



ECOLE NATIONNALE SUPÉRIEURE DE L'AÉRONAUTIQUE ET DE L'ESPACE



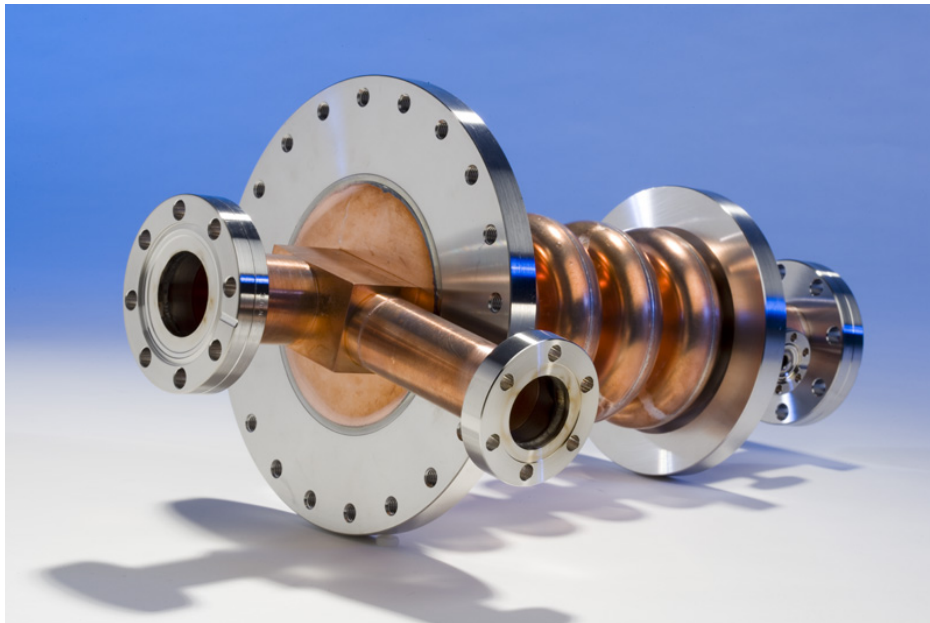
FERMI NATIONAL ACCELERATOR LABORATORY

Study of a 5 cells copper cavity at 3.9GHz

Determination of the cavity transit matrix

Emmanuel BRANLARD

August the 31st 2007



Contents

Notations	1
1 Beam dynamics and introduction to the matrix formalism	2
1.1 Introduction to the theory of beam dynamics	2
1.2 General resolution of the perturbed linear equation of motion	2
1.2.1 Vectorial form	2
1.2.2 Homogeneous equation	3
1.2.3 Particular solution by variation of constant parameter	3
1.2.4 General Solution	3
1.3 Matrix formalism	4
1.3.1 Illustration of matrix use in optics	4
1.3.2 Notation used for the cavity	4
2 Modelling the cavity with five pillboxes	6
2.1 Equation of motion in a pillbox	6
2.2 The One cell pill box model	6
2.3 The One cell pill box transit matrix	7
3 Using ASTRA to determine the cavity matrix	9
3.1 A short description of Astra	9
3.2 Description of the method	9
3.3 The physical limit of this method	9
3.4 Configuring Astra	9
3.4.1 The fields format	9
3.4.2 The fields time phase	10
3.4.3 The fields scaling	11
3.4.4 The input file format	11
3.4.5 The NEWRUN section of the run file	12
3.4.6 The CAVITY section of the run file	12
3.5 Important errors to avoid	12
4 Programs developped and resulting matrix	14
4.1 Description of the different programs	14
4.2 The final cavity matrix	16
Acknowledgements	17
Annexes	17

List of Figures

1	Diagramm of the cavity	1
2	Illustration of the matrix input variable	2
3	Illustration of the matrix formalism in optics	4
4	Ezo (in red) in MV/m and the sine function adjusted in phase without scale . .	11
5	Organigramm of the created programs	14

NOTATIONS

Numerical value and notation The following diagramm introduces the notations that will be used

c	Light velocity	299792458
f	Cavity frequency	3.9 Ghz
λ	Cavity wave length	76.87mm
ω	Cavity proper pulsation	
k	Space pulsation $kc = \omega$	81.738/m
μ_0		$\pi 10^{-7}$
$Scale$	Scaling factor for the field map	$9.3227 10^6$
ϕ	RF oscillator phase	287.9 degrees
x, y, z	Coordinates of a particle, see diagramm below	
p_x, p_y, p_z	Coordinates of a particle, see diagramm below	$p_0 = 15\text{Mev}/c$
l_c	The five cells cavity length $l_c = 5\lambda/2$	192.3mm
L_c	Cavity length including surrounding pipes	300mm
d	Surrounding pipe length	53.85mm
e	Electron charge	$1.6 10^{-19} \text{ C}$
T	Transit time factor	
P_{HFSS}	Power input to the cavity with HFSS	$5.46 10^{-16} \text{ Joules}$

in this paper. Please note that the z axis starts at the entry of the pipe before the 5 cells. Each time we don't use this convention, we will mention it.

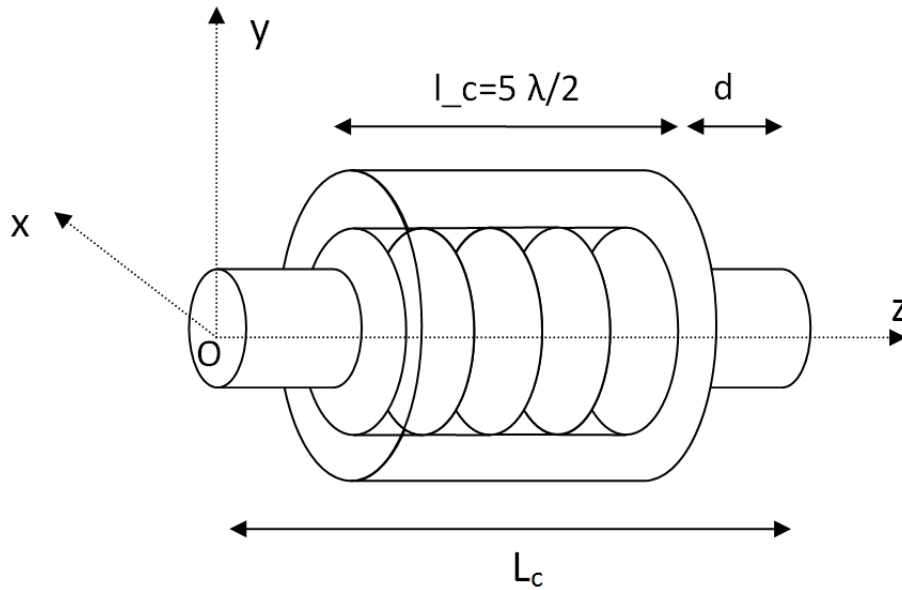


Figure 1: Diagramm of the cavity

1 Beam dynamics and introduction to the matrix formalism

1.1 Introduction to the theory of beam dynamics

In this section I will introduce a theory, which is uncomplete and thus canno't be directly applied to the cavity, but still, it helps me a lot to understand the definition of terms and the basis of the matrix formalism we were using in 6 dimensions. Indeed, the following results doesn't take in account the change of $\delta p/p$, and the z coordinate which are needed to describe the behaviour of the cavity.

In beam dynamics, most of the time we consider one particule, that we will call the reference particule, and which follows a predescribed path. The other particules have trajectories close to the reference particule. In order to simplify the equation of motion, we generally use the curvilinear coordinates system rather than cartesian. The axis, X,Y,S are following the reference particule through its motion. As a result of this, the coordinates of other particules are always relative to the position of the reference particule. When using the equation of motion we also have to be careful that the time of the particule is different of the reference particule. We will not in this paper establish the equation of motion, but we strongly recommand the reading of the section 4.2 and 4.7 of the book from H. Wiedemann [15]. We will try to remember in this section that coordinates used further are coordinates relative to the reference particule. As descibed in figure 2, when we will write x'_{in} , in fact it will correspond to a dx'_{in} and the same for the output.

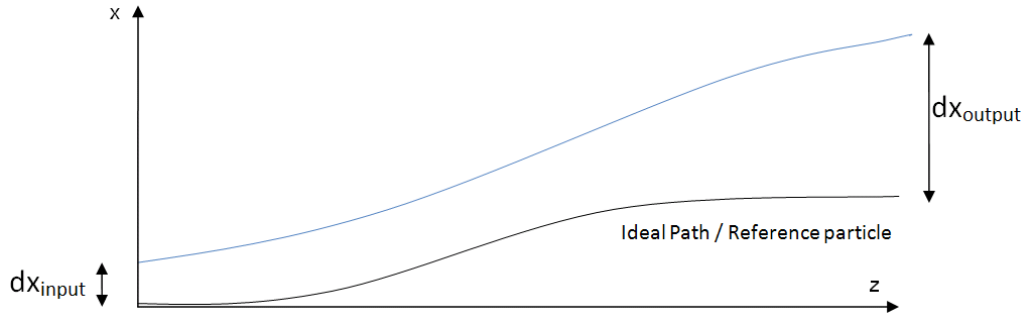


Figure 2: Illustration of the matrix input variable

1.2 General resolution of the perturbated linear equation of motion

1.2.1 Vectorial form

We consider the following motion equation where u stands for x or y, and s is the curvilign coordinate :

$$\begin{aligned} f(s) &= u'' + K(s).u \\ u(s_0) &= u_0 \\ u'(s_0) &= u'_0 \end{aligned}$$

For the resolution we will use the general method of a nth 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 the equation in vector formalism :

$$\begin{pmatrix} u' \\ u'' \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ K(s) & 0 \end{pmatrix} \cdot \begin{pmatrix} u \\ u' \end{pmatrix} + \begin{pmatrix} 0 \\ f(s) \end{pmatrix}$$

Now, we define $X = (u \ u')^T$ to write the following general linear differential equation, but the method that we will use can be written for $X = (u \ u' \ u'' \ \dots \ u^{(n)})^T$ in a more general context of nth degree equation.

$$X'(s) = A(s)X(s) + B(s)$$

1.2.2 Homogeneous equation

We first solve the Homogeneous Equation

$$X'(s) = A(s)X(s)$$

and define the resolvente matrix $R(s) = [C_1 \ \dots \ C_n]$ where $C_i = (x_i(s) \ x'_i(s) \ \dots \ x_i^{(n)}(s))^T$ and $(x_i)_{i \in [1;n]}$ is a base of the vectorial space of solutions. In our example, cos and sin are solutions.

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

As we have $\forall i \ C'_i = A.C_i$ then

$$R' = AR$$

and the general solution will be :

$$X = R.\Omega_0$$

where $\Omega_0 = (u_0 u'_0 \dots u_0^{(n)})^T$.

1.2.3 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 equation.

$$\begin{aligned} X' &= AX + B \\ X &= R.\Omega \end{aligned}$$

Replacing X in the general equation, and remembering that $R' = AR$ we have :

$$R\Omega' = B$$

and 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.

1.2.4 General Solution

We eventually have the general result :

$$X(s) = R(s).(\Omega_0 + \Omega(s))$$

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

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

$$\Omega(s) = \delta \int_{s_0}^s \begin{pmatrix} \frac{S}{\rho_0}(t) \\ \frac{C}{\rho_0}(t) \end{pmatrix} dt$$

And if we define D as :

$$R(s).\Omega(s) = \delta \begin{pmatrix} D(s) \\ D'(s) \end{pmatrix}$$

Then we can add a new initial parameter, $\delta = \frac{\delta p}{p}$ and eventually :

$$\begin{pmatrix} u \\ u' \\ \delta \end{pmatrix} = \begin{pmatrix} C & S & D \\ C' & S' & D' \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} u(s_0) \\ u'(s_0) \\ \delta_0 \end{pmatrix}$$

1.3 Matrix formalism

1.3.1 Illustration of matrix use in optics

According to figure 3, 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 :

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

This is an example which will slightly leads us to the matrix formalism that we will use for the cavity in three dimensions.

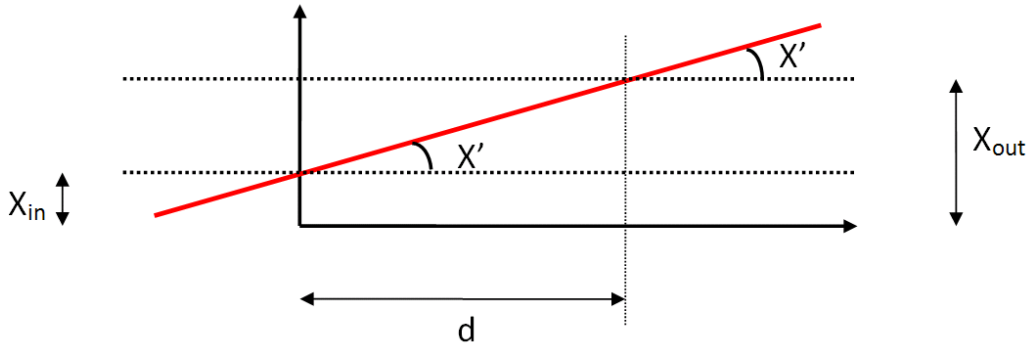


Figure 3: Illustration of the matrix formalism in optics

1.3.2 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 :

$$\begin{aligned} p_x &= \tan(x').p_z \approx x'.p_z \\ p_y &= \tan(y').p_z \approx y'.p_z \\ p_z &= \frac{p_0}{\sqrt{1+\tan^2(x')+\tan^2(y')}} \approx p_0(1 - \frac{x'^2+y'^2}{2}) \approx p_0 \end{aligned} \tag{1}$$

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 curvilign 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 :

$$V = \begin{pmatrix} x & (m) \\ x' & (rad) \\ y & (m) \\ y' & (rad) \\ z & (m) \\ \frac{\delta p}{p} & (\oslash) \end{pmatrix}$$

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 folowing relation for every input vetor V_{in} , which represents a particule a little bit offset relatively to the reference particule, and gives use all the parameters values at the en of the cavity for this same particule, V_{out} :

$$V_{out} = A.V_{in}$$

2 Modelling the cavity with five pillboxes

In this part, we will modelized the cavity by five pillboxes and two drifts pipe on each side of the five cells. 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. In a result of this we will work with a 4D matrix, and we will add the 2 y components at the end.

2.1 Equation of motion in a pillbox

The following calculation were developped in a similar way by Donald A. Edwards [5]. 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 ??, 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 $E' = \frac{\partial E_z}{\partial X}$ assumed to be constant near the axis :

$$E_s(X, t) = E' X \cos(\omega t) \quad (2)$$

$$B_y(X, t) = \frac{E'}{\omega} \cos(\omega t) \quad (3)$$

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

$$\frac{dP_X}{dt} = -eE' \frac{V_s}{\omega} \sin(\omega t) \quad (4)$$

$$\frac{dP_S}{dt} = eE' X \cos(\omega t) + eE' \frac{V_X}{\omega} \sin(\omega t) \quad (5)$$

We will define without justification the transit time factor T :

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

2.2 The One cell pill box model

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

$$E_s = E' X \cos(\omega t - kz) \approx E' X \cos(\omega t) + E' X kz \sin(\omega t) \quad (6)$$

$$B_y = \frac{E'}{\omega} \cos(\omega t - kz) \approx \frac{E'}{\omega} \sin(\omega t) - \frac{E'}{c} z \cos(\omega t) \quad (7)$$

which can be written differently using $s = v_s t \approx ct$ and $\omega = kc$. Let's write equations 4 and 9 for the reference particle, using derivative with respect to s now :

$$\frac{dP_{X,r}}{ds} = \frac{-eE'}{kc} \sin(ks) \quad (8)$$

$$\frac{dP_{S,r}}{ds} = \frac{eE'}{c} X_r \cos(ks) + \frac{eE'}{kc^2} V_{X_r} \sin(ks) \quad (9)$$

$$(10)$$

And for a particle, set at $s_p = s + z$, neglecting second orders terms :

$$\frac{dP_X}{ds} = -\frac{eE'}{kc} \sin(ks) + \frac{eE'}{c} z \cos(ks) \quad (11)$$

$$\frac{dP_S}{ds} = \frac{eE'}{c} X \cos(ks) + \frac{eE'}{kc^2} V_X \sin(ks) \quad (12)$$

Subtracting those two sets of equations gives us

$$\frac{dp_x}{ds} = \frac{eE'}{c} z \cos(ks) \quad (13)$$

$$\frac{dp_s}{ds} = \frac{eE'}{c} X \cos(ks) + \frac{eE'}{kc} \frac{dx}{ds} \sin(ks) \quad (14)$$

Let's integrate 13 from $s = -\lambda/4$ to s , remembering that $k\lambda/4 = \pi/2$:

$$p_x(s) = p_x(-\lambda/4) + \frac{eE'}{ck} z (\sin(ks) + 1) \quad (15)$$

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

$$x'(s) = x'_{in} + \frac{Tp}{2} z \sin(ks) + \frac{Tp}{2} z \quad (16)$$

Which we can integrate again :

$$x(s) = x_{in} + \left(x'_{in} + \frac{Tp}{2} z \right) \left(s + \frac{\lambda}{4} \right) - \frac{Tp}{2k} z \cos(ks) \quad (17)$$

Equations 16 and 17 allows us to integrate equation 9, and eventually :

$$\frac{\delta p}{p} = x_{in} \frac{T}{2} [1 + \sin(ks)] \quad (18)$$

$$+ x'_{in} \frac{T}{2} \left[\left(s + \frac{\lambda}{4} \right) \sin(ks) \right] \quad (19)$$

$$+ z \frac{T^2}{4} \left[\left(s + \frac{\lambda}{4} \right) \sin(ks) - \frac{1}{2k} \sin(2ks) \right] \quad (20)$$

2.3 The One cell pill box 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.

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

Then the matrix of the 5 cells is :

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

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 :

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

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

$$\begin{aligned} d &= 0.05385 \, m \\ \lambda &= 0.07692 \, m \\ T &= -0.6031 \end{aligned}$$

3 Using ASTRA to determine the cavity matrix

3.1 A short description of Astra

Astra is a free particle tracking software which can take in account several particles inside of an electromagnetic fields. All the programs and a precise documentation can be found at <http://www.desy.de/~mpyflo/>. Here we will use the functionality which allows us to directly enter the E/B field map as a data of the program. Astra always uses at least two files. The first one is an .INI file and contains the initial particle distribution in terms of position, momentum, and time. We will name it *input_particles.ini*. The second one is a .IN file which contains all the necessary parameters for astra running. We will name it *run.in*. Examples of these files can be found in annex.

3.2 Description of the method

The main idea is that when you multiply a matrix by the i th canonical base vector you obtain exactly the i th column of the matrix.

$$\begin{pmatrix} A_{11} & \dots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{i1} & \dots & A_{in} \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ A_{n1} & \dots & A_{nn} \end{pmatrix} \cdot \begin{pmatrix} 0 \\ \vdots \\ \lambda \\ \vdots \\ 0 \end{pmatrix} = \lambda \begin{pmatrix} A_{1i} \\ \vdots \\ A_{ii} \\ \vdots \\ A_{ni} \end{pmatrix} \quad (21)$$

Then we just have to input into the astra program the six canonical vectors, collect the output informations given by astra, and then this will give us the numerical cavity matrix.

3.3 The physical limit of this method

As we are treating data with physical units, we need to be careful of using canonical vectors that make sense. As a result of this, we will use the following vectors, and keep this notation along this document.

$$V_0 = \vec{0} \quad V_1 = \begin{pmatrix} 0.001 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad V_2 = \begin{pmatrix} 0 \\ 0.01 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad V_3 = \begin{pmatrix} 0 \\ 0 \\ 0.001 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad V_4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0.01 \\ 0 \\ 0 \end{pmatrix} \quad V_5 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.001 \\ 0 \end{pmatrix} \quad V_6 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.1 \end{pmatrix} \quad (22)$$

For the position offset we have chosen 1mm, for the deflection 10 mrad and for the absolute variation of p we chose a tenth offset. Those values are reasonable given the size of the cavity.

3.4 Configuring Astra

3.4.1 The fields format

The fields were generated with the program HFSS. Astra needs six files to represent the E and B fields. Those might have the same name, contain the "3D" characters at the beginning or at the fifth position of the file name, and end with the extensions .ex, .ey, .ez, .bx, .by, .bz. We used *3D.ex*, but *5cel3D_field.ex* should also work. Then the files have to be in a specific format, with three lines of header, and then the data developed along x in columns and successively along y and z in rows. If $F(i, j, k)$ is the field value at the point $x(i)$, $y(j)$, $z(k)$, N_x the number of x values and respectively for y and z , then the file should have this shape :

$$\begin{bmatrix} Nx & x(1) & x(2) & \dots & x(Nx) \\ Ny & y(1) & y(2) & \dots & y(Ny) \\ Nz & z(1) & z(2) & \dots & z(Nz) \end{bmatrix} \begin{bmatrix} F(1,1,1) & F(2,1,1) & \dots & \dots & F(Nx,1,1) \\ \dots & \dots & \dots & \dots & \dots \\ F(1,Ny,1) & F(2,Ny,1) & \dots & \dots & F(Nx,Ny,1) \\ \dots & \dots & \dots & \dots & \dots \\ F(1,1,Nz) & F(2,1,Nz) & \dots & \dots & F(Nx,1,Nz) \\ \dots & \dots & \dots & \dots & \dots \\ F(1,Ny,Nz) & F(2,Ny,Nz) & \dots & \dots & F(Nx,Ny,Nz) \end{bmatrix}$$

3.4.2 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. Let's call $E_{z0}(z)$ the field map we have, $E_z(z, t)$ the real field and $E_z^p(t)$ the field seen by the particle at t, $E_z^p(z)$ the field seen by the particle at z.

$$E_z(z, t) = E_{z0}(z).sin(\omega t + \phi) \quad (23)$$

Then, as the particle flights at the speed of light velocity, we have $z(t) = ct$ and thus :

$$E_z^p(z) = E_z(z, \frac{z}{c}) = E_{z0}(z).sin(2\pi\frac{z}{\lambda} + \phi) \quad (24)$$

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 ϕ such that the sine is equal to minus one at $L_c/2$, ie :

$$2\pi\frac{L_c}{2\lambda} + \phi = \frac{-\pi}{2} \quad (25)$$

We find $\phi = 287.58$ degrees. Figure 4 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.

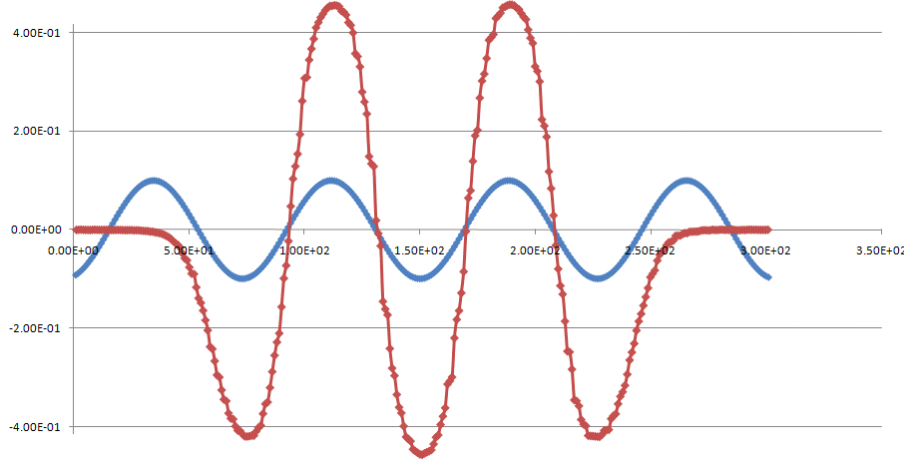


Figure 4: Ezo (in red) in MV/m and the sine function adjusted in phase without scale

3.4.3 The fields scaling

As the fields were generated by , arbitrary values of the power given to the cavity were inputted. 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 his 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 unefficient. We used a method consisting in integrating E_z^p along the axis $x = 1mm$, $y = 0$, and compare the voltage value with the real one : 45kV.

$$V = \int_0^{L_c} E_z^p . dz \quad (26)$$

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

$$V = \sum_i \int_{z_i}^{z_{i+1}} E_z^p(z) . dz = \sum_i E_z^p(z_i) \int_{z_i}^{z_{i+1}} dz = \sum_i E_z^p(z_i) . z_{step} \quad (27)$$

We find $V = 4.8270.10^{-3}V/m$, and a scale of 9.3227e+6 to have the good field amplitude in V/m, which means a scale of 9.327 to have a field in 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 hfss field map represent H, we will also multiply this by μ_0 .

3.4.4 The input file format

Column	1	2	3	4	5	6	7	8	9	10
Parameter	x	y	z	px	py	pz	clock	macro charge	particle index	status flag
Unit	m	m	m	eV/c	eV/c	eV/c	ns	nC		
C Format	6.4e	6.4e	6.4e	6.4e	6.4e	6.4e	6.4e	6.4e	int	int

The particle index could be 1,2,3 or 4 respectively corresponding to electrons, positrons, protons, and hydrogen ions. The status flag we will use will be 3 for particles that we want to track in order to have their trajectory, or 5 if we don't want their trajectory inside the cavity. If a particle is lost during the simulation, a negative status flag will be attributed.

3.4.5 The NEWRUN section of the run file

In this section we configure the way astra will run and track the particles. By setting the parameter *Distribution* = 'input_particle.ini' we define the relative location of the input distribution file relative to the directory where we are running astra. Example : If the astra.exe program is in the root /, the run input file located at /run/astra_run.in, and the distribution file at /distributions/input_particles.ini. Then the parameter in the astra_run.in file has to be *Distribution* = 'distributions/input_particles.ini'. And we run astra by typing, in the root directory : »astra run/astra_run.in.

It is important to set Z_START and Z_STOP, because it defines the range where the particles will be tracked. No particle will be tracked after Z_STOP, and all unlost particles will have a final z around Z_STOP. To generate the output file we need, in this example the file run.0030.001, we have to set *PhaseS* as true in this section. If we also want to track the particle along his trajectory in order to plot it, we have to set *TrackS* to True and set the status flag of the particles we want to track at the value 3.

3.4.6 The CAVITY section of the run file

In this section, we configure the cavity. Here is what it could look like :

```
&CAVITY
LEfield=.T
File_Efield(1) = 'fields/3D' ,
Nue(1)=3.9, Phi(1)=107.11 , C_pos(1)=0.15,
C_NumB(1)=1, C_NoScale(1)=T
/
```

- LEfield is a switch, when set to true the cavity is present, otherwise, astra run as if no cavity fields were present.
- File_Efield defines the file name used for the fields. It can be a relative or absolute path. Here, it means that the fields files are named 3D.ex 3D.ey etc... and are situated in the directory fields in reference to the directory where you are running astra (confer previous section for the distribution file).
- Nue is the RF frequency of the cavity field in GHz
- Phi is the phase in degree added to the RF sine function, which generates the time oscillation inside of the cavity.
- C_pos defines the position of the center of the cavity. Indeed our fields have a z index which goes from -150 mm to 150 mm. Astra will add a 150 z offset.
- C_NumB defines the number of cell/period if we want to repeat several time the field in space.
- C_NoScale when set as true avoid the scaling method used by astra. Indeed we already did a previous scaling of the fields.

3.5 Important errors to avoid

In order to make this work beneficial to further similar studies, we will here describe the different problems we encountered and solutions found.

- **The scale** : The fields given by HFSS have to be scaled as describes before. It seems to be the same scale for E and H. Then H must be multiplied by μ_0 to obtain B.

- **The units** : Astra needs fields files in MV/m and MT scale.
- **The input distribution file** : the first line of this file contains the information of the reference particle. The following lines concerns the other particules and they are all given relatively to the reference particule.
- **The output distribution file run.0030.001** : It has the same structure as the input distribution file. Nevertheless, x,y,px,py are absolute values, and not values with respect to the reference particle. However pz, and z are relative to the reference particle.
- **Using astra programs** : Astra comes with three other programs called *lineplot.exe*, *fieldplot.exe* and *postpro.exe*. To use them you must have in the same directory the file *grfont.dat* that can be found at <http://www.desy.de/~mpyflo/>. Moreover, you have to set your color settings at 16 Bits rather than 32, other wise it won't work.

4 Programs developped and resulting matrix

4.1 Description of the different programs

Several C programs have been written during this study, to save time doing calculation, formatting files, and converting variables. Here is a schematic describing the main relationship between those programs and external files generated or needed for the programs to run.

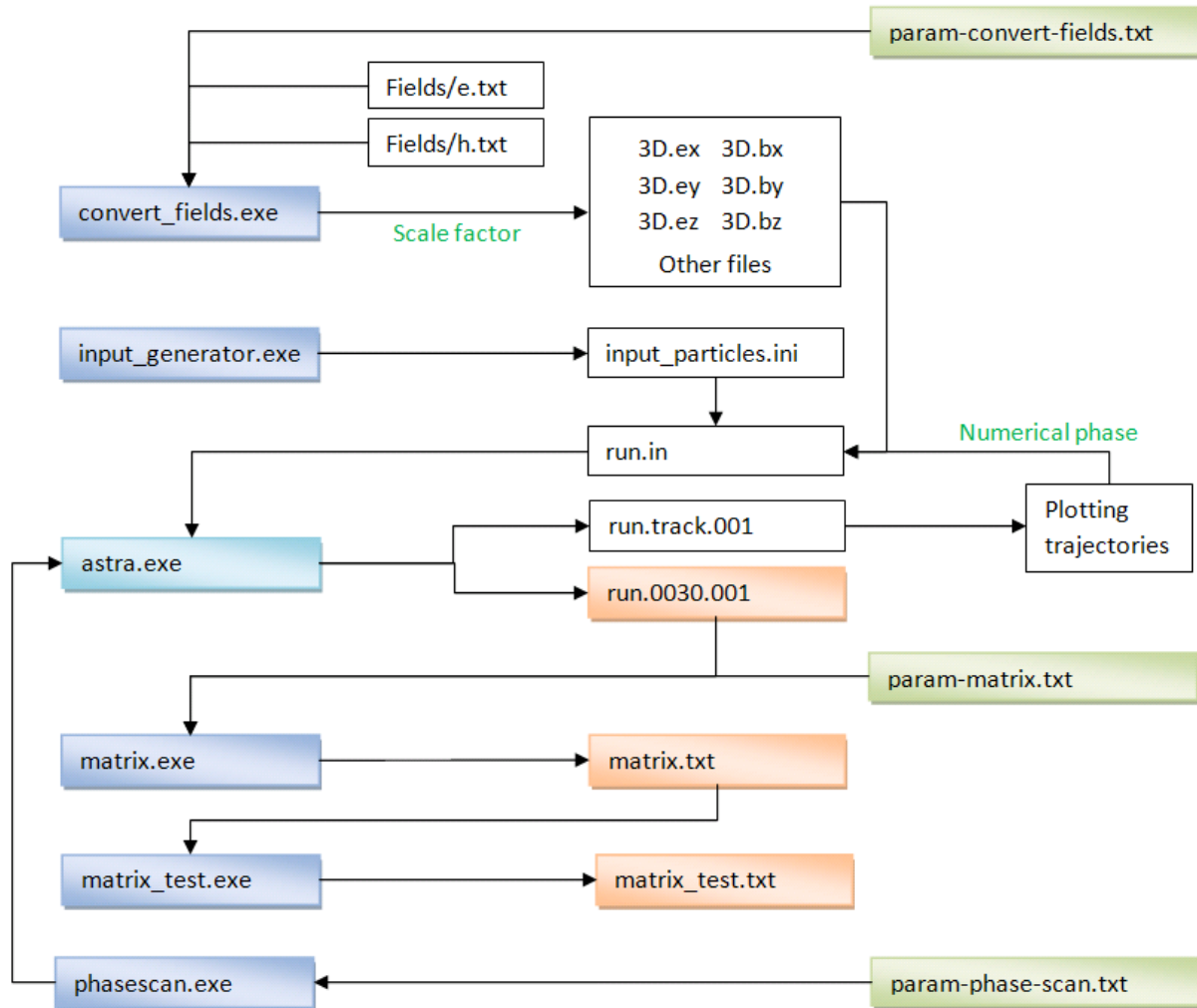


Figure 5: Organigramm of the created programs

Convert fields As described in figure 5, the `Convert_fields.exe` program open the file `param_convert_fields.txt` which shoul have this format :

```

E file location
H file location
xmin xmax xstep
ymin ymax ystep
zmin zmaz zstep
Z offset/Cavity Center position
RF Lambda
Integrate Voltage along the x=1mm axis

```

Numerical values are given in float format, in meters and Volt. The two first lines concern the fields files generated by HFSS. If z doesn't start at 0 but at -15mm then the z offset should be set to 0.150 m. This value is the same than the C_pos of the Cavity section in the astra run file.

The output of the program is the six fields files well formatted and scaled. The electric field is in MV/m and the magnetic field in MT. Moreover, to obtain the B field, the H field has been multiplied by $\mu_0 = 2\pi \cdot 10^{-7}$.

The program also generates the file `axis.ez`, which is the E_z field at $x = 1\text{mm}$. The four columns of this file correspond to x, y, z, E_z , in millimeters and V/m.

The files `by_x_z.dat`, `bz_x_z.dat`, `ez_x_z.dat`, represents respectively B_y, B_z and E_z at $y=0$ in a grid $Z \times X$, which means that the lines correspond to the different values of z , and the columns to the different values of x .

How to run it ? Just type `convert_fields`. Then a prompt will ask you whether you want to give your own scale factor, or whether you want the program to calculate it. It will use the method of integration along the $x = 1\text{mm}$ axis previously described.

Input generator The program is simply called by typing `input_generator` and generates the file `input_particles.ini`. The first line of this file correspond to the reference particule, and the others are given relatively to the first one.

Matrix The matrix program will read the astra output file, convert it to good variables using formula 1. After transposition and normalisation, it will output a matrix, in the file `matrix.txt`. It will ask for one argument which is the location of the file generated by astra of the form : `astra_run_file_name.zzzz.rrr`. Where `zzzz` correspond to the length in cm of the tracking, and `rrr` the run number. In our example it will simply be `run.0030.001`. In our example, we call this program by typing : `>matrix run.0030.001`. Before running the program you must be sure that the file `param_matrix.txt` is present in the same folder than the program and that it is well configured. Here is what this file should look like :

```
X scale factor
X' scale factor
Y scale factor
Y' scale factor
Z scale factor
Delta p over p scale factor
P0
```

$P0$ is the momentum of the beam in eV. The scale factors corresponds to the constants that will be used to get the matrix in good units and scale. In deed, for instance we choosed an input vector of 1mm x offset. If you then refer to the part 3.2 Description of the method, you will see that in this example $\lambda = 0.001$. To have the real matrix column we have to multiply our result by $\frac{1}{\lambda}$ which is here 1000. So what we called X scale factor will be 1000. For x' , the input vector is 0.01radian offset. This leads us to $\lambda = 0.01$, and the x' scale factor will be 100, etc...

Here is the file we use :

```
1000
100
1000
100
1000
10
```

Phase scan The `phase_scan.exe` program should be in the same folder than the `astra.exe` program, the distribution file, the six fields files, and the `param_phase_scan.txt` file. His role is

to run astra for different phase and produce the different tracking and output files in different folders. The program will read the parameter file, which have to contain exactly the following information :

```
Newrun distribution_filename.ini
Newrun Z_min
Newrun Z_max
Cavity field_file_name without any extension
Cavity frequency
Cavity center position
```

Distances are float in meters, and frequency in Ghz. To understand the meaning of each terms, please read the astra instructions about the Newrun and Cavity section (paragraph 3.4.5 and 3.4.6).

Here is an example of this file :

```
input_particles.ini
0
0.30
3D
3.9
0.15
```

While running, the program will first ask for the minimal and maximal phase and then the phase step. After that, it will generates the requiered files in different folders with explicit name, and run astra. Caution : you will have to press a key between all the astra simulations. If you planned to see 100 different phases run, then it will take around 6 minutes and you will have to press a key every 4 seconds...

Dispatch

Matrix test

4.2 The final cavity matrix

$$M_{cav} = \begin{pmatrix} 1.003 & 0.300 & 0.002 & 0.001 & -0.448 & 0.001 \\ 0.020 & 1.004 & -0.002 & 0.002 & -2.991 & -0.000 \\ 0.000 & 0.000 & 1.002 & 0.300 & -0.000 & 0.000 \\ 0.002 & -0.000 & 0.012 & 1.002 & -0.000 & 0.000 \\ -0.001 & -0.001 & 0.000 & -0.001 & 0.998 & 0.000 \\ -3.001 & -0.450 & -0.001 & -0.001 & 0.275 & 0.999 \end{pmatrix}$$

Acknowledgements

I would like to personally thank : Helen Edwards, Don Edwards, Ray Fliller, Timothy Koeth, James Santucci, Philippe Piot, and all the Fermilab employees that helped me during this internship.

Annexes

A more precise numerical calculation of V

Previously in this report, we described a integration method for obtaining V with considered that E_{z0} and the sine function were constant between two numerical step. Here, let's do it more precisely, integrating the sine. We will start from this general expression for E_z

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

Remembering that, the particle sees the following field when it travels at c :

$$E_z^p(z) = E_z(z, \frac{z}{c})$$

Then :

$$V = \int_0^{l_c} E_z^p \cdot dz = \sum_i \int_{z_i}^{z_{i+1}} E_z^p(z) \cdot dz = \sum_i E_{z0}(z_i) \int_{z_i}^{z_{i+1}} \sin(kz + \phi) dz$$

And finally :

$$V = - \sum_i \frac{E_{z0}(z_i)}{k} [\cos(kz_{i+1} + \phi) - \cos(kz_i + \phi)] dz \approx 4.8216 \cdot 10^{-3}$$

This gives us a scale factor of 9.3330, which should be more correct than the other numerical result of 9.3227.

Analytical expression of V

From an analytical modelization of the field map, let's find an analytical expression which gives us the V and the scale factor. Here, we won't take into account the two surrounding pipes. Our z axis starts at the enter of the first cell. According to the field map, we can modelise E_{z0} , at $x = 1\text{m}$, like this :

$$E_{z0}(z) = -E_0 \sin(kz)$$

Numerically, we have, in the field map given by HFSS : $E_0 = 5 \cdot 10^{-2} \text{V/m}$. Then, we multiply it by a negative sine to get the real time dependent field :

$$E_z(z, t) = -E_{z0}(z) \sin(\omega t)$$

Once again, let's remember that, the particle sees the following field when it travels at c :

$$E_z^p(z) = E_z(z, \frac{z}{c})$$

We can try to evaluate V :

$$V = \int_0^{5\lambda/2} E_z^p \cdot dz = \int_0^{5\lambda/2} E_0 \sin^2(kz) dz$$

An integration by part, gives us :

$$V = E_0 l_c - V$$

And finally :

$$V = \frac{E_0 l_c}{2} \approx 4.8075 \cdot 10^{-3}$$

This gives us a scale factor of 9.3607, which is close to the numerical one 9.3227.

References

- [1] C. Bovet, R. Gouiran, and K. H. Reich I. Gumowski. *A selection of formulae and data useful for the design of A.G. Synchrotrons*. European Organization for Nuclear research, 1970.
- [2] Jean-Paul Carneiro. Etude experimentale du photo-injecteur de fermilab, 2001.
- [3] M. Cornacchia and P. Emma. Transverse to longitudinal emittance exchange. *the American Physical Society*, (084001), 2002.
- [4] Astra distribution website. <http://www.desy.de/mpyflo/>.
- [5] Donald A. Edwards. *Notes on transit in deflecting mode pillbox cavity*. 2007.
- [6] Ray P. Fliller et al. Transverse to longitudinal emittance exchange beamline at the a0 photoinjector. *PAC07 Albuquerque*, 2007.
- [7] John David Jackson. *Classical Electrodynamics*. Manhattan Music, 1975.
- [8] K.-J. Kim. Transverse-transverse and transverse-longitudinal phase space converters for enhanced beam applications. *PAC07 Albuquerque*, 2007.
- [9] Timothy W. Koeth. A copper 3.9 ghz tm110 cavity for emittance exchange. *PAC07 Albuquerque*, 2007.
- [10] M. McAshan and R. Wanzenberg. Rf design of a transverse mode cavity for kaon separation. *Fermilab*, (TM-2144), 2001.
- [11] Robert B. Muchmore. *Essentials of Microwaves*. John Wiley and Sons, 1953.
- [12] Hasan Padamsee, Jens Knobloch, and Tom Hays. *RF Superconductivity for accelerators*. Wiley-Interscience, 1998.
- [13] P.Emma, Z.Huang, K.-J. Kim, and P. Piot. Transverse to longitudinal emittance exchange to improve performance of high gain free electron lasers. *The American Physical Society*, (100702), 2006.
- [14] Y.-E Sun. Longitudinal and transverse emittance exchange : transfer matrix and phase space. (EX-002), 2007.
- [15] Helmut Wiedemann. *Particle Accelerator Physics I*. Springer, 1999.