Faktor2: Usage

Warner Bruns, CERN*

November 15, 2007

Abstract

This writeup describes the usage of FAKTOR2. The syntax and semantics of its input is described. The resulting data is described.

*This work is supported by the Commission of the European Communities under the Framework Programme “Structuring the European Research Area”, contract number RIDS-011899
Contents

1 Introduction ................................. 1
   1.1 General remarks about the input .................... 1
       1.1.1 Comments .................................. 1
       1.1.2 M4 variables and expressions ................. 1

2 f2-2d, the 2D variant .................... 2
   2.1 -general ..................................... 2
   2.2 -debug ........................................ 3
   2.3 -beam .......................................... 4
   2.4 -ionisation .................................... 5
   2.5 -photoelectrons ................................ 6
   2.6 -impact ........................................ 8
   2.7 -bfield ........................................ 12
   2.8 -monitor,-charge ................................ 12
   2.9 -monitor,-witnesses ............................ 13
   2.10 -mesh ......................................... 14
   2.11 -mesh,-refineregion ............................ 16
   2.12 -mesh,-brick .................................. 17
   2.13 -mesh,-ellipse ................................ 18
   2.14 -mesh,-gcylinder ................................ 18
   2.15 -mesh,-material ................................ 19
   2.16 File format of the main result file main.data .... 21
   2.17 2D Examples .................................. 22
       2.17.1 Electron cloud buildup in an elliptical waveguide 22
       2.17.2 Trapped Ions ............................... 28

3 f2-3d, the 3D variant .................... 35
   3.1 -mesh .......................................... 35
   3.2 -mesh,-refineregion ............................. 36
   3.3 -mesh,-brick .................................... 38
   3.4 -mesh,-ellipse .................................. 38
   3.5 -mesh,-gcylinder ............................... 39
1 Introduction

Faktor2 is an implementation of the Particle in Cell algorithm to simulate the buildup of electron clouds. Faktor2 can also compute the trajectories of ions trapped in the potential field of a beam.

The charges move in a volume which is bounded by perfectly conducting material. The shape of the geometry is arbitrary. The volume may be partly filled with a dielectric material. An arbitrary number of electrodes with prescribed potential may be present.

The charges are accelerated by their self forces and by the forces of a beam field. A static magnetic field may be prescribed.

There are two variants of Faktor2. A 2D variant and a 3D variant. The first section describes the 2D variant.

1.1 General remarks about the input

The input consists of keywords followed by numbers or strings. The numbers must be simple numbers, not expressions. But one can have expressions in an input file, which is preprocessed by some preprocessor which can evaluate expressions. In this document, the preprocessor is assumed to be M4. M4 is a variant of the m4 preprocessor, expanded such that it can handle floating point expressions. The original unix m4 command can only handle integer expressions.

1.1.1 Comments

Everything behind a # or a sequence of two hyphens -- is considered a comment and is skipped by the parser. The # also introduces a comment for M4. Everything behind a # is not expanded by M4.

1.1.2 M4 variables and expressions.

A variable is introduced and its value is defined for M4 by a sequence like define(Name, Value). Name must be a simple string consisting only of alphanumeric characters. Value may be any string, containing anything, except #) and the newline character. After defining that variable with name Name, each occurrence of the string Name is replaced by the string Value.

Numeric expressions are evaluated by M4 if they are enclosed by eval( and ).
2 f2-2d, the 2D variant

2.1 -general

In this section you give the values for general parameters of the computation. An example input is given below.

```
-genera1
  outfile= /tmp/UserName/Directory-of-results  # String
cmax= 2.6e3  # the maximum time to be simulated.
    # Given in C * TMax [metres].
text= "Some string describing the simulation parameters."
nclouds= 50000
combine= yes
plots= no
```

Specifies that the general parameters of the simulation shall be specified. Internally, this has the effect that the parser calls the 'GeneralEditor' subroutine.

-outfile= SOME-STRING Specifies the directory where the resultfiles shall be written to. This directory must exist. It is NOT created by the program.

-ctmax= SOME-NUMBER Specifies after what simulated range the program will stop the simulation. This range is given in metres and is the time times the velocity of light after which the simulation shall stop. If you want to simulate a bunch train which has a length of 100 metres, the value for ctmax shall be 100. ctmax= 100.

-text= "Some text" Specifies some text which is written to some kind of resultfiles, ie. the ones which can be plotted by plotmtv.

-nclouds= SOME-NUMBER Specifies how many macro particles shall be allowed in the simulation. If the number of macroparticles exceeds this number, some macroparticles are combined or are deleted. What happens depends on the value given for combine= no, combine= yes.

-combine= no, combine= yes Specifies what strategy to use to reduce the number of macroparticles below the number given by nclouds= SOME-NUMBER. If combine= no during the simulation macroparticles are blindly deleted until the number of macroparticles is sufficiently low. The deleted charge is spread evenly over the remaining charges. If combine= yes during the simulation macroparticles are merged with macroparticles which are positionally nearby and also have almost the same velocity, ie. are nearby in phase space.

-plots= no, plots= yes Specifies whether debug plots shall be produced. These plots are mainly for debugging and are not useful for the general user.

Default values

```
outfile= /tmp/Username/--some-directory--/
ctmax= 1.0e30
text= "no description given"
```
nclouds = 50000
combine = no
plots = no

2.2 -debug

In this section you give the values for some parameters which were useful for debugging.
In general, none of the default parameters shall be changed, as the default values are
good values. An example input is given below.

```
-debug
  staticamplitude = 1  # Factor to apply for the self force computation.
  temamplitude = 1  # Factor to apply for the force due to the
                   # TEM-field of the beam.
  normdt = 10        # Relative time step width.
  vinterpolation = 2 # What variant of force interpolation to use
  cloudsize = 1.0    # size of the macroparticles relative to their
                     # enclosing grid.
  pictures = no      # create debugging plots of internal fields.
  verbose = no       # If yes, be verbose while computing.
```

-debug  Specifies that the internal parameters of the simulation shall be changed.
Internally, this has the effect that the parser calls the 'DebugEditor' subroutine.
staticamplitude = SOME-NUMBER  If SOME-NUMBER is zero, no self force due to the
electrostatic field of the charges itself is applied. The particles then only are accelerated
due to the TEM-field of the bunches. If SOME-NUMBER is nonzero, the electrostatic field
of the macroparticles is computed and used for accelerating the particles. The true
acceleration is applied when SOME-NUMBER is 1.

temamplitude = SOME-NUMBER  If SOME-NUMBER is zero, the TEM-field is not used to
accelerate the particles. If SOME-NUMBER is 1, the true acceleration due to the TEM-field
is applied to the macro particles.
normdt = SOME-NUMBER  Specifies a factor by which the overall timestep is changed.
When SOME-NUMBER is 1, the overall timestep is such that a particle traveling with
the speed of light cannot traverse a cell of the coarsest grid within a timestep. When
SOME-NUMBER is 10, the overall timestep is 10 times as large.
vinterpolation = 1, vinterpolation = 2  When vinterpolation = 1, each compo-
nent of the force acting on a macroparticle is interpolated from the four(2D) or eight(3D)
nearest fieldcomponents in the grid. This does NOT give zero self force for a particle of
cloudsize=1 far away from boundaries. When vinterpolation = 2, each component of
the force acting on a macroparticle is interpolated from the six(2D) or twelve(3D) nearest
fieldcomponents in the grid. This does give zero self force for a particle of cloudsize=1
far away from boundaries.
cloudsize = SOME-NUMBER  Specifies the cloudsize of a macro particle relative to the
size of the grid where it is located in. This parameters affects over how many cells
the particles charged is distributed. \texttt{cloudsize= 1} together with \texttt{vinterpolation= 2} deposits the charge of a macroparticle such that its self force is zero.

\texttt{pictures= no, pictures= yes}  If \texttt{pictures= yes}, plotfiles for various internal states are created. They are created in the current directory.

\texttt{verbose= no, verbose= yes}  If \texttt{verbose= yes}, a lot of internal states are written to standard output.

\textbf{Default values}

\begin{verbatim}
staticamplitude= 1
temamplitude= 1
normdt= 10
vinterpolation= 2
cloudsize= 1.0
pictures= no
verbose= no
\end{verbatim}

\subsection*{2.3 -beam}

In this section you give the values for parameters which describe the beam. The beam is assumed to be rigid, and its velocity is the velocity of light. The beam consists of bunches. Each bunch has the same shape.

\begin{verbatim}
-beam
    bunchelectrons= -11.5e10 # number of negative elemental charges
    # in a bunch.
    xsigma= 1e-3, ysigma= 0.1e-3, zsigma= 20e-3
    bunchspacing= 7.48
    irep= 10   # Number of repetitions
    ito1= 44   # Number of bunches in the first part of train
    dtn1= 1   # relative population in the first part of train
    ito2= 15   # number of bunches in the second part of train
    dtn2= 0   # relative population in the second part of train
\end{verbatim}

\texttt{-beam}  Specifies that the parameters of the bunch train shall be specified. Internally, this has the effect that the parser calls the 'BeamEditor' subroutine.

\texttt{bunchelectrons= BUNCHELECTRONS}  Specifies the number of negatively charged elemental charges in a single bunch of the bunch train. For a proton beam with 11.5e10 protons per bunch, the parameter shall be negative 11.5e10, ie. \texttt{bunchelectrons= -11.5e10}.

\texttt{xsigma= XSIGMA, ysigma= YSIGMA, zsigma= ZSIGMA}  The parameters for a single bunch. The sigmas are the sigmas of a beam with gaussian density in x, y and z. All sigmas are measured in metres.

\texttt{bunchspacing= SPACING}  Specifies the distance between two bunches. The value is given in metres.
irep=IREP, ito1=ITO1, ditn1=DITN1, ito2=ITO2, ditn2=DITN2  
IREP specifies how many bunch trains shall be considered. Each bunchtrain consists of ITO1 bunches with a population of DITN1 * BUNCHELECTRONS negatively charged elemental charges, followed by ITO2 bunches with a population of DITN2 * BUNCHELECTRONS negatively charged elemental charges.

Default values

# No defaults for bunchelectrons, xsigma ysigma, zsigma, bunchspacing
irep= 4,  
ito1= 80, ditn1= 1.0  
ito2= 40, ditn2= 0.0

2.4 -ionisation

In this section you specify the parameters for modeling a gas ionisation. The macroparticles are created at positions that form a density like the x-y density of the primary beam, they are created at the position of the beam. If ions and electrons are created, they are created as pairs, ions and electrons at the same positions such that the initial net charge density is zero. The macroparticles which model the gas ionisation can consist of electrons and positively charged ions. The mass of the ions can be arbitrary. The creation of the positively charged ions can be suppressed. The creation of the electrons can be suppressed.

# Gasionisation
-ionisation
  ions= no  
electrons= yes  
ionmass= 28  
# The mass of the ions (if they are created).

# pairsperbunch is the number of macroparticles per bunch passage
pairsperbunch= 2000
pairsperparticlepermetre= 0.25e-6 # Probability

# An initial charge can be put into the computational volume
# to speed up the first electron cloud buildup.
initialcharge= eval( - 1e-4 * 1.6e-19 ) # As/m

-ionisation  
Specifies that the parameters of the gasionisation shall be specified. Internally, this has the effect that the parser calls the 'Gas ionisationEditor' subroutine. 
ions= yes, ions= no  
If ions=yes, positively charged ions are created and tracked.. 
The ratio of charge to mass of these ions will be the charge of a proton divided by (mass of a proton multiplied by the value given by ionmass). For creating and tracking CO ions with 28 nucleons, one specifies ions=yes, ionmass=28.
electrons= yes, electrons= no  If electrons=yes, negatively charged electrons are created and tracked.

pairsperbunch= NUMBER  Specifies the number macroparticle pairs that shall be created per bunch passage.

pairsperparticlepermetre= PROBABILITY  Specifies the probability of the gasionisation. The number of created electron ions pairs per metre and per bunch is PROBABILITY times the number of particles per bunch.

initialcharge=CHARGEVALUE  An homogeneous initial charge cloud can be specified which can be used to speed up the buildup of the electron cloud by seeding. If specified, the charge is spread homogeneously over the computational volume.

Default values

ions= no
electrons= yes
ionmass= 28
pairsperbunch= 500
pairsperparticlepermetre= 0.25e-6
initialcharge= 0

2.5  -photoelectrons

In this section you specify the parameters for modeling the creation of photoelectrons. The electrons are created when the primary beam passes. The number of photoelectrons created, and the number of macroparticles used to model the photoelectrons can be specified. The initial position of the photoelectrons are computed by seeding them in a user specified circular region, and moving them from that region away until they come near to a metallic boundary. These positions are the initial positions of the macroparticles used to model the photoelectrons.

define(PEY, 0.001)
define( AWIDTH, eval(2*22e-3) )
define( BWIDTH, 30e-3 )
define( CWIDTH, eval(2*18e-3) )

define( AHEIGHT, eval(2*18e-3) )
define( BHEIGHT, eval(2*3e-3) )
define( CHEIGHT, eval(2*18e-3) )

-photoelectrons

# Description of a photoelectron distribution which
# consists of 10 percent of the photoelectrons created with a
# circular distribution
# and 90% created with a distribution where all photoelectrons
Figure 1: Grid of an elliptical chamber with antechamber. The crosses are at the positions where the photoelectrons are initially created during the first bunch passage. The macroparticles in the elliptical chamber are created with a circular distribution. The macroparticles in the antechamber are created with a gaussian like distribution.

```plaintext
# are created on the side x>AWIDTH/2 + BWIDTH

macroparticlesperbunch= 500,
electronsperparticlepermetre= eval( 0.1 * PEY)
#
# circular distribution:
# the particles are moved away from the center of the startregion:
#
xstart= 0, ystart= 0, rstart= eval(AHEIGHT/4),
xpush= 0, ypush= 0
    # circular distribution, because xpush=xstart, ypush=ystart
doit # Record current parameter set, switch to next set.

macroparticlesperbunch= 500,
electronsperparticlepermetre= eval( 0.9 * PEY)
#
# Elliptic distribution:
# The particles are moved away from a point
# outside of the start region.
#
  xstart= eval(AWIDTH/2 + BWIDTH + CWIDTH/2), ystart= 0
  rstart= eval(BHEIGHT)
xpush= eval(AWIDTH/2 + BWIDTH), ypush= 0
  doit # Record current parameter set, switch to next set.
```
-photoelectrons  Specifies that the parameters of the photoelectron modelling shall be specified. Internally, this has the effect that the parser calls the 'PhotoElectronEditor' subroutine.

electronsperparticlepermetre= PROBABILITY  Specifies the probability of the photoelectron creation. The number of created electrons per metre is PROBABILITY times the number of particles per bunch.

macroparticlesperbunch= NUMBER  Specifies the number macroparticles that shall be created per bunch passage.

xstart=X0, ystart=Y0, rstart=R0, xpush=X1, ypush=Y1  The macroparticles are initially created at positions evenly distributed within a circular region with radius R0 with center (x,y)=(X0,Y0). Just after creation, the macroparticles are moved on straight lines away from the position (x,y)=(X1,Y1) until they are near to a electric conducting material.

doit  The current parameters are recorded as the parameters of a photoelectron distribution. A new set of parameters can be edited.

**Default values**

```plaintext
# The default is to not create photo electrons,
# as no 'doit' is implied.
macroparticlesperbunch= 500
electronsperparticlepermetre= 0.001
xstart= 0, ystart= 0, rstart= 1e-3
xpush= 0, ypush= 0
# no 'doit' is implied.
```

### 2.6 -impact

In this section you specify the parameters for modeling the creation of electrons when electrons hit the boundary of the beampipe.

-impact

```plaintext
yieldmax= 1.4
maxatev= 230  #
escdf= 0.7  # Reflectivity at zero energy.

# What description to take for a position dependent yield.
yieldfile= pathname/some-impactfile

# Where to log the dose of charges hitting the wall.
logimpactfile= pathname/filename

logxlow= eval( - 1.1 * HORIZONTAL/2 )
logxhigh= eval( 1.1 * HORIZONTAL/2 )
logdx= eval(HORIZONTAL / 100)
```
-impact Specifies that the parameters of the impact modelling shall be specified. Internally, this has the effect that the parser calls the 'ImpactEditor' subroutine.

yieldmax=SEYMAX, maxatev=MAXATEV, escdf=ESCDF Specifies the secondary emission yield at positions where it is not specified via the information in the yieldfile. The secondary emission yield is modeled with a function which depends on the energy of the particles impacting. The function has its maximum at an energy of MAXATEV electron volts and the value of the secondary emission yield is SEYMAX there. ESCDF specifies the value of reflectivity at zero energy.

yieldfile= YIELDFILE When YIELDFILE is not -none-, the secondary emission yield as a function of x-y is read from file. The expected file format is such that a previously written LOGFILE can be used.

logimpactfile= LOGFILE When LOGFILE is not -none-, the charge and energy of the macroparticles hitting the wall are integrated. Every XXX timesteps these integrated values are stored to file in a self explanatory format. The data is recorded on a grid which is specified by logxlow=XLOW, logxhigh=XHIGH, logdx=DX, logylow=YLOW, logyhigh=YHIGH, logdy=DY.
The borders of the grid are given by logxlow=XLOW, logxhigh=XHIGH and logylow=YLOW, logyhigh=YHIGH. The x- and y-spacing of that grid is given by logdx=DX, logdy=DY.
The impactfiles will have a content like:

```
33000  1.2867660E-07  1.0000E+00 <- it, t, Scale
-1.0000E-01 -1.0000E-01  1.0000E-03  1.0000E-03 <- x0, y0, dx, dy
200  200 <- nx, ny
```
# The following are the data of the cells, written by a loopnest like:
# DO iy= 0, ny, 1
#    DO ix= 0, nx, 1
#       IF (Cell(ix,iy)% iCount .GT. 0) THEN
#          WRITE (iUnit,*) ix,iy, Cell(ix,iy)%iCount, Cell(ix,iy)%Data(:)
#       ENDIF
#    ENDDO
# ENDDO
###
# ix iy iCount Dose Energy NewSEY UsedSEY SummedDose
###
# Start of data:

Default values
yieldfile= -none-
logimpactfile= -none-
logxlow= -1e20
logxhigh= 1e20
logylow= -1e20
logyhigh= 1e20
logdx= 1e20
logdy= 1e20
Figures 2 and 3 show the modeled dependence of the secondary emission yield on the parameters yieldmax, maxatev and escdf.

Figure 2: Secondary emission yield (SEY) as a function of energy. Four different sets of parameters are used.

Figure 3: Secondary emission yield (SEY) as a function of energy. Zoom of the dependence for lower energies. Four different sets of parameters are used.
2.7 -bfield

In this section you specify the parameters of a homogeneous static magnetic field.

```
-bfield
  bsx= 0
  bsy= 1.9 # Value in Tesla [Vs/m**2]
? return
```

2.8 -monitor,-charge

In this section what properties of the charges shall be monitored and written to files. Currently, the monitoring of charge densities as aequichargeplots, the monitoring of the macroparticles state as a table and the monitoring of HISTOGRAMDATA as plotmtv files is implemented. The average charge density within circles of two different radii is recorded. These radii are specified in this section.

```
-monitor
  -charge
    histogramfilenames= SomePath/SomeFile
    particlefilenames= SomePath/SomeOtherFile
    densityfilenames= SomePath/SomeThirdFile

    firststored= 0
    laststored= 1e9
    distance= eval(BUNCHSEPARATION/10) # 0.1 # c*dt

    radius1= 1e-3
    radius2= 2e-3
```

-charge specifies that the parameters of the monitor facilities shall be specified. Internally, this has the effect that the parser calls the 'ChargeMonitorEditor' subroutine. The -charge is a subsection of the -monitor section, but it should also work to only specify -charge.

-charge Specifies that the parameters of the monitor facilities shall be specified. Internally, this has the effect that the parser calls the 'ChargeMonitorEditor' subroutine. The -charge is a subsection of the -monitor section, but it should also work to only specify -charge.

histogramfilenames=PATHNAME If PATHNAME is not -none-, information about the distribution of the charge per macroparticle is written at the selected times. The filenames will be PATHNAME-NUMBER.mtv, where NUMBER is a 9 digit number. This has been implemented for debugging purposes and is not very useful.

particlefilenames=PATHNAME If PATHNAME is not -none-, the macroparticles state is written at the selected times. The filenames will be PATHNAME-NUMBER.mtv, where NUMBER is a 9 digit number. The file format is self explanatory. The information is: The actual time, and the charge, position and velocity of each macroparticle at that time. The beginning of such a file looks as follows:


4.1079169E-01 <= ct
# Charge, (Pos(i), Velo(i), i=1,3)
-1.74E-14  1.36E-03  2.21E+03  8.23E-03 -1.49E+06  1.10E-07  4.36E+03
-1.74E-14  -3.73E-03  9.60E+03  -7.33E-03  1.43E+06 -8.94E-07 -1.21E+04

densityfilenames=PATHNAME  If PATHNAME is not -none-, the charge density is written at the selected times in a format suitable to be plotted by plotmtv. The filenames will be PATHNAME-NUMBER.mtv, where NUMBER is a 9 digit number.
firststored=CT0  The first position when the files shall be written. The value given is measured in metres, and is interpreted as a time times velocity of light. If CT0 is 1 metre, the first set of files is written when the first primary bunch has traveled over a distance of 1 metre.
distance=DCT  The datasets are written at times when the first primary bunch has reached positions CTi= CT0 + i*DCT. DCT is measured in metres, and is interpreted as a time times velocity of light.
laststored=CTMAX  No dataset will be written after the first bunch has traveled more than CTMAX.
radius1=R1, radius2=R2  The average charge density within circles of two different radii is recorded. R1 and R2 are these radii. The average charge densities are written to the OUTFILE/main.data file. They are in the 11.th and 12.th column.

Default values

    densityfilenames= -none-
    particlefilenames= -none-
    histogramfilenames= -none-
    firststored= 1e30
    laststored= 1e30
    distance= 1e30
    radius1= 1
    radius2= 1

2.9 -monitor,-witnesses

In this section you may specify the properties of witness charges. These charges are created within some region at some time. They are accelerated by the same field as the main charges, but they do not create an electric field. Their charge is ignored. The positions of these witness particles are written at selected times to files. Each file contains only data of one witness particle. The file format is suitable to be plotted by plotmtv.

    -monitor
    -witnesses
        chargeovermass= eval( 1.6e-19 / ( IONMASS * 1836 * 0.911e-30 ) )

13
\[ dx = \text{eval}(\, 2\times \text{XSIGMA} / 20 \,) \]
\[ \text{bbxlow} = \text{eval}(\, -2\times \text{XSIGMA} \,), \text{bbxhigh} = \text{eval}(\, 2\times \text{XSIGMA} \,) \]
\[ \text{bbylow} = \text{eval}(\, -2\times \text{YSIGMA} \,), \text{bbyhigh} = \text{eval}(\, 2\times \text{YSIGMA} \,) \]
\[ \text{ctmin} = \text{CT} \]
\[ \text{name} = \text{Set1} \]
\[ \text{doit} \]
\[ dx = \text{eval}(\, \text{XSIGMA}/2 / 20 \,) \]
\[ \text{bbxlow} = \text{eval}(\, -\text{XSIGMA}/2 \,), \text{bbxhigh} = \text{eval}(\, \text{XSIGMA}/2 \,) \]
\[ \text{bbylow} = \text{eval}(\, -\text{YSIGMA}/2 \,), \text{bbyhigh} = \text{eval}(\, \text{YSIGMA}/2 \,) \]
\[ \text{ctmin} = \text{CT} \]
\[ \text{name} = \text{Set2} \]
\[ \text{doit} \]

\textit{-witnesses} \hspace{1em} \text{Specifies that the parameters of witness particles shall be specified. Internally, this has the effect that the parser calls the 'WitnessesEditor' subroutine.}

\textit{chargeovermass=QOM} \hspace{1em} \text{The ratio of charge over mass of the witness particles. This ratio defines how the particles are accelerated by the field. A ratio of chargeovermass=\[ \frac{1}{\text{NUMBER} \times 1836 \times 0.911 \times 30} \] is the value to specify for witness particles which move like single ionised positive ions with NUMBER protons and neutrons. A ratio of chargeovermass=\[ -\frac{1.6 \times 10^{-19}}{0.911 \times 30} \] is the value to specify for witness particles which move like electrons.}

\textit{dx=DX} \hspace{1em} \text{The distance between the starting point of the particles.}

\textit{bbxlow=X0, bbylow=Y0, bbyhigh=X1, bbyhigh= Y1} \hspace{1em} \text{The witness particles are created in a rectangular region encloses by X0, X1, Y0, Y1. The distance between the initial positions is DX.}

\textit{ctmin=CT} \hspace{1em} \text{The witness particles are created after simulating a time in which the relativistic charge has traveled a distance of CT.}

\textit{name=NAME} \hspace{1em} \text{The data of the witness particles are written to files}

\textit{doit} \hspace{1em} \text{The over parameters are recorded as the parameters of a set of witness particles. The next set of parameters can be entered.}

\textbf{Default values}

\[ \text{-- no default values.} \]

\textbf{2.10 -mesh}

In this section you specify the parameters which define the mesh. The borderplanes of the mesh are specified and the grid spacing. The mesh may be recursively refined. The regions where refinement occurs may may be specified.
-mesh

define( SPACING, eval( 0.0175 / 11 ) )
xmin= eval( -( 0.0760 + 2*SPACING ))
xmax= eval( ( 0.0760 + 2*SPACING ))
ymin= eval( -( 0.0175 + 2*SPACING ))
ymax= eval( ( 0.0175 + 2*SPACING ))

spacing= SPACING

refinelevels= 1
refinebeam= no
refinewall= yes

-mesh Specifies that the parameters of the mesh shall be specified. Internally, this has the effect that the parser calls the 'MeshEditor' subroutine.

xmin=XMIN, xmax=XMAX, ymin=YMIN, ymax=YMAX, spacing=SPACING Specifies the borderplanes of the mesh. The grid at refinelevel 0 will have

- its lowest x-plane at x=XMIN,
- its lowest y-plane at y=YMIN,
- its highest x-plane at x=XMAX,
- its highest y-plane at y=YMAX.

The grid spacing is homogeneous, all x-planes have the same distance between them, the distance will be about SPACING. All y-planes have the same distance between them, the distance will be about SPACING.

refinelevels=LEVEL If LEVEL is greater than 0, the grid will be refined at some places. These places may be the region where the primary beam is traveling, regions near the beampipe’s wall, or user defined regions. If LEVEL is 1, only cells of the grid of level 0 are considered for refinement. These refined cells are the grid of level 1. If LEVEL is N, cells of grids of level N-1 are considered for refinement.

refinebeam=yes | refinebeam=no When specified as refinebeam=yes, cells where a significant fraction of the primary beam is traveling are marked for refinement.

refinewall=yes | refinewall=no When specified as refinewall=yes, cells which are near to the beampipe’s wall are marked for refinement.

Default values

xmin= -1e30, ymin= -1e30, xmax= 1e30, ymax= 1e30, spacing= 1e30
refinelevels= 0
refinebeam= no
refinewall= no
2.11 -mesh, -refinerregion

In this section you specify a region which shall be refined. If you do not specify any such region, the whole computational volume is considered as candidate for refinement. If you specify such a region, only the gridcells which are within such a region are candidates for refinement.

-mesh

-refinerregion
  xlow= -0.03, xhigh= 0.03
  ylow= -1000, yhigh= 1000
  forcerefine= no
  doit

-refinerregion Specifies that the parameters of the refinement region shall be specified.

xlow=XLOW, ylow=YLOW, xhigh=XHIGH, yhigh=YHIGH Specifies the borderplanes of the region.

forcerefine=yes | forcerefine=no If forcerefine=yes, all cells within the region are marked for refinement. If forcerefine=no, the cells within the region are mere candidates for refinement, ie. the must also fulfil some other condition to be marked for refinement. Eg. they must be near to the beampipe’s wall or they must carry a substantial fraction of the primary’s beam current.

doit This enters the current values specified into the list of refinement regions. The parameters of the next region may be edited.

Default values There are no default values.

Example

-mesh
  spacing= 0.45e-3
  xmin= -0.0169, xmax= 0.0169
  ymin= -0.0099, ymax= 0.0099

  refinelevels= 1
  refinebeam= no, refinewall= yes

-refinerregion
  xlow= -0.5e-2, xhigh= 0.5e-2
  ylow= -1000, yhigh= 1000
  forcerefine= no
  doit

-brick
Figure 4: Grid of an elliptical chamber with parts of the grid near the wall refined.

2.12 -mesh, -brick

In this section you specify a rectangular region which shall be filled with some material.

-mesh
   -brick
       material= 1
       xlow= -1000
       ylow= -1000
       xhigh= 1000
       yhigh= 1000
       doit
-brick    Specifies that the parameters of the rectangular region. Internally, this has the effect that the parser calls the 'BrickEditor' subroutine.

material=MAT    Specifies which material index shall be used for points within the rectangular region. The properties of the material can be specified in the section -material. The default values for materials with index 0 and index 1 are: index 0 corresponds to vacuum, index 1 corresponds to a perfectly conducting material with potential zero.

xlow=XLOW, ylow=YLOW, xhigh=XHIGH, yhigh=YHIGH    Specifies the borderplanes of the region.

doit    This enters the current values specified into the list of rectangular regions. The parameters of the next brick can be entered.

Default values    There are no default values.

2.13 -mesh, -ellipse

In this section you specify an elliptical region which shall be filled with some material.

-mesh

-ellipse

material= 0
xcenter= 0, ycenter= 0
xextension= 0.0760
yextension= 0.0175
doit

-ellipse    Specifies that the parameters of an elliptical region shall be specified. Internally, this has the effect that the parser calls the 'EllipseEditor' subroutine.

material= MAT    Specifies which material index shall be used for points within the elliptical region.

xcenter=X0, ycenter=Y0    Specifies the center of the elliptical region.

xextension=XWIDTH, yextension=YHEIGHT    Specifies the x-extension and y-extension of the ellipse.

doit    This enters the current values specified into the list of elliptical regions. The parameters of the next ellipse can be entered.

Default values    There are no default values.

2.14 -mesh, -gcylinder

In this section you specify a general region enclosed by a polygon which shall be filled with some material.
-mesh
  -gcylinder
    material = 0
    px = 6.2157732E-02, py = -2.4675742E-02, addpoint
    px = 6.2499452E-02, py = -2.3292582E-02, addpoint
    px = 6.2821433E-02, py = -2.1904696E-02, addpoint
    px = 6.3123606E-02, py = -2.0512361E-02, addpoint
    px = 6.3405916E-02, py = -1.9115865E-02, addpoint
    px = 6.3668303E-02, py = -1.7715488E-02, addpoint
    px = 6.3910708E-02, py = -1.6311513E-02, addpoint
    px = 6.4133085E-02, py = -1.4904228E-02, addpoint
    px = 6.4335391E-02, py = -1.3493918E-02, addpoint
    px = 6.4517580E-02, py = -1.2080869E-02, addpoint
    px = 6.4679623E-02, py = -1.0665366E-02, addpoint
    px = 6.4821482E-02, py = -9.2476998E-03, addpoint
    px = 6.4943127E-02, py = -7.8281555E-03, addpoint
    px = 6.5044545E-02, py = -6.4070229E-03, addpoint
    px = 6.5125696E-02, py = -4.9845893E-03, addpoint
    doit

-gcylinder  Specifies that the parameters of an region enclosed by an arbitrary
            polygon shall be specified.
material= MAT  Specifies which material index shall be used for points within
            the elliptical region.
px=Xi, py=Yi, addpoint  Specifies the coordinates of the next point of the enclos-
                       ing polygon. When the parser encounters addpoint, the number of points on the
px= Xi, py= Yi are recorded for the new point.
doit  This enters the current values specified into the list of general regions. The
parameters of the next gcylinder can then be entered.

Default values  There are no default values.

2.15 -mesh, -material

In this section you specify the physical parameters for materials. There may be up to
11 different materials. The material indices must be between 0 and 10.

-mesh
  -material
    material = 3, epsr = 1, muer = 1, fixedpotential = no
    material = 4, fixedpotential = yes, potential = -100
    material = 5, fixedpotential = yes, potential = 100
material=MAT  Specifies that the following specifications shall be applied to the ma-
material with index MAT.
epsr=EPSR, muer=MUER  Specification of the relative permittivity and permeability
of the material.

Default values  All material parameters are initially:

    epsr= 1, muer= 1, fixedpotential= no, potential= 0

Material 1 is an exception, it is initialised to be an electric conducting material with
potential 0.

    material= 1, fixedpotential= yes, potential= 0
2.16 File format of the main result file main.data

At the beginning of the dataset, the input as read by f2-2d is replicated as comments. The rest of the file consists of 12 columns.

- 1: jb: Number of bunch
- 2: ib: number of dataset
- 3: t: time
- 4: nPositive: Number of positively charged macroparticles
- 5: nNegative: Number of negatively charged macroparticles
- 6: ChargePositive: Total charge in positive macroparticles, As/Metres
- 7: ChargeNegative: Total charge in negative macroparticles, As/Metres
- 8: DensityOnAxis: Charge density on the axis, As/metres**3
- 9: Joule/s/metre: Power deposited on the wall
- 10: Joule/metre: Integrated power deposited on the wall
- 11: DensityNearAxis1: Charge density within radius1 of section -monitor, As/Metres**3
- 12: DensityNearAxis2: Charge density within radius2 of section -monitor, As/Metres**3

The file format is readable by gnuplot.
2.17 2D Examples

2.17.1 Electron cloud buildup in an elliptical waveguide

# M4 < Ellipse.M4 | f2-2d | tee logfile

define(NCLOUDS, 50000)

    # Definition of a symbol which is used several times.
    # For the definition in what directory to write the 'main.data' file,
    # and for some monitor files.
    define(OUTFILE, /tmp/UserName/Ellipse)

    # gnuplot
    # plot '/tmp/UserName/main.data' u ($3*3e8):(-$7/1.6021892e-19)

    -debug
        staticamplitude= 1
        temamplitude= 1

        # for vinterpolation=2, cloudsize=1.0 is optimal.

        vinterpolation= 2
        cloudsize= 1.0       # relative Cloudsize
        normdt= 10           # relative dt

    -general
        plots= yes              # Generate plots of the mesh.
        outfile= OUTFILE        # Where to store the results
        ctmax= 2.6e3            # After which time to stop the simulation.
        text= "Ellipse; nclouds= NCLOUDS"
        nclouds= NCLOUDS
        combine= yes            # If there are more clouds than NCLOUDS,
                                # combine nearby clouds to decrease the number below
                                # NCLOUDS.

    -beam
        # Number of negative elementary charges in the bunch.
        bunchelectrons= -11.5e10

        # Gaussian width of the bunch
xsigma= 0.003, ysigma= 0.0015, zsigma= 0.2

# properties of the bunch train.
bunchspacing= 7.48 # Bunchspacing in metres (c*t).
irep= 4 # Four bunchtrains
ito1= 72, ditn1= 1 # Each bunchtrain has 72 bunches, relative charge of '1'.
ito2= 21, ditn2= 0 # followed by 21 empty bunches, relative charge of '0'.

# Show the parameters.

# properties of the gas-ionisation.
-ionisation

# properties of a homogeneous background charge (electrons),
# which are present at the beginning of the simulation.
# This is for seeding. Not really needed.
initialcharge= eval( -1e4 * 1.6e-19 ) # As/m

ions= yes, ionmass= 12 # What is the relative mass of ions created by
# gasionisation?
ions= no # Override the previous 'ions=yes'.
# No ions will be created and therefore are also not
# tracked. This speeds up the computation somewhat.
# In the case of electron cloud buildup, the ion density
# is much smaller than the electron density anyway,
# and does not have a substantial effect.

electrons= yes # Create (and track) electrons via gasionisation.
pairsperbunch= 500.0 # create 500 macropairs per bunch passage.
pairsperparticlepermetre= 0.25e-6 # ionisation probability.

# Model the secondary emission when particles hit the walls.
-impact
 yieldmax= 1.5 # what is the maximum of the SEY?
 maxatev= 200 # At what energy?
 escdf= 0.7 # What is the reflectivity at zero energy?

# Show the parameters.
-bfield
    bsx= 0  # No x-component of the static magnetic field.
    bsy= 1.0  # y-component is 1 Tesla
    bsy= 1.0  # Gibt recht kleine Werte
    bsy= 0.01
?  # Show the parameters

# -monitor: What additional results shall be written to files?
-monitor
-charge
    radius1= 1e-3  # The first radius in which the charge is monitored.
    radius2= 5e-3  # The second radius

define(DCT, eval(7.48/10) )
    firststored= eval( 590 * DCT )
    laststored= eval( 620 * DCT )
    distance= DCT
    histogramfilenames= -none-  # no monitoring of the histogram
    particlefilenames= -none-  # no storing of the particles properties.

    # The charge densities shall be written to files
    # in the resultfiles directory whichs name has been specified
    # via 'outfile= XXX' in the section general.
    # Within that directory, the files will be named
    # 'Q-iSeq-*.mtv'.
    densityfilenames= Q-iSeq-

?  # Show the parameters.

# ############
# Specify properties of the mesh.
-mesh
define( SPACING, eval( 0.0175 / 11 ) )

    # The extension of the mesh.
    # The grid should be slightly larger than the vacuum contained.
    xmin= eval( -( 0.0760 + 2*SPACING ) )
    xmax= eval( ( 0.0760 + 2*SPACING ) )
    ymin= eval( -( 0.0175 + 2*SPACING ) )
    ymax= eval( ( 0.0175 + 2*SPACING ) )
    spacing= SPACING

    refinelevels= 1  # How many recursion levels may be applied.
refinebeam= no  # Do not refine where the primary beam is.
refinewall= yes  # Refine near the wall.
? # Show the parameters.

# Where to refine.
# Several regions may be specified.
-refineregion
    xlow= -0.03, xhigh= 0.03
    ylow= -1000
    yhigh= 1000
    forcerefine= no
    doit

# Definition of the material filling.

# Fill the 'universe' with metal.
-brick
    material= 1  # '1' is electric conducting.
    xlow= -1000
    ylow= -1000
    xhigh= 1000
    yhigh= 1000
    doit

# Carve out an ellipse
-ellipse
    material= 0  # '0' is vacuum.
    xcenter= 0
    ycenter= 0
    xextension= 0.0760
    yextension= 0.0175
    doit

#############
# definition of the electric properties of materials.
-material
    material= 0, epsr= 1, muer= 1, fixedpotential= no, ?
    material= 1, fixedpotential= yes, potential= 0, ?
    material= 3, fixedpotential= yes, potential= +1, ?
    material= 4, fixedpotential= yes, potential= -1, ?
##############

-base  # Back to the base level.

?
doit       # Compute.

The result file is `/tmp/UserName/Ellipse/main.data`. The file's content is described in Section 2.16. To get a plot of the total electrons per metre as a function of distance of the primary beam, start gnuplot and give gnuplot the input:

```
plot '/tmp/UserName/Ellipse/main.data' u ($3*3e8):(-$7/1.6e-19)
```

Figure 5: Electrons per metre in the elliptical chamber. The x-axis is c*t.
Figure 6: Grid of the elliptical chamber. The mesh near the wall is refined in regions where $-3cm < x < 3cm$. 
2.17.2 Trapped Ions

```plaintext
# M4 < Krakowski-0.M4 | f2-2d | tee Ausgabe-Krakowski

# define(IONMASS, eval(12+16)) # C + O
define(IONMASS, 2) # H2

define(INF, 10000)

-general
    outfile= /tmp/UserName/Krakowski-ionmass-IONMASS
    ctmax= 3e3 # Distance to simulate, C*T.
    combine= yes ### If too many macroparticles are in the simulation,
        ### merge macroparticles that are nearby in phase space.
    nclouds= 10000

# 2007-07-25
#
# - At SOLEIL we have a beam of electrons so the instability is due to
#   ions created by ionisation of the residual gas in the vacuum chamber.
# - There are 7.09E9 electrons per bunch (3/4 filling with a total current
#   of 0.3A).
# - The rms bunch length of 6 mm
# - The distance between bunches is 0.85m
# - There are 312 bunches in a bunchtrain (in 3/4 filling)
# - The beam gap is about 88.5m
# - The cross section of the beampipe is 12.5mm vertically and 35mm
#   horizontally
# - There is not any static magnetic field
#
# In fact, it seems that the pressure is much larger in the three
# in-vacuum insertion devices "U20" that we have at SOLEIL. There the
# pressure is about 5.0E-9 mbar instead of 1.5E-9 mbar elsewhere. Thus, we
# want to simulate the effect of the FBII only in those three U20 (or may
# be put together in one). Besides, there are very heavy ion species there
# (of mass 69 and even 119) which could have a preponderant role in FBII
# but unfortunately the vacuum physicists have trouble determining
# precisely which molecule is responsible for the creation of such ions.
# So for the moment we want to assume that there are about 80% of
# dihydrogen (H2) with a cross section of 0.30 Mbarn (at the energy of
# 2.75 GeV) which means about 2.7E-9 ions created per electrons and per
# meter (assuming a pressure of 5E-9 mbar), and 20% of Carbon Monoxide
# (CO) with a cross section of 1.73 Mbarn which means about 3.9E-9 ions
```
# created per electrons and per meter.
#
# If you need it, the cross-section of the beampipe is the one
# of an elliptic beampipe. But actually the cross-section of the bunches
# is the most important for FBII: x-sigma is 2.6E-4 and y-sigma is 1.1E-5
# for the bunches in the U20.
#

define(XSIGMA, 2.6e-4)
define(YSIGMA, 1.1e-5)
define(ZSIGMA, 6e-3)
define(CT, 0.85)

-beam
  bunchelectrons= 7.09e9 # Number of negative elementary charges.
  xsigma= XSIGMA, ysigma= YSIGMA, zsigma= ZSIGMA
  bunchspacing= CT

  irep= 4
  ito1= 312, ditn1= 1
  ito2= eval(88.5/CT), ditn2= 0

-ionisation
  ions= yes
  ionmass= IONMASS
  electrons= no # yes
  pairsperbunch= 500
  pairsperparticlepermetre= 2.7e-9 # Probability

-impact
  yieldmax= 0 # No simulation of secondary electrons,
  # when yieldmax is zero.

-bfield
  bsx= 0
  bsy= 0 # No magnetic field.

-monitor
  -charge
    radius1= XSIGMA # Charge density within that radius is monitored.
    radius2= XSIGMA # Also the density within that radius.
    densityfilenames= Q-iSeq-
particlefilenames= particles-
histogramfilenames= Histogram-
firststored= 0, laststored= 1e9
distance= eval( CT / 10 ) # 0.1 # c*ct

? define(PROTON_CHARGE, 1.6021892e-19)
define(PROTON_MASS, eval(1836.15 * 0.9109534e-30))  
-witnesses
  chargeovermass= eval( PROTON_CHARGE / ( IONMASS * PROTON_MASS ) )
dx= eval( YSIGMA / 5 )
bbxlow= eval( -10*YSIGMA ), bbxhigh= 0
bbylow= eval( -2*YSIGMA ), bbyhigh= 0
cxmin= CT
cdt= 1
name= set-1
  # doit

# # ############
-mesh
define( RADIUS, eval(20*XSIGMA) )
define( SPACING, eval( XSIGMA/2 ) )
xmin= eval( -( RADIUS + 2*SPACING ) )
xmax= eval( ( RADIUS + 2*SPACING ) )
ymin= eval( -( RADIUS + 2*SPACING ) )
ymax= eval( ( RADIUS + 2*SPACING ) )
 spacing= SPACING
  refinelevels= 3 # 5 gibt 'array access out of bounds'
  refinelevels= 5 # 5: geht nicht vorwaerts bei sunf90
  refinelevels= 4 #
  refinebeam= yes
  refinewall= no
?  -refineregion
  xlow= -INF
  xhigh= INF
  ylow= -INF
  yhigh= INF
  doit

-brick
The result file is `/tmp/UserName/Krakowski-ionmass-2/main.data`. The file's content is described in Section 2.16. To get a plot of the total charge density (in -e-) within the radius \( r \) as a function of distance of the primary beam, start gnuplot and give gnuplot the input:

```
plot '/tmp/UserName/Krakowski-ionmass-2/main.data' using ($3*3e8):($12/1.6021892e-19)
```
Figure 7: Density of elemental charges near the radius as a function of distance of the primary beam. The x-axis is $c*t$. 
Figure 8: The grid used to compute the ion trapping. The mesh near the region where the primary beam is traveling is recursively refined.
Figure 9: Zoom of the grid used to compute the ion trapping. The mesh near the region where the primary beam is traveling is recursively refined.
3 f2-3d, the 3D variant

The 3D-variant allows computation with a true 3D static magnet field. The properties of this 3D-field are specified via a call to a subroutine. Suitable subroutines for specifying a wiggler field and the end field of a dipole are provided.
The data format of the maindata file is the same as of the 2D-variant.
The input for the 3D variant shares the sections -general, -debug, -beam, -ionisation, -photoelectrons, -impact, -bfield, -monitor,-charge, -monitor,-witnesses, -mesh,-material with the 2D variant.

3.1 -mesh

In this section you specify the parameters which define the mesh. The borderplanes of the mesh are specified and the grid spacing. The mesh may be recursively refined. The regions where refinement occurs may be specified.

```
-mesh
define( SPACING, eval( 0.0175 / 11 ) )
xmin= eval( -( 0.0760 + 2*SPACING ) )
xmax= eval( ( 0.0760 + 2*SPACING ) )
ymin= eval( -( 0.0175 + 2*SPACING ) )
ymax= eval( ( 0.0175 + 2*SPACING ) )
zmin= eval( -ZEXT/2 )
zmax= eval( ZEXT/2 )
spacing= SPACING

czmin= magnetic, czmax= magnetic
zspacing= eval( 2 * SPACING )
refinelevels= 1
refinebeam= no
refinewall= yes
```

-mesh  Specifies that the parameters of the mesh shall be specified. Internally, this has the effect that the parser calls the ‘MeshEditor’ subroutine.
xmin=XMIN, xmax=XMAX, ymin=YMIN, ymax=YMAX, zmin=ZMIN, zmax=ZMAX,
spacing=SPACING  Specifies the borderplanes of the mesh. The grid at refinelevel 0 will have

• its lowest x-plane at x=XMIN,
• its lowest y-plane at y=YMIN,
• its lowest z-plane at z=ZMIN,
• its highest x-plane at x=XMAX,
• its highest y-plane at y=YMAX.
• its highest z-plane at z=ZMAX.

The grid spacing is homogeneous, all x-planes have the same distance between them, the distance will be about SPACING. All y-planes have the same distance between them, the distance will be about SPACING.

The z-spacing can be different from the x-y-spacing. The z-spacing is specified via zspacing=DZ. If the z-spacing shall be different, it must be specified after the specification of the x-y-spacing via spacing=DX. If zspacing is not specified, the value as given for spacing is used.

refinelevels=LEVEL If LEVEL is greater than 0, the grid will be refined at some places. These places may be the region where the primary beam is traveling, regions near the beampipe’s wall, or user defined regions. If LEVEL is 1, only cells of the grid of level 0 are considered for refinement. These refined cells are the grid of level 1. If LEVEL is N, cells of grids of level N-1 are considered for refinement.

refinebeam=yes | refinebeam=no When specified as refinebeam=yes, cells where a significant fraction of the primary beam is traveling are marked for refinement.

refinewall=yes | refinewall=no When specified as refinewall=yes, cells which are near to the beampipe’s wall are marked for refinement.

Default values

xmin= -1e30, ymin= -1e30, zmin= -1e30,
xmax= 1e30, ymax= 1e30, zmax= 1e30,
spacing= 1e30, zspacing= 1e30
refinelevels= 0
refinebeam= no
refinewall= no
czmin= electric, czmax= electric

3.2 -mesh, -refineregion

In this section you specify a region which shall be refined. If you do not specify any such region, the whole computational volume is considered as candidate for refinement. If you specify such a region, only the gridcells which are within such a region are candidates for refinement.

-mesh

-refineregion

xlow= -0.03, xhigh= 0.03
ylow= -1000, yhigh= 1000
zlow= -1000, zhigh= 1000
forceredefine= no
doit
-refineregion  Specifies that the parameters of the refinement region shall be specified.
xlow=XLOW, ylow=YLOW, xhigh=XHIGH, yhigh=YHIGH, zlow=ZLOW, zhigh=ZHIGH: Specifies the borderplanes of the region.
forcerefine=yes | forcerefine=no  If forcerefine=yes, all cells within the region are marked for refinement. If forcerefine=no, the cells within the region are mere candidates for refinement, i.e. they must also fulfil some other condition to be marked for refinement. Eg. they must be near to the beampipe’s wall or they must carry a substantial fraction of the primary’s beam current.
doit  This enters the current values specified into the list of refinement regions. The parameters of the next region may be edited.

Default values  There are no default values.

Example

-mesh
  spacing= 0.45e-3
  xmin= -0.0169, xmax= 0.0169
  ymin= -0.0099, ymax= 0.0099
  zmin= -0.05, zmax= 0.05

  refinelevels= 1
  refinebeam= no, refinewall= yes

-refineregion
  xlow= -0.5e-2, xhigh= 0.5e-2
  ylow= -1000, yhigh= 1000
  zlow= -1000, zhigh= 1000
  forcerefine= no
doit

-brick
  material= 1
  xlow= -1000, ylow= -1000, zlow= -1000
  xhigh= 1000, yhigh= 1000, zhigh= 1000
doit

-ellipse
  material= 0
  xcenter= 0, ycenter= 0
  xextension= 0.016, yextension= 0.009
  zextension= 1e6
doit
3.3 -mesh, -brick

In this section you specify a rectangular region which shall be filled with some material.

-mesh
   -brick
       material= 1
       xlow= -1000, ylow= -1000, zlow= -1000
       xhigh= 1000, yhigh= 1000, zhigh= 1000
doit

-brick    Specifies that the parameters of the rectangular region. Internally, this has
the effect that the parser calls the 'BrickEditor' subroutine.
materia l=MAT    Specifies which material index shall be used for points within the rect-
geangular region. The properties of the material can be specified in the section -material.
The default values for materials with index 0 and index 1 are: index 0 corresponds to
vacuum, index 1 corresponds to a perfectly conding material with potential zero.
xlow=XLOW, ylow=YLOW, zlow=ZLOW, xhigh=XHIGH, yhigh=YHIGH, zhigh=ZHIGH: Specifies the borderplanes of the region.
doit    This enters the current values specified into the list of rectangular regions. The
parameters of the next brick can be entered.

Default values  There are no default values.

3.4 -mesh, -ellipse

In this section you specify an ellipsoidal region which shall be filled with some material.

-mesh
   -ellipse
       materia l= 0
       xcenter= 0, ycenter= 0, zcenter= 0
       xextension= 0.0760, yextension= 0.0175, zextension= 1e6
doit

-ellipse    Specifies that the parameters of an ellipsoidal region shall be specified. Internally, this has the effect that the parser calls the 'EllipsoidEditor' subroutine.
materia l= MAT    Specifies which material index shall be used for points within the ellipsoidal region.
xcenter=X0, ycenter=Y0, zcenter=Z0    Specifies the center of the elliptical region.
xextension=XWIDTH, yextension=YHEIGHT, zextension: Specifies the x-y-z-extension of the ellipsoid.
doit    This enters the current values specified into the list of ellipsoidal regions. The
parameters of the next ellipsoid can be entered.

Default values  There are no default values.
3.5 -mesh, -ecylinder

In this section you specify an elliptical cylinder which shall be filled with some material.

-ecylinder
  material= 0
  origin= ( 0, 0, 0 )
  xprime= ( 1, 0, 0 )
  yprime= ( 0, 1, 0 )
  xprimeextension= 0.0760
  yprimeextension= 0.0175
  zprimeextension= 0.10
  doit

Specifies that the parameters of an elliptical cylinder shall be specified. Internally, this has the effect that the parser calls the 'ECylinderEditor' subroutine.

material= MAT  Specifies which material index shall be used for points within the elliptical cylinder region.

origin= (X0, Y0, Z0)  Specifies the center of the footprint of the elliptical cylinder.

xprime= (XXP, YXP, ZXP)  Specifies the direction of the first semiaxis of the elliptical footprint. yprime= (XYP, YYP, ZYP)  Specifies the direction of the second semiaxis of the elliptical footprint. Only the component perpendicular to the first semiaxis is used. The axis of the cylinder points into the direction of the cross product of both semiaxes.

xprimeextension=AAXIS, yprimeextension=BAXIS, zprimeextension=ZHEIGHT:  Specifies the x-y-z-extension of the elliptical cylinder.

doit  This enters the current values specified into the list of elliptical cylinders. The parameters of the next elliptical cylinder can be entered.

Default values  There are no default values.