.

**IC=3.3:** Minimum ($\text{NP}=1$) or maximum value ($\text{NP}=2$) of the local $J$-coordinate of particle $I$.

**IC=3.4:** Absolute value of the difference between local $J$-coordinates of particles respectively $I$ and $K$.

**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 $\text{MCOBJET, KOBJ}=3$).

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 $\text{IMAX}$). 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, $\text{NP}=1$, 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 $\text{NP}=0$. 
IC=10: If $1 \leq I \leq IMAX$ then the value of coordinate type $J$ ($J = 1, 3$ for respectively $S_X, S_Y, S_Z$) of particle number $I$ is constrained.

IC=10.1: Difference between final and initial $J$-spin coordinate of particle $I$ (convenient e.g. for $\vec{n}_0$ spin vector search).

OBJECT DEFINITION

Depending on the type of constraint (see table p. 61), 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 [28]. 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. [29].

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.
4.3 Declaring Options

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.
DESCRIPTION OF THE AVAILABLE PROCEDURES

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 \textit{FIT[2]} procedure as displayed in zgoubi.res, normally under the form

\begin{verbatim}
STATUS OF VARIABLES (Iteration # 95)

<table>
<thead>
<tr>
<th>LMNT</th>
<th>VAR</th>
<th>PARAM</th>
<th>MINIMUM</th>
<th>INITIAL</th>
<th>FINAL</th>
<th>MAXIMUM</th>
<th>STEP</th>
<th>NAME</th>
<th>LBL1</th>
<th>LBL2</th>
</tr>
</thead>
<tbody>
<tr>
<td>145</td>
<td>1</td>
<td>4</td>
<td>-3.000E+03</td>
<td>762.</td>
<td>761.9484791</td>
<td>3.000E+03</td>
<td>1.254E-05</td>
<td>MULTIPOL</td>
<td>HKIC</td>
<td>DHCB02</td>
</tr>
<tr>
<td>192</td>
<td>2</td>
<td>4</td>
<td>-1.000E+03</td>
<td>-231.</td>
<td>-230.9846875</td>
<td>1.000E+03</td>
<td>4.182E-06</td>
<td>MULTIPOL</td>
<td>HKIC</td>
<td>DHCB08</td>
</tr>
<tr>
<td>146</td>
<td>3</td>
<td>4</td>
<td>-1.000E+03</td>
<td>-320.</td>
<td>-319.8554171</td>
<td>1.000E+03</td>
<td>4.182E-06</td>
<td>MULTIPOL</td>
<td>VKIC</td>
<td>DVCB02</td>
</tr>
<tr>
<td>183</td>
<td>4</td>
<td>4</td>
<td>-1.000E+03</td>
<td>528.</td>
<td>527.7249064</td>
<td>1.000E+03</td>
<td>4.182E-06</td>
<td>MULTIPOL</td>
<td>VKIC</td>
<td>DVCB08</td>
</tr>
<tr>
<td>615</td>
<td>5</td>
<td>4</td>
<td>-3.000E+03</td>
<td>308.</td>
<td>307.6860565</td>
<td>3.000E+03</td>
<td>1.254E-05</td>
<td>MULTIPOL</td>
<td>HKIC</td>
<td>DHCFO2</td>
</tr>
<tr>
<td>651</td>
<td>6</td>
<td>4</td>
<td>-1.000E+03</td>
<td>-114.</td>
<td>-113.8490362</td>
<td>1.000E+03</td>
<td>4.182E-06</td>
<td>MULTIPOL</td>
<td>HKIC</td>
<td>DHCFO8</td>
</tr>
<tr>
<td>616</td>
<td>7</td>
<td>4</td>
<td>-1.000E+03</td>
<td>-78.9</td>
<td>-78.88730937</td>
<td>1.000E+03</td>
<td>4.182E-06</td>
<td>MULTIPOL</td>
<td>VKIC</td>
<td>DVCF02</td>
</tr>
<tr>
<td>652</td>
<td>8</td>
<td>4</td>
<td>-1.000E+03</td>
<td>212.</td>
<td>211.8799183</td>
<td>1.000E+03</td>
<td>4.182E-06</td>
<td>MULTIPOL</td>
<td>VKIC</td>
<td>DVCF08</td>
</tr>
</tbody>
</table>

# STATUS OF CONSTRAINTS
# Type I J LMNT# Desired Weight Reached K12 * 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 1899 times
# K12 = 1.68006E-16 Busy...
\end{verbatim}

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 \textit{FIT[2]} variables may belong in [MC]OBJET, \textit{GETFITVAL} may appear right before [MC]OBJET in zgoubi.dat, to allow for its updating.
4.3 Declaring Options

MCDESINT : Monte-Carlo simulation of in-flight decay[31]

As soon as MCDESINT appears 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 \( \tau_1 \). 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 \( D_1 = \frac{p_1}{q} \text{BORO} \) with BORO = reference rigidity, defined in [MC]OBJET), and position \( Y_1, Z_1 \) in the zgoubi frame. 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 decay length \( s_1 \) of particle 1 is related to its center of mass lifetime \( \tau_1 \) by

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

The path length \( s \) up to the decay point is then calculated from a random number \( 0 < R_1 \leq 1 \) by using the exponential decay formula

\[
s = -s_1 \ln R_1
\]

After decay, 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 \( R_2 \) and \( R_3 \). \( \phi \) is a relativistic invariant, and \( \theta \) in the laboratory frame (Fig. 8) is given by

\[
\tan \theta = \frac{1}{\gamma_1} \frac{\sin \theta^*}{\beta_1^* + \cos \theta^*}
\]

\( \beta_2^* \) and momentum \( p_2 \) are given by

\[
\gamma_2^* = \frac{M_2^2 + M_2^2 - M_3^2}{2M_1 M_2} \\
\beta_2^* = \left(1 - \frac{1}{\gamma_2^2}\right)^{1/2} \\
\gamma_2 = \gamma_1 \gamma_2^* \left(1 + \beta_1 \beta_2^* \cos \theta^*\right) \\
p_2 = M_2 \sqrt{\gamma_2^2 - 1}
\]

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 \( D_2 = \frac{p_2}{q} \text{BORO} \) (where BORO is the reference rigidity, see OBJET), while the starting position of \( M_2 \) is \( (Y_1, Z_1, s_1) \).

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 \( F(6, I) - \frac{\text{XPAS}}{2} \), 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; \texttt{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 \texttt{zgoubi} frame.

In \textit{ESL} and \textit{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 (\( I = 1, IMAX \)) is stored in the array \( FDES(6, I) \). For further statistical purposes (\textit{e.g.}, use of \texttt{HISTO}) the daughter particle 2 is tagged with an ‘\( S \)’ standing for “secondary”. When a particle decays, its coordinates \( D, Y, T, Z, P, s \), \textit{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 \texttt{MCDESINT} is allowed and correct. For instance, negative drifts may occur in a structure for some of the particles when using \textit{CHANGREF} (due to the \( Z \)-axis rotation or a negative \( XCE \)), or when using \texttt{DRIFT} with \( XL < 0 \). Provision has been made to take it into account during the \texttt{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 \texttt{MCDESINT}. 
OPTICS : Write out optical functions

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 $KOBJ=5.01$.

Note: The initial values in OBJET[$KOBJ=5.01$] may be the periodic ones, as obtained, for instance, from a first run using MATRIX[$IFOC=11$].

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 4.6.5, page 162, regarding the labelling of keywords).

A third argument, $IMP=1$, will cause saving of the transported beta functions into file zgoubi.OPTICS.out.
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 \( \vec{u} \),

\[
\vec{R}_1 \approx \vec{R}_0 + \vec{u} \Delta s + \ldots + \vec{u}^{(5)} \frac{\Delta s^6}{6!}
\]

\[
\vec{u}_1 \approx \vec{u} + \vec{u}' \Delta s + \ldots + \vec{u}^{(5)} \frac{\Delta s^5}{5!}
\]

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 \textit{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 FORTRAN file block.f.

Note the following:
As concerns the optical elements

\textit{DECAPOLE, DODECAPO, EBMULT, ELMULT, MULTIPOL, OCTUPOLE, QUADRUPO, SEXTUPOL}

field derivatives (see eq. 1.2.10 p. 19, eq. 1.2.15 p. 21,) 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 \textit{ORDRE}, so to avoid pushing Taylor expansions as high as \( \vec{u}^{(5)} \).

\textbf{NOTE} : see also the option \textit{IORDRE} in field map declarations (\textit{DIPOLE-M, TOSCA}, etc.).
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:

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAVITE</td>
<td>mass, charge</td>
</tr>
<tr>
<td>MCDESINT</td>
<td>mass, COM life-time</td>
</tr>
<tr>
<td>SPNTRK</td>
<td>mass, gyromagnetic factor</td>
</tr>
<tr>
<td>SRLOSS</td>
<td>mass, charge</td>
</tr>
<tr>
<td>SYNRAD</td>
<td>mass, charge</td>
</tr>
<tr>
<td>Electric and Electro-Magnetic elements</td>
<td>mass, charge</td>
</tr>
</tbody>
</table>

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.
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 (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 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 \((1 + NPASS) \times IMAX\)
  trajectories, with as many random initial coordinates.

• REBELOTE can be used for multi-turn ray-tracing in circular machines circular machines : normally
  \( K = 99 \) in that case. 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 to 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.

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 KWRT= 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.
**RESET : Reset counters and flags**

Piling up problems in *zgoubi* input data file is allowed, with normally no particular precaution, except that each new problem must begin with a new object definition (using *MCOBJET, OBJET*). Nevertheless, when calling upon certain keywords, then, flags, counters or other integrating procedures may be involved. It may therefore be necessary to reset them. This is the purpose of *RESET* which normally appears right before the object definition and causes each problem to be treated as a new and independent one.

The keywords or procedures of concern and the effect of *RESET* are the following:

- **CHAMBR** : number of stopped particles reset to 0; *CHAMBR* option switched off
- **COLLIMA** : number of stopped particles reset to 0
- **HISTO** : histograms are emptied
- **INTEG** : number of particles out of field map boundaries reset to 0
- **MCDESINT** : decay in flight option switched off, counter reset
- **PICKUPS** : pick-up signal calculation switched off
- **SCALING** : scaling functions disabled
- **SPNTRK** : spin tracking option switched off
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, RF parameters in CAVITE. It is normally intended to be declared right after the object definition, and used in conjunction with REBELOTE, for the simulation of multi-turn tracking - possibly including acceleration cycles.

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 varied will result in the same timing law being applied to all MULTIPOL’s in the zgoubi optical structure data file. Subsets can be selected by labeling keywords in the data file (section 4.6.5, page 162) and adding the corresponding LABEL(s) in the SCALING declarations (two LABEL’s maximum). 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, NFMAX, to the number NF of families that can be declared as subject to a scaling law, NFMAX can be changed in the FORTRAN include file MXFS.H. A scaling law can be comprised of up to NT successive timings, between two successive timings, a linear interpolation law is used to determine the scaling factor.

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

```
SCALING - Scaling
1 4 Active. NF = 4 families of elements are concerned, as listed below
QUADRUPO QFA QFB
  2 - Quadrupoles labeled ‘QFA’ and Quadrupoles labeled ‘QFB’
  18131.E-3 24176.E-3 The field increases (linearly) from 18131E-3 to 24176E-3
     1 6379 from turn 1 to turn 6379
MULTIPOL QDA QDB
  2 - Multipoles labeled ‘QDA’ and Multipoles labeled ‘QDB’
  18131.E-3 24176.E-3 Fields increase from 18131E-3 to 24176E-3
     1 6379 (∀ i = 1, 10 poles) from turn 1 to turn 6379
BEND
  2 - All BEND’s (regardless of any LABEL)
  18131.E-3 24176.E-3 As above
     1 6379
CAVITE - Accelerating cavity
  3
  1 1.22 1.33352 The synchronous rigidity (Bρ) increases,
     1 1200 6379 from (Bρ)s to 1.22 *(Bρ)s, from turn 1 to 1200, and
        1 6379 from 1.22 *(Bρ)s to 1.33352 *(Bρ)s 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 Bi are updated in the following way. Let SCALE(TIMING = N) be the updating scale factor

\[
\text{SCALE}(N) = 18.131 \frac{24.176 - 18.131}{1 + 6379 - 1} (N - 1)
\]

and then

\[
B(N) = \text{SCALE}(N) B_0 \\
B_i(N) = \text{SCALE}(N) B_{i0}
\]

The RF frequency is computed using

\[
f_{RF} = \frac{\hbar c}{2 \pi} \sqrt{\frac{q(Bp)_s}{(q^2(Bp)_s^2 + (Mc^2)^2)^{1/2}}}
\]

where the rigidity is updated in the following way. Let (Bp)_so be the initial rigidity (namely, (Bp)_so = BORO as defined in the keyword OBJET for instance). Then, at turn number N,
if $1 \leq N \leq 1200$ then, $SCALE(N) = 1 + \frac{1.22 - 1}{1 + 1200 - 1} (N - 1)$

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

and then,

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

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,

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

- $NT = -2$ : 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:

  MULTIPOL COH1
  1.10
  ./Csnk3D/bump_centered.scal
  1 2

  File name
  Column numbers in the file : col. 2 gives the scaling factor at rigidity given by col. 1.

Notes:

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

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

2. It may happen that some optical elements won’t scale, for source code development reasons. This should be paid attention to.
**SPNTRK : Spin tracking**

The keyword *SPNTRK* permits switching on the spin tracking option. It also permits the attribution of an initial spin component 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 = 5**: random generation of *IMAX* initial spin conditions as described in Fig. 9. Given a mean polarization axis (S) defined by its angles $T_0$ and $P_0$, and a cone of angle $A$ with respect to this axis, the *IMAX* spins are sorted randomly in a Gaussian distribution

$$p(a) = \exp \left[ -\frac{(A - a)^2}{2 \delta A^2} \right] / \delta A \sqrt{2\pi}$$

and within a cylindrical uniform distribution around the (S) axis. Examples of simple distributions available by this mean are given in Fig. 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 (S) defined by $T_0$ and $P_0$.](image-url)
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 $\delta A = 0$. 

$\delta A$
SRLOSS : Synchrotron radiation loss [17]

The keyword \texttt{SRLOSS} allows activating or stopping (option \texttt{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.

\texttt{SRLOSS} must be preceded by \texttt{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 \texttt{SRPRNT}.
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 :

\[
\begin{align*}
(ELx, ELy, ELz) & : \text{electric field vector } \vec{E} \text{ (eq. 3.2.1)} \\
(btx, bty, btz) & = \frac{1}{c} \times \text{ particle velocity} \\
(gx, gy, gz) & = \frac{d}{dt} \vec{\beta} = \text{ particle acceleration (eq. 3.2.3)} \\
\Delta \tau & = \text{ observer time increment (eq. 3.2.2)} \\
t' & = \tau - \tau(t')/c = \text{ retarded (particle) time} \\
(rt x, rty, rtz) & : \vec{R}(t), \text{ particle to observer vector (eq. 3.2.4)} \\
(x, y, z) & = \text{ particle coordinates} \\
\Delta s & = \text{ step size in the magnet (fig. 2)} \\
NS & = \text{ step number} \\
I & = \text{ particle number} \\
LET(I) & = \text{ tagging letter} \\
IEX(I) & = \text{ stop flag (see section 4.6.10)}
\end{align*}
\]

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.
4.4 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 25), 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, $B_{\rho_{\text{ref}}}$, and follows the latter so to preserve $\rho = B_{\rho_{\text{ref}}}/B_0$ and the orbit deviation $L/\rho$. In particular,
  - in the absence of acceleration, $B_{\rho_{\text{ref}}} \equiv \text{BORO}$, with \text{BORO} the quantity appearing in the object definition using [MC]OBJET,
  - in presence of acceleration using CAVITE, $B_{\rho_{\text{ref}}}$ is changed to $\text{BORO} \times D_{\text{ref}}$ at each passage in the cavity, with $D_{\text{ref}}$ the relative synchronous momentum increase, a quantity that zgoubi updates at cavity traversal.
- The field indices, quadrupole $K_1$ and sextupole $K_2$, are derived from the reference rigidity, $B_{\rho_{\text{ref}}}$, via momentum-dependent polynomials, taken from Ref. [32].
- 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 25. 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 multipole data in AGSQUAD, compared to MULTIPOL.

The field in AGSQUAD is computed using transfer functions from the ampere-turns in the coils to magnetic field that account for the non-linearity of the magnetic permeability [33].
AIMANT : Generation of dipole mid-plane 2-D map, polar frame

The keyword \textit{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 \textit{DIPOLE-M}. In addition, a similar modelling, that however skips the stage of an intermediate mid-plane field map, can be found in \textit{DIPOLE[S]}. The extent of the map is defined by the following parameters, as shown in Figs. 11A and 11B,

- $\textit{AT}$ : total angular aperture
- $\textit{RM}$ : mean radius used for the positioning of field boundaries
- $\textit{RMIN, RMAX}$ : 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,

- $\textit{ACENT}$ : arbitrary angle, used for the positioning of the EFB's.
- $\omega$ : azimuth of an EFB with respect to $\textit{ACENT}$
- $\theta$ : 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

\[
B_Z = F(R, \theta) * B_0 * \left( 1 + N * \frac{R - RM}{RM} + B * \left( \frac{R - RM}{RM} \right)^2 + G * \left( \frac{R - RM}{RM} \right)^3 \right) \tag{4.4.1}
\]

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

\textbf{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 \( \xi \).

\textbf{If} \( \xi \geq 0 \), \( F \) is a second order type fringe field (Fig. 12) given by

\[
F = \frac{1}{2} \left( \frac{\lambda - s}{\lambda^2 - \xi^2} \right)^2 \quad \text{if} \quad 0 \leq s \leq \xi \tag{4.4.2}
\]

\[
F = 1 - \frac{1}{2} \left( \frac{\lambda - s}{\lambda^2 - \xi^2} \right)^2 \quad \text{if} \quad -\lambda \leq s \leq -\xi \tag{4.4.3}
\]

where \( s \) is the distance to the EFB, and

\[
F = \frac{1}{2} + \frac{s}{\lambda + \xi} \quad \text{if} \quad 0 \leq s \leq \xi \tag{4.4.4}
\]

\[
F = \frac{1}{2} - \frac{s}{\lambda + \xi} \quad \text{if} \quad -\xi \leq s \leq 0 \tag{4.4.5}
\]

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.
Figure 11: A: Parameters used to define the field map and geometrical boundaries.
B: Parameters used to define the field map and fringe fields.
If \( \xi = -1 \), \( F \) is an exponential type fringe field (Fig. 12) given by [34]

\[
F = \frac{1}{1 + \exp P(s)} \quad (4.4.6)
\]

where \( s \) is the distance to the EFB, and

\[
P(s) = C_0 + C_1 \left( \frac{s}{\lambda} \right) + C_2 \left( \frac{s}{\lambda} \right)^2 + C_3 \left( \frac{s}{\lambda} \right)^3 + C_4 \left( \frac{s}{\lambda} \right)^4 + C_5 \left( \frac{s}{\lambda} \right)^5 \quad (4.4.7)
\]

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 \( \bar{B} \) 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 \( \text{shift} \). \( s \) 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. 4.4.1) is (Fig. 13)

\[
F(R, \theta) = F_E * F_S * F_L
\]

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

\[
\delta R = \frac{R_{\text{MAX}} - R_{\text{MIN}}}{I_{\text{MAX}} - 1}
\]

and the angular step by

\[
\delta \theta = \frac{A T}{I_{\text{MAX}} - 1}
\]

where, \( R_{\text{MIN}} \) and \( R_{\text{MAX}} \) are the lower and upper radial limits of the field map, and \( A T \) is its total angular aperture (Fig. 11B). \( I_{\text{MAX}} \) and \( I_{\text{AMAX}} \) 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

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

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

\[
B_Z * \left( 1 + \frac{\Delta B_Z}{B_0} \frac{\theta - \theta_0}{A T} \right)
\]
Figure 12: Second order type fringe field (upper plot) and exponential type fringe field (lower plot).

Figure 13: Effective value of $F(R, \theta)$ 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) [35]

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

\[
B_Z \ast \left( 1 + F\theta \ast FR \ast \frac{\Delta B_Z}{B_0} \right)
\]

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

**Extrapolation Off Median Plane**

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

![Figure 14: A second order profile shim. The shim is centered at \( \frac{R_1 + R_2}{2} \) and \( \frac{\theta_1 + \theta_2}{2} \).](image)
AUTOREF : Automatic transformation to a new reference frame

AUTOREF positions the new reference frame following 3 options:

If $I = 1$, AUTOREF is equivalent to

$$\text{CHANGREF}[XCE = 0, YCE = Y(1), ALE = T(1)]$$

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

$$\text{CHANGREF}[XW, YW, T(1)]$$

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

$$\text{CHANGREF}[XW, YW, T(I1)]$$

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.
**BEAMBEAM : Beam-beam lens**

*BEAMBEAM* is a beam-beam lens simulation, a point transform [36].

Upon option using *SPNTRK*, *BEAMBEAM* will include spin kicks, after modelling as described in Ref. [37].
**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 *DIPOLES* 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 machine cell for instance, see section 4.6.7, p. 163).

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 *θ* satisfies
\[
XL = 2 \frac{B1 \cdot BORO}{B1} \sin θ/2.
\]

Then follows the description of the entrance and exit EFB’s 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 ±*X_E* (±*X_S*) on both sides of the entrance (exit) EFB, the fringe field model is used (same as for *QUADRUPO*, Fig. 35, p. 133) ; elsewhere, the field is supposed to be uniform.

If *λ_E* (resp. *λ_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 *B1* is non-zero) is simulated at magnet entrance and exit by a kick
\[
P_2 = P_1 - Z_1 \tan(ε/ρ)
\]
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, *ρ* the local horizontal bending radius and *ε* the wedge angle experienced by the particle ; *ε* 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 4.6.7).

![Figure 15: Geometry and parameters of BEND: XL = length, θ = deviation, WE, WS are the entrance and exit wedge angles. The motion is computed in the Cartesian frame (O, X, Y, Z)](image-url)
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).

```fortran
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

```fortran
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 4.6.10, p. 166), 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 can be defined with the flag \( ID \), as follows (Fig. 16).

**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 16: 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 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 4.6.9). It 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 of a problem, as used when computing optical strengths from field values, sections 1.2.1-1.2.3, is specified in the object definition by \([MC]OBJET\). However, in many cases – options as described below – that reference rigidity will be updated upon crossing the cavity, by the amount of the synchronous rigidity increase as induced by the cavity, namely,

\[
B_{\rho_{ref}} = BORO \rightarrow B_{\rho_{ref}} = BORO + \delta B_{\rho_s}
\]

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 4.6.9. CAVITE may conveniently be located at the end of the optical structure, otherwise its phasing has to be indicated. The synchrotron motion of any of the \( IMAX \) particles of a beam is obtained from the following mapping

\[
\begin{align*}
\phi_2 - \phi_1 &= 2\pi f_{RF} \left( \frac{\ell}{\beta_c} - \frac{L}{\beta_{sc}} \right) \\
W_2 - W_1 &= q\hat{V} \sin \phi_1
\end{align*}
\]

where
- \( \phi \) = RF phase; \( \phi_2 - \phi_1 \) = variation of \( \phi \) between two traversals
- \( W \) = kinetic energy; \( W_2 - W_1 \) = energy gain at a traversal of CAVITE
- \( L \) = length of the synchronous closed orbit (to be calculated by prior ray-tracing, see the bottom NOTE)
- \( \ell \) = orbit length of the particle between two traversals
- \( \beta_c \) = velocity of the (virtual) synchronous particle
- \( \beta_{sc} \) = velocity of the particle
- \( \hat{V} \) = peak RF voltage
- \( q \) = particle electric charge.

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

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

where
- \( h \) = harmonic number of the R.F
- \( M \) = mass of the particle
- \( c \) = velocity of light.

The current rigidity \((B\rho)_s\) of the synchronous particle 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 \( \beta_s \) 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

\[ \phi_1 = \phi_s = \text{synchronous phase} \]

\[ (B\rho)_{1s} = \text{BORO} \]

- For any of the \( I = 1, \text{IMAX} \) particles of the beam

\[ \phi_{1I} = \phi_s = \text{synchronous phase} \]

\[ (B\rho)_{1I} = \text{BORO} \ast D_I \]

where the quantities \text{BORO} and \( D_I \) are given in the object description.

**Calculation of the Coordinates**

Let \( p_I = \left[ p_{XI}^2 + p_{YI}^2 + p_{ZI}^2 \right]^{1/2} \) be the momentum of particle \( I \) at the exit of the cavity, while \( p_{I0} = \left[ p_{X0}^2 + p_{Y0}^2 + p_{Z0}^2 \right]^{1/2} \) 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

\[
p_{XI} = \left[ p_I^2 - (p_{I0}^2 - p_{X0}^2) \right]^{1/2}
\]

\[
p_{YI} = p_{YI0}, \quad \text{and} \quad p_{ZI} = p_{ZI0} \quad \text{(longitudinal kick)}
\]

\[
X_I = X_{I0}, \quad Y_I = Y_{I0} \quad \text{and} \quad Z_I = Z_{I0} \quad \text{(zero length cavity)}
\]

and for the angles (see Fig. 1)

\[
T_I = \text{Atg} \left( \frac{p_{YI}}{p_{XI}} \right)
\]

\[
P_I = \text{Atg} \left( \frac{p_{ZI}}{\left(p_{XI} + p_{YI}^2\right)^{1/2}} \right) \quad \text{(damping of the transverse motion)}
\]

**If IOPT = 2 :** the same simulation of a synchrotron RF cavity as for \( IOPT = 1 \) is performed, except that the keyword \text{SCALING} \text{ (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 given in the input data. 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.

**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.

If \( \text{IOPT} = 6 \): allows reading the RF frequency and/or phase law from an external file (with name normally “zgoubi.freqLaw.In”). See routines `cavite.f` and `scalin.f` for details. Was first used for acceleration in scaling FFAG [45].

If \( \text{IOPT} = 7 \): fixed frequency RF, quasi- or isochronous acceleration. Was first used for quasi-isochronous, fixed frequency acceleration in the EMMA prototype linear FFAG [46]. Can be used for cyclotron acceleration.

**NOTE. Calculation of the closed orbit:**

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 $L$ (the $F(6, 1)$ coordinate in the FORTRAN).
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

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

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

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 $-XE$ to 0 and from $XL$ to $XL + XS$, 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. 27, and 11, p. 82). 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 4.6.10) 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

\[
T_2 = T_1 - ALE \\
Y_2 = \frac{(Y_1 - YCE) \cos T_1 + XCE \sin T_1}{\cos T_2} \\
DL^2 = (XCE - Y_2 \sin ALE)^2 + (YCE - Y_1 + Y_2 \cos ALE)^2 \\
Z_2 = Z_1 + DL \cot P_1 \\
S_2 = S_1 + DL \cos P_1 \\
P_2 = P_1
\]

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" 

| OBJECT | Electron, Ekin=15MeV. |
| 1 | 1 |
| 2 | 2.00 0.00 0.00 1.00 1.00 1.00 1.00 1.00 1.00 |
| MARKER | BEG .plt -> list into zgoubi.plt. |
| DRIFT | 10 cm drift. |
| CHANGREF | 0.0 -6.34165 First : half Z-rotation. |
| CHANGREF | 0.0 1.0 Next : Y-shift. |
| MULTIPOL | Combined function multipole, dipole + quadrupole. |
| 5 10.0 2.064995867082342 2.00 0.00 0.00 0.00 |
| 5 0.0 1.0 1.00 1.00 1.00 1.00 1.00 1.00 1.00 |
| 4 1.0555 2.00 1.0105 1.0105 1.0105 1.0105 |
| 4 1.0555 2.00 1.0105 1.0105 1.0105 1.0105 |
| 4 1.0555 2.00 1.0105 1.0105 1.0105 1.0105 |
| 0 0 0 0 0 0 0 0 0 0 |
| DRIFT | 10 cm drift. |
| MARKER | END .plt .plt" => list into zgoubi.plt. |

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

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

Using CHANGREF, "New Style"

'OBJET'
51.71103865921708 Electron, Ekin=15MeV.
2
1 1 One particle, with
2 0. 0.0 0.0 0.0 1. 'R' Y_0=2 cm, other coordinates zero.
1 1 1 1 1 1
'MARKER' BEG .plt -> list into zgoubi.plt.
'DRIFT' 10 cm drift.
10.

'CHANGREF'
ZR -6.34165 YS 1. First half Z-rotate, Next Y-shift.
'MULTIPOL' Combined function multipole, dipole + quadrupole.
2 -> list into zgoubi.plt.
5 10. 2.064995867082342 2. 0. 0. 0. 0. 0. 0. 0. 0.
0 0 5. 1.1 1.00 1.00 1.00 1.00 1. 1. 1. 1.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.
0 0 5. 1.1 1.00 1.00 1.00 1.00 1.00 1. 1. 1. 1.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.
0 0 0 0 0 0 0 0 0 0 .1 step size
1 0. 0. 0.

'CHANGREF'
YS -1. ZR -6.34165 First Y-shift back, next half Z-rotate.
'DRIFT' 10 cm drift.
10.

'FAISCEAU'

'END'
CIBLE or TARGET: Generate a secondary beam following target interaction

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

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

The geometry of the interaction is shown in Fig. 18.

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

The position of \( B \) downstream is deduced from that of \( A \) upstream by a transformation equivalent to two transformations using \( \text{CHANGREF} \), namely

\[
\text{CHANGREF}(XCE = YCE = 0, \ ALE = \beta)
\]

followed by

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

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

\[
M_1 c^2 + W_2 = W_3 + W_4
\]

\[
p_4^2 = p_2^2 + p_3^2 - 2p_2p_3 \cos(\theta - T)
\]

---

**Figure 18:** 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
4.4 Optical Elements and Related Numerical Procedures

**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 \(Y_C, Z_C\) and has transverse dimensions \(\pm Y_L\) and \(\pm Z_L\) such that any particle will be stopped if its coordinates \(Y, Z\) satisfy

\[
(Y - Y_C)^2 \geq Y_L^2 \quad \text{or} \quad (Z - Z_C)^2 \geq Z_L^2 \quad \text{if} \quad IFORM = 1
\]

\[
\frac{(Y - Y_C)^2}{Y_L^2} + \frac{(Z - Z_C)^2}{Z_L^2} \geq 1 \quad \text{if} \quad IFORM = 2
\]

**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

\[
(h \leq h_{\text{min}} \text{ or } h \geq h_{\text{max}}) \quad \text{or} \quad (v \leq v_{\text{min}} \text{ or } v \geq v_{\text{max}})
\]

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

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

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

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 4.6.10) 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. 19)

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 $Z$

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

with $G_0 = \frac{B_0}{R_d}$

The modelling of the fringe field form factor $G(X)$ is described under QUADRUPO, p. 132.

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 19: Decapole magnet
**DIPOLE**: Dipole magnet, polar frame

*DIPOLE* provides a model of a dipole field, possibly with 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* has a version, *DIPOLES*, that allows overlapping of fringe fields in a configuration of neighboring magnets.

The dimensioning of the magnet is defined by (Fig. 11, p. 82)

\[\begin{align*}
AT & : \text{total angular aperture} \\
RM & : \text{mean radius used for the positioning of field boundaries}
\end{align*}\]

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.

\[\begin{align*}
ACENT & : \text{arbitrary inner angle, used for EFB’s positioning} \\
\omega & : \text{azimuth of an EFB with respect to } ACENT \\
\theta & : \text{angle of an EFB with respect to its azimuth (wedge angle)} \\
R_1, R_2 & : \text{radius of curvature of an EFB} \\
U_1, U_2 & : \text{extent of the linear part of an EFB.}
\end{align*}\]

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

\[
B_Z(R, \theta) = F(R, \theta) \cdot B_0 \cdot \left(1 + N \left(\frac{R - RM}{RM}\right) + B \left(\frac{R - RM}{RM}\right)^2 + G \left(\frac{R - RM}{RM}\right)^3\right) \tag{4.4.8}
\]

where \(N, B\) and \(G\) are respectively the first, second and third order field indices and \(F(R, \theta)\) 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. 84) given by [34]

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

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 \(\text{shift}\). \(s\) 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. 4.4.8) is
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. 4.4.8 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. 20. 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.

![Interpolation method](image)

**Figure 20:** 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.
4.4 Optical Elements and Related Numerical Procedures

**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 EFB’s positioning
- **ω**: azimuth of an EFB with respect to **ACENT**
- **θ**: angle of an EFB with respect to its azimuth (wedge angle)
- **R₁, R₂**: radius of curvature of an EFB
- **U₁, U₂**: extent of the linear part of an EFB.

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

\[
B_Z(R, \theta) = F(R, \theta) \ast B_0 \ast \left(1 + N \ast \left(\frac{R - RM}{RM}\right) + B \ast \left(\frac{R - RM}{RM}\right)^2 + G \ast \left(\frac{R - RM}{RM}\right)^3\right) (4.4.9)
\]

where \(N, B, G\) are respectively the first, second and third order field indices and \(F\) 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 [34]

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

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 \(s - shift\) 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. 4.4.9) is

\[
F(R, \theta) = F_E \ast F_S \ast F_L
\]

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
\[ \delta R = \frac{R_{\text{MAX}} - R_{\text{MIN}}}{I_{\text{MAX}} - 1} \]

and the angular step by

\[ \delta \theta = \frac{A_T}{I_{\text{AMAX}} - 1} \]

where \( R_{\text{MIN}} \) and \( R_{\text{MAX}} \) are the lower and upper radial limits of the field map, and \( A_T \) is its total angular aperture (Fig. 11B). \( I_{\text{MAX}} \) and \( I_{\text{AMAX}} \) 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 \( N_{\text{BS}} \), so as to simulate field defects or shims.

If \( N_{\text{BS}} = -2 \), 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

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

If \( N_{\text{BS}} = -1 \), the perturbation is proportional to \( \theta - \theta_0 \), and \( B_Z \) is replaced by

\[ B_Z * \left( 1 + \frac{\Delta B_Z}{B_0} \frac{\theta - \theta_0}{A_T} \right) \]

If \( N_{\text{BS}} \geq 1 \), then \( N_{\text{BS}} \) shims are introduced at positions \( \frac{R_1 + R_2}{2}, \frac{\theta_1 + \theta_2}{2} \) (Fig. 14) [35]

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

\[ B_Z * \left( 1 + F_{\theta} * F_R * \frac{\Delta B_Z}{B_0} \right) \]

where \( F_{\theta} = 0 \) or \( F_R = 0 \) outside the shim, and \( F_{\theta} = 1 \) and \( F_R = 1 \) 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 \( I_{\text{ORDRE}} (I_{\text{ORDRE}}=2, 25 \text{ or } 4, \text{ see section 1.4.2}) \). 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.
**4.4 Optical Elements and Related Numerical Procedures**

**DIPOLES**: Dipole magnet \( N \)-tuple, polar frame \([38, 39]\)

DIPOLES 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\(^5\).

The dimensioning of the magnet is defined by

\[
\begin{align*}
AT & : \text{total angular aperture} \\
RM & : \text{mean radius used for the positioning of field boundaries}
\end{align*}
\]

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. 4.4.10, 4.4.11) 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. 82, and Fig. 21)

\[
\begin{align*}
ACNi & : \text{arbitrary inner angle, used for EFB's positioning} \\
\omega & : \text{azimuth of an EFB with respect to } ACN \\
\theta & : \text{angle of an EFB with respect to its azimuth (wedge angle)} \\
R_1, R_2 & : \text{radius of curvature of an EFB} \\
U_1, U_2 & : \text{extent of the linear part of an EFB}
\end{align*}
\]

**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 either form, upon option,

\[
\begin{align*}
(i) \quad B_{Zi}(R, \theta) &= B_{Z0,i} \mathcal{F}_i(R, \theta) \left(1 + b_1(R - RM_i)/RM_i + b_2(R - RM_i)^2/RM_i^2 + \ldots\right) (4.4.10) \\
(ii) \quad B_Z(R, \theta) &= B_{Z0,i} + \sum_{i=1}^{N} \mathcal{F}_i(R, \theta) \left(b_1(R - RM_i) + b_2(R - RM_i)^2 + \ldots\right) (4.4.11)
\end{align*}
\]

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 channeled dipoles, cyclotron or FFAG magnets, etc.

![Figure 21: Definition of a dipole triplet using the DIPOLES or FFAG procedures.](image)

**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.

\(^5\)FFAG can be referred to as another instance of a procedure based on such method.
$F$ is an exponential type fringe field (Fig. 12, page 84) given by [34]

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

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

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

In addition, $g$ is made dependent of $R$ (a way to simulate the effect of variable gap size on fringe field extent), under the form

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

This dependence is accounted for rigorously if the interpolation method (see below) is used, or else to order zero (derivatives of $g(R)$ are not considered) if the analytic method (below) is used.

Let $F_E$ (respectively $F_S$) be the fringe field 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

$$F_i(R, \theta) = F_E * F_S$$  \hspace{1cm} (4.4.12)

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

$$B_Z(R, \theta) = \sum_{i=1,N} B_{Z_i}(R, \theta)$$  \hspace{1cm} (4.4.13)

$$\frac{\partial^{k+l}B_Z(R, \theta)}{\partial \theta^k \partial r^l} = \sum_{i=1,N} \frac{\partial^{k+l}B_{Z_i}(R, \theta)}{\partial \theta^k \partial r^l}$$  \hspace{1cm} (4.4.14)

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. 4.4.13), 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 [40] :

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
4.4 Optical Elements and Related Numerical Procedures

\[
x(R, \theta) = \cos(ACN - \theta) - RM
\]
\[
y(R, \theta) = R \sin(ACN - \theta)
\]
\[
x_P(R, \theta) = \sin(u) \left( y(R, \theta) - y_b \right)/2 + x_b \sin^2(u) + x(R, \theta) \cos^2(u)
\]
\[
y_P(R, \theta) = \sin(u) \left( x(R, \theta) - x_b \right)/2 + y_b \cos^2(u) + y(R, \theta) \sin^2(u)
\]

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

\[
\frac{\partial u + v x}{\partial \theta} \frac{\partial u + v y}{\partial r}, \frac{\partial u + v x_0}{\partial \theta} \frac{\partial u + v y_0}{\partial r}, \frac{\partial u + v d}{\partial \theta} \frac{\partial u + v p}{\partial r}.
\]

Interpolation method:

The expression \(B_z(R, \theta)\) in Eq. 4.4.13 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. 22. 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 22: 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. 23)

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 $Z$

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

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

The modelling of the fringe field form factor $G(X)$ is described under QUADRUPO, p. 132.

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

$$B_X = 0$$
$$B_Y = G_0(5Y^4 - 10Y^2 Z^2 + Z^4) Z$$
$$B_Z = G_0(Y^4 - 10Y^2 Z^2 + 5Z^4) Y$$

Figure 23: 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. The associated equations of motion are (Fig. 24)

\[
\begin{align*}
Y_2 &= Y_1 + XL \cdot \tan T \\
Z_2 &= Z_1 + \frac{XL}{\cos T} \cdot \tan P \\
SAR_2 &= SAR_1 + \frac{XL}{\cos T \cdot \cos P}
\end{align*}
\]

![Figure 24: Transfer of particles in a drift space.](image)
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 rotation $\frac{\pi}{2n}$ rotation ($n = 1, 2, 3, \ldots$) (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 25: An example of $\vec{E}$, $\vec{B}$ multipole: the achromatic quadrupole (known for its allowing null second order chromatic aberrations [41]).
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 $X_1$ and $V_1$ ($X_2$ and $V_2$). The distance between the two electrodes is $D$, and their inner radius is $R_0$ (Fig. 26). The model for the electrostatic potential along the axis is [42]

$$V(X) = \frac{V_2 - V_1}{2} \frac{\omega x}{R_0} \left[ \frac{V_1 + V_2}{2} \right]$$

if $D = 0$

$$V(X) = \frac{V_2 - V_1}{2} \frac{1}{2\omega D/R_0} \left[ \frac{V_1 + V_2}{2} \right]$$

if $D \neq 0$

($x =$ distance from half-way between the electrodes ; $\omega = 1.318$ ; th = hyperbolic tangent ; 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 $\frac{V_1 + V_2}{2}$ which disappears when differentiating).

Use PARTICUL prior to EL2TUB, for the definition of particle mass and charge.

Figure 26: 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 \( L_1, L_2, \ldots, L_N \). \( D \) is the mirror/lens gap. The model for the \( Y \)-independent electrostatic potential is (after Ref. [43, p.412])

\[
V(X, Z) = \sum_{i=2}^{N} \frac{V_i - V_{i-1}}{\pi} \arctan \frac{\sinh(\pi(X - X_{i-1})/D)}{\cos(\pi Z/D)}
\]

where \( V_i \) are the potential at the \( N \) electrodes (and normally \( V_1 = 0 \) refers to the incident beam energy), \( X_i \) are the locations of the zero-length slits, \( X \) is the distance from the origin taken at the first slit (located at \( X_1 \equiv 0 \) 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 25).

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

In the mirror mode (option \( MT = 11 \) for vertical mid-plane or \( MT = 12 \) for horizontal mid-plane) stepwise integration starts at \( X = -L_1 \) (entrance of the first electrode) and terminates either when back to \( X = -L_1 \) or when reaching \( X = L - L_1 \) (end of the \( N - th \) electrode). In the latter case particles are stopped with their index \( IEX \) set to -8 (see section 4.6.10 on page 166). Normally \( X_1 \) should exceed \( 3D \) (enough that \( V(X < X_1) \) 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 \( X = -L_1 \) (entrance of the first electrode) and terminates either when reaching \( X = L - L_1 \) (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 27: 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 [43]**

The device works as mirror or lens, horizontal or vertical. It is made of $N$ 2-plate electrodes and has mid-plane symmetry$^6$.

Electrode slits are circular, concentric with radii $R_1$, $R_2$, ..., $R_{N-1}$, $D$ is the mirror/lens gap. The model for the mid-plane ($Z = 0$) radial electrostatic potential is (after Ref. [43, p.443])

$$V(r) = \sum_{i=2}^{N} \frac{V_i - V_{i-1}}{\pi} \arctan \left( \sinh \frac{\pi(r - R_{i-1})}{D} \right)$$

where $V_i$ are the potential at the $N$ electrodes (and normally $V_1 = 0$ refers to the incident beam energy). $r$ is the current radius.

The mid-plane field $\vec{E}(r)$ and its $r$-derivatives are first derived by differentiation, then $\vec{E}(r, Z)$ and derivatives are obtained from Taylor expansions and Maxwell relations. Eventually a transformation to the rotating frame provides $\vec{E}(X, Y, Z)$ and derivatives as involved in eq. 1.2.15.

Stepwise integration starts at entrance (defined by $RE, TE$) of the first electrode and terminates when rotation of the reference rotating frame ($RM, X, Y$) has reached the value $AT$. Normally, $R_1 - RE$ and $R_1 - RS$ should both exceed $3D$ (so that potential tails have negligible effect in terms of trajectory behavior).

Positioning of the element is performed by means of $KPOS$ (see section 4.6.7).

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

---

$^6$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

$$\vec{M}_E = \vec{E}_1 + \vec{E}_2 + \vec{E}_3 + \ldots + \vec{E}_{10}$$

$$\frac{\partial \vec{M}_E}{\partial X} = \frac{\partial \vec{E}_1}{\partial X} + \frac{\partial \vec{E}_2}{\partial X} + \frac{\partial \vec{E}_3}{\partial X} + \ldots + \frac{\partial \vec{E}_{10}}{\partial X}$$

$$\frac{\partial^2 \vec{M}_E}{\partial X \partial Z} = \frac{\partial^2 \vec{E}_1}{\partial X \partial Z} + \frac{\partial^2 \vec{E}_2}{\partial X \partial Z} + \frac{\partial^2 \vec{E}_3}{\partial X \partial Z} + \ldots + \frac{\partial^2 \vec{E}_{10}}{\partial X \partial Z}$$

etc.

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 25), 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 $[41, 44]$ $K_n = \frac{1}{2} \frac{\gamma}{\gamma^2 - 1} \frac{V_n}{R_0^n}$

($V_n =$ 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 $RX_i$ 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 29: An electric multipole combining skew-quadrupole ($\vec{E}_2 \neq \vec{0}$, $R_2 = \pi/4$) and skew-octupole ($\vec{E}_4 \neq \vec{0}$, $R_4 = \pi/8$) components ($\vec{E}_1 = \vec{E}_3 = \vec{E}_5 = ... = \vec{E}_{10} = \vec{0}$) [44].
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

```fortran
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 \( I_X \) 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 \textit{ENORM} prior to ray-tracing. As well the longitudinal coordinate \( X \) is normalized with a \textit{XNORM} coefficient (useful to convert to centimeters, the working units in \textit{zgoubi}).

\( X \)-cylindrical symmetry is assumed, resulting in \( E_Y \) and \( E_Z \) 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 \textit{CARTEMES} by means of the flag \textit{ID} and coefficients \( A, B, C, A', B', C' \).

Use \textit{PARTICUL} prior to \textit{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 [47].

*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;
- $MOD \leq 19:$ Cartesian mesh;
- $MOD \geq 20:$ cylindrical mesh.

- $MOD=0:$ a single map, including QF and QD, is read for tracking.
- $MOD=22:$ a single map, including QF and QD, is read.
- $MOD=24:$ field at particle is interpolated from a (QF,QD) pair of maps, closest to current $(Y_F, Y_D)$ value, taken from of set of (QF,QD) pairs registered in file *FNAME*.

The parameters that move/position the maps, as $(Y_F, Y_D)$, are accessible from the FIT, allowing to adjust the cell tunes.

*EMMA* works much like *TOSCA*. Refer to that keyword, and to the *FORTRAN* file *emmac.f*, for details.
**FFAG : FFAG magnet, \( N \)-tuple [38, 39]**

*FFAG* works much like *DIPOLES* 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 *DIPOLES* 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. [38]. 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 *DIPOLE*, *DIPOLE-M*) (see Fig. 11-A page 82, and Fig. 30)

- \( ACN_i \): arbitrary inner angle, used for EFB’s 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 30: Definition of a dipole \( N \)-tuple (\( N = 3 \), a triplet here) using the *DIPOLES* or *FFAG* procedures.](image)

**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

\[ B_{Zi}(R, \theta) = B_{Z0,i} F_i(R, \theta) (R/R_M)^{K_i}, \]

wherein \( B_{Z0,i} \) is a reference field, at reference radius \( R_M \), whereas \( F(R, \theta) \) is calculated as described below.

**Calculation of \( F_i(R, \theta) \)**

The fringe field coefficient \( F_i(R, \theta) \) associated with a dipole is computed as in the procedure *DIPOLES* (eq. 4.4.12), including (rigorously if the interpolation method is used, see page 106, or to order zero if the analytic method is used, see page 107) radial dependence of the gap size.
\[ g(R) = g_0 \left( RM/R \right)^\kappa \]  

(4.4.15)

so to simulate the effect of gap shaping on \( B_{Z_1}(R, \theta) \mid_R \) 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 **DIPOLES** procedure (see page 106).

**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 [39, 45]

**FFAG-SPI** works much like **FFAG** as to the field modelling, apart from the axial dependence of the field.

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 (Fig. 31). This is done in the way described below.

The dimensioning of the magnet is defined by

\[
AT : \text{total angular aperture} \\
RM : \text{mean radius used for the positioning of field boundaries}
\]

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

\[
ACN_i : \text{arbitrary inner angle, used for EFB’s positioning} \\
\omega : \text{azimuth of an EFB with respect to } ACN \\
\xi : \text{spiral angle}
\]

with $ACN_i$ and $\omega$ as defined in Fig. 31 (similar to what can be found in Figs. 30 and 11-A).

![Figure 31: A $N$-tuple spiral sector FFAG magnet ($N = 3$ here, simulating active field clamps at entrance and exit side of a central dipole).](image)

### Calculation of the Field From a Single Dipole

The magnetic field is calculated in polar coordinates. At all $(R, \theta)$ in the median plane ($Z = 0$), the magnetic field due a single one (index $i$) of the dipoles of a $N$-tuple FFAG magnet is written
\[ B_{Z_1}(R, \theta) = B_{Z_0,i} F_i(R, \theta) \left( \frac{R}{R_M} \right)^{K_i} \]

wherein \( B_{Z_0,i} \) is a reference field, at reference radius \( R_M \), whereas \( F(R, \theta) \) is calculated as described below.

**Calculation of \( F_i(R, \theta) \)**

The fringe field coefficient \( F_i(R, \theta) \) associated with a dipole is computed as in the procedure DIPOLES (eq. 4.4.12), including radial dependence of the gap size

\[ g(R) = g_0 \left( \frac{R_M}{R} \right)^{\kappa} \]  \hspace{1cm} (4.4.16)

so to simulate the effect of gap shaping on \( B_{Z_1}(R, \theta) \) 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 DIPOLES procedure (see page 106).
MAP2D : 2-D Cartesian uniform mesh field map - arbitrary magnetic field [48]

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 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’:

```fortran
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)
      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 25, 1.4.3 page 29). 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’:

```fortran
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 25, 1.4.3 page 29). 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 LABELs. Using '.plt' as a second LABEL will cause storage of current coordinates into zgoubi.plt.
MULTIPOLE : Magnetic multipole

The simulation of multipolar magnetic field $\vec{M}$ by MULTIPOLE 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

$$\vec{M} = \vec{B}_1 + \vec{B}_2 + \vec{B}_3 + \ldots + \vec{B}_{10}$$

$$\frac{\partial \vec{M}}{\partial X} = \frac{\partial \vec{B}_1}{\partial X} + \frac{\partial \vec{B}_2}{\partial X} + \frac{\partial \vec{B}_3}{\partial X} + \ldots + \frac{\partial \vec{B}_{10}}{\partial X}$$

$$\frac{\partial^2 \vec{M}}{\partial X \partial Z} = \frac{\partial^2 \vec{B}_1}{\partial X \partial Z} + \frac{\partial^2 \vec{B}_2}{\partial X \partial Z} + \frac{\partial^2 \vec{B}_3}{\partial X \partial Z} + \ldots + \frac{\partial^2 \vec{B}_{10}}{\partial X \partial Z}$$

etc.

The independent components $\vec{B}_1, \vec{B}_2, \vec{B}_3, \ldots, \vec{B}_{10}$ 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 $E_2 (E_3, \ldots, E_{10})$ at entrance, and $S_2 (S_3, \ldots, S_{10})$ at exit, such that the extent is $\lambda_E * E_2 (\lambda_E * E_3, \ldots, \lambda_E * E_{10})$ at entrance and $\lambda_S * S_2 (\lambda_S * S_3, \ldots, \lambda_S * S_{10})$ 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 $RX_i$ 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 4.6.7).
4.4 Optical Elements and Related Numerical Procedures

**OCTUPOLE**: Octupole magnet (Fig. 32)

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 \( Z \)

\[
V(X, Y, Z) = \left( G - \frac{G''}{20} (Y^2 + Z^2) + \frac{G'''}{960} (Y^2 + Z^2)^2 \right) (Y^3 Z - YZ^3)
\]

with \( G_0 = \frac{B_0}{R_0^3} \)

The modelling of the fringe field form factor \( G(X) \) is described under QUADRUPO, p. 132.

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

\[
\begin{align*}
B_X &= 0 \\
B_Y &= G_0 (3Y^2 - Z^2) Z \\
B_Z &= G_0 (Y^2 - 3Z^2) Y
\end{align*}
\]

Figure 32: Octupole magnet
POISSON : Read magnetic field data from POISSON output

This keyword allows reading a field profile $B(X)$ from POISSON output. Let $FNAME$ be the name of this output file (normally, $FNAME = outpoi.lis$); the data are read following the FORTRAN statements here under:

```fortran
I = 0
11 CONTINUE
   I = I + 1
101 FORMAT(I1, I3, I4, E15.6, 2F11.5, 2F12.3)
   GOTO 11
10 CONTINUE
...
```

where $X(I)$ is the longitudinal coordinate, and $B(I)$ is the Z component of the field at a node (I) of the mesh. K’s and R’s are dummy variables appearing in the POISSON output file outpoi.lis but not used here.

From this field profile, a 2-D median plane map is built, with a rectangular and uniform mesh; mid-plane symmetry is assumed. The field at each node $(X_i, Y_j)$ of the map is $B(X_i)$, independent of $Y_j$ (i.e., the distribution is uniform in the $Y$ direction).

For the rest, POISSON works in a way similar to CARTEMES.
POLARMES: 2-D polar mesh magnetic field map

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. 33).

---

**Figure 33:** 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

$$B = B_0 \left( U + \frac{N}{R_0} Y + \frac{B}{R_0^2} Y^2 + \frac{G}{R_0^3} Y^3 \right)$$

$$\frac{\partial B}{\partial Y} = B_0 \left( \frac{N}{R_0} + 2 \frac{B}{R_0^2} Y + 3 \frac{G}{R_0^3} Y^2 \right)$$

$$\frac{\partial^2 B}{\partial Y^2} = B_0 \left( 2 \frac{B}{R_0^2} + 6 \frac{G}{R_0^3} Y \right)$$

$$\frac{\partial^3 B}{\partial Y^3} = 6B_0 \frac{G}{R_0^3}$$

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. 34)

The length of the magnet XL is the distance between the effective field boundaries (EFB), Fig. 35. 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 EFB’s, in the field fall off regions, will be \( \pm X_E \) at the entrance, and \( \pm X_S \) at the exit (Fig. 35), 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 EFB’s, \( \vec{B}(X, Y, Z) \) 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 \( Z \).

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

where \( G \) is the gradient on axis [34]:

\[
G(s) = \frac{G_0}{1 + \exp P(s)} \quad \text{with} \quad G_0 = \frac{B_0}{R_0}
\]

and,

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

where, \( s \) 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

\[
B_X = 0 \\
B_Y = G_0 Z \\
B_Z = G_0 Y
\]
Figure 34: Quadrupole magnet

Figure 35: Scheme of the longitudinal field gradient $G(X)$.

$O(X)$ 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 $-XE$ to $XL + XS$, 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. 36).

\[
X = -R \cos \left( \frac{\omega S}{\beta c} + \epsilon \right) - \frac{\alpha S}{\omega \beta c} + \frac{C_1}{\omega} \\
Y = R \sin \left( \frac{\omega S}{\beta c} + \epsilon \right) - \frac{\alpha}{\omega^2} - \frac{C_2}{\omega} + Y_0 \\
Z = S \sin(P_0) + Z_0
\]

where, \( S \) is the path length in the separator, \( \alpha = -\frac{E c^2}{\gamma} \), \( \omega = -\frac{B c^2}{m \gamma} \), \( C_1 = \beta \sin(T_0) \cos(P_0) \) and \( C_2 = \frac{1}{\beta c \cos(T_0) \cos(P_0)} \) 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 36: Horizontal separation between a wanted particle, \( (W) \), and an unwanted particle, \( (U) \). \( (W) \) undergoes a linear motion while \( (U) \) undergoes a cycloidal motion.
SEXTUPOLE: Sextupole magnet (Fig. 37)

The meaning of parameters for SEXTUPOLE is the same as for QUADRUPO.

In fringe field regions the magnetic field \(\mathbf{B}(X, Y, Z)\) and its derivatives up to fourth order are derived from the scalar potential approximated to 7th order in \(Y\) and \(Z\)

\[
V(X, Y, Z) = \left( G - \frac{G''}{16} (Y^2 + Z^2) + \frac{G'''}{640} (Y^2 + Z^2)^2 \right) \left( Y^2 Z - \frac{Z^3}{3} \right)
\]

with \(G_0 = \frac{B_0}{R_0^2}\)

The modelling of the fringe field form factor \(G(X)\) is described under QUADRUPO, p. 132.

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

\[
\begin{align*}
B_X &= 0 \\
B_Y &= 2G_0 Y Z \\
B_Z &= G_0 (Y^2 - Z^2)
\end{align*}
\]
SOLENOID : Solenoid (Fig. 38)

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. 38).

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

Method 1 : \( \vec{B}(X, r) \), \( r = (Y^2 + Z^2)^{1/2} \) and its derivatives up to second order at all \( (X, Y, Z) \) are calculated following Ref. [49], 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 [50].

This analytical model for the solenoidal field allows simulating an extended range of coil geometries (length, radius) provided that the coil thickness is small enough compared to the mean radius \( R_0 \).

In particular the field on-axis writes (taking \( X = r = 0 \) at the center of the solenoid)

\[
B_X(X, r = 0) = \frac{\mu_0 NI}{2XL} \left[ \frac{XL/2 - X}{\sqrt{(XL/2 - X)^2 + R_0^2}} + \frac{XL/2 + X}{\sqrt{(XL/2 + X)^2 + R_0^2}} \right]
\]

and yields the magnetic length

\[
L_{mag} = \frac{\int_{-\infty}^{\infty} B_X(X, r < R_0) dX}{B_X(X = r = 0)} = XL \sqrt{1 + \frac{4R_0^2}{XL^2}} > XL \quad (4.4.17)
\]

with in addition

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

Method 2 : The second method available uses eq. 4.4.17 above as a 1-D model and uses off-axis extrapolation to derive the field and its derivatives at all \( (X, Y, Z) \), following the method described in section 1.3.1.

Figure 38: Solenoidal magnet.
SPINR : Spin rotation

Spin precession, a local transformation.

The precession is defined by its axis and its value.
TOSCA: 2-D and 3-D Cartesian or cylindrical mesh field map

TOSCA is dedicated to the reading and treatment of 2-D or 3-D Cartesian or cylindrical mesh field maps as delivered by the TOSCA magnet computer code standard output.

A pair of flags, MOD, MOD2, determine whether Cartesian or Z-axis cylindrical mesh is used, and the nature of the field map data set.

The total number of field data files to be read is determined by the MOD 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. 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':

```fortran
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)
    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 that these are binary files, FNAME must begin with ‘B_’ or ‘b_’.

In addition to the MOD=1, 2 cases above, one can have MOD=12 and in that case a single file contains the all 3-D field map. See table below and the FORTRAN subroutine fmapw.f and its entries FMAPR, FMAPR2, for more details, in particular the formatting of the field map data file(s).

7Use MAP2D in case non-zero $B_X$, $B_Y$ are to be taken into account in a 2-D map.
### MOD MOD2

<table>
<thead>
<tr>
<th>MOD ≤ 19 : Cartesian mesh</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0 and IZ = 1</td>
<td>none</td>
<td>2-D map, a single data file for $B_Z(X,Y)</td>
</tr>
<tr>
<td>0 and IZ &gt; 1</td>
<td>none</td>
<td>3-D map, 1+IZ/2 data files of upper half of magnet, one per $(X,Y)</td>
</tr>
<tr>
<td>1</td>
<td>1, 2, 3</td>
<td>As previous case, just a different reading format</td>
</tr>
<tr>
<td>0</td>
<td>none</td>
<td>2- or 3-D map, IZ data files, one per $(X,Y)$ plane, no symmetry assumed</td>
</tr>
<tr>
<td>12</td>
<td>1, 2, 3</td>
<td>As previous case, just a different reading format</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>3-D map, single file, upper half of magnet, symmetry with respect to $(X,Y)$ mid-plane</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>3-D map, single file, whole magnet, no symmetry assumed</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>3-D map, single file, 1/8th of the magnet, symmetry wrt. $(X,Y), (X,Z), (Y,Z)$ planes</td>
</tr>
<tr>
<td>3</td>
<td>none</td>
<td>AGS main magnet field map, 2-D, mid-plane symmetry assumed</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MOD ≥ 20 : Cylindrical mesh</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>20, 21</td>
<td>3-D map, single file, half a magnet, cyl. symmetry with respect to $(Y,Z)$ plane</td>
<td></td>
</tr>
<tr>
<td>22, 24</td>
<td>3-D map, single file, half a magnet, cyl. symmetry with respect to $(X,Y)$ mid-plane</td>
<td></td>
</tr>
</tbody>
</table>

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 a 3-D field map) are normalized by the $X$-, $Y$-, $Z$-NORM coefficient (useful to convert to centimeters, the working units in zgoubi).

At each step of the trajectory of a particle inside the map, the field and its derivatives are calculated
- in the case of 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 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.

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.
TRANSMA T : Matrix transfer

TRANSMA T performs a second order transport of the particle coordinates in the following way

\[ X_i = \sum_j R_{ij} X^0_j + \sum_{j,k} T_{ijk} X^0_j X^0_k \]

where, \( X_i \) stands for any of the current coordinates \( Y, T, Z, P \), path length and momentum dispersion, and \( X^0_i \) 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 [28]. 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 TRANSMA T.
TRAROT: Translation-Rotation of the reference frame

UNDER DEVELOPMENT. Check before use.

This procedure transports particles into a new frame by translation and rotation. Effect on spin tracking, particle decay and gas-scattering are taken into account (but not on synchrotron radiation).
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 $X_1$ (resp. $X_2$, $X_3$). The distance between the electrodes is $D$. The potentials are $V_1$ and $V_2$. The inner radius is $R_0$ (Fig. 39). The model for the electrostatic potential along the axis is \[ V(x) = \frac{V_2 - V_1}{2\omega D} \left[ \ln \frac{\cosh \frac{\omega (x + \frac{X_2}{2} + D)}{R_0}}{\cosh \frac{\omega (x + \frac{X_2}{2})}{R_0}} + \ln \frac{\cosh \frac{\omega (x - \frac{X_2}{2} - D)}{R_0}}{\cosh \frac{\omega (x - \frac{X_2}{2})}{R_0}} \right] \]

($x =$ 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 $X_1 + X_2 + X_3 + 2D$; stepwise integration starts at entrance of the first electrode and terminates at exit of the third one.

Figure 39: 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. 40).

Figure 40: 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. 36). It must be preceded by PARTICUL for the definition of particle mass and charge.

The length $X_L$ of the element is the distance between its entrance and exit EFB's. 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 [34]

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

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., $X_L \leq X_E + X_S$), 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.
**YM Y:** Reverse signs of \( Y \) and \( Z \) reference axes

YM Y performs a 180° rotation of particle coordinates with respect to the X-axis, as shown in Fig. 41. This is done by means of a change of sign of \( Y \) and \( Z \) axes, and therefore coordinates, as follows:

\[
Y_2 = -Y_1, \quad T_2 = -T_1, \quad Z_2 = -Z_1 \quad \text{and} \quad P_2 = -P_1
\]

Figure 41: The use of \( YMY \) in a sequence of two identical dipoles of opposite signs.
4.5 Output Procedures

These procedures are dedicated to the storage or printing of particle coordinates, histograms, spin coordinates, etc. They may be called for at any spot in the data pile.
FAISCEAU, FAISCNL, FAISTORE : Print/Store particle coordinates

- **FAISCEAU** can be introduced anywhere 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** produces a lot more information on particles at current location, including spin components, decay distance, mass, charge, etc. (see list below), and stores it in a dedicated file **FNAME** (advised name is **FNAME** = ‘zgoubi.fai’ (formatted write) or ‘b_zgoubi.fai’ (binary write) if post-processing with zpop should follow). This file may further on be read by means of **OBJET**, option **KOBJ**= 3, or used for other purposes such as graphics (see Part D of the Guide).

The data written to that file are formatted and ordered according to the **FORTRAN** sequence in the subroutine impfai.f, where details and possible updates are to found. The following is an instance:

```fortran
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), -1.0D0 + FO(1, I), (FO(J, I), J = 2, MXJ),
     tag, initial D-1,Y,T,Z,P,S,Time
  2 -1.0D0 + F(1, I), F(2, I), F(3, I), (F(J, I), J = 4, MXJ),
     current D_i,Y,T,Z,P,S,Time
  > (SI(J, I), J = 1, 4), (SF(J, I), J = 1, 4),
     spin components, initial and current energy
  > ENEKI, ENERG,
  > I, IREP(I), SORT(I), (AMQ(J, I), J = 1, 5), RET(I), DPR(I), PS,
     particle#, loss S, mass, Q, G, lifetime
  5 BORO, IPASS, NOEL, KLEY, LBL1, LBL2, LET(I)
  6 BORO, pass#, element#, keyword, labels
  7 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), -1.0D0 + FO(1, I), (FO(J, IT), J = 2, MXJ),
     tag, initial D-1,Y,T,Z,P,S,Time
  2 -1.0D0 + F(1, I), F(2, I), F(3, I),
     current D_i,Y,T,Z,P,S,Time
  > (SI(J, I), J = 1, 4), (SF(J, I), J = 1, 4),
     spin components, initial and current energy
  > ENEKI, ENERG,
  > I, IREP(I), SORT(I), (AMQ(J, I), J = 1, 5), RET(I), DPR(I), PS,
     particle#, loss S, mass, Q, G, lifetime
  7 BORO, IPASS, NOEL,
  8 TX1, KLEY, TX1, TX1, TX1, TX1, TX1, TX1, LET(I), TX1
     BORO, pass#, element#, keyword, labels
  1 CONTINUE
ENDIF
110 FORMAT (1X,
    C1 KEX, XXXO, (FO(J, IT), J = 2, MXJ),
    > 1F, 1X, I2, 7(1X, E16.8)
    C2 XXX, Y, T * 1.0D3,
    > 1F, 1X, E24.16
    C3 2, P = 1.0D3, SAR, TAR
    > , 4(1X, E24.16)
    C4 Sx0, SY0, S20, So, SX, SY, S2, 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)' "FRMFAI.H"
    INCLUDE "FRMFAI.H"
    INCLUDE "FRMFAI.H"
    a continuation
    1 CONTINUE
```

The meaning of the main data is the following (see the keyword **OBJET**):
4.5 Output Procedures

- **LET**\((I)\)
  : one-character string, for tagging particle number \(I\)

- **IEX, I, IREP\((I)\)**
  : flag, particle number, index

- **FO\((1 − 6, I)\)**
  : coordinates \(D, Y, T, Z, P\) and path length at the origin of the structure

- **\(F(1 − 6, I)\)**
  : *idem* at the current position

- **SORT\((I)\)**
  : path length at which the particle has possibly been stopped
  (see **CHAMBR** or **COLLIMA**)

- **RET\((I), DPR(I)\)**
  : synchrotron phase space coordinates; \(RET = \text{phase (radian)}, DPR = \text{momentum dispersion (MeV/c)}\) (see **CAVITE**)

- **IPASS**
  : turn number (see **REBELOTE**)

- **etc.**

- **FAISTORE** has an effect similar to **FAISCNL**, with two more features.
  - On the first data line, **FNAME** may be followed by a series of up to 10 **LABEL**s. 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 input 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.

**Note**

Binary storage 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 \(b_{\text{FNAME}}\) or \(B_{\text{FNAME}}\) (e.g., ‘b_zgoubi.fai’). The **FORTRAN 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_’.
FOCALE, IMAGE[S] : Particle coordinates and beam size; localization and size of horizontal waist

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

IMAGE computes the location and size of the closest horizontal waist.

IMAGES has the same effect as IMAGE, but, in addition, for a non-monochromatic beam it calculates as many waists as there are distinct momenta in the beam, provided that the object has been defined with a classification of momenta (see OBJET, KOBJ= 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, FOCALE) or of each group of momenta (IMAGES)

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

  $Y = Y_1 + X \cdot \tan T_1$

  where $Y_1$ and $T_1$ are the coordinates of the first particle of the beam (IMAGE, FOCALE) or the first particle of each group of momenta (IMAGES).

- **Transverse position of the center of mass of the waist (IMAGE[S]) or of the beam (FOCALE), with respect to the reference trajectory**

  $YM = \frac{1}{N} \sum_{i=1}^{N} (Y_i + X \cdot \tan T_i) - Y = \frac{1}{N} \sum_{i=1}^{N} Y M_i$

- **FWHM of the image (IMAGE[S]) or of the beam (FOCALE), and total width, respectively, $W$ and $WT$**

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

  $WT = \max(YM_i) - \min(YM_i)$
FOCALEZ, IMAGE[S]Z : Particle coordinates and beam size ; localization and size of vertical waist

Similar to FOCALE and IMAGE[S], but the calculations are performed with respect to the vertical coordinates $Z_i$ and $P_i$, in place of $Y_i$ and $T_i$. 
**HISTO : 1-D histogram**

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.
IMAGE[S][Z] : Localization and size of vertical waists
See FOCALE[Z].
**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 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, 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,

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

  In addition, if **OBJET**, **KOBJ** = 5.01 is used (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 **MATRIX**

  \[
  \begin{pmatrix}
  -R_{11}^2 -2R_{11}R_{12} & 0 & R_{12}^2 \\
  R_{21}^2 & -2R_{21}R_{22} & R_{22}^2 \\
  0 & 0 & 0
  \end{pmatrix}
  \]

  at **OBJET**

  \[
  \Delta \phi_Y = \text{Atan} \left( \frac{R_{12}}{R_{11}\beta_{Y,\text{obj}} - R_{12}\alpha_{Y,\text{obj}}} \right), \quad \Delta \phi_Z = \text{Atan} \left( \frac{R_{34}}{R_{33}\beta_{Z,\text{obj}} - R_{34}\alpha_{Z,\text{obj}}} \right),
  \]

  \[
  (4.5.1)
  \]

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

  and print these out.

- **If** **IORD** = 2, 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.

An automatic generation of an appropriate object for the use of **MATRIX** can be obtained using the procedure **OBJET** (pages 49, 229), as follows

- **If** **IORD** = 1, use **OBJET**(**KOBJ** = 5.01, **NN**=01,99]), that generates up to 99*11 sets of initial coordinates. In this case, up to ninety nine matrices may be calculated, each one wrt. to the reference trajectory of concern.

- **If** **IORD** = 2, use **OBJET**(**KOBJ** = 6) that generates 61 sets of 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 [30].

If $IORD = 2$ additional periodic parameters are computed such as chromaticities, beta-function momentum dependence, etc.

Addition of $zgoubi.MATRIX.out$ next to $IORD$, $IFOC$ will cause stacking of $MATRIX$ output data into $zgoubi.MATRIX.out$ file (convenient for use with e.g. gnuplot type of data treatment software).
PICKUPS: Beam centroid path; closed orbit

PICKUPS computes the beam centroid location, from average value of particle coordinates as observed at LABEL'ed keywords.

In conjunction with REBELOTE, this procedure computes by the same method the closed orbit in the periodic structure.

The LABEL list of concern follows the keyword PICKUPS.
4.5 Output Procedures

PLOTDATA : Intermediate output for the PLOTDATA graphic software [52]

PLOTDATA was at the origin implemented for the purpose of plotting particle coordinates using the TRIUMF PLOTDATA package. 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 FORTRAN subroutine pltdat.f.
SPNPRNL, SPNSTORE, SPNPRT : Print/Store spin coordinates

- **SPNPRT** can be introduced anywhere in a structure. It produces a listing (into zgoubi.res) of the initial and actual coordinates and modulus of the spin of the IMAX particles, at the location where it stands, together with their Lorentz factor γ and other infos as the mean values of the spin components.

- **SPNPRNL** has similar effect to **SPNPRT**, 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 FORTRAN sequence found in the subroutine spnprn.f, with meaning of printed quantities as follows :

  \[
  \begin{align*}
  LET(I), IEX(I) &: \text{ tagging character and flag (see OBJET)} \\
  SI(1-4,I) &: \text{ spin components } SX, SY, SZ \text{ and modulus, at the origin} \\
  SF(1-4,I) &: \text{ idem at the current position} \\
  GAMMA &: \text{ Lorentz relativistic factor} \\
  I &: \text{ particle number} \\
  IMAX &: \text{ total number of particles ray-traced (see OBJET)} \\
  IPASS &: \text{ turn number (see REBELOTE)}
  \end{align*}
  \]

- **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 LABEL’s 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 ≥ 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 FORTRAN WRITE output list is the same as in the FORMATTED case above.
4.5 Output Procedures

**SRPRNT : Print SR loss statistics**

`SRPRNT` may be introduced anywhere in a structure. It produces a listing (into zgoubi.res) of current state of statistics on several parameters related to SR loss presumably activated beforehand with keyword `SRLOSS`. 
TWISS : Calculation of periodic optical parameters

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.
- A first pass (the only one if KTW=1) through the structure allows computing the periodic optical functions from the rays.
- The periodic dispersions are used to define chromatic closed orbits at ±δp/p. A second and a third pass (which terminate the process if KTW=2) with chromatic objects centered respectively on ±δp/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.
4.6 Complements Regarding Various Functionalities

4.6.1 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

\[ B_{\rho_{\text{ref}}} = BORO \rightarrow B_{\rho_{\text{ref}}} = BORO + \delta B_{\rho_s} \]

with normally \( \delta B_{\rho_s} \) 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 4.6.6).

4.6.2 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 4.6.6), 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).

4.6.3 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 42.

![Figure 42: 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.](image)

4.6.4 Checking Fields and Trajectories Inside Optical Elements

In all optical elements, an option *IL* is available. It is normally set to *IL = 0* and in this case has no effect.
IL = 1 causes a print in zgoubi.res of particle coordinates and field along trajectories in the optical element. In the meantime, a calculation and summation of the values of $\nabla \cdot \vec{B}$, $\nabla \times \vec{B}$ and $\nabla^2 \vec{B}$ (same for $\vec{E}$) at all integration steps is performed, which allows a check of the behavior of $\vec{B}$ (or $\vec{E}$) in field maps (all these derivatives should normally be zero).

IL = 2 causes a print of particle coordinates and other informations into the file zgoubi.plt at each integration step; this information can further be processed with zpop. In order to minimize the volume of that storage file (when dealing with small step size, large number of particles, etc.) it is possible to print out every other $10^n$ integration step by taking $IL = 2 \times 10^n$ (for instance, $IL = 200$ would cause output into zgoubi.plt every 100 other step).

When dealing with field maps (e.g., CARTEMES, ELREVL, TOSCA), an option index $IC$ is available. It is normally set to $IC = 0$ and in this case has no effect.

$IC = 1$ causes a print of the field map in zgoubi.res.

$IC = 2$ will cause a print of field maps into the file zgoubi.map which can further be processed with zpop for plotting and other data treatment purposes.

### 4.6.5 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 4.6.4, $IL = 2$ option).

A keyword in zgoubi.dat accepts two LABEL’s. The first one is used for the above mentioned purposes, the second one (set to “.plt” in that case) is used essentially with MARKER 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).

### 4.6.6 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 70), 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.

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.

---

8See Part D of the Guide.
4.6 Complements Regarding Various Functionalities

**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).

### 4.6.7 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. 43. The effect is equivalent to a \( \text{CHANGREF}(XCE, YCE, ALE) \) upstream of the optical element, followed by \( \text{CHANGREF}(XCS, YCS, ALS = -ALE) \) downstream of it, with computed \( XCS, YCS \) values as schemed in Fig. 43.

\( KPOS = 3 \) option is available for a limited number of magnets (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 44).

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 \( \theta_Z/2 \) such that \( L = 2\frac{BORO}{B1} \sin(\theta_Z/2) \), wherein \( L \) = geometrical length, \( BORO = \) reference rigidity as defined in OBJET.
Next, the optical element is Y-shifted by $YCE$ ($XCE$ is not used) in a direction orthogonal to the new magnet axis (i.e., at an angle $ALE + \pi/2$ wrt. the $X$ axis of the incoming reference frame).

$KPOS = 4$ applies to a limited number of optical elements, as AGSMM (AGS main magnet). By default, it aligns the magnet in a way similar to $KPOS = 3$, with reference frame $Z$-rotated by $\theta_Z/2$ as drawn from $L = 2 \frac{BORO}{B_1} \sin(\theta_Z/2)$. However magnet mis-alignment (alignment errors) are handled in a specific way, as follows.
All 6 types of misalignments, namely, X-, Y-, Z-shift, X-, Y-, Z-rotation, can be accounted for, in an arbitrary order. They are specified using the “new style” CHANGREF method as described in page 96. Longitudinal rotation “XR” is taken wrt. the longitudinal axis, whereas radial and axial rotations, “YR” and “ZR”, are with respect to an axis going through the center of the magnet.

Figure 45: Pitch angle, \( \varphi \), in the “YR” type of rotation, using \( KPO\text{S} = 4 \). \( M \) is the new position, in the rotated plane \((Y, Z_{\text{new}})\), of a particle with velocity \( \vec{v}/M_oM \) located at former position \( M_o \) in the old, \((Y, Z_{\text{old}})\), plane. \( M', m, m' \) are projections of \( M, M'' \) is projection of \( M' \). \( T \) and \( P \) are the horizontal and vertical angles as defined in Fig. 1. \( p \) is the projection of \( P \).

Figure 46: Positioning of a polar field map, using \( KPO\text{S} = 1 \).
Description of the Available Procedures

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. 46; 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 82A).

4.6.8 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 \( XPAS = \#|C|S \), 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.

4.6.9 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 MCOBJECT together with REBELOTE, put at the end of the optical structure, with its argument NPASS being the number of passes through REBELOTE, and \((NPASS+1) \times IMAX\) 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.

4.6.10 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 27). In all other cases the particle will be stopped.

4.6.11 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 MCOBJECT.
PART B
Keywords and input data formatting
### Glossary of Keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGSMEM</td>
<td>AGS main magnet</td>
<td>175</td>
</tr>
<tr>
<td>AGSQUAD</td>
<td>AGS quadrupole</td>
<td>176</td>
</tr>
<tr>
<td>AIMANT</td>
<td>Generation of dipole mid-plane 2-D map, polar frame</td>
<td>177</td>
</tr>
<tr>
<td>AUTOREF</td>
<td>Automatic transformation to a new reference frame</td>
<td>181</td>
</tr>
<tr>
<td>BEAMBEAM</td>
<td>Beam-beam lens</td>
<td>182</td>
</tr>
<tr>
<td>BEND</td>
<td>Bending magnet, Cartesian frame</td>
<td>183</td>
</tr>
<tr>
<td>BINARY</td>
<td>BINARY/FORMATTED data converter</td>
<td>184</td>
</tr>
<tr>
<td>BREVOL</td>
<td>1-D uniform mesh magnetic field map</td>
<td>185</td>
</tr>
<tr>
<td>CARTEMES</td>
<td>2-D Cartesian uniform mesh magnetic field map</td>
<td>186</td>
</tr>
<tr>
<td>CAVITE</td>
<td>Accelerating cavity</td>
<td>188</td>
</tr>
<tr>
<td>CHAMBR</td>
<td>Long transverse aperture limitation</td>
<td>189</td>
</tr>
<tr>
<td>CHANGREF</td>
<td>Transformation to a new reference frame</td>
<td>190</td>
</tr>
<tr>
<td>CIBLE</td>
<td>Generate a secondary beam following target interaction</td>
<td>191</td>
</tr>
<tr>
<td>COLLIMA</td>
<td>Collimator</td>
<td>192</td>
</tr>
<tr>
<td>DECAPOLE</td>
<td>Decapole magnet</td>
<td>193</td>
</tr>
<tr>
<td>DIPOLE</td>
<td>Dipole magnet, polar frame</td>
<td>194</td>
</tr>
<tr>
<td>DIPOLE-M</td>
<td>Generation of dipole mid-plane 2-D map, polar frame</td>
<td>195</td>
</tr>
<tr>
<td>DIPOLES</td>
<td>Dipole magnet $N$-tuple, polar frame</td>
<td>197</td>
</tr>
<tr>
<td>DODECAPO</td>
<td>Dodecapole magnet</td>
<td>199</td>
</tr>
<tr>
<td>DRIFT</td>
<td>Field free drift space</td>
<td>200</td>
</tr>
<tr>
<td>EBMULT</td>
<td>Electro-magnetic multipole</td>
<td>201</td>
</tr>
<tr>
<td>EL2TUB</td>
<td>Two-tube electrostatic lens</td>
<td>203</td>
</tr>
<tr>
<td>ELMIR</td>
<td>Electrostatic N-electrode mirror/lens, straight slits</td>
<td>204</td>
</tr>
<tr>
<td>ELMIRC</td>
<td>Electrostatic N-electrode mirror/lens, circular slits</td>
<td>205</td>
</tr>
<tr>
<td>ELMULT</td>
<td>Electric multipole</td>
<td>206</td>
</tr>
<tr>
<td>ELREVOL</td>
<td>1-D uniform mesh electric field map</td>
<td>207</td>
</tr>
<tr>
<td>EMMA</td>
<td>2-D Cartesian or cylindrical mesh field map for EMMA FFAG</td>
<td>208</td>
</tr>
<tr>
<td>END</td>
<td>End of input data list</td>
<td>212</td>
</tr>
<tr>
<td>ESL</td>
<td>Field free drift space</td>
<td>200</td>
</tr>
<tr>
<td>FAISCNL</td>
<td>Store particle coordinates in file FNAME</td>
<td>209</td>
</tr>
<tr>
<td>FAISTORE</td>
<td>Store coordinates every $1P$ other pass at labeled elements</td>
<td>209</td>
</tr>
<tr>
<td>FFAG</td>
<td>FFAG magnet, $N$-tuple</td>
<td>210</td>
</tr>
<tr>
<td>FFAG-SPI</td>
<td>Spiral FFAG magnet, $N$-tuple</td>
<td>211</td>
</tr>
<tr>
<td>FIN</td>
<td>End of input data list</td>
<td>212</td>
</tr>
<tr>
<td>FIT,FIT2</td>
<td>Fitting procedure</td>
<td>213</td>
</tr>
<tr>
<td>FOCALE</td>
<td>Particle coordinates and horizontal beam size at distance $XL$</td>
<td>215</td>
</tr>
<tr>
<td>FOCALEZ</td>
<td>Particle coordinates and vertical beam size at distance $XL$</td>
<td>215</td>
</tr>
<tr>
<td>GASCAT</td>
<td>Gas scattering</td>
<td>216</td>
</tr>
<tr>
<td>GETFIT,VALID</td>
<td>Get values of variables as saved from former FIT[2] run</td>
<td>217</td>
</tr>
<tr>
<td>HISTO</td>
<td>1-D histogram</td>
<td>218</td>
</tr>
<tr>
<td>IMAGE</td>
<td>Localization and size of horizontal waist</td>
<td>219</td>
</tr>
<tr>
<td>IMAGES</td>
<td>Localization and size of horizontal waists</td>
<td>219</td>
</tr>
<tr>
<td>IMAGESZ</td>
<td>Localization and size of vertical waists</td>
<td>219</td>
</tr>
<tr>
<td>IMAGEZ</td>
<td>Localization and size of vertical waist</td>
<td>219</td>
</tr>
<tr>
<td>MAP2D</td>
<td>2-D Cartesian uniform mesh field map - arbitrary magnetic field</td>
<td>220</td>
</tr>
<tr>
<td>MAP2D-E</td>
<td>2-D Cartesian uniform mesh field map - arbitrary electric field</td>
<td>221</td>
</tr>
<tr>
<td>MARKER</td>
<td>Marker</td>
<td>222</td>
</tr>
<tr>
<td>Keywords and input data formatting</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MATRIX</strong> Calculation of transfer coefficients, periodic parameters</td>
<td>223</td>
<td></td>
</tr>
<tr>
<td><strong>MCDESINT</strong> Monte-Carlo simulation of in-flight decay</td>
<td>224</td>
<td></td>
</tr>
<tr>
<td><strong>MCOBJET</strong> Monte-Carlo generation of a 6-D object</td>
<td>225</td>
<td></td>
</tr>
<tr>
<td><strong>MULTIPOL</strong> Magnetic multipole</td>
<td>228</td>
<td></td>
</tr>
<tr>
<td><strong>OBJET</strong> Generation of an object</td>
<td>229</td>
<td></td>
</tr>
<tr>
<td><strong>OBJETA</strong> Object from Monte-Carlo simulation of decay reaction</td>
<td>231</td>
<td></td>
</tr>
<tr>
<td><strong>OCTUPOLE</strong> Octupole magnet</td>
<td>232</td>
<td></td>
</tr>
<tr>
<td><strong>OPTICS</strong> Write out optical functions</td>
<td>233</td>
<td></td>
</tr>
<tr>
<td><strong>ORDRE</strong> Taylor expansions order</td>
<td>234</td>
<td></td>
</tr>
<tr>
<td><strong>PARTICUL</strong> Particle characteristics</td>
<td>235</td>
<td></td>
</tr>
<tr>
<td><strong>PICKUPS</strong> Beam centroid path: closed orbit</td>
<td>236</td>
<td></td>
</tr>
<tr>
<td><strong>PLODATA</strong> Intermediate output for the PLODATA graphic software</td>
<td>237</td>
<td></td>
</tr>
<tr>
<td><strong>POISSON</strong> Read magnetic field data from POISSON output</td>
<td>238</td>
<td></td>
</tr>
<tr>
<td><strong>POLARMES</strong> 2-D polar mesh magnetic field map</td>
<td>239</td>
<td></td>
</tr>
<tr>
<td><strong>PS170</strong> Simulation of a round shape dipole magnet</td>
<td>240</td>
<td></td>
</tr>
<tr>
<td><strong>QUADISEX</strong> Sharp edge magnetic multipoles</td>
<td>241</td>
<td></td>
</tr>
<tr>
<td><strong>QUADRUPLO</strong> Quadrupole magnet</td>
<td>242</td>
<td></td>
</tr>
<tr>
<td><strong>REBELOTE</strong> ‘Do it again’</td>
<td>244</td>
<td></td>
</tr>
<tr>
<td><strong>RESET</strong> Reset counters and flags</td>
<td>245</td>
<td></td>
</tr>
<tr>
<td><strong>SCALING</strong> Power supplies and R.F. function generator</td>
<td>246</td>
<td></td>
</tr>
<tr>
<td><strong>SEPARA</strong> Wien Filter - analytical simulation</td>
<td>247</td>
<td></td>
</tr>
<tr>
<td><strong>SEXQUAD</strong> Sharp edge magnetic multipole</td>
<td>248</td>
<td></td>
</tr>
<tr>
<td><strong>SEXTUPOLE</strong> Sextupole magnet</td>
<td>249</td>
<td></td>
</tr>
<tr>
<td><strong>SOLENOID</strong> Solenoid</td>
<td>251</td>
<td></td>
</tr>
<tr>
<td><strong>SPIN</strong> Spin rotation</td>
<td>250</td>
<td></td>
</tr>
<tr>
<td><strong>SPNPRLN</strong> Store spin coordinates into file FNAME</td>
<td>252</td>
<td></td>
</tr>
<tr>
<td><strong>SPNSTORE</strong> Store spin coordinates every IP other pass at labeled elements</td>
<td>252</td>
<td></td>
</tr>
<tr>
<td><strong>SPNPRT</strong> Print spin coordinates</td>
<td>252</td>
<td></td>
</tr>
<tr>
<td><strong>SPNTRK</strong> Spin tracking</td>
<td>253</td>
<td></td>
</tr>
<tr>
<td><strong>SRLOSS</strong> Synchrotron radiation loss</td>
<td>254</td>
<td></td>
</tr>
<tr>
<td><strong>SRPRNT</strong> Print SR loss statistics</td>
<td>255</td>
<td></td>
</tr>
<tr>
<td><strong>SYNRADE</strong> Synchrotron radiation spectral-angular densities</td>
<td>256</td>
<td></td>
</tr>
<tr>
<td><strong>TARGET</strong> Generate a secondary beam following target interaction</td>
<td>191</td>
<td></td>
</tr>
<tr>
<td><strong>TOSCA</strong> 2-D and 3-D Cartesian or cylindrical mesh field map</td>
<td>257</td>
<td></td>
</tr>
<tr>
<td><strong>TRANSMAT</strong> Matrix transfer</td>
<td>258</td>
<td></td>
</tr>
<tr>
<td><strong>TRAROT</strong> Translation-Rotation of the reference frame</td>
<td>259</td>
<td></td>
</tr>
<tr>
<td><strong>TWISS</strong> Calculation of periodic optical parameters</td>
<td>260</td>
<td></td>
</tr>
<tr>
<td><strong>UNDULATOR</strong> Undulator magnet</td>
<td>261</td>
<td></td>
</tr>
<tr>
<td><strong>UNIPOT</strong> Unipotential cylindrical electrostatic lens</td>
<td>262</td>
<td></td>
</tr>
<tr>
<td><strong>VENUS</strong> Simulation of a rectangular shape dipole magnet</td>
<td>263</td>
<td></td>
</tr>
<tr>
<td><strong>WIENFILT</strong> Wien filter</td>
<td>264</td>
<td></td>
</tr>
<tr>
<td><strong>YMY</strong> Reverse signs of Y and Z reference axes</td>
<td>265</td>
<td></td>
</tr>
</tbody>
</table>
# Optical elements versus keywords

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

<table>
<thead>
<tr>
<th>Element</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGS main magnet</td>
<td>AGSMM</td>
</tr>
<tr>
<td>Decapole</td>
<td>DECAPOLE, MULTIPOL</td>
</tr>
<tr>
<td>Dipole[s]</td>
<td>AIMANT, BEND, DIPOLE[S], DIPOLE-M, MULTIPOL, QUADISEX</td>
</tr>
<tr>
<td>Dodecapole</td>
<td>DODECAPPO, MULTIPOL</td>
</tr>
<tr>
<td>FFAG magnets</td>
<td>DIPOLES, FFAG, FFAG-SPI, MULTIPOL</td>
</tr>
<tr>
<td>Helical dipole</td>
<td>HELIX</td>
</tr>
<tr>
<td>Multipole</td>
<td>MULTIPOL, QUADISEX, SEXQUAD</td>
</tr>
<tr>
<td>Octupole</td>
<td>OCTUPOLE, MULTIPOL, QUADISEX, SEXQUAD</td>
</tr>
<tr>
<td>Quadrupole</td>
<td>QUADRUPOL, MULTIPOL, QUADISEX, SEXQUAD, AGSQUAD</td>
</tr>
<tr>
<td>Sextupole</td>
<td>SEXTUPOLE, MULTIPOL, QUADISEX, SEXQUAD</td>
</tr>
<tr>
<td>Skew multipoles</td>
<td>MULTIPOL</td>
</tr>
<tr>
<td>Solenoid</td>
<td>SOLENOID</td>
</tr>
<tr>
<td>Undulator</td>
<td>UNDULATOR</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Element</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-tube (bipotential) lens</td>
<td>EL2TUB</td>
</tr>
<tr>
<td>3-tube (unipotential) lens</td>
<td>UNIPOT</td>
</tr>
<tr>
<td>Decapole</td>
<td>ELMULT</td>
</tr>
<tr>
<td>Dipole</td>
<td>ELMULT</td>
</tr>
<tr>
<td>Dodecapole</td>
<td>ELMULT</td>
</tr>
<tr>
<td>Multipole</td>
<td>ELMULT</td>
</tr>
<tr>
<td>N-electrode mirror/lens, straight slits</td>
<td>ELMIR</td>
</tr>
<tr>
<td>N-electrode mirror/lens, circular slits</td>
<td>ELMIRC</td>
</tr>
<tr>
<td>Octupole</td>
<td>ELMULT</td>
</tr>
<tr>
<td>Quadrupole</td>
<td>ELMULT</td>
</tr>
<tr>
<td>R.F. (kick) cavity</td>
<td>CAVITE</td>
</tr>
<tr>
<td>Sextupole</td>
<td>ELMULT</td>
</tr>
<tr>
<td>Skew multipoles</td>
<td>ELMULT</td>
</tr>
</tbody>
</table>

### Using field maps

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

## ELECTRO-MAGNETIC ELEMENTS

<table>
<thead>
<tr>
<th>Element</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decapole</td>
<td>EBMULT</td>
</tr>
<tr>
<td>Dipole</td>
<td>EBMULT</td>
</tr>
<tr>
<td>Dodecapole</td>
<td>EBMULT</td>
</tr>
<tr>
<td>Multipole</td>
<td>EBMULT</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Octupole</td>
<td>EBMULT</td>
</tr>
<tr>
<td>Quadrupole</td>
<td>EBMULT</td>
</tr>
<tr>
<td>Sextupole</td>
<td>EBMULT</td>
</tr>
<tr>
<td>Skew multipoles</td>
<td>EBMULT</td>
</tr>
<tr>
<td>Wien filter</td>
<td>SEPARA, WIENFILT</td>
</tr>
</tbody>
</table>
INTRODUCTION

Hereafter 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  

$IL = 1, 2 \times 10^9$ : print field and coordinates along trajectories.  

0-2[$\times 10^9$]  I

MOD[MOD2], dL,  

$R_0, dB1, dB2, dB3$  

Type of magnet model 1 [type of back-leg winding model 2] ;  

2 no dim., cm.  3*no dim.  I, [I], 5*E

NBLW,  

NBLW times : NW, I  

Number of back-leg windings ; for each back-leg winding :  

$\leq 2, NBLW \times$  (any, Amp.)  I, NBLW \times (I, E)

Entrance face  

$X_E, \lambda_E, E_2, E_3$  

Integration zone ; fringe field extent :  

dipole fringe field extent = $\lambda_E$ ;  

quadrupole fringe field extent = $\lambda_E \times E_2$ ;  

sextupole fringe field extent = $\lambda_E \times E_3$ (sharp edge if field extent is zero)

NCE, $C_0 - C_5$  

same as QUADRUPO

Exit face  

$X_S, \lambda_S, S_2, S_3$  

Integration zone ; as for entrance

R1, R2, R3  

Skew angles of field components  

3*rad 10*E

XPAS  

Integration step

KPOS, XCE,  

YCE, ALE

KPOS=1 : element aligned, 2 : misaligned ;  

shifts, tilt (unused if KPOS=1).  

KPOS = 3 : effective only if $B_1 \neq 0$ :  

entrance and exit frames are shifted by YCE and tilted wrt. the magnet by an angle of  

• either ALE if ALE$\neq 0$  

• or 2 Arcsin($B_1 XL / 2 BORO$) if ALE=0

KPOS = 4 : same as KPOS = 3 however with possible X- or Y- or Z-misalignment or -rotation (under development)

1 $MOD=1$ : centered multipole model ; $MOD=2$ : long-shifted dipole model ; $MOD=3$ : short-shifted dipole model.

2 $MOD2 = 0$ (default) : user defined back-leg windings (defined in routine agsblw.f) ; $MOD2 = 1$ : 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 = 10 ; MM_B04CF : NBLW = 1, SIGN = -1.D0, NW = 10 ; MM_B05A : NBLW = 1, SIGN = 1.D0, NW = 10 ; MM_B06A : NBLW = 1, SIGN = 1.D0, NW = 12 ; MM_B07CD : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM_B08C : NBLW = 1, SIGN = 1.D0, NW = 5 ; MM_B09BF : NBLW = 1, SIGN = 1.D0, NW = 6 ; MM_B10B : NBLW = 1, SIGN = 1.D0, NW = 5 ; MM_L06A : NBLW = 1, SIGN = 1.D0, NW = 5 ; MM_L07C : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM_L12CF : NBLW = 1, SIGN = 1.D0, NW = 5 ; MM_L14C : NBLW = 1, SIGN = -1.D0, NW = 5 ; MM_L15A : 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_E08BD : NBLW = 1, SIGN = 1.D0, NW = 5 ; 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.

MOD2 = 1 : User defined - implementation to be completed.
**AGSQUAD**  
**AGS quadrupole**

<table>
<thead>
<tr>
<th><strong>IL</strong></th>
<th>$IL = 1, 2\times10^n$</th>
<th>Print field and coordinates along trajectories.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$XL, R_0, IW_1, IW_2, IW_3$</td>
<td>$2\times10^n$</td>
<td>0-2</td>
</tr>
<tr>
<td>$dIW_1, dIW_2, dIW_3$</td>
<td>$3\times10^n$</td>
<td>1</td>
</tr>
</tbody>
</table>

Relative error on currents:

| **dIW_1, dIW_2, dIW_3** | $3\times10^n$ | 3 |

**Entrance face**

<table>
<thead>
<tr>
<th>$X_E, \lambda_E$</th>
<th>Integration zone; fringe field extent.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(sharp edge if field extent is zero)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>NCE, C_0 - C_5</strong></th>
<th>Same as QUADRUPO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0-6, 6*no dim.</td>
</tr>
</tbody>
</table>

**Exit face**

<table>
<thead>
<tr>
<th>$X_S, \lambda_S$</th>
<th>Integration zone; as for entrance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2<em>cm, 9</em>no dim.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>NCS, C_0 - C_5</strong></th>
<th>0-6, 6*no dim.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>I, 6*E</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>R1</strong></th>
<th>Roll angle</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10*rad</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>XPAS</strong></th>
<th>Integration step</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>KPOS, XCE, YCE, ALE</strong></th>
<th>$KPOS=1$: element aligned, 2: misaligned;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>shifts, tilt (unused if $KPOS=1$)</td>
</tr>
</tbody>
</table>

|                | 1-2, 2*cm, rad | I, 3*E |

Keywords and input data formatting
AIMANT

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

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

**NFACE, IC, IL**
Number of field boundaries
- \( IC = 1, 2 \): print field map
- \( IL = 1, 2 \times 10^n \): print field and coordinates along trajectories.

**IAMAX, IRMAX**
Azimuthal and radial number of nodes of the mesh
\( \leq 400, \leq 10^4 \)

**B₀, N, B, G**
Field and field indices
kG, \( 3\times \text{no dim.} \)

**AT, ACENT, RM, RMIN, RMAX**
Mesh parameters: total angle of the map; azimuth for EFBs positioning; reference radius; minimum and maximum radii

**ENTRANCE FIELD BOUNDARY**

\( \lambda, \xi \)
Fringe field extent (normally \( \approx \) gap size); flag:
- \( \xi \geq 0 \): second order type fringe field with linear variation over distance \( \xi \)
- \( \xi = -1 \): exponential type fringe field:
  \[ F = (1 + \exp(P(s)))^{-1} \]
  \[ P(s) = C_0 + C_1 \left( \frac{s}{\lambda} \right) + C_2 \left( \frac{s}{\lambda} \right)^2 + \ldots + C_5 \left( \frac{s}{\lambda} \right)^5 \]

\( NC, C_0 - C_5, \text{shift} \)
NC = 1 + degree of \( P(s) \); \( C_0 \) to \( C_5 \): see above; EFB shift (ineffective if \( \xi \geq 0 \))

\( \omega^+, \theta, R_1, U_1, U_2, R_2 \)
Azimuth of entrance EFB with respect to ACENT; wedge angle of EFB; radii and linear extents of EFB (use \( |U_1,2| = \infty \) when \( R_{1,2} = \infty \))

(Note: \( \lambda = 0, \omega^+ = \text{ACENT and } \theta = 0 \) for sharp edge)

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

\( \lambda, \xi \)
Fringe field parameters

\( NC, C_0 - C_5, \text{shift} \)

\( \omega^-, \theta, R_1, U_1, U_2, R_2 \)
Positioning and shape of the exit EFB

(Note: \( \lambda = 0, \omega^- = -\mathcal{A}T+\text{ACENT and } \theta = 0 \) for sharp edge)
If NFACE = 3

LATERAL FIELD BOUNDARY
(See ENTRANCE FIELD BOUNDARY)
Next 3 records only if NFACE = 3

\( \lambda, \xi \)
Fringe field parameters

cm, (cm) 2*E

\( NC, C_0 - C_5, \) shift
0-6, 6*no dim., cm 1, 7*E

\( \omega, \theta, R_1, U_1, U_2, R_2, \)
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

-2 \(-6 \) or \( \geq 1 \) 1

If NBS = 0
Normal value. No other record required

If NBS = -2
The map is modified as follows :

\( R_0, \Delta B / B_0 \)
\( B \) transforms to \( B \cdot \left( 1 + \frac{\Delta B}{B_0} \frac{R-R_0}{R_{MAX}-R_{MIN}} \right) \)
cm, no dim. 2*E

If NBS = -1
the map is modified as follows :

\( \theta_0, \Delta B / B_0 \)
\( \theta \) transforms to \( \theta \cdot \left( 1 + \frac{\Delta B}{B_0} \frac{\theta-\theta_0}{\Delta \theta} \right) \)
deg, no dim. 2*E

If NBS \( \geq 1 \)
Introduction of NBS shims

For I = 1, NBS
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

\( IORDRE \)
Degree of interpolation polynomial :
2 = second degree, 9-point grid
25 = second degree, 25-point grid
4 = fourth degree, 25-point grid

I

XPAS
Integration step
cm E

KPOS
Positioning of the map, normally 2. Two options :
1-2 I

If KPOS = 2
Positioning as follows :
RE, TE, RS, TS
Radius and angle of reference, respectively,
at entrance and exit of the map.
cm, rad, cm, rad 4*E

If KPOS = 1
Automatic positioning of the map, by means of
reference relative momentum
no dim. E

\( DP \)
A : Parameters used to define the field map and geometrical boundaries.
B : Parameters used to define the field map and fringe fields.
Second order type fringe field.

Exponential type fringe field.
**AUTOREF**  Automatic transformation to a new reference frame

$I$  
1: Equivalent to \( \text{CHANGREF} (XCE = 0, YCE = Y(1), ALE = T(1)) \)

2: Equivalent to \( \text{CHANGREF} (XW, YW, T(1)), \) with \((XW, YW)\)
being the location of the intersection (waist) of particles 1, 4 and 5
(useful with \(\text{MATRIX}\), for automatic positioning of the first order focus)

3: Equivalent to \( \text{CHANGREF} (XW, YW, T(I1)) \), with \((XW, YW)\)
being the location of the intersection (waist) of particles \(I1, I2\) and \(I3\)
(for instance: \(I1 = \text{central trajectory}, I2\) and \(I3 = \text{paraxial trajectories}
that intersect at the first order focus)

<table>
<thead>
<tr>
<th>If (I = 3)</th>
<th>Next record only if (I = 3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I1, I2, I3)</td>
<td>Three particle numbers</td>
</tr>
</tbody>
</table>

If \(I = 3\) Next record only if \(I = 3\)

\(I1, I2, I3\) Three particle numbers

\(3*(1-\text{IMAX})\) \(3*I\)
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Units</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW, I</td>
<td>0/1: off/on; beam intensity.</td>
<td>0-2, Amp</td>
<td>I, E</td>
</tr>
<tr>
<td>(\alpha_Y, \beta_Y, \epsilon_Y,norm/\pi)</td>
<td>Beam parameters, horizontal.</td>
<td>- , m, m.rad</td>
<td>3*E</td>
</tr>
<tr>
<td>(\alpha_Z, \beta_Z, \epsilon_Z,norm/\pi)</td>
<td>Beam parameters, vertical.</td>
<td>- , m, m.rad</td>
<td>3*E</td>
</tr>
<tr>
<td>(\sigma_X, \sigma_{dp/p})</td>
<td>rms bunch length; rms momentum spread.</td>
<td>m, -</td>
<td>2*E</td>
</tr>
<tr>
<td>(C, \alpha)</td>
<td>Ring circumference; momentum compaction.</td>
<td>m, -</td>
<td>2*E</td>
</tr>
<tr>
<td>(Q_Y, Q_Z, Q_s)</td>
<td>Tunes, horizontal, vertical, synchrotron.</td>
<td>- , - , -</td>
<td>3*E</td>
</tr>
<tr>
<td>(A_Y, A_Z, A_X)</td>
<td>Amplitudes, horizontal, vertical, longitudinal.</td>
<td>- , - , -</td>
<td>3*E</td>
</tr>
</tbody>
</table>
**BEND**  
**Bending magnet, Cartesian frame**

- **IL**  
  \[ IL = 1, 2 \times 10^6 \] : print field and coordinates along trajectories.  
  \[ 0-2 \times 10^6 \]  
  \[ I \]

- **XL, Sk, B1**  
  Length ; skew angle ; field  
  \[ \text{cm, rad, kG} \]  
  \[ 3*E \]

- **Entrance face**:
  \[ X_E, \lambda_E, W_E \]  
  Integration zone extent ; fringe field extent (normally \( \approx \) gap height ; zero for sharp edge) ; wedge angle  
  \[ \text{cm, cm, rad} \]  
  \[ 3*E \]

- **N, C_0-C_5**  
  Unused ; fringe field coefficients :  
  \[ B(s) = B1 F(s) \]  
  \[ F(s) = 1/(1 + \exp(P(s))) \]  
  \[ P(s) = \sum_{i=0}^{5} C_i(s/\lambda)^i \]  
  \[ \text{unused, 6*no dim.} \]  
  \[ 1, 6*E \]

- **Exit face**:
  \[ X_S, \lambda_S, W_S \]  
  See entrance face  
  \[ \text{cm, cm, rad} \]  
  \[ 3*E \]

- **XPAS**  
  Integration step  
  \[ \text{cm} \]  
  \[ E \]

- **KPOS, XCE, YCE, ALE**  
  \( KPOS = 1 \) : element aligned, \( 2 \) : misaligned ; shifts, tilt (unused if \( KPOS = 1 \))  
  \[ 1-2, 2*\text{cm, rad} \]  
  \[ 1, 3*E \]

- **KPOS = 3**  
  entrance and exit frames are shifted by \( YCE \)  
  and tilted wrt. the magnet by an angle of  
  - either \( ALE \neq 0 \)  
  - or \( 2 \arcsin(B1XL / 2B1ORO) \) if \( ALE = 0 \)

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 \((O, X, Y, Z)\)
**Keywords and input data formatting**

**BINARY**  
**BINARY/FORMATTED** data converter

\( NF[J], \ NCol, \ NHDR \)

Number of files to convert [READ format type, see below], of data columns, of header lines.

\( \leq 20, \geq 1, 0 \rightarrow 9 \ 3^*1 \)

**The next \( NF \) lines :**

\( FNAME \)

Name of the file to be converted. File content is assumed binary  
"iff" name begins with "B_" or "b_.", assumed formatted otherwise.

A80

**READ format :**

If FRM not given  
Format is "*"

If FRM=1  
Format is "1X,7E11.*"
**BREVOL**

1-D uniform mesh magnetic field map

X-axis cylindrical symmetry is assumed

| **IC, IL** | \( IC = 1, 2 \): print the map \( IL = 1, 2 \times 10^n \): print field and coordinates along trajectories. | 0-2, 0-2[\times 10^n] | 2*I |
| **BNORM, XN** | Field and X-coordinate normalization coeff. | 2*no dim. | 2*E |
| **TITL** | Title. Start with “FLIP” to get field map X-flipped. | A80 |
| **IX** | Number of longitudinal nodes of the map | \( \leq 400 \) | I |
| **FNAME [, SUM]** | File name | A80 |
| **ID, A, B, C** | Integration boundary. Ineffective when \( ID = 0 \). | \( \geq -1, 2 \times \text{no dim.} \) | I,3*E |
| \( [, A', B', C', B''', 	ext{etc.}, 	ext{if } ID \geq 2] \) | as for CARTEMES \( \text{cm} [,2 \times \text{no dim.}, \text{cm, etc.}] \) | [.3*E,etc.] |
| **IORDRE** | Unused | 2, 25 or 4 | I |
| **XPAS** | Integration step | cm | E |
| **KPOS, XCE, YCE, ALE** | \( KPOS=1 \): element aligned, \( 2 \): misaligned; shifts, tilt (unused if \( KPOS=1 \)) | \( 1-2, 2 \times \text{cm, rad} \) | I, 3*E |

---

1. **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.’.

2. Summing (summing) field maps is possible. To do so, pile up file names with ‘SUM’ following each name but the last one, e.g., in the following example, 3 field maps are read and summed:

- myMapFile1 SUM
- myMapFile2 SUM
- myMapFile3 SUM

(all maps must all have their mesh defined in identical coordinate frame).
Keywords and input data formatting

CARTEMES

**2-D Cartesian uniform mesh magnetic field map**
mid-plane symmetry is assumed

**IC, IL**

IC = 1, 2 : print the map
IL = 1, 2 [×10^n] : print field and coordinates along trajectories.

**BNORM, XN, YN**

Field and X-, Y-coordinate normalization coeffs.

**TITL**

Title. Start with “FLIP” to get field map X-flipped.

**IX, JY**

Number of longitudinal (IX) and transverse (JY) nodes of the map ≤ 400, ≤ 200

**FNAME**

File name

**ID, A, B, C**

Integration boundary. Normally ID = 0.

**[A', B', C', A'']**

ID = -1 : integration in the map begins at entrance boundary defined by \(AX + BY + C = 0\).

ID = 1 : integration in the map is terminated at exit boundary defined by \(AX + BY + C = 0\).

ID ≥ 2 : entrance (A, B, C) and up to ID - 1 exit (A', B', C', A'', B'', etc.) boundaries

**IORDRE**

Degree of interpolation polynomial (see DIPOLE-M)

**XPAS**

Integration step cm

**KPOS, XCE, YCE, ALE**

KPOS=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if KPOS=1)

2 FNAME (e.g., spec2.map) contains the field data. These must be formatted according to the following FORTRAN sequence:

```fortran
OPEN (UNIT = NL, FILE = FNAME, STATUS = ‘OLD’ [,FORM=’UNFORMATTED’])
IF (BINARY) THEN
  READ(NL) (Y(J), J=1, JY)
ELSE
  READ(NL100) (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(NL101) X(I), (BMES(I,J), J=1, JY)
  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 $AX + BY + C' = 0$; 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.
CAVITE

Accelerating cavity
\( \Delta W = qV \sin(2\pi fh \Delta t + \varphi_s) \) and other voltage and frequency laws.

\( \text{IOPT}[i] \)
Option. \( i = 1 \) causes info output into zgoubi.CAVITE.out

If \( \text{IOPT}=0 \)
Element inactive

If \( \text{IOPT}=1 \)

\( f_{RF} \) follows the timing law given by \text{SCALING}

If \( \text{IOPT}=2 \)

\( f_{RF} \) follows \( \Delta W_s = qV \sin \phi_s \)

If \( \text{IOPT}=3 \)

No synchrotron motion: \( \Delta W = qV \sin \phi_s \)

If \( \text{IOPT}=6 \)

Read RF frequency and/or phase law from external file, “zgoubi.freqLaw.In”.

If \( \text{IOPT}=7 \)

Quasi- or isochronous acceleration.

1 Use \text{PARTICUL} to declare mass and charge.

2 For ramping the R.F. frequency following \( B_\rho(t) \), use \text{SCALING}, with family \text{CAVITE}.
Keywords and input data formatting

CHAMBR

Long transverse aperture limitation

IA

0: element inactive
1: (re)definition of the aperture
2: stop testing and reset counters, print information on stopped particles.

IFORM[J], C1, C2, C3, C4

IFORM = 1: rectangular aperture:

IFORM = 2: elliptical aperture.

J = 0, default: opening is \( \pm YL = \pm C1, \pm ZL = \pm C2 \), centered at \( YC = C3, ZC = C4 \).

J = 1: opening is in \( Y : [C1, C2] \), in \( Z : [C3, C4] \).

1 Any particle out of limits is stopped.

2 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, \( YC \approx RM \)).
Keywords and input data formatting

**CHANGREF**  Transformation to a new reference frame

**“Old Style”** (Figure below):

\( XCE, YCE, ALE \) Longitudinal and transverse shifts, followed by \( Z \)-axis rotation

\( 2^* \text{cm}, \text{deg} \quad 3^*E \)

**“New Style”** (example below). In an arbitrary order, up to 9 occurrences of:

\( XS 'val', YS 'val', ZS 'val', \)

\( XR 'val', YR 'val', ZR 'val' \)

\( \text{cm or deg} \quad \text{up to 9}^* (A2,E) \)

---

**Example:**

```
Using CHANGREF "New Style
OBJET"
51.71103865921708 Electron, Ekin=15MeV.
2
1 1
2. 0. 0.0 0.0 0.0 1. 'R'
1 1 1 1 1 1
'MARKER' BEG .plt -> list into zgoubi.plt.
'DRIFT'
10.
'CHANGREF'
ZR -6.34165 YS 1. First half Z-rotate, Next Y-shift.
'CHANGREF'
0. 1. 0.
'MULTIPOL' Combined function multipole, dipole + quadrupole.
2
5 10. 2.064995867082342 2. 0. 0. 0. 0. 0. 0. 0.
0 0 5. 1.1 1.00 1.00 1.00 1.00 1.1 1. 1. 1.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.
0 0 5. 1.1 1.00 1.00 1.00 1.00 1.1 1. 1. 1.
4 .1455 2.2670 -.6395 1.1558 0. 0. 0.
0 0 0 0 0 0 0 0 0 0 0 0 0 0
.1 step size
1 0. 0. 0.
'CHANGREF'
YS -1. ZR -6.341 First Y-shift back, next half Z-rotate.
'DRIFT'
10.
'FAISCEAU'
'END'
```

---

Parameters in the CHANGREF procedure.
Keywords and input data formatting

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 

\[ 5 \times \frac{MeV}{c^2}, 2^\circ \text{deg}, 7^\circ \text{E} \]

\[ NT, NP \]

Number of samples in \( T \) and \( P \) coordinates after CIBLE

\[ 2^*1 \]

\[ TS, PS, DT \]

Sample step sizes; tilt angle

\[ 3^*\text{mrad}, 3^*\text{E} \]

\[ BORO \]

New reference rigidity after CIBLE

\[ \text{kG.cm} \]

\[ E \]

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

IA
0 : element inactive
1 : element active
2 : element active and print information on stopped particles

Physical-space collimation
IFORM[J], C1, C2, C3, C4

IFORM = 1 : rectangular aperture ;
J = 0, default : opening is \( \pm YL = \pm C1, \pm ZL = \pm C2 \),
centered at \( YC = C3, ZC = C4 \).
J = 1 : opening is, in Y : \([C1, C2]\), in Z : \([C3, C4]\)

Longitudinal collimation
IFORM[J], Hmin, Hmax

IFORM = 6 or 7 for horizontal variable resp by S or Time,
J=1 or 2 for vertical variable resp by \( 1+dp/p \), kinetic-E (MeV) ;
horizontal and vertical limits

Phase-space collimation
IFORM, \( \alpha, \beta, \epsilon/\pi, N_{\sigma} \)

IFORM = 11, 14 : horizontal collimation ; horizontal ellipse parameters (unused if 14) \(^2\), emittance, cut-off
IFORM = 12, 15 : vertical collimation ; vertical ellipse parameters (unused if 15) \(^2\), emittance, cut-off
IFORM = 13, 16 : longitudinal collimation ; to be implemented

Any particle out of limits is stopped.

The rejection boundary is the \( rms \) ellipse matched to the particle distribution.
### Decapole Magnet

**IL**

\[ IL = 1, 2 \times 10^n \] : print field and coordinates along trajectories.

<table>
<thead>
<tr>
<th>Format</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Int.</td>
<td>0-2 \times 10^n</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
</tr>
</tbody>
</table>

**XL, R_0, B_0**

Length : radius and field at pole tip

<table>
<thead>
<tr>
<th>Format</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Int.</td>
<td>2 cm, kG</td>
</tr>
<tr>
<td>E</td>
<td>3 E</td>
</tr>
</tbody>
</table>

**X_E, \lambda_E**

Integration zone extent : fringe field extent \((\lesssim 2R_0, \lambda_E = 0 \text{ for sharp edge})\)

<table>
<thead>
<tr>
<th>Format</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Int.</td>
<td>2 cm</td>
</tr>
<tr>
<td>E</td>
<td>2 E</td>
</tr>
</tbody>
</table>

**NCE, C_0 - C_5**

\(NCE = \text{unused}\)

\(C_0 - C_5 = \text{Fringe field coefficients such that}\)

\[ G(s) = G_0 / (1 + \exp P(s)), \quad G_0 = B_0 / R_0^4 \]

\[ P(s) = \sum_{i=0}^{5} C_i (s/\lambda)^i \]

<table>
<thead>
<tr>
<th>Format</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>I, 6 E</td>
</tr>
<tr>
<td>E</td>
<td>6 no dim.</td>
</tr>
</tbody>
</table>

**X_S, \lambda_S**

Exit face : see entrance face

<table>
<thead>
<tr>
<th>Format</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Int.</td>
<td>2 cm</td>
</tr>
<tr>
<td>E</td>
<td>2 E</td>
</tr>
</tbody>
</table>

**NCS, C_0 - C_5**

<table>
<thead>
<tr>
<th>Format</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>I, 6 E</td>
</tr>
<tr>
<td>E</td>
<td>6 no dim.</td>
</tr>
</tbody>
</table>

**XPAS**

Integration step

<table>
<thead>
<tr>
<th>Format</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm</td>
<td>E</td>
</tr>
</tbody>
</table>

**KPOS, XCE, YCE, ALE**

\(KPOS = 1 : \text{element aligned, 2 : misaligned}\)

shifts, tilt (unused if \(KPOS = 1\))

<table>
<thead>
<tr>
<th>Format</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>I, 3 E</td>
</tr>
<tr>
<td>E</td>
<td>1-2, 2 cm, rad</td>
</tr>
</tbody>
</table>

---

![Diagram of Decapole Magnet](image-url)
DIPOLE

Dipole magnet, polar frame

\[ B_Z = FB_0 \left( 1 + N \left( \frac{R-RM}{RM} \right) + B \left( \frac{R-RM}{RM} \right)^2 + G \left( \frac{R-RM}{RM} \right)^3 \right) \]

**IL**

**IL** = 1, 2 \( \times 10^n \): print field and coordinates along trajectories. 0-2 \( \times 10^n \) I

**AT, RM**

Total angular extent of the dipole; reference radius  deg, cm 2*E

**ACENT, B0, N, B, G**

Azimuth for positioning of EFBs; field and field indices  deg., kG, 3*no dim. 5*E

### ENTRANCE FIELD BOUNDARY

\( \lambda, \xi \)

Fringe field extent (normally \( \simeq \) gap size); unused. cm, unused 2*E

Exponential type fringe field \( F = 1 / (1 + \exp(P(s))) \)

with \( P(s) = C_0 + C_1 \left( \frac{\xi}{\lambda} \right) + C_2 \left( \frac{\xi}{\lambda} \right)^2 + \ldots + C_5 \left( \frac{\xi}{\lambda} \right)^5 \)

**NC, C0 – C5, shift**

Unused; \( C_0 \) to \( C_5 \): see above; EFB shift 0-6, 6*no dim., cm 1,7*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 \( |U_{1,2}| = \infty \) when \( R_{1,2} = \infty \))

2*deg, 4*cm 6*E

### EXIT FIELD BOUNDARY

(See ENTRANCE FIELD BOUNDARY)

\( \lambda, \xi \)

Fringe field parameters cm, unused 2*E

**NC, C0 – C5, shift**

0-6, 6*no dim., cm 1, 7*E

\( \omega^-, \theta, R_1, U_1, U_2, R_2 \)

Positioning and shape of the exit EFB 2*deg, 4*cm 6*E

### LATERAL FIELD BOUNDARY

(See ENTRANCE FIELD BOUNDARY)

\( \lambda, \xi \)

LATERAL EFB is inhibited if \( \xi = 0 \) cm, unused 2*E

**NC, C0 – C5, shift**

0-6, 6*no dim., cm 1, 7*E

\( \omega^-, \theta, R_1, U_1, U_2, R_2, RM \)

Positioning and shape of the exit EFB 2*deg, 5*cm 7*E

**IORDRE, Resol**

Degree of interpolation polynomial:

2 = second degree, 9-point grid

25 = second degree, 25-point grid

4 = fourth degree, 25-point grid; resolution of flying mesh is XPA/Resol

2, 25 or 4 ; no dim. I, E

**XPAS**

Integration step cm E

**KPOS**

Positioning of the map, normally 2. Two options:

1-2 I

**If KPOS = 2**

RE, TE, RS, TS

Radius and angle of reference, respectively, at entrance and exit of the map. cm, rad, cm, rad 4*E

**If KPOS = 1**

**DP**

Automatic positioning of the map, by means of reference relative momentum no dim. E
DIPOL-M

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

\[ B_Z = \mathcal{F} B_0 \left( 1 + N \left( \frac{R - R_{M3}}{R_{M3}} \right) + B \left( \frac{R - R_{M3}}{R_{M3}} \right)^2 + G \left( \frac{R - R_{M3}}{R_{M3}} \right)^3 \right) \]

**NFACE, IC, IL**

Number of field boundaries

\[ IC = 1, 2 : \text{print field map} \]

\[ IL = 1, 2 : \text{print field and coordinates on trajectories} \]

**IAMAX, IRMAX**

Azimuthal and radial number of nodes of the mesh

\[ \leq 400, \leq 200 \]

**B0, N, B, G**

Field and field indices

\[ \text{kg}, \text{3*no dim.} \]

**AT, ACENT, RM, RMIN, RMAX**

Mesh parameters:

- total angle of the map
- azimuth for positioning of EFBs
- reference radius
- minimum and maximum radii

**ENTRANCE FIELD BOUNDARY**

\( \lambda, \xi \)

Fringe field extent (normally \( \simeq \) gap size) ; unused.

\[ \text{cm, unused} \]

Exponential type fringe field

\[ F = \frac{1}{\left(1 + \exp(P(s))\right)} \]

with

\[ P(s) = C_0 + C_1 \left( \frac{s}{\lambda} \right) + C_2 \left( \frac{s}{\lambda} \right)^2 + \ldots + C_5 \left( \frac{s}{\lambda} \right)^5 \]

\[ NC, C_0 - C_5, \text{shift} \]

Unsed ; \( C_0 \) to \( C_5 \) : see above ; EFB shift

\[ 0-6, 6*\text{nodim., cm} \]

\( \omega^+, \theta, R_1, U_1, U_2, R_2 \)

Azimuth of entrance EFB with respect to ACENT ; wedge angle of EFB ; radii and linear extents of EFB (use \( |U_{1,2}| = \infty \) when \( R_{1,2} = \infty \))

\[ 2^\circ \text{deg}, 3^\circ \text{cm} \]

(Note : \( \lambda = 0, \omega^+ = \text{ACENT} \) and \( \theta = 0 \) for sharp edge)

**EXIT FIELD BOUNDARY**

(See ENTRANCE FIELD BOUNDARY)

\[ \lambda, \xi \]

Fringe field parameters

\[ \text{cm, unused} \]

\[ NC, C_0 - C_5, \text{shift} \]

0-6, 6*nodim., cm

\[ \omega^-, \theta, R_1, U_1, U_2, R_2 \]

Positioning and shape of the exit EFB

\[ 2^\circ \text{deg}, 4^\circ \text{cm} \]

(Note : \( \lambda = 0, \omega^- = -AT + \text{ACENT} \) and \( \theta = 0 \) for sharp edge)

**IF NFACE = 3**

(See ENTRANCE FIELD BOUNDARY)

Next 3 records only if NFACE = 3

\[ \lambda, \xi \]

Fringe field parameters

\[ \text{cm, (cm)} \]

\[ NC, C_0 - C_5, \text{shift} \]

0-6, 6*nodim., cm

\[ \omega^-, \theta, R_1, U_1, U_2, R_2, \text{RM3} \]

Positioning and shape of the lateral EFB ; RM3 is the radial position on azimuth ACENT

\[ 2^\circ \text{deg}, 5^\circ \text{cm} \]

**NBS**

Option index for perturbations to the field map

\[ \text{normally} \, 0 \]

**IF NBS = 0**

Normal value. No other record required

**IF NBS = -2**

The map is modified as follows:

\[ R_0, \Delta B/B_0 \]

\[ B \text{ transforms to } B * \left( 1 + \frac{\Delta B}{B_0} \frac{R - R_{M3}}{R_{M3} - R_{MIN}} \right) \]

\[ \text{cm, no dim.} \]

**IF NBS = -1**

The map is modified as follows:

\[ \theta_0, \Delta B/B_0 \]

\[ B \text{ transforms to } B * \left( 1 + \frac{\Delta B}{B_0} \frac{\theta - \theta_0}{\Delta T} \right) \]

\[ \text{deg, no dim.} \]
If NBS $\geq 1$  
Introduction of NBS shims

For I = 1, NBS  
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

$IORDRE$  
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  
1

$XPAS$  
Integration step  
cm  
E

$KPOS$  
Positioning of the map, normally 2. Two options :  
1-2  
I

If KPOS = 2  
Positioning as follows :  
Radius and angle of reference, respectively,  
at entrance and exit of the map.  
5*E

If KPOS = 1  
Automatic positioning of the map, by means of  
reference relative momentum  
no dim.  
E

$DP$  
Reference relative momentum  
no dim.  
E
Dipoles

Dipole magnet $N$-tuple, polar frame

(i) $B_Z = \sum_{i=1}^{N_i} B_{Z0,i} \mathcal{F}_i(R, \theta) \left(1 + b_1(R - R_{M,i})/R_{M,i} + b_2(R - R_{M,i})^2/R_{M,i}^2 + \ldots\right)$

(ii) $B_Z = B_{Z0,i} + \sum_{i=1}^{N_i} \mathcal{F}_i(R, \theta) \left(b_1(R - R_{M,i}) + b_2(R - R_{M,i})^2 + \ldots\right)$

$I_L$ $I_L = 1, 2[\times 10^6]$ : print field and coordinates along trajectories. 0-2[\times 10^6] 1

$N, AT, RM$ Number of magnets in the $N$-tuple ; total angular extent of the dipole ; reference radius no dim., deg, cm 1, 2*E

Repeat $N$ times the following sequence

$ACN, \delta RM, B_0,$ $ind, b_i, (i = 1, ind)$ Positioning of EFBs : azimuth, $RM_i = RM + \delta RM$ ; field ; number of, and field coefficients deg., cm, kG, (ind + 1)*no dim. $ind*E$

$g_0, \kappa$ Fringe field extent ($g = g_0 (RM/R)^\omega$) cm, no dim. 2*E

Exponential type fringe field $F = 1 / (1 + \exp(P(s)))$

with $P(s) = C_0 + C_1 (s/g) + C_2(s/g)^2 + \ldots + C_5(s/g)^5$

$NC, C_0 - C_5, \text{shift}$ Unused ; $C_0$ to $C_5$ : see above ; EFB shift 0-6, 6*no dim., cm 1,7*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 $|U_{1,2}| = \infty$ when $R_{1,2} = \infty$) 2*deg, 4*cm 6*E

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

EXIT FIELD BOUNDARY

(See ENTRANCE FIELD BOUNDARY)

$g_0, \kappa$ 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 : $g_0 = 0, \omega^- = -AT^+ ACENT, \theta = 0$ and KIRD=0 for sharp edge)

LATERAL FIELD BOUNDARY to be implemented - following data not used

$g_0, \kappa$ cm, no dim. 2*E

$NC, C_0 - C_5, \text{shift}$ 0-6, 6*no dim., cm 1,7*E

$\omega^-, \theta, R_1, U_1, U_2, R_2, R_3$ 2*deg, 5*cm 7*E

End of repeat

1 Non-zero $\delta RM$ requires KIRD= 2, 4 or 25.
**KIRD[n], Resol**

KIRD=0: analytical computation of field derivatives;
n=0: default, $B_Z$ formula (i) above, n=1: $B_Z$ formula (ii).

Resol = 2/4 for 2nd/4th order field derivatives computation

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

**XPAS**
Integration step cm E

**KPOS**
Positioning of the magnet, normally 2. Two options:

**If KPOS = 2**

$RE, TE, RS, TS$
Radius and angle of reference, respectively, at entrance and exit of the magnet cm, rad, cm, rad 4*E

**If KPOS = 1**

$DP$
Automatic positioning of the magnet, by means of reference relative momentum no dim. E
DODECAPO  |  Dodecapole magnet

**IL**  
*IL = 1, 2[×10ⁿ]*: print field and coordinates along trajectories.  
0-2[×10ⁿ]  |  I

**XL, R₀, B₀**  
Length: radius and field at pole tip  
2*cm, kG  |  3*E

**XE, λE**  
Integration zone extent; fringe field extent (< 2R₀, λE = 0 for sharp edge)  
2*cm  |  2*E

**NCE, C₀ − C₅**  
*NCE = unused*  
6*no dim.  |  I, 6*E

**Xₐ, λₐ**  
Exit face: see entrance face  
2*cm  |  2*E

**NCS, C₀ − C₅**  
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 (unused if KPOS=1)  
1-2, 2*cm, rad  |  I, 3*E

---

Keywords and input data formatting
DRIFT, ESL

Field free drift space

\[ XL \]

length cm E
EBMULT  

**Electro-magnetic multipole**

\[ IL = 1, 2 \times 10^9 \] : print field and coordinates along trajectories.

**Electric poles**

\[ X_L, R_0, E_1, E_2, ..., E_{10} \]: Length of element ; radius at pole tip ; field at pole tip for dipole, quadrupole, ..., 20-pole electric components

**Entrance face**

\[ X_E, \lambda_E, E_2, ..., E_{10} \]: Integration zone ; fringe field extent :
dipole fringe field extent = \( \lambda_E \);
quadrupole fringe field extent = \( \lambda_E \times E_2 \);
... 20-pole fringe field extent = \( \lambda_E \times E_{10} \)
(for any component : sharp edge if field extent is zero)

\[ NCE, C_0 - C_5 \]: same as QUADRUPO

**Exit face**

\[ X_S, \lambda_S, S_2, ..., S_{10} \]: Integration zone ; as for entrance

\[ NCS, C_0 - C_5 \]: 0-6, 6*no dim. 1,6*E

\[ R_1, R_2, R_3, ..., R_{10} \]: Skew angles of electric field components

\[ X_L, R_0, B_1, B_2, ..., B_{10} \]: Length of element ; radius at pole tip ; field at pole tip for dipole, quadrupole, ..., 20-pole magnetic components

**Entrance face**

\[ X_E, \lambda_E, E_2, ..., E_{10} \]: Integration zone ; fringe field extent :
dipole fringe field extent = \( \lambda_E \);
quadrupole fringe field extent = \( \lambda_E \times E_2 \);
... 20-pole fringe field extent = \( \lambda_E \times E_{10} \)
(for any component : sharp edge if field extent is zero)

\[ NCE, C_0 - C_5 \]: same as QUADRUPO

**Magnetic poles**

\[ X_L, R_0, B_1, B_2, ..., B_{10} \]: Length of element ; radius at pole tip ; field at pole tip for dipole, quadrupole, ..., 20-pole magnetic components

\[ X_E, \lambda_E, E_2, ..., E_{10} \]: Integration zone ; fringe field extent :
dipole fringe field extent = \( \lambda_E \);
quadrupole fringe field extent = \( \lambda_E \times E_2 \);
... 20-pole fringe field extent = \( \lambda_E \times E_{10} \)
(for any component : sharp edge if field extent is zero)

\[ NCE, C_0 - C_5 \]: same as QUADRUPO

---

1 Use PARTICUL to declare mass and charge.
**Exit face**

- Integration zone: as for entrance

**NCS, C₀ = C₅**

- Skew angles of magnetic field components

**XPAS**

- Integration step

**KPOS, XCE, YCE, ALE**

  - KPOS=1: element aligned, 2: misaligned
  - shifts, tilt (unused if KPOS=1)

---

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xₛ, λₛ, S₂, ..., S₁₀</td>
<td>Integration zone</td>
<td>2<em>cm, 9</em>no dim.</td>
</tr>
<tr>
<td>NCS, C₀ − C₅</td>
<td>Skew angles of magnetic field components</td>
<td>0-6, 6*no dim.</td>
</tr>
<tr>
<td>R₁, R₂, R₃, ..., R₁₀</td>
<td>Integration step</td>
<td>10*rad</td>
</tr>
<tr>
<td>XPAS</td>
<td>Integration step</td>
<td>cm</td>
</tr>
<tr>
<td>KPOS, XCE, YCE, ALE</td>
<td>Integration step</td>
<td>1-2, 2*cm, rad</td>
</tr>
</tbody>
</table>

**Diagram:**

- Electrode
- Magnetic pole
- 7
- N
- Φ₂
- R₀
- Φ₁
- Φ₂
- Φ₂
- N
EL2TUB 1

Two-tube electrostatic lens

**IL**

$IL = 1, 2 \times 10^n$ : print field and coordinates along trajectories. \[0-2 \times 10^n\] I

**$X_1, D, X_2, R_0$**

Length of first tube ; distance between tubes ;
length of second tube ; inner radius

3*m 4*E

**$V_1, V_2$**

Potentials

2*V 2*E

**XPAS**

Integration step

cm E

**KPOS, XCE, YCE, ALE**

KPOS=1 : element aligned, 2 : misaligned ;
shifts, tilt (unused if KPOS=1)

1-2, 2*cm, I, 3*E rad

---

Two-electrode cylindrical electric lens.

1 Use PARTICUL to declare mass and charge.
**Keywords and input data formatting**

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

**IL**  
$IL = 1, 2\times10^9$ : print field and coordinates along trajectories.  
0-2$\times10^9$  
I

**N, L1, ..., 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, ..., 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 (unused if $KPOS=1$); $3$: 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**

\( IL = 1, 2 \times 10^6 \) : print field and coordinates along trajectories.

**R1, R2, AT, D**

Radius of first and second slits; total deviation
angle; gap

**V − VA, VB − V**

Potential difference

**XPAS**

Integration step

**KPOS**

Normally \( KPOS = 2 \) for positioning;

**RE, TE, RS, TS**

Radius and angle at respectively entrance and exit.

Electrostatic N-electrode mirror/lens, circular slits, in the case \( N = 3 \), in horizontal mirror mode.
ELMULT ¹ Electric multipole

**IL**

\[ IL = 1, 2 \times 10^n \] : print field and coordinates along trajectories. 0-2\[ \times 10^n \] I

**XL, R₀, E₁, E₂, ..., E₁₀**
Length of element ; radius at pole tip ; field at pole tip for dipole, quadrupole, ..., dodecapole components

| 2*cm, 10*V/m | 12*E |

**Entrance face**

\[ X_E, \lambda_E, E_2, ..., E_{10} \]

Integration zone ; fringe field extent ;
dipole fringe field extent = \( \lambda_E \); quadrupole fringe field extent = \( \lambda_E \times E_2 \);
...
20-pole fringe field extent = \( \lambda_E \times E_{10} \)
(sharp edge if field extent is zero)

| 2*cm, 9*no dim. | 11*E |

**NCE, C₀ - C₅**
same as QUADRUPO

| 0-6, 6*no dim. | 1, 6*E |

**Exit face**

\[ X_S, \lambda_S, S_2, ..., S_{10} \]
Integration zone ; as for entrance

| 2*cm, 9*no dim. | 11*E |

**NCS, C₀ - C₅**

| 0-6, 6*no dim. | 1, 6*E |

**R₁, R₂, R₃, ..., R₁₀**
Skew angles of field components

| 10*rad | 10*E |

**XPAS**
Integration step

| cm | E |

**KPOS, XCE, YCE, ALE**

\( KPOS=1 \) : element aligned, \( 2 \) : misaligned ;
shifts, tilt (unused if \( KPOS=1 \))

| 1-2, 2*cm, rad | 1, 3*E |

¹ Use PARTICUL to declare mass and charge.
**ELREVOL**  

1-D uniform mesh electric field map  
X-axis cylindrical symmetry is assumed

**IC, IL**  
\[ IC = 1, 2 : \text{print the map} \]  
\[ IL = 1, 2 \times 10^n : \text{print field and coordinates along trajectories.} \]

**ENORM, X-NORM**  
Field and X-coordinate normalization coeff.  
\[ 2^{*}\text{no dim.} \]

**TITL**  
Title. Start with “FLIP” to get field map X-flipped.  
\[ A^{80} \]

**IX**  
Number of longitudinal nodes of the map  
\[ \leq 400 \]

**FNAME**  
File name  
\[ A^{80} \]

**ID, A, B, C**  
Integration boundary. Ineffective when \( ID = 0 \).  
\[ ID = -1, 1 \text{ or } \geq 2 : \text{as for CARTEMES} \]

**IORDRE**  
Unused  
\[ 2, 25 \text{ or } 4 \]

**XPAS**  
Integration step  
\[ \text{cm} \]

**KPOS, XCE, YCE, ALE**  
\( KPOS=1 \) : element aligned, 2 : misaligned ;  
shifts, tilt (unused if \( KPOS=1 \))  
\[ 1-2, 2^{*}\text{cm, rad} \]

---

1 Use **PARTICUL** to declare mass and charge.

2 **FNAME** (e.g., e-lens.map) contains the field data. These must be formatted according to the following **FORTRAN** sequence:

```fortran
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.’
EMMA 2-D Cartesian or cylindrical mesh field map for EMMA FFAG

**IC, IL**
See CARTEMES 0-2, 0-2\(\times 10^n\) 2*I

**BNORM, XN, YN, ZN**
Field and X-, Y-, Z-coordinate normalization coefficients 4*no dim. 4*E

**TITL**
Title. Start with “FLIP” to get field map X-flipped A80

**IX, IY, IZ, MOD[i]**
Number of nodes of the mesh in the X, Y and Z directions, \(IZ = 1\) for single 2-D map;
\(MOD:\) operational and map FORMAT reading mode 1
\(MOD \leq 19:\) Cartesian mesh;
\(MOD \geq 20:\) cylindrical mesh;
\(i,\) optional, tells the reading FORMAT, default is ‘*’.

**FNAME** 1
Names of the NF 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, interpolated 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...

**ID, A, B, C**
Integration boundary. Ineffective when \(ID = 0\). \(I, 2*\)no dim. 1,3*E

\([A', B', C', B'', etc., if \(ID \geq 2\)]\)

**IORDRE**
If \(IZ = 1\) : as in CARTEMES 2, 25 or 4 I
If \(IZ \neq 1\) : unused

**XPAS**
Integration step cm \([2*no dim., [3*E, etc.]]\)

**KPOS, XCE, YCE, ALE**
\(KPOS=1:\) element aligned, 2 : misaligned ; 1-2, 2*cm, rad I, 3*E
Shifts, tilt (unused if \(KPOS=1\))

---

1 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.

**FAISCNL**  
**Store particle coordinates in file FNAME**

- **FNAME**: Name of storage file  
  (e.g., zgoubi.fai, or b2goubi.fai for binary storage).

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

- **FNAME**: Name of storage file (e.g. zgoubi.fai) [, label(s) of the element(s) at the exit of which the store occurs (10 labels maximum)]. If either FNAME or first LABEL is 'none' then no storage occurs. Store occurs at all elements if first LABEL is 'all' or 'ALL'.

- **IP**: Store every IP other pass (when using REBELOTE with NPASS \( \geq IP - 1 \)).

---

1 Stored data can be read back from FNAME using OBJET, KOBJ = 3.
**FFAG**

**UNDER DEVELOPMENT**

\[ B_Z = \sum_{i=1}^{N} B_{Z_0,i} F_i(R, \theta) \left( \frac{R_{M,i}}{R} \right)^K_i \]

**IL**

IL = 1, 2[\times 10^n] : print field and coordinates along trajectories. 0-2[\times 10^n] 1

**N, AT, RM**

Number of dipoles in the FFAG N-tuple ;
total angular extent of the dipole ; reference radius

deg, cm 1, 2*E

Repeat N times the following sequence

ACN, δRM, B\_Z0, K

Azimuth for dipole positioning ; \( R_{M,i} = R + \delta RM \) ; field at \( R_{M,i} \) ; index
deg, cm, kG, 4*E

**ENTRANCE FIELD BOUNDARY**

\( g_0, \kappa \)

Fringe field extent \((g = g_0 \left( \frac{RM}{R} \right)^\kappa)\)

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 1,7*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 \(| U_{1,2} | = \infty \) when \( R_{1,2} = \infty \))

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

**EXIT FIELD BOUNDARY**

(See ENTRANCE FIELD BOUNDARY)

\( g_0, \kappa \)

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

(Note : \( g_0 = 0, \omega^- = -AT + ACENT, \theta = 0 \) 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**

KIRD=0 : analytical computation of field derivatives ;
Resol = 2/4 for 2nd/4th order field derivatives computation
KIRD2, 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 ; I, E

**XPAS**

Integration step
cm E

**KPOS**

Positioning of the magnet, normally 2. Two options :
1-2 I

**If KPOS = 2**

RE, TE, RS, TS

Radius and angle of reference, respectively,
at entrance and exit of the magnet

cm, rad, cm, rad 4*E

**If KPOS = 1**

DP

Automatic positioning of the magnet, by means of
reference relative momentum

no dim. E
FFAG-SPI  Spiral FFAG magnet, N-tuple
UNDER DEVELOPMENT
\[ B_Z = \sum_{i=1}^{N} B_{Z0,i} F_i(R,\theta) (R/R_{M,i})^{K_i} \]

**IL**
\[ IL = 1, 2\times [10^n] : print field and coordinates along trajectories. \]
0-2\times [10^n]  I

**N, AT, RM**
Number of dipoles in the FFAG N-tuple ;
total angular extent of the dipole ; reference radius
no dim.,  I, 2\times E

deg, cm

Repeat N times the following sequence

**ACN, \delta RM, B_{Z0, K}**
Azimuth for dipole positioning ; \( R_{M,i} = R_M + \delta R_M \);
field at \( R_{M,i} \) ; index
deg, cm, kG, 4\times E

**g_0, \kappa**
Fringe field extent (\( g = g_0 (R_M/R)^\kappa \))
cm, no dim. 2\times E

**NC, C_0 - C_5, shift**
Unused ; \( C_0 \) to \( C_5 \) : fringe field coefficients ; EFB shift
0-6, 6*no dim, cm 1,7\times E

**\omega^+, \xi, 4 dummies**
Azimuth of entrance EFB with respect to ACN ;
spiral angle ; 4 \times unused
2*deg, 4*unused 6\times E

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

**EXIT FIELD BOUNDARY**

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

**KIRD, Resol**
KIRD=0 : analytical computation of field derivatives ;
Resol = 2/4 for 2nd/4th order field derivatives computation
KIRD2, 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 ; I, E

**XPAS**
Integration step
cm  E

**KPOS**
Positioning of the magnet, normally 2. Two options :
1-2  I

**If KPOS = 2**
\( RE, TE, RS, TS \)
Radius and angle of reference, respectively,
at entrance and exit of the magnet
cm, rad, cm, rad 4\times E

**If KPOS = 1**
\( DP \)
Automatic positioning of the magnet, by means of
reference relative momentum
no dim.  E
FIN, END

End of input data list

Any information in zgoubi.dat following these keywords will be ignored
**FIT, FIT2**

**Fitting procedure**

**NV**

Number of physical parameters to be varied

\[ \leq 20 \] 1

**For I = 1, NV**

repeat NV times the following sequence

either:

*IR, IP, XC, DV*

Number of the element in the structure;

number of the physical parameter in the element;

coupling switch (off = 0);

variation range (±)

\[ \leq MXL, \leq MXD, 2*I, 2*E \]

or:

*IR, IP, XC, [V_{\min}, V_{\max}]*

\[ \leq MXL, \leq MXD, 2*I, 3*E \]

**NC \[,\] penalty [\,ITER]]** 3

Number of constraints \[,\] penalty \[,\] number of iterations]].

\[ \leq 20 \, [10^{-n} \, [\, > 0]] \] 1 \, E \, [I]

**For I = 1, NC**

repeat NC times the following sequence:

*IC, I, J, IR, V^{4}, WV, NP [\, p_{i}(i = 1, NP)]*

*IC, I and J define the type of constraint (see table below);*

*IR : number of the element after which the constraint applies;*

*V : value; W : weight (the stronger the lower WV);*

*NP : number of parameters; if NP \geq 1, p_{i}(i = 1, NP) : parameter values.*

\[ 0-5, 3*(>0), 4*I, 2*E, \] 4*I, NP*E

\[ \leq \text{no dim.}, \text{curr. units} \]

---

1 MXL value is set in include file MXLD.H.
2 MXD value is set in include file MXLD.H. Data is of the form “integer.iii” with i a 1-digit integer.
3 FIT[2] will stop when the sum of the squared residuals gets <penalty, or when the maximum allowed number of iterations is reached.
4 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.
<table>
<thead>
<tr>
<th>Type of constraint</th>
<th>Parameters defining the constraints</th>
<th>Constraint</th>
<th>Parameter(s) values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>σ-matrix</strong></td>
<td></td>
<td>( \sigma_{I,J} ) (( \sigma_{11} = \beta_Y, \sigma_{12} = \sigma_{21} = \alpha_Y, \text{etc.} ))</td>
<td>OBJET/KOBJ=5,6</td>
</tr>
<tr>
<td>Periodic parameters</td>
<td>0.N</td>
<td>( \sigma_{I,J} ) (( \sigma_{11} = \cos\mu_Y + \alpha_Y \sin\mu_Y, \text{etc.} ))</td>
<td>OBJET/KOBJ=5.0N</td>
</tr>
<tr>
<td>(N=1-9 for MATRIX block 1-9))</td>
<td></td>
<td>( Y\text{-tune} = \mu_Y / 2\pi )</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( Z\text{-tune} = \mu_Z / 2\pi )</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \cos(\mu_Y) )</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \cos(\mu_Z) )</td>
<td></td>
</tr>
<tr>
<td><strong>First order transport coeff.</strong></td>
<td>1</td>
<td>Transport coeff. ( R_{I,J} )</td>
<td>OBJET/KOBJ=5</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>( i \neq 8 : \text{YY-determinant} ; i=8 : \text{YZ-det.} )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>( j \neq 7 : \text{ZZ-determinant} ; j=7 : \text{ZY-det.} )</td>
<td></td>
</tr>
<tr>
<td><strong>Second order transport coeff.</strong></td>
<td>2</td>
<td>Transport coeff. ( T_{I,j,k} )</td>
<td>OBJET/KOBJ=6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( j = [J/10], k = J - 10[J/10] )</td>
<td></td>
</tr>
<tr>
<td><strong>Trajectory coordinates</strong></td>
<td>3</td>
<td>( F(J,I) )</td>
<td>[MC]OBJET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( S_{\text{up}}[F(J,I)] )</td>
<td>( i=1,\text{IMAX} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \text{Dist}[F(J,I)] )</td>
<td>( i=1,12,df )</td>
</tr>
<tr>
<td></td>
<td>3.1</td>
<td>( F(J,I) - F_J^\text{O} )</td>
<td>3 11 12 df</td>
</tr>
<tr>
<td></td>
<td>3.2</td>
<td>(</td>
<td>F(J,I) - F_J^\text{O}</td>
</tr>
<tr>
<td></td>
<td>3.3</td>
<td>( \text{min.} \text{ or max.} \text{ value of} F(J,I) )</td>
<td>1 1-2</td>
</tr>
<tr>
<td></td>
<td>3.4</td>
<td>(</td>
<td>F(J,I) - F_J^\text{O}</td>
</tr>
<tr>
<td><strong>Ellipse parameters</strong></td>
<td>4</td>
<td>( \sigma_{I,J} ) (( \sigma_{11} = \beta_Y, \sigma_{12} = \sigma_{21} = \alpha_Y, \text{etc.} ))</td>
<td>OBJET/KOBJ=8 ; MCOBJET/KOBJ=3</td>
</tr>
<tr>
<td><strong>Number of particles</strong></td>
<td>5</td>
<td>( N_{\text{survived}}/\text{IMAX} )</td>
<td>OBJET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( N_{\text{in best \text{t}<em>{Y,Z,X}}}/N</em>{\text{survived}} )</td>
<td>MCOBJET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( N_{\text{in best \text{t}<em>{Y,Z,X, rms}}}/N</em>{\text{survived}} )</td>
<td>MCOBJET</td>
</tr>
<tr>
<td><strong>Spin</strong></td>
<td>10</td>
<td>( S_{X,Y,Z}(I) ), (</td>
<td>\vec{S}(I)</td>
</tr>
<tr>
<td></td>
<td>10.1</td>
<td>(</td>
<td>S_{X,Y,Z}(I) - SO_{X,Y,Z}(I)</td>
</tr>
</tbody>
</table>
FOCALE  
Particle coordinates and horizontal beam size at distance $XL$  

$XL$  
Distance from the location of the keyword  
cm  
E

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

$XL$  
Distance from the location of the keyword  
cm  
E
<table>
<thead>
<tr>
<th>GASCAT</th>
<th>Gas scattering</th>
</tr>
</thead>
<tbody>
<tr>
<td>KGA</td>
<td>Off/On switch</td>
</tr>
<tr>
<td>AI, DEN</td>
<td>Atomic number : density</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0, 1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2*E</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**GETFITVAL**  
Get values of variables as saved from former FIT[2] run

**FNAME**  
Name of storage file. Zgoubi will proceed silently if not found.
HISTO

1-D histogram

\( J, X_{\text{min}}, X_{\text{max}}, \)

\( NBK, NH \)

\( J = \) 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:
  \( 21(S_X), 22(S_Y), 23(S_Z), 24(<S>) \)

\( X_{\text{min}}, X_{\text{max}} = \) 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)

\( NBL, KAR, \)

\( NORM, TYP \)

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

\( 1-24, 2^* \)
\( 1, 2^*E, 2^*I \)
\( < 120, 1-5 \)

\( \text{normally 10-40, I, A1, A1 char., 1-2, P-S-Q} \)
<table>
<thead>
<tr>
<th>IMAGE</th>
<th>Localization and size of horizontal waist</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMAGES</td>
<td>Localization and size of horizontal waists</td>
</tr>
<tr>
<td></td>
<td>For each momentum group, as classified by means of OBJET, KOBJ = 1, 2 or 4</td>
</tr>
<tr>
<td>IMAGESZ</td>
<td>Localization and size of vertical waists</td>
</tr>
<tr>
<td></td>
<td>For each momentum group, as classified by means of OBJET, KOBJ = 1, 2 or 4</td>
</tr>
<tr>
<td>IMAGEZ</td>
<td>Localization and size of vertical waist</td>
</tr>
</tbody>
</table>
MAP2D  

2-D Cartesian uniform mesh field map - arbitrary magnetic field

\begin{tabular}{lll}
\textbf{IC, IL} & IC = 1, 2 : print the field map & 0-2, 0-2[$\times 10^n$] 2*I \\
                  & IL = 1, 2[$\times 10^n$] : print field and coordinates along trajectories & \\
\textbf{BNORM, XN,YN} & Field and X-,Y-coordinate normalization coeffs. & 3*no dim. 3*E \\
\textbf{TITL} & Title. Start with “FLIP” to get field map X-flipped. & A80 \\
\textbf{IX, JY} & Number of longitudinal and horizontal-transverse nodes of the mesh (the \(Z\) elevation is arbitrary) & \(\leq 400, \leq 200\) 2*I \\
\textbf{FNAME} & File name & A80 \\
\textbf{ID, A, B, C} & Integration boundary. Ineffective when \(ID = 0\). & \(\geq -1, 2\) no dim., \(cm, [2\) no dim., \(cm, etc.]\) 1,3*E \\
\textbf{IORDRE} & Degree of polynomial interpolation, 2nd or 4th order. & 2, 4 I \\
\textbf{XPAS} & Integration step & cm E \\
\textbf{KPOS, XCE, YCE, ALE} & \(KPOS=1\) : element aligned, 2 : misaligned ; shifts, tilt (unused if \(KPOS=1\)) & 1-2, 2*cm, rad I, 3*E \\
\end{tabular}

\(^1\) \textit{FNAME} (e.g., magnet.map) contains the field map data. These must be formatted according to the following \textsc{fortran} read sequence (normally compatible with \textsc{tosca} code \textsc{output} details and possible updates are to be found in the source file ‘fmapw.f’):

\begin{verbatim}
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)
      ENDIF
1 CONTINUE
\end{verbatim}

where \(X(I), Y(J)\) are the longitudinal, horizontal coordinates in the at nodes \((I,J)\) of the mesh, \(SZ(1)\) is the vertical elevation of the map, and \(BX, BY, BZ\) are the components of the field.

For binary files, \textit{FNAME} must begin with ‘B’, a logical flag ‘Binary’ will then automatically be set to ‘.TRUE.’
MAP2D-E  

2-D Cartesian uniform mesh field map - arbitrary electric field

**IC, IL**  
IC = 1, 2: print the field map  
IL = 1, 2*10^9: print field and coordinates along trajectories  

**ENORM, X-,Y-NORM**  
Field and X-, Y-coordinate normalization coeffs.  

**TITL**  
Title. Start with “FLIP” to get field map X-flipped.  

**IX, JY**  
Number of longitudinal and horizontal-transverse nodes of the mesh (the Z elevation is arbitrary)  

**FNAME**  
File name  

**ID, A, B, C**  
Integration boundary. Ineffective when ID = 0.  

**IORDRE**  
Degree of polynomial interpolation, 2nd or 4th order.  

**XPAS**  
Integration step  

**KPOS, XCE, YCE, ALE**  
KPOS=1: element aligned, 2: misaligned; shifts, tilt (unused if KPOS=1)

---

1. 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**

Options:

- **IORD = 0**: Same effect as FAISCEAU
- **IORD = 1** (normally using OBJET, KOBJ = 5): First order transfer matrix; beam matrix, phase advance if using OBJET, KOBJ = 5.01; if IFOC > 10: periodic beam matrix, tune numbers
- **IORD = 2** (normally using OBJET, KOBJ = 6): First order transfer matrix $[R_{ij}]$, second order array $[T_{ijk}]$ and higher order transfer coefficients; if IFOC > 10: periodic parameters,

**IFOC**

- **IFOC = 0**: matrix at actual location, reference ≡ particle # 1
- **IFOC = 1**: matrix at the closest first order horizontal focus, reference ≡ particle # 1
- **IFOC = 10 + NPER**: same as IFOC = 0, and also calculates the Twiss parameters, tune numbers, etc.

(assuming that the DATA file describes one period of a NPER-period structure).

Including 'zgoubi.MATRIX.out' will cause printout to zgoubi.MATRIX.out file
MCDESINT \(^1\)  \textbf{Monte-Carlo simulation of in-flight decay}  
\(M1 \rightarrow M2 + M3\)

\([\text{INFO},] \) \(^2\)  \(M2, M3, \tau_2\) \(^3\)  \[Switch,\] ; masses of the two decay products;  
\(\text{COM lifetime of particle 2}\)  
\(I1, I2, I3\)  \(\text{Seeds for random number generators}\)  
\([-,-,] 2^*\text{MeV}/c^2, s\)  
\([A4,] 3^*\text{E}\)  
\(3^*\approx 10^6\)  
\(3^*1\)

Particle 1 decays into 2 and 3; \texttt{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.

\(^1\) MCDESINT must be preceded by \texttt{PARTICUL}, for the definition of the mass and lifetime of the incoming particle M1.  
\(^2\) Presence of 'INFO' will cause more info on decay kinematics parameters to be printed into zgoubi.res at each decay.  
\(^3\) \(\tau_2\) 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

BORO
Reference rigidity
kG.cm

KOBJ
Type of support of the random distribution

KOBJ = 1 : window
KOBJ = 2 : grid
KOBJ = 3 : phase-space ellipses

IMAX
Number of particles to be generated
≤ 10^4

KY, KT, KZ, KP, KX, KD
Type of probability density
6*(1-3) 6*I

Y₀, T₀, Z₀, P₀,
X₀, D₀
Mean value of coordinates (D₀ = Bρ/BORO)
m, rad, m, no dim.

If KOBJ = 1

In a window

δY, δT, δZ, δP,
δX, δD
Distribution widths, depending on KY, KT etc.
m, rad, m, no dim.

NₙY, NₙT, NₙZ, NₙP,
NₙX, NₙD
Sorting cut-offs (used only for Gaussian density)
units of σₙY, σₙT, etc.

N₀, C₀, C₁, C₂, C₃
Parameters involved in calculation of P(D)
no dim.

IR₁, IR₂, IR₃
Random sequence seeds
3* ≈ 10^6

1 Let x = Y, T, Z, P or X, KY, KT, KZ, KP and KX can take the values
1 : uniform, p(x) = 1/2δx if −δx ≤ x ≤ δx
2 : Gaussian, p(x) = exp(−x²/2δx²)/δx√2π
3 : parabolic, p(x) = 3(1 − x²/δx²)/4δx if −δx ≤ x ≤ δx

KD can take the values
1 : uniform, p(D) = 1/2δD if −δD ≤ x ≤ δD
2 : exponential, p(D) = No exp(C₀ + C₁l + C₂l² + C₃l³) if −δD ≤ x ≤ δD
3 : kinematic, D = δD * T
If KOBJ = 2  

On a grid

\[ IY, IT, IZ, IP, \]
\[ IY, ID \]

Distances between bars

\[ PY, PT, PZ, PP, \]
\[ PX, PD \]

Width of the bars (±) if uniform,

\[ \delta Y, \delta T, \delta Z, \delta P, \]
\[ \delta X, \delta D \]

Sigma value if Gaussian distribution

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

Sorting cut-offs (used only for Gaussian density)

\[ N_0, C_0, C_1, C_2, C_3 \]

Parameters involved in calculation of \( P(D) \)

\[ IR1, IR2, IR3 \]

Random sequence seeds

If KOBJ = 3  

On a phase-space ellipse

\[ \alpha Y, \beta Y, \varepsilon Y / \pi, N_{\sigma Y} \]
\[ [N'_{\sigma Y} \text{ if } N_{\sigma Y} < 0] \]

Ellipse parameters and

\[ \text{emittance, } Y-T \text{ phase-space ; cut-off} \]

\[ m, \text{ units of } \sigma(\varepsilon Y) \]

\[ \alpha Z, \beta Z, \varepsilon Z / \pi, N_{\sigma Z} \]
\[ [N'_{\sigma Z} \text{ if } N_{\sigma Z} < 0] \]

Ellipse parameters and

\[ \text{emittance, } Z-P \text{ phase-space ; cut-off} \]

\[ m, \text{ units of } \sigma(\varepsilon Z) \]

\[ \alpha X, \beta X, \varepsilon X / \pi, N_{\sigma X} \]
\[ [N'_{\sigma X} \text{ if } N_{\sigma X} < 0] \]

Ellipse parameters and

\[ \text{emittance, } X-D \text{ phase-space ; cut-off} \]

\[ m, \text{ units of } \sigma(\varepsilon X) \]

\[ IR1, IR2, IR3 \]

Random sequence seeds

1  

Similar possibilities, non-random, are offered with OBJET, KOBJ=8 (p. 230)

2  

Works with Gaussian density type only : sorting 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 \( N_{\sigma_Y} > 0 \), or, if \( N_{\sigma_Y} < 0 \) sorting within the ring

\[ |N_{\sigma_Y}, N'_{\sigma_Y}| \]
A scheme of input parameters to \textit{MCOBJET} when \textit{KOBJ}=2.

Top : Possible distributions of the $Y$ coordinate
Bottom : A 2-D grid in $(Y, Z)$ space.
**MULTIPOL** Magnetic Multipole

**IL**

\[ IL = 1, 2 \times 10^n \] : print field and coordinates along trajectories

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-2[\times10^n]</td>
<td>I</td>
</tr>
</tbody>
</table>

**XL, R_0, B_1, B_2, ..., B_{10}**

Length of element ; radius at pole tip ; field at pole tip for dipole, quadrupole, ..., dodecapole components

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2<em>cm,10</em>kG</td>
<td>12*E</td>
</tr>
</tbody>
</table>

**Entrance face**

**X_E, \lambda_E, E_2, ..., E_{10}**

Integration zone ; fringe field extent : dipole fringe field extent = \lambda_E ; quadrupole fringe field extent = \lambda_E \times E_2 ; ...

20-pole fringe field extent = \lambda_E \times E_{10}

(sharp edge if field extent is zero)

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2<em>cm,9</em>no dim.</td>
<td>11*E</td>
</tr>
</tbody>
</table>

**NCE, C_0 - C_5**

same as QUADRUPO

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-6, 6*no dim.</td>
<td>I, 6*E</td>
</tr>
</tbody>
</table>

**Exit face**

**X_S, \lambda_S, S_2, ..., S_{10}**

Integration zone ; as for entrance

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2<em>cm, 9</em>no dim.</td>
<td>11*E</td>
</tr>
</tbody>
</table>

**NCS, C_0 - C_5**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-6, 6*no dim.</td>
<td>I, 6*E</td>
</tr>
</tbody>
</table>

**R_1, R_2, R_3, ..., R_{10}**

Skew angles of field components

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10*rad</td>
<td>10*E</td>
</tr>
</tbody>
</table>

**XPAS**

Integration step

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm</td>
<td>E</td>
</tr>
</tbody>
</table>

**KPOS, XCE, YCE, ALE**

KPOS=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if KPOS=1)

for QUADRUPO.

KPOS = 3 : effective only if B_1 \neq 0 : entrance and exit frames are shifted by YCE and tilted wrt. the magnet by an angle of

- either ALE if ALE\neq0
- or 2Arcsin(B_1 XL / 2BORO) if ALE=0

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3, 2*cm, rad</td>
<td>I, 3*E</td>
</tr>
</tbody>
</table>
OBJET  Generation of an object

BORO  Reference rigidity  kG.cm  E


If KOBJ = 1[01]  [Non- Symmetric object

IY, IT, IZ, IP, IX, ID  Ray-Tracing assumes mid-plane symmetry  1Y*IT*IZ*IP*IX*ID ≤ 10^4  6*1

Total number of points in ±Y, ±T, ±Z, ±P (+Z, +P with KOBJ = 1.01), ±X, and ±D coordinates (Y ≤ 20,.....ID ≤ 20)

PY, PT, PZ, PP, PX, PD  Step size in Y, T, Z, P, X and momentum (PD = δBρ/BORO)  2(cm,mrad), cm, no dim.  6*E

YR, TR, ZR, PR, XR, DR  Reference (DR = Bρ/BORO)  2(cm,mrad), cm, no dim.  6*E

If KOBJ = 2[01]  All the initial coordinates must be entered explicitly

IMAX, IDMAX  total number of particles ; number of distinct momenta (if IDMAX > 1, group particles of same momentum)  IMAX ≤ 10^4  2*I

For I = 1, IMAX  Repeat IMAX times the following line

Y, T, Z, P, X, D, LET  Coordinates and tagging of the IMAX particles ;  2(cm,mrad), cm, no dim., 6*E, A1

If KOBJ = 2.01 input units are different :  2(m,rad), m, no dim.,

IEX(I = 1, IMAX)  IMAX times 1 or -9. If IEX(I) = 1 trajectory I is ray-traced, it is not if IEX(I) = -9.  1 or -9  IMAXI

If KOBJ=3[NN, NN=00...03]  Reads coordinates from a storage file

IT1, IT2, ITStep  Read particles numbered IT1 to IT2, step ITStep  ≥ 1, ≥ IT1, ≥ 1  3*I

(For more than 10^4 particles stored in FNAME, use ‘REBELOTE’)

IP1, IP2, IPStep  Read particles that belong in pass numbered ≥ 1, ≥ IP1, ≥ 1  3*I

YP, TF, ZF, PF, XF, DF, TF, TAG  Scaling factor. TAG-ing letter : no effect if '*', 7*no.dim, char. 7*E, A1

otherwise only particles with TAG≡LET are retained.

YR, TR, ZR, PR, XR, DR, TR  Reference. Given the previous line of data, all coordinate C is transformed to C*CF+CR  2(cm, mrad), cm, no dim.,μs

InitC  0 : set new R0 = old R0, new R = old R ;  0-1  I

1 : set new R0 = old R0, new R = old R ;

2 : save old R in new R0, set new R = old R0.

FNAME  File name (e.g., zgoubi.fai)  A80

(ND in KOBJ=3.NN determines storage FORMAT)

If KOBJ = 5[NN, NN=01,99]  Generation of 11 particles, or 11*NN if I ≥ 2 (for use with MATRIX, IORD = 1)
PY, PT, PZ, PP, PX, PD  Step sizes in Y, T, Z, P, X and D  2(cm, mrad), cm, no dim.  6*E

YR, TR, ZR, PR, XR, DR  Reference trajectory (DR = Bρ/BORO)  2(cm, mrad), cm, no dim.  6*E

If KOBJ = 5.01  additional data line:

\( a_Y, \beta_Y, a_Z, \beta_Z, a_X, \beta_X, D_Y, D'_Y, D_Z, D'_Z \)

Initial beam ellipse parameters  1  2(no dim., m), ?, ?, 6*E, 2(m, rad)  4*E

If KOBJ = 5.NN,  NN=02-99  i=1 to 98 (if, resp. NN=02 to 99) additional data lines:

YR, TR, ZR, PR, XR, DR  Reference trajectory # i (DR = Bρ/BORO)  2(cm, mrad), cm, no dim.  6*E

If KOBJ = 6  Generation of 61 particles (for use with MATRIX, IORD = 2)

PY, PT, PZ, PP, PX, PD  Step sizes in Y, T, Z, P, X and D  2(cm, mrad), cm, no dim.  6*E

YR, TR, ZR, PR, XR, DR  Reference trajectory ; DR = Bρ/BORO  2(cm, mrad), cm, no dim.  6*E

If KOBJ = 7  Object with kinematics

IY, IT, IZ, IP, IX, ID  Number of points in \( \pm Y, \pm T, \pm Z, \pm P, \pm X \); ID is not used  \( IY*IT*IZ*IP*IX*ID \leq 10^4 \)  6*I

PY, PT, PZ, PP, PX, PD  Step sizes in Y, T, Z, P and X; PD = kinematic coefficient, such that \( D(T) = DR + PD*T \)  2(cm, mrad), cm, mrad\(^{-1}\)  6*E

YR, TR, ZR, PR, XR, DR  Reference (DR = Bρ/BORO)  2(cm, mrad), cm, no dim.  6*E

If KOBJ = 8  Generation of phase-space coordinates on ellipses  2

IY, IZ, IX  Number of samples in each 2-D phase-space ; if zero the central value (below) is assigned  \( 0 \leq IX, IY, IZ \leq IMAX \),  \( 1 \leq IX*IY*IZ \leq IMAX \)  3*I

\( Y_0, T_0, Z_0, P_0, X_0, D_0 \)  Central values (\( D_0 = B\rho/BORO \))  m, rad, m, rad, m, no dim.  6*E

\( a_Y, \beta_Y, \varepsilon_Y/\pi \)  ellipse parameters and emittances  no dim., m, m  3*E

\( a_Z, \beta_Z, \varepsilon_Z/\pi \)  no dim., m, m  3*E

\( a_X, \beta_X, \varepsilon_X/\pi \)  no dim., m, m  3*E

---

1 They can be transported by using MATRIX

2 Similar possibilities, random, are offered with MCOBJET, KOBJ=3 (p. 226)
<table>
<thead>
<tr>
<th><strong>OBJETA</strong></th>
<th>Object from Monte-Carlo simulation of decay reaction</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>M</strong>&lt;sub&gt;1&lt;/sub&gt; + <strong>M</strong>&lt;sub&gt;2&lt;/sub&gt; −→ <strong>M</strong>&lt;sub&gt;3&lt;/sub&gt; + <strong>M</strong>&lt;sub&gt;4&lt;/sub&gt; and <strong>M</strong>&lt;sub&gt;4&lt;/sub&gt; −→ <strong>M</strong>&lt;sub&gt;5&lt;/sub&gt; + <strong>M</strong>&lt;sub&gt;6&lt;/sub&gt;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>BORO</strong></th>
<th>Reference rigidity</th>
<th>kG.cm</th>
<th>E</th>
</tr>
</thead>
</table>

| **IBODY, KOBJ** | Body to tracked: **M**<sub>3</sub> (**IBODY**=1), **M**<sub>5</sub> (**IBODY**=2) **M**<sub>6</sub> (**IBODY**=3); type of distribution for **Y**<sub>0</sub> and **Z**<sub>0</sub> : uniform (**KOBJ**= 1) or Gaussian (**KOBJ**= 2) |
| **IBODY, KOBJ** | 1-3,1-2 |

| **IMAX** | Number of particles to be generated (use ‘**REBELOTE**’ for more) | \( \leq 10^4 \) |

| **M**<sub>1</sub> − **M**<sub>6</sub> | Rest masses of the bodies | 6*GeV/c<sup>2</sup> |
| **T**<sub>1</sub> | Kinetic energy of incident body | GeV |

| **Y**<sub>0</sub>, **T**<sub>0</sub>, **Z**<sub>0</sub>, **P**<sub>0</sub>, **D**<sub>0</sub> | 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 |
| **Y**<sub>0</sub>, **T**<sub>0</sub>, **Z**<sub>0</sub>, **P**<sub>0</sub>, **D**<sub>0</sub> | 2(cm,mrad), 5*E |

| **δ**<sub>Y</sub>, **δ**<sub>T</sub>, **δ**<sub>Z</sub>, **δ**<sub>P</sub>, **δ**<sub>D</sub> | 2(cm,mrad), no dim. |

| **XL** | Half length of object: \( -XL \leq X_0 \leq XL \) (uniform random distribution) | cm |

| **IR1, IR2** | Random sequence seeds | \( 2^* \sim 0^6 \) | 2*I |
Octupole

Octupole magnet

**IL**

IL = 1, 2\([\times 10^9]\) : print field and coordinates along trajectories. 0-2\([\times 10^9]\) I

**XL, R₀, B₀**

Length ; radius and field at pole tip of the element 2\(^{\text{cm}}, \text{kG}\) 3\(^{\text{E}}\)

**Entrance face :**

Xₜ, λₜ

Integration zone ; Fringe field extent (λₜ = 0 for sharp edge) 2\(^{\text{cm}}\) 2\(^{\text{E}}\)

NCE, C₀ – C₅

NCE = unused
C₀ – C₅ = fringe field coefficients such that : \(G(s) = G₀/(1 + \exp P(s))\), with \(G₀ = B₀/R₀^3\) and \(P(s) = \sum_{i=0}^{5} Cᵢ(s/λ)^i\)

any, 6\(^{\text{no dim.}}\) I, 6\(^{\text{E}}\)

**Exit face :**

Xₛ, λₛ

Parameters for the exit fringe field ; see entrance 2\(^{\text{cm}}\) 2\(^{\text{E}}\)

NCS, C₀ – C₅

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 ; 1-2, 2\(^{\text{cm}}, \text{rad}\) I, 3\(^{\text{E}}\)

shifts, tilt (unused if KPOS=1)

Octupole magnet
Keywords and input data formatting

**OPTICS**  
**Write out optical functions**

**IOPT, label, IMP**  

\( IOPT = 0/1 : \) Off/On. Transport the beam matrix;  

\( \text{0-1, string, 0-1} \)  

I, A, I  

'label' : Can be 'all', 'ALL', or existing 'LABEL_1(NOEL)';  

IMP = 1 causes storage of optical functions in zgoubi.OPTICS.out.
<table>
<thead>
<tr>
<th>ORDRE</th>
<th>Taylor expansions order</th>
</tr>
</thead>
<tbody>
<tr>
<td>IO</td>
<td>Taylor expansions of $\vec{R}$ and $\vec{u}$ up to $\vec{u}^{(IO)}$ (default is $IO = 4$)</td>
</tr>
</tbody>
</table>
PARTICUL Particle characteristics

$M$, $Q$, $G$, $\tau$, $X$  Mass ; charge ; gyromagnetic factor ; COM life-time ; unused  \( \text{MeV/c}^2 \), C, no dim., s  5*E

If $M$ is of the form $\{M_1 \ M_2\}$, then when masses are assigned to particles from a previously defined object, the first half of the particles are given the mass $M_1$, and the second half are given the mass $M_2$.

If $Q$ is zero, the reference charge is left unchanged.

NOTE: Only the parameters of concern need their value be specified (for instance $M$, $Q$ when electric lenses are used); others can be set to zero.
PICKUPS

Beam centroid path; closed orbit

\[ N \]

0 : inactive
\[ \geq 1 \] : total number of \( LABLE \)’s
at which beam centroid is to be recorded
\[ \geq 0 \] I

For \( I = 1, N \)
A list of \( N \) records follows

\( LABLE \)’s
\( N \) labels at which beam centroid is to be recorded
\( N \) string(s) \( N*A10 \)
Keywords and input data formatting

PLOTDATA

Intermediate output for the PLOTDATA graphic software

To be documented.
**POISSION**

Read magnetic field data from **POISSION** output

**IC, IL**

\[ IC = 1, 2 \] : print the field map

\[ IL = 1, 2 \times 10^6 \] : print field and coordinates along trajectories.

**BNORM, XN, YN**

Field and X-, Y-coordinate normalization coeffs.

**TITL**

Title. Start with “FLIP” to get field map X-flipped

**IX, IY**

Number of longitudinal and transverse nodes of the uniform mesh

\[ \leq 400, \leq 200 \]

**FNAME**

File name

**ID, A, B, C**

Integration boundary. Ineffective when \[ ID = 0 \].

\[ ID = -1, 1 \text{ or } \geq 2 \] : as for **CARTEMES**

**IORDRE**

Degree of interpolation polynomial as for **DIPOLE-M**

\[ 2, 25 \text{ or } 4 \]

**XPAS**

Integration step

\[ \text{cm} \]

**KPOS, XCE, YCE, ALE**

\[ KPOS=1 \] : element aligned, \[ 2 \] : misaligned ;

shifts, tilt (unused if \[ KPOS=1 \])

\[ 1-2, 2 \text{ cm, rad} \]

---

1 **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”:

\[
\begin{align*}
I = 0 \\
11 \text{ CONTINUE} \\
I = I+1 \\
101 \text{ FORMAT(I1, I3, I4, E15.6, 2F11.5, 2F12.3)} \\
\text{GOTO II} \\
10 \text{ CONTINUE}
\end{align*}
\]

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 **POISSION** output file outpoi.lis, not used here.
Keywords and input data formatting

POLARMES  2-D polar mesh magnetic field map
mid-plane symmetry is assumed

\text{IC, IL} \quad IC = 1, 2: \text{print the map}
\quad IL = 1, 2[\times 10^n]: \text{print field and coordinates along trajectories.}

\text{BNORM, AN,RN} \quad \text{Field and A-,R-coordinate normalization coeffs.}

\text{TITL} \quad \text{Title. Start with “FLIP” to get field map X-flipped}

\text{IA, JR} \quad \text{Number of angular and radial nodes of the mesh}
\quad \leq 400, \leq 200

\text{FNAME} \quad \text{File name}

\text{ID, A, B, C} \quad \text{Integration boundary. Ineffective when } ID = 0.
\quad ID = -1, 1 \text{ or } \geq 2: \text{as for CARTEMES}

\text{IORDRE} \quad \text{Degree of interpolation polynomial}
\quad (\text{see DIPOLE-M})

\text{XPAS} \quad \text{Integration step}
\quad \text{cm}

\text{KPOS} \quad \text{as for DIPOLE-M. Normally 2.}
\quad \text{1-2}

\text{RE, TE, RS, TS} \quad \text{cm, rad, cm, rad}

\text{D P} \quad \text{no dim.}

\text{1 FNAME (e.g., spes2.map) contains the field data.}
\text{These must be formatted according to the following FORTRAN read sequence - details}
\text{and possible updates are to be found in the source file ‘fmapw.f’.}

\text{OPEN (UNIT = NL, FILE = FNAME, STATUS = ‘OLD’ [FORM=’UNFORMATTED’])}
\text{IF (BINARY) THEN}
\quad \text{READ(NL) (Y(J), J=1, JY)}
\quad \text{ELSE}
\quad \text{READ(NL,100) (Y(J), J=1, JY)}
\text{ENDIF}

\text{100 FORMAT(10 F8.2)}
\quad \text{DO 1 I = 1,IX}
\quad \text{IF (BINARY) THEN}
\quad \quad \text{READ (NL) X(I), (BMES(I,J), J=1, JY)}
\quad \text{ELSE}
\quad \quad \text{READ(NL,101) X(I), (BMES(I,J), J=1, JY)}
\quad \text{ENDIF}

\text{101 FORMAT(10 F8.1)}
\quad \text{1 CONTINUE}

where \(X(I)\) and \(Y(J)\) are the longitudinal and transverse coordinates and \text{BMES} is the \(Z\) field component at a node \((I, J)\)
of the mesh. For binary files, \text{FNAME} must begin with ‘B,’ or ‘b,’. ‘Binary’ will then automatically be set to ‘.TRUE.’
PS170

Simulation of a round shape dipole magnet

$IL$  
$IL = 1, 2 \times 10^n$: print field and coordinates along trajectories.  
0-2\times10^n  
I

$XL, R_0, B_0$  
Length of the element, radius of the circular dipole, field  
2*cm, kG  
3*E

$XPAS$  
Integration step  
cm  
E

$KPOS, XCE, YCE, ALE$  
$KPOS=1$: element aligned, 2: misaligned; shifts, tilt (unused if $KPOS=1$)  
1-2, 2*cm, rad  
I, 3*E

Scheme of the PS170 magnet simulation.
### QUADISEX

**Sharp edge magnetic multipoles**

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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(IL)</td>
<td>(IL = 1, 2 \times 10^9) : print field and coordinates along trajectories.</td>
</tr>
<tr>
<td>(XL, R_0, B_0)</td>
<td>Length of the element ; normalization distance ; field</td>
</tr>
<tr>
<td>(N, EB1, EB2, EG1, EG2)</td>
<td>Coefficients for the calculation of (B).</td>
</tr>
<tr>
<td>(KPOS, XCE, YCE, ALE)</td>
<td>Integration step</td>
</tr>
</tbody>
</table>

**Input Data Formatting**

- **IL**: 0-2 \([\times 10^9]\) I
- **XL, R_0, B_0**: 2 cm, kG, 3 E
- **N, EB1, EB2, EG1, EG2**: 5 no dim., 5 E
- **KPOS=1, XCE, YCE, ALE**: 1-2, 2 cm, rad, I, 3 E
### QUADRUPO

**Quadrupole magnet**

| **IL**  | ÍL = 1, 2[×10⁶] : print field and coordinates along trajectories. | 0-2[×10⁶] | I |
| **XL, R₀, B₀** | Length : radius and field at pole tip | 2*cm, kG | 3*E |
| **Xₑ, λₑ** | Integration zone extent ; fringe field extent (≃ 2R₀, λₑ = 0 for sharp edge) | 2*cm | 2*E |
| **NCE, C₀ − C₅** | NCE = unused. C₀ − C₅ = Fringe field coefficients such that \( G(s) = \frac{G₀}{1 + \exp P(s)} \), with \( G₀ = B₀/R₀ \) and \( P(s) = \sum_{i=0}^{5} Cᵢ (s/λ)ⁱ \) | any, 6*no dim. | 1, 6*E |
| **Xₛ, λₛ** | See entrance face | 2*cm | 2*E |
| **NCS, C₀ − C₅** | 0-6, 6*no dim. | 1, 6*E |
| **XPAS** | Integration step | cm | E |
| **KPOS, XCE, YCE, ALE** | KPOS=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if KPOS=1) | 1-2, 2*cm, rad | 1, 3*E |
Scheme of the elements QUADRUPO, SEXTUPOLE, 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 \(-XE\) to \(XL + XS\), by means of automatic prior and further \(XE\) and \(XS\) translations.
REBELOTE  
Jump to the beginning of zgoubiinput data file

NPASS, KWRIT, K[n], [Label1 [Label2]]

NPASS : Number of runs; KWRIT = 1.1 (resp. 0.0) switches
(inhibits) FORTRAN WRITEs to .res and to screen;
K option:
K = 0 : initial conditions (coordinates and spins)
are generated following the regular functioning
of object definitions. If random generators are
used (e.g. in MCOBJET) their seeds will not be reset.
K = 22 : next run will account for new parameter values in
zgoubi.dat data list, see below.
K = 99 : coordinates at end of previous pass are used as initial
coordinates for the next pass; idem for spin components.
K = 99.1 : Label1 is expected, subsequent passes will start from
element with Label1 down to REBELOTE and so forth;
K = 99.2 : Label1 and Label2 are expected; last pass (# NPASS+1)
will end at element with Label2 whereupon execution will jump to the keyword
next to REBELOTE and will be carried out down to ‘END’.

If K = 22

NPRM  Number of parameters to be changed for next runs  1

Repeat NPR times the following sequence (tells parameters concerned, and for each its successive values):

LMNT, PRM, NV*Val  Keyword # in zgoubi.dat list; parameter # under that Keyword
(same as for FIT[2], see page 213); NV successive values (if
NV < NPASS then last value is maintained over remaining passes).

1 K=22 is compatible with use of the FIT[2] procedure: e.g., allows successive FITs in a run, with successive sets of optical parameters.
2 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.
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

IOPT, NFAM  

- **IOPT** = 0 (inactive) or 1 (active) ;
- **NFAM** = number of families to be scaled

For NF=1, NFAM :

repeat NFAM times the following sequence :

- **NAMEF [, Lbl [, Lbl]]**  Name of family (i.e., keyword of concern) [, up to 2 labels]
- **NT**  
  - **NT > 0** : number of timings ;
  - **NT = −1** : field scaling factor updated by CAVITE ;
  - **NT = −2** : RF law in CAVITE is read from external data file.

- **SCL(I), I = 1, NT**  Scaling values (a single one, normally 1, if NT = −1).
- **TIM(I), I = 1, NT**  Corresponding timings, in units of turns (1 if NT = −1).
**Keywords and input data formatting**

SEPARA \(^1\) Wien Filter - analytical simulation

\[ IA, XL, E, B, \]

\[ IA = 0 : \text{element inactive} \]
\[ IA = 1 : \text{horizontal separation} \]
\[ IA = 2 : \text{vertical separation} \]

Length of the separator; electric field; magnetic field.

Horizontal separation between a wanted particle, \((W)\), and an unwanted particle, \((U)\).

\((W)\) undergoes a linear motion while \((U)\) undergoes a cycloidal motion.

---

\(^1\) SEPARA must be preceded by PARTICUL for the definition of mass and charge of the particles.
**SEXQUAD**

**Sharp edge magnetic multipole**

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

**IL**

\( IL = 1, 2 \times 10^9 \) : print field and coordinates along trajectories.

**XL, R_0, B_0**

Length of the element; normalization distance; field

\( 2 \times 10^9 \) I

**N, EB1, EB2, EG1, EG2**

Coefficients for the calculation of \( B \).

if \( Y > 0 \) : \( B = EB1 \) and \( G = EG1 \);

if \( Y < 0 \) : \( B = EB2 \) and \( G = EG2 \).

**XPAS**

Integration step

\( 5 \times \text{no dim.} \) E

**KPOS, XCE, YCE, ALE**

\( KPOS = 1 \) : element aligned, 2 : misaligned;

shifts, tilt (unused if \( KPOS = 1 \))

\( 1-2, 2 \times \text{cm, rad} \) I, 3 E
### Sextupole Magnet

**IL**

(IL = 1, 2[×10^9]) : print field and coordinates along trajectories.

| 0-2[×10^9] | I |

**XL**, **R₀**, **B₀**

Length ; radius and field at pole tip of the element

| 2*cm, kG | 3*E |

**Entrance face**

**Xₑ**, **λₑ**

Integration zone ; fringe field extent (λₑ = 0 for sharp edge)

| 2*cm | 2*E |

**NCE, C₀ − C₅**

NCE = unused

C₀ − C₅ = Fringe field coefficients such that

\[
G(s) = G₀/(1 + \exp P(s)), \quad \text{with} \quad G₀ = B₀/R₀²
\]

and

\[
P(s) = \sum_{i=0}^{5} Cᵢ(s/λᵢ)ᵢ
\]

| any, 6* | 1, 6*E | no dim. |

**Exit face**

**Xₛ**, **λₛ**

Parameters for the exit fringe field ; see entrance

| 2*cm | 2*E |

**NCS, C₀ − C₅**

| 0-6, 6*no dim. | 1, 6*E |

**XPAS**

Integration step

| cm | E |

**KPOS, XCE**, **YCE, ALE**

KPOS=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if KPOS=1)

| 1-2, 2*cm, rad | 1, 3*E |

---

**Keywords and input data formatting**

![Sextupole magnet](image.png)
<table>
<thead>
<tr>
<th>SPINR</th>
<th>SPINRTitl</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Theta_x, \Phi_X$</td>
<td>Angles that define the spin precession axis.</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Spin precession angle.</td>
</tr>
</tbody>
</table>
**SOLENOID**

**SOLENOID TID**

- **IL**
  - $IL = 1, 2 \times 10^6$: print field and coordinates along trajectories.
  - 0-2\( \times 10^6 \) I

- **XL, R_0, B_0**
  - Length; radius; asymptotic field ($\mu_0 NI/\chi L$)
  - 2\*cm, kG 3E

- **X_E, X_S**
  - Entrance and exit integration zones
  - 2\*cm 2E

- **XPAS**
  - Integration step
  - cm E

- **KPOS, XCE, YCE, ALE**
  - $KPOS = 1$: element aligned, 2: misaligned;
  - 1-2, 2\*cm, rad 1, 3E
  - shifts, tilt (unused if $KPOS = 1$)
Keywords and input data formatting

**SPNPRNL**  
Store spin coordinates in file *FNAME*

*FNAME*  
Name of storage file (*e.g.*, zgoubi.spn)  

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FNAME</td>
<td>Name of storage file (<em>e.g.</em>, zgoubi.spn)</td>
<td>A80</td>
</tr>
</tbody>
</table>

**SPNSTORE**  
Store spin coordinates every *IP* other pass

*FNAME*  
Name of storage file (*e.g.*, zgoubi.spn) [ : label(s) of the element(s) at the exit of which the store occurs (10 labels maximum)].  

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FNAME</td>
<td>Name of storage file (<em>e.g.</em>, zgoubi.spn)</td>
<td>A80</td>
</tr>
<tr>
<td>[, LABEL(s)]</td>
<td>Label(s) of the element(s) at the exit of which the store occurs</td>
<td>[, 10*A10]</td>
</tr>
</tbody>
</table>

*IP*  
Store every *IP* other pass (when using *REBELOTE* with *NPASS* ≥ *IP* − 1).

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>Store every <em>IP</em> other pass</td>
<td>I</td>
</tr>
</tbody>
</table>

**SPNPRT**  
Print spin coordinates

Print spin coordinates into zgoubi.res, at the location where this keyword is introduced in the structure.

1 *FNAME* = 'none' will inhibit printing.

2 If first *LABEL* = 'none' then printing will be inhibited.
SPNTRK \(^1\)  
Spin tracking

**KSO**  
Various options for initial conditions  
1-5  
I

**If KSO = 1 – 3**  
*KSO = 1* (respectively 2, 3) : all particles have their spin automatically set to \((1,0,0) – \text{longitudinal}\) [respectively \((0,1,0) – \text{horizontal}\) and \((0,0,1) – \text{vertical}\)]

**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 = 5**  
Random distribution in a cone (see figure)  
Enter the following two sequences :

\(TO, PO, A, \delta A\)  
Angles of average polarization :
\(A = \text{angle of the cone ; } \delta A = \text{standard deviation of distribution around } A\)

\(IR\)  
Random sequence seed
\(\leq 10^6\)  
I

---

\(^1\) SPNTRK must be preceded by PARTICUL for the definition of \(G\) and mass.
<table>
<thead>
<tr>
<th><strong>SRLOSS</strong></th>
<th><strong>Synchrotron radiation loss</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>KSR[i]</strong></td>
<td>Switch: $i = 1$ causes info output into zgoubi.SRLOSS.out</td>
</tr>
<tr>
<td><strong>STR1, STR2</strong></td>
<td>Options: $STR1 = 'ALL'$ or 'all' or a particular KEYWORD; $STR2 = 'scale'$ will scale fields with energy loss.</td>
</tr>
<tr>
<td><strong>Option, seed</strong></td>
<td>1: loss entails dp only</td>
</tr>
</tbody>
</table>
SRPRNT  Print SR loss statistics  into zgoubi.res
### SYNRAD

**Synchrotron radiation spectral-angular densities**

<table>
<thead>
<tr>
<th>KSR</th>
<th>Switch</th>
<th>0-2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>inhibit SR calculations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>start</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>stop</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**If KSR = 0**

| D1, D2, D3 | Dummies                      | 3*E |

**If KSR = 1**

| X0, Y0, Z0 | Observer position in frame of magnet next to SYNRAD | 3*m | 3*E |

**If KSR = 2**

| \(\nu_1, \nu_2, N\) | Frequency range and sampling | 2*eV, no dim. | 2*E, I |
TOSCA 2-D and 3-D Cartesian or cylindrical mesh field map

IC, IL see CARTEMES 0-2, 0-2 2*I

BNORM, XN,YN,ZN Field and X-,Y-,Z-coordinate normalization coefficients 4*no dim. 4*E

TITL Title. Include “FLIP” to get field map X-flipped. Include “HEADER n” in case FNAME starts with n ≥ 1 header lines.

IX, IY,IZ, MOD,MOD2 Number of nodes of the mesh in the X, Y and Z directions, IZ = 1 for single 2-D map ;

MOD : operational and map FORMAT reading mode 2 ;
MOD≤19 : Cartesian mesh ;
MOD≥20 : cylindrical mesh.
MOD≥20, optional, tells the reading FORMAT, default is ‘*’.

FNAME Names of the NF files that contain the 2-D maps, from Z(1) to Z(NF).

(K = 1, NF) If MOD=0 : NF = 1 + [IZ/2], the NF 2-D maps are for 0 ≤ Z ≤ Zmax ,
they are symmetrized with respect to the Z(1) = 0 plane.
If MOD=1 : NF = IZ, no symmetry assumed ; Z(1) = Zmax ,
Z(1+[IZ/2]) = 0 and Z(NF) = −Zmax .
If MOD=12 : a single FNAME file contains the all 3-D volume.
If MOD=20-22 : other symmetry options, see toscp.f routine...

ID,A, B, C [, A’, B’, C’], Integration boundary. Ineffective when ID = 0.
A′′, etc., if ID ≥ 2] ID = -1, 1 or ≥ 2 : as for CARTEMES

≥ -1, cm, I,3*E

IORDRE If IZ = 1 : 3, 4, 25, as in CARTEMES ; unused if IZ ≠ 1.
2, 25 or 4 I

XPAS Integration step cm E

If Cartesian mesh (see MOD) :
KPOS, XCE, YCE, ALE KPOS=1 : element aligned, 2 : misaligned ; shifts, tilt

1-2, 2*cm, rad I, 3*E

If polar mesh :
KPOS as for POLARMES. Normally 2.
1-2 I

If KPOS = 2
RE, TE, RS, TS cm, rad, cm, rad 4*E

1MXX, MXY, IZ may be changed, they are stated in the include file PARIZ.H.

2Each 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, etc.) :

DO K = 1, NF
OPEN (UNIT = NL, FILE = FNAME(K), STATUS = ’OLD’ [.FORM=’UNFORMATTED’])
DO J = 1, JY ; DO 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) node coordinates, field components at node
ELSE
READ(NL,*) Y(J), Z(K), X(I), BY(J,K,I), BZ(J,K,I), BX(J,K,I) node coordinates, field components at node
ENDIF
ENDDO ; ENDDO
NL = NL + 1
ENDDO

For 2-D maps BX and BY are assumed zero at all nodes of the 2-D mesh, regardless of BX(J,1,I), BY(J,1,I) values. For binary files, FNAME must begin with ’B’, or ’b’, .
TRANSMT Matrix transfer

**IORDRE**  Transfer matrix order  1-2  I

**XL**  Length (ineffective, for updating)  m  E

For $IA = 1, 6$:

$R(IA, IB), IB = 1, 6$  First order matrix  m, rad  6 lines

**IF IORDRE = 2**  Following records *only if* IORDRE = 2

$T(IA, IB, IC)$,  Second order matrix, six 6*6 blocks  m, rad  36 lines
<table>
<thead>
<tr>
<th>TRAROT</th>
<th>Translation-Rotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$TX, TY, TZ,$</td>
<td></td>
</tr>
<tr>
<td>$RX, RY, RZ$</td>
<td>Translations, rotations</td>
</tr>
</tbody>
</table>
Keywords and input data formatting

**TWISS**

**Calculation of periodic optical parameters**

**KTW, FacD, FacA**

\[
K_{TW} = 0/1/2/3 : \text{Off} / \text{as MATRIX} / \text{computation of chromaticities} / \text{computation of anharmonicities.}
\]

\[
FacD \times D = \delta p/p \text{ value applied, with } D \text{ the momentum sampling in OBJET} ; \text{FacA : unused.}
\]
UNDULATOR  Undulator magnet

Under development, to be documented
**UNIPOT**

**Unipotential electrostatic lens**

*IL*

IL = 1, 2: print field and coordinates along trajectories

0-2 \( \times 10^n \)

*\( X_1, D, X_2, X_3, R_0 \)*

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;

1-2, 2*cm, rad I, 3*E

Shifts, tilt (unused if KPOS=1)
**VENUS Simulation of a rectangular dipole magnet**

\( IL = 1, 2 \times 10^9 \): print field and coordinates along trajectories. 

- **XL, YL, B_0**: Length; width = ±YL; field
- **XPAS**: Integration step
- **KPOS, XCE, YCE, ALE**: 
  - **KPOS=1**: element aligned, 2: misaligned; 
  - shifts, tilt (unused if KPOS=1)

**Scheme of VENUS rectangular dipole.**
<table>
<thead>
<tr>
<th><strong>WIENFILT</strong></th>
<th><strong>Wien filter</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IL</strong></td>
<td>$IL = 1, 2 \times 10^n$: print field and coordinates along trajectories.</td>
</tr>
<tr>
<td><strong>XL, E, B, HV</strong></td>
<td>Length; electric field; magnetic field; option: element inactive ($HV = 0$) horizontal; $(HV = 1)$ or vertical ($HV = 2$) separation</td>
</tr>
<tr>
<td><strong>XE, (\lambda_{E_E}), (\lambda_{B_E})</strong></td>
<td>Integration zone extent; fringe field extents, E and B respectively ($\approx$ gap height)</td>
</tr>
<tr>
<td><strong>CE0–CE5</strong></td>
<td>Fringe field coefficients for E</td>
</tr>
<tr>
<td><strong>CB0–CB5</strong></td>
<td>Fringe field coefficients for B</td>
</tr>
<tr>
<td><strong>XS, (\lambda_{E_S}), (\lambda_{B_S})</strong></td>
<td>See entrance face</td>
</tr>
<tr>
<td><strong>CE0–CE5</strong></td>
<td>6*no dim.</td>
</tr>
<tr>
<td><strong>CB0–CB5</strong></td>
<td>6*no dim.</td>
</tr>
<tr>
<td><strong>XPAS</strong></td>
<td>Integration step</td>
</tr>
<tr>
<td><strong>KPOS, XCE, YCE, ALE</strong></td>
<td>$KPOS = 1$: element aligned, 2: misaligned; shifts, tilt (unused if $KPOS = 1$)</td>
</tr>
</tbody>
</table>

1 Use **PARTICUL** to declare mass and charge.
YMY

Reverse signs of \(Y\) and \(Z\) axes

Equivalent to a 180° rotation with respect to \(X\)-axis

The 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 [53], 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. 47.

Example 2: calculates the first and second order transfer matrices of an 800 MeV/c kaon beam line [54] 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. 48.

Example 3: illustrates the use of MCDESINT and REBELOTE with a simulation of the in-flight decay

\[ K \rightarrow \mu + \nu \]

in the SATURNE Laboratory spectrometer SPES 3 [31]. The angular acceptance of SPES 3 is ±50 mrd horizontally and ±50 mrd vertically; its momentum acceptance is ±40%. The bending magnet is simulated with DIPOLE. The layout of SPES 3 is given in Fig. 49.

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 [55].

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, 56]. 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. 51, 52 (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. 53).

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 [53].

Figure 47: SPES 2 Layout.
zgoubi data file.

SPES2 QGD SPECTROMETER, USING FIELD MAPS; MONTE-CARLO OBJECT WITH MOMENTUM GRID.

1  MONTE CARLO IMAGES IN SPES 2

REFERENCE RIGIDITY.

DISTRIBUTION IN GRID.

MONTE-CARLO OBJECT WITH MOMENTUM GRID.

FOR P(D) (UNUSED)

SEEDS.

HISTO

REFERENCES:

HISTO OF D.

HISTO OF THETA.

HISTO OF PHI.

HISTOGRAMME DE LA COORDONNEE D

HISTOGRAMME DE LA COORDONNEE THETA

HISTOGRAMME DE LA COORDONNEE PHI

PARAMETRES PHYSIQUES DE LA DISTRIBUTION:

PARAMETRES PHYSIQUES DE LA DISTRIBUTION:

PARAMETRES PHYSIQUES DE LA DISTRIBUTION:

END

Excerpt from zgoubi.res:

Histograms of initial beam coordinates.

2 HISTO

HISTOGRAMME DE LA COORDONNEE D

HISTOGRAMME DE LA COORDONNEE THETA

HISTOGRAMME DE LA COORDONNEE PHI

TOTAL COMPTAGE : 10000 SUR 10000

NUMERO DU CANAL MOYEN : 51

COMPTAGE AU * : 2038

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 COORDONNEE THETA

HISTOGRAMME DE LA COORDONNEE PHI

TOTAL COMPTAGE : 10000 SUR 10000

NUMERO DU CANAL MOYEN : 51

COMPTAGE AU * : 2038

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
Excerpt from zgoubi.res: the final momentum resolution histogram at the spectrometer focal surface.
2 TRANSFER MATRICES ALONG A TWO-STAGE SEPARATION KAON BEAM LINE

800 MeV/c kaon beam line at BNL Alternating Gradient Synchrotron [54]. The line includes two separation Wien filters.

Layout of LEB3 beamline.

Figure 48: 800 MeV/c kaon beam line layout.
The text content appears to be a list of parameters and coefficients related to a particle physics or accelerator physics context, specifically referring to calculations of transfer coefficients. The numbers and values listed seem to be associated with various objects such as magnets, slits, and focus points, which are common in particle accelerator design. The text includes terms like "transfer coefficients," "load file," and "karm m & q," which are typically used in simulations and calculations for beam lines in high-energy physics experiments. The specific values and parameters listed are likely to be used in simulations to predict and optimize the behavior of particles as they pass through the accelerator setup.
Excerpt of zgoubi.res: first and second order transfer matrices and higher order coefficients at the end of the line

**First order coefficients (MKSA):**

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.60453</td>
<td>-4.453245E-02</td>
</tr>
<tr>
<td>-2.05368</td>
<td>3.270335</td>
</tr>
<tr>
<td>2.40935E-05</td>
<td>-8.68775E-07</td>
</tr>
<tr>
<td>1.18529E-06</td>
<td>-2.35393E-07</td>
</tr>
<tr>
<td>-0.387557</td>
<td>2.31953E-02</td>
</tr>
</tbody>
</table>

**Second order coefficients (MKSA):**

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 11 3.74</td>
<td>21 -1.78</td>
</tr>
<tr>
<td>1 12 -1.78</td>
<td>22 -530</td>
</tr>
<tr>
<td>1 13 1.399E-02</td>
<td>23 -1.308E-03</td>
</tr>
<tr>
<td>1 14 1.45E-02</td>
<td>24 -2.77E-02</td>
</tr>
<tr>
<td>1 15 0.00</td>
<td>25 0.00</td>
</tr>
<tr>
<td>1 16 36.3</td>
<td>26 12.3</td>
</tr>
</tbody>
</table>

**Higher order coefficients (MKSA):**

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y/Y3 5.84</td>
<td></td>
</tr>
<tr>
<td>Y/T3 9.40E+05</td>
<td></td>
</tr>
<tr>
<td>Y/Z3 0.71</td>
<td></td>
</tr>
<tr>
<td>Y/P3 0.42</td>
<td></td>
</tr>
<tr>
<td>T/Y3 -18607</td>
<td></td>
</tr>
<tr>
<td>T/T3 1.05E+05</td>
<td></td>
</tr>
<tr>
<td>T/Z3 -0.10</td>
<td></td>
</tr>
<tr>
<td>T/P3 5.26E-02</td>
<td></td>
</tr>
<tr>
<td>Z/Y3 32.16</td>
<td></td>
</tr>
<tr>
<td>Z/T3 18.42</td>
<td></td>
</tr>
<tr>
<td>Z/Z3 -872.5</td>
<td></td>
</tr>
<tr>
<td>Z/P3 -785.2</td>
<td></td>
</tr>
<tr>
<td>P/Y3 15.46</td>
<td></td>
</tr>
<tr>
<td>P/T3 7.53</td>
<td></td>
</tr>
<tr>
<td>P/Z3 -409.98</td>
<td></td>
</tr>
<tr>
<td>P/P3 -389.2</td>
<td></td>
</tr>
</tbody>
</table>

**First order symplectic conditions (expected values = 0):**

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 11 -0.145</td>
<td>21 2.158E-02</td>
</tr>
<tr>
<td>3 12 2.158E-02</td>
<td>22 2.14E-02</td>
</tr>
<tr>
<td>3 13 20.6</td>
<td>23 3.5E-02</td>
</tr>
<tr>
<td>3 14 0.00</td>
<td>24 0.00</td>
</tr>
<tr>
<td>3 15 0.00</td>
<td>25 0.00</td>
</tr>
<tr>
<td>3 16 -0.201</td>
<td>26 0.00</td>
</tr>
</tbody>
</table>

**Second order symplectic conditions (expected values = 0):**

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 11 -9.25E-02</td>
<td>21 1.14E-02</td>
</tr>
<tr>
<td>4 12 1.14E-02</td>
<td>22 1.04E-02</td>
</tr>
<tr>
<td>4 13 10.7</td>
<td>23 0.787</td>
</tr>
<tr>
<td>4 14 0.00</td>
<td>24 0.00</td>
</tr>
<tr>
<td>4 15 0.00</td>
<td>25 0.00</td>
</tr>
<tr>
<td>4 16 -1.207</td>
<td>26 0.00</td>
</tr>
</tbody>
</table>

**Higher order symplectic conditions (expected values = 0):**

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 11 568</td>
<td>21 -7.67</td>
</tr>
<tr>
<td>5 12 -7.67</td>
<td>22 0.00</td>
</tr>
<tr>
<td>5 13 -5.97E-02</td>
<td>23 1.28E-03</td>
</tr>
<tr>
<td>5 14 -5.97E-02</td>
<td>24 6.94E-04</td>
</tr>
<tr>
<td>5 15 0.00</td>
<td>25 0.00</td>
</tr>
<tr>
<td>5 16 -251</td>
<td>26 2.77</td>
</tr>
</tbody>
</table>
3 IN-FLIGHT DECAY IN SPES 3

Figure 49: Layout of SPES 3 spectrometer.
**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 + NEUTRINO DECAY

`ESL` 4
77.3627 STOPS ABBREVIATED MUONS.
1
1 100. 10. 245. 0.

`DIPOLE` 6
80. 208.5 TOTAL ANGLE AT, CENTRAL RADIUS RM
33. 30. 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 1E6
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 TILT ANGLE OF FOCAL PLANE.
0. 0. -49.

`HISTO` 9 TOTAL SPECTRUM (PION + MUON).
2 -170. 130. 60 1
20 `y` 1 `Q`
20 `P` 1 `P`
20 `y` 1 `S`
20 `d` 1 `S`

`HISTO` 10 PION SPATIAL SPECTRUM AT FOCAL PLANE.
2 -170. 130. 60 2
20 `P` 1 `P`

`HISTO` 11 MUON SPATIAL SPECTRUM AT FOCAL PLANE.
2 -170. 130. 60 3
20 `y` 1 `S`

`HISTO` 12 MUON MOMENTUM SPECTRUM AT FOCAL PLANE.
2 -170. 130. 60 3
20 `y` 1 `S`

`REBELOTE` (49+1) RUNS = CALCULATION OF TRAJECTORIES.
49 0.1 0 (49+1)*200
Excerpt of zgoubi.res: histograms of primary and secondary particles at focal surface of SPES3.

**HISTOGRAMME DE LA COORDONNEE Y**

DANS LA FENETRE : -1.7E+02 / 1.7E+00 (CM)

<table>
<thead>
<tr>
<th>TOTAL COMPTAGE</th>
<th>605 SUR 10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOMBRE DU CANAL MOYEN</td>
<td>55</td>
</tr>
<tr>
<td>VAL. PHYS. AU *</td>
<td>-2.5E+00 (CM)</td>
</tr>
<tr>
<td>RESOLUTION PAR CANAL</td>
<td>5.0E+00 (CM)</td>
</tr>
</tbody>
</table>

PARAMETRES PHYSIQUES DE LA DISTRIBUTION :

RESOLUTION PAR CANAL : 2.5E-02

VAL. PHYS. AU " " : 8.25E-01

COMPTAGE AU " " : 16

NUMERO DU CANAL MOYEN : 46

TOTAL COMPTAGE : 605 SUR 10000

PARTICULES SECONDAIRES

HISTOGRAMME DE LA COORDONNEE D

NORMALISE

<table>
<thead>
<tr>
<th>TOTAL COMPTAGE</th>
<th>605 SUR 10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOMBRE DU CANAL MOYEN</td>
<td>50</td>
</tr>
<tr>
<td>VAL. PHYS. AU *</td>
<td>-2.5E+00 (CM)</td>
</tr>
<tr>
<td>RESOLUTION PAR CANAL</td>
<td>5.0E+00 (CM)</td>
</tr>
</tbody>
</table>

PARAMETRES PHYSIQUES DE LA DISTRIBUTION :

RESOLUTION PAR CANAL : 2.5E-02

VAL. PHYS. AU " " : 8.25E-01

COMPTAGE AU " " : 16

NUMERO DU CANAL MOYEN : 46

TOTAL COMPTAGE : 605 SUR 10000

SIGMA = 2.2849E-01

MOYENNE = 8.1693E-01

MIN = 3.7184E-01, MAX = 1.3837E+00, MAX-MIN = 1.0119E+00

COMPTAGE = 605 PARTICULES

SIGMA = 5.4452E+01 (CM)

MOYENNE = -2.2782E+01 (CM)

MIN= -1.6687E+02, MAX = 9.4131E+01, MAX-MIN = 2.6100E+02 (CM)

COMPTAGE = 605 PARTICULES
4 USE OF THE FITTING PROCEDURE

Figure 50: 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.

```
MATCHING A SYMMETRIC QUADRUPOLE TRIPLET

'OBJEY' 1
2501.73 RIGIDITY (kg.cm)
5 11 PARTICLES GENERATED FOR USE OF MATRIX
2. 2. 2. 2. 0.001
3. 0. 0. 0. 0. 1.

'ESL' 2

200.

'QUADRUPO' 3

40. 15. -7.
0.
30.

'ESL' 4

200.

'QUADRUPO' 5

40. 15. 3.
0.
30.

'ESL' 6

200.

'QUADRUPO' 7

40. 15. -7.
0.
30.

'ESL' 8

200.

'MATRIX' 9

1 0

'FITZ' VARY B IN QUADS FOR FIT OF R12 AND R34
2
# OF VARIABLES
3 12 7.12 2. SYMMETRIC TRIPLET => QUADS #1 AND #3 ARE COUPLED
3 12 0. 2. PMTR #12 OF ELEMENTS #3, 5 AND 7 IS FIELD VALUE
2 1.8-10 # OF CONSTRAINTS, PENALTY
1 1 2 8 16.6 1. CNSTRNT #1 : R12=16.6 AFTER LAST DRIFT (LMNT #8)
1 1 3 4 8 -0. 1. CNSTRNT #2 : R34=-0.8 AFTER LAST DRIFT

'END'
```
Excerpt of zgoubi.res : first order transfer matrices prior to and after fitting.

TRANSFER MATRIX WITH STARTING CONDITIONS:

<table>
<thead>
<tr>
<th></th>
<th>5.43427</th>
<th>17.0254</th>
<th>0.00000</th>
<th>0.00000</th>
<th>0.00000</th>
<th>0.00000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.67580</td>
<td>5.43425</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td>0.00000</td>
<td>0.00000</td>
<td>-1.27003</td>
<td>-0.974288</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td>0.00000</td>
<td>0.00000</td>
<td>-0.629171</td>
<td>-1.27004</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>1.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>1.00000</td>
</tr>
</tbody>
</table>

STATE OF VARIABLES AFTER MATCHING:

<table>
<thead>
<tr>
<th>LMNT</th>
<th>VAR</th>
<th>PARAM</th>
<th>MINIMUM</th>
<th>INITIAL</th>
<th>FINAL</th>
<th>MAXIMUM</th>
<th>STEP</th>
<th>NAME</th>
<th>LBL1</th>
<th>LBL2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>12</td>
<td>-21.0</td>
<td>-7.00</td>
<td>-6.972765137</td>
<td>7.00</td>
<td>1.707E-04</td>
<td>QUADRUPO</td>
<td>3</td>
<td>*</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>12</td>
<td>-3.00</td>
<td>3.00</td>
<td>3.229344585</td>
<td>9.00</td>
<td>1.266E-04</td>
<td>QUADRUPO</td>
<td>5</td>
<td>*</td>
</tr>
</tbody>
</table>

STATUS OF CONSTRAINTS (Target penalty = 1.0000E-10)

<table>
<thead>
<tr>
<th>TYPE</th>
<th>I</th>
<th>J</th>
<th>LMNT#</th>
<th>DESIRED WEIGHT</th>
<th>REACHED KI2</th>
<th>* Parameter(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>1.6600000E+01</td>
<td>1.0000E+00</td>
<td>1.6599981E+01</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
<td>8</td>
<td>-8.8000000E-01</td>
<td>1.0000E+00</td>
<td>-8.8000964E-01</td>
</tr>
</tbody>
</table>

Fit reached penalty value 8.4374E-11

MATRIX, WITH NEW VARIABLES:

<table>
<thead>
<tr>
<th></th>
<th>5.27056</th>
<th>16.6000</th>
<th>0.00000</th>
<th>0.00000</th>
<th>0.00000</th>
<th>0.00000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.61443</td>
<td>5.27450</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td>0.00000</td>
<td>0.00000</td>
<td>-1.24205</td>
<td>-0.880010</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td>0.00000</td>
<td>0.00000</td>
<td>-0.622553</td>
<td>-1.24620</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>1.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>1.00000</td>
</tr>
</tbody>
</table>
5 MULTITURN SPIN TRACKING IN SATURNE 3 GeV SYNCHROTRON

Figure 51: 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 $Z_0 = 4.58 \times 10^{-3}$ 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, $\alpha_Z = 0.444$, $\gamma_Z = 0.582$ m$^{-1}$, $\varepsilon_Z / \pi = 12 \times 10^{-6}$ m.rad in agreement with matrix calculations.

(C) Fractional tune numbers obtained by Fourier analysis for $\varepsilon_Y / \pi = \varepsilon_Z / \pi \approx 12 \times 10^{-6}$ m.rad: $\nu_Y = 0.63795$, $\nu_Z = 0.60912$ (the integer part is 3 for both).

(D) Longitudinal phase-space (“$DP_{\text{phase}}$” in Zgoubi notations): particles with initial momentum dispersion of $5 \times 10^{-4}$ (1), $1 \times 10^{-3}$ (2), $1.65 \times 10^{-3}$ (3) (out of acceptance), are accelerated at 1405 eV/turn ($\dot{B} = 2.1$ T/s); analytical calculations give accordingly momentum acceptance of $1.65 \times 10^{-3}$.

Figure 52: Crossing of $\gamma G = 7 - \nu_Z$, at $\dot{B} = 2.1$ T/s.

(A) $\varepsilon_Z / \pi = 12.2 \times 10^{-6}$ m.rad. The strength of the resonance is $| \varepsilon | = 3.3 \times 10^{-4}$. As expected from the Froissart-Stora formula [57] the asymptotic polarization is about 0.44.

(B) The emittance is now $\varepsilon_Z / \pi = 1.2 \times 10^{-6}$ 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 $10^{-3}$. 


SATURN. CROSSING Gamma=7-MHz, Hz=1.60877 (perturbed)

"SORTER" 5015.388 834.04 MeV, proton

2.6E-02 6.58E-02 ,1.00 "o" Epillon/Yp = 0. (Closed orbit)

0.570 0.390 ,1.005 "o" 1.047 0.180 ,1.001 "o"

14.04 1.05 ,1.016 "o"

1 1 1 1 "SCALING"

1 4 "MULTIPOL"

5015.388-3 5034.391E-3

FROM 834.041 TO 838.077 MeV

"QUADRUPO" QP 1 21

71.6256

曝1|曝2|曝3|曝4

2999 0.1 99 TOTAL NUMBER OF TURNS = 3000

6000. 0. SIN(phs) = .234162, dE=1.40497 keV/Turn.

20.8. .0427050667

4.2481 1.8639 -5.572 -3904 0. 0. 0.

20.8. .0427050667 20.8.

4.2481 1.8639 -5.572 -3904 0. 0. 0.

30[120]30 bend 3 0 0 .0 .-1.634595048

JSI SD 2

71.6256 "BEND" DIP 3 4 3

8

247.30309 0. 1.57776

20.8 .0427050667

4.2481 1.8639 -5.572 -3904 0. 0. 0.

20.8 .0427050667 20.8.

4.2481 1.8639 -5.572 -3904 0. 0. 0.

30[120]30 bend 3 0 0 .0 .-1.634595048

JSI SD 2

71.6256 "BEND" DIP 3 4 3

9

48.4273 10. 1.76533 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

6.1122 6.2671 -1.4982 3.5882 -2.1209 1.723

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

6.1122 6.2671 -1.4982 3.5882 -2.1209 1.723

30[50]30 Quad

1 0. 0.

"JIS" SD 2

71.6256 "BEND" DIP 3 4 3

10

247.30309 0. 1.57776

20.8 .0427050667

4.2481 1.8639 -5.572 -3904 0. 0. 0.

20.8 .0427050667 20.8.

4.2481 1.8639 -5.572 -3904 0. 0. 0.

30[120]30 bend 3 0 0 .0 .-1.634595048

JSI SD 2

71.6256 "BEND" DIP 3 4 3

13

46.723 10. 763495

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

6.1122 6.2671 -1.4982 3.5882 -2.1209 1.723

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

6.1122 6.2671 -1.4982 3.5882 -2.1209 1.723

30[50]30 Quad

1 0. 0.

"JIS" SD 2

71.6256 "BEND" DIP 3 4 3

30

247.30309 0. 1.57776

20.8 .0427050667

4.2481 1.8639 -5.572 -3904 0. 0. 0.

20.8 .0427050667 20.8.

4.2481 1.8639 -5.572 -3904 0. 0. 0.

30[120]30 bend 3 0 0 .0 .-1.634595048

JSI SD 2

71.6256 "BEND" DIP 3 4 3

31

283
Figure 53: 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 \( \pm 3 \times 10^{-4} \).

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.
MICROBEAM LINE, WITH A MAGNETO-ELECTROSTATIC QUADRUPOLE DUOSET.

"MCOBJET" RANDOM OBJECT DEFINITION 1

RIGIDITY (20keV PROTONS). 1 DISTRIBUTION IN WINDOW.
200 NUMBER OF PARTICLES.
UNIFORM DISTRIBUTION.
1 1 1 1 1 1 CENTRAL VALUE, AND
HALF WIDTH OF DISTRIBUTION.
10 10 10 10 10 10 FOR P(D) - UNIFORM.
9939.57 549720 472584 SEEDS.

"PARTICUL" PARTICLE MASS AND CHARGE 2

938.2723 1.60217733E-19 FOR INTEGRATION IN E-FIELD.

"DRIFT" DRIFT. 3

500.

"EBMULT" FIRST MAGNETO-ELECTROSTATIC 4 QUADRUPOLE.

10.2 10. 0. -9272.986 0. 0. 0. 0. 0. 0. ELECTRIC Q-POLE COMPONENT.

10.2 10. 0. 1.89493 0. 0. 0. 0. 0. 0. MAGNETIC Q-POLE COMPONENT.

"DRIFT" DRIFT. 6

4.9

"EBMULT" SECOND MAGNETO-ELECTROSTATIC 7 QUADRUPOLE.

10.2 10. 0. 13779.90 0. 0. 0. 0. 0. 0. RAYS ARE STORED IN rays.out FOR FURTHER PLOTTING.

19 0.1 0 A TOTAL OF 200 * (19+1) PARTICLES.

"END" 8

---

zgoubi.data file.

zgoubi.res file.

---

Multiple pass, from element # 1 : MCOBJET /label1=RANDOM /label2=OBJECT to REBELOTE /label1=RANDOM /label2=OBJECT ending at pass # 19 through the optical structure

Total of 3800 particles have been launched

---

1 MCOBJET RANDOM OBJECT

Reference magnetic rigidity = 20.435 KXm

Object built up of 200 particles

Distribution in a Window

Central values (MKSA units):

Vo, To, Po, To, SO, No, Woch : 0.000 0.000 0.000 0.000 0.000 1.000

Cut-offs ( +/- Widths ) :

Ny, Nt, Nz, Np, Nx, NBR/BORO : 1.00 1.00 1.00 1.00 1.00 1.00

Type of sorting :

Y, T, Z, P, X, D : Uniform Uniform Uniform Uniform Uniform Uniform

---

2 PARTICUL PARTICLE MASS

---

Reference data:

mass, rigidity (kG.cm) : 20.435000 -p/q, such that dev.=B * L/rigidity
mass (MeV/c2) : 938.2720
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) : 3.9999376335E-02 -T[eV]=(gamma+1)/gamma, such that dev.=E/L/rigidity

---
### Micro-Beam Focusing with $B \times E$ Quadrupoles

<table>
<thead>
<tr>
<th>Trajectory</th>
<th>Length (cm)</th>
<th>Cumulative Length of Optical Axis (m)</th>
<th>Time (for ref. rigidity &amp; particle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 drift</td>
<td>50.000000</td>
<td>0.00000000</td>
<td>2.53635E-06</td>
</tr>
<tr>
<td>4 drift</td>
<td>59.000000</td>
<td>0.00000000</td>
<td>2.85698E-06</td>
</tr>
<tr>
<td>5 EBMULT FIRST ELECTRO-MAGNET</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multipoles</td>
<td>Length: 10.200000 cm</td>
<td>E-DIPOLE: 0.0000000E+00 V/m</td>
<td></td>
</tr>
<tr>
<td>entrance</td>
<td>B-DIPOLE: 0.0000000E+00 kG</td>
<td>E-QUADRUPOLE: 1.8949300E+00 kG</td>
<td></td>
</tr>
<tr>
<td>exit</td>
<td>E-QUADRUPOLE: 1.8949300E+00 kG</td>
<td>E-QUADRUPOLE: 1.8949300E+00 kG</td>
<td></td>
</tr>
<tr>
<td>6 drift</td>
<td>4.900000</td>
<td>0.00000000</td>
<td>2.93298E-06</td>
</tr>
<tr>
<td>7 EBMULT SECOND ELECTRO-MAGNET</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multipoles</td>
<td>Length: 10.200000 cm</td>
<td>E-DIPOLE: 0.0000000E+00 V/m</td>
<td></td>
</tr>
<tr>
<td>entrance</td>
<td>B-DIPOLE: 0.0000000E+00 kG</td>
<td>E-QUADRUPOLE: -2.8159200E+00 kG</td>
<td></td>
</tr>
<tr>
<td>exit</td>
<td>E-QUADRUPOLE: -2.8159200E+00 kG</td>
<td>E-QUADRUPOLE: -2.8159200E+00 kG</td>
<td></td>
</tr>
</tbody>
</table>

### Notes
- Drifts have lengths of 50.000000 cm, 59.000000 cm, 4.900000 cm, and 5.69200000 cm, 5.84300000 cm, respectively.
- The entrance and exit field models are sharp edge, FINTE, FINTS, gap: 0.0000E+00 0.0000E+00 5.0000E+00.
- The integration step is 0.8000 cm for drifts 3, 4, and 6.
- The integration step is 0.8000 cm for drift 5.
- The cumulated length of the optical axis is calculated at each drift.
- The time (for ref. rigidity & particle) is given for each trajectory.
HISTOGRAMME DE LA COORDONNÉE V
PARTICULS PRIMAIRES ET SECONDAIRES
DANS LA FENETRE : -5.0000E-06 / 5.0000E-06 (CM)
NORMALISÉ

TOTAL COMPTAGE : 4000 SUR 4000
NUMÉRO DU CANAL MOYEN : 51
COMPTAGE AU " " : 109
VAL. PHYS. AU " " : 0.000E+00 (CM)
RESOLUTION PAR CANAL : 1.667E-07 (CM)
PARAMÈTRES 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,P,S,IEX : 1.0002E+00 9.0257E-07 -2.3996E-01 -1.0770E-06 1.7947E+00 6.09300E+02 1

HISTOGRAMME DE LA COORDONNÉE Z
PARTICULS PRIMAIRES ET SECONDAIRES
DANS LA FENETRE : -5.0000E-06 / 5.0000E-06 (CM)
NORMALISÉ

TOTAL COMPTAGE : 4000 SUR 4000
NUMÉRO DU CANAL MOYEN : 51
COMPTAGE AU " " : 169
VAL. PHYS. AU " " : 0.000E+00 (CM)
RESOLUTION PAR CANAL : 1.667E-07 (CM)
PARAMÈTRES 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,P,S,IEX : 1.0002E+00 9.0257E-07 -2.3996E-01 -1.0770E-06 1.7947E+00 6.09300E+02 1

--------- FIN D'EFFET DE 'REBELOTE' ---------
IL Y A EU 20 PASSAGES DANS LA STRUCTURE
PARTICULS ENVOYÉES : 4000

Rebelote Run Again.
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 [58]. 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.
- 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).

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.
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.
References

F. Méot, Generalization of the Zgoubi method for ray-tracing to include electric fields, NIM A 340 (1994) 594-604.
[13] F. Méot, Spin tracking simulations in AGS based on ray-tracing methods, Tech. Note C-AD/AP/452, BNL C-AD (2009);
[28] L. Farvacque et al., Beta user’s guide, Note ESRF-COMP-87-01, 1987;
J. Payet, IRF/LNS, CEA Saclay, private communication;
[36] Files developed by Simon White, January 2012. Including beam-beam spin kick after Ref. [37].


outpoi.lis, 128


PICKUPS, 156, 162, 236

PLOTDATA, 157, 237

POISSON, 128, 238

POLARMES, 129, 166, 239

positioning of optical elements, 163

PS170, 130, 240

QUADISEX, 24, 25, 131, 241

QUADRUPO, 25, 57, 58, 68, 95, 100, 115, 126, 127, 132, 135, 163, 242

REBELOTE, 46, 51, 54, 57, 57, 70, 72, 92, 149, 152, 156, 158, 162, 163, 166, 209, 229, 244, 252, 269

reference rigidity, 46, 49, 73, 92, 161

RESET, 71, 245

SCALING, 59, 72, 92, 93, 162, 163, 188, 245, 246

SEPARA, 134, 247

SEXQUAD, 24, 131, 248

SEXTUPO, 25, 68, 135, 243, 249

SOLENOID, 136, 251

spin kick, beam-beam, 87, 182

spin tracking, 33, 70, 74, 96, 141, 147, 152, 158, 166, 218, 244, 253, 269

spin, \( \vec{n}_0 \) vector, 62

SPINR, 137, 250

SPNPRNL, 158, 252, 292

SPNPRT, 76, 158, 252

SPNSTORE, 158, 252, 292

SPNTRK, 54, 70, 74, 166, 245, 253

SRLOSS, 76, 254, 255

SRPRNT, 159, 255

stopped particles, 70, 71, 95, 99, 149, 166, 189, 192

storage files, 291

synchrotron motion, 70, 72, 92, 188, 244, 246

synchrotron radiation, 37, 292

synchrotron radiation loss, 76, 159, 254

synchrotron radiation spectra, 77, 256

SYNRAD, 77, 256

tag, symmetry index, 49

tag, tag character, 47, 49, 50, 66, 77, 148, 149, 152, 158

TARGET, 98, 191

time varying fields, 161

TOSCA, 24, 58, 68, 138, 162, 163, 257, 269, 292

TRANSMAT, 140, 258

TRAROT, 141, 259

TWISS, 51, 160, 260

UNDULATOR, 142, 261

UNIPOT, 24, 143, 262

variable (FIT, FIT2), 57, 57, 213

VENUS, 24, 25, 144, 263

WIENFILT, 25, 145, 264

XCE, 163

XPAS