Hands-on Astra Tutorial

Zachary J. Liptak
With thanks to Abhay Deshpande for material

August 30, 2025

About ASTRA

What is ASTRA?

- **A S**pace Charge **TR**acking **A**lgorithm
- Developed by Prof. Klaus Flöttman at DESY for beam tracking simulations
- Freely available for download and use, with pre-compiled binaries
- Written in FORTRAN, but you don't need to actually know FORTRAN to use it!

Programs

We'll actually be using several different programs:

- generator: Generates particle distribution
  (You can also provide a distribution manually)
- astra: Runs the tracking algorithm

There are also several different graphics programs used for representing beam and field distributions:

- fieldplot: Plot EM fields
- lineplot: Plot beam parameters (size, emittance, &c.)
- postpro†: Plot beam profiles

The graphics programs are based off of PGPLOT packages.
†: This seems to have some issues on recent Mac builds due to some XQuartz compatibility problem

Getting ASTRA

ASTRA is free and readily available for download and use on Windows, Linux and Mac. Windows and Linux (64-bit) are the recommended platforms, as those are the ones most often supported/updated.
It's availble at `https://www.desy.de/~mpyflo/`.
Note that there is also a very thorough manual at the "Astra documentation" section, along with some examples.
The "Utility programs" link contains some scripts to convert between ASTRA and SDDS (e.g., for ELEGANT) and HFSS for those who want to use it with other tracking codes.

Installing ASTRA

Windows (Assume we're using C:\\as the default)
Create a folder in C:\\called ASTRA and download all the .exe files into it. You can (and should) also download the manual for reference.
To make running these programs easier in the future, you can add them to your path by searching for "Environment Variables" in your system settings and adding C:\\ASTRA to the "Path" option.
Alternately, you can use the command line to do the same:
setx path "%PATH%;C:\\ASTRA"

Installing ASTRA on Windows

Now you should be able to run the .exe programs you downloaded.
In the command prompt, type

```
generator name-of-file.in
astra name-of-file.in
```

(For now, try without the file names to see if they work, we'll make inputs in just a bit)
You can see results in a similar way after running the above programs:

```
lineplot filename.run-number
fieldplot filename.run-number
postpro filename.run-number
```

Installing ASTRA

Linux/Mac

Similar to Windows, you can download the pre-compiled executables.

Create a folder such as ∼/ASTRA and download the executables to it.

(∼ means your home directory, e.g., /usr/yourname/home)

you may need to do chmod 755 astra (and generator, and...) to be able to run them.

Add this folder to your PATH by adding

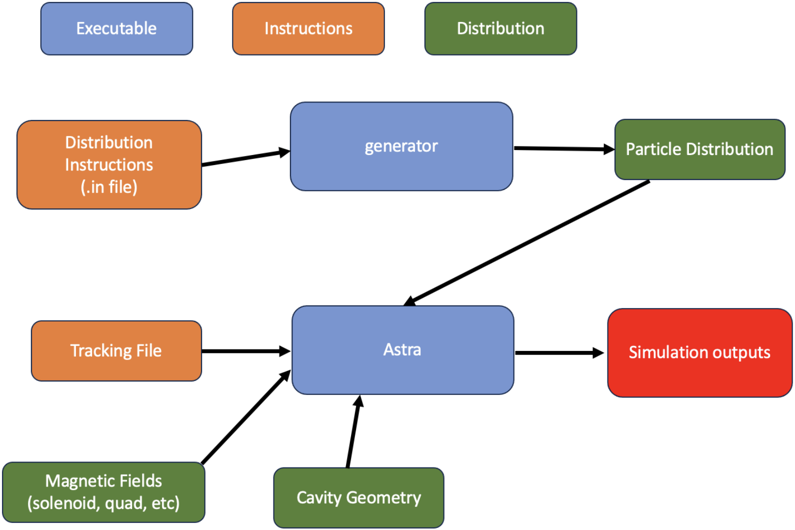
export PATH=∼/Astra:$PATH

to the file .bash_profile, either with your text editor of choice.

Installing on Linux/Mac

In the directory you made, you can do ./astra (or ./generator, &c.) to run the programs; once the folder has been added to your PATH, you will be able to run anywhere by simply typing **astra** (**generator**, **fieldplot**...).
(Mac users may also need to right click on the icon in the Finder window and choose 'open with' in order to force MacOS to let you open it.)

ASTRA Program Flow



ASTRA Inputs

On previous slides we mentioned input files, e.g.:
generator **input-file.in**
So what do those contain?

Generating a Bunch

We can make a particle bunch using the generator program. As an example, let's make one by providing laser parameters that will generate the beam in our cathode:

Transverse

- $\sigma_x = \sigma_y = 0.25\,\text{mm}$ (Radial Profile)
- Radius is $2\sigma_x = 0.5\,\text{mm}$
- FWHM $= \sqrt{3} \times r = .866$ mm

Longitudinal

- $\sigma_z = 5.5\,\text{ps}$ (Gaussian Profile)
- Charge $= 0.5$ nC/bunch

Defining our Parameters

Parameters in ASTRA are given in *Namelists*. There are a number of defined Namelists for inputs such as beam generation, magnet fields, &c. that we can use for defining our beam.
Namelists have the form:

&NAME
arguments
.
.
/ (← this ending / is necessary!)

The &INPUT namespace

Let's define the laser beam we wrote up above, using the &INPUT namelist:

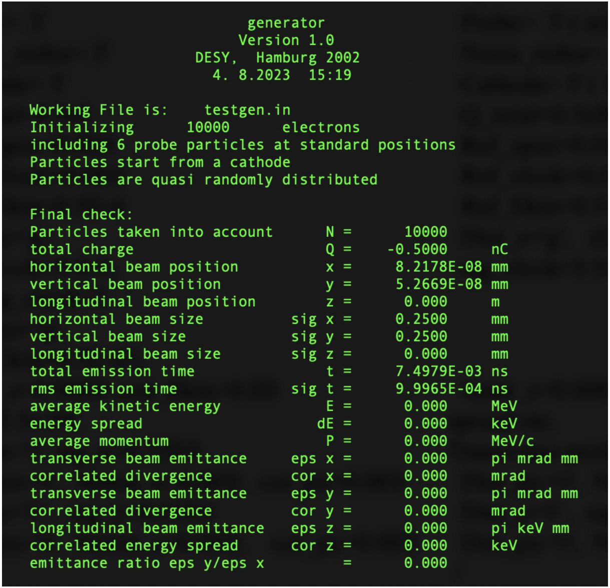
```
&INPUT
FNAME=rfg_10000_1nC.ini ! defines the name of the distribution
IPart=10000 ! sets no. of particles
Species='electrons' ! define particle species
Q_total=0.5e0 ! Set total charge
Dist_x='r', sig_x=0.25 ! Set $\sigma_x$ and choose radial dist.
Dist_px='r', Nemit_x=0.0, cor_px=0.0
Dist_y='r', sig_y=0.25 ! Set $\sigma_y$, radial dist.
Dist_py='r', Nemit_y=0.0, cor_py=0.0
sig_clock=5.0E-3 ! Z distribution, defined as a laser pulse time
Dist_z='g' ! Gaussian
Dist_pz='g'
/
```

Using Generator to make your bunch

Save your generator file as a .in file: source_1nC.in
Run Generator with this file as input: (./generator source_1nC.in on linux)
The output will be a distribution file: rfg_10000_1nC.ini (or whatever you specified in your file)
Now you can view this with postpro:
postpro rfg_10000_1nC.ini

The Particle Distribution

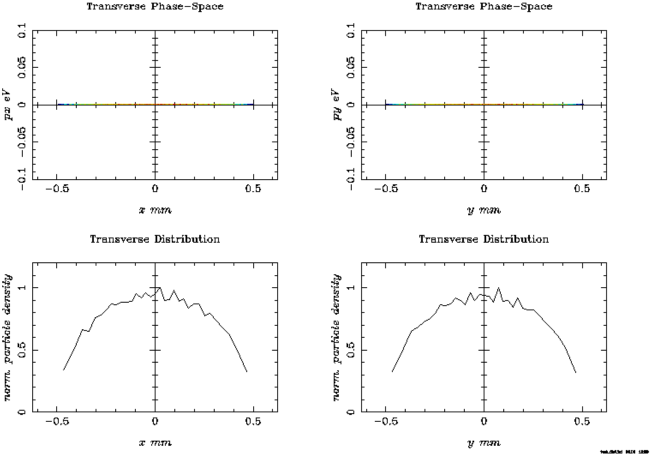
Upon entering the above, we get the following output:



```
                          generator
                         Version 1.0
                     DESY,  Hamburg 2002
                       4. 8.2023   15:19

 Working File is:    testgen.in
 Initializing      10000       electrons
 including 6 probe particles at standard positions
 Particles start from a cathode
 Particles are quasi randomly distributed

 Final check:
 Particles taken into account      N =      10000
 total charge                      Q =    -0.5000     nC
 horizontal beam position          x =     8.2178E-08 mm
 vertical beam position            y =     5.2669E-08 mm
 longitudinal beam position        z =      0.000     m
 horizontal beam size          sig x =     0.2500     mm
 vertical beam size            sig y =     0.2500     mm
 longitudinal beam size        sig z =      0.000     mm
 total emission time               t =     7.4979E-03 ns
 rms emission time             sig t =     9.9965E-04 ns
 average kinetic energy            E =      0.000     MeV
 energy spread                    dE =      0.000     keV
 average momentum                  P =      0.000     MeV/c
 transverse beam emittance     eps x =      0.000     pi mrad mm
 correlated divergence         cor x =      0.000     mrad
 transverse beam emittance     eps y =      0.000     pi mrad mm
 correlated divergence         cor y =      0.000     mrad
 longitudinal beam emittance   eps z =      0.000     pi keV mm
 correlated energy spread      cor z =      0.000     keV
 emittance ratio eps y/eps x         =      0.000
```

The Particle Distribution

Outputs from postpro:



Transverse Phase–Space

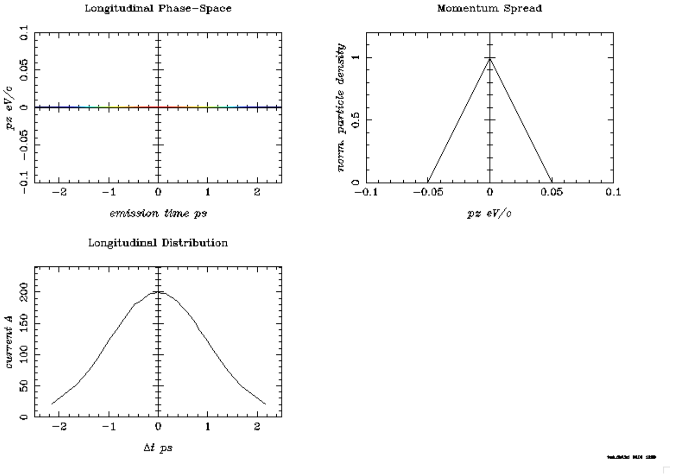
Transverse Phase–Space

Transverse Distribution

Transverse Distribution

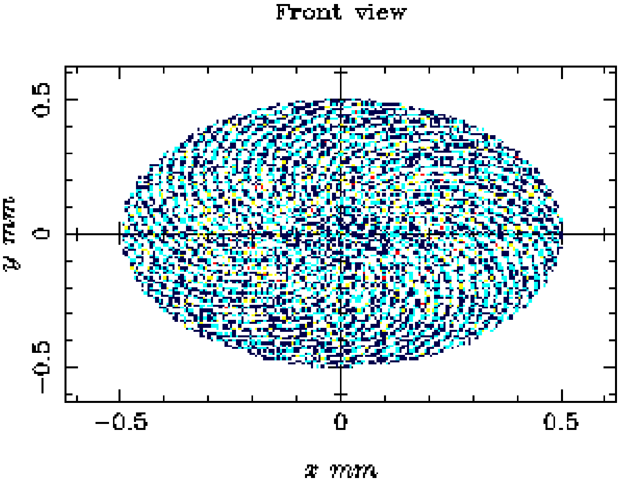
The Particle Distribution

Outputs from postpro:



The Particle Distribution

Outputs from postpro:

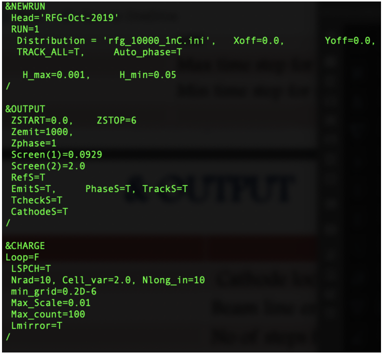


Tracking

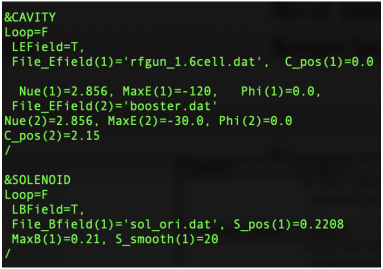
We can build our tracking simulation now using a set of relevant namelists:

- &NEWRUN:
  Defines run, specifies what parameters to evaluate, &c.
- &SCAN:
  Defines scans over some parameter (e.g., phase)
- &CHARGE:
  Specifies parameters for space charge calculation
- &CAVITY:
  Define accelerating fields, RF Gun, &c.
- &SOLENOID:
  Define solenoid field
- &QUADRUPOLE:
  Define Quadrupole fields

Example File



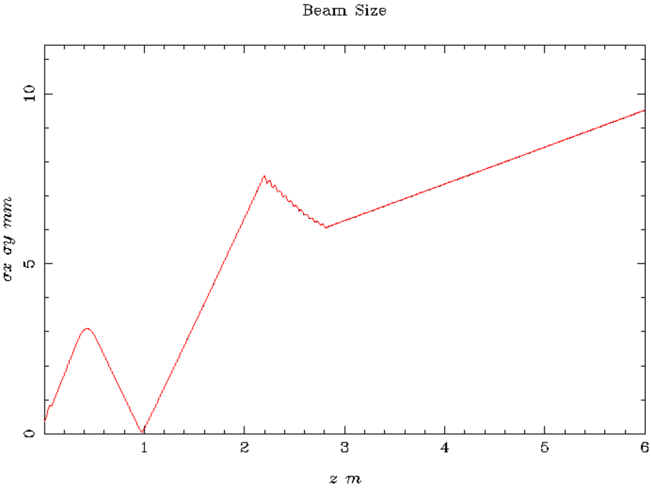
```
&NEWRUN
 Head='RFG-Oct-2019'
 RUN=1
  Distribution = 'rfg_10000_1nC.ini',   Xoff=0.0,       Yoff=0.0,
  TRACK_ALL=T,      Auto_phase=T

   H_max=0.001,     H_min=0.05
/

&OUTPUT
 ZSTART=0.0,    ZSTOP=6
 Zemit=1000,
 Zphase=1
 Screen(1)=0.0929
 Screen(2)=2.0
 RefS=T
 EmitS=T,      PhaseS=T, TrackS=T
 TcheckS=T
 CathodeS=T
/

&CHARGE
Loop=F
 LSPCH=T
 Nrad=10, Cell_var=2.0, Nlong_in=10
 min_grid=0.2D-6
 Max_Scale=0.01
 Max_count=100
 Lmirror=T
/
```



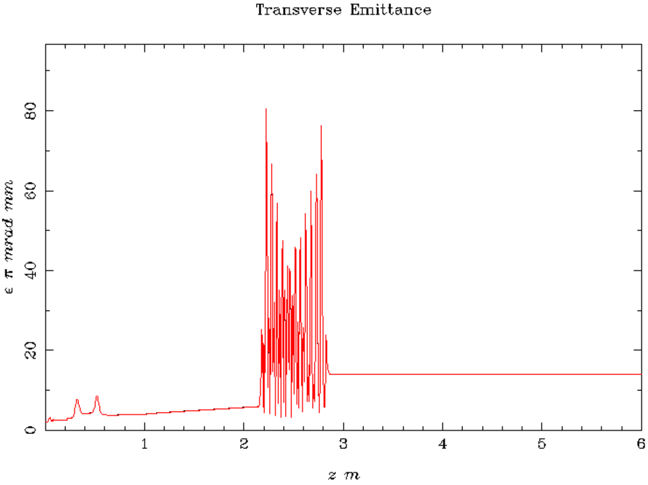
```
&CAVITY
Loop=F
 LEField=T,
 File_Efield(1)='rfgun_1.6cell.dat',  C_pos(1)=0.0

  Nue(1)=2.856, MaxE(1)=-120,    Phi(1)=0.0,
 File_EField(2)='booster.dat'
Nue(2)=2.856, MaxE(2)=-30.0, Phi(2)=0.0
C_pos(2)=2.15
/

&SOLENOID
Loop=F
 LBField=T,
 File_Bfield(1)='sol_ori.dat', S_pos(1)=0.2208
 MaxB(1)=0.21, S_smooth(1)=20
/
```

And running this with **astra input_file.in** we get:

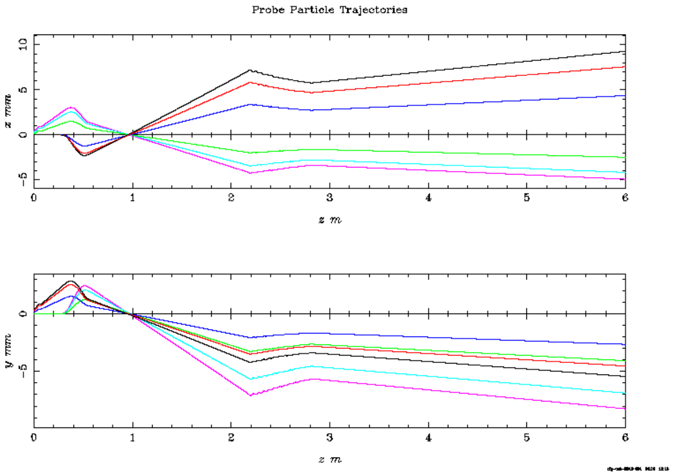
Tracking Outputs



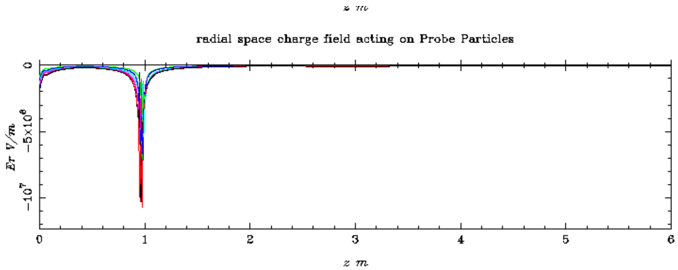
Tracking Outputs



Tracking Outputs

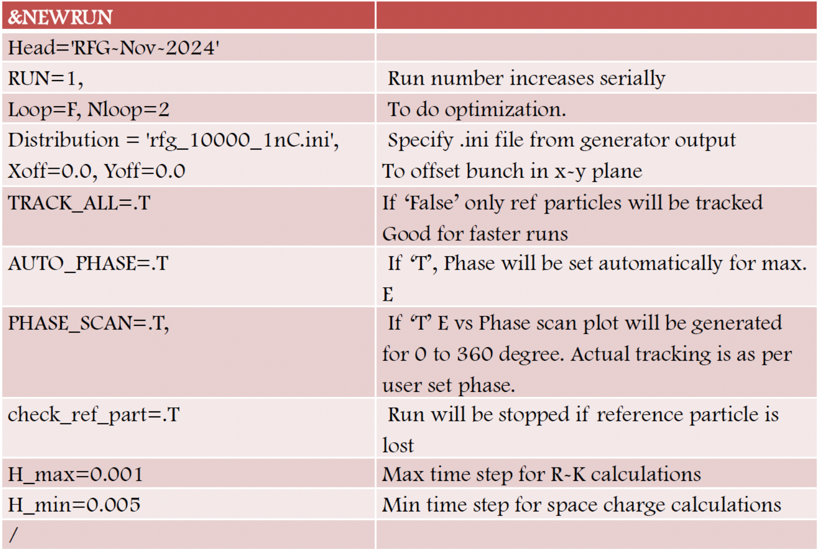


Tracking Outputs



&NEWRUN

From page 54 of the manual:

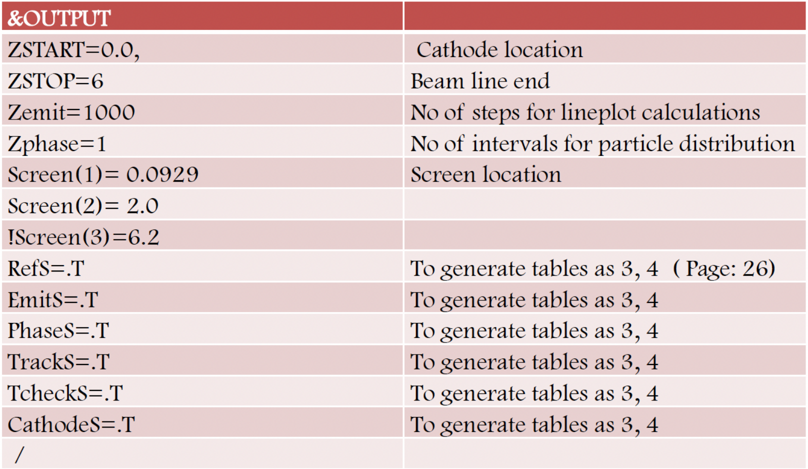


&NEWRUN	
Head='RFG-Nov-2024'	
RUN=1,	Run number increases serially
Loop=F, Nloop=2	To do optimization.

Distribution = 'rfg_10000_1nC.ini',
Xoff=0.0, Yoff=0.0
	Specify .ini file from generator output
	To offset bunch in x-y plane

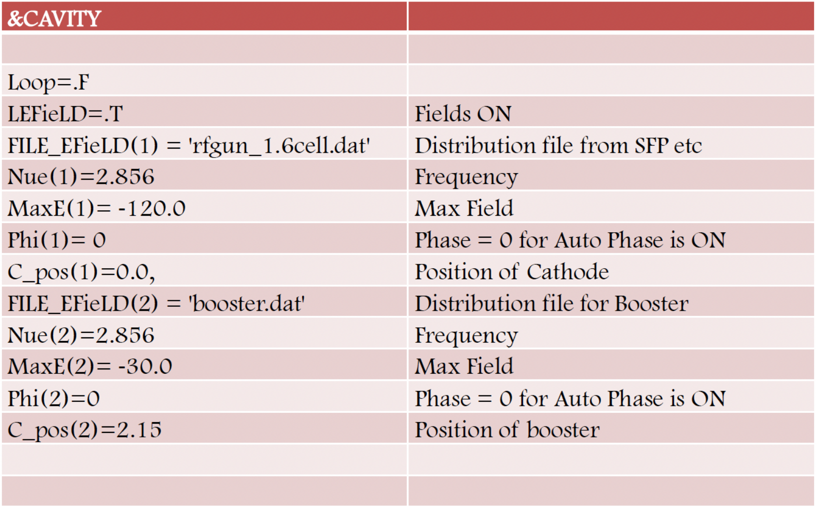
TRACK_ALL=.T	If 'False' only ref particles will be tracked Good for faster runs
AUTO_PHASE=.T	If 'T', Phase will be set automatically for max. E
PHASE_SCAN=.T,	If 'T' E vs Phase scan plot will be generated for 0 to 360 degree. Actual tracking is as per user set phase.
check_ref_part=.T	Run will be stopped if reference particle is lost
H_max=0.001	Max time step for R-K calculations
H_min=0.005	Min time step for space charge calculations
/	

&OUTPUT



&OUTPUT	
ZSTART=0.0,	Cathode location
ZSTOP=6	Beam line end
Zemit=1000	No of steps for lineplot calculations
Zphase=1	No of intervals for particle distribution
Screen(1)= 0.0929	Screen location
Screen(2)= 2.0	
!Screen(3)=6.2	
RefS=.T	To generate tables as 3, 4 (Page: 26)
EmitS=.T	To generate tables as 3, 4
PhaseS=.T	To generate tables as 3, 4
TrackS=.T	To generate tables as 3, 4
TcheckS=.T	To generate tables as 3, 4
CathodeS=.T	To generate tables as 3, 4
/	

&Cavity



&CAVITY	
Loop=.F	
LEFieLD=.T	Fields ON
FILE_EFieLD(1) = 'rfgun_1.6cell.dat'	Distribution file from SFP etc
Nue(1)=2.856	Frequency
MaxE(1)= -120.0	Max Field
Phi(1)= 0	Phase = 0 for Auto Phase is ON
C_pos(1)=0.0,	Position of Cathode
FILE_EFieLD(2) = 'booster.dat'	Distribution file for Booster
Nue(2)=2.856	Frequency
MaxE(2)= -30.0	Max Field
Phi(2)=0	Phase = 0 for Auto Phase is ON
C_pos(2)=2.15	Position of booster

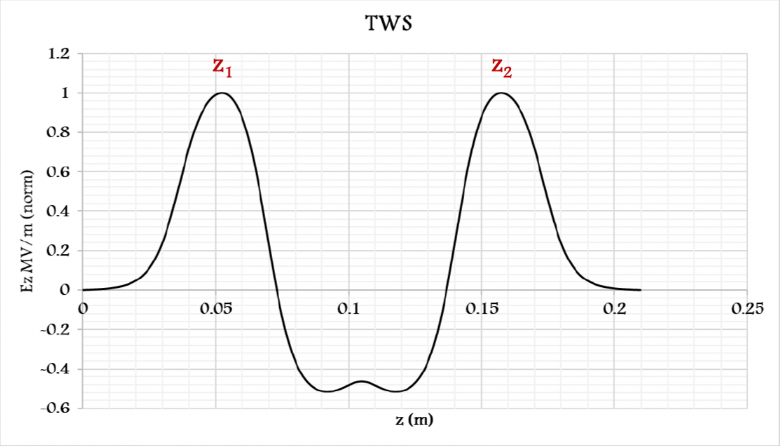
TW Linac

You can define a Traveling Wave field by providing the on-axis longitudinal field for at least one period.
We can do this with the input file: TWS_sband.dat
We add a first line of the form:
z_1 z_2 n m
where z_1 and z_2 are the start and end points of the cells, and n and m are two integers characterizing the phase advance per cell:

$$\phi = \frac{2\pi n}{m} \tag{1}$$

TW Linac

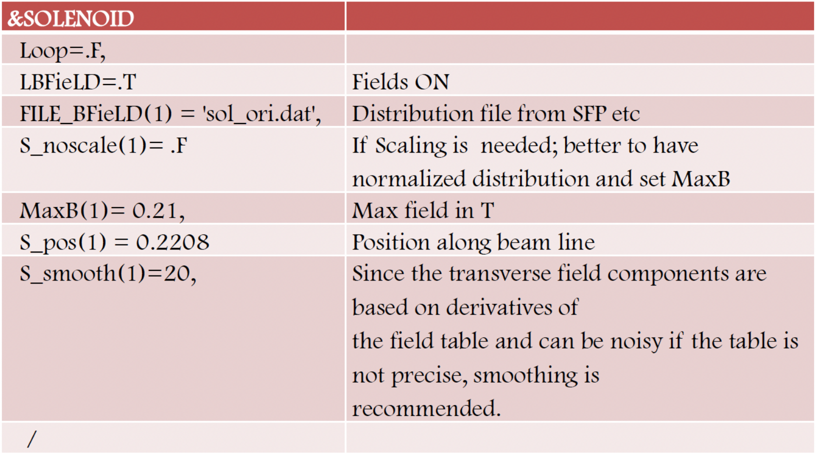
z_1	z_2	n	m	Mode
0.0524	0.15739	1	1	$2\pi/3$



TW Linac

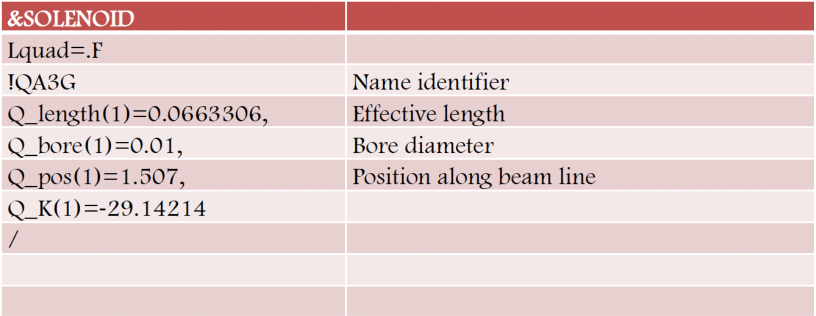
We can define the field and place the linac in &CAVITY:
FILE_EField(2) = 'TWS_Sband.dat'
Nue(2)=2.856, MaxE(2) = 30.0, Phi(2) = -110.0
C_pos(2) = 2.15, C_numb(2) = 9
C_pos(i) defines the z position at which the cavity is placed;
C_numb(j) gives the total number of cells

&Solenoid



&SOLENOID	
Loop=.F,	
LBFieLD=.T	Fields ON
FILE_BFieLD(1) = 'sol_ori.dat',	Distribution file from SFP etc
S_noscale(1)= .F	If Scaling is needed; better to have normalized distribution and set MaxB
MaxB(1)= 0.21,	Max field in T
S_pos(1) = 0.2208	Position along beam line
S_smooth(1)=20,	Since the transverse field components are based on derivatives of the field table and can be noisy if the table is not precise, smoothing is recommended.
/	

&Quadrupole

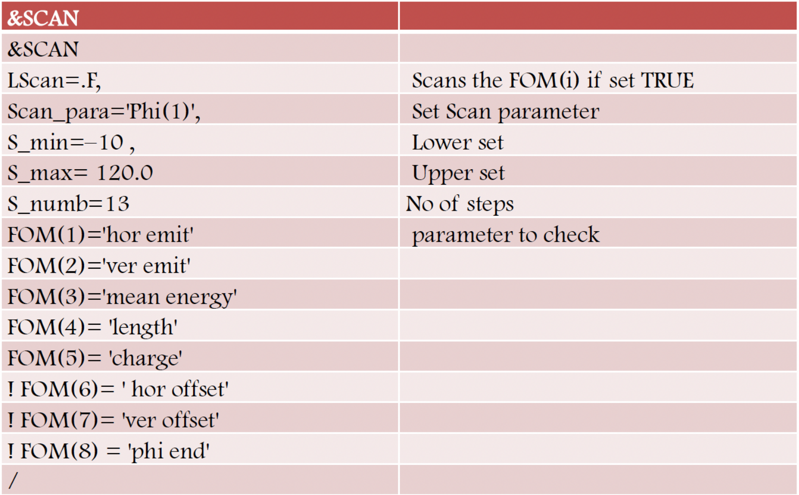


&SOLENOID	
Lquad=.F	
!QA3G	Name identifier
Q_length(1)=0.0663306,	Effective length
Q_bore(1)=0.01,	Bore diameter
Q_pos(1)=1.507,	Position along beam line
Q_K(1)=-29.14214	
/	

Optimizing

How can we use ASTRA to optimize our beamline?
i.e. to make sure that we have the least emittance, beam size...?
We can use &SCAN to define our parameter and scan for the best value.

&Scan

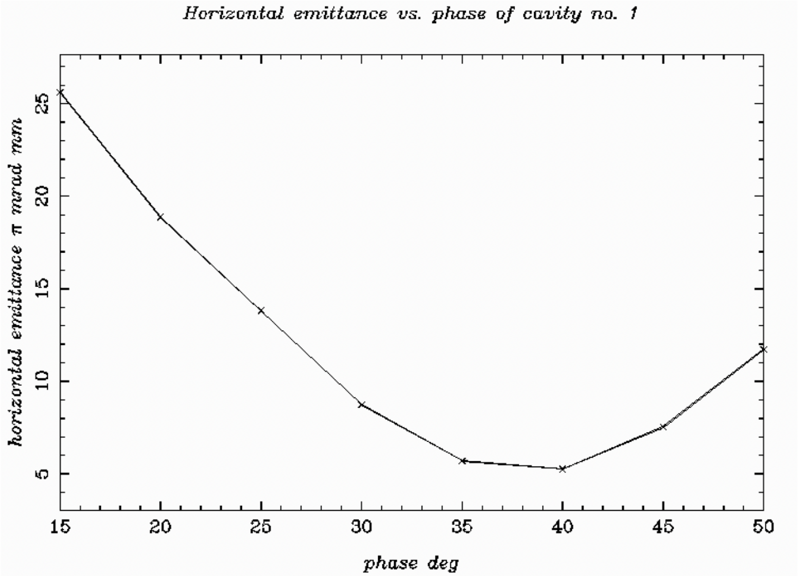


&SCAN	
&SCAN	
LScan=.F,	Scans the FOM(i) if set TRUE
Scan_para='Phi(1)',	Set Scan parameter
S_min=–10 ,	Lower set
S_max= 120.0	Upper set
S_numb=13	No of steps
FOM(1)='hor emit'	parameter to check
FOM(2)='ver emit'	
FOM(3)='mean energy'	
FOM(4)= 'length'	
FOM(5)= 'charge'	
! FOM(6)= ' hor offset'	
! FOM(7)= 'ver offset'	
! FOM(8) = 'phi end'	
/	

Emittance Scan

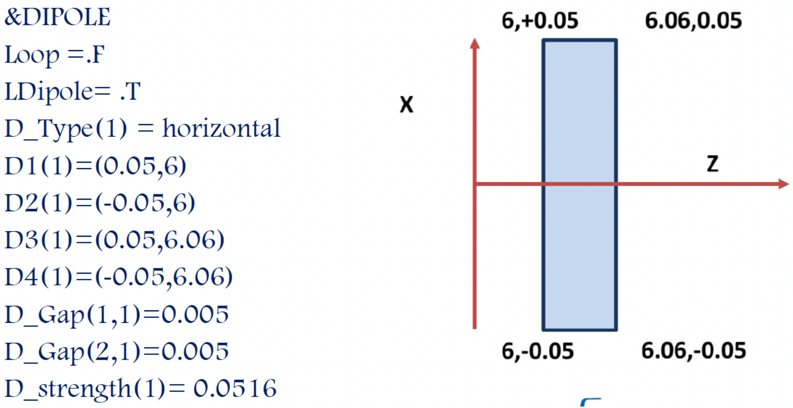
```
in &SCAN:
LScan=.T
Scan_para='Phi(1)', ! ¡– scan over the phase
S_min=15, S_max=50, S_numb=8 ! ¡– min, max, steps
FOM(1)='hor emit' ! ¡– main figure of merit FOM(2)='ver emit' !
¡– FOM 2 .
.
/
```

Emittance Scan



Dipoles

We can include dipoles by defining their entrance and exit faces. To do so, we have to provide the four corners of the magnet (as seen from above) in (x,z) coördinates:



```
&DIPOLE
Loop =.F
LDipole= .T
D_Type(1) = horizontal
D1(1)=(0.05,6)
D2(1)=(-0.05,6)
D3(1)=(0.05,6.06)
D4(1)=(-0.05,6.06)
D_Gap(1,1)=0.005
D_Gap(2,1)=0.005
D_strength(1)= 0.0516
```