On the implementation of a 3D space charge algorithm to understand and further study the physics of linear accelerators

Emmanuel Branlard

August 2009

Fermilab supervisor: Helen Edwards & Philippe Piot
Supaero supervisor: Jean-Claude Mollier
Contents

Introduction 2
Notations 4

I A quick introduction to beam physics 6

1 Quick memo on electromagnetism and special relativity 8
   1.1 Basic notions of Electromagnetism and charged particles 8
   1.2 Basic notions of relativity 9

2 Beam dynamics - linear formalism 11
   2.1 Introduction 11
   2.2 A particular case: Hill’s equation - Betatron functions 12
   2.3 General resolution of the perturbated linear equation of motion 13

3 Phase space and matrix formalism 16
   3.1 Transportation matrix 16
   3.2 Beam matrix and phase space 17

II Implementation of the space charge algorithm 21

4 On the implementation of a 3D algorithm in Astra 23
   4.1 Wider prospects for the algorithm 23
   4.2 General description of the method 24
   4.3 Detailed description of the method 25

5 Space Charge calculation 28
   5.1 Poisson equation 28
   5.2 Resolution of poisson screened equation with the use of Fourier transform 29
   5.3 Resolution of poisson screened equation with the use of Green function 29

III On the benchmarking of the space charge algorithm 31

6 Benchmarking of the stand alone space charge algorithm without rotation 33
   6.1 Fields for a cylindrical distribution 33
   6.2 Fields for an ellipsoidal distribution 36
# Table of Contents

16 **ILCT A at New Muon lab (NML)**  
16.1 About ILCT A ........................................... 97  
16.2 Specifications ........................................... 97  
16.3 Coordinates of the principal elements ............ 98  
16.4 Compressor lattice coordinates ....................... 98  
16.5 Simulation ........................................... 99  

**Conclusion** ........................................... 100  

**Acknowledgements** .................................... 101  

**Annexes** ............................................ 103  

A **Documentation of R scripts**  

B **Jacobi diagonalization of a symmetric matrix**  
B.1 Mathematical description of the method ........... 107  
B.2 Fortran source code ................................ 109  

C **The use of Green function for solving differential equations**  

D **A0 line full lattice geometry and coordinates**  

E **More examples of the rotation algorithm benchmarking**  
E.1 Example of an ellipsoid at rest .................... 119  
E.2 Example of a sphere at low energy ............... 120  

F **Useful mathematical formulas**  

G **Longitudinal space charge - Theory vs. Simulation**  
G.1 Cylindrical distribution .............................. 125  
G.2 Ellipsoidal distribution ............................... 127  

H **Astra input files for the two accelerator lines**  
H.1 A0 photo injector ...................................... 128  
H.2 ILCT A at NML ........................................ 132  

I **Automatizing and organizing the simulations**  
I.1 On the need to get organized ......................... 137  
I.2 Changing parameters in any Astra input file ...... 138  
I.3 Application to generate a series of input distribution  
I.4 Performing simulation and storing results in a corresponding folder  

J **Versions history of the 3d space charge algorithm**  

**Bibliography** ........................................ 145
List of Tables

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Parameters used for the benchmarking of the space charge algorithm - cylinder distribution</td>
<td>36</td>
</tr>
<tr>
<td>6.2</td>
<td>Parameters used for the benchmarking of the space charge algorithm - spheroid distribution</td>
<td>40</td>
</tr>
<tr>
<td>6.3</td>
<td>Parameters used for the benchmarking of the space charge algorithm - sphere at two energies</td>
<td>40</td>
</tr>
<tr>
<td>7.1</td>
<td>RMS dimensions used for the study of ellipsoidal distributions</td>
<td>44</td>
</tr>
<tr>
<td>7.2</td>
<td>Structure of the file /results/ez-rms/EllipseXYZ-database.csv</td>
<td>45</td>
</tr>
<tr>
<td>8.1</td>
<td>Parameters used for the benchmarking of the rotation algorithm - example of a coin-shape</td>
<td>54</td>
</tr>
<tr>
<td>8.2</td>
<td>Parameters used for the benchmarking of the rotation algorithm - example of an ellipsoid at high energy</td>
<td>55</td>
</tr>
<tr>
<td>9.1</td>
<td>Structure of the file containing the sigma matrix: run.Sigma.001</td>
<td>58</td>
</tr>
<tr>
<td>9.2</td>
<td>Parameters of the distributions used for our analysis on the sigma matrix</td>
<td>59</td>
</tr>
<tr>
<td>9.3</td>
<td>Structure of the file containing the sigma matrix: run.Sigma.001</td>
<td>61</td>
</tr>
<tr>
<td>10.1</td>
<td>Parameters of the distributions used for our analysis on the sigma matrix</td>
<td>66</td>
</tr>
<tr>
<td>11.1</td>
<td>Structure of the experimental dipoles data files</td>
<td>71</td>
</tr>
<tr>
<td>12.1</td>
<td>Dipole radii needed by astra to fit the compressor geometry</td>
<td>77</td>
</tr>
<tr>
<td>13.1</td>
<td>Dipole radii needed by astra to fit the geometry</td>
<td>83</td>
</tr>
<tr>
<td>15.1</td>
<td>Specifications of A0 beam out of the photoinjector</td>
<td>94</td>
</tr>
<tr>
<td>15.2</td>
<td>Dipoles hard edge coordinates in mm for A0 emittance exchange lattice</td>
<td>96</td>
</tr>
<tr>
<td>16.1</td>
<td>Specifications of NML beam out of the photoinjector and expected range</td>
<td>98</td>
</tr>
<tr>
<td>16.2</td>
<td>Coordinates of beam line elements at NML</td>
<td>98</td>
</tr>
<tr>
<td>16.3</td>
<td>Dipoles hard edge coordinates in mm for NML compressor</td>
<td>99</td>
</tr>
<tr>
<td>D.1</td>
<td>Coordinates of beam line elements at A0 in the emittance exchange configuration</td>
<td>116</td>
</tr>
<tr>
<td>D.2</td>
<td>Coordinates of beam line elements at A0 in the straight ahead configuration</td>
<td>117</td>
</tr>
<tr>
<td>E.1</td>
<td>Parameters used for the benchmarking of the rotation algorithm - example of an ellipsoid at rest with two rotations</td>
<td>119</td>
</tr>
<tr>
<td>E.2</td>
<td>Parameters used for the benchmarking of the rotation algorithm - example of a sphere at low energy</td>
<td>120</td>
</tr>
</tbody>
</table>
**List of Figures**

1. Convention used for the longitudinal phase space ............................................ 5
2. Legend of phase space plots ................................................................................. 5
3. Convention used to define dipole pole tip coordinates ...................................... 5

2.1 Curvilinear system of coordinates used in particle beam dynamics .................... 11
2.2 Illustration of the relative coordinates .............................................................. 12

3.1 Illustration of the matrix formalism in optics - case of a drift ......................... 17
3.2 Different convention for the longitudinal phase space ........................................ 18
3.3 Graphical representation of the Twiss-parameters ............................................ 20

4.1 Problem of unadapted mesh ............................................................................... 23
4.2 Global process of space charge calculation ......................................................... 24
4.3 Frame definition - laboratory - rest frame - bunch frame .................................... 25

6.1 Cylindrical distribution: Longitudinal field $E_z$ along $z$ - comparison between simulation and theory ................................................................. 35
6.2 Form factor data and plot .................................................................................... 38
6.3 Ellipsoidal distribution: Longitudinal field $E_z$ along $z$ - comparison between simulation and theory ................................................................. 39
6.4 Comparison of theoretical (dashed) and numerical longitudinal field ............... 40

7.1 Rms value of the longitudinal field for a spheroidal distribution ....................... 43
7.2 Maximum value of the longitudinal field for a spheroidal distribution ............... 44
7.3 Relative difference between theory and simulation for different ellipsoids for the three components of the field ................................................................. 46
7.4 Rms value of the longitudinal field for a cylindrical distribution ....................... 48
7.5 Maximum value of the longitudinal field for a cylindrical distribution ............... 49

8.1 Canonical directions on which the fields are plotted ......................................... 53
8.2 Coin shape cylinder before and after rotation: canonical directions .................... 55
8.3 Rotation algorithm - Longitudinal Field ............................................................ 55
8.4 Rotation algorithm - Longitudinal Field ............................................................ 56

9.1 Phase space ellipse for a cylinder of emittance $\epsilon_{nx} = 5\text{mmrad:mm}$. for different set of parameters and units ................................................................. 60
9.2 Phase space ellipse for an ellipsoidal distribution with gaussian momentum ....... 61

10.1 Evolution of the transversal space charge for a drift without space charge ........ 65
10.2 Evolution of the longitudinal space charge for a drift without space charge ........ 66
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.3</td>
<td>Evolution of the transversal space charge for a drift without space charge</td>
<td>67</td>
</tr>
<tr>
<td>10.4</td>
<td>Evolution of the beam envelop for a focusing beam - beam waist definition</td>
<td>67</td>
</tr>
<tr>
<td>10.5</td>
<td>Evolution of the transversal space charge for a drift with space charge forces</td>
<td>68</td>
</tr>
<tr>
<td>10.6</td>
<td>Evolution of the longitudinal space charge for a drift with space charge forces</td>
<td>68</td>
</tr>
<tr>
<td>11.1</td>
<td>Illustration of the dependence on the momentum for the bending radius of a dipole</td>
<td>70</td>
</tr>
<tr>
<td>11.2</td>
<td>Definitions of the dipole origins</td>
<td>71</td>
</tr>
<tr>
<td>11.3</td>
<td>Dipoles dimensions at A0</td>
<td>71</td>
</tr>
<tr>
<td>11.4</td>
<td>Field map from the experimental measurement for the dipoles of A0</td>
<td>72</td>
</tr>
<tr>
<td>11.5</td>
<td>Magnetic field map as modelized by Astra for the dipoles of A0</td>
<td>73</td>
</tr>
<tr>
<td>11.6</td>
<td>Comparison between the measurement fields and astra modelized fields</td>
<td>74</td>
</tr>
<tr>
<td>11.7</td>
<td>Fitting the measured fields with a fringe field model</td>
<td>75</td>
</tr>
<tr>
<td>12.1</td>
<td>Evolution of a bunch through a compressor</td>
<td>76</td>
</tr>
<tr>
<td>12.2</td>
<td>Scheme of a compressor made of 4 dipoles and corresponding sign of the magnetic fields</td>
<td>76</td>
</tr>
<tr>
<td>12.3</td>
<td>Reference path through the four dipoles of the compressor</td>
<td>77</td>
</tr>
<tr>
<td>12.4</td>
<td>Deflection angle along the trajectory in the compressor</td>
<td>77</td>
</tr>
<tr>
<td>12.5</td>
<td>Longitudinal phase space before and after the compressor, without space charge forces</td>
<td>78</td>
</tr>
<tr>
<td>12.6</td>
<td>Longitudinal phase space before and after the compressor, with space charge forces</td>
<td>78</td>
</tr>
<tr>
<td>12.7</td>
<td>Size of the bunch after compression for different correlated energy spread</td>
<td>79</td>
</tr>
<tr>
<td>12.8</td>
<td>Emittance after compression for different correlated energy spread</td>
<td>79</td>
</tr>
<tr>
<td>12.9</td>
<td>Transversal phase space for the compressor without space charge forces</td>
<td>80</td>
</tr>
<tr>
<td>12.10</td>
<td>Transversal phase space for the compressor with space charge forces</td>
<td>80</td>
</tr>
<tr>
<td>12.11</td>
<td>Longitudinal phase space for the compressor without space charge forces</td>
<td>81</td>
</tr>
<tr>
<td>12.12</td>
<td>Transversal phase space for the compressor with space charge forces</td>
<td>81</td>
</tr>
<tr>
<td>12.13</td>
<td>Phase space at the end of the compressor with densities on each axis</td>
<td>82</td>
</tr>
<tr>
<td>13.1</td>
<td>Scheme of an emittance exchanger made of 4 dipoles and corresponding sign of magnetic fields</td>
<td>83</td>
</tr>
<tr>
<td>13.2</td>
<td>Reference path through the four dipoles in the dogleg configuration</td>
<td>84</td>
</tr>
<tr>
<td>13.3</td>
<td>Deflection angle along the trajectory through the dogleg</td>
<td>84</td>
</tr>
<tr>
<td>14.1</td>
<td>Fermilab cavity schema and picture</td>
<td>84</td>
</tr>
<tr>
<td>14.2</td>
<td>$E_{45}$ (red) in MV/m and the sine oscillating function (blue) adjusted in phase without scale</td>
<td>87</td>
</tr>
<tr>
<td>15.1</td>
<td>Dipoles hard edge coordinates for A0 emittance exchange lattice</td>
<td>95</td>
</tr>
<tr>
<td>16.1</td>
<td>NML facility</td>
<td>97</td>
</tr>
<tr>
<td>16.2</td>
<td>Dipoles hard edge coordinates for NML compressor lattice</td>
<td>98</td>
</tr>
<tr>
<td>A.1</td>
<td>Screen generated by the function ppSummaryPlot</td>
<td>105</td>
</tr>
<tr>
<td>A.2</td>
<td>Screen generated by the function ppSigmaPlot</td>
<td>106</td>
</tr>
<tr>
<td>D.1</td>
<td>Scheme of beam line elements at A0</td>
<td>118</td>
</tr>
<tr>
<td>E.1</td>
<td>Ellipsoid at rest with two rotations: the two rotation steps</td>
<td>119</td>
</tr>
<tr>
<td>E.2</td>
<td>Ellipsoid at rest with two rotations: canonical directions</td>
<td>120</td>
</tr>
<tr>
<td>E.3</td>
<td>Rotation algorithm - Longitudinal Field</td>
<td>120</td>
</tr>
<tr>
<td>E.4</td>
<td>Sphere at low energy: canonical directions</td>
<td>121</td>
</tr>
<tr>
<td>E.5</td>
<td>Rotation algorithm - Longitudinal Field along longitudinal direction</td>
<td>121</td>
</tr>
<tr>
<td>E.6</td>
<td>Rotation algorithm - Longitudinal Field along transverse direction</td>
<td>121</td>
</tr>
<tr>
<td>E.7</td>
<td>Rotation algorithm - Transverse Field in y direction</td>
<td>122</td>
</tr>
</tbody>
</table>
G.1 Cylindrical distribution - $E_z$ along $z$ for different $\sigma_z$ ......................... 125
G.2 Cylindrical distribution - $E_z$ along $z$ for different $\sigma_z$ ......................... 126
G.3 Ellipsoidal distribution - $E_z$ along $z$ for different $\sigma_z$ ......................... 127
G.4 Ellipsoidal distribution - $E_z$ along $z$ for different $\sigma_z$ ......................... 127
Introduction

**From particle accelerator to simulation**  Particle accelerators consist in controlled acceleration, guidage and storage of a particle beam until it reaches the design requirements at a target location. A particle beam is usually made of several bunches of particles emitted at different time. At Fermilab, a linear particle accelerator, called A0, has been designed for fundamental research in photo-injectors. The particles accelerated are electrons emitted by a laser-excited photo-cathode. The typical bunch charge at A0 is $-0.25 \text{nC}$. It is left to the reader to calculate that in a bunch, the number of particles, each interacting with the others and submitted to external forces, is $10^{9}$. This is far from a 2-body problem, removing any hope of theoretical solution for the equation of motions, and thus, leading us to a numerical resolution of the problem.

**Space charge calculation**  Space charge forces are the interaction forces between the particles of the bunch. These internal forces are often neglected as their amplitude might be small compared to the external fields applied to the particles. Nevertheless, for a better modelization of the bunch evolution in the accelerator, these forces should be taken into account, especially if the bunch is compressed in one or another direction, through a compressor or a focusing element. We will further see, that even in a single drift, space charge forces should not be neglected. Astra is a free particle tracking program, that offers the opportunity to calculate space charge with a two dimensionnal algorithm. A 3D algorithm is currently being implemented, and we propose here an improvement of this one to allow space charge calculation for bunches going in any directions. This report will discuss the implementation of this algorithm and its testing.

**Contents**  This report is divided into 5 main parts, the whole trying to retrace the coherent story of my work at Fermilab. The first part will quickly present some theoretical background required for the basic understanding of this report. We will try to systematically mention references, for the curiosity of the reader to be satisfied if more complete answers are needed. The second part will focus on the first and main objective of our work: the improvement of the space charge algorithm implemented in Astra. General and technical descriptions of the method will be presented. The algorithm implemented, a succession of tests has been applied in order to validate it. Particular attention has been paid to compare theoretical expressions and simulations results. Presenting this testing phase, our third part will try to convince the reader of the validity of the algorithm. The following step consists in testing the algorithm on specific components found in particle accelerator. The main component will be a dipole, as it clearly introduced change of beam direction. Eventually, the geometry and specifications of two different accelerators have been studied in order to simulate them.

**What is new ?**  As we mentionned above the aim of our work was to modify the 3D algorithm present in Astra, for distributions that goes in any directions. The hypothesis of the beam going mainly in a straight line is thus dropped. On top of that, a meshing adapted to the bunch direction is used, resulting in a better precision for a same computational cost. We first implemented a stand alone algorithm that we integrated in Astra in a second step. Discrepancies appeared between my
algorithm and the 3D algorithm currently in development. A long phase of debugging revealed an error of sign in the way the Lorentz transformation was done in the development version of the 3D algorithm from Klaus Floettman. Apart from this error, the two algorithms provides similar results. To allow a fast interpretation of the data, the comparison between several algorithm versions or several Astra simulations, multiple scripts have been written, representing more than 3000 lines of code. These scripts were written in the language $\textit{R}$ [7]. A modification of the way Astra outputs sigma matrices has been performed, for it to be in the most commonly used variables for phase space. Eventually we also studied the dipole fields from measurement data, and compared them with the one modelized in Astra. From our analysis, it appeared that the modelization of the dipoles should be improved in Astra. I hope that part of my work will help for the general development of this beautiful and free program that is Astra.

On the context and philosophy of this report I am studying at the french graduate school SUPAERO(ISAE) that delivers an aerospace engineering master's degree. SUPAERO offers the opportunity to do, before the achievements of the studies, a gap year of 2 internships instead of a 6 months master thesis. I am glad to have been offered a position at Fermilab for one of these two internships. The writing of this report is thus required by Fermilab and by my school for partial validation of my studies. Nevertheless, I will not report only the results from my work, but I will try to present what I have learned, what helped me to understand, and try to develop this with my own philosophy, making it a personal report. As a result of this, one can probably find that this report is unbalanced, going sometimes into useless details, while sometimes skipping a lot of theoretical and physical background. I personally like figures and formulas better than long sentences, but above all, I like to understand where they come from. When discovering a subject, answers to small questions and demonstrations of small results, are the main ingredients to satisfy curiosity, develop sensitivity and better understanding of more complex systems. I thus like to report the answers from small questions I raised, because they are the ones that really matter for me as they helped me in my understanding. Small results don’t ask much to be demonstrated, referring sometimes to beautiful mathematical notions, so I will report these results for them being daily useful to me, and hopefully, to somebody else. I owe most of these answers to discussions with Philippe Piot and Helend Edwards, that I would like to thank now, before reaching the acknowledgment part of this report.
### MAIN NOTATIONS, VALUES AND CONVENTIONS

- \( x, y, z \) Coordinates of a particle
- \( p_x, p_y, p_z \) Momentum of a particle \hspace{1cm} \( p_0 = 15 \text{MeV/c at A0} \)
- \( c \) Light velocity \hspace{1cm} 299792458 m/s
- \( \mathcal{E} \) Energy
- \( \beta \) Relative velocity \hspace{1cm} \( \approx 1 \) for relativistic speed
- \( \gamma \) Relativistic factor \hspace{1cm} \( \approx 30 \) for 15MeV
- \( \epsilon_0 \) Permittivity of a material (vacuum) \hspace{1cm} 4\pi \times 10^{-7}
- \( \mu_0 \) Permeability of a material (vacuum) \hspace{1cm} 4\pi \times 10^{-7}
- \( e \) Electron charge \hspace{1cm} \(-1.60218 \times 10^{-19} \) C
- \( m_e \) Electron rest mass \hspace{1cm} 9.10938215 \times 10^{-31} \) kg
- \( m_e c \) Momentum normalization factor \hspace{1cm} 0.510998 \approx 0.511 \) MeV/c
- \( \vec{p} \) Normalized momentum \hspace{1cm} \( \vec{p} / m_e c = \gamma \beta \), \( \vec{p} = 30.35 \) for 15MeV
- \( Q \) Bunch charge \hspace{1cm} 1-10 nC
- \( \sigma_{x,y,z} \) Standard deviation of the bunch radii
- \( \vec{E} \) Electric field
- \( \vec{B} \) Magnetic field

#### CAVITY PARAMETERS

- \( l_c \) The five cells cavity length \( l_c = 5\lambda /2 \) \hspace{1cm} 192.3mm
- \( L_c \) Cavity length including surrounding pipes \hspace{1cm} 300nm
- \( f \) Cavity frequency \hspace{1cm} 3.9 Ghz
- \( \lambda \) Cavity wave length \hspace{1cm} 76.87mm
- \( \omega \) Cavity proper pulsation \hspace{1cm} 81.738/m
- \( Scale \) Scaling factor for the field map \hspace{1cm} 9.3227 \times 10^6

#### ABBREVIATIONS

- A0 Linear accelerator facility at Fermilab
- NML New Muon Laboratory
- ILCTA International Linear Collider Testing Accelerator

### On the RMS definition and notation

We define the rms value of a statistical variable \( X \) as:

\[
\sigma_X = \sqrt{\langle X^2 \rangle_n - \langle X \rangle^2_n} \tag{1}
\]

where the brackets stands for the expected value, and the subscript \( n \), stands for normalized. If \( f \) is a normalized probability density function:

\[
\langle g \rangle_n = \int_\Omega g f \tag{2}
\]

If this function is not normalised, we will normalized it as follows:

\[
\langle g \rangle_n = \frac{\langle g \rangle}{\langle 1 \rangle} \tag{3}
\]

In the case of centered statistical variables, \( \langle g \rangle_n = 0 \) and thus the standard deviation reduces to:

\[
\sigma_g^2 = \langle g^2 \rangle_n \tag{4}
\]

Eventhough this is common, one has to pay attention not to do it systematically because it is not a general case.
Phase space convention used

Figure 1: Convention used for the longitudinal phase space

Figure 2: Legend of phase space plots. Most of the time, plots will be in plane $x-x'$, $y-y'$ and $z-z'$, where $z' = \delta p/p$. In this document phase space plots will be displayed in three different ways. Either a single ellipse (left plot), which is the canonical translation of the sigma matrix (beam matrix). Either a scatter corresponding to each particle coordinates, with the canonical ellipse (middle plot). Or, more informations can be provided by showing the density of the particles with faded colors (right plot). Red color translating a high density of particles, as opposite to lighter colors. The canonical ellipse is also plotted.

Dipole points definition

Figure 3: Convention used to define dipole pole tip coordinates. Geometrical input and output points are also drawn
Part I

A quick introduction to beam physics
# Table of Contents

1 Quick memo on electromagnetism and special relativity 8
   1.1 Basic notions of Electromagnetism and charged particles .......................... 8
      1.1.1 Electromagnetic fields: Maxwell equations .................................. 8
      1.1.2 Lorentz Force .......................................................... 8
      1.1.3 Equation of motion ...................................................... 8
   1.2 Basic notions of relativity .................................................................. 9
      1.2.1 Relativity parameters and useful formulas ..................................... 9
      1.2.2 Lorentz transformation ......................................................... 9

2 Beam dynamics - linear formalism 11
   2.1 Introduction ............................................................................. 11
      2.1.1 Coordinate system ........................................................... 11
      2.1.2 Equation of motion .......................................................... 12
   2.2 A particular case: Hill’s equation - Betatron functions ......................... 12
   2.3 General resolution of the perturbated linear equation of motion ................ 13

3 Phase space and matrix formalism 16
   3.1 Transportation matrix ............................................................... 16
      3.1.1 From equations of motion to matrix formalism ................................. 16
      3.1.2 Interpretation in term of optics .............................................. 17
   3.2 Beam matrix and phase space ...................................................... 17
      3.2.1 Phase space ...................................................................... 18
      3.2.2 Beam emittance ................................................................. 18
      3.2.3 Beam matrix ................................................................. 19
      3.2.4 Two dimensional beam matrix - Relation with Twiss-parameters .......... 19
      3.2.5 Relation with transport matrix .............................................. 20
      3.2.6 Four dimensional beam matrix - introduction to Emittance exchange .... 20
Quick memo on electromagnetism and special relativity

1.1 Basic notions of Electromagnetism and charged particles

1.1.1 Electromagnetic fields: Maxwell equations

In a vacuum environment, electric charges and currents are related to electric fields $\mathbf{E}$ and magnetic fields $\mathbf{B}$ through the Maxwell equations:

\[
\begin{align*}
\nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} \\
\nabla \cdot \mathbf{E} &= \frac{\rho}{\varepsilon_0} \\
\nabla \times \mathbf{B} &= \mu_0 \mathbf{j} + \varepsilon_0 \mu_0 \frac{\partial \mathbf{E}}{\partial t}
\end{align*}
\]

where $\rho$ is the density of the charges, $\mathbf{v}$ the velocity, $\mathbf{j} = \rho \mathbf{v}$ the current, $\varepsilon$ and $\mu$ the permittivity and permeability of the material (here the vacuum).

1.1.2 Lorentz Force

A particle of charge $q$ with velocity $\mathbf{v}$ in an electromagnetic field, receives the external Lorentz force:

\[
\mathbf{F} = q \left( \mathbf{E} + \mathbf{v} \times \mathbf{B} \right)
\] (1.1)

Charges can be expressed as multiples of the charge of an electron, $q = ne$. The charge of an electron $e$ being the elementary unit of electric charge: $e = -1.60217733 \times 10^{-19} C$. For high velocities and energies, a numerical application can reveal that for an equivalent force, it would be easier and less expensive if the force is generated by a magnetic field, rather than by an electric field. In general, the guidance and focusing of a beam in an accelerator will thus be performed by magnetic elements. Nevertheless, at low energies, electric fields at radio frequencies will be preferred and thus relegated to the acceleration task.

1.1.3 Equation of motion

\[
\frac{d\mathbf{p}}{dt} = \mathbf{F}
\] (1.2)

Where $\mathbf{p}$ is the momentum. In the physics of accelerators, relativistic definition of the momentum is used (see section 1.2).
1.2 Basic notions of relativity

1.2.1 Relativity parameters and useful formulas

The definition of the relativistic momentum is expressed in function of the rest mass \( m_0 \) (for an electron \( m_e = 9.10938 \times 10^{-31} \text{kg} \)):

\[
\vec{p} = \gamma m_0 \vec{v}
\]  

(1.3)

The relativistic factor \( \gamma \) being:

\[
\gamma = \frac{1}{\sqrt{1 - \beta^2}}
\]  

(1.4)

The relative velocity \( \beta \) being:

\[
\vec{\beta} = \frac{\vec{v}}{c}
\]  

(1.5)

The energy of a particle of rest mass \( m_0 \) travelling at \( \vec{v} \) is:

\[
E = \gamma m_0 c^2
\]  

(1.6)

The basic unit of energy used is the energy of a particle of charge \( e \) would gain while being accelerated between two conducting plates at a potential difference of one Volt: one electron volt (eV). By introducing the definitions of \( \gamma \), \( \beta \) and \( p \), we have the following expression for the energy:

\[
E^2 = c^2 p^2 + m_0^2 c^4
\]  

(1.7)

\[
\vec{p} = \gamma m_0 \vec{v} = \gamma m_0 c \vec{\beta} = \frac{E}{c} \vec{\beta}
\]  

(1.8)

The latter expression justify the expression of the momentum in the unit eV/c, instead of the classic mechanics one: kg m/s. In the case of a conserved linear momentum, without external forces we can write:

\[
\vec{v} = \frac{d \vec{r}}{dt} = \frac{c^2 \vec{p}}{E}
\]  

(1.9)

\[
\vec{r} = \text{cst} + \frac{c^2 \vec{p}}{E} t
\]  

(1.10)

1.2.2 Lorentz transformation

Space Transformation

In the four dimensional space, we consider two reference frames \( \mathcal{R} \) and \( \mathcal{R}' \) with a relative velocity \( \vec{v} \) between them. The coordinates of a point in each frame are \( X = (t,x,y,z) \), and \( X' = (t',x',y,z') \). The construction of Lorentz group transformation leads to the expression of any general Lorentz transformation:

\[
\begin{align*}
L &= -\omega \cdot S - \zeta \cdot K \\
A &= e^L
\end{align*}
\]  

(1.11)

(1.12)

where the coordinates from a frame to another are transformed by matricial multiplication by the Lorentz matrix \( A \):

\[
X' = A \cdot X
\]  

(1.13)

The 4×4 matrix \( L \), has null diagonal term. Its other terms can be expressed by six parameters (degree of freedom) and each of them can be decomposed on a base of 6 canonical 4×4 matrices \( S_i \) and \( K_i \), \( i \in \{1;2;3\} \), the first ones being antisymmetric contrary to the second ones that are symmetric (refer to [21] for the expression of these matrices). \( \omega \) and \( \zeta \) are 3 dimensional vectors that can be respectively interpreted in term of rotation and boost of the coordinate axis of the frame \( \mathcal{R}' \). The product \( \omega \cdot S \) and
\( \zeta \cdot K \) are thus tensorial products so that the result is a \( 4 \times 4 \) matrix.

For a boost without rotation (\( \omega = 0 \)) in an arbitrary direction, the boost vector can be written in terms of the relative velocity \( \vec{\beta} \):

\[
\zeta = \tanh^{-1}(\beta) \frac{\vec{\beta}}{\beta} \tag{1.14}
\]

The Lorentz transformation matrix reduces then:

\[
A_{\text{boost}}(\vec{\beta}) = \begin{pmatrix}
\gamma & -\gamma \beta_x & -\gamma \beta_y & -\gamma \beta_z \\
-\gamma \beta_x & \gamma^2 \left(1 + \frac{\beta^2}{\gamma^2} \right) & (\gamma - 1) \frac{\beta_x \beta_y}{\gamma^2} & (\gamma - 1) \frac{\beta_x \beta_z}{\gamma^2} \\
-\gamma \beta_y & (\gamma - 1) \frac{\beta_y \beta_x}{\gamma^2} & \gamma^2 \left(1 + \frac{\beta^2}{\gamma^2} \right) & (\gamma - 1) \frac{\beta_y \beta_z}{\gamma^2} \\
-\gamma \beta_z & (\gamma - 1) \frac{\beta_z \beta_x}{\gamma^2} & (\gamma - 1) \frac{\beta_z \beta_y}{\gamma^2} & \gamma^2 \left(1 + \frac{\beta^2}{\gamma^2} \right) \end{pmatrix} \tag{1.15}
\]

**Field Transformation**

Choosing the prime notation to refer to parameters in the moving frame, the fields in the laboratory frame are obtained using the following Lorentz transformation[21]:

\[
\vec{E} = \gamma \left( \vec{E}' - \vec{\beta} \times \vec{B}' \right) - \frac{\gamma^2}{\gamma + 1} \left( \vec{\beta} \cdot \vec{E}' \right) \vec{\beta} \tag{1.16}
\]

\[
\vec{B} = \gamma \left( \vec{B}' + \vec{\beta} \times \vec{E}' \right) - \frac{\gamma^2}{\gamma + 1} \left( \vec{\beta} \cdot \vec{B}' \right) \vec{\beta} \tag{1.17}
\]

Using the hypothesis that \( \vec{B}' = 0 \), the formulation reduces to:

\[
\vec{E} = \gamma \vec{E}' - \frac{\gamma^2}{\gamma + 1} \left( \vec{\beta} \cdot \vec{E}' \right) \vec{\beta} \tag{1.18}
\]

\[
\vec{B} = \gamma \vec{\beta} \times \vec{E}' \tag{1.19}
\]

**Longitudinal field and pure longitudinal momentum** It can be noticed that in the common situation where \( \vec{\beta} = \beta \vec{e}_z \), the field in the laboratory frame will be reduced to:

\[
E_z = \frac{\gamma + \gamma^2 (1 - \beta^2)}{\gamma + 1} E'_z = E'_z \tag{1.20}
\]

Thus, the two fields are the same for any velocity in the hypothesis of pure longitudinal momentum.
Chapter 2

Beam dynamics - linear formalism

2.1 Introduction

2.1.1 Coordinate system

Beam dynamics consists of the study of a bunch of particles, bunch among which one particle is labelled as the reference particle. The reference particle follows a pre-described path, which is the ideal path that the bunch would follow. Two frames are then defined. Cartesian coordinates are associated to the laboratory frame $\Sigma$, with axis $X$, $Y$ and $Z$, the later being the main direction of propagation for a linear accelerator. A second frame is defined, $\Sigma^*$, which moves along the reference particle trajectory. The axis associated with $\Sigma^*$, are the curvilinear axis $X^*$, $Y^*$ and $S^*$, following the reference particle in its motion, and forming a right handed coordinate system. $s$ being the curvilinear coordinate, we define the curvature vector for the reference particle as:

$$\vec{k} = -\frac{\vec{S}(s)}{ds} \quad (2.1)$$

In this frame, transverse coordinates $(x, y)$ of particles are expressed with respect to the reference particle coordinates. When using the equation of motion we have to be careful that the time of the particle is different than the time of the reference particle. As we are mainly studying linear accelerator, $S$ and $Z$ will be most of the time the same, so that we might use the $z$ coordinate instead of $s$ in some local cases.

![Curvilinear system of coordinates](image)

Figure 2.1: Curvilinear system of coordinates used in particle beam dynamics. The coordinates are given in a frame $\Sigma^*$ that moves along the reference trajectory.

We will try to remember in this section that coordinates used further are coordinates relative to the reference particle. As described in figure 2.2, when we will write $x'_m$, in fact it will correspond to a $dx'^m_m$ and it is the same for the output.
2.1.2 Equation of motion

We will not establish in this paper the equations of motion, but we strongly recommend the reading of the section 4.2 and 4.7 from H. Wiedemann’s book [43]. Nevertheless, I will present the resolution method I used, because it really helped me for my understanding of linear formalism, and according to me, justifies immediately the use of matrix formalism. From the previously cited reference, the perturbated linear equation of motion is:

\[ f(s) = u'' + K(s)u \]  \hspace{1cm} (2.2)

where \( u \) stands for \( x \) or \( y \), and \( s \) is the longitudinal coordinate. We chose to introduce since the beginning the dispersive term \( f \), which is the illustration that particles in a beam have different deflections due to their different energies. The finite energy spread, or chromatic error can be at a first order assessed with the term \( f(s) = \delta/\rho \), where \( \delta = \Delta p/p \). The resolution of equation 2.2 will follow in the next sections.

2.2 A particular case: Hill’s equation - Betatron functions

The resolution of the homogeneous equation will be performed in a general way in the next section, but only to focus and the the general solution of the perturbated equation. In beam physics a particular case of the homogeneous equation of motion was studied with attention, and a specific formalism has been introduced: the betatron functions. We will introduce it quickly in this section as they will be mentioned later in this document. Removing the dispersive term the trajectory of a particle satisfy the following equation of motion:

\[ u'' + K(s)u = 0 \]  \hspace{1cm} (2.3)

Assuming \( K(s) = K(-s) \), this equation is called Hill’s differential equation. If \( K \) were a constant (for instance positive), we know that the harmonic solution would be of the form \( A \cos(\omega s) + B \sin(\omega s) \), which can also be written \( \tilde{A} \cos(\omega s + \tilde{B}) \). This justify the use of an ansatz solution for equation 2.3 of the form:

\[ u(s) = \sqrt{\epsilon} \sqrt{\beta(s)} \cos(\psi(s) + \psi_0) \]  \hspace{1cm} (2.4)

Introducing this form in equation 2.3 will introduce several conditions for equation 2.4 to be a solution. Nevertheless we will not perform this operations here, and refer the reader to the following reference[43]. \( \psi \) is defined as the phase advance, and calculated as:

\[ \psi(s) = \int_0^s \frac{dt}{\beta(t)} \]  \hspace{1cm} (2.5)
and two other parameters are introduced in this formalism:

\[
\begin{align*}
\alpha(s) &= -\frac{1}{2} \frac{d\beta(s)}{ds} \\ 
\gamma(s) &= \frac{1 + \alpha(s)^2}{\beta(s)}
\end{align*}
\] (2.6) (2.7)

After some algebra[43], it can be shown that the previous sets of equations leads to the following, in which we recognize the equation of an ellipse:

\[
\gamma u^2 + 2\alpha uu' + \beta u' = \epsilon
\] (2.8)

It should not be forgotten here that, all symbols are functions of \(s\), except \(\epsilon\) which is an integration constant. Thus, equation 2.8 is the constant of motion to the homogeneous equation, called the Courant-Snyder invariant. As we will further see the the Courant-Snyder invariant \(\epsilon\), is the emittance. \(\alpha\), \(\beta\) and \(\gamma\) are called Twiss-parameters, defining the shape and orientation of the ellipse. Physical interpretations of these coefficients and their relations with the beam envelope in phase space will be described in the next section. Unfortunately, we will not go further in the study of betatron function in beam dynamics.

### 2.3 General resolution of the perturbed linear equation of motion

**Vectorial form**

Boundaries conditions must be added for the problem to be solvable:

\[
\begin{align*}
 f(s) &= u'' + K(s).u \\
 u(s_0) &= u_0 \\
 u'(s_0) &= u'_0
\end{align*}
\]

For the resolution we will use the general method of a \(n\)th degree linear differential equation using the Resolvent matrix. This method is very nice and has the advantage of giving a result directly on matrix form. To get started, let’s write equation 2.2 in vectorial formalism:

\[
\begin{pmatrix}
u' \\
u''
\end{pmatrix} =
\begin{pmatrix}0 & 1 \\K(s) & 0
\end{pmatrix}
\begin{pmatrix}u \\
u'
\end{pmatrix} +
\begin{pmatrix}0 \\f(s)
\end{pmatrix}
\] (2.9)

Now, we define \(X = (u \ u')^T\), and equation 2.9 reduces to the following general linear differential equation:

\[
X'(s) = A(s)X(s) + B(s)
\] (2.10)

The method we will further describe can apply to any vector \(X = (u \ u'' \ \ldots \ \u^{(n)})^T\) in a more general context of a \(n^{th}\) degree differential equation.

**Homogeneous equation**

We first solve the Homogeneous Equation:

\[
X'(s) = A(s)X(s)
\] (2.11)

and define the resolvent matrix \(R(s) = [C_1 \ \ldots \ \ C_n]\) where \(C_i = (x_{i}(s) \ x_{i}'(s) \ \ldots \ x_{i}^{(n)}(s))^T\) and \((x_i)_{i\in[1,n]}\) is a base of the vectorial space of solutions. These solutions are called principal solutions. As we have \(\forall i \ C_i' = A.C_i\) then

\[
R(s)' = A(s).R(s)
\]
and the general solution will be:

\[ X(s) = R(s) \Omega_0 \]

where \( \Omega_0 = (u_0 \ u'_0 \ ... \ u_0^{(n)})^T \), is a constant vector defined by the boundary conditions.

**Particular solution by variation of constant parameter**

Let’s find a solution that can be written \( X(s) = R(s) \Omega(s) \) and solve the general system.

\[
\begin{align*}
X' &= AX + B \\
X &= R \Omega
\end{align*}
\]

Replacing 2.13 in the general equation 2.12, and remembering that \( R' = AR \) we have:

\[ R \Omega' = B \]

The invertibility of \( R \), given by the fact that it contains base vectors leads us to:

\[ \Omega(s) = \int_{s_0}^{s} R^{-1}(t)B(t)dt \]

and the particular solution is defined.

**General Solution**

We eventually find the general solution as the sum of the particular solution and the homogeneous solution:

\[ X(s) = R(s).(\Omega_0 + \Omega(s)) \]

**Going back to two dimensions**

In two dimension, the resolvant matrix \( R(s) = [C_1 \ ... \ C_n] \), can be written, without loss of generality:

\[
R(s) = \begin{pmatrix} C(s) & S(s) \\ C'(s) & S'(s) \end{pmatrix}
\]

\[
R^{-1}(s) = \begin{pmatrix} S'(s) & -S(s) \\ -C'(s) & C(s) \end{pmatrix}
\]

Now, in our particular case, \( f(s) = \frac{1}{\rho_0}(s) \delta \)

And by assuming \( \delta \) stay constant between \( s_0 \) and \( s \):

\[ \Omega(s) = \int_{s_0}^{s} R^{-1}(t) \left( \begin{array}{c} 0 \\ \frac{\delta}{\rho_0}(t) \end{array} \right) dt = \delta \int_{s_0}^{s} \left( \begin{array}{c} \frac{-S(t)}{\rho_0(t)} \\ \frac{C(t)}{\rho_0(t)} \end{array} \right) dt \]

We can now define D, the dispersion function, from:

\[ R(s) \Omega(s) = \delta \begin{pmatrix} D(s) \\ D'(s) \end{pmatrix} \]

So that the dispersion function is:

\[ D(s) = \int_{s_0}^{s} \frac{1}{\rho_0(t)} [S(s)C(t) - C(s)S(t)] dt \]

Then we can add a new initial parameter, \( \delta = \frac{\delta p}{p} \) and eventually the general solution 2.15 can be written:

\[
\begin{pmatrix} u \\ u' \\ \delta \end{pmatrix} = \begin{pmatrix} C & S & D \\ C' & S' & D' \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} u(s_0) \\ u'(s_0) \\ \delta_0 \end{pmatrix}
\]
Particular case of $K(s) = k$ constant

It is straightforward to find principal solutions for equation 2.9.

For $k < 0$  \[ C(s) = \cosh \sqrt{-k} (s - s_0) \quad S(s) = \frac{\sinh \sqrt{-k} (s - s_0)}{\sqrt{-k}} \]

For $k > 0$  \[ C(s) = \cos \sqrt{k} (s - s_0) \quad S(s) = \frac{\sin \sqrt{k} (s - s_0)}{\sqrt{k}} \]

For $k = 0$  \[ C(s) = 1 \quad S(s) = (s - s_0) \]

These results close our section on general beam dynamics to now focus on matrix formalism and beam optics. Indeed, the case $K = k$ is a good approximation for most accelerator components, and the distinction between the cases $k < 0$, $k > 0$ and $k = 0$ respectively reflects the distinction between defocusing elements, focusing elements, and a simple drift.
Chapter 3

Phase space and matrix formalism

Linear theory introduced in the previous section calls for the use of matrix formalism for its linearity. From this, we will define the transformation matrix, which allows to determine the trajectory of a single particle by simple matrix multiplication. Then, we will introduce the phase space, and the ellipse representation of the beam, that allow to reduce the problem of the study of numbers of particle, to a single envelop determined by six parameters. This will also be the perfect time for us to introduce the notion of emittance, which is a key element in phase space.

3.1 Transportation matrix

3.1.1 From equations of motion to matrix formalism

In the previous section, the resolution of the homogeneous equation of motion revealed, that in a none dispersive system, the coordinates $u$ and $u'$ of a particle are entirely determined by initial conditions and the knowledge of principals solution. These functions, $C$ and $S$, which depend only on the system, determine entirely the $R$ matrix:

$$ R(s) = \begin{pmatrix} C(s) & S(s) \\ C'(s) & S'(s) \end{pmatrix} $$  \hspace{1cm} (3.1)

In the case of $K = k$ constant, the functions $C$ and $S$ are sine like functions, thus, easy to determine. This transformation matrix provides for any vector of input coordinates, the coordinates of the particle at any position is known by doing the simple multiplication:

$$ \begin{pmatrix} u(s) \\ u'(s) \end{pmatrix} = \begin{pmatrix} C(s) & S(s) \\ C'(s) & S'(s) \end{pmatrix} \cdot \begin{pmatrix} u_0 \\ u'_0 \end{pmatrix} $$  \hspace{1cm} (3.2)

Most of the time, the matrix is used to determine coordinates at the output of the element, so that the dependence in $s$ is dropped, replaced by the length of the component, the $R$ matrix being constant:

$$ X_{out} = R.X_{in} $$  \hspace{1cm} (3.3)

The 'R' matrix are called transport matrix. The matrix formalism offers the opportunity to follow any particle trajectory along a beam line by repeated matrix multiplications from element to element. If the beam line is made of $n$ elements, including drifts, and if an $R$ matrix can be associated with all elements(i.e. $K = cst$ on each element), a global transport matrix for the beam line can be easily defined: $R = R_n.R_{n-1}...R_2.R_1$. This kind of method is used by some algorithm like TRACE3D[8], to perform particle tracking.
3.1.2 Interpretation in term of optics

Case of a drift  According to figure 3.1, if the red line stands for the beam trajectory through a simple drift, we have the following result in two dimensions with the approximation of small angles:

$$
\begin{align*}
(x_{\text{out}}, x'_{\text{out}}) &= (x_{\text{in}} + \tan(x')d \approx x + d.x') \\
&= \begin{pmatrix} 1 & d \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_{\text{in}} \\ x'_{\text{in}} \end{pmatrix}
\end{align*}
$$

In this example we recognize the matrix corresponding to the case $K = 0$, developed in the previous chapter.

![Figure 3.1: Illustration of the matrix formalism in optics - case of a drift. It appear clear that for a linear trajectory (in red), $x'$ stays constant, while the increase of $x$ is a function of the drift length $d$.](image)

Thin lens approximation  Principal solutions for $K = k$ constant, have been given in the previous chapter. The corresponding transport matrix for an element of length $l$ are:

For $k < 0$ (defocusing element)

$$
R_D = \begin{pmatrix}
\cosh \sqrt{|k|}l & \frac{1}{\sqrt{|k|}} \sinh \sqrt{|k|}l \\
\sqrt{|k|} \sinh \sqrt{|k|}l & \cosh \sqrt{|k|}l
\end{pmatrix}
$$

(3.4)

For $k > 0$ (focusing element)

$$
R_F = \begin{pmatrix}
\cos \sqrt{k}l & \frac{1}{\sqrt{k}} \sin \sqrt{k}l \\
-\sqrt{k} \sin \sqrt{k}l & \cos \sqrt{k}l
\end{pmatrix}
$$

(3.5)

If the length of the element tend to 0 ($l \rightarrow 0$), we have $\sin \sqrt{k}l \sim \sqrt{k}l$, $\cos \sqrt{k}l \sim 1$, and we can write the previous matrix in the thin lens approximation:

$$
R_F = \begin{pmatrix}
1 & 0 \\
\pm 1 & 1
\end{pmatrix}
$$

(3.6)

where we defined the focal length $f^{-1} = kl$. One can recognize the matrix of a focusing/defocusing lens in the previous equation, revealing one again the analogy between optical and particle beams.

3.2 Beam matrix and phase space

The understand the collective motion of a beam one require the trajectory of each single particle along with its momentum. To fully describe the state of a particle, and to describe it in a representative way, a six-dimensional space is introduced: the phase space. Nevertheless, the knowledge of this coordinates for each single particle represent too much information, so that only the envelope of the beam is represented in the phase space. In practice, different curvatures can be found in the shape of the beam, nevertheless the canonical form chosen would be an ellipse and its representation by the so-called sigma-matrix.
3.2.1 Phase space

In the theory of beam physics, the three cartesian coordinates are not enough to fully describe the state of the particle, so that a system of 6 coordinates is used: \((x, x', y, y', s, \delta)\). Each coordinate being expressed with respect to the reference particle. Thus, unlike curvilinear coordinates, the coordinate \(s\) is the longitudinal distance to the reference particle. This is the tricky part, when studying phase space coordinates, one has to keep in mind this definition. When confusion can be made, we will write these coordinates as: \((\tilde{x}, \tilde{y}, \tilde{s})\) to emphasize their relative definition. \(x'\) and \(y'\) represent the angle relative to their designed momentum. The transverse momenta \(p_x\) and \(p_y\) are also used. \(\delta = \Delta p/p\) is the normalised momentum deviation. Energy deviation can also be used \(\Delta E/E\) without significative difference remembering that \(cp = \beta E\). This 6-dimensional space is called phase space and will be further used to study the evolution of the envelope of a bunch through an accelerator. In absence of coupling between the different plane, the phase space may be split into three independent 2-dimensional phase planes. This hypothesis is often used, eventhough coupling terms are always present, and might sometime be significant.

Two conventions exists for the longitudinal phase space. Indeed, let’s consider a particle which is in front of the reference particle in term of the longitudinal coordinate: \(z > z_0\), then this particle will arrive before the reference particle, so its time of arrival is less than the time of arrival of the reference particle: \(t < t_0\), or in term of phase \(\Phi < \Phi_0\). This results in two different phase space representations as one can see on figure 3.2. To go from one representation to the other we just have to flip the scheme horizontally. As a result of this, the shape of the ellipse is modified. If one is not careful, this could lead to completely different interpretations while analyzing a phase space scheme. In Astra, an in the rest of this document the "z convention" will be used.

![Figure 3.2: Different convention for the longitudinal phase space. (left) Convention used by astra and further in this document: z is is used as reference - (right) Convention using the difference of phase, or arrival time with respect to the reference particle](image)

3.2.2 Beam emittance

Each particle of a beam is a point in the phase space. The envelope of all these points represents the region occupied by the beam in phase space. This region is called beam emittance, and its value is normalized proportionally to the surface area. In the case of absence of coupling between the different phase plane, three independent beam emittances are defined. In each plane, the region delimited by the particles will be represented by an ellipse for its easy analytical description. The general equation of a 2D phase space ellipse is:

\[
\gamma x^2 + 2\alpha xx' + \beta x'^2 = \epsilon
\]  

(3.7)

The notation \(\alpha, \beta, \gamma, \epsilon\) are commonly used. They indeed recall the notations used when we introduce the betatron functions, the Courant-Snyder invariant, and the Twiss-parameters in section 2.2. From geometric properties of an ellipse we have the relation:

\[
\beta \gamma - \alpha^2 = 1
\]

(3.8)
The area defined by this two dimensional ellipse is:

$$A = \int_{\text{ellipse}} dx dx' = \pi \epsilon$$  \hfill (3.9)

The emittance, for instance in the plane \((x, x')\), can be expressed as the root mean square value (rms):

$$\epsilon_x = \sqrt{\langle x^2 \rangle \langle x'^2 \rangle - \langle xx' \rangle^2} = \sqrt{\sigma_{xx}^2 - \sigma_{xx'}^2} \hfill (3.10)$$

Nevertheless, the canonical parameters related to each other are \(x\) and \(p_x\), instead of \(x\) and \(x'\). As a result of this, a normalised emittance is defined as:

$$\epsilon_{nx} = \frac{1}{m_0 c} \sqrt{\langle x^2 \rangle \langle p_x^2 \rangle - \langle xp_x \rangle^2} \hfill (3.11)$$

The relation between the two is:

$$\epsilon_{nx} = \beta \gamma \epsilon_x \hfill (3.12)$$

### 3.2.3 Beam matrix

The general equation of an \(n\)-dimension ellipse can be written as:

$$u^T \Sigma^{-1} u = 1 \hfill (3.13)$$

where \(u \in \mathbb{R}^n\) and \(\sigma \in \mathbb{R}^n \times \mathbb{R}^n\). The ellipse volume is:

$$V = \frac{\pi^{n/2}}{\Gamma(1 + n/2)} \sqrt{\det \Sigma} \hfill (3.14)$$

In the context of phase space, a maximum of six dimensions will be used with \(u = (x, x', y, y', z, z')^T\). The \(\Sigma\)-matrix will be also called the beam matrix.

### 3.2.4 Two dimensional beam matrix - Relation with Twiss-parameters

The equation of the phase space ellipse in \(\sigma\)-notation is as following:

$$u^T \Sigma^{-1} u = 1 \hfill (3.15)$$

Let’s identify it with the previously introduced equation of the ellipse in Twiss-parameters:

$$\gamma x^2 + 2\alpha xx' + \beta x'^2 = \epsilon \hfill (3.16)$$

The link between these two equation can be obtained by equaling the two ellipse areas:

$$\frac{A}{\pi} = \sqrt{\det \Sigma} = \sqrt{\sigma_{11} \sigma_{22} - \sigma_{12}^2} \quad \text{and} \quad \frac{A}{\pi} = \epsilon \hfill (3.17)$$

This leads to the following relation:

$$\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix} = \epsilon \begin{pmatrix} \beta & -\alpha \\ -\alpha & \gamma \end{pmatrix} \hfill (3.18)$$

One has to be careful not to be confused between elements of the \(\Sigma\) matrix, and the standard deviation their are related to: \(\sigma_{11} = \sigma_{22}^x\), \(\sigma_{22} = \sigma_{xx'}^x\), and \(\sigma_{12} = \sigma_{xx'}\). For this reason, we will explicitly write down the relations described by equation 9.7 and schematize their geometrical interpretation in figure 3.3.
σₓ = √ε/β \hspace{1cm} \text{(rms beam envelope)} \hspace{1cm} (3.19)

σₓ′ = √ε/γ \hspace{1cm} \text{(rms beam divergence)} \hspace{1cm} (3.20)

σₓ = √ε/β \hspace{1cm} \text{(rms beam envelope)} \hspace{1cm} (3.21)

3.2.5 Relation with transport matrix

If the transportation matrix from a point 0 to a point 1 is known, then the beam matrix a point 1 Σ₁ can be related to the beam matrix at point 0 with the following relation:

Σ₁ = MΣ₀Mᵀ \hspace{1cm} (3.22)

Given an input sigma matrix, one can try to find which succession of elements will provide a transportation matrix M, such that the output sigma matrix will satisfy specific requirements.

3.2.6 Four dimensional beam matrix - introduction to Emittance exchange

To simplify the study of the beam matrix evolution as it propagates through a beam line, we will assume the initial beam matrix to be diagonal by block:

Σᵢₙ = \begin{pmatrix} Σₓ & 0 \\ 0 & Σₚ \end{pmatrix} \hspace{1cm} (3.23)

Emittance exchange is complete if after a transformation, the output sigma matrix is of the form:

Σᵢₜ = \begin{pmatrix} Σₚ & 0 \\ 0 & Σₓ \end{pmatrix} \hspace{1cm} (3.24)

Writing the transportation matrix M in a general form:

M = \begin{pmatrix} a₁₁ & a₁₂ & b₁₁ & b₁₂ \\ a₂₁ & a₂₂ & b₂₁ & b₂₂ \\ c₁₁ & c₁₂ & d₁₁ & d₁₂ \\ c₂₁ & c₂₂ & d₂₁ & d₂₂ \end{pmatrix} \hspace{1cm} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \hspace{1cm} (3.25)

Equation 3.22 results in:

Σᵩₒᵤᵗ = \begin{pmatrix} AΣₓAᵀ + BΣₚBᵀ & AΣₓCᵀ + BΣₚDᵀ \\ CΣₓAᵀ + DΣₚBᵀ & CΣₓCᵀ + DΣₚDᵀ \end{pmatrix} \hspace{1cm} (3.26)

Complete emittance exchange is reached if A = 0 & D = 0

Figure 3.3: Graphical representation of the Twiss-parameters
Part II

Implementation of the space charge algorithm
# Table of Contents

4 On the implementation of a 3D algorithm in Astra

4.1 Wider prospects for the algorithm .................................................. 23
  4.1.1 The need to adapt the meshing ................................................. 23
  4.1.2 No momentum direction favoured ........................................... 23

4.2 General description of the method .................................................. 24

4.3 Detailed description of the method .................................................. 25
  4.3.1 Frame definitions ........................................................................ 25
  4.3.2 Boosting the distribution to the rest frame ................................... 25
  4.3.3 Rotation of the bunch to its principal axis ................................... 25
  4.3.4 Diagonalization algorithm .......................................................... 26
  4.3.5 Space charge calculation ............................................................. 26
  4.3.6 The 3d space charge algorithm ..................................................... 27
  4.3.7 Solving the equations of motion ................................................... 27

5 Space Charge calculation ........................................................................ 28

  5.1 Poisson equation .............................................................................. 28
  5.2 Resolution of poisson screened equation with the use of Fourier transform .............................................. 29
  5.3 Resolution of poisson screened equation with the use of Green function .................................................... 29
On the implementation of a 3D algorithm in Astra

4.1 Wider prospects for the algorithm

4.1.1 The need to adapt the meshing

The 3d space charge calculation implemented in Astra uses rectangular mesh, each mesh having the same dimension. Even though this can be satisfying for cylindrical or spherical distribution which keeps a trajectory close to one of the laboratory axis (e.g. z), a lack of efficiency will be found for distorted distributions or trajectories which are not parallel to one of the frame axis. This is illustrated in figure 4.1 where the rectangular meshing along the frame axis results in a lot of empty mesh. Since the number of meshes in each direction strongly influence the computational cost, an adapted meshing should be applied. Reducing the number of empty mesh and increasing the resolution of the mesh on the distribution by keeping the implemented method of rectangular meshing will be performed by rotating the frame axis. As illustrated in figure 4.1, this will result in a better adapted meshing, increasing the accuracy of the calcul for a same computational cost (number of mesh)

![Figure 4.1: Problem of unadapted mesh. On the left, one can see that a lot of cells are empty, so that computational time is lost, and excessive number of particles are present in other cells, resulting in a bad numerical solution. An adapted mesh as presented on the right figure, would solve these problems.](image)

4.1.2 No momentum direction favorised

Astra was implemented to study particles travelling along the z axis, the main component of momentum being $\beta_z$. Thus several approximations were done knowing that few deviations from the z axis would occurred, mainly $\vec{\beta} \approx \beta_z \vec{e}_z$. As soon as elements such as dipoles are introduced in the
beam lattice, the main direction of the momentum is changed and these approximations are not valid any more. Moreover, the introduction of a dipole will introduce situations of undapted meshing as described in the previous section 4.1.1. The algorithm we implemented does not favorised any direction of momentum and designs an adapted grid around the distribution that is not parallel to the frame axis. A similar approach is actually followed by Klaus Floettman for the next version of Astra, but has not been released at the time this paper is written.

### 4.2 General description of the method

The prospects previously described will imply several requirements, one of them being of course the possibility of importing our algorithm in Astra. A stand alone main program has been implemented in order to test our space charge calculation algorithm before its incorporation in Astra’s program infrastructure. The adaptation of the meshing will require an interface between the absolute coordinates and the rotated grid. Indeed, Astra uses absolute laboratory coordinates whereas the grid is defined in a rotated frame. This interface will be provided by storing the coordinates of the corner of the grid and the components of the three axis defining the rotated grid (see figure 4.3). The other main change, would be the Lorentz transformation of the coordinates and fields, so that they could take into account any momentum direction. The global process of space charge calculation is schematised on figure 4.2 where the steps concerning the lorentz boost transformation and the rotation of the frame are presented. Attention has to be paid on these different steps because from their simple appearance, underlying subtle manipulation of coordinates and components are required. For instance, after the resolution of Poisson algorithm, the fields are known on each points of the rotated grid. These fields have components in the rotated frame and are expressed with respect to coordinates also in the rotated frame.

![Figure 4.2: Global process of space charge calculation. Poisson equation has to be solved in the rest frame, where by hypothesis, there is no motion of particles. In this frame, a frame adapted to the bunch is found by diagonalization of the covariance matrix of the particles.](image)

August 2009 24 Emmanuel Branlard
4.3 Detailed description of the method

4.3.1 Frame definitions

Figure 4.3: Frame definition - laboratory - rest frame - bunch frame

The laboratory frame $R$, the rest frame $R'$. We will note respectively $(e)$, $(e')$ and $(e'_b)$ the canonical bases adapted to the laboratory frame, the rest frame and the main directions of the bunch.

4.3.2 Boosting the distribution to the rest frame

The Lorentz space transformation that should be applied to the particle has been studied in a previous section. For the numerical situation, the time dependence is removed so that, the matrix of equation 1.15 reduces to:

$$A(\beta) = \begin{pmatrix}
1 + (\gamma - 1)\frac{\beta^2}{\gamma^2} & (\gamma - 1)\frac{\beta_x\beta_y}{\gamma^2} & (\gamma - 1)\frac{\beta_x\beta_z}{\gamma^2} \\
(\gamma - 1)\frac{\beta_y\beta_x}{\gamma^2} & 1 + (\gamma - 1)\frac{\beta_y^2}{\gamma^2} & (\gamma - 1)\frac{\beta_y\beta_z}{\gamma^2} \\
(\gamma - 1)\frac{\beta_z\beta_x}{\gamma^2} & (\gamma - 1)\frac{\beta_z\beta_y}{\gamma^2} & 1 + (\gamma - 1)\frac{\beta_z^2}{\gamma^2}
\end{pmatrix} \tag{4.1}$$

Coordinates of a particle $X$ in the rest frame $R'$ are obtained with the following multiplication:

$$X|_{e'} = A \cdot X|_{e} \tag{4.2}$$

4.3.3 Rotation of the bunch to its principal axis

To adapt the meshing to the distribution, one need to detect the distribution main axis. For this, we used the covariance matrix of the particles. Let $C$ being this $3 \times 3$-matrix. For this method, there should be no difference between using coordinates in the rest frame or in the laboratory frame. In our case, we chose the coordinates in the rest frame $R'$.

$$C = \text{Cov} \{ x'_i, y'_i, z'_i \} \tag{4.3}$$

$C$ is by definition a symmetric real matrix, hence diagonalizable to a matrix $D$. Once we find the eigen vectors of this matrix, we have the principal axis of the distribution. Thus diagonalizing the covariance matrix is an elegant method to further adapt the meshing on the distribution. The diagonalization process is described in the following section 4.3.4 and in annex B. Rotations and base transformation can be confusing, but not if one takes the time to write the proper formalism:

$$P|_{e'_b} = \text{Mat} (\text{Id}, e'_b, e') \tag{4.4}$$

$$P|_{e'_b} = \text{Mat} (\text{Id}, e', e'_b) = P|_{e'_b}^{-1} \tag{4.5}$$
WHERE \( P|_{e'}^b \) IS THE TRANSFORMATION MATRIX FROM THE BASE \((e')\) TO THE BASE \((e'_b)\). WE WILL IMMEDIATELY NOTE THAT THESE MATRIX ARE ROTATIONS, I.E. ORTHOGONAL MATRIX, SUCH THAT \( P^{-1} = P^T \) AND \( \det P = 1 \). THE DIAGONALIZATION OF \( C \) CAN BE WRITTEN:

\[
D|_{e'_b} = P|_{e'_b}^e \cdot C|_{e'}^e \cdot P^T|_{e'_b}^e
\]

(4.6)

LET US DEFINE \( E'_{b,1}|_{e'}^e, E'_{b,2}|_{e'}^e, E'_{b,3}|_{e'}^e \) THE THREE NORMALISED EIGEN VECTORS OF \( C \) FORMING THE BASE \((e'_b)\), WHICH COORDINATES ARE EXPRESSED IN THE BASE \((e')\). THE 3\( \times \)3-MATRIX FORMED BY THE THREE COLUMN VECTORS \( E'_{b,i} \) IS:

\[
P|_{e'_b}^e = \begin{pmatrix}
E'_{b,1} & E'_{b,2} & E'_{b,3}
\end{pmatrix}
\]

(4.7)

AND TO SIMPLIFY NOTATIONS WE WILL FURTHER WRITE IT \( R \): ACCORDING TO THE PREVIOUS FORMALISM WE THEN HAVE:

\[
R = P|_{e'_b}^e
\]

(4.8)

\[
D = R^T \cdot C \cdot R
\]

(4.9)

AND EVENTUALLY, LET’S WRITE DOWN THE FACT THAT, FOR A GIVEN VECTOR \( X \) EXPRESSED IN THE REST BASE \((e'_b)\), ITS COORDINATES IN \((e')\) ARE:

\[
X|_{e'_b} = P|_{e'_b}^e \cdot X|_{e'} = R^T \cdot X|_{e'}
\]

(4.10)

NOW THAT THIS ALGEBRA IS CLEAR, THE IMPLEMENTATION OF THE ALGORITHM IS STRAIGHTFORWARD

- Compute the covariance matrix
- Diagonalize it and store the eigenvectors as the rotation matrix \( R \)
- Multiply the coordinates of the particles in the rest frame by the transposed rotation matrix
- Go on with the 3D Poisson solver in this bunch-adapted base

### 4.3.4 Diagonalization algorithm

A choice between the several algorithms of diagonalization has been made. From our analysis, the Jacobi algorithm has been selected as it takes into account the fact that the matrix is symmetric. We developed in annex B the complete Jacobi method, and provide as well the source code I wrote in fortran to implement this algorithm. The other methods considered were:

- The Givens reduction: which is a variant of the Jacobi method, which requires a finite number of iteration and return a tridiagonal matrix
- The Householder algorithm: reduces a symmetric matrix \( A \) to tridiagonal form by orthogonal transformations.
- The factorization method: another common technique consisting in decomposing the matrix into a left factor and a right factor. This method, is more subtle and does not exploit the symmetric form of the matrix.

The Jacobi method has been selected for its simplicity of principle, and its rapidity to diagonalize a 3\( \times \)3 symmetric matrix.

### 4.3.5 Space charge calculation

The 2D space charge algorithm of Astra is a cylindrical symmetric algorithm. Each grid consist in a cylindrical ring in the radial direction forming a slice in the longitudinal direction. After “boosting” the grid in the rest frame, a static field calculation is performed, with the assumption that the charge is constant in each ring. Fields are calculated at the center of each grid and transformed back in the laboratory frame. To determine the fields in arbitrary location, a cubic spline interpolation is used. Outside of the grid, an \(1/r\) extrapolation is used.
4.3.6 The 3d space charge algorithm

Follow the resolution method of Poisson's equation developed in chapter 5 of this document. Its main hypothesis is that the particle are at rest after boost, so that Poisson equation is valid. Indeed if the particle are at rest, no courant is present in the distribution and thus Maxwell equations reduces to Poisson equation. This also means that the magnetic field is null in the rest frame $\vec{B}' = 0$.

4.3.7 Solving the equations of motion

Astra uses a Runge-Kutta model to perform the integration of the equations of motion. In these equations, both the contributions of the external and internal fields are taken into account. Nevertheless, internal forces being expected to evolve slowly, the space charge algorithm is not executed at every time step. Different criteria, such as the evolution of the aspect ratio of the bunch, determine whether the space charge algorithm should be called. If not, the previously calculated fields are scaled accordingly to the variation of the beam size.
Chapter 5

Space Charge calculation

The resolution of poisson equation is the usual way to find the electric potential for a given charge distribution.

5.1 Poisson equation

Assuming a linear, isotropic and homogeneous medium, Gauss’ law is written as follows:

\[ \nabla \cdot \vec{E} = \frac{\rho}{\epsilon} \]  \hspace{1cm} (5.1)

In the absence of a changing magnetic field, \( \vec{B} \), Faraday’s law of induction gives:

\[ \nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} = 0 \]  \hspace{1cm} (5.2)

Since the curl of the electric field is zero, it is defined by a scalar electric potential field \( \phi \):

\[ \vec{E} = -\nabla \phi \]  \hspace{1cm} (5.3)

Substituting \( \vec{E} \) provides us with a form of the Poisson equation:

\[ \nabla \cdot \nabla \phi = \nabla^2 \phi = \Delta \phi = -\frac{\rho}{\epsilon}. \]  \hspace{1cm} (5.4)

Where \( \Delta \) is the Laplace operator, which takes the following form in cartesian coordinates:

\[ \Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \]  \hspace{1cm} (5.5)

Equation 5.4 is a particular case of the inhomogeneous partial differential equation named Screened Poisson equation which with usual mathematical notation is the following:

\[ [\Delta - \lambda^2] u(r) = -f(r) \]  \hspace{1cm} (5.6)

where \( \Delta \) is the Laplace operator, \( \lambda \) is a constant, \( f \) is an arbitrary function of position (known as the "source function") and \( u \) is the function to be determined. This equation is defined in unbounded space and is subject to the condition that \( u(r) \) vanishes sufficiently rapidly as \( r \to \infty \).
5.2 Resolution of poisson screened equation with the use of Fourier transform

The resolution of equation 5.6 is performed through a 3d spacial Fourier transformation of function $f(r)$, this operation being possible due to the unbounded domain of definition $-\infty < x, y, z < \infty$. The Fourier transform and its inverse are respectively defined as:

$$\hat{f}(\vec{k}) = \mathcal{F}(f) = \int_{\Omega} f(\vec{r}) e^{-i\vec{k} \cdot \vec{r}} d^3r$$  \hspace{1cm} (5.7)

$$f(\vec{r}) = \mathcal{F}^{-1}(f) = \frac{1}{(2\pi)^3} \int_{\Omega} \hat{f}(\vec{k}) e^{i\vec{k} \cdot \vec{r}} d^3k$$  \hspace{1cm} (5.8)

Note: multiple conventions exists for Fourier transform, one could have used a coefficient $1/(2\pi)^{3/2}$ before the integral for both the Fourier transform and its inverse, or no coefficient at all but an exponential coefficient with $2\pi i$. The choice made here is justified by the normalization convention $\mathcal{F}(\delta(\vec{r})) = 1$.

The Fourier transform of equation 5.6, after two successive integrations and the use of boundary condition, reduces eventually to the algebraic equation:

$$(k^2 + \lambda^2) \hat{u}(\vec{k}) = \hat{f}(\vec{k})$$  \hspace{1cm} (5.10)

Which yields the solution:

$$\hat{u}(\vec{k}) = \frac{\hat{f}(\vec{k})}{k^2 + \lambda^2}$$  \hspace{1cm} (5.11)

The Fourier inverse transformation provides the desired solution:

$$u(\vec{r}) = \mathcal{F}^{-1}(u) = \frac{1}{(2\pi)^3} \int_{\Omega} \hat{f}(\vec{k}) e^{i\vec{k} \cdot \vec{r}} d^3k$$  \hspace{1cm} (5.12)

The integrand is known because $f$ is known and it is straightforward to compute $\hat{f}$. Thus one can be satisfied of this expression. Nevertheless it is interesting to develop the expression of $\hat{f}(\vec{k})$ as a Fourier transform, leading to the double integral:

$$u(\vec{r}) = \frac{1}{(2\pi)^3} \int \int \frac{1}{k^2 + \lambda^2} e^{i\vec{k} \cdot (\vec{r} - \vec{r}')} f(\vec{r}') d^3r' d^3k$$  \hspace{1cm} (5.13)

Expression in which we recognize an integral representation for the Green’s function $G$ for equation 5.6:

$$u(\vec{r}) = \int G(\vec{r}, \vec{r}') f(\vec{r}') d^3r'$$  \hspace{1cm} (5.14)

with

$$G(\vec{r}', \vec{r}) = \frac{1}{(2\pi)^3} \int \frac{1}{k^2 + \lambda^2} e^{i\vec{k} \cdot (\vec{r}' - \vec{r})} d^3k$$  \hspace{1cm} (5.15)

The formalism of Green function is common in differential equation formalism and deserved attention. This approach will be thus studied in the next section.

5.3 Resolution of poisson screened equation with the use of Green function

Poisson’s screened equation uses a linear operator so its solutions are superposable and this suggests a general method for solving this equation. Suppose that we could construct all of the solutions generated
by point sources, provided that they satisfy the appropriate boundary conditions. Any general source function can be built up out of a set of suitably weighted point sources, so the general solution of Poisson’s equation must be expressible as a weighted sum over the point source solutions. Thus, once we know all of the point source solutions we can construct any other solution. In mathematical terminology, we require the solution to

$$\nabla^2 G(\mathbf{r}, \mathbf{r}') = -\delta(\mathbf{r} - \mathbf{r}')$$

which goes to zero as $|\mathbf{r}| \to \infty$. The function $G(\mathbf{r}, \mathbf{r}')$ is the solution generated by a unit point source located at position $\mathbf{r}'$. This function is known as a Green’s function and from now on we will consider it to depend only on the difference $\mathbf{r} - \mathbf{r}'$. In other words

$$G(\mathbf{r}, \mathbf{r}') = G(\mathbf{r} - \mathbf{r}')$$

This reflects the translational invariance of the unbounded domain with the disturbance depending only on the relative separation from the source. The solution generated by a general source function $f(\mathbf{r})$ is simply the appropriately weighted sum of all of the Green’s function solutions:

$$u(\mathbf{r}) = \int_{\Omega} G(\mathbf{r} - \mathbf{r}') f(\mathbf{r}') d^3 r'.$$

The formalism of Green function and the proof of this solution is provided in annex C. As stated above, this is a superposition of screened $1/r$ functions, weighted by the source function $f$ and with $\lambda$ acting as the strength of the screening.

An analytical expression of the Green functions associated to Poisson’s screened equation can be evaluated. The presence of the term $k^2$ suggests the use of spherical polar coordinates $(\rho, \theta, \phi)$, with the polar axis along $\mathbf{R} = \mathbf{r} - \mathbf{r}'$, such that, $\mathbf{R} = \mathbf{R}$, so $\mathbf{R} \cdot \mathbf{R} = \rho R \cos \theta$, and the elementary integration volume being $dV = d\rho d\theta d\phi$. Equation 5.15 becomes:

$$G(\mathbf{R}) = \frac{1}{(2\pi)^3} \int_0^\infty \rho^2 \int_0^\pi \sin \theta \frac{\epsilon \rho R \cos \theta}{\rho^2 + \lambda^2} d\theta d\rho \int_0^{2\pi} d\phi$$

The angular integration present no difficulty, letting us with the $\rho$ integral:

$$G(\mathbf{R}) = \frac{1}{2\pi^2 R} \int_0^\infty \frac{\rho}{\rho^2 + \lambda^2} \frac{e^{i\rho R} - e^{-i\rho R}}{2i} d\rho$$

And the beauty of holomorph analysis to gives us:

$$G(\mathbf{R}) = \frac{1}{4\pi^2 R} \int_{-\infty}^{\infty} \frac{i\rho}{\rho^2 + \lambda^2} e^{i\rho R} d\rho = \frac{1}{4\pi^2 R} \left[ 2\pi i \text{Res} \left( \frac{i\rho e^{i\rho R}}{(\rho + i\lambda)(\rho - i\lambda)}, i\lambda \right) \right] = e^{-\lambda R}$$

with the use of contour techniques around the pole $\rho = i\lambda$ and the residue theorem.

Let’s summarize the previous steps and write the final results for the particular case of interest where $\lambda = 0$, corresponding to the solution of Poisson’s equation:

$$G(\mathbf{R} - \mathbf{R}') = \frac{1}{(2\pi)^3} \int \frac{1}{k^2} e^{i\mathbf{k} \cdot (\mathbf{r} - \mathbf{r}')} d^3 k = \frac{1}{4\pi |\mathbf{r} - \mathbf{r}'|}$$

$$\phi(\mathbf{r}) = \frac{1}{\epsilon_0} \int_{\Omega} \rho(\mathbf{r}') G(\mathbf{r} - \mathbf{r}') d\mathbf{r}' = \frac{1}{\epsilon_0 \rho} G = \frac{1}{4\pi\epsilon_0} \int_{\Omega} \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}'$$

The expression of $\phi$ as the convolution between $\rho$ and $G$ is an interesting results for implementing Poisson’s solver algorithm.
Part III

On the benchmarking of the space charge algorithm
# Table of Contents

## 6 Benchmarking of the stand alone space charge algorithm without rotation  
6.1 Fields for a cylindrical distribution ............................................. 33  
   6.1.1 Theoretical expression of the potential ........................................ 33  
   6.1.2 Theoretical expression of the Fields for the cylinder ....................... 34  
   6.1.3 Graphical comparisons .......................................................... 34  
6.2 Fields for an ellipsoidal distribution ........................................... 36  
   6.2.1 Theoretical expression of the potential for a spheroid ..................... 36  
   6.2.2 Theoretical expression of the Fields for a spheroid and an ellipsoid .... 37  
   6.2.3 Graphical comparisons of the fields .......................................... 39

## 7 Rms field values - Simulation vs. theory  
7.1 On the RMS definition ............................................................... 41  
7.2 Spheroidal distribution .............................................................. 41  
7.3 Ellipsoidal distribution .............................................................. 44  
7.4 Cylindrical distribution .............................................................. 46  
7.5 Results summary ...................................................................... 49  
   7.5.1 Fields calculation ............................................................... 49  
   7.5.2 RMS calculation ................................................................. 50

## 8 Benchmarking of the stand alone space charge algorithm with rotation  
8.1 Testing method ...................................................................... 52  
   8.1.1 Principle ....................................................................... 52  
   8.1.2 Method used ................................................................. 52  
   8.1.3 Technical details ............................................................ 54  
8.2 Overview of different comparisons performed .................................... 54  
   8.2.1 Exemple of a coin-shape cylinder ........................................... 54  
   8.2.2 Exemple of an ellipsoid at high energy ..................................... 55

## 9 Including our algorithm in Astra - Modifications done and problem encountered  
9.1 Astra conventions ................................................................. 57  
9.2 Organizations of the source files ................................................. 57  
9.3 Learning from my mistakes ......................................................... 58  
9.4 Note on the sigma matrix outputed by Astra .................................... 58  
   9.4.1 Analysing the previous algorithm without the source code ............... 58  
   9.4.2 Modifying the source code to output a standard sigma matrix ........... 61
Benchmarking of the stand alone space charge algorithm without rotation

In order to validate the stand alone space charge algorithm, comparisons have been done with theoretical expressions of space charge fields. For cylindrical and ellipsoidal distributions, analytical expressions of the electric fields are available in the absence of external forces and when all particles are at rest. Comparisons with other algorithms have also been considered, but confronted to a short period of time and satisfied by the good agreement of our results with the theory, it has been eventually rejected to focus on the simulation of beam line components. On top of all the graphical comparisons made for all the fields components, we focused our comparisons on the longitudinal field $E_z$ and reported our results in this chapter. We will first derive the expression of the theoretical potentials and fields for a cylindrical distribution, in order to graphically compare the resulting fields for different values of the bunch size. In a second time, the same thing is done for an ellipsoidal distribution. Eventually, the calculation of the RMS value of the longitudinal fields on all the particles is done numerically and analytically for further comparisons with previous work of H. Edwards and Y-E Sun[12].

6.1 Fields for a cylindrical distribution

6.1.1 Theoretical expression of the potential

Given a homogeneous distribution of charges in a cylindrical volume with revolution symmetry around the longitudinal axis $Oz$, the potential on any point $z_0$ of this axis can be expressed using the Biot and Savart law:

$$
\Phi(0, z_0) = \frac{\rho}{4\pi\epsilon_0} \int \int \frac{2\pi r}{r^2 + (z - z_0)^2} \, dr \, dz
$$

(6.1)

Where $\rho$ is the density of charges. For a cylinder of semi-axis $b$ and radius $a$ with $N$ charges we have:

$$
\rho = \frac{Ne}{2\pi a^2}
$$

(6.2)

We first integrate equation 6.1 along $r$, between $r = 0$ and $r = a$, and then do the change of variable $Z = z_0 - z$:

$$
\Phi(0, z_0) = \frac{\rho}{2\epsilon_0} \int_{z_0 - b}^{z_0 + b} \left( \sqrt{a^2 + Z^2} - |Z| \right) dZ
$$

(6.3)

The integration of the second term is:

$$
\int_{z_0 - b}^{z_0 + b} |Z| dZ = \left\{ \begin{array}{ll}
    z_0^2 + b^2 & |z_0| \leq b \\
    2z_0b & |z_0| \geq b
  \end{array} \right.
$$

(6.4)

(6.5)
The first part is calculated either by blindly applying a formula we provide in annex F or by simply writing \( \sqrt{1 + z^2} = 1 + z^2 / \sqrt{1 + z^2} \), leading to so a sum of two integrals, where the integration by part of the second one leads to the original integral:

\[
\int \sqrt{1 + z^2} = \left[ \text{arg sinh} \ z \right] + \left[ z \sqrt{1 + z^2} \right] - \int \sqrt{1 + z^2} \quad (6.6)
\]

Eventually, for a cylindrical distribution, the potential inside (\(|z_0| < b\), subscript “int”), and outside (\(|z_0| > b\), subscript “ext”) the distribution are:

\[
\Phi(0, 0, z_0)_{\text{int}} = \Phi_0 + \frac{\rho}{2\epsilon_0} \left\{ - \left( \frac{z_0^2 + b^2}{a} \right) + \frac{1}{2} \left[ a^2 \left( \text{arg sinh} \ \frac{z_0 + b}{a} - \text{arg sinh} \ \frac{z_0 - b}{a} \right) \right] + (z_0 + b) \sqrt{(z_0 + b)^2 + a^2} - (z_0 - b) \sqrt{(z_0 - b)^2 + a^2} \right\} \quad (6.7)
\]

\[
\Phi(0, 0, z_0)_{\text{ext}} = \Phi_0 + \frac{\rho}{2\epsilon_0} \left\{ -2b z_0 + \frac{1}{2} \left[ a^2 \left( \text{arg sinh} \ \frac{z_0 + b}{a} - \text{arg sinh} \ \frac{z_0 - b}{a} \right) \right] + (z_0 + b) \sqrt{(z_0 + b)^2 + a^2} - (z_0 - b) \sqrt{(z_0 - b)^2 + a^2} \right\} \quad (6.8)
\]

The function \( \text{arg sinh} \) is the inverse function of \( \sinh \), the hyperbolic sine.

### 6.1.2 Theoretical expression of the Fields for the cylinder

From the previous expression of the potential, the field component \( E_z = -\frac{\partial \Phi}{\partial z} \) on the axis can be expressed:

\[
E_{z,\text{int}}(0, z_0) = -\frac{\rho}{2\epsilon_0} \left[ -2z_0 + \frac{1}{2} \left( \frac{a^2 + 2(z_0 + b)^2}{\sqrt{(z_0 + b)^2 + a^2}} - \frac{a^2 + 2(z_0 - b)^2}{\sqrt{(z_0 - b)^2 + a^2}} \right) \right] \quad (6.9)
\]

\[
E_{z,\text{ext}}(0, z_0) = -\frac{\rho}{2\epsilon_0} \left[ -2b \text{sign} (z_0) + \frac{1}{2} \left( \frac{a^2 + 2(z_0 + b)^2}{\sqrt{(z_0 + b)^2 + a^2}} - \frac{a^2 + 2(z_0 - b)^2}{\sqrt{(z_0 - b)^2 + a^2}} \right) \right] \quad (6.10)
\]

### 6.1.3 Graphical comparisons

On figure 6.1 one can see the good agreement between the numerical (solid colored line) and the theoretical (dashed lines) fields for different values of \( \sigma_z \), the RMS value of the longitudinal bunch half-length, and \( \sigma_r \), the rms value of the radius. Table 6.1 display the parameters used for the simulation.
CHAPTER 6. BENCHMARKING OF THE STAND ALONE SPACE CHARGE ALGORITHM WITHOUT ROTATION

Figure 6.1: Cylindrical distribution: Longitudinal field $E_z$ along $z$ - comparison between simulation and theory. The theoretical fields are represented with black dashed lines. A perfect agreement is found, but the maximum amplitude of the simulated field is always smaller. This depends on the resolution of the grid, and the smoothing of the fields.
CHAPTER 6. BENCHMARKING OF THE ST AND ALONE SPACE CHARGE ALGORITHM WITHOUT ROTATION

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution type</td>
<td>Uniform cylinder</td>
</tr>
<tr>
<td>Distribution $\sigma_x$</td>
<td>${1; 2; 3; 4}/\sqrt{2}$ mm</td>
</tr>
<tr>
<td>Distribution $\sigma_y$</td>
<td>${1; 2; 3; 4}/\sqrt{2}$ mm</td>
</tr>
<tr>
<td>Distribution $\sigma_z$</td>
<td>${0.1; 0.2; 0.3; 0.4}$ mm</td>
</tr>
<tr>
<td>Distribution energy</td>
<td>15 MeV</td>
</tr>
<tr>
<td>Distribution charge</td>
<td>-0.25 nC</td>
</tr>
<tr>
<td>3D grid used</td>
<td>$N_{x0}=17, N_{xf}=64, N_{z0}=128$</td>
</tr>
</tbody>
</table>

Table 6.1: Parameters used for the benchmarking of the space charge algorithm - cylinder distribution

6.2 Fields for an ellipsoidal distribution

We will call spheroid an ellipsoidal distribution with radial symmetry.

6.2.1 Theoretical expression of the potential for a spheroid

Like in the previous section, the Biot and Savart law can be integrated to obtain a theoretical expression of the potentials and fields for an ellipsoidal distribution. Given a homogeneous distribution of charges in an ellipsoidal volume with revolution symmetry around the longitudinal axis $Oz$, the potential on any point $z_0$ of this axis can be expressed using the Biot and Savart law:

$$\Phi(0, z_0) = \frac{\rho}{4\pi\varepsilon_0} \int r \int \frac{2\pi r}{r^2 + (z - z_0)^2} \, dr \, dz$$  \hspace{0.5cm} (6.11)

Where $\rho$ is the density of charges. For an ellipse of semi-axes lengths $A, B, C$ we have:

$$\rho = N e \left(\frac{4}{3}\pi ABC\right)^{-1}$$  \hspace{0.5cm} (6.12)

The integration on all charges is reduced to the domain defined by the plane ellipse driven by the equation:

$$\frac{r^2}{a^2} + \frac{z^2}{b^2} = 1$$  \hspace{0.5cm} (6.13)

This defines the boundaries used for the integration. We first integrate along $r$, between $r = 0$ and $r^2 = a^2(1 - z^2/b^2)$:

$$\Phi(0, z_0) = \frac{\rho}{2\varepsilon_0} \int_{-b}^{b} \left( \sqrt{a^2 - \frac{a^2 z^2}{b^2} - (z_0 - z)^2} - \frac{z_0 - z}{\sqrt{\frac{b^2}{z^2} - 1}} \arg coth \frac{b}{a} \right) \, dz$$  \hspace{0.5cm} (6.14)

The integration along $z$ requires the distinction between the case $a > b$ and $a < b$. Moreover, one has to distinguish the case $|z_0| < b$, for the potential inside the bunch, and $|z_0| > b$ for the external potential. The integration is a little bit more complicated than the one for the cylindrical case. One can refer to Annex F for some mathematical refreshments. After the integration, the fields are generalized[28] to points off axis but inside the distribution. For $a < b$ and $|z_0| < b$ the potential in the spheroid is:

$$\Phi(x_0, y_0, z_0)_{int} = \Phi_0 - \frac{\rho}{2\varepsilon_0} \left[ \frac{x_0^2 + y_0^2}{2} + \frac{z_0^2}{b^2 - a^2} - \frac{x_0^2 + y_0^2}{2} \left( 1 - \frac{b}{\sqrt{b^2 - a^2}} \arg coth \frac{b}{a} \right) \right]$$  \hspace{0.5cm} (6.15)

While the external potential $|z_0| > b$ on the axis would be:

$$\Phi(0, z_0)_{ext} = \frac{\rho}{2\varepsilon_0} a^2 b \left[ \frac{|z_0|}{b^2 - a^2} + \left( 1 - \frac{z_0^2}{b^2 - a^2} \right) \frac{1}{\sqrt{b^2 - a^2}} \arg coth \frac{|z_0|}{\sqrt{b^2 - a^2}} \right]$$  \hspace{0.5cm} (6.16)
Now, for an ellipsoid of dimensions such that $a > b$, the potential is:

$$
\Phi(x_0, y_0, z_0)_{\text{int}} = \Phi_0 - \frac{\rho}{2\varepsilon_0} \left[ \frac{x_0^2 + y_0^2}{2} + \frac{z_0^2 - \frac{x_0^2 + y_0^2}{2}}{a^2} \left( 1 - \frac{b}{a} \frac{\arccos \frac{b}{a}}{\sqrt{1 - \frac{b^2}{a^2}}} \right) \right]
$$

(6.17)

$$
\Phi(0, z_0)_{\text{ext}} = \frac{\rho}{2\varepsilon_0} a^2 b \left[ \frac{|z_0|}{a^2 - b^2} + \left( 1 + \frac{z_0^2}{a^2 - b^2} \right) \frac{1}{\sqrt{a^2 - b^2}} \frac{\text{arccotg} \frac{|z_0|}{\sqrt{a^2 - b^2}}}{\sqrt{a^2 - b^2}} \right]
$$

(6.18)

The functions $\text{arccosh}$ and $\text{arccoth}$ are the respective inverse functions of $\cosh$ and $\coth$, the hyperbolic cosine and cotangent. The generalization of the external fields is more complicated and we refer the reader to [28].

### 6.2.2 Theoretical expression of the Fields for a spheroid and an ellipsoid

#### Analytical derivation of the potential for a spheroid

From the previous expression of the internal potential, it can be immediately seen that a linear dependence is expected between each field component and its corresponding axes.

$$\overrightarrow{E}_{\text{int}} = -\nabla(\Phi) = \frac{\rho}{\varepsilon_0} (M_x x, M_y y, M_z z)
$$

(6.19)

With the symmetry of the distribution imposing $M_x = M_y$. Deriving equation 6.15 for the case $a < b$ yields to:

$$
M_z = -\frac{1}{b^2} \frac{a^2}{a^2 - b^2} \left( 1 - \frac{b}{a} \frac{\text{arccosh} \frac{b}{a}}{\sqrt{b^2 - a^2}} \right)
$$

(6.20)

$$
M_x = M_y = \frac{1}{2} (1 - M_z)
$$

(6.21)

While for the case $a > b$:

$$
M_z = -\frac{1}{b^2} \frac{a^2}{a^2 - b^2} \left( 1 - \frac{b}{a} \frac{\arccos \frac{b}{a}}{\sqrt{1 - \frac{b^2}{a^2}}} \right)
$$

(6.22)

In [30], referring [22], the following expressions can be found for $M_z$:

$$
M_z = \frac{1 + \Gamma}{1 - \frac{\sqrt{A^2/C^2 - 1}}{\Gamma^3}} \Gamma
$$

(6.23)

with the excentricity $\Gamma = \sqrt{A^2/C^2 - 1}$. Nevertheless, no link between 6.22 and 6.23 has not been done yet.

The expression of the external field being more complicated we will focus on its expression on the ellipse axis ($r = 0$) and its longitudinal component. For the case $a < b$:

$$
E_{z,\text{ext}}(0, z_0) = -\frac{\rho}{2\varepsilon_0} a^2 b \left[ \frac{\text{sign}(z_0)}{b^2 - a^2} + \frac{1}{\sqrt{b^2 - a^2}} \left( \frac{-2z_0}{b^2 - a^2} \frac{\text{arg coth} \frac{|z_0|}{\sqrt{b^2 - a^2}}}{\sqrt{b^2 - a^2}} \right) + \frac{\text{sign}(z_0)}{\sqrt{b^2 - a^2}} \right]
$$

(6.24)

And for $a > b$:

$$
E_{z,\text{ext}}(0, z_0) = -\frac{\rho}{2\varepsilon_0} a^2 b \left[ \frac{\text{sign}(z_0)}{a^2 - b^2} + \frac{1}{\sqrt{a^2 - b^2}} \left( \frac{2z_0}{a^2 - b^2} \text{arccotg} \frac{|z_0|}{\sqrt{a^2 - b^2}} - \frac{\text{sign}(z_0)}{\sqrt{a^2 - b^2}} \right) \right]
$$

(6.25)
General ellipsoid - Introducing a form factor

The linearity of the fields with respect to their axis is still valid in the general case of an ellipsoid. Approximate expressions have been derived by Lapostolle[28] and are used by the software trace 3D[8] to compute the space charge fields. For an ellipsoid of semiaxes \( r_x, r_y, r_z \), the following expressions are given:

\[
E_x(x) = \frac{1}{4\pi\epsilon_0} \frac{3I\lambda}{c\gamma^2} \frac{1 - f}{r_x(r_x + r_y)r_z} x \\
E_y(y) = \frac{1}{4\pi\epsilon_0} \frac{3I\lambda}{c\gamma^2} \frac{1 - f}{r_y(r_x + r_y)r_z} y \\
E_z(z) = \frac{1}{4\pi\epsilon_0} \frac{3I\lambda}{c} \frac{f}{r_xr_yr_z} z
\]  

(6.26)  
(6.27)  
(6.28)

where \( f \) is a function of \( p = \gamma r_z/\sqrt{(r_xr_y)} \), \( \lambda \) the free space wave-length of the RF and \( I \) the average current assuming one bunch is emitted at each period of the RF. As the result of this, in the case of a single bunch, \( \lambda I = cQ \) and eventually equation 6.28 can be written:

\[
E_z(z) = \frac{\rho}{\epsilon_0} f(p) \frac{z}{p}
\]  

(6.29)

The fact that \( f \) is called the form factor appears then obvious.

Figure 6.2(a) gives reference values for the form factor and figure 6.2(b) represent these values spline-interpolated for more precision.

<table>
<thead>
<tr>
<th>( p )</th>
<th>( f )</th>
<th>( 1/p )</th>
<th>( f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.0500</td>
<td>0.9260</td>
<td>0.0500</td>
<td>0.0070</td>
</tr>
<tr>
<td>0.1000</td>
<td>0.8610</td>
<td>0.1000</td>
<td>0.0200</td>
</tr>
<tr>
<td>0.1500</td>
<td>0.8030</td>
<td>0.1500</td>
<td>0.0370</td>
</tr>
<tr>
<td>0.2000</td>
<td>0.7500</td>
<td>0.2000</td>
<td>0.0560</td>
</tr>
<tr>
<td>0.2500</td>
<td>0.7040</td>
<td>0.2500</td>
<td>0.0750</td>
</tr>
<tr>
<td>0.3000</td>
<td>0.6610</td>
<td>0.3000</td>
<td>0.0950</td>
</tr>
<tr>
<td>0.3500</td>
<td>0.6230</td>
<td>0.3500</td>
<td>0.1150</td>
</tr>
<tr>
<td>0.4000</td>
<td>0.5880</td>
<td>0.4000</td>
<td>0.1350</td>
</tr>
<tr>
<td>0.4500</td>
<td>0.5560</td>
<td>0.4500</td>
<td>0.1550</td>
</tr>
<tr>
<td>0.5000</td>
<td>0.5270</td>
<td>0.5000</td>
<td>0.1740</td>
</tr>
<tr>
<td>0.5500</td>
<td>0.5000</td>
<td>0.5500</td>
<td>0.1920</td>
</tr>
<tr>
<td>0.6000</td>
<td>0.4760</td>
<td>0.6000</td>
<td>0.2100</td>
</tr>
<tr>
<td>0.6500</td>
<td>0.4530</td>
<td>0.6500</td>
<td>0.2270</td>
</tr>
<tr>
<td>0.7000</td>
<td>0.4320</td>
<td>0.7000</td>
<td>0.2440</td>
</tr>
<tr>
<td>0.7500</td>
<td>0.4130</td>
<td>0.7500</td>
<td>0.2600</td>
</tr>
<tr>
<td>0.8000</td>
<td>0.3940</td>
<td>0.8000</td>
<td>0.2760</td>
</tr>
<tr>
<td>0.8500</td>
<td>0.3780</td>
<td>0.8500</td>
<td>0.2910</td>
</tr>
<tr>
<td>0.9000</td>
<td>0.3620</td>
<td>0.9000</td>
<td>0.3060</td>
</tr>
<tr>
<td>0.9500</td>
<td>0.3470</td>
<td>0.9500</td>
<td>0.3200</td>
</tr>
<tr>
<td>1.0000</td>
<td>0.3330</td>
<td>1.0000</td>
<td>0.3330</td>
</tr>
</tbody>
</table>

(a)

Figure 6.2: Form factor data and corresponding plot for different value of \( p \)
6.2.3 Graphical comparisons of the fields

On figure 6.3 one can see the good agreement between the numerical (solid colored line) and the theoretical (dashed lines) fields for different values of $\sigma_z$, the RMS value of the bunch length, and $\sigma_r$, the rms value of the radius. Table 6.2 display the parameters used for the simulation.

![Graphical comparisons of the fields](image)

Figure 6.3: Ellipsoidal distribution: Longitudinal field $E_z$ along $z$ - comparison between simulation and theory. The theoretical fields are represented with black dashed lines. A perfect agreement is found, but the maximum amplitude of the simulated field is always smaller. This depends on the resolution of the grid, and the smoothing of the fields.
### Table 6.2: Parameters used for the benchmarking of the space charge algorithm - spheroid distribution

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution type</td>
<td>Uniform ellipsoid</td>
</tr>
<tr>
<td>Distribution $\sigma_x$</td>
<td>${1; 2; 3; 4} /\sqrt{2}$ mm</td>
</tr>
<tr>
<td>Distribution $\sigma_y$</td>
<td>${1; 2; 3; 4} /\sqrt{2}$ mm</td>
</tr>
<tr>
<td>Distribution $\sigma_z$</td>
<td>${0.1; 0.2; 0.3; 0.4}$ mm</td>
</tr>
<tr>
<td>Distribution energy</td>
<td>15 MeV</td>
</tr>
<tr>
<td>Distribution charge</td>
<td>-0.25 nC</td>
</tr>
<tr>
<td>3D grid used</td>
<td>$N_{*0} = 17, N_{*f} = 64, N_{*2} = 128$</td>
</tr>
</tbody>
</table>

Table 6.2: Parameters used for the benchmarking of the space charge algorithm - spheroid distribution

---

### Table 6.3: Parameters used for the benchmarking of the space charge algorithm - sphere at two energies

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution type</td>
<td>Uniform ellipsoid</td>
</tr>
<tr>
<td>Distribution $\sigma_x$</td>
<td>1 mm</td>
</tr>
<tr>
<td>Distribution $\sigma_y$</td>
<td>1 mm</td>
</tr>
<tr>
<td>Distribution $\sigma_z$</td>
<td>1 mm</td>
</tr>
<tr>
<td>Distribution energy</td>
<td>${0; 100}$ MeV</td>
</tr>
<tr>
<td>Distribution charge</td>
<td>-100 nC</td>
</tr>
<tr>
<td>3D grid used</td>
<td>$N_{*0} = 5, N_{*f} = 20, N_{*2} = 32$</td>
</tr>
</tbody>
</table>

Table 6.3: Parameters used for the benchmarking of the space charge algorithm - sphere at two energies

---

Figure 6.4: Comparison of theoretical (dashed) and numerical longitudinal field for an ellipsoidal bunch at two different energies: a) 0 MeV, b) 100 MeV - Values calculated on the grid are displayed with green cross. Note that a small amount of cells has been used here.
Rms field values - Simulation vs. theory

7.1 On the RMS definition

We recall here some notations we introduced at the beginning of this document. We define the rms value of the longitudinal electrical field as:

$$\sigma_{E_z} = \sqrt{\langle E_z^2 \rangle_n - \langle E_z \rangle_n^2}$$

(7.1)

where the brackets stands for the expected value, and the subscript $n$, stands for normalized. If $f$ is a normalized probability density function:

$$\langle g \rangle_n = \int_{\Omega} g f$$

(7.2)

If this function is not normalised, we will normalized it as follows:

$$\langle g \rangle_n = \frac{\langle g \rangle}{\langle 1 \rangle}$$

(7.3)

In this section we will only deal with centered statistical variables, so that $\langle g \rangle_n = 0$ and thus the standard deviation reduces to:

$$\sigma^2_g = \langle g^2 \rangle_n$$

(7.4)

One has to pay attention, because this is not a general case.

7.2 Spheroidal distribution

RMS calculation of the theoretical longitudinal field We have seen in the previous section that the fields inside the ellipsoid are linear with respect to their axis. We will write equation 6.28 as:

$$E_z(z) = -\frac{\alpha}{X_0 Y_0 Z_0} z$$

(7.5)

without expliciting the constants hidden behind the positive constant $\alpha$. The maximum of the longitudinal field in the distribution is:

$$E_{\text{max}} = E_z(z = Z_0) = \frac{\alpha}{X_0 Y_0}$$

(7.6)

Then we can write the field in function of its maximum:

$$E_z(z) = -\frac{E_{\text{max}}}{Z_0} z$$

(7.7)
CHAPTER 7. RMS FIELD VALUES - SIMULATION VS. THEORY

By definition
\[ \langle E_z^2 \rangle = \int_{x,y,z} E_z^2 f(x,y,z) dx dy dz \]  
(7.8)

Projecting the density distribution function of the ellipsoid on the z-axis:
\[ \langle E_z^2 \rangle = \frac{E_{\text{max}}^2}{Z_0} \int_z z^2 f(z) dz = \frac{E_{\text{max}}^2}{Z_0^2} \langle z^2 \rangle \]  
(7.9)

where the projection of the density function of the distribution of a uniform ellipsoid is:
\[ f(z) = \int_{x,y} f(x,y,z) dx dy = \left[ 1 - \left( \frac{z}{Z_0} \right)^2 \right] \]  
A  
(7.10)

The constant A arbitrary chosen at 1. Let’s present some intermediate steps of calculus:
\[ N = \int_{-Z_0}^{Z_0} f(z) dz = \frac{4}{3} Z_0 \]  
(7.11)

\[ \langle z \rangle = \int_{-Z_0}^{Z_0} z f(z) dz = 0 \]  
(7.12)

\[ \langle z^2 \rangle = \int_{-Z_0}^{Z_0} z^2 f(z) dz = \frac{4}{15} Z_0^2 \]  
(7.13)

The rms value of z is then
\[ \langle z^2 \rangle_n = \sigma_z^2 = \frac{\langle z^2 \rangle}{N} = \frac{Z_0^2}{5} \]  
(7.14)

And
\[ \sigma_{E_z}^2 = \frac{E_{\text{max}}^2}{Z_0^2} \sigma_z^2 = \frac{E_{\text{max}}^2}{5} \]  
(7.15)

**RMS calculation of the radius** : We will study the RMS value of \( E_z \) for different radius. We will then use RMS value of the radius. As it is a spheroid we could simply write \( r^2 = x^2 + y^2 = 2x^2 \) and thus:
\[ \langle r^2 \rangle = 2 \langle x^2 \rangle \]  
(7.16)

But let’s be convinced that we are right. Tha calculation is done for the section \( z = 0 \). We write \( x = r \cos \theta \), and calculate the not-normalized rms value of x:
\[ \langle x^2 \rangle = \int_0^{R_0} \int_0^{2\pi} x^2 r dr d\theta = \frac{R_0^4}{4} \int_0^{2\pi} \cos^2 \theta = \frac{R_0^4 \pi}{4} \]  
(7.17)

The normalization being:
\[ \langle 1 \rangle = \pi R^2 \]  
(7.18)

we end up with
\[ \langle x^2 \rangle_n = \frac{R_0^2}{4} \]  
(7.19)

On the other end we have
\[ \langle r^2 \rangle_n = \frac{2\pi R_0^4}{\pi R_0^2} = \frac{R_0^2}{2} \]  
(7.20)

So eventually we have:
\[ \langle r^2 \rangle_n = 2 \langle x^2 \rangle_n \]  
(7.21)

By symmetry, from the calculation of \( \langle z^2 \rangle_n \) we have:
\[ \langle x^2 \rangle_n = \frac{X_0^2}{5} \]  
(7.22)

and thus, as \( X_0 = R_0 \):
\[ \langle r^2 \rangle_n = \frac{2}{5} R_0 \]  
(7.23)
RMS calculation of the simulated field: Two different methods were used for the RMS calculation of the simulated field. The first one, is labelled \textit{(sim.)} in the legend of figure 7.1, and the second one \textit{(sim. from max)}. Going back to the definition of the rms value, the first method consists in calculating the rms value from the field value on each particle. For this, we added a little “FOR” loop at the end of the main of our stand alone algorithm: \textit{astraimpulse\_main.f}. For each component of the two fields, the rms value is calculated as follow:

\[
E_{z,rms} = \frac{\sum_{i=0}^{N_{part}-1} E_{z,i}(x_i, y_i, z_i)^2}{N_{part}}
\]  

(7.24)

where \(N_{part}\) is the number of particle, \(x_i, y_i, z_i\), its coordinates and \(E_{z,i}\) has been calculated with the function \texttt{SCGridInterpolate}. The 6 RMS values are stored in an output file called \textit{RMSFields.out}.

The second method used is based on the theoretical rms value calculated above. Knowing the maximum of the simulated longitudinal field on the z-axis, dividing it by \(\sqrt{5}\) provides a result comparable to the theoretical RMS value. To know the longitudinal field on the z-axis, we will use the file \textit{E\_DIR.out}, which is outputed by our main program. This functionality will be developed in details in section 8.1.3.

Comparison between theory and simulation

![Graph showing comparison between theory and simulation](image)

Figure 7.1: Rms value of the longitudinal field for a spheroidal distribution. The theoretical value is calculated with the formula developed in the previous section, taking \(E_{z,rms} = E_{z,max}/\sqrt{5}\). Calculating the simulated rms value \textit{(sim.)} by averaging the longitudinal field on each particle, provides a really good agreement with the theory. From the maximum of the simulated field, another RMS value can be defined by dividing it by \(\sqrt{5}\). Nevertheless, we previously saw that the simulated field has a smaller maximum. This explains why the values \textit{sim. from max} are smaller. Data points are stored in file \textit{/results/ez-rms/MErmsEllipse.csv}
Figure 7.2: Maximum value of the longitudinal field for a spheroidal distribution. Once again we see that the simulated field has an amplitude smaller than the theoretical one. Note that the fields are here plotted with the real length of the bunch and the real radius of the bunch, instead of RMS values.

### 7.3 Ellipsoidal distribution

For an ellipsoidal distribution, we didn’t derived manually the fields expressions, and we will use equations 6.26-6.28, that uses a form factor, function of the dimensions of the ellipsoid $r_x$, $r_y$, $r_z$. For the comparison between this theory and our simulations, we generated different bunches of different dimensions. All permutations of values of $\sigma_x$, $\sigma_y$, $\sigma_z$ presented in table 7.1 were used, representing 896 distributions.

<table>
<thead>
<tr>
<th>$\sigma_x$</th>
<th>0.50</th>
<th>1.00</th>
<th>1.50</th>
<th>2.00</th>
<th>2.50</th>
<th>3.00</th>
<th>3.50</th>
<th>4.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_y$</td>
<td>0.20</td>
<td>0.40</td>
<td>0.60</td>
<td>0.80</td>
<td>1.00</td>
<td>1.20</td>
<td>1.40</td>
<td>1.60</td>
</tr>
<tr>
<td>$\sigma_z$</td>
<td>0.05</td>
<td>0.10</td>
<td>0.15</td>
<td>0.20</td>
<td>0.25</td>
<td>0.30</td>
<td>0.35</td>
<td>0.40</td>
</tr>
</tbody>
</table>

Table 7.1: RMS dimensions used for the study of ellipsoidal distributions

The difference between $\sigma_x$ and $\sigma_y$ is anecdotal. As a symmetry was expected between x and y, it appeared unnecessary to use the same values for $x$ and $y$. We chose to use values close to the expected range that can be found at $a_0$: $\sigma_x \in 0.5 - 4.0$, $\sigma_y \in 0.2 - 2.0$. But after all, we decided to study the symmetry between $x$ and $y$, so we extended the values of $\sigma_y$.

Our stand alone algorithm has been run on all this distributions, for a grid of $64 \times 64 \times 64$. Maximum fields values and rms values have been extracted from the simulation. Combined with theoretical values, a database has been established. Its structure is described in table 7.2.

August 2009

Emmanuel Branlard
In Table 7.2, DEX stands for the relative difference:

$$E_{x,\text{rms}}^{\text{th}} - DEX = \frac{E_{x,\text{rms}}^{\text{sim}}}{E_{x,\text{rms}}^{\text{th}}}$$  (7.25)

As a result of this DEX, DEY, DEZ translate the percentage of difference between the simulation and the theory. We suggest here, to look at the influence of $\sigma_x$ and $\sigma_y$ on this three parameters, for a given $\sigma_z = 0.1\text{mm}$. These results are plotted on figure 7.3. The best agreement is for $\sigma_x \approx \sigma_y$. One explanation can be that the theory developed by Lapostolle as an extension of the case $\sigma_x = \sigma_y$, is less valid if the bunch does not have revolution symmetry. Nevertheless, we don’t reject the hypothesis that the 3D Poisson solver has less validity in this situation also.
Figure 7.3: Relative difference between theory and simulation for different ellipsoids for the three components of the field. Light color represent a percentage of error of less than 5 %. The three figures show the relative difference between the theory and the simulation: (a) DEZ($E_z$), (b) DEX($E_x$), (c) DY($E_y$). The best agreement is for $\sigma_x \approx \sigma_y$.

7.4 Cylindrical distribution

**RMS calculation of $z$** For a uniformly distributed cylinder, $f$ is a constant not null in the bunch domain:

$$N = \int_{\Omega} f = \int_{x,y,z} c \, dxdydz = \int_{-z_0}^{z_0} Cdz = 2z_0C$$

(7.26)

N is the normalization factor so that $f/N$ is a normalized distribution.

$$\langle z^2 \rangle = \int_{x,y,z} z^2 f \, dxdydz = \int_{-z_0}^{z_0} z^2Cdz = \frac{2}{3}z_0^3C$$

(7.27)

Eventually

$$\sigma_z^2 = \frac{\langle z^2 \rangle}{N} = \frac{z_0^2}{3}$$

(7.28)

Let’s call $l_z$ the bunch length, so that $z_0 = l_z/2$. The previous equation is now:

$$\sigma_z^2 = \frac{l_z^2}{12}$$

(7.29)

**Estimation of the RMS value for the longitudinal field**: Contrary to the ellipsoidal distribution, fields are not linear with respect to their axis. The theoretical RMS calculation is then more
complex. Option 1: If we assume (wrongly) that the longitudinal field is linear with respect to $z$, then (cf calculation for the ellipsoid in the previous section 7.2) we can define the RMS field as:

$$\langle E_z^2 \rangle \approx \frac{E_{z,\text{max}}^2 \langle z^2 \rangle}{r_z^2}$$

(7.30)

which, from the previous section would give us:

$$E_{z,\text{rms}} \approx \frac{E_{z,\text{max}}}{\sqrt{3}}$$

(7.31)

Option 2: From the simulation, calculating the RMS values by summing on all particles provides an RMS field that has an amplitude way smaller than the one suggested in equation 7.31. It appeared that the RMS value is more of the order of:

$$E_{z,\text{rms}} \approx \frac{E_{z,\text{max}}}{\sqrt{12}}$$

(7.32)

Why using a wrong definition? Indeed, this sound controversial. We just want an easy way to compare the theoretical and simulation fields amplitude, and as we are using rms values for $z$ and $r$, it makes more sense to use an rms value for the field also. Nevertheless, a non-controversial plot, with real values (not rms) will be also provided (see figure 7.5).

**RMS estimation of $E_z$** From the two previous paragraph, we eventually have

**RMS calculation of the simulated field**: the two same methods than for the ellipsoid are used (see equation 7.24 and its corresponding paragraph), eventhough the only trustable one is the one calculated by averaging on all the particles. This results will be obvious in the next plot (figure 7.4).
Comparison between theory and simulation

Figure 7.4: Rms value of the longitudinal field for a cylindrical distribution. What we call theory here, corresponds to the theoretical value of the field maximum divided by $\sqrt{12}$, $E_{z \text{rms}} = E_{z \text{th, max}}/\sqrt{12}$.

This empirical relation seems reasonable as it provides results of the same order than the one calculated by averaging the simulated longitudinal field on each particle (sim.). From the maximum of the simulated field, another RMS value can be defined by dividing it by $\sqrt{12}$. Nevertheless, we previously saw that the simulated field has a smaller maximum. This explains why the values sim. from max are smaller than the one calculated from the theoretical field. Data points are stored in file /results/ez-rms/MERmsCylinder.csv.
Figure 7.5: Maximum value of the longitudinal field for a cylindrical distribution. Once again we see that the simulated field has an amplitude smaller than the theoretical one. Note that the fields are here plotted with the real length of the bunch and the real radius of the bunch, instead of RMS values. Data points are stored in file /results/ez-rms/MEmaxCylinder.csv

7.5 Results summary

7.5.1 Fields calculation

For a uniform cylinder of radius $a$ and half length $b$:

$$E_{z,int}(0, z_0) = -\frac{\rho}{2\epsilon_0} \left[-2z_0 + \frac{1}{2} \left( \frac{a^2 + 2(z_0 + b)^2}{\sqrt{(z_0 + b)^2 + a^2}} - \frac{a^2 + 2(z_0 - b)^2}{\sqrt{(z_0 - b)^2 + a^2}} \right) \right]$$ (7.33)

$$E_{z,ext}(0, z_0) = -\frac{\rho}{2\epsilon_0} \left[-2b \text{sign}(z_0) + \frac{1}{2} \left( \frac{a^2 + 2(z_0 + b)^2}{\sqrt{(z_0 + b)^2 + a^2}} - \frac{a^2 + 2(z_0 - b)^2}{\sqrt{(z_0 - b)^2 + a^2}} \right) \right]$$ (7.34)
For a uniform spheroid of radius \( a \) and half length \( b \), the internal fields are:

For \( a < b \):

\[
E_{z,\text{int}}(0, z_0) = -\frac{\rho}{\epsilon_0} \frac{1}{b^2 - a^2} \left( 1 - \frac{b}{a} \arccosh \frac{b}{a} \right) z_0
\]

\[
E_{z,\text{ext}}(0, z_0) = -\frac{\rho}{\epsilon_0} \frac{1}{a^2} \left[ \text{sign}(z_0) \right] \frac{2z_0}{b^2 - a^2} \arg \coth \frac{|z_0|}{\sqrt{b^2 - a^2}} + \text{sign}(z_0) \frac{|z_0|}{\sqrt{b^2 - a^2}}
\]

For \( a > b \):

\[
E_{z,\text{int}}(0, z_0) = -\frac{\rho}{\epsilon_0} \frac{1}{b^2 - a^2} \left( 1 - \frac{b}{a} \arccos \frac{b}{a} \right) z_0
\]

\[
E_{z,\text{ext}}(0, z_0) = -\frac{\rho}{\epsilon_0} \frac{1}{b^2 - a^2} \left[ \text{sign}(z_0) \right] \frac{2z_0}{a^2 - b^2} \arg \cotg \frac{|z_0|}{\sqrt{a^2 - b^2}} - \text{sign}(z_0) \frac{|z_0|}{\sqrt{a^2 - b^2}}
\]

For a uniform ellipsoid of semiaxes \( r_x, r_y, r_z \), the internal fields are:

\[
E_{x,\text{int}}(x) = \frac{1}{4\pi\epsilon_0} \frac{3\lambda}{c^2} \frac{1 - f}{r_x(r_x + r_y + r_z)} x
\]

\[
E_{y,\text{int}}(y) = \frac{1}{4\pi\epsilon_0} \frac{3\lambda}{c^2} \frac{1 - f}{r_y(r_x + r_y + r_z)} y
\]

\[
E_{z,\text{int}}(z) = \frac{1}{4\pi\epsilon_0} \frac{3\lambda}{c} \frac{f}{r_xr_yr_z} z
\]

### 7.5.2 RMS calculation

RMS calculations performed above are simple calculus and integrations. It is good to keep their values in mind, but to be convinced one often do the calculation at each time the value is needed. Let’s write down the previous results in this section, so that we can refer to it later without doing the calculation again.

For a uniform cylinder \((x_0 = y_0 = r_0)\):

\[
x_{\text{rms}} = \sqrt{\langle x^2 \rangle} = \frac{1}{2} x_0
\]

\[
r_{\text{rms}} = \sqrt{\langle r^2 \rangle} = \frac{1}{\sqrt{2}} r_0 (= \sqrt{\frac{1}{2}} x_{\text{rms}})
\]

\[
z_{\text{rms}} = \sqrt{\langle z^2 \rangle} = \frac{1}{\sqrt{3}} z_0
\]

\[
z_{\text{rms}} = \sqrt{\langle z^2 \rangle} = \frac{1}{\sqrt{12}} l_z
\]
For a uniform spheroid ($x_0 = y_0 = r_0$):

\[
x_{\text{rms}} = \sqrt{\langle x^2 \rangle_n} = \frac{1}{\sqrt{5}} x_0
\]
(7.49)

\[
r_{\text{rms}} = \sqrt{\langle r^2 \rangle_n} = \frac{\sqrt{2}}{\sqrt{5}} r_0 \quad (= \sqrt{2} x_{\text{rms}})
\]
(7.50)

\[
z_{\text{rms}} = \sqrt{\langle z^2 \rangle_n} = \frac{1}{\sqrt{5}} z_0
\]
(7.51)

\[
E_{z,\text{rms}} = \sqrt{\langle E_z^2 \rangle_n} = \frac{E_{\text{max}}}{\sqrt{5}}
\]
(7.52)
8.1 Testing method

8.1.1 Principle

Once the space charge algorithm has been tested and validated, the rotation algorithm needed to be tested. For this, we used graphical comparisons. In order to validate our algorithm we want:

- the fields calculated for a non rotated distribution going in a direction parallel to the $z$-axis,
and
- the fields calculated for a non rotated distribution going in a direction parallel to the $z$-axis,
to be the same when the second ones are expressed in the proper rotated frame. Indeed, in the proper rotated frame, the second distribution is exactly the same than the first one, and the problem of solving Poisson equation is exactly the same.

8.1.2 Method used

We introduced a switch in our program allowing or not the rotation of the frame for the space charge calculation. Let’s call algo-norot the algorithm that calculate the space charge without trying to find an adapted frame, and algo-rot, the one that does. With the software $R$ ([7]) we developped a series of scripts to setup the comparison(rotating bunch, generating directions..), and do the post production(rotations and plotting of the fields) of the program output. The main of our stand alone algorithm, astraimpulse_main.f, has been modified to allow the output of fields values on specific points.

We used the following protocol: (figure 8.1 will help the understanding)

Step 1: calculating reference fields values for a non rotated distribution

- Generate a non rotated distribution

- Generate canonical directions (series of points aligned) with respect to the bunch dimensions on which the fields will be calculated. On figure 8.1, each direction is represented with a different color.

- Run algo-norot and output the fields value on these points. This will be the reference values of the space charge fields.

- Plot the fields for each directions
CHAPTER 8. BENCHMARKING OF THE ST AND ALONE SPACE CHARGE ALGORITHM WITH ROTATION

Step 2: calculating fields values for a rotated distribution

- Rotate the previous distribution (position and momentum multiplied by a rotation matrix $R$)
- Generate canonical directions for the rotated bunch. These directions correspond to the non-rotated bunch directions multiplied by $R$.
- Run algo-rot and output the fields value on these directions. Note that the fields returned are expressed in the laboratory frame.
- Plot the fields for each directions

Step 3: rotate the fields from lab frame to rotated frame

- The three Fields components from step 2 are rotated so that their components are expressed in the rotated frame and not in the laboratory frame.
- Plot the rotated fields for each directions. Compare them with the one obtained at step 1.

We will further present several situations of tests, and different fields plots comparing step1 step2 and step3. These comparisons were made for each field component, plotted on each directions, along each frame axis: $E_x$ along $x,y,z$ ... $B_z$ along $x,y,z$. We won’t display all of these fields components. Only one has been selected for each studied case. The reader will notice that the results from step3 and step1 are in each case identical. No other tests were performed, even though we are aware that a simple comparison of plots for different situations can not really be taken for a rigourous demonstration. Nevertheless, more than 30 canonical cases were studied, varying: axis of rotation, number of rotations, size, shape, and energy of the bunch. Each time, a perfect matching in shape and amplitude has been observed between step1 and step3 plots. From this, we can agree on the validity of the rotation algorithm.

Figure 8.1: Canonical directions on which the fields are plotted for the benchmarking of rotation algorithm. Positions of the lines are according to the dimension of the bunch $z/\sigma_z$ and $y/\sigma_r$. (a) Original bunch - (b) rotated bunch
8.1.3 Technical details

All the functions concerning the directions generation and the post-production of the results are located in the R script file `LabFieldsRotCheck.r`. The rotation of the bunch is part of the functions present in the file `bunch.r`.

**Rotations**: One has to be careful to rotate position and momentum. We tried to give explicit file names for the rotated distribution. If `ellipse.ini` is the original distribution file, and we rotate it from an angle of $\pi/4$ around the X axis, the rotated distribution will be: `ellipse-Xpi4.ini`

**Directions definition**: 15 directions are generated, each consisting of a line of $n=100$ points. The length of each line is equal to twice the length of the bunch in this direction. For a non-rotated bunch, the directions are parallel to the laboratory axis. Five directions parallel to X or Y are plotted for $z/\sigma_z \in \{-2;-1;0;1;2\}$. Five directions parallel to Z are plotted for $r/\sigma_r \in \{0;0.37;0.75;1.12;1.5\}$.

**Directions files**: Directions files are stored in the folder `/bunches/directions`, that’s where the fortran program will look for. The name of the file containing the directions has to be the name of the distribution, followed by `-directions`, for instance: `ellipse-Xpi4.ini-directions`. The file consist of three columns of coordinates X, Y, Z and $15 \times n$ lines. The $5 \times n$ first lines concern the directions parallel to X and so on for Y and Z.

**Program output**: The “main” of the stand alone algorithm output a file `E_DIR.out` which contains the three same first columns than the direction files, with the three following columns being the three components of the electrical field expressed in the laboratory frame.

8.2 Overview of different comparisons performed

Two examples of plots comparisons are presented here. Each time, three figures are presented, corresponding from left to right to step2, step3 and step1. As a result of this, the two right plots are expected to be the same, and one can happily notice that they are. The fields of step2 have most of the time a really different shape as they are expressed in the laboratory frame. Two more examples are presented in annex E. The 3D grid used consisted in a total number of grid of 20 in each direction ($N*f=20$), with 5 cells used for the boundary condition on each side ($N*0 = 5$). This provides reasonable results eventhough smoother curves would have been obtained with a higher number of cells.

8.2.1 Exemple of a coin-shape cylinder

The simulation was done with the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution type</td>
<td>Uniform cylinder</td>
</tr>
<tr>
<td>Distribution $\sigma_x$</td>
<td>3 mm</td>
</tr>
<tr>
<td>Distribution $\sigma_y$</td>
<td>3 mm</td>
</tr>
<tr>
<td>Distribution $\sigma_z$</td>
<td>0.1 mm</td>
</tr>
<tr>
<td>Distribution energy $E$</td>
<td>15 MeV</td>
</tr>
<tr>
<td>Distribution charge $Q$</td>
<td>-0.25 nC</td>
</tr>
<tr>
<td>Rotation axis</td>
<td>X</td>
</tr>
<tr>
<td>Rotation angle</td>
<td>$\pi/4$</td>
</tr>
<tr>
<td>3D grid used</td>
<td>$N<em>0 = 5$, $N</em>f=20$, $N*2=32$</td>
</tr>
</tbody>
</table>

Table 8.1: Parameters used for the benchmarking of the rotation algorithm - exemple of a coin-shape
8.2.2 Exemple of an ellipsoid at high energy

The simulation was done with the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution type</td>
<td>Uniform ellipsoid</td>
</tr>
<tr>
<td>Distribution $\sigma_x$</td>
<td>1 mm</td>
</tr>
<tr>
<td>Distribution $\sigma_y$</td>
<td>1 mm</td>
</tr>
<tr>
<td>Distribution $\sigma_z$</td>
<td>3 mm</td>
</tr>
<tr>
<td>Distribution energy $\mathcal{E}$</td>
<td>100 MeV</td>
</tr>
<tr>
<td>Distribution charge $Q$</td>
<td>-0.25 nC</td>
</tr>
<tr>
<td>Rotation axis</td>
<td>X</td>
</tr>
<tr>
<td>Rotation angle</td>
<td>$\pi/6$</td>
</tr>
<tr>
<td>3D grid used</td>
<td>$N<em>0 = 5$, $N</em>f = 20$, $N*2 = 32$</td>
</tr>
</tbody>
</table>

Table 8.2: Parameters used for the benchmarking of the rotation algorithm - exemple of an ellipsoid at high energy
Figure 8.4: Rotation algorithm - Longitudinal Field. (a) Before the rotation of the fields - (b) After the rotation of the fields - (c) Fields from a distribution not rotated
Including our algorithm in Astra - Modifications done and problem encountered

9.1 Astra conventions

By definition the momentum $\vec{p}$ is

$$\vec{p} = \gamma m_0 c \beta = \frac{\xi}{c} \vec{\beta}$$

(9.1)

Astra uses another convention, such that the momentum defined in Astra, further written $\vec{\tilde{p}}$, is defined as:

$$\vec{\tilde{p}} = \frac{\vec{p}}{m_0 c} = \gamma \vec{\beta}$$

(9.2)

As a result of this the calculation of $\gamma$ in Astra is performed by using the following relation:

$$\text{GAMMA} = \left[ \left( \frac{\vec{\tilde{p}}}{m_0 c} \right)^2 + 1 \right]^{1/2} = \left[ \gamma \beta^2 + 1 \right]^{1/2}$$

(9.3)

9.2 Organizations of the source files

- astraimpulse : stand alone program
- astraimpulse_main.f : stand alone main source file
- compile_astra : program to compile the stand alone algorithm
- covariance.f : source file for the covariance calculation
- diago_NxN.f : source file for the diagonalization of a $N \times N$ symmetric matrix.
- inverse.f : source file to invert a matrix
- spch_improve.f : source files containing all the new functions used for our algorithm
- X_em_mon.f : source file in which the output of the sigma matrix has been changed.
- X_modules_rot.f : module ROTBOOST used by my algorithm
- X_spch3d.f : astra space charge 3d source file modified to use our functions
9.3 Learning from my mistakes

- Generating of a distribution with a kinetic energy of 0 implies that the boost matrix is not defined (division by zero)
- SCGridInterpolate requires arguments relative to the bunch center, no absolute positions
- When estimating the min and max of the bunch, the loop must be initialized to the center of the bunch (boosted coordinates for instance)
- Watch out for the parameter Qbunch in Newrun, whatever the distribution, Qbunch is taken
- In the input distribution file, X and Y are not relative to the reference particle, but absolute values. If offset are required, one should input them in the run file. Idem for px and py.

9.4 Note on the sigma matrix outputed by Astra

The reader is invited to read chapter 3.2 for a better understanding on beam matrix and emittance. For convenience, we recall here some formulas:

\[ \epsilon_x = \sqrt{\langle x^2 \rangle \langle x'^2 \rangle - \langle xx' \rangle^2} \]  
\[ \epsilon_{nx} = \frac{1}{m_0 c} \sqrt{\langle x^2 \rangle (p_x^2) - \langle xp_x \rangle^2} \]  
\[ \epsilon_{nx} = \beta \gamma \epsilon_x \]  
\[ \Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix} = \begin{pmatrix} \langle x^2 \rangle & \langle xx' \rangle^2 \\ \langle xx' \rangle^2 & \langle x'^2 \rangle \end{pmatrix} = \epsilon \begin{pmatrix} \beta & -\alpha \\ -\alpha & \gamma \end{pmatrix} \]  
\[ \epsilon = \frac{A}{\pi} = ab \]

9.4.1 Analysing the previous algorithm without the source code

Philippe Piot added to Astra the functionnality of outputing the \(\Sigma\) matrix if the option \texttt{SigmaS}\ is specified in the \texttt{&NEWRUN} section of the run file. The structure of the output file is specified in table 9.1. The matrix being symmetric only 21(6 \(\times\) 6/2 + 6/2) coefficients need to be stored. The two first columns contain the \(z\) position and the energy. The following 21 are the matrix coefficients.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>...</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>...</th>
<th>22</th>
<th>23</th>
</tr>
</thead>
<tbody>
<tr>
<td>(z)</td>
<td>(\epsilon)</td>
<td>(\sigma_{11})</td>
<td>(\sigma_{12})</td>
<td>...</td>
<td>(\sigma_{16})</td>
<td>(\sigma_{22})</td>
<td>(\sigma_{23})</td>
<td>...</td>
<td>(\sigma_{56})</td>
<td>(\sigma_{66})</td>
</tr>
</tbody>
</table>

Table 9.1: Structure of the file containing the sigma matrix: run.Sigma.001

A sigma matrix can be written in different units, and attention has been paid to plot the ellipse with the right dimension and units on each axis. To check this, we calculated directly the sigma matrix from the distribution, and we compared it with the matrix outputed by Astra. We did this for two cylinders with with two different normalized transversal emittances : \(\epsilon_{nx} = 5\) mrad.mm and \(\epsilon_{nx} = 1\) mrad.mm. See table 9.2 for more information on the distributions.
The 2D $x$-block of the matrix outputed by astra were:

$$\Sigma_{x,\epsilon_n=1}^{\text{out}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \times 10^{-6} \quad \text{and} \quad \Sigma_{x,\epsilon_n=5}^{\text{out}} = \begin{pmatrix} 1 & 0 \\ 0 & 25 \end{pmatrix} \times 10^{-6} \quad (9.9)$$

One can easily notice that the square root of the determinant of these matrix provides the normalized emittance in [m.rad]. From the input distribution, the values $\sigma_x$, $\sigma_{p_x}$, $\sigma_{\tilde{p}_x}$ and $\sigma_{x'}$ have been calculated, remembering our notation for the momentum variable in Astra: $\tilde{p}_x = p_x/m_0c = \gamma\beta_x$. The units for these four values are respectively: [m], [eV/c], [ ] and [rad]. From the comparison we concluded that the matrix outputed by astra was the following:

$$\Sigma^{\text{out}}_x = \begin{pmatrix} \langle x^2 \rangle & \langle x\tilde{p}_x \rangle \\ \langle x\tilde{p}_x \rangle & \langle \tilde{p}_x^2 \rangle \end{pmatrix} \quad (9.10)$$

Nevertheless, the canonical variables in phase space are $x - p_x$ instead of $x - \tilde{p}_x$, and the most common set used is $x - x'$. In the first case, one simply goes from dimensionless units [ ] to [MeV/c] by using:

$$p_x [\text{MeV/c}] = (\tilde{p}_x [\text{ ]}) \times (m_0c [\text{MeV/c}]) \quad (9.11)$$

where the product $m_0c = 0.510998 \simeq 0.511$ [MeV/c]. And thus, the sigma matrix in the space $x - p_x$ can be calculated, and has the following terms:

$$\sigma_{12} = \langle xp_x \rangle \simeq 0.511\langle x\tilde{p}_x \rangle \quad (9.12)$$
$$\sigma_{22} = \langle \tilde{p}_x^2 \rangle \simeq 0.511^2 \langle \tilde{p}_x^2 \rangle \quad (9.13)$$

The corresponding ellipse is plotted with horizontal and vertical units being respectively [m] and [MeV/c]. For the phase space $x - x'$ we use the relation:

$$x' = \frac{p_x}{p_z} = \frac{\tilde{p}_x}{\tilde{p}_z} \simeq \frac{\tilde{p}_x}{\langle \tilde{p}_z \rangle} \quad (9.14)$$

where $\langle \tilde{p}_z \rangle$ stands for the mean value of $\tilde{p}_z$. Nevertheless, this value, close to $\gamma\beta$, is unknown in general with the single knowledge of the $\Sigma$ matrix. We will then use another output file of Astra, the file: run.ref.001, which contains the value of $z [\text{m}]$, and $\tilde{p}_z [\text{MeV/c}]$. One has to be careful because the number of lines of run.ref.001 and run.Sigma.001 are in general different, so the $z$ column must be used to find the indices for which the $z$ values of the two files are the closest. We will then write

$$\langle \tilde{p}_z \rangle [\text{ ]} = \frac{\langle p_z \rangle [\text{MeV/c}]}{0.511 [\text{MeV/c}]} \quad (9.15)$$

So that eventually, the $\Sigma$ matrix, in the phase space $x - x'$

$$\sigma_{12} = \langle xx' \rangle \simeq \frac{1}{\langle \tilde{p}_z \rangle /0.511} \langle x\tilde{p}_x \rangle \quad (9.16)$$
$$\sigma_{22} = \langle x'^2 \rangle \simeq \left( \frac{1}{\langle \tilde{p}_z \rangle /0.511} \right)^2 \langle \tilde{p}_x^2 \rangle \quad (9.17)$$

Table 9.2: Parameters of the distributions used for our analysis on the sigma matrix

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Uniform cylinder</td>
</tr>
<tr>
<td>$\sigma_x$</td>
<td>1 mm</td>
</tr>
<tr>
<td>$\sigma_y$</td>
<td>1 mm</td>
</tr>
<tr>
<td>$\sigma_z$</td>
<td>0.4 mm</td>
</tr>
<tr>
<td>$\epsilon_{nx,NEmit_x}$</td>
<td>${1 ; 5 }$ mrad.mm</td>
</tr>
<tr>
<td>Energy $E$</td>
<td>15 MeV</td>
</tr>
<tr>
<td>Charge $Q$</td>
<td>-0.25 nC</td>
</tr>
<tr>
<td>$N_{part}$</td>
<td>10000</td>
</tr>
</tbody>
</table>
The corresponding ellipse is plotted with horizontal and vertical units being respectively [m] and [rad].

To illustrate the difference between the ellipse representation and the real distribution in the phase space we chose first to use a uniform distribution of particles in the transversal direction. In this case, if $\sigma_x = 1$[mm], the radius of the bunch will be $2\sigma_x$. This is observed on figure 9.1 for the three sets of parameters $x - x'$, $x - p_x$, and $x - \tilde{p}_x$. One can verify that the RMS values (i.e. the ellipse semi-axes) $x_{\text{rms}}$, $x'_{\text{rms}}$ correspond exactly to the values contained in the files `run.Xemit.001` and `run.Yemit.001`, confirming that the algorithm plotting the ellipses from the covariance matrix it consistent.

(a)  
(b)  
(c)  

Figure 9.1: Phase space ellipse for a cylinder of emittance $\epsilon_{nx} = 5$mmrad.mm. for different set of parameters and units. (a) $x - x'$ - (b) $x - p_x$ - (c) $x - \tilde{p}_x$. See table 9.2 for details on the cylindrical distribution used. The distribution along $x$ is uniform, as well as the distribution of the momentum $p_x$, thus explaining this rectangular shape when plotted in the phase space. The ellipse plotted in the middle is the translation of the sigma matrix in terms of twist-parameters, with the adapted units for each case. Figure (c) is in the canonical dimensions $x - \tilde{p}_x$, so that, when multiplying the axis lengths of the ellipse, one gets the normalised emittance : 5 mrad.mm
9.4.2 Modifying the source code to output a standard sigma matrix

The main part of this work was not the implementation itself, but the understanding of the units and dimensions as well as what should be plotted as an ellipse in the phase space. The previous section, might be a little hard to read, but it really translates the methodical approach used to be sure that we output and plot the right sigma matrix / ellipse. After modification of the fortran source file X_em_mon.f, Astra is now outputing a file run.Sigmas.001 which has the following structure (table 9.3).

<table>
<thead>
<tr>
<th>Column</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>...</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>...</th>
<th>22</th>
<th>23</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
<td>$z$</td>
<td>PZ_meanU</td>
<td>$\sigma_{11}$</td>
<td>$\sigma_{12}$</td>
<td>...</td>
<td>$\sigma_{16}$</td>
<td>$\sigma_{22}$</td>
<td>$\sigma_{23}$</td>
<td>...</td>
<td>$\sigma_{56}$</td>
<td>$\sigma_{66}$</td>
</tr>
<tr>
<td>Value</td>
<td>$z$</td>
<td>$\gamma/\beta_z$</td>
<td>$\langle x^2 \rangle$</td>
<td>$\langle xx' \rangle$</td>
<td>...</td>
<td>$\langle xz' \rangle$</td>
<td>$\langle x'^2 \rangle$</td>
<td>$\langle x'y \rangle$</td>
<td>...</td>
<td>$\langle zz' \rangle$</td>
<td>$\langle z'^2 \rangle$</td>
</tr>
<tr>
<td>Units</td>
<td>[m]</td>
<td>[m]</td>
<td>[m$^2$]</td>
<td>[m rad]</td>
<td>...</td>
<td>[m]</td>
<td>[m$^2$]</td>
<td>[m rad]</td>
<td>...</td>
<td>[m]</td>
<td>[m$^2$]</td>
</tr>
</tbody>
</table>

Table 9.3: Structure of the file containing the sigma matrix: run.Sigmas.001
Part IV

Study of different components found in particle accelerator
# Table of Contents

## 10 Drift

10.1 Theory without space charge ......................................................... 65  
10.1.1 Simple case .............................................................................. 65  
10.1.2 Introducing transverse momentum correlation in the distribution .......... 66  
10.1.3 Beam waist interpretation ........................................................... 67  
10.2 Influence of the space charge in a drift ............................................ 68

## 11 Dipoles

11.1 Theory ............................................................................................ 69  
11.1.1 Equation of motion ..................................................................... 69  
11.1.2 Successions of dipole - reducing the energy spread ......................... 69  
11.1.3 Fringe fields ............................................................................... 70  
11.1.4 Effective length ......................................................................... 70  
11.2 Dipole fields from measurement ........................................................ 70  
11.2.1 Coordinates conventions ............................................................ 70  
11.2.2 Organization of measurement data ............................................... 71  
11.2.3 Field map from measurement ...................................................... 72  
11.3 Comparisons between dipole measurements and Astra modelization .......... 72  
11.3.1 Modelization of dipoles in Astra .................................................. 72  
11.3.2 Comparison of the different fields ............................................... 73  
11.3.3 Fit with Henge’s field fall-off model ............................................ 74

## 12 Compressors

12.1 Introduction to compressors theory .................................................... 76  
12.2 Adapting simulation parameters to fit reference path .......................... 77  
12.3 Maximum compression ..................................................................... 78  
12.3.1 Introducing correlation directly in the bunch ................................. 78  
12.4 Phase space plots ............................................................................ 80  
12.4.1 Transversal phase space, with and without space charge ................. 80  
12.4.2 Longitudinal phase space, with and without space charge ............... 81  
12.4.3 Projected densities for the phase space with space charge forces at the end of the compressor ........................................................ 82  
12.4.4 Introducing correlation with a cavity ............................................ 82

## 13 Emittance exchanger

13.1 Adapting simulation parameters to fit reference path .......................... 83  
13.2 Emittance measurements method ...................................................... 84

## 14 Study of a cavity

14.1 Fields inside a cavity ....................................................................... 85
14.1.1 Longitudinal component ........................................ 85
14.1.2 Correlation introduced by a cavity between $z$ and the energy .... 85
14.1.3 Determination of the cavity matrix .................................. 87
14.2 An example of transportation matrix: five cell cavity ..................... 88
  14.2.1 Equation of motion in a pillbox ................................... 88
  14.2.2 The One cell pill box model ...................................... 89
  14.2.3 Cavity Transit matrix ............................................. 90
10

Drift

10.1 Theory without space charge

10.1.1 Simple case

We study here the evolution of the phase space for a drift of four meters. Figure 10.1 displays the transversal phase space $x - x'$, while figure 10.2 displays the longitudinal phase space $z - z'$. Refer to table 10.1 for the distribution specification.

![Figure 10.1](image)

Figure 10.1: Evolution of the transversal space charge for a drift without space charge. From left to right, phase space at $z = 0$ m, $z = 2$ m and $z = 4$ m. As expected, the dispersion $x'$ stays constant in the absence of space charge, while $x$ keeps increasing. One can refer to our analogy with optics, in section 3.1.2, figure 3.1.
Figure 10.2: Evolution of the longitudinal space charge for a drift without space charge. From left to right, phase space at $z = 0\text{m}$, $z = 2\text{m}$ and $z = 4\text{m}$. Contrary to the transversal phase space, no change is observed in the longitudinal phase space.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Uniform cylinder</td>
</tr>
<tr>
<td>$\sigma_x$</td>
<td>1 mm</td>
</tr>
<tr>
<td>$\sigma_y$</td>
<td>1 mm</td>
</tr>
<tr>
<td>$\sigma_z$</td>
<td>0.4 mm</td>
</tr>
<tr>
<td>$\epsilon_{nx,\text{NEmit}_x}$</td>
<td>5 mrad.mm</td>
</tr>
<tr>
<td>$\epsilon_{nz,\text{NEmit}_z}$</td>
<td>2 mrad.mm</td>
</tr>
<tr>
<td>Energy $\mathcal{E}$</td>
<td>15 MeV</td>
</tr>
<tr>
<td>Charge $Q$</td>
<td>-0.25 nC</td>
</tr>
<tr>
<td>$N_{\text{part}}$</td>
<td>10000</td>
</tr>
</tbody>
</table>

Table 10.1: Parameters of the distributions used for our analysis on the sigma matrix

10.1.2 Introducing transverse momentum correlation in the distribution

The previous plots were done for a distribution where no correlation existed between $x$ and $p_x$. If a simple linear relation exists, the behavior of the beam is different, and will not always expend if the correlation is negative. This might appear obvious, but when generating distributions for simulations, one can forget about this aspect of the beam. A beam is often modelised as a gaussian beam, and a correlation between $x$ and $p_x$ always exists. This justify the use of quadrupoles in beam line to focus or defocus the beam. Interpretation of these results in term of beam waist is done in the next section.

Figure 10.3 displays the plane $x - x'$ for five different longitudinal positions. The distribution is the same than the previous one (table 10.1), excepts that initially, the correlation $p_x=-0.05 x$ exists.
10.1.3 Beam waist interpretation

By introducing a negative correlation between $x$ and $x'$, the beam is focusing. Nevertheless, the theory of gaussian beams predict that the transversal size of the beam will reach a minimum, the beam waist, before expanding again. This effect is shown on figure 10.4. In the previous section, without any correlation, we were starting the simulation at the beam waist, and thus only observing an expansion of the beam size. As the beam is focusing, the particles are more condensed and thus the space charge repulsion force become more important so that the beam waist is bigger if space charges are taken into account.

Figure 10.3: Evolution of the transversal space charge for a drift without space charge for a focusing beam. From left to right: $z = 0\text{m}$, $z = 1\text{m}$, $z = 2\text{m}$, $z = 3\text{m}$ and $z = 4\text{m}$.

Figure 10.4: Evolution of the beam envelop for a focusing beam - beam waist definition.
10.2 Influence of the space charge in a drift

The same plots than before will be shown for a distribution without correlation between \(x\) and \(x'\), excepts that here, the space charge forces are taken into account.

Figure 10.5: Evolution of the transversal space charge for a drift with space charge forces. From left to right : \(z = 0\)m, \(z = 2\)m and \(z = 4\)m. This time, the dispersion \(x'\) is no longer constant. The space charge forces induce an expansion of the bunch in more directions.

Figure 10.6: Evolution of the longitudinal space charge for a drift with space charge forces. From left to right : \(z = 0\)m, \(z = 2\)m and \(z = 4\)m. This time, the longitudinal phase space is modified during the drift. Remembering that the distribution is a cylinder, and remembering the shape of the longitudinal space charge field for a cylinder, one can understand why the following shape is found.
11.1 Theory

11.1.1 Equation of motion

An ideal dipole consists in a constant magnetic field of constant direction, between two metallic pole tips. The magnetic fields is produced by a coil perpendicular to the pole tips, so that the magnetic lines in the dipole are perpendicular to the two surfaces of the pole tips. In the classic coordinate system, we will have $\mathbf{B} = B_y \mathbf{e}_y$, so that the beam is deviated in the plane $x - z$. Without electric field the equation of motion in gaussian units reduces to (equations 1.1 and 1.2):

$$\frac{d\mathbf{p}}{dt} = q \mathbf{v} \times \mathbf{B}$$ (11.1)

This equation can be integrated numerically to determine the trajectory, in case of a non constant magnetic field. Nevertheless, assuming a constant magnetic field, one can show that from curvilinear coordinates we have:

$$-\mathbf{k} = -\frac{d^2 \mathbf{S}}{ds^2} \approx \frac{1}{\beta cp} \frac{d\mathbf{p}}{dt}$$ (11.2)

Projecting along the $x$ coordinates we have:

$$k_x = \frac{1}{\rho_x} = \frac{q}{cp} B_y$$ (11.3)

Eventually, the radius of the trajectory can be expressed in terms of the magnetic field:

$$\rho = \frac{pc}{qB_y} = \frac{m_0 c^2 \gamma}{qB_y}$$ (11.4)

$B\rho$ is called the magnetic rigidity.

11.1.2 Successions of dipole - reducing the energy spread

From equation 11.4 one can see that the deviation introduced by a dipole depends on the momentum $p$. Thus a sucession of two dipoles will transform a longitudinal spread of momentum $p$ into a spread of transversal position $x$, as illustrated on figure 11.1. This principle is used for emittance exchange doglegs and compressors. The path of the particle of low momentum is longer than the one of high momentum. Thus, if the initial distribution has a negative corelation between position and energy, this correlation will be reduced as the high momentum tail will travel along a smaller distance than the head, catching up with the head of the distribution.
11.1.3 Fringe fields
As one can see on figure 11.4, the magnetic field is not constant at the vicinity of the dipole edges. The field start dropping inside the dipole and continue to have a significant influence outside the dipole metallic frontier. This is called the fringe fields. It is commonly recognize that the fringe fields extend outside the pole tips is a function of the gap between the two pole tips. From this, several empirical models can be suggested. We will here mention the Henge’s field fall off model, which can be expressed as:

\[
B(s) = \frac{B_0}{1 + e^{P_n(s)}}
\]

\[
P_n(s) = \sum_{i=0}^{n} c_i \left( \frac{d(s)}{g} \right)^i
\]

where \(d(s)\) is the distance to the effective field boundary (the magnet edge), \(g\) is a scaling factor of the order of the gap size which determines the fringe fields extent, and the \(c_i\) are coefficients that determines the shape of the fringe field. A reasonable value for the degree of the polynome is \(n = 3\). Astra uses a polynome of degree 1. As we will further see (figure 11.7), this value should be extended in order to allow a better fitting with the experimental data.

11.1.4 Effective length
The effective length is defined as:

\[
l_{\text{eff}} = \frac{1}{B_0} \int B_y dl
\]

It corresponds to a abstract length of the dipole fs the field was constant equal to \(B_0\) inside the dipole, and null outside. As we saw before, this is not likely the case, so that we will always have: \(l_{\text{eff}} > l\), where \(l\) is the real length of the dipole.

11.2 Dipole fields from measurement

11.2.1 Coordinates conventions
The measurements data take as convention for the \(x\) axis, the fact that the edge of the pole tip is at -5 inches. The trapezoidal dipoles measuring 8 inches along the \(x\) axis (figure 11.3), the origin does not correspond to the center of the dipole. I chose to change this convention to use the one drawn on figure 11.2, so that the origin point is the barycenter of the dipole.
11.2.2 Organization of measurement data

Three sets of measurements data were provided, the latest being the data taken by Marcellus Parker while studying fringe fields interactions between the dipoles at A0[36]

- Exp1 : Files provided by Philippe Piot, concerning the trapezoidal dipole TDA006, and the parallelogram TDA004.

- Exp2 : Files provided by Helen Edwards, concerning the trapezoidal dipole TDA011, and the parallelogram TDA003.

- Exp3 : Files provided by Marcellus Parker, concerning only the dipole TDA011, but for a wider range of points.

- ... : At this time, no measurement has been done to study the evolution of the field along the $y$ axis, or to study TDA011 with a wider range of points, but it should be done pretty soon.

After concatenations of original experimental files into bundles, we post-treated them for them to be comparable: same coordinates and same structure. These files are stored in the folder /fields/dipoles and are named dip-paral-expX.dat and dip-trapez-expX.dat for the parallel and trapezoidal dipoles of each experiments. Their structure is as presented in table 11.1. $B_n$ corresponds to the transverse magnetic field measured for a current $I = 4.5\, A$ and normalized at 1, so that $B_n = B_y / \max(B_y)$. $X$ $Y$ $Z$ are expressed in meters with respect to the origin point $(0, 0, 0)$ at the barycenter of the dipole(cf figure 11.2).

<table>
<thead>
<tr>
<th>$X$ [m]</th>
<th>$Y$ [m]</th>
<th>$Z$ [m]</th>
<th>$I$ [A]</th>
<th>$B_n$ [T]</th>
</tr>
</thead>
</table>

Table 11.1: Structure of the experimental dipoles data files
11.2.3 Field map from measurement

It appeared that the three sets of measurements that were given to me, provide all the same results (see the following figure 11.6). Nevertheless, the one from Marcellus (exp3) are the most interesting as they provide more points. On figure 11.4, the fields measured for the trapezoidal dipole during exp3 and for the parallelogram dipole during exp1 are presented.

![Field map from the experimental measurement for the dipoles of A0](image)

Figure 11.4: Field map from the experimental measurement for the dipoles of A0 (a) trapezoidal dipole - (b) Parallelogram dipole. The field plotted here is of course \( B_y \), but normalised so that it’s maximum value is one. On black solid line, the hard edges are plotted. The dashed line are at the distance 1.5\( g \) fromn the hard edges, where \( g \) is the dipole gap.

11.3 Comparisons between dipole measurements and Astra modelization

11.3.1 Modelization of dipoles in Astra

In Astra, dipole magnetic fields are modelized as follow :

\[
B_y = \frac{B_0}{1 + \exp \frac{d(s)}{1.5g}}
\]  

(11.8)

where \( d(s) \) is the distance to the dipole edge, and \( g \) is the gap between the pole tips. The field map for the trapezoidal and parallelogram dipoles of A0 are plotted on figure 11.5.
Figure 11.5: Magnetic field map as modeled by Astra for the dipoles of A0 (a) trapezoidal dipole - (b): Parallelogram dipole. The field plotted here is $B_n$, the transverse magnetic field $B_y$ normalized so that it’s maximum value is one. On black solid line, the hard edges are plotted. The dashed line are at the distance $1.5g$ from the hard edges. At first glance, one can see that important differences are present between the modelization and the experimental fields presented on figure 11.4.

11.3.2 Comparison of the different fields

We are still studying here the trapezoidal dipole of A0. For this dipole, the beam is suppose to enter at $x = -30$mm. We thus plot the different fields we have along this line. Of course, the beam will follow a curve inside the dipole, but for the fields comparison, we will be satisfied by studying the fields along the line $x = -30$mm. For experimental files, the data available are at $x = -25.4$mm. Fields from the three sets of measurements, and fields modeled by astra are presented on figure 11.6.
CHAPTER 11. DIPOLES

Figure 11.6: Comparison between the measurement fields and astra modelized fields. The three measurement data are one above the other, showing the consistency of the measurements that were done with several years of interval and with several dipoles (TDA006 and TDA011). With good eye, one can see, that only the data from exp3 goes far on each side. Nevertheless, no need to have good eye to notice that Astra fields are far from the measurement fields.

We see here that the modelization of the field(fringe field) is bad in Astra. We know that the integral $\int Bdz$, almost entirely determine the trajectory of the beam in the dipole. As a result of this, one can adjust the amplitude of the field in astra, or play with the gap parameter to have tp

11.3.3 Fit with Henge’s field fall-off model

The measured fields from exp3 have been fitted with several Henge’s fringe field model(see section 11.1.3). Henge model is function of a polynome. We fitted the measurement data for Henge fields with polynomials of degree one, three and five. In a sense, Astra corresponds to a Henge model with a polynome of degree 1, without any constant. Nevertheless, the Henge field for a polynome of degree one that fits best the measured field is far from Astra fields. The different fitted fields are plotted on figure 11.7. To compare the different fits, the integral of the field has been calculated, and compared to the integral for the measured field. If $B_{fn}$ is a fitted field, and $B_m$ is the measured field, we define the ratio $R$ as:

$$R = \frac{\int B_{fn}(z)dz}{\int B_{m}(z)dz}$$  \hspace{1cm} \text{(11.9)}$$

note that here we don’t integrate on the trajectory of the particle, but on a straight line defined by $x = -0.03m$
Figure 11.7: Fitting the measured fields with a fringe field model. Note that the origin for the z axis is taken here at the edge of the dipole. For polynomials of degree above 3, the Henge fringe fields model can fit well any experimental fringe field. Indeed for \( n = 5 \), the integral ratio \( R \) show that the two integral differs from one percent. This figure clearly show us the need to implement a better field modelization in Astra.

From figure 11.7 we clearly see that one needs to implement a better field modelization in Astra. The other solution for the simulation, is to increase the field amplitude, of the current by a factor of 1.5(\( =1/R \)), or reduce the radius by a factor of 0.64 (\( =R \)). The underestimation of about 0.64 explains why the radius obtained by Astra when fitting the geometry of a0 or NML are smaller from a factor 1.5 compare to the real radius (see table 12.1 and 13.1).
Chapter 12

Compressors

The compressor studied is the compressor present in the ILC Test Accelerator. For detailed information concerning the geometry of the compressor please refer to section 16.

12.1 Introduction to compressors theory

Figure 12.1: Evolution of a bunch though a compressor. The path of the particles of low momentum is longer than the ones of high momentum. Thus, if the initial distribution has a negative correlation between position and energy, this correlation will be reduced as the high momentum tail will travel along a smaller distance than the head, catching up with the head of the distribution. For a given correlated energy spread, the compression will be minimal. As we go far from this point, overcompression occurs.

Figure 12.2: Scheme of a compressor made of 4 dipoles and corresponding sign of the magnetic fields
12.2 Adapting simulation parameters to fit reference path

The radii of the four dipole magnets have been manually adapted to fit the geometry of the dogleg. The focus has been on introducing an $x$-offset of respectively 0.333 m and $-0.333$ m after the first and second set of two dipoles, and keeping a beamline parallel to the $z$-axis on both sides of these two dipoles sets. Table 12.1 references the values used to obtained the desired reference path. Figure 12.3 represents the simulated reference path, with the exact geometry of the four dipoles. On figure 12.4, the deflection angle $\arctan\left(\frac{dx}{dz}\right)$, is plotted along the trajectory. The deflection angle is $\alpha = 17.36^\circ$ instead of the value $18^\circ$ expected by the designed geometry.

<table>
<thead>
<tr>
<th>Dipole 1</th>
<th>Dipole 2</th>
<th>Dipole 3</th>
<th>Dipole 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radius $\rho$ [m]</td>
<td>0.64</td>
<td>-0.64</td>
<td>-0.64</td>
</tr>
</tbody>
</table>

Table 12.1: Dipole radii needed by astra to fit the compressor geometry

Figure 12.3: Reference path through the four dipoles of the compressor

Figure 12.4: Deflection angle along the trajectory in the compressor
12.3 Maximum compression

12.3.1 Introducing correlation directly in the bunch

Figure 12.5: Longitudinal phase space before and after the compressor, without space charge forces (a) before the compressor - (b) after the compressor

Figure 12.6: Longitudinal phase space before and after the compressor, with space charge forces (a) before the compressor - (b) after the compressor
Figure 12.7: Size of the bunch after compression for different correlated energy spread (a) without space charge - (b) with space charge. The maximum compression is obtained for a correlation of -220 KeV.

Figure 12.8: Emittance after compression for different correlated energy spread (a) without space charge - (b) with space charge.
12.4 Phase space plots

12.4.1 Transversal phase space, with and without space charge

Figure 12.9: Transversal phase space for the compressor without space charge forces. From left to right: before the compressor, at the middle of the four dipoles, and after the compressor.

Figure 12.10: Transversal phase space for the compressor with space charge forces. From left to right: before the compressor, at the middle of the four dipoles, and after the compressor.
### 12.4.2 Longitudinal phase space, with and without space charge

Figure 12.11: Longitudinal phase space for the compressor without space charge forces. From left to right: before the compressor, at the middle of the four dipoles, and after the compressor.

Figure 12.12: Transversal phase space for the compressor with space charge forces. From left to right: before the compressor, at the middle of the four dipoles, and after the compressor.
12.4.3 Projected densities for the phase space with space charge forces at the end of the compressor

Figure 12.13: Phase space at the end of the compressor with densities on each axis (a) Traversal phase space - (b) Longitudinal phase space

12.4.4 Introducing correlation with a cavity

The correlation between the $z'$ and $z$ can be introduced by a cavity. Thus placing a cavity before the compressor, and changing the phase will introduce different correlation between energy and $z$. In section 14.1.2 we indeed saw that the following relation exists for the relative momentum at the output of the cavity:

$$\delta_{out} = \delta_{in} + \alpha z$$  \hfill (12.1)

Nevertheless, few simulations have been done with the case of a cavity and a compressor. The results were the same than the one presented in the previous section.
The dogleg studied is the one present at A0. For detailed information concerning the geometry of the dogleg please refer to section 15.2.

13.1 Adapting simulation parameters to fit reference path

The radii of the four dipole magnets have been manually adapted to fit the geometry of the dogleg. The focus has been on introducing an $x$-offset of 0.333m after each set of two dipoles, and keeping a beamline parallel to the $z$-axis on both sides of these two dipoles sets. Table 13.1 references the values used to obtained the desired reference path. Figure 13.2 represents the simulated reference path, with the exact geometry of the four dipoles. On figure 13.3, the deflection angle $\arctan\frac{dx}{dz}$, is plotted along the trajectory. The deflection angle is $\alpha = 20.94^\circ$ instead of the value $22.5^\circ$ expected by the designed geometry.

![Scheme of an emittance exchanger made of 4 dipoles and corresponding sign of magnetic fields](image)

Table 13.1: Dipole radii needed by astra to fit the geometry

<table>
<thead>
<tr>
<th>Dipole</th>
<th>Radius $\rho$ [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dipole 1</td>
<td>-0.598</td>
</tr>
<tr>
<td>Dipole 2</td>
<td>0.598</td>
</tr>
<tr>
<td>Dipole 3</td>
<td>-0.595</td>
</tr>
<tr>
<td>Dipole 4</td>
<td>0.595</td>
</tr>
</tbody>
</table>
13.2 Emittance measurements method

The Method of Wiedmann, use three screens, to determine the emittance with three measure of sigma x. In reality, this is done with one screen and by varying a quadrupole, and fitting a curve. This requires the knowledge of the transport matrix, which is not a problem, but space charge effects are not taken into account by the transport matrix.

Methode of slits: Slits are introduced in the beam line so that part of the bunch goes through. The “holes” are proportionnals to the dispersion at point A, so by moving the slits the dispersion is known at several locations and thus, an ellipse can be fitted.

At a0, the small intermediate camera are used to measure sigma x and sigma y. For the emittance, Sigma z is measured. The bunch reflects on a metal screen and thus emit a really fast radiation. The radiation is proportional to the duration of the bunch. This radiation, goes into a yag screen which emit electrons, who are then deflected by an electric field. Their image is then screened by a camera. The image is fitted with a gaussian, providing us with sigma z. This is called a streak camera. To determine the energy spread, we use a spectrometer. But this is done three meteres after the streak camera, so charge effects during the drift might strongly affect the dispersion. Moreover the influence of the term sigma z delta is not studied.
Chapter 14

Study of a cavity

Once again in this section we will just present some elements of theory needed for the overall understanding of the phenomenon involved during the propagation of a bunch through the accelerator. The small calculations performed in the first section helped for my understanding of the correlation between z and δ introduced by a cavity for a relativistic bunch.

14.1 Fields inside a cavity

14.1.1 Longitudinal component

In its most general form, the solution of the wave equation is expressed with Bessel functions. For a $TM_{110}$ cavity, it can be shown that the longitudinal field is of the following form:

$$E_z(z, t) = E_{z0}(z).\sin(\omega t + \varphi)$$

(14.1)

where $E_{z0}(z)$ is in a good approximation equal to $E_0 \sin(\omega t)$, with $\omega$ fixing the field oscillation frequency with time, $k$ being the space pulsation, and $\phi$ being a phase constant that can be fixed. The phase value is very important because the phase space evolution through the cavity will mainly depend on the chosen phase.

14.1.2 Correlation introduced by a cavity between $z$ and the energy

Lets introduce a bunch into the cavity and determine its gain in energy. The field seen by a particle P when it is at a position $z$ and a time $t$ inside the cavity is written $E_p^p(t)$. Of course the trajectory of the particle provides a relation $t = f(z)$ such that the field seen by a particle in the cavity is only a function of $z$. In an accelerating cavity this relation would be complex, but as the particles are travelling at a relativistic speed, we will assume that the reference particle has the following trajectory $z(t) = ct$, and the other particles to have the trajectory $z(t) = ct + \tilde{z}$. $\tilde{z}$ being the distance with respect to the reference particle, and the dispersion becoming $k = 2\pi/\lambda = \omega/c$. Given these definitions, the field seen by a particle of the bunch at the position $z$ in the cavity is:

$$E_p^p(z) = E_z(z, \frac{z - \tilde{z}}{c}) = E_0 \sin(kz).\sin(kz - k\tilde{z} + \varphi)$$

(14.2)

The hypothesis of relativistic speed imply that $\tilde{z}$ is a constant for a given particle. We will thus introduced the phase $\tilde{\phi} = \phi + \tilde{\varphi}$ with $\tilde{\varphi} = 2\pi \tilde{z}/\lambda$. This traduce that the difference of arrival time in the cavity between the particles of the bunch can be expressed by a difference of phase in the expression of the field. The total energy gained by a particle during its travel through the cavity is then obtained by calculating the following integral:

$$\int_{z=0}^{z=l_c} [\cos(\phi) - \cos(2kz + \phi)] dz = \frac{1}{2k} [\sin(\phi + 2kl_c) - \sin(\phi)] + l_c \cos(\phi)$$

(14.3)
But as the design of a cavity is such that the length of the cavity $l_c = n\lambda/2$ ($n = 1$ corresponding to the $\pi$ mode) for the bunch to see a field of a constant sign, the term between the brackets reduces to 0. Replacing $\phi$ by its expression leads us eventually with:

$$E_{out}(\tilde{z}) = E_{in}(\tilde{z}) + \rho V_0 \cos(k \tilde{z} + \varphi) \quad (14.4)$$

Thus, for the reference particle:

$$E^p(0) = E^p(0) + \rho V_0 \cos(\varphi) \quad (14.5)$$

The difference of the two last expression, gives us the evolution of the energy space phase coordinates through the cavity:

$$\Delta E = \Delta E_{in}(\tilde{z}) - \rho V k \sin(\varphi) \tilde{z} \quad (14.6)$$

Or, by writing this expression in term of momentum:

$$\delta_{out} = \delta_{in} + \alpha z \quad (14.7)$$

Expression in which we can see that a linear dependence has been introduced by the cavity between the momentum and the longitudinal coordinate. The slope of this correlation, $\alpha$, is directly proportional to $\cos(\varphi)$, where we remind that $\varphi$ is the phase parameter of the cavity. This parameter can easily be adapted to satisfy a requirement on $\delta$ or $z$.

The fields time phase

The fields we have is a picture of the time depending fields. To make them be time dependent astra will multiply them by a time depending sine function. The same is done for the other components of $E$ and $B$. As we included in our field map the two small pipes surrounding the cavity, the best way to adjust the phase is to adjust it in reference to the middle of the cavity. Indeed, at $z = L_c/2 = 150mm$, we are in the middle of the third cell. In order to have the right kick at the right position we chose $\phi$ such that the sine is equal to minus one at $L_c/2$, ie:

$$\frac{2\pi L_c}{2\lambda} + \phi = \frac{-\pi}{2} \quad (14.8)$$

We find $\phi = 287.58$ degrees. Figure 14.2 represents the $E_{z0}(z)$ and the sine function of $z$. The product of the two give us the field seen by the particle with $z$, and we can see that it will be positive all along the trajectory.
The fields scaling

As the fields were generated by , arbitrary values of the power given to the cavity were imputed. As a result of this, the fields have to be scaled to their real values by multiplying all the E and B components by a same scaling factor. Astra has its own scale factor algorithm, but it uses the maximum E value on the cavity axis. With the \( TM_{110} \) mode we use, the electric field is equal to zero on the axis, so this method is inefficient. We used a method consisting in integrating \( \int E_p^z \) along the axis \( x = 1 \text{mm}, y = 0 \), and compare the voltage value with the real one: 45kV.

\[
V = \int_0^{L_e} E_p^z dz
\]

(14.9)

Numerically we have data every \( z_{\text{step}} \), and assuming that the function under the integral is constant between two steps then, it leads us to:

\[
V = \sum_i \int_{z_i}^{z_{i+1}} E_p^z(z) dz = \sum_i E_p^y(z_i) \int_{z_i}^{z_{i+1}} dz = \sum_i E_p^y(z_i) z_{\text{step}}
\]

(14.10)

We find \( V = 4.8270 \times 10^{-3} \text{V/m} \), and a scale of \( 9.32 \times 10^6 \) to have the good field amplitude in \( \text{V/m} \), which means a scale of \( 9.327 \) to have a field in \( \text{MV/m} \) which is what is expected by Astra. We will use the same scale factor for B, which will give us fields in MTesla, which is an non usual unit, but it seems that it is what is needed by astra. As our hfs field map represent H, we will also multiply this by \( \mu_0 \).

14.1.3 Determination of the cavity matrix

Notation used for the cavity

In the previous section, we used \( u'(s) \) where \( s \) is the curviligne coordinate. But we can also interprete it as the angle between the U axis (X or Y axis) and the beam trajectory tangent. This makes the link between the optical example given previously and the beam dynamics section. As a result of this, as the tangent of this angle in radian is equal to the ratio between \( p_x \) and \( p_z \) (respectively \( p_y \) and \( p_z \)), we can write, in the approximation of small deflection around the main trajectory z axis:
\[
\begin{align*}
p_x &= \tan(x').p_z \\ p_y &= \tan(y').p_z \\ p_z &= p_0 \sqrt{1 + \tan^2(x') + \tan^2(y')} \\
&\approx p_0 (1 - \frac{x'^2 + y'^2}{2}) \approx p_0
\end{align*}
\] (14.11)

This will be useful when we will further need to go from \(x'\) to \(p_x\) and \(y'\) to \(p_y\). Now that the link between \(s\) derivative, momentum and angle is made, it is time to introduce the notations for a three dimensional beam. For practical reason, as the curvilinear coordinate \(s\) is really close to \(z\) along the trajectory inside the cavity, we will use \(z\). Thus this will be the structure of the input vector for the cavity matrix, and we will pay attention to the units:

\[
V = \begin{pmatrix}
x & (m) \\
x' & (rad) \\
y & (m) \\
y' & (rad) \\
z & (m) \\
\delta p & (\text{r})
\end{pmatrix}
\]

And our problem is, determining the matrix of the 5 cell cavity, assuming that it is constant if we consider beam trajectories which are really close to the reference particule. This matrix should give us the following relation for every input vector \(V_{in}\), which represents a particule a little bit offset relatively to the reference particule, and gives use all the parameters values at the end of the cavity for this same particule, \(V_{out}\):

\[
V_{out} = A.V_{in}
\]

### 14.2 An example of transportation matrix: five cell cavity

In this part, we will modelized a five cell cavity by five pillboxes and two drifts pipe on each side of the five cells. Such a cavity is present on a0 beam line. We will assume that the \(y\) displacement isn’t influenced by the cavity, which means that in this plane the beam exactly behave like in a drift. As a result of this we will work with a 4D matrix, and we will add the 2 \(y\)-components at the end of our study.

#### 14.2.1 Equation of motion in a pillbox

The following calculation where developped in a similar way by Donald A. Edwards [10]. We kept harmonious notations. Here we will use the \(S\) axis, which will be identical to the \(Z\) axis. The middle of the cell will correspond to \(s = 0\), and the cell length will be \(\lambda/2\). The \(r\) subscript will stands for the reference particle data. \(X\) will correspond to absolute transverse coordinate, and thus \(x \equiv X - X_r\).

From the equations of Padamsee, Knobloch and Hays [35], page 41, for a \(TM_{110}\) mode pillbox, in the paraxial approximation, usual for linear dynamics, we can assume that the fields will be of the following form. As the longitudinal \(E\) field is linear with respect to \(X\) near the longitudinal axis we will use \(E' = \frac{\partial E}{\partial X}\) assumed to be constant near the axis:

\[
\begin{align*}
E_s(X, t) &= E'X \cos(\omega t) \quad (14.12) \\
B_y(X, t) &= \frac{E'}{\omega} \cos(\omega t) \quad (14.13)
\end{align*}
\]

The equations of motion, with the only Lorentz force \(\vec{F} = e(\vec{E} + \vec{v}) \times \vec{B}\) are:

\[
\begin{align*}
\frac{dP_X}{dt} &= -eE'V_s \sin(\omega t) \\
\frac{dP_S}{dt} &= eE'X \cos(\omega t) + eE'V_X \sin(\omega t) \quad (14.15)
\end{align*}
\]
We will define without justification the transit time factor $T$:

$$T = \frac{2eE' pck}{c}$$

### 14.2.2 The One cell pill box model

For a particle leading the reference particle by a distance $z$ we have:

$$E_a = E' X \cos(\omega t - kz) \approx E' X \cos(\omega t) + E' X k z \sin(\omega t) \quad (14.16)$$

$$B_y = \frac{E'}{\omega} \cos(\omega t - kz) \approx \frac{E'}{\omega} \sin(\omega t) - \frac{E'}{c} z \cos(\omega t) \quad (14.17)$$

which can be written differently using $s = v_x t \approx c t$ and $\omega = kc$. Let’s write equations 14.14 and 14.15 for the reference particle, using derivative with respect to $s$ now:

$$\frac{dP_{X,r}}{ds} = -\frac{eE'}{kc} \sin(k s) \quad (14.18)$$

$$\frac{dP_{S,r}}{ds} = \frac{eE'}{c} X_{r} \cos(k s) + \frac{eE'}{kc^2} V_X \sin(k s) \quad (14.19)$$

Subtracting those two sets of equations gives us

$$\frac{dp_x}{ds} = \frac{eE'}{c} z \cos(k s) \quad (14.23)$$

$$\frac{dp_a}{ds} = \frac{eE'}{c} X_{r} \cos(k s) + \frac{eE'}{kc} \frac{dx}{ds} \sin(k s) \quad (14.24)$$

Let’s integrate 14.23 from $s = -\lambda/4$ to $s$, remembering that $k \lambda/4 = \pi/2$:

$$p_x(s) = p_x(-\lambda/4) + \frac{eE'}{ck} z (\sin(k s) + 1) \quad (14.25)$$

Now, as $x'(s) = p_x/p_a$ we have:

$$x'(s) = x'_{in} + \frac{Tp}{2} z \sin(k s) + \frac{Tp}{2} z \quad (14.26)$$

Which we can integrate again:

$$x(s) = x_{in} + \left( x'_{in} + \frac{Tp}{2} z \right) \left( s + \frac{\lambda}{4} \right) - \frac{Tp}{2k} z \cos(k s) \quad (14.27)$$

The integration of 14.24 is now possible thanks to the two last expressions. We will provide details of the algebra here. Inserting expressions 14.26 and 14.27 in equation 14.24 yields to:

$$\frac{dp_a}{ds} = \frac{Tp}{2} \left[ \frac{x_{in} + \left( x'_{in} + \frac{Tp}{2} z \right) \left( s + \frac{\lambda}{4} \right) - \frac{Tp}{2k} z \cos(k s)}{s} \right] \cos(k s) \quad (14.28)$$

$$+ \frac{Tp}{2} \left[ x'_{in} + \frac{Tp}{2} z \sin(k s) + \frac{Tp}{2} z \right] \sin(k s) \quad (14.29)$$
After factorization:
\[
\frac{dp_s}{ds} = x_{in} \frac{T}{2} p [k \cos(ks)] \quad (14.30)
\]
\[
+ x'_{in} \frac{T}{2} p \left[ k \left( s + \frac{\lambda}{4} \right) \cos(ks) + \sin(ks) \right] \quad (14.31)
\]
\[
+ z \frac{T^2}{4} p \left[ k \left( s + \frac{\lambda}{4} \right) \cos(ks) - \cos^2(ks) + \sin^2(ks) + \sin(ks) \right] \quad (14.32)
\]

We use the relation \(-\cos^2(ks) + \sin^2(ks) = -\cos(2ks)\), and the following integrals:
\[
\int_{-\lambda/4}^{s} \sin(ks) = -\frac{\cos(ks)}{k} \quad (14.33)
\]
\[
\int_{-\lambda/4}^{s} \cos(2ks) = \frac{\sin(2ks)}{2k} \quad (14.34)
\]
\[
\int_{-\lambda/4}^{s} k \left( s + \frac{\lambda}{4} \right) \cos(ks) = \left( s + \frac{\lambda}{4} \right) \sin(ks) + \frac{\cos(ks)}{k} \quad (14.35)
\]

It yields:
\[
\frac{\delta p}{p} = x_{in} \frac{T}{2} [1 + \sin(ks)] \quad (14.36)
\]
\[
+ x'_{in} \frac{T}{2} \left[ \left( s + \frac{\lambda}{4} \right) \sin(ks) \right] \quad (14.37)
\]
\[
+ z \frac{T^2}{4} \left[ \left( s + \frac{\lambda}{4} \right) \sin(ks) - \frac{1}{2k} \sin(2ks) \right] \quad (14.38)
\]

### 14.2.3 Cavity Transit matrix

To have the matrix of the entire cell we just have to evaluate the previous expression at \( s = \lambda/4 \) which corresponds to the exit of the cell.

\[
M_{\text{cell}} = \begin{pmatrix}
1 & \lambda/2 & T\lambda/4 & 0 \\
0 & 1 & T & 0 \\
0 & 0 & 1 & 0 \\
T & T\lambda/4 & T^2\lambda/8 & 1
\end{pmatrix}
\]

Then the matrix of the 5 cells is:

\[
M_{\text{cell}}^5 = \begin{pmatrix}
1 & 5\lambda/2 & 25T\lambda & 0 \\
0 & 1 & 5T & 0 \\
0 & 0 & 1 & 0 \\
5T & 25T\lambda/4 & 85T^2\lambda/8 & 1
\end{pmatrix}
\]

The matrix for a drift of \( d \) is:

\[
M_d = \begin{pmatrix}
1 & d & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

Eventually, the matrix of the whole cavity with surrounding pipes is:

\[
M_{\text{cav}} = M_d.M_{\text{cell}}^5.M_d = \begin{pmatrix}
1 & 2d + 5\lambda/2 & 5Td + 25T\lambda/4 & 0 \\
0 & 1 & 5T & 0 \\
0 & 0 & 0 & 0 \\
5T & 5Td + 25T\lambda/4 & 85T^2\lambda/8 & 1
\end{pmatrix}
\]
Before evaluating this matrix let’s write it in 6D, so including $y$ and $y'$ components:

$$
M_{cav} = \begin{pmatrix}
1 & 2d + 5\lambda/2 & 0 & 0 & 5Td + 25T\lambda/4 & 0 \\
0 & 1 & 0 & 0 & 5T & 0 \\
0 & 0 & 1 & 2d + 5\lambda/2 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
5T & 5Td + 25T\lambda/4 & 0 & 0 & 85T^2\lambda/8 & 1
\end{pmatrix}
$$

Numerically:

$$
M_{cav} = \begin{pmatrix}
1 & 0.300 & 0 & 0 & -0.452 & 0 \\
0 & 1 & 0 & 0 & -3.016 & 0 \\
0 & 0 & 1 & 0.300 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
-3.016 & -0.452 & 0 & 0 & 0.297 & 1
\end{pmatrix}
$$

with

$$
d = 0.05385 \text{ m} \\
\lambda = 0.07692 \text{ m} \\
T = -0.6031
$$
Part V

Preliminary steps for the simulation of full accelerators line
# Table of Contents

15 **A0 photo injector**

15.1 Specifications .......................................................... 94
15.2 Geometry ................................................................. 94
  15.2.1 Coordinates of the principal elements ......................... 94
  15.2.2 Dogleg lattice coordinates ..................................... 94
15.3 Simulation ............................................................... 96

16 **ILCTA at New Muon lab (NML)**

16.1 About ILCTA ............................................................ 97
16.2 Specifications .......................................................... 97
16.3 Coordinates of the principal elements .............................. 98
16.4 Compressor lattice coordinates ..................................... 98
16.5 Simulation ............................................................... 99
15.1 Specifications

Table 15.1 presents some specifications for the beam at a0.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Nominal</th>
<th>Expected range</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>charge per bunch</td>
<td>Q</td>
<td>3.2</td>
<td>0.1-10</td>
<td>nC</td>
</tr>
<tr>
<td>bunch per macropulse</td>
<td>( N_b )</td>
<td>2850</td>
<td>1-3000</td>
<td>-</td>
</tr>
<tr>
<td>beam total energy</td>
<td>( \mathcal{E} )</td>
<td>40-50</td>
<td>10-50</td>
<td>MeV</td>
</tr>
<tr>
<td>rms transverse normalized emittance</td>
<td>( \epsilon_{nx}, \epsilon_{ny} )</td>
<td>4-5</td>
<td>4-5</td>
<td>( \mu \text{m} )</td>
</tr>
</tbody>
</table>

Table 15.1: Specifications of A0 beam out of the photoinjector

15.2 Geometry

15.2.1 Coordinates of the principal elements

Please refer to annex D for the coordinates of the beam line at a0.

15.2.2 Dogleg lattice coordinates

Figure 15.1 displays the coordinates of the dogleg at a0, thanks to Bruce Popper (Fermilab)
Table 15.2 reference the coordinates of the dipoles where the origin is taken at the entrance of the first dipole. See figure 3 for convention of notations. The column \( x \) and \( z \) corresponds to the real geometry (cf figure 15.1) expressed with respect to the entrance point in the first dipole. Nevertheless, Astra will refuse this geometry as it induces a fringe field overlapping inside the dipole. To avoid this we will have to shorten the height of the two trapezoidal magnets. The results from this operation correspond to column 3 and 4. The first dipole being at \( z = 5808.80 \text{mm} \), we write the final coordinates with respect to the cathode in the last column of the table.
### Table 15.2: Dipoles hard edge coordinates in mm for A0 emittance exchange lattice

<table>
<thead>
<tr>
<th>Point</th>
<th>$x$</th>
<th>$z$</th>
<th>$x+dx$</th>
<th>$z+dz$</th>
<th>$x+dx$</th>
<th>$z+dz+z_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>D11</td>
<td>-71.66</td>
<td>-19.20</td>
<td>-71.66</td>
<td>5808.80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D21</td>
<td>131.54</td>
<td>35.25</td>
<td>111.54</td>
<td>29.89</td>
<td>111.54</td>
<td>5857.89</td>
</tr>
<tr>
<td>D31</td>
<td>-71.66</td>
<td>260.14</td>
<td>-71.66</td>
<td>5808.80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D41</td>
<td>131.54</td>
<td>205.69</td>
<td>111.54</td>
<td>211.05</td>
<td>111.54</td>
<td>6039.05</td>
</tr>
<tr>
<td>I1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>5828.00</td>
</tr>
<tr>
<td>O1</td>
<td>45.67</td>
<td>228.70</td>
<td>45.67</td>
<td>6056.70</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| D21   | 159.61 | 846.01 | 159.61 | 6674.01 |
| D22   | 413.61 | 779.66 | 413.61 | 6607.66 |
| D32   | 159.61 | 1084.91 | 159.61 | 6912.91 |
| D42   | 413.61 | 1018.55 | 413.61 | 6846.55 |
| I2    | 287.19 | 811.82 | 287.19 | 6639.82 |
| O2    | 333.33 | 1040.06 | 333.33 | 6868.06 |

| D13   | 261.20 | 2349.54 | 261.20 | 8177.54 |
| D23   | 512.20 | 2283.18 | 512.20 | 8111.18 |
| D33   | 261.20 | 2588.44 | 261.20 | 8416.44 |
| D43   | 512.20 | 2522.08 | 512.20 | 8350.08 |
| I3    | 333.33 | 2330.21 | 333.33 | 8158.21 |
| O3    | 378.11 | 2558.81 | 378.11 | 8386.81 |

| D14   | 533.87 | 3164.93 | 3159.57 | 553.87 | 3159.57 | 8987.57     |
| D24   | 737.07 | 3110.48 | 737.07  | 8938.48 |
| D34   | 533.87 | 3335.37 | 3340.73 | 553.87 | 3340.73 | 9168.73     |
| D44   | 737.07 | 3389.82 | 737.07  | 9217.82 |
| I4    | 620.20 | 3141.80 | 620.20  | 8969.80 |
| O4    | 666.65 | 3370.80 | 666.65  | 9198.80 |

Table 15.2: Dipoles hard edge coordinates in mm for A0 emittance exchange lattice

### 15.3 Simulation

No simulation results are available yet, but the input files for astra are available in annex:H
Chapter 16

ILCT A at New Muon lab (NML)

16.1 About ILCT A

Fermilab is currently building an ILC test accelerator, called NML. The facility will be capable of testing superconducting accelerating modules with an ILC-like beam. It is anticipated, as the need for ILC-related tests diminishes, that the facility will start providing beam to support Advanced accelerator R&D projects and experiments. An overview of the ILC test accelerator is shown in Figure 16.1. The accelerator complex incorporates an L-band radiofrequency (rf) gun operating at 1.3 GHz. The bunches are emitted from a CsTe photocathode with a nominal charge of 3.2 nC. The photocathode drive-laser enables the generation of a series of bunches repeated at 3 MHz within a macropulse of nominal duration of 1 ms. The beam energy upon exit from the rf-gun is about 4 MeV. The energy is then boosted to 40-50 MeV by two TESLA-type cavities located downstream of the rf-gun. Eventually a 3rd harmonic cavity will be installed, its main purpose being the linearization of the longitudinal phase. The nominal beam parameters out of the injector are shown in Table 16.1.

Figure 16.1 was taken from the presentation of S. Nagaitsev[34]

16.2 Specifications

Table 16.1 inspired of specifications provided on the website of the workshop “Possible Directions for Advanced Accelerator R&D at the ILC test accelerator at Fermilab”[19] organized by P. Piot, and the presentation [37].

Figure 16.1: NML facility
### 16.3 Coordinates of the principal elements

Table 16.2 is inspired from the presentation given by Y.-E. Sun at Fermilab[41].

<table>
<thead>
<tr>
<th>Element</th>
<th>z center position [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bucking solenoid</td>
<td>-0.0949</td>
</tr>
<tr>
<td>Photocathode</td>
<td>0.0000</td>
</tr>
<tr>
<td>Main solenoid</td>
<td>0.2152</td>
</tr>
<tr>
<td>CC1</td>
<td>2.8000</td>
</tr>
<tr>
<td>CC2</td>
<td>5.5110</td>
</tr>
<tr>
<td>Q103</td>
<td>10.7639</td>
</tr>
<tr>
<td>Q104</td>
<td>10.9647</td>
</tr>
<tr>
<td>Q105</td>
<td>11.1655</td>
</tr>
<tr>
<td>D101</td>
<td>11.7655</td>
</tr>
<tr>
<td>D102</td>
<td>12.6508</td>
</tr>
<tr>
<td>D103</td>
<td>13.6706</td>
</tr>
<tr>
<td>D104</td>
<td>14.5558</td>
</tr>
</tbody>
</table>

Table 16.2: Coordinates of beam line elements at NML

### 16.4 Compressor lattice coordinates

Figure 16.2 has been provided by Steve Wesseln (Fermilab), version from the 15th June 2009.
Table 16.3 reference the coordinates of the dipoles where the origin is taken at the cathode. See figure 3 for convention of notations. The column $x$ and $z$ corresponds to the real geometry (cf figure 16.2) expressed with respect to the entrance point in the first dipole. Nevertheless, in previous simulations from Yin-e Sun[41] a dipole length of 230 mm has been used. We thus add an offset of $dz = \pm 15$ mm to the longitudinal coordinate to satisfy this condition. The center of the first dipole being at $z = 11.7655m$, we write the final coordinates with respect to the cathode in the last column of the table.

![Table 16.3: Dipoles hard edge coordinates in mm for NML compressor](image)

### 16.5 Simulation

No simulation results are available yet, but the input files for astra are available in annex:H
Astra is a non commercial program, but is still a powerful program for linear particle accelerator simulations, and especially its space charge calculation has nothing to envy to its competitors. Few algorithms solve Poisson equation with a three dimensional grid, and few of them try to adapt the meshing to the bunch. The two dimensionnal space charge algorithm of Astra, has proven satisfying results for the last years. Nevertheless, a complete description of the beam requires a three dimensional approach, as the beam is not likely to have or keep revolution symmetry. A three dimensional space charge algorithm has been implemented by Klaus Flottmann. Nevertheless, his first version was not adapting the grid to the bunch in case of a rotated bunch, typically a bunch at the exit of a dipole. From his original 3D space charge algorithm, I corrected small bugs, and implemented a solution to adapt the meshing to the bunch, and thus, increase the quality of the problem resolution for a same amount of grid cells. Throughout this document, we tried to show the validity of our algorithm, with, or without rotation of the bunch. We presented briefly small results concerning different components, trying to focus on the comparison between space charge runs, and runs without space charge. From this, we saw that space charge effects are not negligible, and they clearly introduce modifications in the phase space. From the study of the compressor, this appeared clear. For the two cases, the minimal compression is obtained for the same correlated energy spread, but the compression factor will be at least ten time smaller with space charge effects. Even in a simple drift, space charge effects are non negligible. From our analysis, it appeared necessary to find a way to change the modelization of dipoles in Astra before trying to compare experimental results with simulations. Hopefully our algorithm and our work will help the understanding and the estimation of space charge effects in particle beams.
I would like to thank Helen Edwards who gave me the opportunity to do this internship at Fermilab in the laboratory of A0, and who regularly guided me and advised me with patience, making this project challenging, and thus enjoyable. I would also like to thank Philippe Piot, who spent a lot of time discussing with me, either to help me in the implementation of the algorithm, or to help me in the interpretation of simulation results by explaining me the theory behind it. He replied with attention to each of my questions, and thank to this, it was a real pleasure to study particle physics. I wish to thank Bruce Popper and Steve Wessel, who provided me amazingly quickly with the coordinates and drawing of A0 and NML. Among A0 team, James Santucci, and Amber, helped me several times at work, but I am mainly grateful for their moral support that counted a lot. Again, from A0 team, thank you to Wayne for the different coffee time, and to Elvin Harms for welcoming me in the team, and helping me to settle during the first weeks. I would like to thank all the personnel from Fermilab I met. If it was for scientific or administrative purpose, or if it was in spare time, they all welcomed me with attention. They all participate in making this project a rich adventure. Eventually, my last words would be for my brother, Julien, who has been supporting me all this time, cheering me up, and who did a lot for me to enjoy my stay in the US. Merci Julien.
Annexes
Appendix A

Documentation of R scripts

Astra comes with a variety of programs to interpret the results of one simulation. Interpreting the results for several simulations was required by our study so we decided to implement our own scripts. These scripts were written in the language R (see website [7]). We will try to quickly document how to use these files.

Installing R

R is a free computing program like matlab, multi-platform, that can be downloaded at the following website: http://www.r-project.org/ Installing it should not cause any problem.

Installing libraries required by our scripts

For unix user, it is better to do this as root. Launch R and type:

```
install.package("name", dependencies=T)
```

where "name", stands for the name of the package. The packages known to be required are: rgl, fields, MASS, lattice and ellipse. Note that after installation of these, they are not loaded by default, they have to be loaded each time you run R by typing:

```
library(rgl)
library(fields)
library(MASS)
library(lattice)
library(ellipse)
```

Nevertheless, if you follow the next sections, you won’t have to load them at each time. Indeed, these commands are written in the file mains.r. If you load all my files in the workspace, the libraries will thus be automatically loaded.

Loading my functions into the program

To load any source file the command is

```
source("C:/sources/program.r")
```

All the functions in the file program.r, will be loaded in the workspace.

To automatically load all my scripts(files with extension "r") when R starts, a loop like this will work:

```
for(f in list.files(path="C:/sources/", pattern="\.[r]")){source(f);}
```
You can add this line in the file " /.Rprofile" for linux users or "C:/programfiles/R/etc/Rprofile.site" for windows users for it to be executed automatically at each startup.

**Defining the main workspace**

In file *mains.r*, modify the line :

```r
assign("ROOTDIR","/home/ebranlar/SpaceCharge",envir=.GlobalEnv)
```

And replace "/home/ebranlar/SpaceCharge", by any absolute path from which your work documents will be placed. This will define the global variable "ROOTDIR". Each time you call

```r
setwd.spc(""
```

R will go in this directory. If you call

```r
setwd.spc("simulations/")
```

R will go in the sub directory "simulations" that is hopefully in the directory you gave for your path ("ROOTDIR").

**Principals functions**

We will here describe the functions present in the file *astra-postpro.r*. We will provide an example of how to use these functions in the next section. This source file, is dependent on other source files, so it is better to load all the ".r" scripts, before trying to run them. We will only describe the content of this source file, as it is the one that provides tools to plot the interpretation of Astra results.

- **ppSummaryPlot** : produces the screen interface with the trajectories, beam dimensions, emittances, energies and deviations. (see figure A.1)

- **ppSigmaPlot** : produces the phase space screen interface (see figure A.2)

- **loadCaseGeometry** : load from an astra input file the geometry

- **plot.objects** : after the previous function has been called, one can call plot.objects to display the scene at real scale.

- **plot.all.ellipses** : the functions that is able to plot several phase space ellipses. This functions is called by ppSigmaPlot, but can be stand alone.

- **plot.ellipse** : called by the previous one, this function display one ellipse, and if needed the distribution

- **loadAstraOutput** : load astra output files

- **initPostPro** : empty the workspace, this is necessary between two data interpretation

- **plotvar** : plot a parameter against the other one, for several simulations. This functions is called by ppSummaryPlot, but can be stand alone.

**A small example**

Assuming all my source files are included in the workspace, and assuming that two Astra simulations have been run, and stored in the folders(for instance) "/astra/simul1/" and "/astra/simul2/", copying-pasting this in the R command window will display the two main screens. To see more examples, have a look at files "main-postpro.r", "main-postpro-old.r". They contain all the simulations I used.
APPENDIX A. DOCUMENTATION OF R SCRIPTS

```r
setwd.spch("")
initPostpro() ; Folders=character(); niceruns=character()

Folders[1]=="/astra/simul1"
Folders[2]=="/astra/simul2"

niceruns=c("Simulation 2", "Simulation 2")
assign("NICERUNS", niceruns, envir=GlobalEnv)
assign("CASE", "Drift", envir=GlobalEnv)
assign("DISTR", "Ellipsoid", envir=GlobalEnv)

# loading astra output files
for(f in Folders){
  setwd.spch("")
  loadAstraOutput(f)
}

# loading the geometry
setwd.spch("")
loadCaseGeometry(Folders[1])

# plotting the summary screen
ppSummaryPlot()

# defining the points where the phase space wants to be plotted (in z
# coordinates, in meters)
SIGMAPORTS=c(0, 2, 4)

# plotting the phase space screen
ppSigmaPlot()
```

Figure A.1: Screen generated by the function ppSummaryPlot
Problems, questions

Feel free to contact me.
Appendix B

Jacobi diagonalization of a symmetric matrix

B.1 Mathematical description of the method

The method adopted by most algorithms to reduce a matrix $A$ to a diagonal form is to apply a sequence of similarity transformations:

$$A \rightarrow P_1^{-1} \cdot A \cdot P_1 \rightarrow P_2^{-1} \cdot P_1^{-1} \cdot A \cdot P_1 \cdot P_2 \rightarrow \text{etc.} \quad (B.1)$$

So that eventually we have:

$$D = R \cdot A \cdot R^{-1} \quad (B.2)$$

$$R = P_1 \cdot P_2 \cdot \ldots \quad (B.3)$$

For the Jacobi method, these transformations consist of orthogonal similarities. Each Jacobi transformation, is a plane rotation designed to annihilate one of the off-diagonal matrix elements. Successive transformations undo previously set zeros, but the off-diagonal elements nevertheless get smaller and smaller, until the matrix can be considered diagonal for a given precision. Moreover, the product of the transformations, $R$ in equation B.3, is the matrix of eigenvectors. The Jacobi method is absolutely foolproof for all real symmetric matrices. For matrices of order greater than about 10, the algorithm becomes slower than other algorithms. However, the Jacobi algorithm is much simpler than the more efficient methods.

The basic Jacobi rotation $P_{pq}$ is a matrix of the form

$$P_{pq} = \begin{bmatrix}
1 \\
\vdots \\
c & \cdots & s \\
\vdots \\
-s & \cdots & c \\
\vdots \\
\cdots & \cdots & \cdots & \cdots & 1
\end{bmatrix} \quad (B.4)$$

The numbers $c$ and $s$ are the cosine and sine of a rotation angle $\phi$:

$$c = \cos(\phi) ; \ s = \sin(\phi) \quad (B.5)$$

The Jacobi rotation is used to transform matrix $A$ with the similarity:

$$A' = P_{pq}^T \cdot A \cdot P_{pq} \quad (B.6)$$
Now, $P_{pq}^T \cdot A$ changes only rows $p$ and $q$ of $A$, while $A \cdot P_{pq}$ changes only its columns $p$ and $q$. The result of the product on each side is, for $r \neq p, r \neq q$:

\begin{align}
\begin{aligned}
    a'_{rp} &= ca_{rp} - sa_{rq} \\
    a'_{rq} &= ca_{rq} + sa_{rp}
\end{aligned}
\end{align}

and otherwise:

\begin{align}
\begin{aligned}
    a'_{pp} &= c^2 a_{pp} + s^2 a_{qq} - 2 sca_{pq} \\
    a'_{qq} &= s^2 a_{pp} + c^2 a_{qq} + 2 sca_{pq} \\
    a'_{pq} &= (c^2 - s^2) a_{pq} + sc(a_{pp} - a_{qq})
\end{aligned}
\end{align}

Let’s remember that we want to try to annihilate the off-diagonal elements. $A'$ and $A$ are similar, so they have the same Frobenius norm $||.|.|_F$: the sum of squares of all components. However we can choose $\phi$ such that $a'_{pq} = 0$, in which case $A$ has a larger sum of squares on the diagonal. As we will further see, this is important for the convergence of the method. In order to optimize this effect, $a_{pq}$ should be the largest off-diagonal component, called the pivot. From equation B.11, $a'_{pq} = 0$ yields:

\begin{align}
    a'_{pq} &= \cos(2\phi) a_{pq} + \frac{1}{2} \sin(2\phi)(a_{qq} - a_{pp}) = 0
\end{align}

from which we define $\theta$:

\begin{align}
    \theta &= \cot 2\phi = \frac{c^2 - s^2}{2sc} = \frac{a_{qq} - a_{pp}}{2a_{pq}}
\end{align}

If we let $t = s/c$, the definition of $\theta$ can be rewritten

\begin{align}
    t^2 + 2t\theta - 1 = 0
\end{align}

We choose the smaller root of this equation, where the rotation angle is less than $\pi/4$. This choice gives the most stable reduction. The smaller root is:

\begin{align}
    t &= \frac{\text{sgn}(\theta)}{|\theta| + \sqrt{\theta^2 + 1}}
\end{align}

If $\theta$ is so large that $\theta^2$ would overflow on the computer, we set $t = 1/(2\theta)$. It now follows that

\begin{align}
    c &= \frac{1}{\sqrt{t^2 + 1}} \\
    s &= tc
\end{align}

To minimize numerical roundoff error, equation B.11 is replaced by $a'_{pq} = 0$ The idea in the remaining equations B.7 - B.11 is to set the new quantity equal to the old quantity plus a small correction. Thus we can use $a'_{pq} = 0$ to eliminate $a_{qq}$ from B.9, giving

\begin{align}
    a'_{pp} &= a_{pp} - ta_{pq}
\end{align}

Similarly,

\begin{align}
\begin{aligned}
    a'_{qq} &= a_{qq} + ta_{pq} \\
    a'_{rp} &= a_{rp} - s(a_{rq} + \tau a_{rp}) \\
    a'_{rq} &= a_{rq} + s(a_{rp} - \tau a_{rq})
\end{aligned}
\end{align}

where $\tau(= \tan \phi / 2)$ is defined as

\begin{align}
    \tau &= \frac{s}{1 + c}
\end{align}
One can see the convergence of the Jacobi method by considering the sum of the squares of the off-diagonal elements:

\[ S = \|A\|_F = \sum_{r \neq s} |a_{rs}|^2 \]  
(B.23)

Using equations B.7 - B.11 and the similarity properties, we have:

\[ S' = S - 2|a_{pq}|^2 \]  
(B.24)

The sequence of S's thus decrease monotonically. Since the sequence is bounded below by zero, and since we can choose \( a_{pq} \) to be whatever element we want, the sequence can be made to converge to zero. Eventually one obtains a matrix \( D \) that is diagonal to machine precision. The diagonal elements give the eigenvalues of the original matrix \( A \), since

\[ D = R^T \cdot A \cdot R \]  
(B.25)

where

\[ R = P_1 \cdot P_2 \cdot P_3 \ldots \]  
(B.26)

the \( P_i \)'s being the successive Jacobi rotation matrices. The columns of \( R \) are the eigenvectors.

### B.2 Fortran source code

```fortran
C  ***********************************************************************
C PROGRAM: DIAGO_NxN.f
C AUTHOR: Emmanuel BRANLARD
C DATE: 13/4/09 VERSION: 1
C DESCRIPTION:
C Returns the eigenvalues and normalized eigenvectors of a symmetric
C NxN matrix with the use of the Jacobi algorithm
C ***********************************************************************
C SUBROUTINE DIAGO_NxN(N, A, E, lambda,DEBUG )
C PROGRAM: Subroutines DIAGO_NxN
C AUTHOR: Emmanuel BRANLARD
C DATE: 13/4/09 VERSION: 1
C DESCRIPTION:
C Returns the eigenvalues and normalized eigenvectors of
C a symmetric NxN matrix with the use of the Jacobi algorithm.
C ***********************************************************************
C ARGUMENTS
C N: dimension of the matrix A
C A: symmetric matrix
C E: output for eigenvectors
C lambda: output for eigenvalues
C ***********************************************************************
C PARAMETERS
C INTEGER N
C ARGUMENTS
C DOUBLE PRECISION A(N,N)
C DOUBLE PRECISION E(N,N)
C DOUBLE PRECISION lambda(N)
```

August 2009 109  Emmanuel Branlard
APPENDIX B. JACOBI DIAGONALIZATION OF A SYMMETRIC MATRIX

35       LOGICAL DEBUG
36
37       C LOCAL VARIABLES
38       INTEGER IND(N)
39       LOGICAL CHANGED(N)
40       INTEGER k, l, m, i, state, cpt
41       DOUBLE PRECISION y, t, c, s, p
42
43       c-     write(*,*) "DIAGO_NxN:: Initialization"
44       C Initialize state
45       state=N
46
47       C Initialize E to identity matrix
48       DO k = 1, N
49           E(k, k) = 1.0D0
50       DO l = 1, k-1
51           E(k, l) = 0.0D0
52           E(l, k) = 0.0D0
53       END DO
54       END DO
55
56       C Initialize indexes to the maximum of each row
57       C Initialize lambda to diag(A)
58       C Initialize CHANGED to TRUE
59       DO k = 1, N
60           IND(k) = maxind(N, k, A)
61           lambda(k) = A(k, k)
62           CHANGED(k) = .TRUE.
63       END DO
64       END DO
65
66       IF (DEBUG) write(*,*) "DIAGO_NxN:: Enter diagonalization loop"
67       cpt=0
68       p=1.0D0
69       c       .AND. p>0
70       DO WHILE (state.NE.0 .AND. cpt<(2*N)) !next rotation
71           ! find index (k, l) of pivot p
72           cpt=cpt+1
73           m=1
74           DO k = 2, N-1
75               IF ( abs(A(k, IND(k))) .GT. abs(A(m, IND(m))) ) then
76                   m=k
77             ENDIF
78         END DO
79
80       k=m
81       l=IND(m)
82       p=A(k, l)
83       ! now the pivot is set
84
85       ! calculate cosine and sine
86       y=(lambda(1)-lambda(k))/2
87       t=abs(y) + sqrt(p**2 + y**2)
88       s=sqrt(p**2 + t**2)
89       c=t/s
90       s=p/s
91       t=(p**2)/t
92       IF(y.LT.0) then
93 

August 2009 110 Emmanuel Branlard
APPENDIX B. JACOBI DIAGONALIZATION OF A SYMMETRIC MATRIX

\begin{verbatim}
A(k, l) = 0.0
call update(lambda(k), -t, changed(k), state)
call update(lambda(l), t, changed(l), state)

! rotate rows and columns k and l
DO i = 1, k-1
   call rotate(N, A, i, k, i, l, c, s)
END DO
DO i = k+1, l-1
   call rotate(N, A, k, i, i, l, c, s)
END DO
DO i = l+1, N
   call rotate(N, A, k, i, l, i, c, s)
END DO

! rotate eigenvectors
DO i = 1, N
   call rotate(N, E, i, k, i, l, c, s)
END DO

! rows k, l have changed, update rows ind(k), ind(l)
IND(k) = maxind(N, k, A)
IND(l) = maxind(N, l, A)
END DO

IF (DEBUG) THEN
   write(*,*) "DIAGO_NxN:: Done after ", state, " loops"
   write (*,*) "DIAGO_NxN:: Eigen vectors:"
   write (*,*) "DIAGO_NxN:: E: ",E(1,1), E(1,2), E(1,3)
   write (*,*) "DIAGO_NxN:: E: ",E(2,1), E(2,2), E(2,3)
   write (*,*) "DIAGO_NxN:: E: ",E(3,1), E(3,2), E(3,3)
   write (*,*) "DIAGO_NxN:: Eigen values:"
   write (*,*) "DIAGO_NxN:: L: ",lambda(1), lambda(2), lambda(3)
END IF

END SUBROUTINE

C * End of subroutine DIAGO

C *****************************************************************************

C INTEGER FUNCTION maxind(n, k, A)
C    index of the largest element in row k
INTEGER n
INTEGER k
DOUBLE PRECISION A(n, n)

INTEGER maxind
INTEGER i
maxind = k+1
DO i = k+2, n
   if ( abs(A(k, i)) .GT. abs(A(k, maxind)) ) then
      maxind = i
   endif
END DO
RETURN
END
\end{verbatim}
APPENDIX B. JACOBI DIAGONALIZATION OF A SYMMETRIC MATRIX

```fortran
C
C ************************************************************
C
C SUBROUTINE update(lambda, t, changed, state)
C update the eigenvalues
INTEGER state
LOGICAL changed
DOUBLE PRECISION lambda, t

DOUBLE PRECISION y
y=lambda
lambda=t+y
if (changed .AND. y.EQ.lambda) then
  changed=.FALSE.
  state=state-1;
elseif (.NOT. changed) .AND. y.NE.lambda) then
  changed=.TRUE.
  state=state+1
endif

END SUBROUTINE

C
C ************************************************************
C
C SUBROUTINE rotate(n,A,k,l,i,j,c,s)
C rotate elements A(k,l) and A(i,j) with c=cos, s=sin

INTEGER n,k,l,i,j
DOUBLE PRECISION c,s
DOUBLE PRECISION A(n,n)
DOUBLE PRECISION tmp;
tmp=A(k,l)
A(k,l)=c*tmp -s*A(i,j)
A(i,j)=s*tmp +c*A(i,j)

END SUBROUTINE

C
C ************************************************************
```
The use of Green function for solving differential equations

Green’s function $G(x, \xi)$ is defined as an integral kernel of a linear operator which inverts a differential operator. It can be defined thus as:

$$LG(x, \xi) + \delta(x - \xi) = 0 \quad x, \xi \in \mathbb{R}^n$$  \hspace{0.5cm} (C.1)

Where $L$ is a linear differential operator, $\xi$ is an arbitrary point in $\mathbb{R}^n$ and $\delta$ is the Dirac Delta function. The Green function can be used to solve (weakly) a differential equation of the form:

$$Lu(x) = \psi(x)$$  \hspace{0.5cm} (C.2)

If the differential equation is accompanied by appropriate boundary conditions, and if the green function corresponding to $L$ is known the solution of C.2 has the following integral representation:

$$u(x) = - \int_{\Omega} G(x, \xi) \psi(\xi) \, d\xi$$  \hspace{0.5cm} (C.3)

The demonstration is straightforward. Multiplying equation C.1 by $\psi(\xi)$ and integrating over the bounded space $\Omega$ with respect to $\xi$ we have:

$$\int_{\Omega} \delta(x - \xi) \psi(\xi) \, d\xi = - \int_{\Omega} LG(x, \xi) \psi(\xi) \, d\xi$$  \hspace{0.5cm} (C.4)

Due to the Dirac delta function property, the first term is simply:

$$\int_{\Omega} \delta(x - \xi) \psi(\xi) \, d\xi = \psi(x)$$  \hspace{0.5cm} (C.5)

Or, recalling the differential equation definition C.2:

$$\int_{\Omega} \delta(x - \xi) \psi(\xi) \, d\xi = Lu(x)$$  \hspace{0.5cm} (C.6)

Hence, equation C.4 may be rewritten as:

$$Lu(x) = - \int_{\Omega} LG(x, \xi) \psi(\xi) \, d\xi$$  \hspace{0.5cm} (C.7)

Since $L$ is a linear differential operator, which does not act on the variable of integration we may write:

$$u(x) = - \int_{\Omega} G(x, \xi) \psi(\xi) \, d\xi$$  \hspace{0.5cm} (C.8)
Which is the integral representation presented above for $u(x)$. However, to evaluate this integral knowledge of the explicit form of both $G$ and $\psi$ are required. Furthermore, even if both $G$ and $\psi$ are known, the associated integral may not be a trivial exercise. In addition, every linear differential operator does not admit a Green's Function. It would also be prudent to point out that in general, Green's functions are distributions rather than classical functions.
Appendix D

A0 line full lattice geometry and coordinates

Tables D.2 and D.1, as well as figure D.1 are taken from T. Koeth thesis[27].
## Table D.1: Coordinates of beam line elements at A0 in the emittance exchange configuration

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>s Position [mm]</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Photo cathode</td>
<td>0</td>
<td>not shared</td>
</tr>
<tr>
<td>BPM1</td>
<td>305</td>
<td></td>
</tr>
<tr>
<td>H&amp;VT A9C</td>
<td>3662</td>
<td>H &amp; V Trim dipole</td>
</tr>
<tr>
<td>H&amp;VTBX03</td>
<td>3706</td>
<td>H &amp; V Trim dipole</td>
</tr>
<tr>
<td>X3</td>
<td>3765</td>
<td>OTR &amp; slits</td>
</tr>
<tr>
<td>Q1AX03</td>
<td>3893</td>
<td>Type: blue</td>
</tr>
<tr>
<td>BPM2</td>
<td>3956</td>
<td>not shared</td>
</tr>
<tr>
<td>X4</td>
<td>4149</td>
<td>pneumatic OTR</td>
</tr>
<tr>
<td>X5</td>
<td>4563</td>
<td>pneumatic YAG</td>
</tr>
<tr>
<td>Q1AX05</td>
<td>4739</td>
<td>Type: blue</td>
</tr>
<tr>
<td>X6</td>
<td>5059</td>
<td>pneumatic OTR</td>
</tr>
<tr>
<td>Q2AX06</td>
<td>5358</td>
<td>Type: green</td>
</tr>
<tr>
<td>BPM3</td>
<td>5424</td>
<td>not shared</td>
</tr>
<tr>
<td>X7</td>
<td>5623</td>
<td>OTR &amp; slits</td>
</tr>
<tr>
<td>X8</td>
<td>7927</td>
<td>YAG</td>
</tr>
<tr>
<td>Q1AX08</td>
<td>8061</td>
<td>Type: green</td>
</tr>
<tr>
<td>Q2AX08</td>
<td>8210</td>
<td>Type: green</td>
</tr>
<tr>
<td>BPM4</td>
<td>8292</td>
<td>shared with BPM24</td>
</tr>
<tr>
<td>Q3AX08</td>
<td>8359</td>
<td>Type: green</td>
</tr>
<tr>
<td>X9</td>
<td>8492</td>
<td>pneumatic OTR, streak camera</td>
</tr>
<tr>
<td>X10</td>
<td>8754</td>
<td>pneumatic OTR</td>
</tr>
<tr>
<td>X12</td>
<td>9131</td>
<td>pneumatic OTR</td>
</tr>
<tr>
<td>X13</td>
<td>9294</td>
<td>pneumatic OTR</td>
</tr>
<tr>
<td>X14</td>
<td>9939</td>
<td>pneumatic OTR</td>
</tr>
<tr>
<td>Q1AX14</td>
<td>10065</td>
<td>Type: green</td>
</tr>
<tr>
<td>Q2AX14</td>
<td>10230</td>
<td>Type: green</td>
</tr>
<tr>
<td>BPM6</td>
<td>10304</td>
<td>shared with BPM25</td>
</tr>
<tr>
<td>Q3AX14</td>
<td>10396</td>
<td>Type: green</td>
</tr>
<tr>
<td>X15</td>
<td>10504</td>
<td>pneumatic OTR</td>
</tr>
<tr>
<td>BPM7</td>
<td>10786</td>
<td>shared with BPM26</td>
</tr>
<tr>
<td>Spectrometer input edge</td>
<td>11036</td>
<td>soft edge value</td>
</tr>
<tr>
<td>Spectrometer output edge</td>
<td>11327</td>
<td>soft edge value</td>
</tr>
<tr>
<td>BPM8</td>
<td>11468</td>
<td>Disp. 240 mm, shared with BPM30</td>
</tr>
<tr>
<td>X83</td>
<td>11574</td>
<td>YAG, Dispersion 324 mm</td>
</tr>
<tr>
<td>ELEMENT</td>
<td>s Position [mm]</td>
<td>Notes</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>Photocathode</td>
<td>0</td>
<td>not shared</td>
</tr>
<tr>
<td>BPM1</td>
<td>305</td>
<td>H &amp; V Trim dipole</td>
</tr>
<tr>
<td>H&amp;VTA9C</td>
<td>3662</td>
<td>H &amp; V Trim dipole</td>
</tr>
<tr>
<td>H/VTBX03</td>
<td>3706</td>
<td>Start of EEX line</td>
</tr>
<tr>
<td>X3</td>
<td>3765</td>
<td>OTR &amp; slits</td>
</tr>
<tr>
<td>Q1AX03</td>
<td>3893</td>
<td>Type: blue</td>
</tr>
<tr>
<td>BPM2</td>
<td>3956</td>
<td>not shared</td>
</tr>
<tr>
<td>X4</td>
<td>4149</td>
<td>pneumatic OTR</td>
</tr>
<tr>
<td>X5</td>
<td>4563</td>
<td>pneumatic YAG</td>
</tr>
<tr>
<td>Q1AX05</td>
<td>4739</td>
<td>Type: blue</td>
</tr>
<tr>
<td>X6</td>
<td>5059</td>
<td>pneumatic OTR</td>
</tr>
<tr>
<td>Q2AX06</td>
<td>5358</td>
<td>Type: green</td>
</tr>
<tr>
<td>BPM3</td>
<td>5424</td>
<td>not shared</td>
</tr>
<tr>
<td>X7</td>
<td>5623</td>
<td>OTR &amp; slits</td>
</tr>
<tr>
<td>D1 input edge</td>
<td>5828</td>
<td>soft edge value</td>
</tr>
<tr>
<td>D1 output edge</td>
<td>6121</td>
<td>soft edge value</td>
</tr>
<tr>
<td>BPM24</td>
<td>6445</td>
<td>shared with BPM4</td>
</tr>
<tr>
<td>D2 input edge</td>
<td>6694</td>
<td>soft edge value</td>
</tr>
<tr>
<td>D2 output edge</td>
<td>6986</td>
<td>soft edge value</td>
</tr>
<tr>
<td>BPM25</td>
<td>7081</td>
<td>shared with BPM6</td>
</tr>
<tr>
<td>X21</td>
<td>7195</td>
<td>OTR &amp; YAG</td>
</tr>
<tr>
<td>T M110</td>
<td>7602</td>
<td>center of mid cell</td>
</tr>
<tr>
<td>X22</td>
<td>8007</td>
<td>OTR &amp; YAG</td>
</tr>
<tr>
<td>BPM26</td>
<td>8122</td>
<td>shared with BPM7</td>
</tr>
<tr>
<td>D3 input edge</td>
<td>8217</td>
<td>soft edge value</td>
</tr>
<tr>
<td>D3 output edge</td>
<td>8509</td>
<td>soft edge value</td>
</tr>
<tr>
<td>BPM27</td>
<td>8758</td>
<td>shared with BPM10</td>
</tr>
<tr>
<td>D4 input edge</td>
<td>9082</td>
<td>soft edge value</td>
</tr>
<tr>
<td>D4 output edge</td>
<td>9374</td>
<td>soft edge value</td>
</tr>
<tr>
<td>X23</td>
<td>9582</td>
<td>OTR &amp; slits</td>
</tr>
<tr>
<td>Q1AX23</td>
<td>9713</td>
<td>Type: blue</td>
</tr>
<tr>
<td>BPM28</td>
<td>9782</td>
<td>shared with BPM9</td>
</tr>
<tr>
<td>Q2AX23</td>
<td>9852</td>
<td>Type: blue</td>
</tr>
<tr>
<td>Q3AX23</td>
<td>9993</td>
<td>Type: blue</td>
</tr>
<tr>
<td>X24</td>
<td>10147</td>
<td>OTR &amp; YAG, streak camera</td>
</tr>
<tr>
<td>BPM29</td>
<td>10444</td>
<td>shared with BPM5</td>
</tr>
<tr>
<td>EEX Spectrometer input edge</td>
<td>10546</td>
<td>soft edge value</td>
</tr>
<tr>
<td>EEX Spectrometer output edge</td>
<td>10836</td>
<td>soft edge value</td>
</tr>
<tr>
<td>BPM30</td>
<td>11740</td>
<td>shared with BPM30, Disp. 865 mm</td>
</tr>
<tr>
<td>X84</td>
<td>11845</td>
<td>pneumatic OTR, Disp. 944 mm</td>
</tr>
</tbody>
</table>

Table D.2: Coordinates of beam line elements at A0 in the straight ahead configuration
Figure D.1: Scheme of beam line elements at A0
Appendix E

More exemples of the rotation algorithm benchmarking

E.1 Exemple of an ellipsoid at rest

The simulation was done with the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution type</td>
<td>Uniform ellipsoid</td>
</tr>
<tr>
<td>Distribution $\sigma_x$</td>
<td>1 mm</td>
</tr>
<tr>
<td>Distribution $\sigma_y$</td>
<td>1 mm</td>
</tr>
<tr>
<td>Distribution $\sigma_z$</td>
<td>3 mm</td>
</tr>
<tr>
<td>Distribution energy $\mathcal{E}$</td>
<td>0 MeV</td>
</tr>
<tr>
<td>Distribution charge $Q$</td>
<td>-0.25 nC</td>
</tr>
<tr>
<td>Rotation 1 axis</td>
<td>$Y$</td>
</tr>
<tr>
<td>Rotation 1 angle</td>
<td>$-\pi/3$</td>
</tr>
<tr>
<td>Rotation 2 axis</td>
<td>$Z$</td>
</tr>
<tr>
<td>Rotation 2 angle</td>
<td>$-\pi/4$</td>
</tr>
</tbody>
</table>

3D grid used $N_{*0} = 5, N_{*f} = 20, N_{*2} = 32$

Table E.1: Parameters used for the benchmarking of the rotation algorithm - exemple of an ellipsoid at rest with two rotations

Figure E.1: Ellipsoid at rest with two rotations: the two rotation steps
APPENDIX E. MORE EXEMPLES OF THE ROTATION ALGORITHM BENCHMARKING

Figure E.2: Ellipsoid at rest with two rotations: canonical directions

Figure E.3: Rotation algorithm - Longitudinal Field. (a) Before the rotation of the fields - (b) After the rotation of the fields - (c) Fields from a distribution not rotated

E.2 Exemple of a sphere at low energy

The simulation was done with the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution type</td>
<td>Uniform ellipsoid</td>
</tr>
<tr>
<td>Distribution $\sigma_x$</td>
<td>1 mm</td>
</tr>
<tr>
<td>Distribution $\sigma_y$</td>
<td>1 mm</td>
</tr>
<tr>
<td>Distribution $\sigma_z$</td>
<td>3 mm</td>
</tr>
<tr>
<td>Distribution energy $\mathcal{E}$</td>
<td>0.1 MeV</td>
</tr>
<tr>
<td>Distribution charge $Q$</td>
<td>-0.25 nC</td>
</tr>
<tr>
<td>Rotation axis</td>
<td>X</td>
</tr>
<tr>
<td>Rotation angle</td>
<td>$\pi/6$</td>
</tr>
<tr>
<td>3D grid used</td>
<td>$N_{x0}=5$, $N_{xf}=20$, $N_{x2}=32$</td>
</tr>
</tbody>
</table>

Table E.2: Parameters used for the benchmarking of the rotation algorithm - exemple of a sphere at low energy
APPENDIX E. MORE EXAMPLES OF THE ROTATION ALGORITHM BENCHMARKING

Figure E.4: Sphere at low energy: canonical directions

Figure E.5: Rotation algorithm - Longitudinal Field along longitudinal direction. (a) Before the rotation of the fields - (b) After the rotation of the fields - (c) Fields from a distribution not rotated

Figure E.6: Rotation algorithm - Longitudinal Field along transverse direction. (a) Before the rotation of the fields - (b) After the rotation of the fields - (c) Fields from a distribution not rotated
Figure E.7: Rotation algorithm - Transverse Field in y direction.  (a) Before the rotation of the fields - (b) After the rotation of the fields - (c) Fields from a distribution not rotated
Useful mathematical formulas

<table>
<thead>
<tr>
<th>$f$</th>
<th>$f(z)$</th>
<th>$\Omega$</th>
<th>$\frac{df}{dz}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{arccos/acos/cos}^{-1}$</td>
<td>$\frac{1}{2} \pi + i \ln \left(iz + \sqrt{1 - z^2}\right)$</td>
<td>$\frac{-1}{\sqrt{1 - z^2}}$</td>
<td></td>
</tr>
<tr>
<td>$\text{arcsin/asin/sin}^{-1}$</td>
<td>$\ln \left(z + \sqrt{1 + z^2}\right)$</td>
<td>$\frac{1}{\sqrt{1 + z^2}}$</td>
<td></td>
</tr>
<tr>
<td>$\text{arctan/atan/tan}^{-1}$</td>
<td>$\frac{i}{2} \left(\ln (1 - iz) - \ln (1 + iz)\right)$</td>
<td>$\frac{1}{1 + z^2}$</td>
<td></td>
</tr>
<tr>
<td>$\text{arccotg/acotg/cotg}^{-1}$</td>
<td>$\frac{i}{2} \left(\ln \left(\frac{z - i}{z}\right) - \ln \left(\frac{z + i}{z}\right)\right)$</td>
<td>$\frac{-1}{1 + z^2}$</td>
<td></td>
</tr>
<tr>
<td>$\text{argch/acosh/cosh}^{-1}$</td>
<td>$\ln \left(z + \sqrt{z^2 + 1}\right)$</td>
<td>$\frac{1}{\sqrt{z^2 - 1}}$</td>
<td></td>
</tr>
<tr>
<td>$\text{argcoth/acoth/coth}^{-1}$</td>
<td>$\frac{1}{2} \left(\ln \left(1 + \frac{1}{z}\right) - \ln \left(1 - \frac{1}{z}\right)\right)$</td>
<td>$z \neq 0$</td>
<td>$\frac{1}{1 - z^2}$</td>
</tr>
</tbody>
</table>

\[
\text{arctan}(z) = \frac{\pi}{2} - \arccos \left(\frac{z}{\sqrt{z^2 + 1}}\right) \quad (F.1)
\]

\[
\text{arctan}(x) = \text{sign}(x) \arccos \left(\frac{1}{\sqrt{x^2 + 1}}\right) \quad (F.2)
\]

\[
\arccos(z) = \frac{\pi}{2} - \arctan \left(\frac{z}{\sqrt{1 - z^2}}\right) \quad (F.3)
\]

\[
\arccos(x) = \arctan \left(\frac{\sqrt{1 - x^2}}{x}\right) \quad (F.4)
\]

\[
\ln z = \ln |z| + i \arg(z) \quad (F.5)
\]

Integrals of rational functions, from the free mathematical handbook [1]:

\[
\int r \, dx = \frac{1}{2} \left(xr + a^2 \ln (x + r)\right)
\]

\[
\int \frac{r \, dx}{x} = r - a \ln \left|\frac{a + r}{x}\right| = r - a \text{arsinh} \frac{a}{x}
\]
\[
\begin{align*}
  s &= \sqrt{x^2 - a^2} \\
  \int \frac{s \, dx}{x} &= s - a \arccos \left| \frac{a}{x} \right| \\
  \int \frac{dx}{s} &= \int \frac{dx}{\sqrt{x^2 - a^2}} = \ln \left| \frac{x + s}{a} \right| \\
  R &= \sqrt{ax^2 + bx + c} \\
  \int R \, dx &= \frac{2ax + b}{4a} R + \frac{4ac - b^2}{8a} \int \frac{dx}{R} \\
  \int \frac{dx}{R} &= \frac{1}{\sqrt{a}} \ln \left| 2\sqrt{a}R + 2ax + b \right| \quad (\text{for } a > 0) \\
  \int \frac{dx}{R} &= \frac{1}{\sqrt{a}} \arcsinh \left( \frac{2ax + b}{\sqrt{4ac - b^2}} \right) \quad (\text{for } a > 0, 4ac - b^2 > 0) \\
  \int \frac{dx}{R} &= \frac{1}{\sqrt{a}} \ln \left| 2ax + b \right| \quad (\text{for } a > 0, 4ac - b^2 = 0)
\end{align*}
\]
Longitudinal space charge - Theory vs. Simulation

Longitudinal space charge fields have been calculated theoretically and numerically with distributions of energy 15MeV and charge $Q = -0.25\text{nC}$. The notation $sr$ on the plots stands for $\sigma_r$, the radial bunch length standard deviation. Black dashed line are theoretical fields, as opposed to colored lines. A total of $N\{x, y, z\} = 64$ grids in each directions was used, including $N\{x, y, z\} = 17$ grids on each side for the boundary conditions.

G.1 Cylindrical distribution

Figure G.1: Cylindrical distribution - $E_z$ along $z$ for different $\sigma_z$ (a) $\sigma_z = 0.1\text{mm}$ (b) $\sigma_z = 0.2\text{mm}$
Figure G.2: Cylindrical distribution - $E_z$ along $z$ for different $\sigma_z$  
(a) $\sigma_z = 0.3\text{mm}$ (b) $\sigma_z = 0.4\text{mm}$
### G.2 Ellipsoidal distribution

![Figure G.3](image1)

**Figure G.3:** Ellipsoidal distribution - $E_z$ along $z$ for different $\sigma_z$ (a) $\sigma_z = 0.1\text{mm}$ (b) $\sigma_z = 0.2\text{mm}$.

![Figure G.4](image2)

**Figure G.4:** Ellipsoidal distribution - $E_z$ along $z$ for different $\sigma_z$ (a) $\sigma_z = 0.3\text{mm}$ (b) $\sigma_z = 0.4\text{mm}$. 
Astra input files for the two accelerator lines

H.1 A0 photo injector

```
&NEWRUN
   Head='A0-EEX line //astra 2'
   Version=2
   Run=1
   NLoop=0,
   Distribution = 'radial1k_short.ini'
   Xoff=0.0,
   Yoff=0.0,
   Zoff=0.0,
   Lmagnetized=.F
   EmitS=.T
   PhaseS=.T
   TrackS=.T
   RefS=.T
   TcheckS=.T
   CathodeS=.T
   SigmaS=.T
   TRACK_ALL=.T
   PHASE_SCAN=.F
   AUTO_PHASE=.T
   check_ref_part=.F
   Max_Step=3000000
   H_max=0.05
   H_min=0.005
   QBunch = 0.25
/

&OUTPUT
   ZSTART=0.0
   ZSTOP=12.00
   Zemit=500
   Zphase=1
   Local_Emit=.F
   High_res=.T
   LandFS=.T
   !----------------- beam density monitor -----------------
   Screen (1) = 0.470 ! cross X1
   Screen (2) = 0.630 ! cross X2  (ceramic)
   Screen (3) = 3.765 ! cross X3  (slits H: 0.5, 1.0 mm, V: 1.0 mm)
   Screen (4) = 4.149 ! cross X4
```
APPENDIX H. ASTRA INPUT FILES FOR THE TWO ACCELERATOR LINES

```plaintext
Screen (5) = 4.563 ! cross X5
Screen (6) = 5.358 ! cross X6
Screen (7) = 5.623 ! cross X7
Screen (8) = 7.195 ! cross X21
Screen (9) = 8.007 ! cross X22
Screen (10) = 9.582 ! cross X23
Screen (12) = 10.147 ! cross X24
Screen (13) = 11.845 ! cross XS4

&SCAN
LScan = F
Scan_para = 'MaxB(1)'
S_min = 0, S_max = 0.20, S_numb = 10
FOM(1) = 'bunch charge',
FOM(2) = 'mean energy',
FOM(3) = 'rms energy',
FOM(4) = 'hor spot',
FOM(5) = 'ver spot',
FOM(6) = 'hor emit',
FOM(7) = 'ver emit',
FOM(8) = 'bunch length'
/

&MODULES
/

&ERROR
/

&CHARGE
! L2D_3D = T
! Z_trans = 5
LSPCH = F
Max_scale = 0.1
Max_Count = 100
! 2D space charge (used when LSPCH=T)
Nrad = 7,
Nlong_in = 10
Cell_var = 2.0
min_grid = 0.4D-6
Lmirror = T
Lmirror = T
Linert = F
! 3D space charge (used when LSPCH=T and LSPCH3d=T)
LSPCH3D = F
Nxf = 32, Nxf = 32, Nzfm = 32
Smooth_x = 1
Smooth_y = 1
Smooth_z = 1
/

&CSR
/

&Aperture
/

&CAVITY
LEfield = T
```

August 2009 129 Emmanuel Branlard
APPENDIX H. ASTRA INPUT FILES FOR THE TWO ACCELERATOR LINES

File_Efield (1) = '/fields/ysbalanced.txt',
Nue(1) = 1.3, MaxE(1) = -35, Phi(1) = -12.00, C_pos(1) = 0,
File_Efield (2) = '/fields/cavityMAF.dat',
Nue(2) = 1.3, MaxE(2) = -23, Phi(2) = -9, C_pos(2) = 1.896,

SOLENOID
LBfield = T,
File_Bfield (1) = '/fields/bz_b8_p164_s119',
MaxB(1) = 0.126, S_smooth(1) = 20, S_pos(1) = 0,
File_Bfield (2) = '/fields/bz_b0_p150_s0.bzw',
MaxB(2) = 1.402450e-01, S_smooth(2) = 10, S_pos(2) = 0,

QUADRUPOLE
Loop = F,
Lquad = T,
Q1AX03
Q_pos(1) = 3.873, Q_grad(1) = 0.00, Q_bore(1) = 0.0020, Q_length(1) = 0.102
! Q2AX03
Q_pos(2) = 4.020, Q_grad(2) = 15.742631318, Q_bore(2) = 0.0020, Q_length(2) = 0.102, Q_type(2) = 'sk'
! S1AX04
Q_pos(3) = 4.371, Q_grad(3) = -16.62868254, Q_bore(3) = 0.0020, Q_length(3) = 0.102, Q_type(3) = 'sk'
! S1AX05
Q_pos(4) = 4.722, Q_grad(4) = -10.02350185, Q_bore(4) = 0.0020, Q_length(4) = 0.102, Q_type(4) = 'sk'
! S1AX06
Q_pos(5) = 5.224, Q_grad(5) = 38.10078156, Q_bore(5) = 0.0020, Q_length(5) = 0.102, Q_type(5) = 'sk'
end of skew quadrupoles section

Q1AX06
Q_pos(6) = 5.354, Q_grad(6) = 0.00, Q_bore(6) = 0.0020, Q_length(6) = 0.102
! Q2AX06
Q_pos(7) = 5.492, Q_grad(7) = 0.00, Q_bore(7) = 0.0020, Q_length(7) = 0.102
! chicane is here

Q1AX08
APPENDIX H. ASTRA INPUT FILES FOR THE TWO ACCELERATOR LINES

```
159 Q_pos(8) = 7.609, Q_grad(8) = 0.00, Q_bore(8) = 0.0020, Q_length(8) = 0.102
160 Q_pos(9) = 7.757, Q_grad(9) = 0.00, Q_bore(9) = 0.0020, Q_length(9) = 0.102
161 Q_pos(10) = 7.905, Q_grad(10) = 0.00, Q_bore(10) = 0.0020, Q_length(10) = 0.102
162 ! Q2AX08
163 Q_pos(11) = 8.949, Q_grad(11) = 0.00, Q_bore(11) = 0.0020, Q_length(11) = 0.102
164 Q_pos(12) = 9.090, Q_grad(12) = 0.00, Q_bore(12) = 0.0020, Q_length(12) = 0.102
165 ! Q2AX11
166 Q_pos(13) = 9.231, Q_grad(13) = 0.00, Q_bore(13) = 0.0020, Q_length(13) = 0.102
167 ! Q1AX14
168 Q_pos(14) = 10.085, Q_grad(14) = 0.00, Q_bore(14) = 0.0020, Q_length(14) = 0.102
169 ! Q2AX14
170 Q_pos(15) = 10.253, Q_grad(15) = 0.00, Q_bore(15) = 0.0020, Q_length(15) = 0.102
171 ! Q3A14
172 Q_pos(16) = 10.421, Q_grad(16) = 0.00, Q_bore(16) = 0.0020, Q_length(16) = 0.102
173 /
174 DIPOLE
175 Loop = F
176 Ldipole = T
177 ! geometry modified to avoid overlap
178 D_Type(1) = 'HOR'
179 D1(1) = (-0.0717, 5.8088)
180 D2(1) = (0.1115, 5.8579)
181 D3(1) = (-0.0717, 6.0881)
182 D4(1) = (0.1115, 6.039)
183 D_Gap(1, 1) = 0.0573
184 D_Gap(2, 1) = 0.0573
185 D_radius(1) = -0.598
186 D_Type(2) = 'HOR'
187 D1(2) = (0.1596, 6.674)
188 D2(2) = (0.4136, 6.6077)
189 D3(2) = (0.1596, 6.9129)
190 D4(2) = (0.4136, 6.8466)
191 D_Gap(1, 2) = 0.0573
192 D_Gap(2, 2) = 0.0573
193 D_radius(2) = 0.598
194 D_Type(3) = 'HOR'
195 D1(3) = (0.2612, 8.1775)
196 D2(3) = (0.5122, 8.1112)
197 D3(3) = (0.2612, 8.4164)
198 D4(3) = (0.5122, 8.3501)
199 D_Gap(1, 3) = 0.0573
```
! geometry modified to avoid overlap
D_Type(4) = 'HOR'
D1(4) = (0.5539, 8.9876)
D2(4) = (0.7371, 8.9385)
D3(4) = (0.5539, 9.1687)
D4(4) = (0.7371, 9.2178)
D_Gap(1,4) = 0.0573
D_Gap(2,4) = 0.0573
D_r radius (4) = 0.595
!
D_strength = 0.15
/

H.2 ILCTA at NML

! Nota: the quads are forced to follow a hard edge model with length corresponding to the estimated magnetic length.
!
Head= 'NML';
Version=2
Run=1
NLoop = 0,
Distribution = 'radial1k_short.ini'
Xoff = 0.0,
Yoff = 0.0,
Zoff = 0.0,
Lmagnetized = .F
EmitS = .T
C_Emits = .T,
Lsub_rot = .T,
Lsub_cor = .T,
PhaseS = .T
TrackS = .T
RefS = .T
TcheckS = .T
CathodeS = .T
SigmaS = .T
TRACK_ALL = .T
PHASE_SCAN = .F
AUTO_PHASE = .T
check_ref_part = .F
Max_Step = 3000000
H_max = 0.05
H_min = 0.005
QBunch = 3.2
/

&OUTPUT
ZSTART = 0.0
ZSTOP = 15.00
Zemit = 500
APPENDIX H. ASTRA INPUT FILES FOR THE TWO ACCELERATOR LINES

39  Zphase=1
40  Local_Emit=.F
41  High_res=.T
42  LandFS=.T
43  ! __________ beam density monitor ___________
44  ! Screen (1) = 0.470  ! cross X1
45  ! Screen (2) = 0.630  ! cross X2  (ceramic)
46  ! Screen (3) = 3.765  ! cross X3  (slits H: 0.5, 1.0 mm, V: 1.0 mm)
47  ! Screen (4) = 4.149  ! cross X4
48  ! Screen (5) = 4.563  ! cross X5
49  ! Screen (6) = 5.358  ! cross X6
50  ! Screen (7) = 5.623  ! cross X7
51  ! Screen (8) = 7.195  ! cross X21
52  ! Screen (9) = 8.007  ! cross X22
53  ! Screen (10) = 9.582  ! cross X23
54  ! Screen (11) = 10.147 ! cross X24
55  ! Screen (12) = 10.145 ! cross X54
56
57  &SCAN
58  LScan=.F
59  ! Scan_para='MaxB(1)'
60  ! Scan_para='Phi(1)'
61  ! Scan_para='Phi(1)'
62  ! Scan_min=-180,  S_max=180,  S_numb=36
63  ! S_min=0.08,  S_max=0.2,  S_numb=25
64  ! FOM(1)='bunch charge',
65  ! FOM(2)='mean energy',
66  ! FOM(3)='rms energy',
67  ! FOM(4)='hor spot',
68  ! FOM(5)='ver spot',
69  ! FOM(6)='hor emit',
70  ! FOM(7)='ver emit',
71  ! FOM(8)='bunch length',
72
73  &MODULES
74
75  &ERROR
76
77  &CHARGE
78  ! L2D_3D=.T
79  ! Z_trans=7
80  LSPhC=.F
81  Max_scale=0.1
82  Max_Count=100
83  ! 2D space charge (used when LSPCH=T)
84  Nrad=7,
85  Nlong_in=10
86  Cell_var=2.0
87  min_grid=0.4D-6
88  Lmirror=.T
89  Linert=.F
90  ! 3D space charge (used when LSPCH=T and LSPCH3d=T)
91  LSPCH3D=.F
92  Nxf=32,  Nyf=32,  Nzf=32
93  Smooth_x=1
94
95 August 2009  133  Emmanuel Branlard
APPENDIX H. ASTRA INPUT FILES FOR THE TWO ACCELERATOR LINES

\[ \text{Smooth}_y=1 \]
\[ \text{Smooth}_z=1 \]
\[
\&\text{CSR} \\
\&\text{Aperture} \\
\&\text{CAVITY} \\
\text{Loop} = F, \text{LE field} = T \\
\text{!cavity gun} \\
\text{File\_Efield}(1) = '/fields/cavity\_desygun.dat', \text{C\_smooth}(1)=20, \text{C\_HIGHER\_ORDER}(1)=T \\
\text{Nue}(1) = 1.3, \text{MaxE}(1) = -36, \text{Phi}(1)=0, \text{C\_pos}(1)=0.0, \\
\text{!CC1 - first 9 cells} \\
\text{File\_Efield}(2) = '/fields/cavityMAF.dat', \text{C\_smooth}(2)=20, \text{C\_HIGHER\_ORDER}(2)=T \\
\text{Nue}(2) = 1.3, \text{MaxE}(2) = -24.0, \text{Phi}(2)=0, \text{C\_pos}(2)=2.800, \\
\text{!CC2 - second 9 cells} \\
\text{File\_Efield}(3) = '/fields/cavityMAF.dat', \text{C\_smooth}(3)=20, \text{C\_HIGHER\_ORDER}(3)=T \\
\text{Nue}(3) = 1.3, \text{MaxE}(3) = -48, \text{Phi}(3)=0, \text{C\_pos}(3)=5.511, \\
\text{!SOLENOID} \\
\text{Loop} = T, \text{LB field} = T \\
\text{! bucking solenoid} \\
\text{File\_Bfield}(1) = '/fields/ilcta\_main\_100A.bz', \text{MaxB}(1) = -0.031993, \text{S\_smooth}(1)=20, \text{S\_pos}(1)=-0.0949 \\
\text{! main solenoid} \\
\text{File\_Bfield}(2) = '/fields/ilcta\_main\_100A.bz', \text{MaxB}(2) = 0.1614, \text{S\_smooth}(2)=20, \text{S\_pos}(2)=0.21517 \\
\text{! other config} \\
\text{! File\_Bfield}(1) = '/fields/solenoid\_nml\_198.dat', \text{MaxB}(1) = 0.15, \text{S\_smooth}(1)=10, \text{S\_pos}(1)=0.0 \text{! valley of solenoid scan for emit but not smooth} \\

August 2009 134 Emmanuel Branlard
APPENDIX H. ASTRA INPUT FILES FOR THE TWO ACCELERATOR LINES

```plaintext
! MaxB(1) = 0.16, S_smooth(1) = 10, S_pos(1) = 0.0 ! smooth valley of solenoid scan

QUADRUPOLE
Loop = F
Lquad = F

! Q103
Q_pos(1) = 10.7639, Q_grad(1) = -2.00, Q_bore(1) = 0.0020,
Q_length(1) = 0.113

! Q104
Q_pos(2) = 10.9647, Q_grad(2) = 2.00, Q_bore(2) = 0.0020,
Q_length(2) = 0.113

! Q105
Q_pos(3) = 11.1655, Q_grad(3) = 2.00, Q_bore(3) = 0.0020,
Q_length(3) = 0.113

DIPOLe
Loop = F
Ldipole = T

D_Type(1) = 'HOR'
D1(1) = (-0.24, 11.6505)
D2(1) = (0.24, 11.6505)
D3(1) = (-0.24, 11.8805)
D4(1) = (0.24, 11.8805)
D_Gap(1,1) = 0.053
D_Gap(2,1) = 0.053
D_radius(1) = 0.7374000000

D_Type(2) = 'HOR'
D1(2) = (-0.5276, 12.5358)
D2(2) = (-0.0476, 12.5358)
D3(2) = (-0.5276, 12.7658)
D4(2) = (-0.0476, 12.7658)
D_Gap(1,2) = 0.053
D_Gap(2,2) = 0.053
D_radius(2) = -0.7374000000

D_Type(3) = 'HOR'
D1(3) = (-0.5276, 13.5556)
D2(3) = (-0.0476, 13.5556)
D3(3) = (-0.5276, 13.7856)
D4(3) = (-0.0476, 13.7856)
D_Gap(1,3) = 0.053
D_Gap(2,3) = 0.053
D_radius(3) = -0.7395000000

D_Type(4) = 'HOR'
D1(4) = (-0.24, 14.4408)
D2(4) = (0.24, 14.4408)
D3(4) = (-0.24, 14.6708)
D4(4) = (0.24, 14.6708)
```

August 2009 135 Emmanuel Branlard
<table>
<thead>
<tr>
<th>Line</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>D_Gap(1,4) = 0.053</td>
</tr>
<tr>
<td>22</td>
<td>D_Gap(2,4) = 0.053</td>
</tr>
<tr>
<td>23</td>
<td>D_radius(4) = 0.7395000</td>
</tr>
</tbody>
</table>

/*
Automatizing and organizing the simulations

Three files related to Astra are here discussed:

- the input file for generator - ex: gener-sphere.in
- the distribution file, generated by generator, containing the bunch particles - ex: ellipse113-Ek40-Xpi6.ini
- the input file for Astra, that we will indifferently call simulation file of case file - ex: dipole.in

I.1 On the need to get organized

Astra offers a large set of parameters that can be defined in Astra and generator input files. Nevertheless, it can be tedious to change each of them one by one, especially if one wants to vary a parameter for a lot of different values. For the simulation to be reproducible, and to avoid mistakes, the best solution is to have one file per simulation. The generation of these files by hand is a loss of time, and providing a different, but canonical name to each of them, is tedious. On top of that, the case file is dependent of the distribution file (the 'NEWRUN' section requires the parameter 'Distribution'), whereas we would like to study a case for several distributions. Once again, generating a case file for each distribution multiply the number of files! And eventually, we were comparing different space charge algorithm, requiring us to execute different versions of Astra to be able to compare them.

It then appeared clear that distinctions had to be made to clarify the simulation, and translate better the structure and link between the different objects. We start with the hypothesis that few parameters of a case file will vary from one simulation to another. These parameters will mainly be: Distribution, Nx, Ny, Nz, Lspch, Lspch3D, Qbunch, Phi(1). Allowing us to setup the grid, study a distribution with or without space charge calculation, in 2D or 3D. We came out with the following distinction:

- **cases file** are the file defining the global geometry: how many components and where. They are stored in the 'cases/' folder. They are never changed manually, to keep simulations reproducible.
- The **distribution file** are stored in the 'bunches/' folder.
- The **parameter string** defines which parameters will be changed in the case file.
- The **algorithm string** defines which algorithm will be run to perform the calculation.

In summary we define a simulation as: running an algorithm, with a specific input distribution, for a specific case where different parameters have been changed. The architecture chosen to store output results is:

```
CASE / DISTRIBUTION / PARAMETER-STRING / ALGORITHM
```
Appendix I. Automatizing and Organizing the Simulations

Let’s give an example: the case file dipole.in, contain a dipole at a specific position. We want to test it with the input distribution ellipse113-Ek15.ini, with the algorithm astra-v18 (see J for a description of the versions). We would like to compare the results between Qbunch=10 with space charge and Qbunch=3.2 without space charge. The parameter string will be in the first case Qbunch=10-LSPCH=T and in the second case: Qbunch=3.2-LSPCH=F. The operator writes in the terminal

\$ sim\_paramcomp astra-v18 dipole ellipse113-Ek15 Qbunch=10 LSPCH=T
\$ sim\_paramcomp astra-v18 dipole ellipse113-Ek15 Qbunch=3.2 LSPCH=F

The folders generated to store the results are:

dipole / ellipse113-Ek15 / Qbunch=10-LSPCH=T / astra-v18

dipole / ellipse113-Ek15 / Qbunch=3.2-LSPCH=F / astra-v18

The script sim\_paramcomp is described below in this chapter. As we will further see, shortnames have been implemented to shorten the parameter string. For instance 3D will activate the 3D space charge algorithm, and mgrid, will setup a medium grid (Nxf = 32, Nyf = 32, Nzf = 32). The scripts used to change parameters in case file, generate a serie of different distribution, process the overall simulation depending on the parameter string input by the user are described below.

I.2 Changing parameters in any Astra input file

Astra input files follow the structure of Fortran modules. Fields are coma separated, or written line by line. Comments are done with the character: ‘!’. The script changeParam we wrote is in the awk[38] language. It can modify any parameter of an input file, even if these parameters are on the same line, as long as they are separated by a virgule. One line parameters don’t have to end with a virgule. This follows the convention used by Fortran. The script is not case sensitive.

Examples of call:

\$ awk -f changeParam casefile.in Distribution="ellipse.ini" > casefile-modified.in
\$ awk -f changeParam casefile.in Nxf=32 > casefile-modified.in
\$ awk -f changeParam casefile.in Qbunch=3.2 > casefile-modified.in
\$ awk -f changeParam casefile.in LSPCH3D=T > casefile-modified.in

```
1 #!/bin/awk
2 BEGIN {
3 IGNORECASE=1
4 FS="",""
5 split(ARGV[2], a, ",=")
6 bb=tolower(a[1]) "="
7 }
8 $0 ~ bb { print }
9 for(i=1;i<=NF;i=i+1){
10 .buff=$i;
11  if($i!="!") { split($i,c,"!"); print "!"c[2]; buff=c[1]}
12  if(buff"bb") {
13   print ""a[1]""=a[2]
14  } else {
15   if(length(buff)>0){ gsub( "", "", buff); print " "
16     buff }
17  }
18 }
19 $0 !~ bb { print }
```

August 2009 138 Emmanuel Branlard
I.3 Application to generate a series of input distribution

The script changeParam can be used to change parameter in the case file, but also in the generator input file. For instance, we wrote the following shell script [25] to generate 16 cylindrical distributions with \(\sigma_r \in [1; 4]\) mm and \(\sigma_z \in [0.1; 0.4]\) mm. Note that if \(\sigma_r \in [1; 4]\) then \(\sigma_x \in [1; 4]/\sqrt{2}\). (see section 7.4).

```bash
#!/bin/sh
 generator -file="gener-cylinder.in"
 q=0.25
 ipart=50000
 ekin=15

 temp=$generator-file"-tmp";
 cp $generator-file $temp;
 awk -f changeParam $temp Q_total=$q > $generator-file ; cp $generator-file $temp;
 awk -f changeParam $temp IPart=$ipart > $generator-file ; cp $generator-file $temp;
 awk -f changeParam $temp Ref_Ekin=$ekin > $generator-file ; cp $generator-file $temp;

 for sz in 1 2 3 4
 do
   for sr in 1 2 3 4
   do
     #Compute sx value: sz=sz/sqrt(2).
     #With integer calculation we separate integer part and
     #decimal part
     delta=`expr $sr \* 1000000 / 141421 \* 1000`
     num=`expr $sr \* 1000000000 / 141421`
     int=`expr $num / 1000`
     dec=`expr $num - $delta`
     filename="cylinder"$sr"0"$sz"-Ek"$ekin"-"$ipart ".ini"
     awk -f changeParam $temp sig_x=$ent".$dec > $casefile ; cp $casefile $temp;
     awk -f changeParam $temp sig_y=$ent".$dec > $casefile ; cp $casefile $temp;
     awk -f changeParam $temp sig_z="0.0"$sz > $generator-file ; cp $generator-file $temp;
     awk -f changeParam $temp FNAME="$filename\" > $generator-file ; cp $generator-file $temp;
     echo $filename
     generator $generator-file > /dev/null
   done
 done
 rm $temp
```

I.4 Performing simulation and storing results in a corresponding folder

Examples of the script simul-paramcomp have been given above. We will here provide its corresponding source code that we wrote to perform our simulations easily and massively while keeping an organized structure for storing the results.

```bash
#!/bin/sh
 fordir ="/SpaceCharge/fortran"
bunchdir="/SpaceCharge/bunches"
outdir="/SpaceCharge/simulations/paramcomp"
casedir="/SpaceCharge/cases"
```
APPENDIX I. AUTOMATIZING AND ORGANIZING THE SIMULATIONS

```bash
#STORING DEFAULT ARGUMENTS
export algo=$1
export case=$2
export distr=$3
echo "Algorithm: "$algo
echo "Case: " :$case
echo "Distribution: " :$distr
shift: shift: shift:
casefile=$case",.in"

#CREATING CASE FILE WITH GOOD DISTRIBUTION AND STORING IT IN OUTDIR
awk -f changeParam $casedir/casefile Distribution="$bunchdir/$distr",.in
" " $casefile
params="simul"
while [ "$1" != "nn" ]
do
temp=$casefile"-tmp";
cp $casefile $temp;
case $1
in
  2D)
    echo " 2D";
    params=$params"-2D";
    awk -f changeParam $temp LSPCH=T > $casefile ; cp $casefile $temp;
    awk -f changeParam $temp LSPCH3D=F > $casefile ;;
  3D) echo " 3D";
    params=$params"-3D";
    awk -f changeParam $temp LSPCH=T > $casefile ; cp $casefile $temp;
    awk -f changeParam $temp LSPCH3D=F > $casefile ;;
  0D) echo "Desactivating space charge",
    params="$params"-NOSPCH"
    awk -f changeParam $temp LSPCH=F > $casefile ; cp $casefile $temp;
    awk -f changeParam $temp LSPCH3D=F > $casefile ;;
x sgrid | echo "XS grid"
    params=$params"-xsgrid"
    awk -f changeParam $temp Nxf=8 > $casefile ; cp $casefile $temp
    awk -f changeParam $temp Nxf=8 > $casefile ; cp $casefile $temp
    awk -f changeParam $temp Nxf=8 > $casefile ; cp $casefile $temp
    awk -f changeParam $temp Nxf=8 > $casefile ;
sgrid | echo "S grid"
    params=$params"-sgrid"
    awk -f changeParam $temp Nxf=16 > $casefile ; cp $casefile $temp
    awk -f changeParam $temp Nxf=16 > $casefile ; cp $casefile $temp
    awk -f changeParam $temp Nxf=16 > $casefile ; cp $casefile $temp
    awk -f changeParam $temp Nxf=16 > $casefile ;
mgrid | echo "M grid"
    params=$params"-mgrid"
    awk -f changeParam $temp Nxf=32 > $casefile ; cp $casefile $temp
    awk -f changeParam $temp Nxf=32 > $casefile ; cp $casefile $temp
    awk -f changeParam $temp Nxf=32 > $casefile ; cp $casefile $temp
```
awk -f changeParam $temp Nrad=16 > $casefile; cp $casefile $temp;
awk -f changeParam $temp Nlong_in=32 > $casefile ;
lgrid) echo "L grid";
params=$params"−lgrid"
awk -f changeParam $temp Nxf=64 > $casefile; cp $casefile $temp;
awk -f changeParam $temp Nyf=64 > $casefile; cp $casefile $temp;
awk -f changeParam $temp Nrad=32 > $casefile; cp $casefile $temp;
awk -f changeParam $temp Nlong_in=64 > $casefile ;;
∗) echo "Setting parameter $1";
params=$params"−"$1
awk -f changeParam $temp $1 > $casefile ;;
esac
shift;
done
echo $params
# CREATING ARBORESCENCE
mv $casefile $case"−"$params".in"
outdir=$outdir/$case/$distr/$params/algo
case=$case"−"$params".in"
echo $outdir
mkdir −p $outdir
mv $case $outdir/$case
rm $temp
# COMPILING ASTR A VERSION
echo $fordir
echo "Version :" $algo
cd $fordir/$algo
cp $outdir/$case ./
make
# RUNNING Astra
./Astra $case
# MOVING OUTPUT FILES
mv *.out $outdir
mv *.001 $outdir
rm $case
# RUNNING R FOR DATA INTERPRETATION
# cd $outdir
# echo "BatchMain.AstraPostPro ()" >run.r
# R −f run.r
echo "Done."

August 2009 141 Emmanuel Branlard
Appendix J

Versions history of the 3d space charge algorithm

v12 - directions-out: from now on, we output the fields on specific points defined in the file distrbution.ini-directions
   if distribution.ini is the input distribution file

v13: - !(TO BE IMPROVED) Adding a condition on the covariance matrix before rotating the distribution
   - trying to save nxf, nx0 etc. and to adapt them to the rotated frame, with failure (in vain)

v14: - Modifying get mesh to use the "distance to the corner method" instead of boosting and rotating
   the lab coordinates of each point when GridInterpolate is called.
   - Adding a debug parameter to activate write command
   - DEBUGGING for implementation in astra
   - IMPORTANT: change of SCGRIDINTERPOLATE to version 2.0 (takes only "Field" in arguments)
   - Merging Boost and Rot modules
   - Adding a distinction between Dump and Debug
   - Output files depend on the number of calls

v15: - Harmonization with the work of Klaus on boost matrix:
   - use of B_MAT instead of BoostMat
   - use of SuPx instead of SumPx
   - PROBLEMS of absolute/relative coordinates
     - After boost, the min and max are around Xmi_r, Ymi_r, Z_mi_r
       => we subtract the bunch center Xmi_r while boosting
     - After rotation, R_part should be center on 0
       => XgMi, etc.. initialized at 0
     - SCGridInterpolate receives arguments relative to the bunch center
       => don’t call GetMeshInfo with x-xmeanU

142
v14-alpha: Modification of v14 to take into account the solutions of the problems found when comparing v15 and v00. We don’t use Klaus notations Bmat and SuPx, so to keep harmonious with our previous work.

-function BoostDistr -> BoostAndShiftDistr (we substract Xmi_r, Ymi_r, Zmi_r)

v16: exactly the same as v14-alpha, with a higher version number because preferred to v15.

v00: This is the latest original version from Klaus (SCGridInterpolate 2.0, B_Mat, G_mat)
- I just run this version with our stand-alone main: "astraImpulse_main.f"
- For this, I had to comment the "warning: particle outside grid", because the condition generates a segmentation fault. This is the only modification to the original file.

v00a: NOTE: Field Mat is not computed correctly by Klaus in v00, there are two errors:
- IF THINK non-diagonal terms should be negative => now they are !
- Field_mat(2,2) appeared twice => not any more!

v17: this version is born from the same process that made v16:
- starting from v15 but adding progressively functions from spch_improve.f.

Indeed, the comparison of emittance between v00 v15 and v16 revealed than v16 had a problem.

v17 has then been created to debug and see which step made v16 different than v15.

It turns out the difference was in the field calculation in SCGridInterpolate, due to the fact that the field transformation is a function of gamma:
- v16 algorithm calculates the transformation coeff at each call of SCGridInterpolate
- Klaus algorithm calculates it at each call of SPCH3D

PROBLEM: Gamma changes between two calls of SCGRID... , and this effect is cumulated to the scaling (GUESSING)

- Introducing Matrice multiplication for field transform (harmony with Klaus notation)

v17a: My Field matrix (- on off diagonal turns)
v17b: My field matrix with Klaus convention of sign (+): after testing: same as v15
v17c: Klaus field matrix, should give similar result to v17b and v00a: after testing: same as v17b
v1712: Born from the fusion between v17a and v12. Indeed, it turns out the "distance to the corner method" is not working that well compared to the old fashion method implemented in v12 in the case of spherical rest bunches. In v1712, we try to improve the "distance to the corner method" in GetMeshInfo.

After analysis, I chose to mix the two methods:
- From the old method: Lab Coordinates are transformed to bunch coordinates to be compared with the corner at rest.
- From the more recent: storing the matrix LabToBunchMat=MATMUL(RotMatInv,BoostMat) in order to save computational time.

This results in:
- suppression of Corner and Lambda in X_modules_rot.f, keeping only lambdaRest and CornerRest.
- suppression of function RotInvGrid and BoostInvGrid, replaced by only one: LabToBunch_Grid, which compute the eponyme matrix.

What's new also:
- Renaming function SetRotatedGrid in SaveRotatedGrid
- astraimpulse_main.f: I ASSUME(!!!) X and Y coordinates are relative to the reference particle.

v19: Adding the file X_em_mon.f to our project, and modifying it for it to export gamma Betaz in the second column of the sigma matrix output file.

v20: Modification of X_em_mon.f to output a sigma matrix with coefficient adapted to the phase plans x-x', etc.

v20b: Output the 6 first columns of PARTICLES in a file. This is perform 50 times, with a file name equivalent to the one generated by astra when Zphase >1. We had to do this, because Astra refuses to output these files if the line is more than 10 meters long.

v24: Cleaned version, but nevertheless, we chose to let the Dump and Debug logicals. If this functionnality is not desired any more, one just has to remove all the condition "'if(lDebug)'" or "'if(lDump)'".

Note: BEFORE, all the matricial functions (diago, inverse..) were stored in the folder /fortran/matrix.
NOW, the corresponding fortran files "'.f'" are in the same folder than X_spch3d.f and spch_improve.f.
The makefile has been modified conseuquently.
Bibliography


