Zgoubi Users’ Guide

F. Meot

Collider-Accelerator Department
Brookhaven National Laboratory
Upton, NY 11973

Notice: This document has been authorized by employees of Brookhaven Science Associates, LLC under Contract No. DE-AC02-98CH10886 with the U.S. Department of Energy. The United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this document, or allow others to do so, for United States Government purposes.
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
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].
PART A Description of software contents

GLOSSARY OF KEYWORDS

OPTICAL ELEMENTS VERSUS KEYWORDS

PREFACE TO THE BNL EDITION (2012)

INTRODUCTION TO THE 4th EDITION (1997)

INTRODUCTION

1 NUMERICAL CALCULATION OF MOTION AND FIELDS

1.1 zgoubi Frame

1.2 Integration of the Lorentz Equation

1.2.1 Integration in Magnetic Fields

1.2.2 Integration in Electric Fields

1.2.3 Integration in Combined Electric and Magnetic Fields

1.2.4 Calculation of the Time of Flight

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

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

1.3.2 Extrapolation from 1-D axial field map

1.3.3 Extrapolation From Median Plane Field Models

1.3.4 Extrapolation from Arbitrary 2-D Field Maps

1.3.5 Interpolation in 3-D Field Maps

1.3.6 2-D Analytical Field Models and Extrapolation

1.3.7 3-D Analytical Models of Fields

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

1.4.1 1-D Axial Map, with Cylindrical Symmetry

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

1.4.3 Arbitrary 2-D Map, no Symmetry

1.4.4 3-D Field Map

2 SPIN TRACKING

2.1 Introduction

2.2 Integration in Magnetic Fields

2.3 Integration in Electric Fields

2.4 Integration in Combined Electric and Magnetic Fields

3 SYNCHROTRON RADIATION

3.1 Energy Loss and Dynamical Effects

3.2 Spectral-Angular Radiated Densities

3.2.1 Calculation of the Radiated Electric Field

3.2.2 Calculation of the Fourier Transform of the Electric Field

4 DESCRIPTION OF THE AVAILABLE PROCEDURES

4.1 Introduction

4.2 Definition of an Object

4.3 Declaring Options

4.4 Optical Elements and Related Numerical Procedures

4.5 Output Procedures

4.6 Complements Regarding Various Functionalities

4.6.1 Reference rigidity

4.6.2 Time Varying Fields

4.6.3 Backward Ray-Tracing

4.6.4 Checking Fields and Trajectories Inside Optical Elements

4.6.5 Labeling Keywords

4.6.6 Multi-turn Tracking in Circular Machines

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

4.6.8 Coded Integration Step
PART B  Keywords and input data formatting  

GLOSSARY OF KEYWORDS  

OPTICAL ELEMENTS VERSUS KEYWORDS  

INTRODUCTION  

PART C  Examples of input data files and output result files  

INTRODUCTION  

1  MONTE CARLO IMAGES IN SPES 2  

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

3  IN-FLIGHT DECAY IN SPES 3  

4  USE OF THE FITTING PROCEDURE  

5  MULTITURN SPIN TRACKING IN SATURNE 3 GeV SYNCHROTRON  

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

PART D  Running zgoubi and its post-processor/graphic interface zpop  

INTRODUCTION  

1  GETTING TO RUN zgoubi AND zpop  

1.1  Making the Executable Files zgoubi and zpop  

1.1.1  The transportable package zgoubi  

1.1.2  The post-processor and graphic interface package zpop  

1.2  Running zgoubi  

1.3  Running zpop  

2  STORAGE FILES  

REFERENCES  

INDEX  

REFERENCES  

INDEX
PART A

Description of software contents
Glossary of Keywords

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

\[
\frac{d(m\vec{v})}{dt} = q(\vec{e} + \vec{v} \times \vec{b})
\]  

(1.2.1)

Figure 1: Reference frame and coordinates \((Y, T, Z, P)\) in zgoubi.
- \(OX\) : in the direction of motion,
- \(OY\) : normal to \(OX\),
- \(OZ\) : orthogonal to the \((X, Y)\) plane,
- \(\vec{W}\) : projection of the velocity, \(\vec{v}\), in the \((X, Y)\) plane,
- \(T\) = angle between \(\vec{W}\) and the \(X\)-axis,
- \(P\) = angle between \(\vec{W}\) and \(\vec{v}\).
Noting \( (\cdot)' = \frac{d(\cdot)}{ds} \) and taking
\[
\vec{u} = \frac{\ddot{v}}{v}, \quad ds = v \, dt, \quad \vec{u}' = \frac{d\vec{u}}{ds}, \quad m\vec{v} = mv\vec{u} = q \, B\rho \, \ddot{u}
\] (1.2.2)

where \( B\rho \) is the rigidity of the particle, this equation can be rewritten
\[
(B\rho)' \, \vec{u} + B\rho \, \vec{u}' = \frac{\vec{e}}{v} + \vec{u} \times \vec{b}
\] (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)
\[
\vec{R}(M_1) \approx \vec{R}(M_0) + \vec{u}(M_0) \, \Delta s + \vec{u}'(M_0) \, \frac{\Delta s^2}{2!} + \cdots + \vec{u}'''(M_0) \, \frac{\Delta s^6}{6!}
\] (1.2.4)

The rigidity at \( M_1 \) is obtained in the same way from
\[
(B\rho)(M_1) \approx (B\rho)(M_0) + (B\rho)'(M_0) \, \Delta s + (B\rho)''(M_0) \, \frac{\Delta s^2}{2!} + \cdots + (B\rho)''''(M_0) \, \frac{\Delta s^6}{5!}
\] (1.2.5)

The equation of time of flight is written in a similar manner
\[
T(M_1) \approx T(M_0) + T'(M_0) \, \Delta s + T''(M_0) \, \frac{\Delta s^2}{2!} + \cdots + T''''(M_0) \, \frac{\Delta s^6}{5!}
\] (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.

Figure 2: Position and velocity of a particle in the reference frame.
1.2 Integration of the Lorentz Equation

1.2.1 Integration in Magnetic Fields

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

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

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

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

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

From

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

and by successive differentiation, we get

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

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

\[
(B^0)' \vec{u} + B^0 \vec{u}' = \frac{\vec{e}}{v} \quad (1.2.11)
\]

which, by successive differentiations, gives the recursive relations
\[(B\rho)'\ddot{u} + B\rho \dot{u}' = \frac{\vec{e}}{v}\]

\[(B\rho)''\ddot{u} + 2(B\rho)'^{'}\dot{u}' + B\rho\dddot{u} = \left(\frac{1}{v}\right)' \vec{e} + \frac{\vec{e}'}{v}\]

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

(1.2.12)

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

\[(B\rho)''''''\ddot{u} + 5(B\rho)''''\dot{u}'''' + \text{etc.}\]

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

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

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

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

(1.2.13)

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

\[\vec{u}''''''' = \left(\frac{1}{v}\right)'''' \vec{E} + 4 \left(\frac{1}{v}\right)''' \vec{E}' |_{B\rho} + 6 \left(\frac{1}{v}\right)'' \vec{E}'' |_{B\rho} + 4 \left(\frac{1}{v}\right)' \vec{E}''' |_{B\rho} + \frac{1}{v} \vec{E}''''' |_{B\rho} - 5 \frac{(B\rho)'}{B\rho} \vec{u}''''' - 10 \frac{(B\rho)''}{B\rho} \vec{u}''' - 10 \frac{(B\rho)'''}{B\rho} \vec{u}'' - 5 \frac{(B\rho)''''}{B\rho} \vec{u}'' - \frac{(B\rho)'''''}{B\rho} \vec{u}\]

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

\[d\vec{E} = \frac{\partial \vec{E}}{\partial X} dX + \frac{\partial \vec{E}}{\partial Y} dY + \frac{\partial \vec{E}}{\partial Z} dZ\]  

(1.2.14)

by successive differentiations
\[ \mathbf{E}' = \sum_i \frac{\partial \mathbf{E}}{\partial X_i} u_i \]

\[ \mathbf{E}'' = \sum_{ij} \frac{\partial^2 \mathbf{E}}{\partial X_i \partial X_j} u_i u_j + \sum_i \frac{\partial \mathbf{E}}{\partial X_i} u'_i \] (1.2.15)

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

e tc. 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

\[ \frac{dp^2}{dt} = \frac{dp^2}{dt} \quad \text{i.e.,} \quad \frac{dp}{dt} p = \frac{dp}{dt} \vec{p} \] (1.2.16)

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

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

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

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

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

\[ (B\rho)''' = \left( \frac{1}{v} \right)'' (\vec{e} \cdot \vec{u}) + 2 \left( \frac{1}{v} \right)' (\vec{e} \cdot \vec{u})' + \frac{1}{v} (\vec{e} \cdot \vec{u})'' \] (1.2.18)

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

\[ (B\rho)'''''' = \text{etc.} \]

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

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

\[ (\vec{e} \cdot \vec{u})'' = \vec{e}''' \cdot \vec{u} + 2 \vec{e}'' \cdot \vec{u}' + \vec{e} \cdot \vec{u}'' \] (1.2.19)

e tc.

Note that they can be related to the derivatives of the kinetic energy \(W\) by \(dW = \frac{dp}{dt} \cdot \vec{v} dt = q\vec{e} \cdot \vec{v} dt\) which leads to
\[
\frac{d^{n+1}W}{ds^{n+1}} = q \frac{d^n(\vec{e} \cdot \vec{u})}{ds^n}
\]  
(1.2.20)

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

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

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

\[
\begin{align*}
\left( \frac{1}{v} \right)' &= \frac{1}{c^2} \left( W + m_0 c^2 \right) - q B \rho \\
\left( \frac{1}{v} \right)'' &= \frac{1}{c^2} \left( \frac{\vec{e} \cdot \vec{u}'}{B \rho} - \frac{1}{v} \frac{\left( B \rho \right)'}{B \rho} \right) \\
\left( \frac{1}{v} \right)''' &= \frac{1}{c^2} \left( \frac{\vec{e} \cdot \vec{u}''}{B \rho} - 2 \left( \frac{1}{v} \right) \frac{\left( B \rho \right)'}{B \rho} - \frac{1}{v} \frac{\left( B \rho \right)''}{B \rho} \right) \\
\left( \frac{1}{v} \right)'''' &= \frac{1}{c^2} \left( \frac{\vec{e} \cdot \vec{u}'''}{B \rho} - 3 \left( \frac{1}{v} \right) \frac{\left( B \rho \right)'}{B \rho} - 3 \left( \frac{1}{v} \right) \frac{\left( B \rho \right)''}{B \rho} - \frac{1}{v} \frac{\left( B \rho \right)'''}{B \rho} \right)
\end{align*}
\]  
(1.2.21)

1.2.3 Integration in Combined Electric and Magnetic Fields

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

\[
\begin{align*}
(B \rho)'' \vec{u}' + B \rho \vec{u}'' &= \vec{e} + \vec{u} \times \vec{b} \\
(B \rho)'''' \vec{u}'' + 3(B \rho)'' \vec{u}'' + B \rho \vec{u}''' &= \left( \frac{1}{v} \right)'' \vec{e} + \left( \frac{1}{v} \right)'' \vec{e}' + \left( \frac{1}{v} \right)'' \vec{e}'' + \left( \vec{u} \right) (\vec{u} \times \vec{b})' \\
(B \rho)''''' \vec{u}''' + 6(B \rho)''' \vec{u}''' + 4(B \rho)'' \vec{u}''' + B \rho \vec{u}''''' &= \left( \frac{1}{v} \right)''''' \vec{e} + 3 \left( \frac{1}{v} \right)''''' \vec{e}' + 3 \left( \frac{1}{v} \right)''''' \vec{e}'' + \left( \vec{u} \right) (\vec{u} \times \vec{b})''''
\end{align*}
\]  
(1.2.22)

that provide the derivatives \( \frac{d^n \vec{a}}{ds^n} \) needed in the Taylor expansions (1.2.4)
1.3 Calculation of $\vec{E}$ and $\vec{B}$ Fields and their Derivatives

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

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

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

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

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

\[ \vec{E}^{(n)} \bigg|_{B_\rho} = \frac{1}{B_\rho} \frac{d^n \vec{e}}{ds^n} \text{ and } (\vec{u} \times \vec{B})^{(n)} \bigg|_{B_\rho} = \frac{1}{B_\rho} (\vec{u} \times \vec{b})^{(n)}. \] (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 $dT/ds = 1/v, d^2T/ds^2 = d(1/v)/ds$, 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

\[
\begin{align*}
B_X(X, r) &= B_X(X, 0) - \frac{r^2 \partial^2 B_X}{4 \partial X^2} (X, 0) + \frac{r^4 \partial^4 B_X}{64 \partial X^4} (X, 0) \\
B_r(X, r) &= -\frac{r \partial B_X}{2 \partial X} (X, 0) + \frac{r^3 \partial^3 B_X}{16 \partial X^3} (X, 0) - \frac{r^5 \partial^5 B_X}{384 \partial X^5} (X, 0)
\end{align*}
\]

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

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

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 \( \vec{E} \) and \( \vec{B} \) Fields and their Derivatives

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

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 \textit{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. \( \vec{B} \) and its derivatives at any point \((X, Y, Z)\) are calculated by polynomial interpolation followed by Taylor expansions in \( Z \), without any hypothesis of symmetries (see section 1.4.3 and keywords \textit{MAP2D}, \textit{MAP2D-E}).

1.3.5 Interpolation in 3-D Field Maps [8]

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

1.3.6 2-D Analytical Field Models and Extrapolation

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

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

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

Multipoles

The scalar potential used for the calculation of \( \frac{\partial^{i+j+k} \vec{B}_n(X, Y, Z)}{\partial X^i \partial Y^j \partial Z^k} \) (\( i + j + k = 0 \) to 4) in the case of magnetic and electro-magnetic multipoles with \( 2n \) poles (namely, \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]

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

where \( G(X) \) is the longitudinal gradient, defined at the entrance or exit of the optical element by
\[ G(s) = \frac{G_0}{1 + \exp(P(s))}, \quad G_0 = \frac{B_0}{R_0^n} \tag{1.3.6} \]

where

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

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 \( \Theta_n \) around the \( X \)-axis. If so, the calculation of the field and derivatives in the rotated axis \((X, Y, Z_R)\) 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 \( \Theta_n \).

In particular a skew \( 2n \)-pole component is created by taking \( \Theta_n = \pi/2n \).

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

\[ B(X, 0) = A_0 + A_1 X + A_2 X^2 + A_3 X^3 + A_4 X^4 + A_5 X^5 \tag{1.4.1} \]

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

\[ S = \sum_i (B(X, 0) - B_i)^2 \tag{1.4.2} \]

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

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

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).

\[
\begin{align*}
\text{(A)} & \quad \text{Cartesian mesh, 9-point interpolation grid.} \\
\text{(B)} & \quad \text{Cartesian mesh, 25-point interpolation grid.} \\
\text{(C)} & \quad \text{Polar mesh and moving Cartesian frame.}
\end{align*}
\]

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.

\( \text{A} \) : Cartesian mesh, 9-point interpolation grid.
\( \text{B} \) : Cartesian mesh, 25-point interpolation grid.
\( \text{C} \) : Polar mesh and moving Cartesian frame.

\textbf{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

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

to the second order, or

\[
B(X, Y, 0) = A_{00} + A_{10}X + A_{01}Y + A_{20}X^2 + A_{11}XY + A_{02}Y^2 + A_{30}X^3 + A_{21}X^2Y + A_{12}XY^2 + A_{03}Y^3 + A_{40}X^4 + A_{31}X^3Y + A_{22}X^2Y^2 + A_{13}XY^3 + A_{04}Y^4
\]

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

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

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

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

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)

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

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

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, DIPOLES[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 map is assumed to describe the field is obtained by means of (parameter $B$ the components $(\mathbf{E}, \mathbf{B})$ polynomial interpolation from a $3 \times 3$ grid. 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 $\vec{B}(B_X, B_Y, B_Z)$ 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 $\vec{B}$ and its derivatives at the projection $(X, Y, Z_0)$ of the actual position $(X, Y, Z)$ of a particle is obtained by means of (parameter $IORDRE$ in keyword data list - see for instance MAP2D, MAP2D-E) either a second degree polynomial interpolation from a $3 \times 3$ points grid ($IORDRE=2$), or a fourth degree polynomial interpolation from a $5 \times 5$ points grid ($IORDRE=4$), centered at the node $(i,j)$ closest to the position $(X, Y)$.

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

\[
\begin{align*}
\frac{\partial^3 B}{\partial X^3} &= 1 \frac{\partial^4 B}{\partial X^4} + 3 \frac{\partial^2 B}{\partial X^2} - 2 \frac{\partial B}{\partial X} \\
\frac{\partial^3 B}{\partial Y^3} &= 1 \frac{\partial^4 B}{\partial Y^4} + 3 \frac{\partial^2 B}{\partial Y^2} - 2 \frac{\partial B}{\partial Y} \\
\frac{\partial^4 B}{\partial X^4} &= 1 \frac{\partial^4 B}{\partial X^4} + 6 \frac{\partial^2 B}{\partial X^2} + 3 \frac{\partial B}{\partial X} \\
\frac{\partial^4 B}{\partial Y^4} &= 1 \frac{\partial^4 B}{\partial Y^4} + 6 \frac{\partial^2 B}{\partial Y^2} + 3 \frac{\partial B}{\partial Y}
\end{align*}
\]

**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 $-1$. 
To second order for instance

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

(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.

(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).

![3-D 27-point grid](image)

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,

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

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

\[ S = \sum_{ijk} \left( B_\ell(X, Y, Z) - B_{\ell ijk} \right)^2 \]  
(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]

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

with, in the laboratory frame,

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

t, q, m, c, $\gamma$ and $G$ are respectively the time, charge, relativistic mass, speed of light, Lorentz relativistic factor and anomalous magnetic moment of the particle, $\vec{b}$ and $\vec{e}$ are the fields in the laboratory, $\vec{b}_f$ 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

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

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

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

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

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

2.2 Integration in Magnetic Fields

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

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

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

(2.2.2)

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

\[ \vec{B}_i = (\vec{B} \cdot \vec{u}) \vec{u} \]
\[ \vec{B}'_i = (\vec{B}' \cdot \vec{u} + \vec{B} \cdot \vec{u}'') \vec{u} + (\vec{B} \cdot \vec{u}) \vec{u}' \]
\[ \vec{B}''_i = (\vec{B}'' \cdot \vec{u} + 2\vec{B}' \cdot \vec{u}' + \vec{B} \cdot \vec{u}'') \vec{u} + 2(\vec{B}' \cdot \vec{u} + \vec{B} \cdot \vec{u}'') \vec{u}' + (\vec{B} \cdot \vec{u}) \vec{u}'' \]

(2.2.3)

The quantities \( \vec{u}, \vec{B} \) and their derivatives as involved in these equations are known, being sub-products of the integration of the motion of the particle, eqs. (1.2.8, 1.2.10) p. 19.

2.3 Integration in Electric Fields

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

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

(2.3.1)

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

\[ B\rho \vec{S}' = \vec{S} \times \vec{\omega}_e \]
\[ (B\rho)' \vec{S}' + B\rho \vec{S}'' = \vec{S}' \times \vec{\omega}_e + \vec{S} \times \vec{\omega}_e' \]
\[ (B\rho)' \vec{S}'' + 2(B\rho)' \vec{S}'' + B\rho \vec{S}''' = \vec{S}''' \times \vec{\omega}_e + 2\vec{S}' \times \vec{\omega}_e' + \vec{S} \times \vec{\omega}_e'' \]

(2.3.2)

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

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

(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,

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

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

etc.

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

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

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

### 2.4 Integration in Combined Electric and Magnetic Fields

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

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

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

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

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

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

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

Magnetic field component, eq. (2.2.2)  
Electric field component, eq. (2.3.2)

etc.

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 miniwiggler [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

$$p(k) = \frac{\lambda^{-k}}{k!} \exp(-\lambda)$$  \hspace{1cm} (3.1.1)

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 \( p(\epsilon/\epsilon_c)de/\epsilon_c \) (i.e., the probability of emission of a photon with energy in \([0, \epsilon]\)) writes

\[
P(\epsilon/\epsilon_c) = \frac{3}{5\pi} \int_0^{\epsilon/\epsilon_c} K_{5/3}(x)dx
\]

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

\[
P(\epsilon/\epsilon_c) = \frac{12\sqrt{3}}{5 \Gamma^{1/3}(\frac{1}{3})} \frac{\epsilon^{1/3}}{\epsilon_c}
\]

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 \( \epsilon/\epsilon_c = \left( \frac{5^{2/3} \Gamma^{1/3}(\frac{1}{3})}{12\sqrt{3}P} \right) \) 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 \( \Delta E(eV) = \frac{2}{3} r_0 c^3 B(T)\Delta \theta \), 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 \( B_0 \) (as defined with, e.g., [MC]OBJET)
- critical energy \( \epsilon_c(eV) = \frac{3\gamma^3 c \hbar}{2\rho e} \), with \( \rho = B_0/B \)
- average energy of the photons radiated \( \langle \epsilon \rangle = \frac{8}{15\sqrt{3}}\epsilon_c \)
- rms energy of radiated photons \( \epsilon_{rms} = 0.5591\epsilon_c \)
- number of radiated photons per particle \( N = \Delta E/\langle \epsilon \rangle \).

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

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.

![Diagram](image)

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

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

3.2.1 Calculation of the Radiated Electric Field

The expression for the radiated electric field \(\vec{E}(\vec{n}, \tau)\) as seen by the observer in the long distance approximation is [26]

\[
\vec{E}(\vec{n}, \tau) = \frac{q}{4\pi\varepsilon_0 c} \frac{\vec{n}(t) \times \left[ (\vec{n}(t) - \vec{\beta}(t)) \times d\vec{\beta}/dt \right]}{r(t) \left( 1 - \vec{n}(t) \cdot \vec{\beta}(t) \right)^{3/2}}
\]  \hspace{1cm} (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.
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 \( \vec{e} = 0 \))

\[
\frac{d\vec{\beta}}{dt} = \frac{q}{m} \vec{\beta}(t) \times \vec{b}(t) \tag{3.2.3}
\]

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

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

As an illustration, Fig. 6 has been produced using \texttt{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

\[
\vec{n} = (n_x, n_y, n_z) = (\cos \psi \cos \phi, \cos \psi \sin \phi, \sin \psi) \\
= \left[ 1 - 2(\sin^2 \phi/2 + \sin^2 \psi/2) + 4 \sin^2 \phi/2 \sin^2 \psi/2, \sin \phi(1 - 2 \sin^2 \psi/2), \sin \psi \right] \tag{3.2.5}
\]

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

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].
\[ \phi = \arctan \left( \frac{r_y}{r_x} \right) \text{ and } \psi = \arctan \left( \frac{r_z}{\sqrt{r_x^2 + r_y^2}} \right) \]  

(3.2.6)

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

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

(3.2.7)

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

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

with

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

and

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

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 \( (\varepsilon_x, \xi_x, n_y \beta_y \text{ and } n_z \beta_z \sim 1/\gamma^2) \), plus \( \varepsilon_x \beta_x \sim 1/\gamma^4 \).

The precision of these expressions is directly related to the order at which the series

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

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

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

The Fourier transforms

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

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

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

(3.10)

Fig. 6-right gives a typical example in the case of a short magnet. These Fourier transforms are computed without resorting to FFT techniques, namely from

\[ FT_\omega [\vec{E}(\tau)] \approx \sum \vec{E}(\tau_k) e^{-i\omega \tau_k} \Delta \tau_k \]  

(3.11)

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 \( \Delta \tau_k = \Delta s/\beta c \) 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., $1 - \tilde{n}(t) \cdot \tilde{\beta}(t) \rightarrow 1/2 \gamma^2$, 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 $2 \tau_c = \frac{2 \rho}{3 \gamma^3 c} (1 + \gamma^2 \psi^2)^{3/2}$ 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 \texttt{zgoubi} input data file, \texttt{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 \textit{MATRIX}, etc.
- Monte Carlo distribution, with various distributions as well: 6-D window, ellipsoids, etc.

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

\[ K_Y, K_T, K_Z, K_P, K_X, K_D \]

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

\( K_Y, K_T, K_Z, K_P, K_X \) can take the following values:

1. uniform density, \( p(x) = \frac{1}{2\delta x} \) 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.

\( K_D \) can take the following values:

1. uniform density, \( p(D) = \frac{1}{2\delta D} \) 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 \( T = \text{horizontal angle}, D = \delta D \ast T \).

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 \( x - x_0 \) \( (\equiv Y - Y_0, T - T_0, ...) \) 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 \( K_Y, K_T, ... = 1 \) or 3), and the r.m.s. width for Gaussian distributions \( (K_Y, K_T, ... = 2) \)

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

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

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

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

\( IR1, \ IR2, \ IR3 \quad (all \ \approx 10^6) \)

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

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

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

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

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

**\textbf{KOBJ = 3}**: Distribution of \textit{IMAX} particles inside a 6-D ellipsoid defined by the three sets of data (one set per 2-D phase-space)

\[ \alpha_Y, \ \beta_Y, \ \frac{\varepsilon_Y}{\pi}, \ \ N_{\varepsilon_Y} \quad [\quad N_{\varepsilon_Y}', \ \text{if} \ N_{\varepsilon_Y} < 0] \]

\[ \alpha_Z, \ \beta_Z, \ \frac{\varepsilon_Z}{\pi}, \ \ N_{\varepsilon_Z} \quad [\quad N_{\varepsilon_Z}', \ \text{if} \ N_{\varepsilon_Z} < 0] \]

\[ \alpha_X, \ \beta_X, \ \frac{\varepsilon_X}{\pi}, \ \ N_{\varepsilon_X} \quad [\quad N_{\varepsilon_X}', \ \text{if} \ N_{\varepsilon_X} < 0] \]

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

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

\( N_{\varepsilon_Y}, \ N_{\varepsilon_Z} \) and \( N_{\varepsilon_X} \) are the sorting cut-offs (used only for Gaussian distributions, \( KY, KT, ... = 2 \)).

The sorting is uniform in surface (for \( KY = 1 \) or \( KZ = 1 \) or \( KX = 1 \)) or Gaussian (\( KY = 2 \) or \( KZ = 2 \)), 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 \( \sigma_Y = \sqrt{\beta_Y \varepsilon_Y / \pi}, \ \sigma_T = \sqrt{\frac{(1 + \alpha_Y^2)}{\beta_Y^2}} \varepsilon_Y / \pi, \) and similar relations for \( \sigma_Z, \ \sigma_P, \sigma_X, \sigma_D \).

If \( N_{\varepsilon}' \) is negative, thus the sorting fills the elliptical ring that extends from \( |N_{\varepsilon}| \) to \( N_{\varepsilon}' \) (rather than the inner region determined by the \( N_{\varepsilon} \) cut-off as discussed above).
Figure 7: A scheme of input parameters to \textit{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 \(IY \leq n_Y \ldots ID \leq n_D\) such that \(n_Y \times n_T \times \ldots n_D \leq \max(\text{IMAX})\). One defines the sampling range in each coordinate

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

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

\[
0, \quad \pm PY, \quad \pm 2 \times PY, \quad \ldots, \quad \pm IY/2 \times PY, \\
0, \quad \pm PT, \quad \pm 2 \times PT, \quad \ldots, \quad \pm IT/2 \times PT, \\
0, \quad \pm PZ, \quad \pm 2 \times PZ, \quad \ldots, \quad \pm IZ/2 \times PZ, \\
0, \quad \pm PP, \quad \pm 2 \times PP, \quad \ldots, \quad \pm IP/2 \times PP, \\
0, \quad \pm PX, \quad \pm 2 \times PX, \quad \ldots, \quad \pm IX/2 \times PX, \\
0, \quad \pm PD, \quad \pm 2 \times PD, \quad \ldots, \quad \pm ID/2 \times PD,
\]

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 \(DR \times BORO\), resulting in the rigidity of a particle of initial condition \(I \times PD\) to be \((DR + I \times PD) \times BORO\).

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

\[
0, \quad PZ, \quad 2 \times PZ, \quad \ldots, \quad (IZ - 1) \times PZ, \\
0, \quad PP, \quad 2 \times PP, \quad \ldots, \quad (IP - 1) \times PP,
\]

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

\[ Y, \quad T, \quad Z, \quad P, \quad X, \quad D, \quad 'A' \]

where \(D \times BORO\) is the rigidity (negative value allowed) and ‘A’ is a (arbitrary) tagging character.
The last record $IEX$ (I=1, IMAX) contains IMAX times either the character “1” to indicates that the particle has to be tracked, or “-9” to indicates that the particle should not be tracked.

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

*KOBJ = 2.01*: Same as $KOBJ=2$ except for the units, meter and radian in that case.

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

The next three data lines are:

<table>
<thead>
<tr>
<th>$IT1$, $IT2$, $ITStep$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$IP1$, $IP2$, $IPStep$</td>
</tr>
<tr>
<td>$YF$, $TF$, $ZF$, $PF$, $SF$, $DPF$, $TiF$, $TAG$</td>
</tr>
<tr>
<td>$YR$, $TR$, $ZR$, $PR$, $SR$, $DPR$, $TiR$</td>
</tr>
</tbody>
</table>

$InitC$ followed by the storage file name $FNAME$.

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

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

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

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

The file $FNAME$ must be formatted in the appropriate manner. The following FORTRAN sequence is an instance, details and possible updates are to be found in the source file ‘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,1),J=2,MXJ),
   > -1.D0+F(1,I),F(2,I),F(3,I),
   > (F(J,1),J=4,MXJ),ENEKI,
   > TD,I,IREP(I), SORT(I),D,D,D,D,RET(I), DPR(I),
   > D, D, BORO, IPASS, KLEY,LBL1,LBL2,NOEL
100 FORMAT(1X,
   LET(IT),KEX, 1.D0-FO(1,IT),(FO(J,IT),J=2,MXJ),
   A1,1X,I2,1P,7E16.8,
   > 1,1X, 1P,7E16.8,
   C2 1.D0-F(1,IT),(FO(J,IT),J=2,MXJ),
   > /,3E24.16,
   C3 Z,P+1.D3,SAR, 
   > TAR, DS,
   > /,4E24.16,E16.8,
   C4 KART, IT,IREP(IT), SORT(IT),X, BX,BY,BZ, RET(IT), DPR(IT),
   > /,11,216,7E16.8,
   C5 EX,EX,EX, BORO, IPASS, KLEY, (LABEL(NOEL,I),I=1,2),NOEL
   > /, 4E16.8,16,1X, A8,1X, ZA10, 15
ENDDO
```

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

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

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

```
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

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

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

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

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

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

Note: For details and possible updates in the formatted read of concern in the FORTRAN, regarding options 3.01-3.03, 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 \( DR \times BORO \) 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)
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 \( \varepsilon/\) is the ellipse surface, corresponding to an ellipse with equation

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

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

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

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

\[ Y_0 - \delta Y \leq Y \leq Y_0 + \delta Y \quad \ldots \quad D_0 - \delta D \leq D \leq D_0 + \delta D \]

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

\[ -XL \leq X_0 \leq XL \]

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

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

\[
\text{READ (unit=ln, *) } (X7(I), I=1, NCOL)
\]

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

\[
\text{READ (unit=ln, fmt='(1x, ncol*E11.2)') } (X7(I), I=1, NCOL)
\]

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, *if* 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.
4.3 Declaring Options

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

$IR = \quad$ number of the varied element in the structure

$IP = \quad$ number of the physical parameter to be varied in this element

$XC = \quad$ coupling parameter. Normally $XC = 0$. If $XC \neq 0$, coupling will occur (see below)

followed by, either

$DV = \quad$ allowed relative range of variation of the physical parameter $IP$

or

$[V_{min}, V_{max}] = \quad$ 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>first line 1, 2, 3, ..., 9</td>
</tr>
<tr>
<td></td>
<td>second line 10, 11, 12, 13, ..., 19</td>
</tr>
<tr>
<td></td>
<td>this is a comment a line of comments is skipped</td>
</tr>
<tr>
<td></td>
<td>next line 20, 21, 22, ..., 29</td>
</tr>
<tr>
<td>and so on...</td>
<td></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>IL 1</td>
</tr>
<tr>
<td></td>
<td>XL, R₀, B 10, 11, 12</td>
</tr>
<tr>
<td></td>
<td>Xₑ, λₑ 20, 21</td>
</tr>
<tr>
<td></td>
<td>NCE, C₀, C₁, C₂, C₃, C₄, C₅ 30, 31, 32, 33, 34, 35, 36</td>
</tr>
<tr>
<td></td>
<td>Xₛ, λₛ 40, 41</td>
</tr>
<tr>
<td></td>
<td>NCS, C₀, C₁, C₂, C₃, C₄, C₅ 50, 51, 52, 53, 54, 55, 56</td>
</tr>
<tr>
<td></td>
<td>XPAS 60</td>
</tr>
<tr>
<td></td>
<td>KPOS, XCE, YCE, ALE 70, 71, 72, 73</td>
</tr>
</tbody>
</table>

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

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>NFACE, IC, IL 1, 2, 3</td>
</tr>
<tr>
<td></td>
<td>IAMAX, IRMAX 4, 5</td>
</tr>
<tr>
<td></td>
<td>B₀, N, B, G 6, 7, 8, 9</td>
</tr>
<tr>
<td></td>
<td>AT, ACENT, RM, RMIN, RMAX 10, 11, 12, 13, 14</td>
</tr>
<tr>
<td></td>
<td>λ, ξ 15, 16</td>
</tr>
<tr>
<td></td>
<td>NC, C₀, C₁, C₂, C₃, C₄, C₅, shift 17, 18, 19, 20, 21, 22, 23, 24</td>
</tr>
<tr>
<td></td>
<td>ω, θ, R₁, U₁, U₂, R₂ 25, 26, 27, 28, 29, 30</td>
</tr>
<tr>
<td></td>
<td>etc.</td>
</tr>
</tbody>
</table>
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><code>SCALING</code></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>0.605 0.77 0.879</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 2000 10000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AGSQUAD QV_*</td>
<td>24 25 26</td>
<td>No numbering if 1.10 type of option</td>
</tr>
<tr>
<td>3</td>
<td>27 28 29</td>
<td></td>
</tr>
<tr>
<td>0.587 0.83 0.83</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 2000 10000</td>
<td></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></td>
<td></td>
</tr>
<tr>
<td>MULTIPOL COH2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>./Csnk3D/bump_centered.scal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MULTIPOL KICKH KICV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1 0.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MULTIPOL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.72135154291</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>35</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) \([v_{min}, v_{max}]\) : This specifies the allowed interval of variation.

CONSTRAINTS

The next input data in \(FIT[2]\) is the number of constraints, \(NC\). A list of the available constraints is given in the table page 61; adding or changing a constraint resorts to the \(FORTRAN\) file \(ff.f\).

Each constraint is defined by the following list of data :

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

\(IC=0\) : The coefficients \(\sigma_{11} (\sigma_{33}) = \text{horizontal (vertical) beta values and } \sigma_{22} (\sigma_{44}) = \text{horizontal (vertical) derivatives (} \alpha = -\beta/2\) are obtained by transport of their initial values at line start as introduced using for instance \(OBJET, KOBJ=5.1\).

\(IC=0.1\) : Beam parameters : \(\sigma_{11} = \beta_Y, \sigma_{12} = \sigma_{21} = -\alpha_Y, \sigma_{22} = \gamma_Y, \sigma_{33} = \beta_Z, \sigma_{34} = \sigma_{43} = -\alpha_Z, \sigma_{44} = \gamma_Z\); periodic dispersion : \(\sigma_{16} = D_Y, \sigma_{26} = D_Y', \sigma_{36} = D_Z, \sigma_{46} = D_Z'\), all quantities derived by assuming periodic structure and identifying the first order transfer matrix with the form \(Icos\mu + Jsin\mu\).

\(IC=1, 2\) : The coefficients \(R_{ij}\) and \(T_{ijk}\) are calculated following the procedures described in \(MATRIX\), option \(IFOC=0\). The fitting of the \([R_{ij}]\) matrix coefficients supposes the tracking of particles with paraxial coordinates, normally defined using \(OBJET\) option \(KOBJ=5\) or 6.
<table>
<thead>
<tr>
<th>Type of constraint</th>
<th>Parameters defining the constraints</th>
<th>Object definition (recommended)</th>
</tr>
</thead>
<tbody>
<tr>
<td>σ-matrix</td>
<td>$\sigma_{ij}$ (σ₁₁ = β₁, σ₁₂ = σ₂₁ = α₂₁, etc.)</td>
<td>OBJET/KOBJ=5.6</td>
</tr>
<tr>
<td>Periodic parameters</td>
<td>$\sigma_{ij}$ (σ₁₁ = cos(μ₁), α₁ = sin(μ₁), etc.)</td>
<td>OBJET/KOBJ=5.0N</td>
</tr>
<tr>
<td>N=1-9 for MATRIX block 1-9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>First order transport coeffs.</td>
<td>$i \neq 8$ : YY-determinant ; $j=8$ : YZ-det.</td>
<td>OBJET/KOBJ=5</td>
</tr>
<tr>
<td>Transport coeff.</td>
<td>$j \neq 7$ : ZZ-determinant ; $j=7$ : ZY-det.</td>
<td>OBJET/KOBJ=6</td>
</tr>
<tr>
<td>Second order transport coeffs.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trajectory coordinates</td>
<td>$F(J, I)$</td>
<td>[MC]OBJET</td>
</tr>
<tr>
<td>Ellipse parameters</td>
<td>$\sigma_{ij}$ (σ₁₁ = β₁, σ₁₂ = σ₂₁ = α₂₁, 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>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 (NP=1) or maximum value (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 MCOBJET, KOBJ=3).

The fitting of the $[\sigma_{ij}]$ coefficients supposes the tracking of a relevant population of particles within an appropriate emittance.

IC=5 : The constraint value is the ratio of particles (over IMAX). Three cases possible :

$I = -1$, ratio of particles still on the run.

$I = 1, 2, 3$, maximization of the number of particles encompassed within a given $I$-type (for respectively $Y, Z, D$) phase-space emittance value. Then, NP=1, followed by the emittance value. The center and shape of the ellipse are determined by a matching to the position and shape of the particle distribution.

$I = 4, 5, 6$, same as previous case, except for the ellipse, taken to be the rms matched ellipse to the distribution. Thus NP=0.
IC=10 : If \( 1 \leq I \leq \text{IMAX} \) 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 \textit{OBJET} with either \( KOBJ = 5 \) or \( KOBJ = 6 \), or from particle distributions and in this case need object definition using for instance \textit{OBJET} with \( KOBJ = 8 \), \textit{MCOBJET} with \( KOBJ = 3 \).

**THE FITTING METHODS**

The \textit{FIT} procedure was drawn from the matrix transport code BETA [28]. It is a direct sequential minimization of the quadratic sum of all errors (\textit{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 \textit{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.
GASCAT: Gas scattering

Modification of particle momentum and velocity vector, performed at each integration step, under the effect of scattering by residual gas.

*Installation is to be completed.*
GETFITVAL : Get values of variables as saved from former FIT[2] run

This keyword allows reading, from a file whose name needs be specified, parameter values to be assigned to optical elements in zgoubi.dat.

That file is expected to contain a copy-paste of the data under the FIT[2] procedure as displayed in zgoubi.res, normally under the form

```
# STATUS OF VARIABLES (Iteration # 95)
  LMNT VAR PARAM MINIMUM INITIAL FINAL MAXIMUM STEP NAME LBL1 LBL2
145 1 4 -3.000E+03  762. 761.9484791 3.000E+03 1.254E-05 MULTIPOL HKIC DHCB02
182 2 4 -1.000E+03  231. -230.9846875 1.000E+03 4.182E-06 MULTIPOL HKIC DHCB08
146 3 4 -1.000E+03  320. -319.8554171 1.000E+03 4.182E-06 MULTIPOL VKIC DVCB02
183 4 4 -1.000E+03  528. 527.7249064 1.000E+03 4.182E-06 MULTIPOL VKIC DVCB08
615 5 4 -3.000E+03  308. 307.6860565 3.000E+03 1.254E-05 MULTIPOL HKIC DHCFO2
651 6 4 -1.000E+03  114. 113.8490362 1.000E+03 4.182E-06 MULTIPOL HKIC DHCF08
616 7 4 -1.000E+03  -78.9 -78.88730937 1.000E+03 4.182E-06 MULTIPOL VKIC DVCF02
652 8 4 -1.000E+03  212. 211.8789183 1.000E+03 4.182E-06 MULTIPOL VKIC DVCF08
```

```
# STATUS OF CONSTRAINTS
# TYPE I J LMNT# DESIRED WEIGHT REACHED KI2 * Parameter(s)
# 3 1 2 127 0.00000000E+00 1.0000E+00 1.00680881E-08 6.0335E-01 *
# 3 1 3 127 0.00000000E+00 1.0000E+00 7.01014051E-09 2.9250E-01 *
# 3 1 4 127 0.00000000E+00 1.0000E+00 2.91843831E-10 5.0696E-04 *
# 3 1 5 127 0.00000000E+00 1.0000E+00 3.11423811E-10 5.7727E-04 *
# 3 1 2 436 0.00000000E+00 1.0000E+00 3.84383781E-09 8.7944E-02 *
# 3 1 3 436 0.00000000E+00 1.0000E+00 1.57730111E-09 1.4808E-02 *
# 3 1 4 436 0.00000000E+00 1.0000E+00 2.20812721E-10 2.9022E-04 *
# 3 1 5 436 0.00000000E+00 1.0000E+00 5.79305521E-11 1.9975E-05 *
```

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 not need any commenting.

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

$$1 \rightarrow 2 + 3$$

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 $D_1 = \frac{p_1}{q}$ BORO), 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

$$\gamma_2^* = \frac{M_2^2 + M_2^2 - M_3^2}{2M_1 M_2}$$

$$\beta_2^* = \left(1 - \frac{1}{\gamma_2^*} \right)^{1/2}$$

$$\gamma_2 = \gamma_1 \gamma_2^* \left(1 + \beta_1 \beta_2^* \cos \theta^* \right)$$

$$p_2 = M_2 \sqrt{\gamma_2^2 - 1}$$

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 $D_2 = \frac{p_2}{q}$ 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.
In ESL and CHANGREF, $F(6, I)$ is compared to $s$ at the end of the element. If the decay occurs inside the element, the particle is considered as having decayed at its actual limit path length $s$, thus its coordinates at $s$ are recalculated by translation.

The limit path length of all particles ($I = 1, IMAX$) is stored in the array $FDES(6, I)$. For further statistical purposes (e.g., use of HISTO) the daughter particle 2 is tagged with an 'S' standing for “secondary”. When a particle decays, its coordinates $D, Y, T, Z, P, s, time$ at the decay point are stored in $FDES(J, I)$, $J = 1, 7$.

A note on negative drifts:
The use of negative drifts with MCDESINT is allowed and correct. For instance, negative drifts may occur in a structure for some of the particles when using CHANGREF (due to the $Z$-axis rotation or a negative XCE), or when using DRIFT with $XL < 0$. Provision has been made to take it into account during the MCDESINT procedure, as follows.

If, due to a negative drift, a secondary particle reaches back the decay location of its parent particle, then the parent particle is “resurrected “ with its original coordinates at that location, the secondary particle is discarded, and ray-tracing resumes in a regular way for the parent particle which is again allowed to decay, after the same path length. This procedure is made possible by prior storage of the coordinates of the parent particles (in array $FDES(J, I)$) each time a decay occurs.

Negative steps ($XPAS < 0$) in optical elements are not compatible with MCDESINT.
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 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

**DECAPOLE, DODECAPO, EBMULT, ELMULT, MULTIPOL, OCTUPOLE, QUADRUPO, SEXTUPOLE**

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 ORDRE, so to avoid pushing Taylor expansions as high as $\vec{u}^{(5)}$.

**NOTE**: see also the option IORDRE in field map declarations (**DIPOLE-M, TOSCA**, etc.).
PARTICUL : Particle characteristics

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:

- CAVITE : mass, charge
- MCDESINT : mass, COM life-time
- SPNTRK : mass, gyromagnetic factor
- SRLOSS : mass, charge
- SYNRAD : mass, charge
- Electric and Electro-Magnetic elements : mass, charge

The declaration of PARTICUL must precede these keywords.

If PARTICUL is omitted, which is in general the case when ray-tracing ions in purely magnetic optical assemblies, then zgoubi, since it only knows the rigidity, will skip the computation of such quantities as time of flight.
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 $\text{Max(IMAX)}$ particles are to be tracked. Thus, when the following random procedures are used : *MCOBJET, OBJETA, MCDESINT, SPNTRK* ($KSO = 5$), their random seeds are not reset and independent statistics will add up. This includes Monte Carlo simulations, in beam lines : normally $K = 0$. $NPASS$ runs through the same structure, from *MCOBJET* to *REBELOTE* will follow, resulting in the calculation of $(1+NPASS) \times \text{IMAX}$ 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 $\text{IMAX}$ 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 $KW\text{RIT} = i,j$, 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 MC_OBJET, 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.

<table>
<thead>
<tr>
<th>SCALING</th>
<th>- Scaling</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 4</td>
<td>Active. NF = 4 families of elements are concerned, as listed below</td>
</tr>
<tr>
<td>QUADRUPO QFA QFB</td>
<td>- Quadrupoles labeled 'QFA' and Quadrupoles labeled 'QFB'</td>
</tr>
<tr>
<td>2</td>
<td>NT = 2 timings</td>
</tr>
<tr>
<td>18131.E-3</td>
<td>from turn 1 to turn 6379</td>
</tr>
<tr>
<td>24176.E-3</td>
<td>The field increases (linearly) from 18131E-3<em>B0 to 24176E-3</em>B0</td>
</tr>
<tr>
<td>1</td>
<td>from turn 1 to turn 6379</td>
</tr>
<tr>
<td>MULTIPOL QDA QDB</td>
<td>- Multipoles labeled 'QDA' and Multipole labeled 'QDB'</td>
</tr>
<tr>
<td>2</td>
<td>NT = 2 timings</td>
</tr>
<tr>
<td>18131.E-3</td>
<td>Fields increase from 18131E-3<em>Bi to 24176E-3</em>Bi (∀i = 1, 10 poles)</td>
</tr>
<tr>
<td>24176.E-3</td>
<td>from turn 1 to turn 6379</td>
</tr>
<tr>
<td>1</td>
<td>from turn 1 to turn 6379</td>
</tr>
<tr>
<td>BEND</td>
<td>- All BEND's (regardless of any LABEL)</td>
</tr>
<tr>
<td>2</td>
<td>NT = 2 timings</td>
</tr>
<tr>
<td>18131.E-3</td>
<td>As above</td>
</tr>
<tr>
<td>24176.E-3</td>
<td>From turn 1 to turn 6379</td>
</tr>
<tr>
<td>1</td>
<td>As above</td>
</tr>
<tr>
<td>CAVITE</td>
<td>- Accelerating cavity</td>
</tr>
<tr>
<td>3</td>
<td>NT = 3 timings</td>
</tr>
<tr>
<td>1 1.22</td>
<td>The synchronous rigidity (Bρ), increases,</td>
</tr>
<tr>
<td>133352</td>
<td>from (Bρ)s to 1.22*(Bρ)s, from turn 1 to 1200, and</td>
</tr>
<tr>
<td>1 1200</td>
<td>from 1.22*(Bρ)s to 1.33352*(Bρ)s from turn 1200 to 6379</td>
</tr>
<tr>
<td>6379</td>
<td></td>
</tr>
</tbody>
</table>

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

\[
f_{RF} = \frac{hc}{L} \frac{q(Bp)_s}{(q^2(Bp)_s^2 + (Mc^2)^2)^{1/2}}
\]

where the rigidity is updated in the following way. Let (Bp)ss be the initial rigidity (namely, (Bp)ss = 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, $SCALE(N) = 1.22 + \frac{1.3352 - 1.22}{1 + 6379 - 1200} (N - 1200)$

and then,

$$(B\rho)_s(N) = SCALE(N) \cdot (B\rho)_{s0}$$

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 File name
1 2 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 $\delta A = 0$. 
**SRLOSS : Synchrotron radiation loss [17]**

The keyword *SRLOSS* allows activating or stopping (option \( KSR = 1, 0 \) respectively) stepwise tracking of energy loss by stochastic emission of photons in magnetic fields, following the method described in section 3.1.

It can be chosen to allow radiation in the sole dipole fields, or in all types of fields regardless of their multipole composition. It can also be chosen to allow for the radiation induced transverse kick.

*SRLOSS* must be preceded by *PARTICUL* for defining mass and charge values as they enter in the definition of SR parameters.

Statistics on SR parameters are computed and updated while tracking, the results of which can be obtained by means of the keyword *SRPRNT*. 
SYNRAD : Synchrotron radiation spectral-angular densities

The keyword SYNRAD enables (or disables) the calculation of synchrotron radiation (SR) electric field and spectral angular energy density. It must be preceded by PARTICUL for defining mass and charge values, as they enter in the definition of SR parameters.

SYNRAD is supposed to appear a first time at the location where SR should start being taken into account, with the first data KSR set to 1. It results in on-line storage of the electric field vector and other relevant quantities in zgoubi.sre, as step by step integration proceeds. The observer position (XO, YO, ZO) is specified next to KSR.

Data stored in zgoubi.sre :

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

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 \( \int_{\nu_1}^{\nu_2} \frac{\partial^3 W}{\partial \phi \partial \psi \partial \nu} \) (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 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].
4.4 Optical Elements and Related Numerical Procedures

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

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

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

The extent of the map is defined by the following parameters, as shown in Figs. 11A and 11B,

- $A T$: total angular aperture
- $R M$: mean radius used for the positioning of field boundaries
- $R M I N, R M A X$: 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,

- $A C E N T$: arbitrary angle, used for the positioning of the EFB’s.
- $\omega$: azimuth of an EFB with respect to $A C E N T$
- $\theta$: 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

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

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, $\lambda$, is associated (Figs. 11A and 11B), and a fringe field coefficient $F$ is calculated. In the following $\lambda$ stands for either $\lambda_E$ (Entrance), $\lambda_S$ (Exit) or $\lambda_L$ (Lateral EFB).

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

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

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

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

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

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

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

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

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]

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

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

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

The values of the coefficients $C_0$ to $C_5$ should be such that the derivatives of $B_Z$ with respect to $s$ be negligible at $s = \pm \lambda$, so as not to perturb the extrapolation of $\vec{B}$ out of the median plane.

It is also possible to simulate a shift of the EFB, by giving a non zero value to the parameter $\text{shift}$. $s$ 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)

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

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

The Mesh of the Field Map

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

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

and the angular step by

$$\delta \theta = \frac{AT}{IA_{\text{MAX}} - 1}$$

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

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

Figure 13: Effective value of $\mathcal{F}(R, \theta)$ for overlapping fringe fields $F_1$ and $F_2$ centered at $O_1$ and $O_2$. 
If $NBS \geq 1$, then $NBS$ shims are introduced at positions $\frac{R_1 + R_2}{2}$, $\frac{\theta_1 + \theta_2}{2}$ (Fig. 14) [35]

The initial field map is modified by shims with second order profiles given by

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

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 $I_1$, is set to zero.

If $I = 3$, it is equivalent to

$$\text{CHANGREF}[XW, YW, T(I_1)]$$

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 $I_1$, $I_2$ and $I_3$ specified as data, while $T(I_1)$ 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 *DIPOLE[S]* that uses a polar frame). As a consequence, having particle coordinates at entrance or exit of the magnet referring to the curved main direction of motion may require using *KPOS*, in particular *KPOS*=3 (in a circular 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 $\theta$ satisfies $XL = \frac{2 BORO}{B1} \sin \theta/2$.

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 $\pm X_E(\pm X_S)$ on both sides of the entrance (exit) EFB, the fringe field model is used (same as for *QUADRUPO*, Fig. 35, p. 133) ; elsewhere, the field is supposed to be uniform.

If $\lambda_E$ (resp. $\lambda_S$) is zero sharp edge field model is assumed at entrance (resp. exit) of the magnet and $X_E$ (resp. $X_S$) is forced to zero. In this case, the wedge angle vertical first order focusing effect (if $B1$ is non zero) is simulated at magnet entrance and exit by a kick $P_2 = P_1 - Z_1 \tan(\epsilon/\rho)$ applied to each particle ($P_1$, $P_2$ are the vertical angles upstream and downstream the EFB, $Z_1$ the vertical particle position at the EFB, $\rho$ the local horizontal bending radius and $\epsilon$ the wedge angle experienced by the particle ; $\epsilon$ depends on the horizontal angle $T$).

Magnet (mis-)alignment is assured by *KPOS*. *KPOS* also allows some degrees of automatic alignment useful for periodic structures (section 4.6.7).

---

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

CAVITE provides a simulation of a (zero length) accelerating cavity; it can be used in conjunction with keywords REBELOTE and SCALING for the simulation of multi-turn tracking with synchrotron or fixed field (FFAG, cyclotron) acceleration (see section 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,

$$B_{\rho_{ref}} = B_{ORO} \rightarrow B_{\rho_{ref}} = B_{ORO} + \delta B_{\rho_s}$$

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

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

where

- $\phi$ = RF phase; $\phi_2 - \phi_1$ = variation of $\phi$ between two traversals
- $W$ = kinetic energy; $W_2 - W_1$ = energy gain at a traversal of CAVITE
- $\mathcal{L}$ = length of the synchronous closed orbit (to be calculated by prior ray-tracing, see the bottom NOTE)
- $\ell$ = orbit length of the particle between two traversals
- $\beta_c$ = velocity of the (virtual) synchronous particle
- $\beta_{s,c}$ = 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

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

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 $(B\rho)_s = B_{ORO} \cdot SCALE(TIMING)$ (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 $B_{ORO}$ 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

  $$
  \phi_1 = \phi_s = \text{synchronous phase}
  $$

  $$(B\rho)_1 = BORO
  $$

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

  $$
  \phi_{1I} = \phi_s = \text{synchronous phase}
  $$

  $$(B\rho)_{1I} = BORO * D_I
  $$

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

**Calculation of the Coordinates**

Let $p_I = [p_{X_I}^2 + p_{Y_I}^2 + p_{Z_I}^2]^{1/2}$ be the momentum of particle $I$ at the exit of the cavity, while $p_{I0} = [p_{X_{I0}}^2 + p_{Y_{I0}}^2 + p_{Z_{I0}}^2]^{1/2}$ is its momentum at the entrance. The kick in momentum is assumed to be fully longitudinal, resulting in the following relations between the coordinates at the entrance (denoted by the index zero) and at the exit

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

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

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

and for the angles (see Fig. 1)

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

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

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

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

where the synchronous phase $\phi_s$ is 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 ±*YL* and ±*ZL* such that any particle will be stopped if its coordinates *Y, Z* satisfy

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

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

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 ± 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

\[
T_2 = T_1 - ALE
\]

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

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

\[
Z_2 = Z_1 + DL \tan P_1
\]

\[
S_2 = S_1 + \frac{DL}{\cos P_1}
\]

\[
P_2 = P_1
\]

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”

```
OBJET
51.7123965921708 Electron, Ekin=15MeV.
2 1 1
1 1 1 1 1 1 1 1 1
MARKER BEG .plt -> list into zgoubi.plt.

 DRIFT 10 cm drift.
10.

 CHANGREF
 0. 0. -6.34165 First : half Z-rotation.

 CHANGREF
 0. 1. 0. Next : Y-shift.

 MULTIPOL
 2 -> list into zgoubi.plt.

 5 10. 2.064995867082342 2. 0. 0. 0. 0. 0. 0.
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0.1455 2.2670 -.6395 1.1558 0. 0. 0.
 0 0 0

 MARKER END .plt*.plt => list into zgoubi.plt.

 END
```

```
Zgoubi|Zpop

Y_Lab (m) vs. X_Lab (m)

Trajectory

Optical Axis

Note : The square markers scheme the stepwise integration in case of ±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.

1 1
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

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.
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 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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laboratory momentum</td>
<td>$p_1 \equiv 0$</td>
</tr>
<tr>
<td>Rest mass</td>
<td>$M_1$</td>
</tr>
<tr>
<td>Total energy in laboratory</td>
<td>$M_1c^2$</td>
</tr>
</tbody>
</table>

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

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

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

$$M_1c^2 + W_2 = W_3 + W_4$$

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

![Figure 18: Scheme of the principles of CIBLE (TARGET)](image)

$A, T =$ position, angle of incoming particle 2 in the entrance reference frame

$P =$ position of the interaction

$B, T =$ position, angle of the secondary particle in the exit reference frame

$\theta =$ angle between entrance and exit frames

$\beta =$ tilt angle of the target
COLLIMA : Collimator

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

Physical Aperture

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

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

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

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 \( Z \)

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

with \( G_0 = \frac{B_0}{R_0^4} \)

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

\[
\begin{align*}
B_X &= 0 \\
B_Y &= 4G_0(Y^2 - Z^2)YZ \\
B_Z &= G_0(Y^4 - 6Y^2Z^2 + Z^4)
\end{align*}
\]

Figure 19: Decapole magnet
4.4 Optical Elements and Related Numerical Procedures

**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

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

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]

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

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

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

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

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

![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.](image)

**Extrapolation Off Median Plane**

From the vertical field \( \vec{B} \) and derivatives in the median plane, first a transformation from polar to Cartesian coordinates is performed, following eqs (1.4.9 or 1.4.10), then, extrapolation off median plane is performed by means of Taylor expansions, following the procedure described in section 1.3.3.
DIPOLE-M : Generation of dipole mid-plane 2-D map, polar frame

DIPOLE-M is a more recent, simpler and improved version of AIMANT.

The keyword DIPOLE-M provides an automatic generation of a dipole field map in polar coordinates. The extent of the map is defined by the following parameters, as shown in Figs. 11A and 11B.

\[ \begin{align*}
A_T & : \text{total angular aperture} \\
RM & : \text{mean radius used for the positioning of field boundaries} \\
RMIN, RMAX & : \text{minimum and maximum radii}
\end{align*} \]

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.

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

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

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

where \(N\), \(B\) and \(G\) 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 \(\text{shift}\). \(s\) 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. At any node of the map mesh, the resulting value of the fringe field coefficient (eq. 4.4.9) is

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

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

and the angular step by

\[ \delta \theta = \frac{AT}{IA_{\text{MAX}} - 1} \]

where \( R_{\text{MIN}} \) and \( R_{\text{MAX}} \) are the lower and upper radial limits of the field map, and \( AT \) is its total angular aperture (Fig. 11B). \( IR_{\text{MAX}} \) and \( IA_{\text{MAX}} \) are the total number of nodes in the radial and angular directions.

Simulating Field Defects and Shims

Once the initial map is calculated, it is possible to modify it by means of the parameter \( NBS \), so as to simulate field defects or shims.

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

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

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

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

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

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

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

Extrapolation Off Median Plane

The vector field \( \vec{B} \) and its derivatives in the median plane are calculated by means of a second or fourth order polynomial interpolation, depending on the value of the parameter \( \text{IORDRE} \) (\( \text{IORDRE}=2, 25 \) 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.
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

- $AT$ : total angular aperture
- $RM$ : mean radius used for the positioning of field boundaries

For each one of the $N = 1$ to 5 dipoles of the $N$-tuple, the 2 effective field boundaries (entrance and exit EFBs) from which the dipole field (eqs. 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)

- $ACNi$ : 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

**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,

(i) \[ B_{Zi}(R, \theta) = B_{Z0,i} \mathcal{F}_i(R, \theta) \left(1 + b_1_i (R - RM_i)/RM_i + b_2_i (R - RM_i)^2/RM_i^2 + \ldots \right) \] (4.4.10)

(ii) \[ B_{Z}(R, \theta) = B_{Z0,i} + \sum_{i=1}^{N} \mathcal{F}_i(R, \theta) \left( b_1_i (R - RM_i) + b_2_i (R - RM_i)^2 + \ldots \right) \] (4.4.11)

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 chicaned dipoles, cyclotron or FFAG magnets, etc.

---

**Figure 21**: Definition of a dipole triplet using the **DIPOLES** or FFAG procedures.

**Calculation of the Fringe Field Coefficient**

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

$^5$**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

$$\mathcal{F}_i(R, \theta) = F_E \ast F_S$$  \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

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

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

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

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

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

which, in turn, intervene in the derivatives of the compound functions

\[ \frac{\partial u}{\partial \theta} \frac{\partial F(R, \theta)}{\partial x}, \frac{\partial u}{\partial r} \frac{\partial P(R, \theta)}{\partial y} \]

**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.

![Interpolation method](image)

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.
4.4 Optical Elements and Related Numerical Procedures

**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 $Z$

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

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

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

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

![Figure 23: Dodecapole magnet](image-url)
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.
EBMULT : Electro-magnetic multipole

**EBMULT** simulates an electro-magnetic multipole, by addition of electric ($\vec{E}$) and magnetic ($\vec{B}$) multipole components (dipole to 20-pole). $\vec{E}$ and its derivatives $\frac{\partial^{i+j+k}\vec{E}}{\partial X^i \partial Y^j \partial Z^k} (i + j + k \leq 4)$ are derived from the general expression of the multipole scalar potential (eq. 1.3.5), followed by a $\frac{\pi}{2n}$ rotation ($n = 1, 2, 3, ...$) (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 $D = 0$

$$V(X) = \frac{V_2 - V_1}{2} \frac{1}{2 \omega D/R_0} \frac{\ln}{\frac{\sinh}{\cosh}} \frac{x + D}{R_0} \frac{\cosh}{\frac{x - D}{R_0}} \left[ + \frac{V_1 + V_2}{2} \right]$$ 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 $\left[ \frac{V_1 + V_2}{2} \right]$ which disappears when differentiating).

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

![Figure 26: Two-electrode cylindrical electric lens.](image)
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 \textsc{Particul} prior to \textsc{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 \textsc{Changref}, or using \textsc{Kpos}=3 with \( YCE \) and \( ALE=\) half-deviation matched to the reference trajectory.](image-url)
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 \( R_1, R_2, \ldots, R_{N-1}, D \) 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 \textit{PARTICUL} prior to \textit{ELMIRC}, for the definition of particle mass and charge.

![Figure 28: Electrostatic N-electrode mirror/lens, circular slits, in the case \( N = 3 \), in horizontal mirror mode.](image)

\(^6\)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]

\[
K_n = \frac{1}{2} \frac{\gamma}{\gamma^2 - 1} \frac{V_n}{R_0^2}
\]

(\( V_n = \) 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 \( E_2 (E_3, \ldots, E_{10}) \) at entrance, and \( S_2 (S_3, \ldots, S_{10}) \) at exit, such that the fringe field extent is \( \lambda_E \ast E_2 (\lambda_E \ast E_3, \ldots, \lambda_E \ast E_{10}) \) at entrance and \( \lambda_S \ast S_2 (\lambda_S \ast S_3, \ldots, \lambda_S \ast S_{10}) \) 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 ($\vec{E}_1 = \vec{E}_3 = \vec{E}_5 = \ldots = \vec{E}_{10} = \vec{0}$) [44].
4.4 Optical Elements and Related Numerical Procedures

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 \( IX \) is the number of nodes along the (symmetry) \( X \)-axis, \( X(I) \) their coordinates, and \( EX(I) \) are the values of the \( X \) component of the field. \( EX \) is normalized with \( ENORM \) prior to ray-tracing. As well the longitudinal coordinate \( X \) is normalized with a \( XNORM \) coefficient (useful to convert to centimeters, the working units in *zgoubi*).

\( X \)-cylindrical symmetry is assumed, resulting in \( EY \) and \( EZ \) taken to be zero on axis. \( \vec{E}(X,Y,Z) \) and its derivatives along a particle trajectory are calculated by means of a 5-points polynomial interpolation followed by second order off-axis extrapolation (see sections 1.3.1 and 1.4.1).

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

Use **PARTICUL** prior to **ELREVOL**, for the definition of particle mass and charge.
EMMA : 2-D Cartesian or cylindrical mesh field map for EMMA FFAG

*EMMA* is dedicated to the reading and treatment of 2-D or 3-D Cartesian mesh field maps representing the EMMA FFAG cell quadrupole doublet [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.
- $MOD=1$: a single map, interpolated from QF[$Y_F$] and QD[$Y_D$] ones, is built.
- $MOD=22$: 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 FNAM E.

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

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

For each one of the \( N = 1 \) to (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)

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

![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

\[
B_{Zi}(R, \theta) = B_{Zi,0} \mathcal{F}_i(R, \theta) (R/R_M)^K_i
\]

wherein \(B_{Zi,0}\) is a reference field, at reference radius \(R_M\), whereas \(\mathcal{F}(R, \theta)\) is calculated as described below.

**Calculation of \(\mathcal{F}_i(R, \theta)\)**

The fringe field coefficient \(\mathcal{F}_i(R, \theta)\) associated with a dipole is computed as in the procedure *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.
\[ g(R) = g_0 (RM/R)^\kappa \]  

so to simulate the effect of gap shaping on \( B_{Zi}(R,\theta) \) 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 \textit{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

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

For each one of the $N = 1$ to (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

\[
\begin{align*}
ACN_i & : \text{ arbitrary inner angle, used for EFB’s positioning} \\
\omega & : \text{ azimuth of an EFB with respect to } ACN \\
\zeta & : \text{ spiral angle}
\end{align*}
\]

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
\[ B_{Z_i}(R, \theta) = B_{Z0,i} \mathcal{F}_i(R, \theta) \left( \frac{R}{R_M} \right)^{K_i} \]

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

**Calculation of \( \mathcal{F}_i(R, \theta) \)**

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

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

(4.4.16)

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

**Calculation of the Full Field From All \( N \) Dipoles**

For the rest, namely calculation of the full field at particle position, as resulting from the \( N \) dipoles, calculation of the mid-plane field derivatives, extrapolation off median plane, etc., things are performed in the same manner as for the \textit{DIPOLES} procedure (see page 106).
4.4 Optical Elements and Related Numerical Procedures

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 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 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

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

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

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

The independent components $\vec{B}_1, \vec{B}_2, \vec{B}_3, ..., \vec{B}_{10}$ 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 $E_2 (E_3, ..., E_{10})$ at entrance, and $S_2 (S_3, ..., S_{10})$ at exit, such that the 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 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).
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 $Z$

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

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

$$B_X = 0$$
$$B_Y = G_0(3Y^2 - Z^2)Z$$
$$B_Z = G_0(Y^2 - 3Z^2)Y$$

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, \( FNAME = \text{outpoi.lis} \) ), the data are read following the FORTRAN statements here under:

\[
\begin{align*}
I & = 0 \\
11 & \text{ CONTINUE} \\
I & = I + 1 \\
& \quad \text{READ(LUN,101,ERR=10,END=10) K, K, R, X(I), R, R, B(I)} \\
101 & \quad \text{FORMAT(I1, I3, I4, E15.6, 2F11.5, 2F12.3)} \\
& \quad \text{GOTO 11} \\
10 & \text{ CONTINUE} \\
& \quad \ldots
\end{align*}
\]

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 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 $X_L$. 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, $\lambda \simeq 2 \times R_0$).

When fringe fields overlap inside the magnet ($XL \leq X_E + X_S$), the gradient $G$ is expressed as

\[
G = G_E + G_S - 1
\]

where, $G_E$ is the entrance gradient and $G_S$ is the exit gradient.

If $\lambda_E = 0$ ($\lambda_S = 0$), the field at entrance (exit) is considered as sharp edged, and then $X_E(X_S)$ is forced to zero (for the mere purpose of saving computing time).

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

\[
\begin{align*}
B_X &= 0 \\
B_Y &= G_0 Z \\
B_Z &= G_0 Y
\end{align*}
\]
4.4 Optical Elements and Related Numerical Procedures

Figure 34: Quadrupole magnet

Figure 35: Scheme of the longitudinal field gradient $G(X)$. $(OX)$ is the longitudinal axis of the reference frame $(0, X, Y, Z)$ of zgoubi. The length of the element is XL. Trajectories are ray-traced from $-X_E$ to $XL + X_S$, by means of respectively prior and final automatic change of frame.
SEPARA : Wien Filter - analytical simulation

Note : simulation by stepwise integration can be found in WIENFILTER.

SEPARA provides an analytic simulation of an electrostatic separator. Input data are the length \( L \) of the element, the electric field \( E \) and the magnetic field \( B \). The mass \( m \) and charge \( q \) of the particles are entered by means of the keyword PARTICUL.

The subroutines involved in SEPARA solve the following system of three equations with three unknown variables \( S \), \( Y \), \( Z \) (while \( X \equiv L \)), that describe the cycloidal motion of a particle in \( \vec{E}, \vec{B} \) static fields (Fig. 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 \( C_2 = \beta c \cos(T_0) \cos(P_0) \) 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.
The meaning of parameters for SEXTUPO is the same as for QUADRUPO.

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

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

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 ($\lambda_E = \lambda_S = 0$), $\vec{B}(X,Y,Z)$ in the magnet is given by

$$B_X = 0$$
$$B_Y = 2G_0YZ$$
$$B_Z = G_0(Y^2 - Z^2)$$
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)

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

and yields the magnetic length

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

with in addition

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

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$^7$. 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)
    100 FORMAT(1X,6E11.2)
    ENDIF
  1 CONTINUE
```

$IX$ ($JY$, $KZ$) is the number of longitudinal (transverse horizontal, vertical) nodes of the 3-D uniform mesh. For letting zgoubi know 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).

$^7$Use MAP2D in case non-zero $B_X$, $B_Y$ are to be taken into account in a 2-D map.
## 4.4 Optical Elements and Related Numerical Procedures

### MOD MOD2

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

<table>
<thead>
<tr>
<th>MOD ≥ 20</th>
<th>Cylindrical mesh</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>
</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>
</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.
TRANSMAT : Matrix transfer

TRANSMAT 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 TRANSMAT.
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 [51]

$$V(x) = \frac{V_2 - V_1}{2\omega D} \left[ \ln \frac{\cosh \left( \frac{x + X_2}{2} + \frac{D}{R_0} \right)}{\cosh \left( \frac{x + X_2}{2} \right) R_0} + \ln \frac{\cosh \left( \frac{x - X_2}{2} - \frac{D}{R_0} \right)}{\cosh \left( \frac{x - X_2}{2} \right) R_0} \right]$$

($x =$ distance from the center of the central electrode ; $\omega = 1.318$ ; $\cosh =$ hyperbolic cosine), from which the field $\vec{E}(X, Y, Z)$ and its derivatives are deduced following the procedure described in section 1.3.1.

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

The total length of the lens is $X_1 + X_2 + X_3 + 2D$; 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.
4.4 Optical Elements and Related Numerical Procedures

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 \( XL \) 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

\[
B_0 = -\frac{E_0}{\beta_{WC}}
\]

for the selection of “wanted” particles of velocity \( \beta_{WC} \). 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., \( XL \leq X_E + X_S \)), the field fall-off is expressed as

\[
F = F_E + F_S - 1
\]

where \( F_E(F_S) \) is the value of the coefficient respective to the entrance (exit) EFB.

If \( \lambda_E = 0 \) (\( \lambda_S = 0 \)) for either the electric or magnetic component, then both are considered as sharp edge fields and \( X_E(X_S) \) is forced to zero (for the purpose of saving computing time). In this case, the magnetic wedge angle vertical first order focusing effect is simulated at entrance and exit by a kick \( P_2 = P_1 - Z_1 \tan(\epsilon/\rho) \) applied to each particle (\( P_1, P_2 \) are the vertical angles upstream and downstream the EFB, \( Z_1 \) the vertical particle position at the EFB, \( \rho \) the local horizontal bending radius and \( \epsilon \) the wedge angle experienced by the particle; \( \epsilon \) depends on the horizontal angle \( T \)). This is not done for the electric field, however it is advised not to use a sharp edge electric dipole model since this entails non symplectic mapping, and in particular precludes accounting for momentum effects of the non zero longitudinal electric field component.
YMY: Reverse signs of $Y$ and $Z$ reference axes

YMY performs a 180° 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.D0+FO(1,I),(FO(J,I),J=2,MXJ),
        2 -1.D0+F(1,I),F(2,I),F(3,I),(F(J,I),J=4,MXJ),
        3 (SI(J,I),J=1,4),(SF(J,I),J=1,4),
        4 ENEKI,ENERG,
        5 I,IREP(I), SORT(I),(AMQ(J,I),J=1,5),RET(I),DPR(I),PS,
        6 BORO, IPASS, NOEL, KLEY,LBL1,LBL2,LET(I)
     2 CONTINUE
ELSE
  DO 1 I=1,IMAX
     P = BORO * CL9 * F(1,I) * AMQ(2,I)
     ENERG = SQRT(P*P + AMQ(1,I)*AMQ(1,I))
     ENEKI = ENERG - AMQ(1,I)
     WRITE(NFAI,110)
        1 IEX(I),-1.D0+FO(1,I),(FO(J,IT),J=2,MXJ)
        2 -1.D0+F(1,I),F(2,I),F(3,I),(F(J,I),J=4,MXJ),
        3 (SI(J,I),J=1,4),(SF(J,I),J=1,4),
        4 ENEKI,ENERG,
        5 I,IREP(IT), SORT(IT),(AMQ(J,I),J=1,5),RET(IT),DPR(IT),PS,
        6 BORO, IPASS, NOEL,
        7 TX1,KLEY, TX1, TX1, LBL1, TX1, TX1, LBL2, TX1, TX1, LET(IT), TX1
     1 CONTINUE
ENDIF
110 FORMAT(1X,
      C1 KEX, XXXO, (FO(J,IT),J=2,MXJ)
      > 1P, 1X,I2, 7(1X,E16.8)
      C2 XXX,Y,T*1.D3,
      > ,3(1X,E24.16)
      C3 2,P*1.D3,SAR,TAR
      > ,4(1X,E24.16)
      C4 SXo, SYo, SZo, So, SX, SY, SZ, S
      > ,8(I8,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(I8,6), 9(1X,E16.8)
      C7 BORO, IPASS, NOEL,
      > ,1X,E16.8, 2(I8,6)
      C8 'KLEY', ('LABEL(NOEL,I)',I=1,2), 'LET(IT)'
      > ,1X,A1,A10,A1, 2(I8,A1,A10,A1),
      > 1X,3A1)
```

The meaning of the main data is the following (see the keyword **OBJET**)

- **KEX** : keyword, initial D-1,Y,T,2,P,S,Time
- **XXX** : tag, initial D-1,Y,T,2,MXJ
- **Y** : current D_i,Y,T,2,P,S,Time
- **T** : spin components, initial and current energy
- **I** : particle#,loss S,mass,Q,G,life time
- **IREP(IT)**
- **SORT(IT)**
- **(AMQ(J,I),J=1,5)**
- **RET(IT)**, **DPR(IT)**, **PS**
- **BORO**, **pass#**, **element#**, **keyword**, **labels**
4.5 Output Procedures

\begin{itemize}
  \item \texttt{LET}(I) : one-character string, for tagging particle number \( I \)
  \item \texttt{IEX}, \( I \), \texttt{IREP}(I) : flag, particle number, index
  \item \texttt{FO}(1 \rightarrow 6, I) : coordinates \( D, Y, T, Z, P \) and path length at the origin of the structure
  \item \texttt{F}(1 \rightarrow 6, I) : \textit{idem} at the current position
  \item \texttt{SORT}(I) : path length at which the particle has possibly been stopped (see \textsc{Chambr} or \textsc{Collima})
  \item \texttt{RET}(I), \texttt{DPR}(I) : synchrotron phase space coordinates ; \texttt{RET} = phase (radian), \texttt{DPR} = momentum dispersion (MeV/c) (see \textsc{Cavite})
  \item \texttt{IPASS} : turn number (see \textsc{Rebelete})
  \item \texttt{ipass} etc.
\end{itemize}

- \texttt{FAISTORE} has an effect similar to \texttt{FAISCNl}, with two more features.
  - On the first data line, \texttt{FNAME} may be followed by a series of up to 10 \texttt{LABEL}’s. If there is no label, the print occurs by default at the location of \texttt{FAISTORE} ; if there are labels the print occurs right downstream of all optical elements wearing those labels (and no longer at the \texttt{FAISTORE} location).
  - The next data line gives a parameter, \texttt{IP} : printing will occur at pass 1 and then at every \( IP \) other pass, if using \textsc{Rebelete} with \texttt{NPASS} \( \geq IP - 1 \).

For instance the following data input in \texttt{zgoubi.dat}:

\begin{verbatim}
FAISTORE
zgoubi.fai    HPCKUP    VPCKUP
12
\end{verbatim}

will result in output prints into \texttt{zgoubi.fai}, at pass 1 and then at every 12 other pass, each time elements of the \texttt{zgoubi.dat} data list labeled either \texttt{HPCKUP} or \texttt{VPCKUP} are encountered.

\textbf{Note}

Binary storage can be obtained from \texttt{FAISCNL} and \texttt{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 \textsc{zpop}.

This is achieved by giving the storage file a name of the form \texttt{b.FNAME} or \texttt{B.FNAME} (e.g., ‘b_zgoubi.fai’). The \texttt{FORTRAN WRITE} list is the same as in the \texttt{FORMATTED} case above.

This is compatible with the \texttt{READ} statements in \textsc{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, zgouib 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**:
  \[
  \text{FOCALE} : \quad X = XL \\
  \text{IMAGE[S]} : \quad X = -\frac{\sum_{i=1}^{N} Y_i \cdot \tan T_i - \left(\sum_{i=1}^{N} Y_i \cdot \sum_{i=1}^{N} \tan T_i\right)}{\sum_{i=1}^{N} \tan^2 T_i - \left(\sum_{i=1}^{N} \tan T_i\right)^2 / N} \\
  Y = Y_1 + X \cdot \tan T_1
  \]
  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**
  \[
  Y_M = \frac{1}{N} \sum_{i=1}^{N} (Y_i + X \cdot \tan T_i) - Y = \frac{1}{N} \sum_{i=1}^{N} Y \cdot M_i
  \]

- **FWHM of the image (IMAGE[S]) or of the beam (FOCALE), and total width, respectively, $W$ and $WT$**
  \[
  W = 2.35 \left(\frac{1}{N} \sum_{i=1}^{N} Y \cdot M_i^2 - Y \cdot M^2\right)^{\frac{1}{2}} \\
  WT = \max(YM_i) - \min(YM_i)
  \]
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 ($S = \text{path length}$; $D$ 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

\[
\begin{align*}
Y^+ &= \left( \frac{Y}{T_0} \right) T_0 + \left( \frac{Y}{T_0^2} \right) T_0^2 + \left( \frac{Y}{T_0^3} \right) T_0^3; \\
Y^- &= -\left( \frac{Y}{T_0} \right) T_0 + \left( \frac{Y}{T_0^2} \right) T_0^2 - \left( \frac{Y}{T_0^3} \right) T_0^3
\end{align*}
\]

will yield, neglecting third order terms,

\[
R_{11} = \frac{Y}{T_0} = \frac{Y^+ - Y^-}{2T_0}
\]

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

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

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

(4.5.1)

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

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\[.NN, NN=01,99\]), 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, \textit{IFOC}, acts as follows

- If \textit{IFOC} = 0, the transfer coefficients are calculated at the location of \textit{MATRIX}, and with respect to the reference trajectory. For instance, $Y^+$ and $T^+$ above are defined for particle number $i$ as $Y^+ = Y^+(i) - Y(\text{Ref})$, and $T^+ = T^+(i) - T(\text{ref})$.

- If \textit{IFOC} = 1, the transfer coefficients are calculated at the horizontal focus closest to \textit{MATRIX} (determined automatically), while the reference direction is that of the reference particle. For instance, $Y^+$ is defined for particle number $i$ as $Y^+ = Y^+(i) - Y_{\text{focus}}$, while $T^+$ is defined as $T^+ = T^+(i) - T(\text{ref})$.

- If \textit{IFOC} = 2, no change of reference frame is performed: the coordinates refer to the current frame. Namely, $Y^+ = Y^+(i)$, $T^+ = T^+(i)$, etc.

\textbf{Periodic Structures}

- If \textit{IFOC} = 10 + \textit{NPeriod}, then, from the 1-turn transport matrix as obtained in the way described above, \textit{MATRIX} calculates periodic parameters characteristic of the structure such as optical functions and tune numbers, assuming that it is \textit{NPeriod}-periodic, and in the coupled hypothesis, based on the Edwards-Teng method [30].

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

Addition of \textit{zgoubi.MATRIX.out} next to \textit{IORD}, \textit{IFOC} will cause stacking of \textit{MATRIX} output data into \textit{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.
PLOTDATA : Intermediate output for the PLOTDATA graphic software [52]

PLOTDATA was at the origin implemented for the purpose of plotting particle coordinates using the TRI-UMF 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 $\gamma$ 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 $\text{FNAME}$ (should post-processing with `zpop` follow, advised name is $\text{FNAME} = \text{‘zgoubi.spn’ (formatted write) or ‘b_zgoubi.spn’ (binary write)}$). The data are formatted and ordered according to the **FORTRAN** sequence found in the subroutine `spnprn.f`, with meaning of printed quantities as follows:

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

- **SPNSTORE** has an effect similar to **SPNPRNL**, with two more features.
  - On the first data line, $\text{FNAME}$ 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 $\text{NPASS} \geq IP - 1$.

    For instance the following data input in zgoubi.dat:
    ```
    SPNSTORE
    zgoubi.spn   HPCKUP   VPCKUP
    12
    ```

    will result in output prints into zgoubi.spn, every 12 other pass, each time elements of the zgoubi.dat data list labeled either $\text{HPCKUP}$ or $\text{VPCKUP}$ 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 $\text{b_FNAME}$ or $\text{B_FNAME}$ (e.g., ‘b_zgoubi.spn’). The **FORTRAN WRITE** output list is the same as in the **FORMATTED** case above.
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 $B_{\rho_{\text{ref}}}$ as defined in [MC]OBJET. However using *CAVITE* and indirectly *SCALING*, may affect the reference rigidity, following

$$B_{\rho_{\text{ref}}} = B_{\text{BORO}} \rightarrow B_{\rho_{\text{ref}}} = B_{\text{BORO}} + \delta B_{\rho_{s}}$$

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.

See Part D of the Guide.
Synchrotron acceleration can be simulated, using following the procedure:

- CAVITE appears in the zgoubi.dat data list (normally before REBELOTE), with option IOPT ≠ 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 CHANGREF(XCE, YCE, ALE) upstream of the optical element, followed by CHANGREF(XCS, YCS, ALS = -ALE) downstream of it, with computed XCS, YCS values as schemed in Fig. 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 ≠0,
- or, if ALE=0 by half the Z-rotation $\theta_Z/2$ such that $L = 2 \frac{BRO}{B1} \sin(\theta_Z/2)$, wherein $L$ = geometrical length, BRO = 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 on 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.

![Diagram](image)

**Figure 45**: Pitch angle, $\varphi$, in the “YR” type of rotation, using $KPOS = 4$. $M$ is the new position, in the rotated plane $(Y, Z_{\text{new}})$, of a particle with velocity $\overrightarrow{v}/M_oM$ 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 $P$.

![Diagram](image)

**Figure 46**: Positioning of a polar field map, using $KPOS = 1$. 
Polar Coordinates

If \( KPOS = 1 \), the element is positioned automatically in such a way that a particle entering with zero initial coordinates and \( 1 + DP = B\rho / \text{BORO} \) relative rigidity will reach position \((RM, AT/2)\) 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 \( XPAS = #E|C|S \), with \( E, C, S \) integers, \( E \) is the number of steps in the entrance fringe field region, \( C \) is the number of steps in the magnet body, and \( S \) is the number of steps in the exit fringe field region.

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 MCOBJET together with REBELOTE, put at the end of the optical structure, with its argument NPASS being the number of passes through REBELOTE, and \((NPASS + 1) * IMAX\) the number of particles to be ray-traced. In order that new initial conditions \((D, Y, T, Z, P, X)\) be generated at each pass, \( K = 0 \) has to be specified in REBELOTE.

Statistics on coordinates, spins, and other histograms can be performed by means of such procedures as HISTO, SPNTRK, etc. that stack the information from pass to pass.

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 \( \text{BORO} < 0 \) or \( D = B\rho / \text{BORO} < 0 \) when defining the initial coordinates with OBJET or MCOBJET.
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>AGSMM</td>
<td>AGS main magnet</td>
<td>175</td>
</tr>
<tr>
<td>AGSQAD</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>EMMMA</td>
<td>2-D Cartesian or cylindrical mesh field map for EMMMA 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>FAINSCAU</td>
<td>Print particle coordinates</td>
<td>209</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 $IP$ 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>GETFITVAL</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>Command</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>MATRIX</td>
<td>Calculation of transfer coefficients, periodic parameters</td>
<td>223</td>
</tr>
<tr>
<td>MCDESINT</td>
<td>Monte-Carlo simulation of in-flight decay</td>
<td>224</td>
</tr>
<tr>
<td>MCOBJET</td>
<td>Monte-Carlo generation of a 6-D object</td>
<td>225</td>
</tr>
<tr>
<td>MULTIPOLE</td>
<td>Magnetic multipole</td>
<td>228</td>
</tr>
<tr>
<td>OBJET</td>
<td>Generation of an object</td>
<td>229</td>
</tr>
<tr>
<td>OCTUPOLE</td>
<td>Object from Monte-Carlo simulation of decay reaction</td>
<td>231</td>
</tr>
<tr>
<td>OPTICS</td>
<td>Wien filter</td>
<td>232</td>
</tr>
<tr>
<td>ORDRE</td>
<td>Taylor expansions order</td>
<td>234</td>
</tr>
<tr>
<td>PARTICUL</td>
<td>Particle characteristics</td>
<td>235</td>
</tr>
<tr>
<td>PICKUPS</td>
<td>Beam centroid path; closed orbit</td>
<td>236</td>
</tr>
<tr>
<td>PLOTDATA</td>
<td>Intermediate output for the PLOTDATA graphic software</td>
<td>237</td>
</tr>
<tr>
<td>POISSON</td>
<td>Read magnetic field data from POISSON output</td>
<td>238</td>
</tr>
<tr>
<td>POLARMS</td>
<td>2-D polar mesh magnetic field map</td>
<td>239</td>
</tr>
<tr>
<td>PS170</td>
<td>Simulation of a round shape dipole magnet</td>
<td>240</td>
</tr>
<tr>
<td>QUADISEX</td>
<td>Sharp edge magnetic multipoles</td>
<td>241</td>
</tr>
<tr>
<td>QUADROPO</td>
<td>Quadrupole magnet</td>
<td>242</td>
</tr>
<tr>
<td>REBELOTE</td>
<td>‘Do it again’</td>
<td>244</td>
</tr>
<tr>
<td>RESET</td>
<td>Reset counters and flags</td>
<td>245</td>
</tr>
<tr>
<td>SCALING</td>
<td>Power supplies and R.F. function generator</td>
<td>246</td>
</tr>
<tr>
<td>SEPARA</td>
<td>Wien Filter - analytical simulation</td>
<td>247</td>
</tr>
<tr>
<td>SEXQUAD</td>
<td>Sharp edge magnetic multipole</td>
<td>248</td>
</tr>
<tr>
<td>SEXTUPOL</td>
<td>Sextupole magnet</td>
<td>249</td>
</tr>
<tr>
<td>SOLENOID</td>
<td>Solenoid</td>
<td>251</td>
</tr>
<tr>
<td>SPINR</td>
<td>Spin rotation</td>
<td>250</td>
</tr>
<tr>
<td>SPNPRNL</td>
<td>Store spin coordinates into file FNAME</td>
<td>252</td>
</tr>
<tr>
<td>SPNSTORE</td>
<td>Store spin coordinates every $IP$ other pass at labeled elements</td>
<td>252</td>
</tr>
<tr>
<td>SPNPT</td>
<td>Print spin coordinates</td>
<td>252</td>
</tr>
<tr>
<td>SPNTRK</td>
<td>Spin tracking</td>
<td>253</td>
</tr>
<tr>
<td>SRLOSS</td>
<td>Synchrotron radiation loss</td>
<td>254</td>
</tr>
<tr>
<td>SRFRTN</td>
<td>Print SR loss statistics</td>
<td>255</td>
</tr>
<tr>
<td>SYNRAD</td>
<td>Synchrotron radiation spectral-angular densities</td>
<td>256</td>
</tr>
<tr>
<td>TARGET</td>
<td>Generate a secondary beam following target interaction</td>
<td>191</td>
</tr>
<tr>
<td>TOSCA</td>
<td>2-D and 3-D Cartesian or cylindrical mesh field map</td>
<td>257</td>
</tr>
<tr>
<td>TRANSMAT</td>
<td>Matrix transfer</td>
<td>258</td>
</tr>
<tr>
<td>TRAROT</td>
<td>Translation-Rotation of the reference frame</td>
<td>259</td>
</tr>
<tr>
<td>TWISS</td>
<td>Calculation of periodic optical parameters</td>
<td>260</td>
</tr>
<tr>
<td>UNIDM</td>
<td>Undulator magnet</td>
<td>261</td>
</tr>
<tr>
<td>UNIPOT</td>
<td>Unipotential cylindrical electrostatic lens</td>
<td>262</td>
</tr>
<tr>
<td>VENUS</td>
<td>Simulation of a rectangular shape dipole magnet</td>
<td>263</td>
</tr>
<tr>
<td>WIENFILT</td>
<td>Wien filter</td>
<td>264</td>
</tr>
<tr>
<td>YMY</td>
<td>Reverse signs of $Y$ and $Z$ reference axes</td>
<td>265</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 Type</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<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 magnets</td>
<td>DIOPOLES, 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>OCTUPOLE, MULTIPOLE, QUADISEX, SEXQUAD</td>
</tr>
<tr>
<td>Quadrupole</td>
<td>QUADRUPO, MULTIPOLE, SEXQUAD, AGSQUAD</td>
</tr>
<tr>
<td>Sextupole</td>
<td>SEXTUPOLE, MULTIPOLE, 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

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

### ELECTRO-MAGNETIC ELEMENTS

<table>
<thead>
<tr>
<th>Element Type</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

$IL$  
$IL = 1, 2 \times 10^3$ : print field and coordinates along trajectories.  
0-2$[\times 10^3]$  

$MOD[MOD2], dL, R_0, dB1, dB2, dB3$  
Type of magnet model $^1$ [type of back-leg winding model $^2$] ;  
unused ; pole tip radius, 10 cm if set to zero ; relative error on dipole, quadrupole, sextupole component.  
2$\times$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 :  
number of windings, current.  
$\leq 2, NBLW \times $ (any, Amp.) (I, E)

$X_E, \lambda_E, E_2, E_3$  
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  
0-6, 6*no dim.  I, 6*E

$X_S, \lambda_S, S_2, S_3$  
Integration zone ; as for entrance  
2$\times$cm, 2$\times$no dim.  4$^*$E

$R_1, R_2, R_3$  
Skew angles of field components  
3*rad 10$^*$E

$XPAS$  
Integration step  
cm E

$KPOS, XCE, YCE, ALE$  
$KPOS=1$ : 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 Arccsin($B_1 XL / 2BRO$) if $ALE=0$  
$KPOS = 4$ : same as $KPOS=3$ however  
with possible X- or Y- or Z-misalignment or -rotation  
(under development)

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

$^2MOD2 = 0$ (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 = 10 ;  
$MM_{A20BD}$ : NBLW = 1, SIGN = 1.D0, NW = 10 ;  
$MM_{B02BF}$ : NBLW = 2, SIGN = 1.D0, NW1 = 12, NW2 = 6 ;  
$MM_{B04BD}$ : NBLW = 1, SIGN = 1.D0, NW = 10 ;  
$MM_{B05A}$ : NBLW = 1, SIGN = 1.D0, NW = 10 ;  
$MM_{B06A}$ : NBLW = 1, SIGN = 1.D0, NW = 5 ;  
$MM_{B07CD}$ : NBLW = 1, SIGN = 1.D0, NW = 5 ;  
$MM_{E02BD}$ : NBLW = 1, SIGN = 1.D0, NW = 5 ;  
$MM_{F01BF}$ : NBLW = 1, SIGN = 1.D0, NW = 5 ;  
$MM_{F04CAD}$ : NBLW = 1, SIGN = 1.D0, NW = 5 ;  
$MM_{F05A}$ : NBLW = 1, SIGN = 1.D0, NW = 5 ;  
$MM_{F06A}$ : NBLW = 1, SIGN = 1.D0, NW = 5 ;  
$MM_{F07CD}$ : 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.
<table>
<thead>
<tr>
<th><strong>AGSQUAD</strong></th>
<th><strong>AGS quadrupole</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IL</strong></td>
<td><em>IL</em> = 1, 2×10⁰ : print field and coordinates along trajectories.</td>
</tr>
<tr>
<td><strong>XL, R₀, IW₁, IW₂, IW₃</strong></td>
<td>Length of element ; radius at pole tip ; current in windings ;</td>
</tr>
<tr>
<td><strong>dIW₁, dIW₂, dIW₃</strong></td>
<td>relative error on currents.</td>
</tr>
</tbody>
</table>

**Entrance face**

| **Xₑ, λₑ** | Integration zone ; fringe field extent. (sharp edge if field extent is zero) | 2cm, 9 no dim. | 11E |

**Exit face**

| **NCE, C₀ − C₅** | Same as QUADRUPO | 0-6, 6 no dim. | I, 6E |
| **Xₛ, λₛ** | Integration zone ; as for entrance | 2cm, 9 no dim. | 11E |
| **NCS, C₀ − C₅** | 0-6, 6 no dim. | I, 6E |
| **R₁** | Roll angle | 10 rad | 10E |
| **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, 3E |
AIMANT

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

\[ B_z = F B_0 \left( 1 - N \left( \frac{R - R_M}{R_M} \right) + B \left( \frac{R - R_M}{R_M} \right)^2 + G \left( \frac{R - R_M}{R_M} \right)^3 \right) \]

**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*no dim.

**AT, ACENT, RM, RMIN, RMAX**
Mesh parameters: total angle of the map; azimuth for EFBs positioning; reference radius; minimum and maximum radii
2*deg, 3*cm

**ENTRANCE FIELD BOUNDARY**

\( \lambda, \xi \)
Fringe field extent (normally \( \approx \) gap size); flag:
- if \( \xi \geq 0 \): second order type fringe field with linear variation over distance \( \xi \)
- if \( \xi = -1 \): exponential type fringe field:

\[ F = \left( 1 + \exp(P(s)) \right)^{-1} \]

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

**NC, C0 − C5, shift**
NC = 1 + degree of \( P(s) \); \( C_0 \) to \( C_5 \): see above; EFB shift (ineffective if \( \xi \geq 0 \))
0-6, 6*no dim., cm

**ω+, θ, R1, U1, U2, R2**
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

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

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

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

**NC, C0 − C5, shift**
0-6, 6*no dim., cm

**ω−, θ, R1, U1, U2, R2**
Positioning and shape of the exit EFB
2*deg, 4*cm

(Note: \( \lambda = 0, \omega^- = -AT+ACENT \) and \( \theta = 0 \) for sharp edge)
**Keywords and input data formatting**

**IF NFACEx = 3**

LATERAL FIELD BOUNDARY

(See ENTRANCE FIELD BOUNDARY)

Next 3 records *only if NFACEx = 3*

\( \lambda, \xi \)

Fringe field parameters

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

Positioning and shape of the lateral EFB ;

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

RM3 is the radial position on azimuth ACENT

\( NBS \)

Option index for perturbations to the field map

\[-2 \leq 0 \text{ or } \geq 1\]

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

**If NBS = -1**

The map is modified as follows:

\( \theta_0, \Delta B/B_0 \)

\( B \) transforms to \( B \ast \left( 1 + \frac{\Delta B}{B_0} \frac{\theta - \theta_0}{AT} \right) \)

**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

\( \gamma, \alpha, \mu, \beta \)

Geometrical parameters of the shim

**IORDRE**

Degree of interpolation polynomial :

\( 2 \) = second degree, 9-point grid

\( 25 \) = second degree, 25-point grid

\( 4 \) = fourth degree, 25-point grid

\( XPAS \)

Integration step

**KPOS**

Positioning of the map, normally 2. Two options :

**If KPOS = 2**

Positioning as follows :

\( RE, TE, RS, TS \)

Radius and angle of reference, respectively, at entrance and exit of the map.

**If KPOS = 1**

Automatic positioning of the map, by means of reference relative momentum

\( DP \)

No dim.
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**

1. Equivalent to `CHANGREF (XCE = 0, YCE = Y(1), ALE = T(1))`  
   1-2 1

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

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

<table>
<thead>
<tr>
<th>I</th>
<th>1 : Equivalent to <code>CHANGREF</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>XCE = 0, YCE = Y(1), ALE = T(1))</code></td>
</tr>
</tbody>
</table>

<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>

<table>
<thead>
<tr>
<th></th>
<th>3*(1-IMAX)</th>
<th>3*I</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEAMBEAM</td>
<td>Beam-beam lens</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>SW, I</td>
<td>0/1 : off/on ; beam intensity. Use SPNTRK to activate spin kicks.</td>
<td></td>
</tr>
<tr>
<td>$\alpha_Y, \beta_Y, \epsilon_Y, \text{norm}/\pi$</td>
<td>Beam parameters, horizontal.</td>
<td></td>
</tr>
<tr>
<td>$\alpha_Z, \beta_Z, \epsilon_Z, \text{norm}/\pi$</td>
<td>Beam parameters, vertical.</td>
<td></td>
</tr>
<tr>
<td>$\sigma_X, \sigma_{dp/p}$</td>
<td>$rms$ bunch length ; $rms$ momentum spread.</td>
<td></td>
</tr>
<tr>
<td>$C, \alpha$</td>
<td>Ring circumference ; momentum compaction.</td>
<td></td>
</tr>
<tr>
<td>$Q_Y, Q_Z, Q_s$</td>
<td>Tunes, horizontal, vertical, synchrotron.</td>
<td></td>
</tr>
</tbody>
</table>
**BEND**

**Bending magnet, Cartesian frame**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Units</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>( IL )</td>
<td>( IL = 1, 2 \times 10^6 ) : print field and coordinates along trajectories.</td>
<td>0-2 ( \times 10^6 )</td>
<td>1</td>
</tr>
<tr>
<td>( XL, Sk, B1 )</td>
<td>Length ; skew angle ; field</td>
<td>cm, rad, kG</td>
<td>3*E</td>
</tr>
<tr>
<td>( X_E, \lambda_E, W_E )</td>
<td>Integration zone extent ; fringe field extent (normally ( \simeq ) gap height ; zero for sharp edge) ; wedge angle</td>
<td>cm, cm, rad</td>
<td>3*E</td>
</tr>
<tr>
<td>( N, C_0-C_5 )</td>
<td>Unused ; fringe field coefficients : ( B(s) = B1 F(s) ) with ( F(s) = 1/(1 + \exp(P(s)) ) and ( P(s) = \sum_{i=0}^5 C_i(s/\lambda)^i )</td>
<td>unused, 6*no dim.</td>
<td>1, 6*E</td>
</tr>
<tr>
<td>( X_S, \lambda_S, W_S )</td>
<td>See entrance face</td>
<td>cm, cm, rad</td>
<td>3*E</td>
</tr>
<tr>
<td>( N, C_0-C_5 )</td>
<td>unused. 6*no dim.</td>
<td>1, 6*E</td>
<td></td>
</tr>
<tr>
<td>( XPAS )</td>
<td>Integration step</td>
<td>cm</td>
<td>E</td>
</tr>
</tbody>
</table>

**KPOS, XCE, YCE, ALE**

\( KPOS = 1 \) : element aligned, 2 : misaligned ; shifts, tilt (unused if \( KPOS = 1 \))

\( KPOS = 3 \) :
- 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 \( \arcsin(B1XL / 2BRO) \) 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 \((O, X, Y, Z)\).
**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.  
\[\leq 20, \geq 1, 0 - 9 \ 3*1\]

**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\[×10^n\] : print field and coordinates along trajectories.

**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  
\( ≤ 400 \)  I

**FNAME [, SUM] 1, 2**  

File name  
A80

**ID, A, B, C**  

Integration boundary. Ineffective when ID = 0.  
\( ≥ −1, 2*\) no dim., cm \[,2*\no \ dim., cm, etc.\]\( I,3*E \)

\([, A', B', C', B''\text{, etc.}, \text{if } ID ≥ 2]\)

**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)  
I-2, 2*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:

```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 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. Superimposing (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

(all maps must all have their mesh defined in identical coordinate frame).
CARTEMES

2-D Cartesian uniform mesh magnetic field map
mid-plane symmetry is assumed

IC, IL

IC = 1, 2 : print the map
IL = 1, 2\times10^{n} : print field and coordinates along trajectories.

BNORM, XN, YN

Field and X-, Y-coordinate normalization coeff.s.

TITL

Title. Start with “FLIP” to get field map X-flipped.

IX, JY

Number of longitudinal (IX) and transverse (JY) nodes of the map

\leq 400, \leq 200

FNAME

File name

ID, A, B, C

Integration boundary. Normally ID = 0.

\geq -1, 2*no dim., 1, 3*E

\text{, etc., if } ID \geq 2

IORDRE

Degree of interpolation polynomial (see DIPOLE-M)

\leq 400, \leq 200

XPAS

Integration step cm

YCE, ALE

shifts, tilt (unused if KPOS=1)

IORDRE

Degree of interpolation polynomial (see DIPOLE-M)

\leq 400, \leq 200

FNAME

File name

ID, A, B, C

Integration boundary. Normally ID = 0.

\geq -1, 2*no dim., 1, 3*E

\text{, etc., if } ID \geq 2

IORDRE

Degree of interpolation polynomial (see DIPOLE-M)

\leq 400, \leq 200

XPAS

Integration step cm

YCE, ALE

shifts, tilt (unused if KPOS=1)

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

\begin{verbatim}
OPEN (UNIT = NL, FILE = FNAME, STATUS = ‘OLD’ [,FORM=’UNFORMATTED’])
IF (BINARY) THEN
READ(NL) (Y(J), J=1, JY)
ELSE
READ(NL,100) (Y(J), J=1, JY)
ENDIF
100 FORMATE(10 F8.2)
DO 1 I=1,IX
IF (BINARY) THEN
READ(NL,101) X(I), (BMES(I,J), J=1, JY)
ELSE
READ(NL,101) X(I), (BMES(I,J), J=1, JY)
ENDIF
1 CONTINUE
\end{verbatim}

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**

\[ \Delta W = qV \sin(2\pi f \Delta t + \phi_s) \]

and other voltage and frequency laws.

**IOPT[i]**

Option. \( i = 1 \) causes info output into zgoubi.CAVITE.out

<table>
<thead>
<tr>
<th>IOPT</th>
<th>0-7</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta W = )</td>
<td>qV \sin(2\pi f \Delta t + \phi_s)</td>
<td>and other voltage and frequency laws.</td>
</tr>
</tbody>
</table>

If IOPT=0

Element inactive

\( X, X \)

Unused

\( X, X \)

Unused

If IOPT=1

\( f_{RF} \) follows the timing law given by **SCALING**

\( \mathcal{L}, h \)

Reference closed orbit length ; harmonic number

m, no dim. 2*E

\( \tilde{V}, X \)

R.F. peak voltage ; unused

V, unused 2*E

If IOPT=2

\( f_{RF} \) follows \( \Delta W_s = q\tilde{V}\sin\phi_s \)

\( \mathcal{L}, h \)

Reference closed orbit length ; harmonic number

m, no dim. 2*E

\( \tilde{V}, \phi_s \)

R.F. peak voltage ; synchronous phase

V, rad 2*E

If IOPT=3

No synchrotron motion : \( \Delta W = q\tilde{V}\sin\phi_s \)

\( X, X \)

Unused ; unused

2*unused 2*E

\( \tilde{V}, \phi_s \)

R.F. peak voltage ; synchronous phase

V, rad 2*E

If IOPT=6

Read RF frequency and/or phase law from external file, “zgoubi.freqLaw.In”.

\( \mathcal{L}, E_k \)

Orbit length and kinetic energy at start of acceleration.

m, MeV 2*E

\( \tilde{V}, \Phi_s \)

R.F. peak voltage ; synchronous phase.

V, rad 2*E

If IOPT=7

Quasi- or isochronous acceleration.

\( X, E_k \)

Unused ; RF frequency :

- , Hz 2*E

\( \tilde{V}, \Phi_s \)

R.F. peak voltage ; synchronous phase.

V, rad 2*E

---

1. Use **PARTICUL** to declare mass and charge.
2. For ramping the R.F. frequency following \( B_{\rho}(t) \), use **SCALING**, with family **CAVITE**.

---

---
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 \ \simeq \ 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° cm, deg 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’**

Parameters in the **CHANGREF** procedure.

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 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.
10 step size
1 0. 0. 0.
'CHANGREF'
YS -1. ZR -6.341 First Y-shift back, next half Z-rotate.
'DRIFT'
10.
'FAISCEAU'
'END'
```
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 
ergy; scattering angle; angle of the target

\[ NT, NP \]

Number of samples in \( T \) and \( P \) coordinates after CIBLE

\[ TS, PS, DT \]

Sample step sizes; tilt angle

\[ BORO \]

New reference rigidity after CIBLE

Scheme of the principles of CIBLE (TARGET)

\[ A, T = \text{position, angle of incoming particle 2 in the entrance reference frame} \]
\[ P = \text{position of the interaction} \]
\[ B, T = \text{position, angle of the secondary particle in the exit reference frame} \]
\[ \theta = \text{angle between entrance and exit frames} \]
\[ \beta = \text{tilt angle of the target} \]
COLLIMA Collimator

IA 0: element inactive
1: element active
2: element active and print information on stopped particles

Physical-space collimation

$\text{IFORM}[J, C^1, C^2, C^3, C^4]$  
$\text{IFORM} = 1$: rectangular aperture;  
$\text{J} = 0$, default: opening is $\pm YL = \pm C^1, \pm ZL = \pm C^2$, centered at $YC = C^3, ZC = C^4$.  
$\text{J} = 1$: opening is, in $Y: [C^1, C^2]$; in $Z: [C^3, C^4]$

Longitudinal collimation

$\text{IFORM}, H_{\text{min}}, H_{\text{max}}, V_{\text{min}}, V_{\text{max}}, J=1$ or 2 for vertical variable resp.

Phase-space collimation

$\text{IFORM}, \alpha, \beta, \epsilon/\pi, N_{\sigma}$  
$\text{IFORM} = 11, 14$: horizontal collimation; horizontal ellipse parameters (unused if 14), 2, emittance, cut-off  
$\text{IFORM} = 12, 15$: vertical collimation; vertical ellipse parameters (unused if 15), 2, emittance, cut-off  
$\text{IFORM} = 13, 16$: longitudinal collimation; to be implemented

---

1 Any particle out of limits is stopped.
2 The rejection boundary is the rms ellipse matched to the particle distribution.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Format</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IL</strong></td>
<td>$IL = 1, 2\times 10^n$</td>
<td>$0.2 \times 10^n$</td>
<td>I</td>
</tr>
<tr>
<td><strong>XL, R_0, B_0</strong></td>
<td>Length: radius and field at pole tip</td>
<td>$2^\text{cm}, \text{kG}$</td>
<td>$3^E$</td>
</tr>
<tr>
<td><strong>X_E, \lambda_E</strong></td>
<td>Entrance face: Integration zone extent; fringe field extent ($\leq 2R_0, \lambda_E = 0$ for sharp edge)</td>
<td>$2^\text{cm}$</td>
<td>$2^E$</td>
</tr>
<tr>
<td><strong>NCE, C_0 - C_5</strong></td>
<td>$NCE = \text{unused}$</td>
<td>unused, $6^E$</td>
<td></td>
</tr>
<tr>
<td><strong>X_S, \lambda_S</strong></td>
<td>Exit face: see entrance face</td>
<td>$2^\text{cm}$</td>
<td>$2^E$</td>
</tr>
<tr>
<td><strong>NCS, C_0 - C_5</strong></td>
<td>Exit face: see entrance face</td>
<td>$0-6, 6^\text{no dim.}$</td>
<td>$3^E$</td>
</tr>
<tr>
<td><strong>XPAS</strong></td>
<td>Integration step</td>
<td>cm</td>
<td>E</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>
<td>1-2, $2^\text{cm}, \text{rad}$</td>
<td>$3^E$</td>
</tr>
</tbody>
</table>

![Diagram of a decapole magnet](image)
**DIPOL**

**Dipole magnet, polar frame**

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

**IL**

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

**AT, RM**

Total angular extent of the dipole ; reference radius \( \text{deg, cm} \) 2*E

**ACENT, B₀, N, B, G**

Azimuth for positioning of EFBs ; field and field indices \( \text{deg., kG, 3*no dim.} \) 5*E

**ENTRANCE FIELD BOUNDARY**

\( \lambda, \xi \)

Fringe field extent (normally \( \simeq \) gap size) ; unused. \( \text{cm, unused} \) 2*E

Exponential type fringe field \( F = \frac{1}{1 + \exp(P(s))} \)

with \( P(s) = C_0 + C_1 \left( \frac{s}{\lambda} \right) + C_2 \left( \frac{s}{\lambda} \right)^2 + \ldots + C_5 \left( \frac{s}{\lambda} \right)^5 \)

**NC, C₀ – C₅, shift**

Unused ; \( C_0 \) to \( C_5 \) : see above ; EFB shift \( 0-6, 6*\text{no dim.}, \text{cm} \) 1,7*E

**ω⁺, θ, R₁, U₁, U₂, R₂**

Azimuth of entrance EFB with respect to \( \text{ACENT} \) ; wedge angle of EFB ; radii and linear extents of EFB (use \( |U_{1,2}| = \infty \) when \( R_{1,2} = \infty \)) \( 2*\text{deg}, 4*\text{cm} \) 6*E

**EXIT FIELD BOUNDARY**

(See ENTRANCE FIELD BOUNDARY)

**LATERAL FIELD BOUNDARY**

(See ENTRANCE FIELD BOUNDARY)

\( \lambda, \xi \)

Fringe field parameters \( \text{cm, unused} \) 2*E

**NC, C₀ – C₅, shift**

\( 0-6, 6*\text{no dim.}, \text{cm} \) 1,7*E

**ω⁻, θ, R₁, U₁, U₂, R₂**

Positioning and shape of the exit EFB \( 2*\text{deg}, 4*\text{cm} \) 6*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 XPAS/Resol

\( \text{2, 25 or 4 ; no dim.} \) I, E

**XPAS**

Integration step \( \text{cm} \) 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. \( \text{cm, rad, cm, rad} \) 4*E

**IF KPOS = 1**

Automatic positioning of the map, by means of reference relative momentum \( \text{no dim.} \) E
**DIPOLE-M**

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

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

**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

\[ \text{kg}, 3^*\text{no dim.} \]

**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 \( \approx \) gap size); unused.

Exponential type fringe field \( F = 1 / (1 + \exp(P(s))) \)

with \( P(s) = C_0 + C_1 \left( \frac{s}{\lambda} \right) + C_2 \left( \frac{s}{\lambda} \right)^2 + \ldots + C_5 \left( \frac{s}{\lambda} \right)^5 \)

\[ NC, C_0 - C_5, \text{shift} \]

Unused; \( C_0 \) to \( C_5 \): see above; EFB shift

\[ 0-6, 6^*\text{no dim., cm} \]

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

Azimuth of entrance EFB with respect to ACENT; wedge angle of EFB; radii and linear extents of EFB (use \(| U_{1,2} | = \infty \) when \( R_{1,2} = \infty \))

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

**EXIT FIELD BOUNDARY**

(See ENTRANCE FIELD BOUNDARY)

\( \lambda, \xi \)

Fringe field parameters

\[ \text{cm, unused} \]

\[ 2^*\text{E} \]

\[ NC, C_0 - C_5, \text{shift} \]

\[ 0-6, 6^*\text{nodim., cm} \]

\[ 1.7^*\text{E} \]

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

Positioning and shape of the exit EFB

\[ 2^*\text{deg}, 4^*\text{cm} \]

\[ 6^*\text{E} \]

(Note: \( \lambda = 0, \omega^- = -AT + ACENT \) and \( \theta = 0 \) for sharp edge)

**IF NFACES = 3**

**LATERAL FIELD BOUNDARY**

(See ENTRANCE FIELD BOUNDARY)

Next 3 records only if NFACES = 3

\( \lambda, \xi \)

Fringe field parameters

\[ \text{cm, (cm)} \]

\[ 2^*\text{E} \]

\[ NC, C_0 - C_5, \text{shift} \]

\[ 0-6, 6^*\text{no dim., cm} \]

\[ 1.7^*\text{E} \]

\( \omega^- , \theta , R_1, U_1, U_2, R_2, \text{RM3} \)

Positioning and shape of the lateral EFB:

RM3 is the radial position on azimuth ACENT

\[ 2^*\text{deg}, 5^*\text{cm} \]

\[ 7^*\text{E} \]

**NBS**

Option index for perturbations to the field map

\[ \text{normally 0} \]

\[ 1 \]

**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 \( B \ast \left( 1 + \frac{\Delta B}{B_0} \frac{R - R_{\text{RM}}}{R_{\text{RM}} - R_{\text{MIN}}} \right) \)

\[ \text{cm, no dim.} \]

\[ 2^*\text{E} \]

**IF NBS = -1**

The map is modified as follows:

\[ R_0, \Delta B / B_0 \]

\( B \) transforms to \( B \ast \left( 1 + \frac{\Delta B}{B_0} \frac{\theta - \theta_0}{AT} \right) \)

\[ \text{deg, no dim.} \]

\[ 2^*\text{E} \]
### Keywords and input data formatting

**If NBS ≥ 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

- $XPAS$
  - Integration step

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

  **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 $DP$
    - Reference relative momentum
      - no dim.  E
**DIPOLES**

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

(i) \[ B_Z = \sum_{i=1}^{N} B_{Z0,i} F_i(R, \theta) \left( 1 + b_1 (R - R M_i)/R M_i + b_2 (R - R M_i)^2/R M_i^2 + ... \right) \]

(ii) \[ B_Z = B_{Z0,i} + \sum_{i=1}^{N} F_i(R, \theta) \left( b_1 (R - R M_i) + b_2 (R - R M_i)^2 + ... \right) \]

**IL**

\[ IL = 1, 2 \times 10^n : \text{print field and coordinates along trajectories.} \]

**N, AT, RM**

Number of magnets in the N-tuple ;

<table>
<thead>
<tr>
<th>Field</th>
<th>Coordinates along trajectories</th>
</tr>
</thead>
<tbody>
<tr>
<td>deg, cm</td>
<td>1, 2*E</td>
</tr>
</tbody>
</table>

Repeat \(N\) times the following sequence

ACN, \(\delta R M\), \(B_0\), \(ind, b_i\), \((i = 1, ind)\)

Positioning of EFBs : azimuth, \(RM_i = RM + \delta RM\) ; field ;

<table>
<thead>
<tr>
<th>Number of, and field coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>((ind + 1))*no dim.</td>
</tr>
</tbody>
</table>

**ENTRANCE FIELD BOUNDARY**

\(g_0, \kappa\)

Fringe field extent \((g = g_0 (RM/R)^c)\) cm, no dim. 2*E

Exponential type fringe field \(F = 1 / (1 + \exp(P(s)))\)

with \(P(s) = C_0 + C_1 (\frac{s}{g}) + C_2 (\frac{s}{g})^2 + ... + C_5 (\frac{s}{g})^5\)

**NC, \(C_0 - C_5\), shift**

Unused ; \(C_0\) to \(C_5\) : see above ; EFB shift

<table>
<thead>
<tr>
<th>Field</th>
<th>Coordinates along trajectories</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-6, 6*no dim., cm</td>
<td>1, 7*E</td>
</tr>
</tbody>
</table>

**\(\omega^+, \theta, R_1, U_1, U_2, R_2\)**

Azimuth of entrance EFB with respect to ACN ;

<table>
<thead>
<tr>
<th>Azimuth of entrance EFB with respect to ACN</th>
</tr>
</thead>
<tbody>
<tr>
<td>2<em>deg, 4</em>cm</td>
</tr>
</tbody>
</table>

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)

**LATERNAL 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

**\(\omega^-, \theta, R_1, U_1, U_2, R_2, R_3\)**

2*deg, 5*cm 7*E

(Note : \( g_0 = 0, \omega^- = -AT^- ACENT, \theta = 0 \) and KIRD=0 for sharp edge)

**End of repeat**

---

1 Non-zero \(\delta R M\) requires KIRD= 2, 4 or 25.
**Keywords and input data formatting**

**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 \( \text{XPAS}/\text{Resol} \)
  - KIRD=2 or 25: second degree, 9- or 25-point grid
  - KIRD=4: fourth degree, 25-point grid

**XPAS**

Integration step

- cm

**KPOS**

Positioning of the magnet, normally 2. Two options:

- If **KPOS = 2**
  - Positioning as follows:
    - \( R_E, T_E, R_S, T_S \)
    - Radius and angle of reference, respectively, at entrance and exit of the magnet
  - cm, rad, cm, rad

- If **KPOS = 1**
  - Automatic positioning of the magnet, by means of reference relative momentum
  - no dim.

**E**
### DODECAPO Magnet

#### IL

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print field and coordinates along trajectories.</td>
<td>0-2[$10^5$]</td>
<td>I</td>
</tr>
</tbody>
</table>

#### Xl, R0, B0

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length: radius and field at pole tip</td>
<td>2*cm, kG</td>
<td>3*E</td>
</tr>
</tbody>
</table>

#### XE, λE

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entrance face</td>
<td>2*cm</td>
<td>2*E</td>
</tr>
<tr>
<td>Integration zone extent: fringe field extent ($\leq 2R_0, \lambda_E = 0$ for sharp edge)</td>
<td>2*cm</td>
<td>2*E</td>
</tr>
</tbody>
</table>

#### NCE, C0 - C5

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCE = unused</td>
<td>unused</td>
<td>I, 6*E</td>
</tr>
<tr>
<td>Fringe field coefficients such that $G(s) = G_0/(1 + \exp P(s))$, with $G_0 = B_0/R_0^5$ and $P(s) = \sum_{i=0}^{5} C_i(s/\lambda)^i$</td>
<td>6*no dim.</td>
<td></td>
</tr>
</tbody>
</table>

#### XS, λS

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exit face: see entrance face</td>
<td>2*cm</td>
<td>2*E</td>
</tr>
</tbody>
</table>

#### NCS, C0 - C5

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exit face: see entrance face</td>
<td>0-6, 6*no dim.</td>
<td>I, 6*E</td>
</tr>
</tbody>
</table>

#### XPAS

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integration step</td>
<td>cm</td>
<td>E</td>
</tr>
</tbody>
</table>

#### KPOS, XCE, YCE, ALE

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element aligned, 2: misaligned; shifts, tilt (unused if $KPOS=1$)</td>
<td>1-2, 2*cm, rad</td>
<td>I, 3*E</td>
</tr>
</tbody>
</table>
Keywords and input data formatting

**DRIFT, ESL**

Field free drift space

\[ XL \text{ length cm} \quad E \]
EBMULT

**Electro-magnetic multipole**

*IL*  
*IL* = 1, 2[×10^n] : print field and coordinates along trajectories.  
0-2[×10^n]

**Electric poles**

*XL, R₀, E₁, E₂, ..., E₁₀*  
Length of element ; radius at pole tip ;  
field at pole tip for dipole, quadrupole,  
..., 20-pole electric components  
2*cm, 10*V/m  
12*E

**Entrance face**

*Xₑ, λₑ, E₂, ..., E₁₀*  
Integration zone ; fringe field extent :  
dipole fringe field extent = λₑ ;  
quadrupole fringe field extent = λₑ * E₂ ;  
...  
20-pole fringe field extent = λₑ * E₁₀  
(for any component : 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**

*Xₛ, λₛ, S₂, ..., S₁₀*  
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 electric field components  
10*rad  
10*E

**Magnetic poles**

*XL, R₀, B₁, B₂, ..., B₁₀*  
Length of element ; radius at pole tip ;  
field at pole tip for dipole, quadrupole,  
..., 20-pole magnetic components  
2*cm, 10*kG  
12*E

**Entrance face**

*Xₑ, λₑ, E₂, ..., E₁₀*  
Integration zone ; fringe field extent :  
dipole fringe field extent = λₑ ;  
quadrupole fringe field extent = λₑ * E₂ ;  
...  
20-pole fringe field extent = λₑ * E₁₀  
(for any component : 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

---

1 Use PARTICUL to declare mass and charge.
### Keywords and input data formatting

<table>
<thead>
<tr>
<th>Keyword(s)</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_S, \lambda_S, S_2, \ldots, S_{10}$</td>
<td>Exit face: Integration zone; as for entrance</td>
<td>2<em>cm, 9</em>no dim. 11*E</td>
</tr>
<tr>
<td>$NCS, C_0 - C_5$</td>
<td></td>
<td>0-6, 6<em>no dim. 1, 6</em>E</td>
</tr>
<tr>
<td>$R_1, R_2, R_3, \ldots, R_{10}$</td>
<td>Skew angles of magnetic field components</td>
<td>10<em>rad 10</em>E</td>
</tr>
<tr>
<td>$XPAS$</td>
<td>Integration step</td>
<td>cm E</td>
</tr>
<tr>
<td>$KPOS, XCE, YCE, ALE$</td>
<td>$KPOS=1$: element aligned, 2: misaligned; shifts, tilt (unused if $KPOS=1$)</td>
<td>1-2, 2<em>cm, rad 1, 3</em>E</td>
</tr>
</tbody>
</table>

![Diagram of an Electrode and Magnetic Pole](image-url)
EL2TUB  

Two-tube electrostatic lens

$IL$  
$IL = 1, 2 \times 10^9$ : print field and coordinates along trajectories.  
$0-2 \times 10^9$  
$I$

$X_1, D, X_2, R_0$  
Length of first tube ; distance between tubes ; length of second tube ; inner radius  
$3m$  
$4E$

$V_1, V_2$  
Potentials  
$2V$  
$2E$

$XPAS$  
Integration step  
$cm$  
$E$

$KPOS, XCE, YCE, ALE$  
$KPOS=1$ : element aligned, 2 : misaligned ; shifts, tilt (unused if $KPOS=1$)  
$1-2, 2cm, 1, 3E$

Two-electrode cylindrical electric lens.

1 Use PARTICUL to declare mass and charge.
<table>
<thead>
<tr>
<th><strong>IL</strong></th>
<th>Electrostatic N-electrode mirror/lens, straight slits</th>
</tr>
</thead>
<tbody>
<tr>
<td>$IL = 1, 2 \times 10^9$</td>
<td>print field and coordinates along trajectories. 0-2[$\times 10^9$] I</td>
</tr>
<tr>
<td>N, L1, ..., LN, D, MT</td>
<td>Number of electrodes; electrode lengths; gap; mode (11/H-mir, 12/V-mir, 21/V-lens, 22/H-lens) 2 - 7, N<em>m, m I, N</em>E, E, I</td>
</tr>
<tr>
<td>V1, ..., VN</td>
<td>Electrode potentials (normally $V1 = 0$) N<em>V N</em>E</td>
</tr>
<tr>
<td>XPAS</td>
<td>Integration step cm E</td>
</tr>
<tr>
<td>KPOS, XCE, YCE, ALE</td>
<td>KPOS=1: element aligned; 2: misaligned; 1-2, 2<em>cm, rad I, 3</em>E shifts, tilt (unused if KPOS=1); 3: automatic positioning, $YCE =$ pitch, $ALE =$ half-deviation</td>
</tr>
</tbody>
</table>

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

**IL**

\[ IL = 1, 2 \times 10^n \] : print field and coordinates along trajectories.  

0-2\([\times 10^n]\) I

**R1, R2, AT, D**

Radius of first and second slits; total deviation angle; gap  

4*m 4*E  

2*m, rad, m 4*E

**V – VA, VB – V**

Potential difference  

2*V 2*E

**XPAS**

Integration step  

cm E

**KPOS**

Normally \( KPOS = 2 \) for positioning;  

1-2 I

**RE, TE, RS, TS**

Radius and angle at respectively entrance and exit.  

cm, rad, cm, rad 4*E

---

**Diagram Description**

Electrostatic N-electrode mirror/lens, circular slits, in the case \( N = 3 \), in horizontal mirror mode.
**ELMULT**

**Electric multipole**

<table>
<thead>
<tr>
<th><strong>IL</strong></th>
<th>IL = 1, 2([\times10^n]) : print field and coordinates along trajectories.</th>
<th>0-2([\times10^n])</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>XL, R₀, E₁, E₂, ..., E₁₀</strong></td>
<td>Length of element ; radius at pole tip ; field at pole tip for dipole, quadrupole, ..., dodecapole components</td>
<td>2<em>cm, 10</em>V/m</td>
<td>12*E</td>
</tr>
</tbody>
</table>

**Entrance face**

| **XE, λE, E₂, ..., E₁₀** | Integration zone ; fringe field extent ; dipole fringe field extent = λE ; quadrupole fringe field extent = λE * E₂ ; ... 20-pole fringe field extent = λE * E₁₀ (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. | I, 6*E |

**Exit face**

| **XS, λS, S₂, ..., S₁₀** | Integration zone ; as for entrance | 2*cm, 9*no dim. | 11*E |
| **NCS, C₀ - C₅** | 0-6, 6*no dim. | I, 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  
**1-D uniform mesh electric field map**  
X-axis cylindrical symmetry is assumed

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Value Range</th>
<th>Units</th>
</tr>
</thead>
</table>
| IC, IL   | IC = 1, 2: print the map  
IL = 1, 2×10^n: print field and coordinates along trajectories. | 0-2, 0-2×10^n | 2*I |
| ENORM, X-NORM | 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 | ≤ 400 | I |
| FNAME   | File name | A80 |
| ID, A, B, C | Integration boundary. Ineffective when ID = 0. | ≥ -1, 2*no dim., cm | 1,3*E |
| [. A', B', C', B'' etc., if ID ≥ 2] | | 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*cm, rad | 1, 3*E |

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:

```
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[$\times 10^m$] 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 i
MOD ≤ 19 : Cartesian mesh ;
MOD ≥ 20 : cylindrical mesh ;
i, optional, tells the reading FORMAT, default is ’*’.

**FNAME**
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.
ID = -1, 1 or ≥ 2 : as for CARTEMES
≥ -1, 2*no dim., 1-2, 2*cm, rad 1, 3*E

**IORDRE**
If IZ = 1 : as in CARTEMES
If IZ ≠ 1 : unused 2, 25 or 4 I

**XPAS**
Integration step cm E

**KPOS, XCE, YCE, ALE**
KPOS=1 : element aligned, 2 : misaligned ;
KPOS=1 : element aligned, 2 : misaligned ;
shifts, tilt (unused if KPOS=1) 1-2, 2*cm, rad I, 3*E

---

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.
<table>
<thead>
<tr>
<th><strong>FAISCEAU</strong></th>
<th><strong>Print particle coordinates</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Print particle coordinates at the location where the keyword is introduced in the structure.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>FAISCNL</strong></th>
<th><strong>Store particle coordinates in file FNAME</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FNAME</strong>&lt;sup&gt;1&lt;/sup&gt;</td>
<td>Name of storage file (e.g., zgoubi.fai, or bzgoubi.fai for binary storage).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>FAISTORE</strong></th>
<th><strong>Store coordinates every IP other pass [, at elements with appropriate label]</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FNAME</strong>&lt;sup&gt;1&lt;/sup&gt; [, <strong>LABEL(s)</strong>]</td>
<td>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 <strong>FNAME</strong> or first <strong>LABEL</strong> is 'none' then no storage occurs. Store occurs at all elements if first <strong>LABEL</strong> is 'all' or 'ALL'.</td>
</tr>
<tr>
<td><strong>IP</strong></td>
<td>Store every IP other pass (when using <strong>REBELOTE</strong> with <strong>NPASS</strong> ≥ <strong>IP</strong> – 1).</td>
</tr>
</tbody>
</table>

<sup>1</sup> Stored data can be read back from **FNAME** using **OBJET, KOBJ** = 3.
**FFAG magnet, \textit{N}-tuple**

**UNDER DEVELOPMENT**

\[ B_Z = \sum_{i=1}^{N} B_{Z0_i} \mathcal{F}_i(R, \theta) (R/R_{M,i})^{K_i} \]

**IL**

\[ IL = 1, 2 \times 10^n \] : print field and coordinates along trajectories.

**N, AT, RM**

Number of dipoles in the FFAG \textit{N}-tuple ; no dim.

total angular extent of the dipole ; reference radius

\[ \text{deg, cm} \]

Repeat \textit{N} times the following sequence

**ACN, \delta RM, B_{Z0}, K**

Azimuth for dipole positioning ; \( R_{M,i} = R_M + \delta RM \) ;
field at \( R_{M,i} \) ; index

\[ \text{deg, cm, kG, no dim.} \]

**ENTRANCE FIELD BOUNDARY**

\( g_0, \kappa \)

Fringe field extent (\( g = g_0 (R_M/R)^\kappa \))

\[ \text{cm, no dim.} \]

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

Unused ; \( C_0 \) to \( C_5 \) : fringe field coefficients ; EFB shift

\[ 0-6, 6 \times \text{no dim, cm} \]

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

Azimuth of entrance EFB with respect to \textit{ACN} ;

wedge angle of EFB ; radii and linear extents of EFB (use \( |U_{1,2}| = \infty \) when \( R_{1,2} = \infty \))

(Note : \( g_0 = 0, \omega^+ = \text{ACENT}, \theta = 0 \) and KIRD=0 for sharp edge)

**EXIT FIELD BOUNDARY**

(See ENTRANCE FIELD BOUNDARY)

\( g_0, \kappa \)

\[ \text{cm, no dim} \]

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

\[ 0-6, 6 \times \text{no dim, cm} \]

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

\[ 2 \times \text{deg, 4 \times cm} \]

(Note : \( g_0 = 0, \omega^- = -AT + \text{ACENT}, \theta = 0 \) and KIRD=0 for sharp edge)

**LATERAL FIELD BOUNDARY**

to be implemented - following data not used

\( g_0, \kappa \)

\[ \text{cm, no dim} \]

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

\[ 0-6, 6 \times \text{no dim, cm} \]

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

\[ 2 \times \text{deg, 4 \times cm} \]

**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 \( \text{XPAS}/\text{Resol} \)

KIRD=2 or 25 : second degree, 9- or 25-point grid

KIRD=4 : fourth degree, 25-point grid

**XPAS**

Integration step

\[ \text{cm} \]

**KPOS**

Positioning of the magnet, normally 2. Two options :

\[ 1-2 \]

**If KPOS = 2**

\( RE, TE, RS, TS \)

Radius and angle of reference, respectively, at entrance and exit of the magnet

\[ \text{cm, rad, cm, rad} \]

**If KPOS = 1**

\( DP \)

Automatic positioning of the magnet, by means of reference relative momentum

\[ \text{no dim.} \]
**FFAG-SPI**  
**Spiral FFAG magnet, N-tuple**  
**UNDER DEVELOPMENT**  
\[ B_Z = \sum_{i=1}^{N} B_{Z0,i} \mathcal{F}_i(R, \theta) (R/R_{M,i})^{K_i} \]

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

**N, AT, RM**  
Number of dipoles in the FFAG 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\( Z_0, K \)**  
Azimuth for dipole positioning ; \( R_{M,i} = R_M + \delta R_M \) ;  
field at \( R_{M,i} \) ; index  
deg, cm, kG, no dim.  
4*E

**ENTRANCE FIELD BOUNDARY**

\( g_0, \kappa \)  
Fringe field extent \( (g = g_0 (R_M/R)^\kappa) \)  
cm, no dim.  
2*E

\( NC, C_0 - C_5, \) shift  
Unused ; \( C_0 \) to \( C_5 \) : fringe field coefficients ; EFB shift  
0-6, 6*no dim, cm  
1,7*E

\( \omega^+, \xi, \) 4 dummies  
Azimuth of entrance EFB with respect to \( ACN \) ;  
spiral angle ; 4 \times unused  
2*deg, 4*unused  
6*E

**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

\( \omega^-, \xi, \) 4 dummies  
2*deg, 4*unused  
6*E

**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  
1

**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
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 \( \leq 20 \)  

**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 (±)  

\( \leq MXL, \leq MXD, 2*E \)

**or :**

\[ IR, IP, XC, [V_{min}, V_{max}] \]

\( \leq MXL, \leq MXD, 2*E \)

\( NC \), penalty \([, ITER]\) \(3\)

Number of constraints \([, \) penalty \([, \) number of iterations]\]. \( \leq 20 \), \([10^{-n}, > 0]\)  

**For I = 1, NC**

repeat NC times the following sequence :  

\[ IC, I, J, IR, V, WV, NP [, p_i (i = 1, NP)] \]

\( IC, I \) and \( J \) define the type of constraint (see table below) ;  
\( IR \) : number of the element after which the constraint applies ;  
\( V \) : value ; \( WV \) : weight (the stronger the lower \( WV \))  
\( NP \) : number of parameters ; if \( NP \geq 1, p_i (i = 1, NP) \) : parameter values.

\( 2*\text{no dim.}, \text{curr. units} \)

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>Object definition (recommended)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sigma )-matrix</td>
<td>( \sigma_{IJ} ) (( \sigma_{11} = \beta_Y, \sigma_{12} = \sigma_{21} = \alpha_Y ), etc.)</td>
<td>OBJET/KOBJ=5.6</td>
</tr>
<tr>
<td>Periodic parameters</td>
<td>( Y )-tune = ( \mu_Y / 2\pi ) ( \text{for MATRIX block 1-9} )</td>
<td>OBJET/KOBJ=5.0N</td>
</tr>
<tr>
<td>First order transport coeffs.</td>
<td>( i \neq 8 : \text{YY-determinant} ; i=8 : \text{YZ-determinant} )</td>
<td>OBJET/KOBJ=5</td>
</tr>
<tr>
<td>Second order transport coeffs.</td>
<td>( j \neq 7 : \text{ZZ-determinant} ; j=7 : \text{ZY-determinant} )</td>
<td>OBJET/KOBJ=6</td>
</tr>
<tr>
<td>Trajectory coordinates</td>
<td>( F(J, I) ) ( &lt; F(J, i) &gt;_{i=1,IMAX} ) ( \text{Sup}(F(J, i))</td>
<td>i=1,IMAX )</td>
</tr>
<tr>
<td></td>
<td>( \text{Dist}(F(J, I)</td>
<td>i=1,12,dt )</td>
</tr>
<tr>
<td></td>
<td>( F(J, I) - F_0(J, I) )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( F(J, I) - F(J, K)</td>
<td>(K = 1 - IMAX) )</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>S_{X,Y,Z}(I) - \tilde{S}(I)</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>S_{X,Y,Z}(I) - \tilde{S}(I)</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>\tilde{S}(I)</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>S_{X,Y,Z}(I) - \tilde{S}(I)</td>
</tr>
</tbody>
</table>

**Keywords and input data formatting**

- **Type of constraint**: Object definition
- **Parameters defining the constraints**: # values
- **Object definition (recommended)**: OBJET/KOBJ=5.6, OBJET/KOBJ=5.0N, OBJET/KOBJ=5, OBJET/KOBJ=6, [MC]OBJET
<table>
<thead>
<tr>
<th><strong>FOCALE</strong></th>
<th><strong>Particle coordinates and horizontal beam size at distance</strong> $XL$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$XL$</td>
<td>Distance from the location of the keyword cm E</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>FOCALEZ</strong></th>
<th><strong>Particle coordinates and vertical beam size at distance</strong> $XL$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$XL$</td>
<td>Distance from the location of the keyword cm E</td>
</tr>
<tr>
<td>Keywords</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>GASCAT</td>
<td>Gas scattering</td>
</tr>
<tr>
<td>KGA</td>
<td>Off/On switch</td>
</tr>
<tr>
<td>AI, DEN</td>
<td>Atomic number; density</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.
Keywords and input data formatting

**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**
For each momentum group, as classified by means of $OBJET, KOBJ = 1, 2$ or $4$.
### MAP2D

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

| **IC, IL** | $IC = 1, 2 : \text{print the field map}$ | $0-2, 0-2 \times \times 10^n$ | **2*I** |
| **IL** | $IL = 1, 2 \times 10^n : \text{print field and coordinates along trajectories}$ |

| **BNORM, XN, YN** | Field and X-, Y-coordinate normalization coeffs. | $3 \times \text{no dim.}$ | **3*E** |

| **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) | $\leq 400, \leq 200$ | **2*I** |

| **FNAME** | File name | A80 |

| **ID, A, B, C** | Integration boundary. Ineffective when $ID = 0$. | $\geq -1, 2 \times \text{no dim.}$ | **I,3*E** |

| $[A', B', C', B''$, etc., if $ID \geq 2]$ | cm $[2 \times \text{no dim.}, \text{cm}, \text{etc.}]$ |

| **IORDRE** | Degree of polynomial interpolation, 2nd or 4th order. | 2, 4 | **I** |

| **XPAS** | Integration step | cm | **E** |

| **KPOS, XCE, YCE, ALE** | $KPOS=1 : \text{element aligned, 2 : misaligned ;}$ | $1-2, 2 \times \text{cm, rad}$ | **I, 3*E** |

| shifts, tilt (unused if $KPOS=1$) |

---

1. **FNAME** (e.g., magnet.map) contains the field map data. These must be formatted according to the following FORTRAN read sequence (normally compatible with TOSCA code OUTPUTS - details and possible updates are to be found in the source file ‘fmapw.f’):

   ```fortran
   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
   100 FORMAT (1X, 6E11.4)
   ENDIF
   1 CONTINUE
   ```

   where $X(I), Y(J)$ are the longitudinal, horizontal coordinates in the at nodes $(I, J)$ of the mesh, $Z(1)$ is the vertical elevation of the map, and $BX, BY, BZ$ are the components of the field.

   For binary files, **FNAME** must begin with ’B’ or ’b’; a logical flag ’Binary’ will then automatically be set to ’.TRUE.’
### MAP2D-E

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

| **IC, IL** | **IC = 1, 2**: print the field map  
**IL = 1, 2[×10^n]**: 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 \( ID = 0. \)  
\( ID = -1, 1 \) or \( \geq 2 \) : as for CARTEMES** |  
| **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)** |  

| **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 \((I, J)\); \( Z(1) \) is the vertical elevation of the map, and \( E_X, E_Y, E_Z \)
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.’

---

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’.

---

*Keywords and input data formatting*
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**

<table>
<thead>
<tr>
<th>IORD, IFOC</th>
<th>Options:</th>
</tr>
</thead>
<tbody>
<tr>
<td>[. zgoubi.MATRIX.out]</td>
<td>0-2, 0-1 or &gt; 10 2*1 [.A]</td>
</tr>
</tbody>
</table>

**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 = 0:** matrix at actual location,
reference \(\equiv\) particle # 1

**IFOC = 1:** matrix at the closest first order horizontal focus,
reference \(\equiv\) 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 Monte-Carlo simulation of in-flight decay
M₁ → M₂ + M₃

INFO, ² M₂, M₃, τ₂ ³ [Switch] ; masses of the two decay products;
COM lifetime of particle 2
J₁, J₂, J₃ Seeds for random number generators

Particle 1 decays into 2 and 3 ; zgoubi then calculates trajectory of 2, while 3 is discarded. θ and φ are the scattering angles of particle 2 relative to the direction of the incoming particle 1. They transform to T₂ and P₂ in Zgoubi frame.

¹ MCDESINT must be preceded by 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 τ₂ 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

<table>
<thead>
<tr>
<th><strong>BORO</strong></th>
<th>Reference rigidity</th>
<th>kG.cm</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>KOBJ</strong></td>
<td>Type of support of the random distribution</td>
<td>1-3</td>
<td>I</td>
</tr>
<tr>
<td>KOBJ= 1 : window</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KOBJ= 2 : grid</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KOBJ= 3 : phase-space ellipses</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>IMAX</strong></td>
<td>Number of particles to be generated</td>
<td>(\leq 10^4)</td>
<td>I</td>
</tr>
<tr>
<td><strong>KY, KT, KZ, KP, KX, KD</strong></td>
<td>Type of probability density</td>
<td>6*(1-3)</td>
<td>6*I</td>
</tr>
</tbody>
</table>

**Y_0, T_0, Z_0, P_0, X_0, D_0**  
Mean value of coordinates \((D_0 = B\rho/BORO)\)  
m, rad, m, rad, m, no dim.

**If KOBJ = 1**  
In a window

| \(\delta Y, \delta T, \delta Z, \delta P\) | Distribution widths, depending on \(KY, KT\) etc. | m, rad, m, rad, m, no dim. | 6*E |
| \(\delta X, \delta D\) | | | |
| **N_{SY}, N_{ST}, N_{SZ}, N_{SP}, N_{SX}, N_{SD}\)** | Sorting cut-offs (used only for Gaussian density) | units of \(\sigma_Y, \sigma_T\), etc. | 6*E |
| **N_0, C_0, C_1, C_2, C_3\)** | Parameters involved in calculation of \(P(D)\) | no dim. | 5*E |
| **IR1, IR2, IR3\)** | Random sequence seeds | \(3^*\approx 10^6\) | 3*I |

---

1. Let \(x = Y, T, Z, P\) or \(X, KY, KT, KZ, KP\) and \(KX\) can take the values

1. uniform, \(p(x) = 1/2\delta x\) if \(-\delta x \leq x \leq \delta x\)
2. Gaussian, \(p(x) = \exp(-x^2/2\delta x^2)/\delta x\sqrt{2\pi}\)
3. parabolic, \(p(x) = 3(1 - x^2/\delta x^2)/4\delta x\) if \(-\delta x \leq x \leq \delta x\)

**K D** can take the values

1. uniform, \(p(D) = 1/2\delta D\) if \(-\delta D \leq x \leq \delta D\)
2. exponential, \(p(D) = \exp(C_0 + C_1 l + C_2 l^2 + C_3 l^3)\) if \(-\delta D \leq x \leq \delta D\)
3. kinematic, \(D = \delta D * T\)
If KOBJ = 2  

On a grid  

\( IY, IT, IZ, IP, IX, ID \)  

Number of bars of the grid  

\( PY, PT, PZ, PP, PX, PD \)  

Distances between bars  

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

Width of the bars (±) if uniform,  

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)  

Units of \( \sigma_Y, \sigma_T, \text{etc.} \)  

\( N_0, C_0, C_1, C_2, C_3 \)  

Parameters involved in calculation of \( P(D) \)  

\( IR1, IR2, IR3 \)  

Random sequence seeds  

6*I  

6*E  

6*E  

ibidem  

6*E  

5*E  

3*\( \simeq 10^6 \)  

3*I  

If KOBJ = 3  

On a phase-space ellipse  

\( \alpha_Y, \beta_Y, \epsilon_Y / \pi, N_{\sigma_Y}, \)  

Ellipse parameters and  

Emittance, Y-T phase-space ; cut-off  

\( N_{\sigma_Y} \) if \( N_{\sigma_Y} < 0 \)  

\( \)  

4*E [E]  

4*E [E]  

4*E [E]  

4*E [E]  

3*\( \simeq 10^6 \)  

3*I  

\[ 1 + \frac{\alpha_Y^2}{\beta_Y^2} \frac{Y^2}{2} + 2 \alpha_Y \beta_Y YT + \frac{\epsilon_Y}{\pi} T^2 = \frac{\epsilon_Y}{\pi} \]  

if \( N_{\sigma_Y} > 0 \), or, if \( N_{\sigma_Y} < 0 \) sorting within the ring  

\[ |N_{\sigma_Y}, N_{\sigma_Y}'| \]  

\[ 1 + \frac{\alpha_Y^2}{\beta_Y^2} \frac{Y^2}{2} + 2 \alpha_Y \beta_Y YT + \frac{\epsilon_Y}{\pi} T^2 = \frac{\epsilon_Y}{\pi} \]  

if \( N_{\sigma_Y} > 0 \), or, if \( N_{\sigma_Y} < 0 \) sorting within the ring  

\[ |N_{\sigma_Y}, N_{\sigma_Y}'| \]  

\[ 1 + \frac{\alpha_Y^2}{\beta_Y^2} \frac{Y^2}{2} + 2 \alpha_Y \beta_Y YT + \frac{\epsilon_Y}{\pi} T^2 = \frac{\epsilon_Y}{\pi} \]  

if \( N_{\sigma_Y} > 0 \), or, if \( N_{\sigma_Y} < 0 \) sorting within the ring  

\[ |N_{\sigma_Y}, N_{\sigma_Y}'| \]  

\[ 1 + \frac{\alpha_Y^2}{\beta_Y^2} \frac{Y^2}{2} + 2 \alpha_Y \beta_Y YT + \frac{\epsilon_Y}{\pi} T^2 = \frac{\epsilon_Y}{\pi} \]  

if \( N_{\sigma_Y} > 0 \), or, if \( N_{\sigma_Y} < 0 \) sorting within the ring  

\[ |N_{\sigma_Y}, N_{\sigma_Y}'| \]  

\[ 1 + \frac{\alpha_Y^2}{\beta_Y^2} \frac{Y^2}{2} + 2 \alpha_Y \beta_Y YT + \frac{\epsilon_Y}{\pi} T^2 = \frac{\epsilon_Y}{\pi} \]  

if \( N_{\sigma_Y} > 0 \), or, if \( N_{\sigma_Y} < 0 \) sorting within the ring  

\[ |N_{\sigma_Y}, N_{\sigma_Y}'| \]  

Similar possibilities, non-random, are offered with OBJET, KOBJ=8 (p. 230)  

Works with Gaussian density type only : sorting within the ellipse frontier  

\[ 1 + \frac{\alpha_Y^2}{\beta_Y^2} \frac{Y^2}{2} + 2 \alpha_Y \beta_Y YT + \frac{\epsilon_Y}{\pi} T^2 = \frac{\epsilon_Y}{\pi} \]  

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 *MCOBJET* when *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  

0-2\( \times 10^n \)  

**XL, R0, B1, B2, ..., B10.**  

Length of element; radius at pole tip; field at pole tip for dipole, quadrupole, ..., dodecapole components  

2\*cm,10\*kG  

**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 = \( \lambda_E \times E_2 \); ...  

2\*cm,9\*no dim.  

**NCE, C_0 - C_5**  

same as **QUADRUPO**

0-6, 6*no dim.  

**Exit face**

\( X_S, \lambda_S, S_2, ..., S_{10} \)  

Integration zone; as for entrance  

2\*cm, 9*no dim.  

**R1, R2, R3, ..., R10**  

Skew angles of field components  

10*rad  

**XPAS**  

Integration step  

cm 

**KPOS, XCE, YCE, ALE**

\( KPOS=1 \): element aligned; 2: misaligned; shifts, tilt (unused if \( KPOS=1 \))  

\( KPOS = 3 \): effective only if \( B1 \neq 0 \); 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\( \text{Arcsin}(B1 XL / 2BRO) \) if \( ALE=0 \)
**OBJET**  
Generation of an object

**BORO**  
Reference rigidity  
kG.cm E

**KOBJ[.K2]**  
Option index [.More options]  
1-6 1

**If KOBJ = 1[.01]**  
[Non-] Symmetric object

**IY, IT, IZ, IP, IX, ID**  
Ray-Tracing assumes mid-plane symmetry  
Total number of points in \(\pm Y, \pm T, \pm Z, \pm P\)  
\(+Z, +P\) with KOBJ = 1.01, \(\pm X\).  
and \(\pm D\) coordinates (\(\pm Y \leq 20, \ldots, \pm D \leq 20\))

**PY, PT, PZ, PP, PX, PD**  
Step size in \(Y, T, Z, P, X\) and momentum  
\((PD = \delta B_{p}/BORO)\)  
\(2(cm,mrad), \text{cm, no dim.} 6^*E\)

**YR, TR, ZR, PR, XR, DR**  
Reference (\(DR = B_{p}/BORO\))  
\(2(cm,mrad), \text{cm, no dim.} 6^*E\)

**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)

**For I = 1, IMAX**  
Repeat IMAX times the following line

**Y, T, Z, P, X, D, LET**  
Coordinates and tagging of the IMAX particles ;  
If KOBJ = 2.01 input units are different :  
\(2(cm,mrad), \text{cm, no dim.,} 6^*E, A1\)

**IEX(I = 1, IMAX)**  
IMAX times 1 or -9. If \(IEX(I) = 1\) trajectory \(I\) is  
ray-traced, it is not if \(IEX(I) = -9\).

**If KOBJ=3[.NN,  
NN=00...03]**  
Reads coordinates from a storage file  
\(NN=00\) (default) : [b_]zgoubi.fai like data file FORMAT  
\(NN=01 : \text{read FORMAT is 'READ(NL, *) Y, T, Z, P, S, DP'}\)  
\(NN=02 : \text{read FORMAT is 'READ(NL, *) X, Y, Z, PX, PY, PZ'}\)  
\(NN=03 : \text{read FORMAT is 'READ(NL, *) DP, Y, T, Z, P, S, TIME, MASS, CHARGE'}\)

**IT1, IT2, ITStep**  
Read particles numbered IT1 to IT2, step ITStep  
\((\text{For more than} \ 10^4 \ \text{particles stored in FNAME, use 'REBELOTE'})\)

**IP1, IP2, IPStep**  
Read particles that belong in pass numbered  
\(\geq 1, \geq IP1, \geq 1\)  
\(\geq 1, \geq IP2, \text{step IPStep}\)

**YF, TF, ZF, PF,  
XF, DF, TF, TAG**  
Scaling factor. TAG-ing letter : no effect if '*'  
otherwise only particles with \(\text{TAG=}LET\) are retained.

**YR, TR, ZR, PR,  
XR, DR, TR**  
Reference. Given the previous line of data,  
all coordinate C is transformed to \(C*CF+CR\ \text{cm, no dim.\mu s}\)

**InitC**  
0 : set \(\text{new } R_{0} = \text{old } R_{0}, \text{new } R = \text{old } R\) ;  
1 : set \(\text{new } R_{0} = \text{old } R_{0}, \text{new } R = \text{old } R\) ;  
2 : save \(\text{old } R\) in \(\text{new } R_{0}\), set \(\text{new } R = \text{old } R_{0}\).

**FNAME**  
File name (e.g., zgoubi.fai)  
\(A80\)  
\(\text{NN in KOBJ=3.NN determines storage FORMAT})\)

**If KOBJ = 5[.NN,  
NN=01,99]**  
Generation of 11 particles, or 11*NN if \(I \geq 2\) (for use with MATRIX, IORD = 1)
Step sizes in $Y$, $T$, $Z$, $P$, $X$ and $D$  

Reference trajectory $(DR = B\rho/BORO)$

If $KOBJ = 5.01$

Initial beam ellipse parameters $^1$

Additional data line:

Reference trajectory $(DR = B\rho/BORO)$

If $KOBJ = 5.0N$

Additional data lines:

Reference trajectory $^{#i} (DR = B\rho/BORO)$

If $KOBJ = 6$

Generation of 61 particles (for use with $MATRX$, $IORD = 2$)

Step sizes in $Y$, $T$, $Z$, $P$, $X$ and $D$  

Reference trajectory $(DR = B\rho/BORO)$

If $KOBJ = 7$

Object with kinematics

Number of points in $\pm Y$, $\pm T$, $\pm Z$, $\pm P$, $\pm X$; $ID$ is not used

Step sizes in $Y$, $T$, $Z$, $P$, $X$; $PD$ = kinematic coefficient, such that $D(T) = DR + PD*T$

Reference $(DR = B\rho/BORO)$

If $KOBJ = 8$

Generation of phase-space coordinates on ellipses $^2$

Number of samples in each 2-D phase-space; if zero the central value (below) is assigned

Central values $(D_0 = B\rho/BORO)$

Ellipse parameters and emittances

They can be transported by using $MATRX$

Similar possibilities, random, are offered with $MCOBJET$, $KOBJ=3$ (p. 226)

$^1$ They can be transported by using $MATRX$

$^2$ Similar possibilities, random, are offered with $MCOBJET$, $KOBJ=3$ (p. 226)
<table>
<thead>
<tr>
<th><strong>OBJJETA</strong></th>
<th><strong>Object from Monte-Carlo simulation of decay reaction</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$M_1 + M_2 \rightarrow M_3 + M_4$ and $M_4 \rightarrow M_5 + M_6$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>BORO</strong></th>
<th><strong>Reference rigidity</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>kG.cm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>IBODY, KOBJ</strong></th>
<th><strong>Body to tracked:</strong> $M_3$ (IBODY=1), $M_5$ (IBODY=2), $M_6$ (IBODY=3) : type of distribution for $Y_0$ and $Z_0$ : uniform (KOBJ= 1) or Gaussian (KOBJ= 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-3,1-2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>IMAX</strong></th>
<th><strong>Number of particles to be generated (use 'REBELOTE' for more)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\leq 10^4$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>$M_1 - M_6$</strong></th>
<th><strong>Rest masses of the bodies</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$6^*\text{GeV/c}^2$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>$T_1$</strong></th>
<th><strong>Kinetic energy of incident body</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GeV</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>$Y_0, T_0, Z_0, P_0, D_0$</strong></th>
<th><strong>Only those particles in the range</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$Y_0 - \delta Y \leq Y \leq Y_0 + \delta Y$</td>
</tr>
<tr>
<td></td>
<td>$D_0 - \delta D \leq D \leq D_0 + \delta D$</td>
</tr>
<tr>
<td></td>
<td>will be retained</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>$\delta Y, \delta T, \delta Z, \delta P, \delta D$</strong></th>
<th><strong>2(cm,mrad), no dim.</strong></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>$XL$</strong></th>
<th><strong>Half length of object:</strong> $-XL \leq X_0 \leq XL$ (uniform random distribution)**</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>$IR1, IR2$</strong></th>
<th><strong>Random sequence seeds</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$2^*\sim 0^6$</td>
</tr>
</tbody>
</table>
**OCTUPOLE**

**Octupole magnet**

**IL**

$IL = 1, 2 \times 10^6$ : print field and coordinates along trajectories.

0-2[$\times 10^6$] I

**XL, R₀, B₀**

Length ; radius and field at pole tip of the element

2*cm, kG 3*E

**Entrance face :**

$X_E, \lambda_E$

Integration zone ;

Fringe field extent ($\lambda_E = 0$ for sharp edge)

2*cm 2*E

**NCE, C₀ − C₅**

$NCE = \text{unused}$

$C₀ − C₅ = \text{fringe field coefficients}$

such that : $G(s) = G₀/(1 + \exp P(s))$, with $G₀ = B₀/R₀^3$

and $P(s) = \sum_{i=0}^{5} C_i (s/\lambda)^i$

any, 6*no dim. I, 6*E

**Exit face :**

$X_S, \lambda_S$

Parameters for the exit fringe field ; see entrance

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

---

Octupole magnet
OPTICS Write out optical functions

\( I_{OPT} \), label, \( I_{MP} \)

\( I_{OPT} = 0/1 \) : Off/On. Transport the beam matrix ;

0-1, string, 0-1 I, A, I

'label' : Can be 'all', 'ALL', or existing 'LABEL.1(NOEL)' ;

\( I_{MP} = 1 \) 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 $\tilde{R}$ and $\tilde{u}$ up to $\tilde{u}^{(IO)}$ (default is $IO = 4$)</td>
</tr>
</tbody>
</table>
Keywords and input data formatting

PARTICUL  Particle characteristics

$M, Q, G, \tau, X$  Mass ; charge ; gyromagnetic factor ; COM life-time ; unused  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 *LABEL’s* at which beam centroid is to be recorded

For \( I = 1, N \), a list of \( N \) records follows

**LABEL’s**  
N labels at which beam centroid is to be recorded

N string(s)  
N*A10
<table>
<thead>
<tr>
<th>Keywords and input data formatting</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PLOTDATA</strong></td>
</tr>
</tbody>
</table>

*To be documented.*
### POISSON

**Read magnetic field data from POISSON output**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC, IL</td>
<td>Print the field map. IC = 1, 2: print the field map. IL = 1, 2[×10^6]: print field and coordinates along trajectories.</td>
<td>0-2, 0-2[×10^6] 2*I</td>
</tr>
<tr>
<td>BNORM, XN,YN</td>
<td>Field and X-, Y-coordinate normalization coeffs.</td>
<td>3<em>no dim. 3</em>E</td>
</tr>
<tr>
<td>TITL</td>
<td>Title. Start with “FLIP” to get field map X-flipped</td>
<td>A80</td>
</tr>
<tr>
<td>IX, IY</td>
<td>Number of longitudinal and transverse nodes of the uniform mesh</td>
<td>≤ 400, ≤ 200 2*I</td>
</tr>
<tr>
<td>FNAME</td>
<td>File name</td>
<td>A80</td>
</tr>
<tr>
<td>ID, A, B, C</td>
<td>Integration boundary. Ineffective when ID = 0.</td>
<td>≥ −1, 2<em>no dim., cm [,2</em>no dim., cm, etc.] I,3*E</td>
</tr>
<tr>
<td>[A', B', C', B'', etc., if ID ≥ 2]</td>
<td>as for CARTEMES</td>
<td></td>
</tr>
<tr>
<td>IORDRE</td>
<td>Degree of interpolation polynomial as for DIPOLE-M</td>
<td>2, 25 or 4 I</td>
</tr>
<tr>
<td>XPAS</td>
<td>Integration step</td>
<td>cm E</td>
</tr>
<tr>
<td>KPOS, XCE, YCE, ALE</td>
<td>KPOS=1: element aligned, 2: misaligned; shifts, tilt (unused if KPOS=1)</td>
<td>1-2, 2<em>cm, rad I, 3</em>E</td>
</tr>
</tbody>
</table>

---

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”:

```fortran
I = 0
11 CONTINUE
I = I+1
READ(LUN,101,ERR=99,END=10) K, K, X(I), R, R, B(I)
101 FORMAT(I1, I3, I4, E15.6, 2F11.5, 2F12.3)
GOTO 11
```

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

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

mid-plane symmetry is assumed

**IC, IL**  
\( IC = 1, 2 \): print the map  
\( IL = 1, 2 \times 10^n \): print field and coordinates along trajectories.

**BNORM, AN,RN**  
Field and A-,R-coordinate normalization coeffs.  
\[ 3^* \text{no dim.} \]

**TITL**  
Title. Start with “FLIP” to get field map X-flipped  
A80

**IA, JR**  
Number of angular and radial nodes of the mesh  
\[ \leq 400, \leq 200 \]

**FNAME**  
File name  
A80

**ID, A, B, C**  
Integration boundary. Ineffective when \( ID = 0 \).  
\[ ID = -1, 1 \text{ or } \geq 2 \text{ : as for CARTEMES} \]

\[ \geq -1, 2^* \text{no dim.}, \text{cm} \]

\[ \text{cm}, 2^* \text{no dim.}, \text{cm}, \text{etc.} \]

**IORDRE**  
Degree of interpolation polynomial  
(see DIPOLE-M)  
2, 25 or 4

**XPAS**  
Integration step  
cm

**KPOS**  
\text{as for DIPOLE-M. Normally 2.}  
1-2

**FNAME**  
(e.g., spes2.map) contains the field data.  
These must be formatted according to the following FORTRAN read sequence - details and possible updates are to be found in the source file ’fmapw.f’:

\[
\text{OPEN (UNIT = NL, FILE = FNAME, STATUS = ‘OLD’ [,FORM=’UNFORMATTED’])}
\]

\[
\text{IF (BINARY) THEN}
\]

\[
\text{READ(NL) (Y(J), J=1, JY)}
\]

\[
\text{ELSE}
\]

\[
\text{READ(NL,100) (Y(J), J=1, JY)}
\]

\[
\text{ENDIF}
\]

\[
100 \quad \text{FORMAT(10 F8.2)}
\]

\[
\text{DO 1 I = 1,IX}
\]

\[
\text{IF (BINARY) THEN}
\]

\[
\text{READ (NL) X(I), (BMES(I,J), J=1, JY)}
\]

\[
\text{ELSE}
\]

\[
\text{READ(NL,101) X(I), (BMES(I,J), J=1, JY)}
\]

\[
\text{ENDIF}
\]

\[
101 \quad \text{FORMAT(10 F8.1)}
\]

\[
1 \quad \text{CONTINUE}
\]

where \( X(I) \) and \( Y(J) \) are the longitudinal and transverse coordinates and \( BMES \) is the \( Z \) field component at a node \( (I,J) \) of the mesh. For binary files, FNAME must begin with ‘B’, or ‘b’, ‘Binary’ will then automatically be set to ‘.TRUE.’
**PS170**

Simulation of a round shape dipole magnet

**IL**

ILI = 1, 2 \times 10^n \) : print field and coordinates along trajectories.

**XL, R_0, B_0**

Length of the element, radius of the circular dipole, field

**XPAS**

Integration step

**KPOS, XCE, YCE, ALE**

KPOS=1 : element aligned, 2 : misaligned ;

shifts, tilt (unused if KPOS=1)

---

Scheme of the PS170 magnet simulation.
**QUADISEX**  

**Sharp edge magnetic multipoles**  

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

**IL**  

\( IL = 1, 2 \times 10^9 \): print field and coordinates along trajectories. \( 0-2 \times 10^9 \) I

**XL, R₀, B₀**  

Length of the element; normalization distance; field 2*cm, kG 3*E

**N, EB₁, EB₂, EG₁, EG₂**  

Coefficients for the calculation of B. 5*no dim. 5*E

if \( Y > 0 \): \( B = EB₁ \) and \( G = EG₁ \);  
if \( Y < 0 \): \( B = EB₂ \) and \( G = EG₂ \).

**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
<table>
<thead>
<tr>
<th><strong>QUADRUPO</strong></th>
<th><strong>Quadrupole magnet</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><em>IL</em></td>
<td><em>IL</em> = 1, 2 [10^n] : print field and coordinates along trajectories.</td>
</tr>
<tr>
<td><em>XL</em>, <em>R_0</em>, <em>B_0</em></td>
<td>Length ; radius and field at pole tip</td>
</tr>
<tr>
<td><em>X_E</em>, <em>λ_E</em></td>
<td>Integration zone extent ; fringe field extent (≈ 2<em>R_0</em>, <em>λ_E</em> = 0 for sharp edge)</td>
</tr>
<tr>
<td><em>NCE</em>, <em>C_0</em> − <em>C_5</em></td>
<td><em>NCE</em> = unused</td>
</tr>
<tr>
<td><em>X_S</em>, <em>λ_S</em></td>
<td>See entrance face</td>
</tr>
<tr>
<td><em>XPAS</em></td>
<td>Integration step</td>
</tr>
<tr>
<td><strong>KPOS</strong>, <strong>XCE</strong>, <strong>YCE</strong>, <strong>ALE</strong></td>
<td><em>KPOS</em>=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if <em>KPOS</em>=1)</td>
</tr>
</tbody>
</table>

**Entrance face :**

- *2*cm, kG
- any, 6*no dim. I, 6*E

**Exit face**

- *2*cm
- 0-6, 6*no dim. I, 6*E

- cm
- E

- 1-2, 2*cm, rad I, 3*E
Keywords and input data formatting

A Quadrupole magnet

Scheme of the elements QUADRUPO, SEXTUPOLE, OCTUPOLE, DECAPELE, DODECAPO, and MULTIPOL

The length of the element is $XL$, but trajectories are calculated from $-X_E$ to $XL + X_S$, by means of automatic prior and further $X_E$ and $X_S$ translations.
REBELOTE Jump to the beginning of zgoubi input 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);  
\[NFAM = \text{number of families to be scaled}\]  

**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**  

\[NT > 0: \text{number of timings}\]  
\[NT = -1: \text{field scaling factor updated by CAVITE}\]  
\[NT = -2: \text{RF law in CAVITE is read from external data file}\]  

**SCL(I), I = 1, NT**  
Scaling values (a single one, normally 1, if \(NT = -1\)). relative \(NT\)\(^*\)E

**TIM(I), I = 1, NT**  
Corresponding timings, in units of turns (1 if \(NT = -1\)). turn number \(NT\)\(^*\)I
Keywords and input data formatting

Wien Filter - analytical simulation

$IA, XL, E, B,$

$IA = 0$: element inactive
$IA = 1$: horizontal separation
$IA = 2$: vertical separation ;
Length of the separator ; electric field ; magnetic field.

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**

\[ B_Z |_{Z=0} = B_0 \left( \frac{N}{R_0} Y + \frac{P}{R_0^2} Y^2 + \frac{Q}{R_0^3} Y^3 \right) \]

**IL**

\( IL = 1, 2 \times 10^9 \) : print field and coordinates along trajectories. 0-2\( \times 10^9 \) I

**XL, R_0, B_0**

Length of the element; normalization distance; field 2\( \times \)cm, kG 3\( \times \)E

**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 \( G = EG2 \).

\( N \), EB1, EB2, EG1, EG2

5\( \times \)no dim. 5\( \times \)E

**XPAS**

Integration step cm E

**KPOS, XCE, YCE, ALE**

**KPOS=1** : element aligned, 2 : misaligned;

1-2, 2\( \times \)cm, rad I, 3\( \times \)E

Shifts, tilt (unused if \( KPOS=1 \))
### Sextupole Magnet

#### IL

$IL = 1, 2 \times 10^9$: print field and coordinates along trajectories.  
0-2 $\times 10^9$  
I

#### XL, $R_0, B_0$

Length; radius and field at pole tip of the element  
2 cm, kG  
3 E

#### Entrance face:

$X_E, \lambda_E$

Integration zone; fringe field  
extent ($\lambda_E = 0$ for sharp edge)  
2 cm  
2 E

$NCE, C_0 - C_5$

$NCE = \text{unused}$  
$C_0 - C_5 = \text{Fringe field coefficients such that}$  
$G(s) = G_0/(1 + \exp P(s))$, with $G_0 = B_0/R_0^2$  
and $P(s) = \sum_{i=0}^{5} C_i(s/\lambda)^i$

$X_S, \lambda_S$

Parameters for the exit fringe field; see entrance  
2 cm  
2 E

$NCS, C_0 - C_5$

0-6  
6* no dim.  
I, 6 E

$XPAS$

Integration step  
cm  
E

$KPOS, XCE, YCE, ALE$

$KPOS=1$: element aligned; 2: misaligned;  
1-2, 2 cm, rad  
1, 3 E

---

**Sextupole magnet**
<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

<table>
<thead>
<tr>
<th>Description</th>
<th>Values</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>$XL, R_0, B_0$</strong></td>
<td>Length : radius : asymptotic field ($=\mu_0NI/XL$)</td>
</tr>
<tr>
<td><strong>$X_E, X_S$</strong></td>
<td>Entrance and exit integration zones</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>

![Diagram of a solenoid](image-url)
Keywords and input data formatting

SPNPRNL  Store spin coordinates in file FNAME

FNAME 1 Name of storage file (e.g., zgoubi.spn) A80

SPNSTORE  Store spin coordinates every IP other pass

FNAME 1 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)]. [, 10*A10]

IP Store every IP other pass (when using REBELOTE with NPASS ≥ IP − 1). I

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**

Spin tracking

<table>
<thead>
<tr>
<th>KSO</th>
<th>Various options for initial conditions</th>
<th>1-5</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>If KSO = 1 – 3</td>
<td>KSO = 1 (respectively 2, 3) : all particles have their spin automatically set to (1,0,0) – longitudinal [respectively (0,1,0) – horizontal and (0,0,1) – vertical]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If KSO = 4</td>
<td>Repeat IMAX times (corresponding to the IMAX particles in 'OBJET') the following sequence :</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(S_X, S_Y, S_Z)</td>
<td>X, Y and Z initial components of the initial spin.</td>
<td>3*no dim.</td>
<td>3*E</td>
</tr>
<tr>
<td>If KSO = 4.1</td>
<td>X, Y and Z components of the initial spins. These will be assigned to all particles.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If KSO = 5</td>
<td>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)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(IR)</td>
<td>Random sequence seed (\leq 10^6)</td>
<td></td>
<td>I</td>
</tr>
</tbody>
</table>

\(^1\) SPNTRK must be preceded by PARTICUL for the definition of \(G\) and mass.
### SRO LOSS

**SRLOSS**

<table>
<thead>
<tr>
<th>Keywords and input data formatting</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SRLOSS</strong></td>
</tr>
</tbody>
</table>

**KSR[i]**

Switch: \( i = 1 \) causes info output into zgoubi.SRLOSS.out

\[ 0 \leq 1 \leq 2^*1 \]

**STR1, STR2**

Options:
- `STR1 = 'ALL'` or `all` or a particular KEYWORD
- `STR2 = 'scale'` will scale fields with energy loss

**Option, seed**

- 1: loss entails dp only
- 2: loss entails dp and kick angle

\[ 1 \leq 3, , > 10^5 \leq 1 \]
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>I</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**

- **$D_1, D_2, D_3$** Dummies $3^*E$

**If KSR = 1**

- **$X_0, Y_0, Z_0$** 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 $2^*eV, no\ dim.$ $2^*E, I$
Keywords and input data formatting

**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  
MOD ≤ 19 : Cartesian mesh  
MOD ≥ 20 : cylindrical mesh.  
MOD2, 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).  
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 toscap.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  
2*no dim. [idem] [.3*E,etc.]

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

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’):

```
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,’.
<table>
<thead>
<tr>
<th><strong>TRANSMAT</strong></th>
<th><strong>Matrix transfer</strong></th>
</tr>
</thead>
</table>

**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  
6*E each

**IF IORDRE = 2**  
Following records *only if* IORDRE = 2

$T(IA, IB, IC)$,  
Second order matrix, six 6*6 blocks  
m, rad  
36 lines  
6*E each
<table>
<thead>
<tr>
<th>TRAROT</th>
<th>Translation-Rotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$TX, TY, TZ, RX, RY, RZ$</td>
<td>Translations, rotations</td>
</tr>
</tbody>
</table>
**TWISS**

**Calculation of periodic optical parameters**

*KTW, FacD, FacA*

\[ KTW = 0/1/2/3 : \text{Off / as MATRIX} / \text{computation of chromaticities / computation of anharmonicities.} \]

\[ FacD \times D = \delta p/p \text{ value applied, with } D \text{ the momentum sampling in OBJET ; } FacA : \text{ unused.} \]
UNDULATOR  Undulator magnet

Under development, to be documented
**UNIPOT**  

**Unipotential electrostatic lens**

\[ IL = 1, 2 : \text{print field and coordinates along trajectories} \quad \text{0-2}[\times10^6] \]

\[ X_1, D, X_2, X_3, R_0 \quad \text{Length of first tube ; distance between tubes ; length of second and third tubes ; radius} \quad 5^m \quad 5^E \]

\[ V_1, V_2 \quad \text{Potentials} \quad 2^V \quad 2^E \]

\[ XPAS \quad \text{Integration step} \quad \text{cm} \quad \text{E} \]

\[ KPOS, XCE, YCE, ALE \quad \text{KPOS=1 : element aligned, 2 : misaligned ; shifts, tilt (unused if KPOS=1)} \quad 1-2, 2^\text{cm, rad} \quad 1, 3^E \]
VENUS Simulation of a rectangular dipole magnet

*IL* \( I_L = 1, 2 \times 10^9 \): print field and coordinates along trajectories. 0-2\( \times 10^9 \) I

*XL, YL, B_0* Length ; width = \( \pm Y_L \) ; field 2*cm, kG 3*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)

Scheme of VENUS rectangular dipole.
**WIENFILT**

**Wien filter**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Default Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( IL )</td>
<td>( IL = 1, 2 \times 10^6 ) : print field and coordinates along trajectories.</td>
<td>0-2 ( \times 10^6 )</td>
<td>I</td>
</tr>
<tr>
<td>( XL, E, B, HV )</td>
<td>Length : electric field ; magnetic field ; option : element inactive ( (HV = 0) ) horizontal ( (HV = 1) ) or vertical ( (HV = 2) ) separation</td>
<td>m, V/m, T, 0-2</td>
<td>3*E, I</td>
</tr>
<tr>
<td>( X_E, \lambda_{E_E}, \lambda_{B_E} )</td>
<td>Integration zone extent : fringe field extents, ( E ) and ( B ) respectively (( \approx ) gap height)</td>
<td>3*cm</td>
<td>3*E</td>
</tr>
<tr>
<td>( C_{E_0} - C_{E_5} )</td>
<td>Fringe field coefficients for ( E )</td>
<td>6*no dim.</td>
<td>6*E</td>
</tr>
<tr>
<td>( C_{B_0} - C_{B_5} )</td>
<td>Fringe field coefficients for ( B )</td>
<td>6*no dim.</td>
<td>6*E</td>
</tr>
<tr>
<td>( X_S, \lambda_{E_S}, \lambda_{B_S} )</td>
<td>See entrance face</td>
<td>3*cm</td>
<td>3*E</td>
</tr>
<tr>
<td>( C_{E_0} - C_{E_5} )</td>
<td>Fringe field coefficients for ( E )</td>
<td>6*no dim.</td>
<td>6*E</td>
</tr>
<tr>
<td>( C_{B_0} - C_{B_5} )</td>
<td>Fringe field coefficients for ( B )</td>
<td>6*no dim.</td>
<td>6*E</td>
</tr>
<tr>
<td>( XPAS )</td>
<td>Integration step</td>
<td>cm</td>
<td>E</td>
</tr>
<tr>
<td>( KPOS, XCE, YCE, ALE )</td>
<td>( KPOS = 1 ) : element aligned, 2 : misaligned ; shifts, tilt (unused if ( KPOS = 1 ))</td>
<td>1-2, 2*cm, rad</td>
<td>I, 3*E</td>
</tr>
</tbody>
</table>

---

1 Use **PARTICUL** to declare mass and charge.
YMY

Reverse signs of $Y$ and $Z$ axes

Equivalent to a $180^\circ$ rotation with respect to $X$-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 $\pm 50$ mrd horizontally and $\pm 50$ mrd vertically; its momentum acceptance is $\pm 40\%$. The bending magnet is simulated with DIPOLE. The layout of SPES 3 is given in Fig. 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 $R_{12}=16.6$ and $R_{34}=-.88$ 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 QOD SPECTROMETER, USING FIELD MAPS; MONTE-CARLO OBJECT WITH MOMENTUM GRID.**

**REFERENCE RIGIDITY.**

**DISTRIBUTION IN GRID.**

**NUMBER OF PARTICLES.**

**CENTRAL VALUES OF BARS.**

**NUMBER OF BARS IN MOMENTUM.**

**SPACE BETWEEN MOMENTUM BARS.**

**NUMBER OF BARS.**

**SORTING CUTOFFS (UNUSED) FOR P(D) (UNUSED)**

**SEEDS.**

**HISTO 1**

**REFERENCE RIGIDITY.**

**DISTRIBUTION IN GRID.**

**NUMBER OF PARTICLES.**

**CENTRAL VALUES OF BARS.**

**NUMBER OF BARS IN MOMENTUM.**

**SPACE BETWEEN MOMENTUM BARS.**

**NUMBER OF BARS.**

**SORTING CUTOFFS (UNUSED) FOR P(D) (UNUSED)**

**SEEDS.**

**HISTO 2**

**HISTO OF D.**

**HISTO 3**

**HISTO OF THETA.**

**HISTO 4**

**HISTO OF PHI.**

**DRIFT 5**

**CARTEMES QUADRUPOLE MAP.**

**0 0 IC IL.**

**BNORM, XNorm, YNorm.**

**CONCORDE ++++**

**IX Y.**

**FIELD MAP FILE NAME, QUADRUPOLE**

**NO LIMIT PLANE.**

**IORDRE.**

**XPAS.**

**KPOS.**

**DRIFT 6**

**CHANGREF POSITIONING OF THE**

**EXIT FRAME.**

**FIELD MAP FILE NAME, FIRST DIPOLE**

**NO LIMIT PLANE.**

**DRIFT 7**

**CHANGREF POSITIONING OF THE**

**EXIT FRAME.**

**FIELD MAP FILE NAME, SECOND DIPOLE**

**NO LIMIT PLANE.**

**HISTO 8**

**HISTO OF Y : SHOWS THE RESOLUTION**

**OF THE SPECTROMETER.**

---

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

---

**HISTO 2**

**HISTO DE LA COORDONNEE D PRINCIPALE DANS LA FENETRE : 0.9970 / 1.003**

**TOTAL COMPTAGE : 10000 SUR 10000**

**NUMERO DU CANAL MOYEN : 51**

**COMPTAGE AU " " : 2038**

**VAL. PHYS. AU " " : 1.0000**

**RESOLUTION PAR CANAL : 7.500E-0**

**PARAMETRES PHYSIQUES DE LA DISTRIBUTION :**

**COMPTAGE = 10000 PARTICULES**

**MIN = 0.9980 , MAX = 1.002 , MAX-MIN = 4.0000E-03**

**MOYENNE = 1.000**

**SIGMA = 1.4108E-03**

**TRAJ 1 IEX,D,Y,T,Z,P,S,**

---

**HISTO 3**

**HISTO DE LA COORDONNEE THETA PRINCIPALE DANS LA FENETRE : -60.00 / 60.00 (MRD)**

**TOTAL COMPTAGE : 10000 SUR 10000**

**NUMERO DU CANAL MOYEN : 51**

**COMPTAGE AU " " : 128**

**VAL. PHYS. AU " " : 3.331E-15 (MRD)**

**RESOLUTION PAR CANAL : 1.50 (MRD)**

**PARAMETRES PHYSIQUES DE LA DISTRIBUTION :**

**COMPTAGE = 10000 PARTICULES**

**MIN = -49.99 , MAX = 50.00 , MAX-MIN = 99.98 (MRD)**

**MOYENNE = 0.3320 (MRD)**

**SIGMA = 29.04 (MRD)**

**TRAJ 1 IEX,D,Y,T,Z,P,S,**
HISTOGRAMME DE LA COORDONNEE PHI
PARTICULES PRIMAIRES ET SECONDAIRES
DANS LA FENETRE : -60.00 / 60.00 (MRD)
NORMALISE

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

PARAMETRES PHYSIQUES DE LA DISTRIBUTION :
COMPTAGE = 10000 PARTICULES
MIN = -50.00 , MAX = 49.99 , MAX-MIN = 99.99 (MRD)
MOYENNE = 0.2838 (MRD)
SIGMA = 28.75 (MRD)

TRAJ 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.
800 MeV/c kaon beam line at BNL Alternating Gradient Synchrotron [54]. The line includes two separation Wien filters.

Figure 48: 800 MeV/c kaon beam line layout.
zgoubi data file.

800 HW/KE KAM BEAM LINE. CALCULATION OF TRANSFER COEFFICIENTS.

1. "COLLIMA" 2. FIRST VERTICAL FOCUS, 22
2. 14.6 -156.1 0.0.
3. "DRIFT" 23
4. 0.0.
5. "QUADRUPO" Q5 24
6. 0.0.
7. "MULTIPOL" 25
8. 0.0.
9. "SEXTUPOLE, COM pensation of 26
10. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0.0. 0.0.0. 0.0.
11. 4.0.2490 5.3630 -2.4100 0.9870 0.0. 0.0. 0.0. 0.0. 0.0.0. 0.0.
12. 0.0.
13. "OCTUPOLE" 27
14. 0.0.
15. "DRIFT" 28
16. 0.0.
17. "QUADRUPOLE" Q7 29
18. 0.0.
19. "MULTIPOL" 30
20. 0.0.
21. "SEXTUPOLE" 31
22. 0.0.
23. "OCTUPOLE" 32
24. 0.0.
25. "DRIFT" 33
26. 0.0.
27. "QUADRUPOLE" Q8 34
28. 0.0.
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>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.60453</td>
<td>-4.431245E-02</td>
</tr>
<tr>
<td>-2.05368</td>
<td>-4.700735</td>
</tr>
<tr>
<td>2.249356E-05</td>
<td>-8.477757E-07</td>
</tr>
<tr>
<td>1.183590E-05</td>
<td>-4.256398E-07</td>
</tr>
<tr>
<td>-0.387577</td>
<td>2.319538E-02</td>
</tr>
<tr>
<td>0.00000</td>
<td>0.00000</td>
</tr>
</tbody>
</table>

**SECOND ORDER COEFFICIENTS (MKSA):**

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.147E-02</td>
<td>3.501E-02</td>
</tr>
<tr>
<td>6.87E-03</td>
<td>1.29E-03</td>
</tr>
<tr>
<td>2.11E-04</td>
<td>2.41E-04</td>
</tr>
<tr>
<td>5.70E-05</td>
<td>7.10E-05</td>
</tr>
<tr>
<td>1.00E00</td>
<td>1.00E00</td>
</tr>
</tbody>
</table>

**HIGHER ORDER COEFFICIENTS (MKSA):**

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y/Y3</td>
<td>584.8</td>
</tr>
<tr>
<td>Y/T3</td>
<td>8.40277E-05</td>
</tr>
<tr>
<td>Y/Z3</td>
<td>0.70473</td>
</tr>
<tr>
<td>Y/P3</td>
<td>0.42144</td>
</tr>
<tr>
<td>T/Y3</td>
<td>-1.9407</td>
</tr>
<tr>
<td>T/T3</td>
<td>1.04027E-05</td>
</tr>
<tr>
<td>T/Z3</td>
<td>-1.10234</td>
</tr>
<tr>
<td>T/P3</td>
<td>5.25793E-02</td>
</tr>
<tr>
<td>Z/Y3</td>
<td>32.161</td>
</tr>
<tr>
<td>Z/T3</td>
<td>10.425</td>
</tr>
<tr>
<td>Z/Z3</td>
<td>-872.50</td>
</tr>
<tr>
<td>Z/P3</td>
<td>-785.20</td>
</tr>
<tr>
<td>P/Y3</td>
<td>15.460</td>
</tr>
<tr>
<td>P/T3</td>
<td>7.5244</td>
</tr>
<tr>
<td>P/Z3</td>
<td>-409.99</td>
</tr>
<tr>
<td>P/P3</td>
<td>-389.15</td>
</tr>
</tbody>
</table>

DetY-1 = -0.1170246601, DetZ-1 = 0.0000034613

R12 = 0 at 0.1647 m, R34 = 0 at -0.6034E-01 m
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
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 .5e-3 .5e-2 .5e-3 0. 0.4 WIDTH OF BARS.
1 1 1 1 1 1 CUT-OFFS (UNUSED)
9 9. 9. 9. 9. UNUSED.
186387 548728 472874 SEEDS.
'PARTICUL' 2
139.6000 0. 0. 26.03E-9 0. PION MASS AND LIFE TIME
'MCDESINT' 3
105.66 0. PION -> MUON + NEUTRINODECAY
136928 768370 548375
'ESL' 4
77.3627 STOPS ABBERRANT MUONS.
1 1 100. 10. 245. 0.
'DIPOLE' 6
2 80. 208.5 TOTAL ANGLE AT, CENTRAL RADIUS RM
33. 30. 0. 0. CENTRAL ANGLE, FIELD, INDICES
46. -1. ENTRANCE EFB
14552 5.21405 -3.38307 14.0629 0. 0. 0.
15. 0. -65. 0. 0. -65.
46. -1. EXIT EFB
14552 5.21405 -3.38307 14.0629 0. 0. 0.
-15. 69. 85. 0. 1.6E 1.6E
0. 0. LATERAL EFB (INHIBITED)
14552 5.21405 -3.38307 14.0629 0. 0. 0.
-15. 69. 85. 0. 1.6E 1.6E
2 10.0 2ND DEGREE INTERP., MESH 0.4 CM
4. STEP SIZE
2 164.755 .479966 233.554 -.057963
'CHAMBR' 7
1 100. 10. 245. 0.
'CHANGREF' 8 TILT ANGLE OF FOCAL PLANE.
0. 0. -49.
'HISTO' TOTAL SPECTRUM (PION + MUON).
2 -170. 130. 60 1
20 'V' 1 'Q'
'HISTO' PION SPATIAL SPECTRUM
2 -170. 130. 60 2
20 'P' 1 'P'
'HISTO' MUON SPATIAL SPECTRUM
2 -170. 130. 60 3
20 'y' 1 'S'
'HISTO' MUON MOMENTUM SPECTRUM
1 .2 1.7 60 3
20 'd' 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**
PARTICULES PRIMAIRES
DANS LA FENETRE : -1.7000E+02 / 1.3000E+02 (CM)
NORMALISE

TOTAL COMPTAGE : 9282 SUR 10000
HISTOGRAMME DE LA COORDONNEE Y
PARTICULES SECONDAIRES
DANS LA FENETRE : -1.7000E+02 / 1.3000E+02 (CM)
NORMALISE

TOTAL COMPTAGE : 9282 SUR 10000

PARAMETRES PHYSIQUES DE LA DISTRIBUTION :
COMPTAGE = 9282 PARTICULES
MIN = -6.5836E+01, MAX = -4.3504E+01, MAX-MIN = 2.6120E+02 (CM)
MOYENNE = -4.9278E+01 (CM)
SIGMA = 5.3114E+01 (CM)

**HISTOGRAMME DE LA COORDONNEE D**
PARTICULES SECONDAIRES
DANS LA FENETRE : -1.7000E+02 / 1.3000E+02 (CM)
NORMALISE

TOTAL COMPTAGE : 9282 SUR 10000
HISTOGRAMME DE LA COORDONNEE D
PARTICULES PRIMAIRES
DANS LA FENETRE : -1.7000E+02 / 1.3000E+02 (CM)
NORMALISE

TOTAL COMPTAGE : 9282 SUR 10000

PARAMETRES PHYSIQUES DE LA DISTRIBUTION :
COMPTAGE = 9282 PARTICULES
MIN = -6.5836E+01, MAX = -4.3504E+01, MAX-MIN = 2.6120E+02 (CM)
MOYENNE = -4.9278E+01 (CM)
SIGMA = 5.3114E+01 (CM)
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.

Zgoubi data file.

MATCHING A SYMMETRIC QUADRUPOLE TRIPLET

`OBJET` 1
2501.73 RIGIDITY (kG.cm)
11 PARTICLES GENERATED FOR USE OF MATRIX
2. 2. 2. 0. .001
0. 0. 0. 0. 1.
`ESL` 2
200.
`QUADRUPO` 3
0
40. 15. -7.
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
5.
1 0. 0. 0.
`ESL` 4
30.
`QUADRUPO` 5
0
40. 15. 3.
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
5.
1 0. 0. 0.
`ESL` 6
30.
`QUADRUPO` 7
0
40. 15. -7.
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
5.
1 0. 0. 0.
`ESL` 8
200.
`MATRIX` 9
1 VARY B IN QUADS FOR FIT OF R12 AND R34
2 # OF VARIABLES
3 12 7.12 2.
5 12 0. 2.
PMTR #12 OF ELEMENTS #3, 5 AND 7 IS FIELD VALUE
2 1.5-10 # OF CONSTRAINTS, PENALTY
1 1 2 8 16.6 1.
CNSTRNT #1 : R12=16.6 AFTER LAST DRIFT (LIMIT #8)
1 3 4 8 -.8 1.
CNSTRNT #2 : R34=-.88 AFTER LAST DRIFT
.END`
Excerpt of zgoubi.res: first order transfer matrices prior to and after fitting.

TRANSFER MATRIX WITH STARTING CONDITIONS:

```
TRANSFER MATRIX WITH STARTING CONDITIONS:

MATRICE DE TRANSFERT ORDRE 1 ( MKSA )

5.43427  17.0254  0.00000  0.00000  0.00000  0.00000
1.67580  5.43425  0.00000  0.00000  0.00000  0.00000
0.00000  0.00000  -1.27003  -0.974288  0.00000  0.00000
0.00000  0.00000  -0.629171  -1.27004  0.00000  0.00000
0.00000  0.00000  0.00000  0.00000  1.00000  0.00000
0.00000  0.00000  0.00000  0.00000  0.00000  1.00000
```

STATE OF VARIABLES AFTER MATCHING:

```
STATE OF VARIABLES AFTER MATCHING:

LMNT VAR PARAM MINIMUM INITIAL FINAL MAXIMUM STEP NAME LBL1 L BL2
3  1  12 -21.0 -7.00 -6.972765137 7.00 1.707E-04 QUADRUPO 3 *
7  1  120 -6.97 -7.00 -6.972765137 7.00 1.707E-04
5  2  12 -3.00 3.00 3.229344585 9.00 1.266E-04 QUADRUPO 5 *
STATUS OF CONSTRAINTS (Target penalty = 1.0000E-10)

TYPE I J LMNT# DESIRED WEIGHT REACHED KI2 * Parameter(s)
1  1  2  8 1.6600000E+01 1.0000E+00 1.6599981E+01 * 0 :
1  3  4  8 -8.8000000E-01 1.0000E+00 -8.8000964E-01 * 0 :
Fit reached penalty value 8.4374E-11
```

MATRICE, WITH NEW VARIABLES:

```
MATRICE, WITH NEW VARIABLES:

5.27056  16.6000  0.00000  0.00000  0.00000  0.00000
1.61443  5.27450  0.00000  0.00000  0.00000  0.00000
0.00000  5.00000  -1.24205  -1.24620  0.00000  0.00000
0.00000  0.00000  -0.622553  -1.24620  0.00000  0.00000
0.00000  0.00000  0.00000  0.00000  1.00000  0.00000
0.00000  0.00000  0.00000  0.00000  0.00000  1.00000
```
5 MULTITURN SPIN TRACKING IN SATURNE 3 GeV SYNCHROTRON

Figure 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 $\gamma_Z Z^2 + 2 \alpha_Z ZZ' + \beta_Z Z'^2 = \varepsilon_Z / \pi$ 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 $\varepsilon_Y / \pi = \varepsilon_Z / \pi \simeq 12 \times 10^{-6}$ m.rad: $\nu_Y = 0.63795, \nu_Z = 0.60912$ (the integer part is 3 for both).

(D) Longitudinal phase-space ("$DP, phase$" in Zgoubi notations): particles with initial momentum dispersion of $5 \times 10^{-4}$ (1), $10^{-3}$ (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 $10^{-3}$. 
283

zgoubi data file (beginning and end).
SATURNE. CROSSING GammaG=7-NUz, NUz=3.60877(perturbed)
’OBJET’
5015.388
834.04 MeV, proton
2
4 1
6.2E-02
6.5E-02
.458 0. 0. 1.00
’o’
EpsilonY/pi ˜ 0. (Closed orbit)
0.356
0.379
.458 0. 0. 1.0005 ’1’
0.647
0.689
.458 0. 0. 1.001
’2’
1.024
1.09
.458 0. 0. 1.0016 ’3’
1 1 1 1
’SCALING’
1 4
MULTIPOL
2
CROSSING GammaG=7-Nuz+/-14E, E=3.3E-4
5015.388E-3 5034.391E-3
AT 2.1 T/s, IN 3442 MACHINE TURNS,
1
3442
FROM 834.041 TO 838.877 MeV
QUADRUPO
2
5015.388E-3 5034.391E-3
1
3442
BEND
2
5015.388E-3 5034.391E-3
1
3442
CAVITE
2
1.
1.00378894
RELATIVE CHANGE OF SYNCHRONOUS RIGIDITY
1
3442
’PARTICUL’
938.2723 1.6021892E-19 1.7928474 0. 0.
’SPNTRK’
3
’QUADRUPO’
QP 1
5
0
46.723 10. .763695
.763695 = FIELD FOR BORO=1 T.m
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
#30|50|30 Quad
1 0. 0. 0.
’ESL’
SD 2
6
71.6256
’BEND’
DIP 3 4 3
7
0
247.30039 0. 1.57776
20. 8. .04276056667
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
20. 8. .04276056667
20. 8.
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
#30|120|30
bend
3 0. 0. 0. -.1963495408
’ESL’
SD 2
8
71.6256
’MULTIPOL’
QP 5
9
0
48.6273 10. 0. -.765533 0. 0. 0. 0.
0. 0. 0. 0.
0. 0. 0. 0. 0. .0 .0 0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0.
0. 0. 0. 0.
#30|50|30 Quad
1 0. 0. 0.
’ESL’
SD 2
10
71.6256
’BEND’
DIP 3 4 3
11
0
247.30039 0. 1.57776
20. 8. .04276056667
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
20. 8. .04276056667
20. 8.
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
#30|120|30
bend
3 0. 0. 0. -.1963495408
’ESL’
SD 2
12
71.6256
’QUADRUPO’
QP 1
13
0
46.723 10. .763695
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
#30|50|30 Quad
1 0. 0. 0.
’ESL’
SD 2
14
71.6256
’BEND’
DIP 3 4 3
15
0
247.30039 0. 1.57776
20. 8. .04276056667
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
20. 8. .04276056667
20. 8.
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
#30|120|30
bend
3 0. 0. 0. -.1963495408
’ESL’
SD 2
16
71.6256
’MULTIPOL’
QP 5
17
0
48.6273 10. 0. -.765533 0. 0. 0. 0.
0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0.
0. 0. 0. 0.
#30|50|30 Quad
1 0. 0. 0.

’ESL’
SD 2
71.6256
’BEND’
DIP 3 4 3
0
247.30039 0. 1.57776
20. 8. .04276056667
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
20. 8. .04276056667
20. 8.
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
#30|120|30
bend
3 0. 0. 0. -.1963495408
’ESL’
SD 2
71.6256
’QUADRUPO’
QP 1
0
46.723 10. .763695
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
#30|50|30 Quad
1 0. 0. 0.
’ESL’
392.148
’MULTIPOL’
QP 5
0
48.6273 10. 0. -.765533 0. 0. 0. 0.
0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0.
0. 0. 0. 0.
#30|50|30 Quad
1 0. 0. 0.
’ESL’
392.148
’QUADRUPO’
QP 1
0
46.723 10. .763695
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
#30|50|30 Quad
1 0. 0. 0.
’ESL’
SD 2
71.6256
’BEND’
DIP 3 4 3
0
247.30039 0. 1.57776
20. 8. .04276056667
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
20. 8. .04276056667
20. 8.
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
#30|120|30
bend
3 0. 0. 0. -.1963495408
’ESL’
SD 2
71.6256
’MULTIPOL’
QP 5
0
48.6273 10. 0. -.765533 0. 0. 0. 0.
0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0.
6 .1122 6.2671 -1.4982 3.5882 -2.1209 1.723
0. 0. 0. 0. 0. 0.
0. 0. 0. 0.
#30|50|30 Quad
1 0. 0. 0.
’ESL’
SD 2
71.6256
’BEND’
DIP 3 4 3
0
247.30039 0. 1.57776
20. 8. .04276056667
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
20. 8. .04276056667
20. 8.
4 .2401 1.8639 -.5572 .3904 0. 0. 0.
#30|120|30
bend
3 0. 0. 0. -.1963495408

18
19

20
21

22
23

24
25

26
27

28
29

30
31

:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
’ESL’
392.148
’CAVITE’
1
105.5556848673
6000.
0.
’FAISCNL’
b_zgoubi.fai
’SPNPRNL’
zgoubi.spn
’SPNPRT’
’REBELOTE’
2999 0.1 99
’END’

84
85
3.
SIN(phis) = .234162, dE=1.40497 keV/Turn.
86
87
88
90
TOTAL NUMBER OF TURNS = 3000
91


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 DOUBLET.

**MCOBJET**

**RANDOM OBJECT DEFINITION**

1

**DISTRI**

2

**UTION IN WINDOW.**

200

**NUMBER OF PARTICLES.**

1 **UNIFORM DISTRIBUTION.**

0. 0. 0. 0. 0. 0.

1 **HALF WIDTH OF DISTRIBUTION.**

0.0003

10. 10. 10. 10. 10. 10.

8 9 9. 9.

194837 549728 472874

**SEED**

1

**PARTICUL**

**PARTICLE MASS AND CHARGE**

1

**DISTRIBUTION IN WINDOW.**

200

**NUMBER OF PARTICLES.**

1 **UNIFORM DISTRIBUTION.**

0. 0. 0. 0. 0. 0.

1 **HALF WIDTH OF DISTRIBUTION.**

0.0003

10. 10. 10. 10. 10. 10.

8 9 9. 9.

194837 549728 472874

**SEED**

1

**PARTICUL**

**PARTICLE MASS AND CHARGE**

2

**DRIFT**

3

**DRIFT**

4

**DRIFT**

5

**DRIFT**

6

**DRIFT**

7

**DRIFT**

8

**HISTO**

**HISTOGRAM OF THE Y COORDINATE.**

20 **Y** 1 **Q**

**HISTO**

**HISTOGRAM OF THE Z COORDINATE.**

20 **Z** 1 **Q**

**FAISCNL**

**RAYS ARE STORED IN rays.out FOR FURTHER PLOTTING.**

20 **V** 1 **Q**

**REBEL**

**RUN AGAIN, FOR RAY-TRACING.**

19 0.1 0 **A TOTAL OF 200*(19+1) PARTICLES.**

**END**

285

**zgoubi data file.**

**MICROBEAM LINE, WITH A MAGNETO-ELECTROSTATIC QUADRUPOLE DOUBLET.**

**MCOBJET**

**RANDOM OBJECT DEFINITION**

1

20.435 **RIGIDITY (20keV PROTONS).**

1 **DISTRIBUTION IN WINDOW.**

200 **NUMBER OF PARTICLES.**

1 **UNIFORM DISTRIBUTION.**

0. 0. 0. 0. 0. 0.

1 **HALF WIDTH OF DISTRIBUTION.**

0.0003

10. 10. 10. 10. 10. 10.

8 9 9. 9.

194837 549728 472874

**SEED**

1

**PARTICUL**

**PARTICLE MASS AND CHARGE**

2

**DRIFT**

3

**DRIFT**

4

**DRIFT**

5

**DRIFT**

6

**DRIFT**

7

**DRIFT**

8

25 **HISTO**

**HISTOGRAM OF THE Y COORDINATE.**

20 **Y** 1 **Q**

**HISTO**

**HISTOGRAM OF THE Z COORDINATE.**

20 **Z** 1 **Q**

**FAISCNL**

**RAYS ARE STORED IN rays.out FOR FURTHER PLOTTING.**

20 **V** 1 **Q**

**REBEL**

**RUN AGAIN, FOR RAY-TRACING.**

19 0.1 0 **A TOTAL OF 200*(19+1) PARTICLES.**

**END**

**zgoubi.res file.**

**1 MCOBJET RANDOM OBJECT**

**Reference magnetic rigidity = 20.435 K0xM**

**Object built up of 200 particles**

**Distribution in a Window**

**Central values (MKSA units):**

**Yo, To, Zo, Po, Xo, BR/BORO : 0.000 0.000 0.000 0.000 0.000 1.000**

**Width ( +/- , MKSA units ) :**

**DY, DT, DZ, DP, DX, DBR/BORO : 0.00 2.000E-04 0.00 2.000E-04 0.00 3.000E-04**

**Cut-offs ( * +/-Width ) :**

**NY, NT, NZ, NP, NX, NBR/BORO : 1.00 1.00 1.00 1.00 1.00 1.00**

**Type of sorting :**

**Y, T, Z, P, X, D : Uniform Uniform Uniform Uniform Uniform Uniform**

**2 PARTICUL PARTICLE MASS**

**Particle properties :**

**Mass = 938.272 MeV/c2**

**Charge = 1.602177E-19 C**

**Reference data :**

**mag. rigidity (K0.cm) : 20.435000 -g/p, such that dev.=p/g/riidity**

**mass (MeV/c2) : 938.2720**

**momentum (MeV/c) : 6.126262**

**energy, total (MeV) : 936.29230**

**energy, kinetic (MeV) : 1.69596908E-02**

**beta = v/c : 6.99999999E-03**

**gamma : 1.00000000000009**

**beta* gamma : 6.99999999999999**

**electric rigidity (MeV) : 3.99999999999999**

**zgoubi.data file.**

**MICROBEAM LINE, WITH A MAGNETO-ELECTROSTATIC QUADRUPOLE DOUBLET.**

**MCOBJET**

**RANDOM OBJECT DEFINITION**

1

20.435 **RIGIDITY (20keV PROTONS).**

1 **DISTRIBUTION IN WINDOW.**

200 **NUMBER OF PARTICLES.**

1 **UNIFORM DISTRIBUTION.**

0. 0. 0. 0. 0. 0.

1 **HALF WIDTH OF DISTRIBUTION.**

0.0003

10. 10. 10. 10. 10. 10.

8 9 9. 9.

194837 549728 472874

**SEED**

1

**PARTICUL**

**PARTICLE MASS AND CHARGE**

2

**DRIFT**

3

**DRIFT**

4

**DRIFT**

5

**DRIFT**

6

**DRIFT**

7

**DRIFT**

8

**HISTO**

**HISTOGRAM OF THE Y COORDINATE.**

20 **Y** 1 **Q**

**HISTO**

**HISTOGRAM OF THE Z COORDINATE.**

20 **Z** 1 **Q**

**FAISCNL**

**RAYS ARE STORED IN rays.out FOR FURTHER PLOTTING.**

20 **V** 1 **Q**

**REBEL**

**RUN AGAIN, FOR RAY-TRACING.**

19 0.1 0 **A TOTAL OF 200*(19+1) PARTICLES.**

**END**
3 DRIFT DRIFT. 3
Drift, length = 500.00000 cm
TRAJ #1: T, D, Y, T, Z, P, S, time : 1 2.999865E-04 -2.027052E-02 -4.054104E-02 -9.359114E-02 -1.871823E-01 5.000001E+02 2.55365E+00
Cumulative length of optical axis = 5.000000 m at ; Time for ref. rigidity & particle = 2.534417E-04 s

4 DRIFT DRIFT. 4
Drift, length = 59.00000 cm
TRAJ #1: T, D, Y, T, Z, P, S, time : 1 2.999865E-04 -2.266244E-02 -4.054104E-02 -1.046349E-01 -1.871823E-01 5.5900001E+02 2.85498E+00
Cumulative length of optical axis = 5.590000 m at ; Time for ref. rigidity & particle = 2.855839E-06 s

5 EBMULT FIRST ELECTROMAGNETIC ELEMENT:
--- MULTIPOLAR:
Length of element = 10.200000 cm
Bore radius RO = 10.000 cm
E-DIPOLAR = 0.0000000E+00 V/m
E-QUADRUPOLE = -9.2729860E+03 V/m
E-SEXTUPOLE = 0.0000000E+00 V/m
E-OCTUPOLE = 0.0000000E+00 V/m
E-DECAPOLE = 0.0000000E+00 V/m
E-DODECAPOLE = 0.0000000E+00 V/m
E-14-POLE = 0.0000000E+00 V/m
E-16-POLE = 0.0000000E+00 V/m
E-18-POLE = 0.0000000E+00 V/m
E-20-POLE = 0.0000000E+00 V/m
Entrance/exit field models are sharp edge
FINTE, FINTS, gap : 0.0000E+00 0.0000E+00 5.0000E+00
--- MULTIPOLAR:
Length of element = 10.200000 cm
Bore radius RO = 10.000 cm
B-DIPOLE = 0.0000000E+00 kG
B-QUADRUPOLE = 1.8949300E+00 kG
B-SEXTUPOLE = 0.0000000E+00 kG
B-OCTUPOLE = 0.0000000E+00 kG
B-DECAPOLE = 0.0000000E+00 kG
B-DODECAPOLE = 0.0000000E+00 kG
B-14-POLE = 0.0000000E+00 kG
B-16-POLE = 0.0000000E+00 kG
B-18-POLE = 0.0000000E+00 kG
B-20-POLE = 0.0000000E+00 kG
Entrance/exit field models are sharp edge
FINTE, FINTS, gap : 0.0000E+00 0.0000E+00 5.0000E+00
*** Warning : sharp edge model, vertical wedge focusing approximated with first order kick. FINTE at entrance = 0.000
*** Warning : sharp edge model, vertical wedge focusing approximated with first order kick. FINT at exit = 0.000
Integration step : 0.8000 cm
Cumulative length of optical axis = 5.69200000 m ; Time for ref. rigidity & particle = 2.907949E-06 s

6 DRIFT DRIFT. 6
Drift, length = 4.90000 cm
TRAJ #1: T, D, Y, T, Z, P, S, time : 1 2.839285E-04 -1.310701E-02 9.564949E-01 -1.603604E-01 -5.590035E+00 5.7410014E+02 2.93210E+00
Cumulative length of optical axis = 5.74100000 m at ; Time for ref. rigidity & particle = 2.932982E-06 s

7 EBMULT SECOND ELECTROMAGNETIC ELEMENT:
--- MULTIPOLAR:
Length of element = 10.200000 cm
Bore radius RO = 10.000 cm
E-DIPOLAR = 0.0000000E+00 V/m
E-QUADRUPOLE = 1.3779900E+04 V/m
E-SEXTUPOLE = 0.0000000E+00 V/m
E-OCTUPOLE = 0.0000000E+00 V/m
E-DECAPOLE = 0.0000000E+00 V/m
E-DODECAPOLE = 0.0000000E+00 V/m
E-14-POLE = 0.0000000E+00 V/m
E-16-POLE = 0.0000000E+00 V/m
E-18-POLE = 0.0000000E+00 V/m
E-20-POLE = 0.0000000E+00 V/m
Entrance/exit field models are sharp edge
FINTE, FINTS, gap : 0.0000E+00 0.0000E+00 5.0000E+00
--- MULTIPOLAR:
Length of element = 10.200000 cm
Bore radius RO = 10.000 cm
B-DIPOLE = 0.0000000E+00 kG
B-QUADRUPOLE = -2.8159200E+00 kG
B-SEXTUPOLE = 0.0000000E+00 kG
B-OCTUPOLE = 0.0000000E+00 kG
B-DECAPOLE = 0.0000000E+00 kG
B-DODECAPOLE = 0.0000000E+00 kG
B-14-POLE = 0.0000000E+00 kG
B-16-POLE = 0.0000000E+00 kG
B-18-POLE = 0.0000000E+00 kG
B-20-POLE = 0.0000000E+00 kG
Entrance/exit field models are sharp edge
FINTE, FINTS, gap : 0.0000E+00 0.0000E+00 5.0000E+00
*** Warning : sharp edge model, vertical wedge focusing approximated with first order kick. FINTE at entrance = 0.000
*** Warning : sharp edge model, vertical wedge focusing approximated with first order kick. FINT at exit = 0.000
Integration step : 0.8000 cm
Cumulative length of optical axis = 5.84300000 m ; Time for ref. rigidity & particle = 2.985092E-06 s
HISTOGRAMME DE LA COORDONNÉE Y
PARTICULES PRIMAIRES ET SECONDAIRES
DANS LA FENETRE : -5.0000E-06 / 5.0000E-06 (CM)
NORMALISE

TOTAL COMPTAGE : 4000 SUR 4000
NUMERO DU CANAL MOYEN : 51
COMPTAGE AU " " : 109
VAL. PHYS. AU " " : 0.000E+00 (CM)
RESOLUTION PAR CANAL : 1.667E-07 (CM)
PARAMETRES PHYSIQUES DE LA DISTRIBUTION :
COMPTAGE = 4000 PARTICULES
MIN = -3.4326E-06, MAX = 3.4347E-06, MAX-MIN = 6.8674E-06 (CM)
MOYENNE = -2.8531E-08 (CM)
SIGMA = 1.8619E-06 (CM)

TRAJ #1 D,Y,T,P,S,IEX : 1.0002E+00 9.0257E-07 -2.3996E-01 -1.0770E-06 1.7947E+00 6.09300E+02

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

TOTAL COMPTAGE : 4000 SUR 4000
NUMERO DU CANAL MOYEN : 51
COMPTAGE AU " " : 169
VAL. PHYS. AU " " : 0.000E+00 (CM)
RESOLUTION PAR CANAL : 1.667E-07 (CM)
PARAMETRES PHYSIQUES DE LA DISTRIBUTION :
COMPTAGE = 4000 PARTICULES
MIN = -1.9150E-06, MAX = 1.9110E-06, MAX-MIN = 3.8260E-06 (CM)
MOYENNE = -3.8539E-09 (CM)
SIGMA = 1.1232E-06 (CM)

TRAJ #1 D,Y,T,P,S,IEX : 1.0002E+00 9.0257E-07 -2.3996E-01 -1.0770E-06 1.7947E+00 6.09300E+02

REBELOTE RUN AGAIN
**** FIN D'EFFET DE 'REBELOTE' ****
IL Y À EU 20 PASSAGES DANS LA STRUCTURE
# PARTICULES ENVOYEES : 4000

Pgm principal : arrêt sur clé REBELOTE
PART D

Running zgoubi and its post-processor/graphic interface zpop
INTRODUCTION

The *zgoubi* package, including this guide, examples, the *zpop* graphic/analysis post-processor, is available on web [5].

The *zgoubi* *FORTRAN* package is transportable; it has been compiled, linked and executed over the years on several types of systems (e.g. CDC, CRAY, IBM, DEC, HP, SUN, VAX, UNIX, LINUX).

An additional *FORTRAN* code, *zpop*, allows the post-processing and graphic treatment of *zgoubi* output files. *zpop* has been routinely used on DEC, HP, SUN stations, and is now on UNIX and LINUX systems.

1 GETTING TO RUN zgoubi AND zpop

1.1 Making the Executable Files zgoubi and zpop

1.1.1 The transportable package zgoubi

Compile and link the *FORTRAN* source files (normally, just run the Makefile), to create the executable *zgoubi*.

*zgoubi* is written in standard *FORTRAN*, mostly 77, therefore it does not require linking to any library.

1.1.2 The post-processor and graphic interface package zpop

Compile the *FORTRAN* source files (normally, just run the Makefile).

Link *zpop* with the graphic library, libminigraf.a [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 `zgoubi.fai`) with particle and spin coordinates and a lot of other informations.
- **Keywords SPNPRNL, SPNSTORE**: fill a `.spn` type file (normally named `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

[36] Files developed by Simon White, January 2012. Including beam-beam spin kick after Ref. [37].


PICKUPS, 156, 162, 236
PLOTDATA, 157, 237
POISSON, 128, 238
POLARMS, 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, 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
SEXTUPOL, 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
SRPRINT, 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
outpoi.lis, 128
negative, 66
XPAS, coded, 166
XPAS, negative, 161
YCE, 163
YMY, 146, 265
zgoubi, 291
zgoubi.dat, 149, 158, 291
zgoubi.f, 291
zgoubi.fai, 148, 149, 158, 209, 229, 292
zgoubi.map, 162, 292
zgoubi.plt, 125, 162, 222, 292
zgoubi.res, 162, 291
zgoubi.spl, 158, 252, 292
zgoubi.src, 77
zpop, 39, 148, 158, 162, 291
XCE, 163
XPAS