How to use MAFOT

Code author: Andreas Wingen
Email: wingen@fusion.gat.com

Relevant Publications: (use [1] to cite the MAFOT code itself please)

[1] A. Wingen et al., *High resolution numerical studies of separatrix splitting due to non-axisymmetric perturbation in DIII-D*, **Nuclear Fusion 49**, 55027 (2009) https://doi.org/10.1088/0029-5515/49/5/055027

[2] A. Wingen et al., *Footprint structures due to resonant magnetic perturbations in DIII-D*, **Physics of Plasmas 16**, 42504 (2009) https://doi.org/10.1063/1.3099053

[3] A. Wingen et al., *Numerical Modeling of Edge-Localized-Mode Filaments on Divertor Plates Based on Thermoelectric Currents*, **PRL 104**, 175001 (2010) https://doi.org/10.1103/PhysRevLett.104.175001

[4] A. Wingen et al., *Impact of plasma response on plasma displacements in DIII-D during application of external 3D perturbations*, **Nuclear Fusion 54**, 64007 (2014) https://doi.org/10.1088/0029-5515/54/6/064007

[5] A. Wingen et al., *Heat flux modeling using ion drift effects in DIII-D H-mode plasmas with resonant magnetic perturbations*, **Physics of Plasmas 21**, 12509 (2014) https://doi.org/10.1063/1.4862034

[6] A. Wingen et al., *New heat flux model for non-axisymmetric divertor infrared structures*, **Nuclear Fusion 61**, 16018 (2021) https://doi.org/10.1088/1741-4326/abbfe9

[7] A. Wingen et al., *Development and validation of non-axisymmetric heat flux simulations with 3D fields using the HEAT code*, **Nuclear Fusion 65**, 086029 (2025) https://doi.org/10.1088/1741-4326/adeff1

Setup:

Check the Readme.txt in the MAFOT source folder.

Table of Contents

1. *Tools in MAFOT* 2
2. *Required files* 3
3. *GUI* 4
4. *Control file* 8
5. *Output file structure* 9
6. *Plot MAFOT output* 10
7. *Xpand* 11

1. Tools in MAFOT

All tools have command line options. Use the -h flag to show help:

```
#> dtfoot_mpi -h
```

The tool binaries have a machine prefix:
dt = DIII-D, iter = ITER, mast = MAST, nstx = NSTX, cmod = C-Mod, **any = agnostic**

The binaries that use MPI for parallel processing have the suffix: _mpi

plot - **returns the Poincaré poloidal cross-section**
For a given toroidal angle this tool traces field-lines (particle drift orbits) and simulates the Poincaré plot.

fix - **returns periodic fixed points (hyperbolical and elliptical)**
It uses a 2D Newton method to search for periodic fixed points. Note that the Newton method requires an initial guess, given by a rectangular search grid, but the algorithm can leave the grid area. For periodicity = 1 the search stops after it finds the first point.

man - **returns the stable and unstable manifolds related to a hyperbolic fixed point**
It needs the fixed point found by “fix”. It uses a step size controlled algorithm to trace the manifold until it intersects three times with the vessel wall. Note that if the manifold is too long for your purposes, just truncate the file.

laminar - **returns the connection length and penetration depth in a poloidal cross-section**
It calculates the data on a rectangular grid, suitable for 2D contour plots. Note that if the field line does not intersect with the wall after the given limit of toroidal iterations, the integration of that field line stops and the length and minimum penetration up to this point is returned.

foot - **returns the connection length and penetration depth footprint on the divertor target**
same as laminar, but on the vessel wall. As coordinates it uses a length parameter along the wall (t, y-axis) and the toroidal angle (phi, x-axis). The plotting script d3dplot.py automatically converts t back into a meaningful length.
Tip: since one rarely knows the proper limits in t, make a very low-resolution run along the entire t-range, plot it with the physical = 0 flag (see options in d3dplot.py), locate the footprint and thereby the proper t-limits. Then do the full resolution run.

structure - **traces the full 3D orbit.**
This computes a full 3D orbit in x,y,z as well as R,phi,Z. Default output is every 10 degrees, but this can be changed with command line flags. This tool is not in the GUI, but can use the same control file as plot.

2. Required files

MAFOT runs are controlled by a control file. Their filenames always start with an underscore and end with '.dat'. The file is generated by the GUI. Example: _plot.dat

MAFOT always needs a g-file. If the code exits directly after launching, check the shot, time and path to the g-file. If the problem persist, check the header line in the g-file, sometimes (rarely) is not standard. This can cause problems. The line should look like this:

```
  EFITD    03/13/2012    #148712  4101ms           3 129 129
```

MAFOT further needs the following files in its current working directory, depending on the *Field* you want to run:

Field: -4=GPEC, -3=VMEC, -2=SIESTA, -1=gfile, M3DC1: 0=Eq, 1=I-coil, 2=both

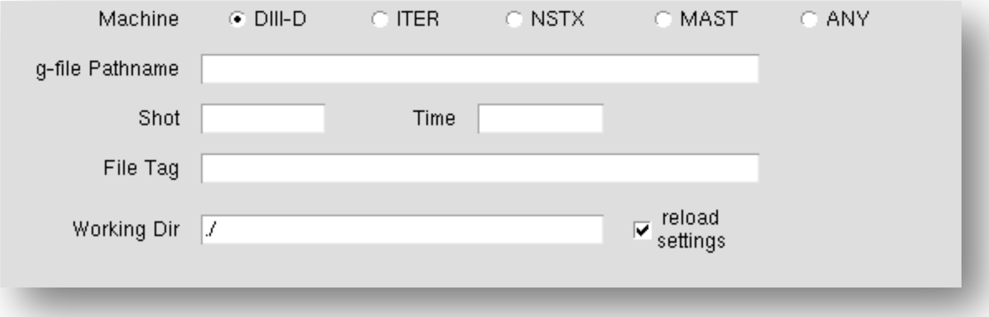
Field:	Files:
g-file Vacuum	**diiidsup.in** (or itersup.in or nstxsup.in): containing the perturbation-coil currents. There are examples for every machine in the MAFOT/example folder.
M3D-C1	**m3dc1sup.in**: a list of the following structure `path/to/files/C1.h5       <scaling factor>    <phase in deg>` You can list up to 10 C1.h5 files, each in a new line. MAFOT will combine the perturbations from all using the provided scaling factor and phase. **All .h5 output files**
VMEC	**wout.nc** and **xpand.dat**: the first is the VMEC wout-file, the second is the output of *xpand*.
SIESTA	**siesta.dat**: the file with the B-field on a 3D rectangular grid, that Siesta optionally returns. Note that SIESTA only works inside the Siesta last closed flux surface so far.
GPEC	**gpecsup.in**: similar as for M3D-C1. The GPEC data file **gpec.out**

3. GUI

The GUI runs in Python. Launch the GUI by calling:

```
#> mafot_gui.py &
```

Common section:



Machine ● DIII-D ○ ITER ○ NSTX ○ MAST ○ ANY

g-file Pathname

Shot Time

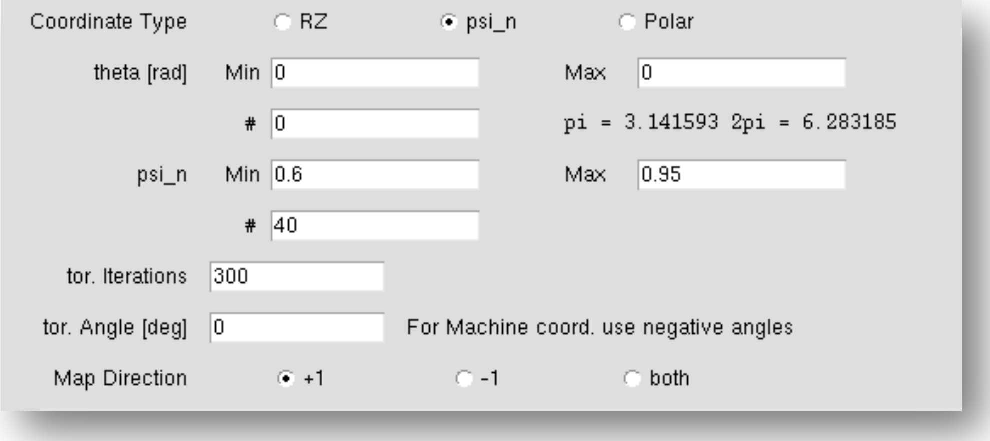
File Tag

Working Dir ./ ☑ reload settings

- Select a machine.
- Enter the path to the g-file. A drop-down menu will appear with auto-complete options.
- Shot and Time will populate automatically.
- *File Tag* is an arbitrary string, appended to all output files
- *Working Dir* selects where to run MAFOT.

The GUI reads already available control files to provide the last settings you used. To prevent that behavior, uncheck the *relod settings* box before changing *Working Dir*.

plot Tab:



Coordinate Type ○ RZ ● psi_n ○ Polar

theta [rad] Min 0 Max 0

# 0 pi = 3.141593 2pi = 6.283185

psi_n Min 0.6 Max 0.95

# 40

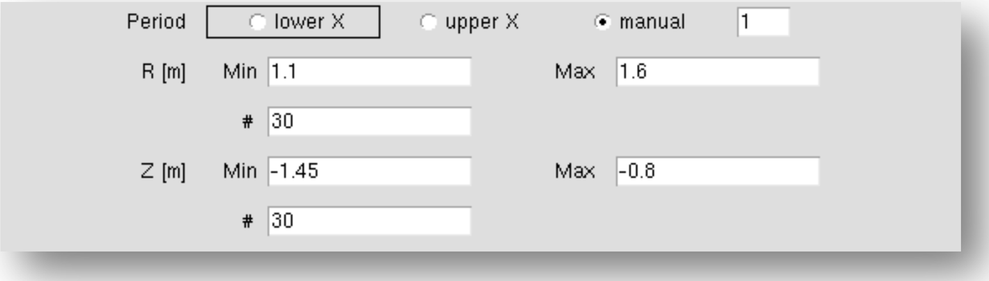
tor. Iterations 300

tor. Angle [deg] 0 For Machine coord. use negative angles

Map Direction ● +1 ○ -1 ○ both

- Select your coordinate system: (R,Z), (theta, psi_n), or (theta, r)
  - theta is the geometric poloidal angle
- Select the ranges and number of grid points (#)
  - Values of pi and 2*pi are provided for copy/paste.
- *tor. Iterations*: number of toroidal turns
- *tor Angle*: mathematically oriented (ccw) toroidal angle phi
  - Use negative for DIII-D machine angles
- *Map Direction*: integrate field line (orbit) in this phi direction

fix Tab:



Period ○ lower X ○ upper X ● manual 1

R [m] Min 1.1 Max 1.6

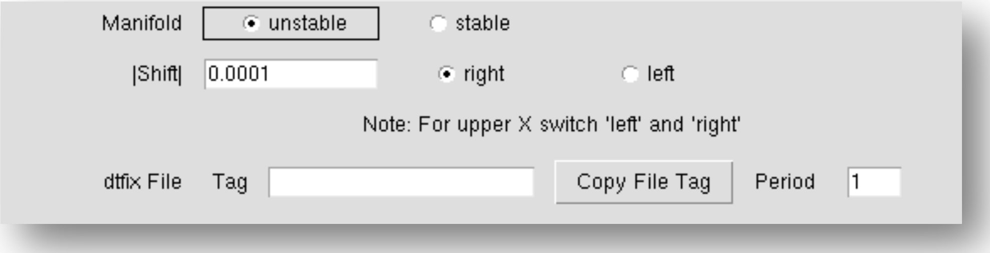
# 30

Z [m] Min -1.45 Max -0.8

# 30

- Select periodicity and search grid. Defaults for lower and upper X-point are provided.
- The rest is same as plot Tab.

man Tab:



Manifold ● unstable ○ stable

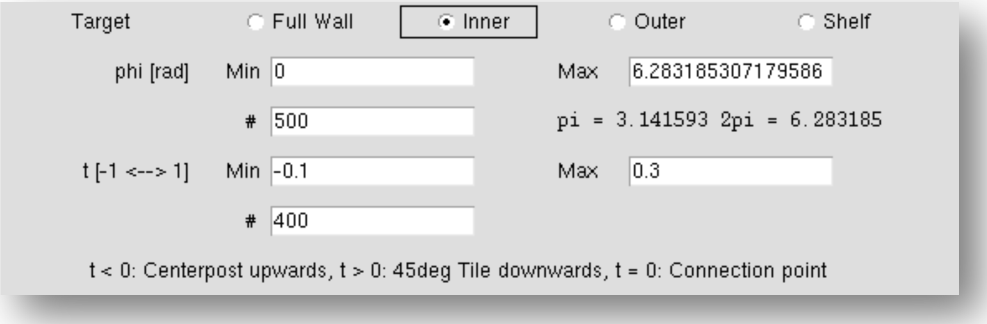
|Shift| 0.0001 ● right ○ left

Note: For upper X switch 'left' and 'right'

dtfix File Tag Copy File Tag Period 1

- Select the type of manifold.
- *Shift*: a guess of how much to move away from X-point. Default usually works fine.
    - Smaller *Shift* makes manifold more accurate, but computation takes longer.
- *right/left*: direction of shift; this selects which branch of the manifold you get.
    - For the separatrix X-point, one branch goes straight to the wall, while the other goes around the plasma; you want the latter.
- *Tag*: man needs the result of fix. Give the *File Tag* you used for fix.
    - The *copy* button fills in the tag that is currently in the common section.
    - *Period*: fixed point periodicity
- The rest is same as plot Tab.

foot Tab:



Target ○ Full Wall ● Inner ○ Outer ○ Shelf

phi [rad] Min 0 Max 6.283185307179586

# 500 pi = 3.141593 2pi = 6.283185

t [-1 <--> 1] Min -0.1 Max 0.3

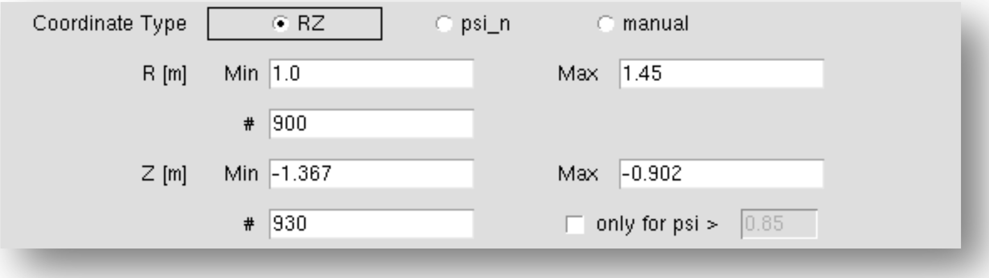
# 400

t < 0: Centerpost upwards, t > 0: 45deg Tile downwards, t = 0: Connection point

- Select the target, meaningful defaults are provided
- Select the grid range.

  - Remember that t is a dimensionless length parameter along the wall. Use the tip given in Sec. 1.
  - The text gives some idea of where t is at the wall.
- The rest is same as plot Tab.

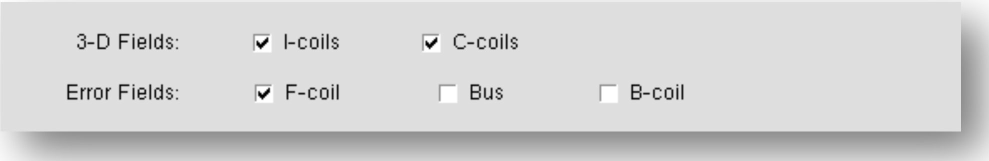
laminar Tab:



Coordinate Type ● RZ ○ psi_n ○ manual
R [m] Min 1.0 Max 1.45
# 900
Z [m] Min -1.367 Max -0.902
# 930 □ only for psi > 0.85

- Select your coordinate system: (R,Z) or (theta, psi_n) and grid range.
- The rest is same as plot section.

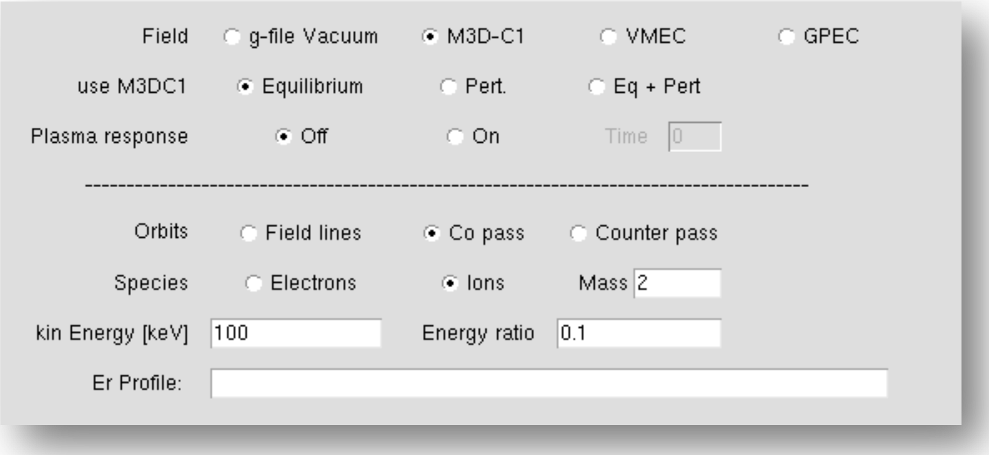
3D Fields section:



3-D Fields: ☑ I-coils ☑ C-coils
Error Fields: ☑ F-coil □ Bus □ B-coil

This is different for different machines, as it depends on their respective 3D coil sets. Check the boxes of the coils or error fields to include. This is all vacuum fields only.

Unfolding sections:



Field ○ g-file Vacuum ● M3D-C1 ○ VMEC ○ GPEC
use M3DC1 ● Equilibrium ○ Pert. ○ Eq + Pert
Plasma response ● Off ○ On Time 0

Orbits ○ Field lines ● Co pass ○ Counter pass
Species ○ Electrons ● Ions Mass 2
kin Energy [keV] 100 Energy ratio 0.1
Er Profile:

- *Field*: source of the magnetic field
- *Orbits*: field lines or guiding center drift particles
  - Particles can move in either direction along their orbit.
    - *Co pass*: with Bphi,
    - *Counter pass*: against Bphi direction
    - Note, that switching *Map Direction* flips *Co* and *Counter*.

Depending on selections for *Field* and *Orbits*, new options unfold and become available.

As example for *M3D-C1*:

- *Equilibrium* uses only the axisymmetric part in C1.h5,
- *Pert.* uses the g-file equilibrium + the perturbation from C1.h5,
- *Eq + Pert* uses equilibrium and perturbation from C1.h5
    - Typically *Eq* is very similar to g-file, but it is a resolve of the Grad Shafranov
- Plasma response: turn on/off effects of plasma response.
    - The off case should give the same result than g-file Vacuum with coils on.

If you use *M3D-C1*: *Pert.* or *Eq+Pert*, the vacuum coils will be set to off, regardless of your selection in the 3D Fields section

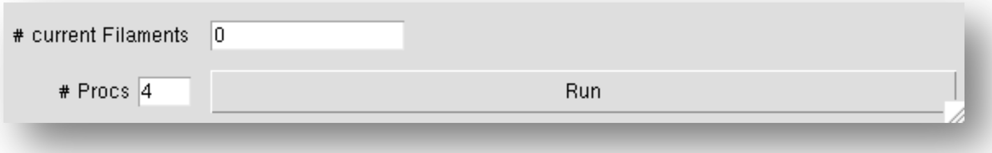
For *VMEC* or *GPEC*:

- Path and file names for the magnetic field data files.

If you use particle orbits, then specify

- *Species* and their atomic *Mass (ions only)*
- *Kinetic Energy* of particles
- *Energy ratio* of how much energy goes into translation vs. gyration.
- Path and file name of a radial electric field profile
    - Same format as for M3D-C1 profile files
    - This is optional

Bottom section



# current Filaments 0

Procs 4

- *# current Filaments*: number of arbitrary line currents to add as an additional perturbation. Requires additional files!
- *# Procs*: if tool is parallel, then you can give the number of openMPI processes here
- *Run Button*: generates control file and launches MAFOT

Launching the code manually

Use calls like

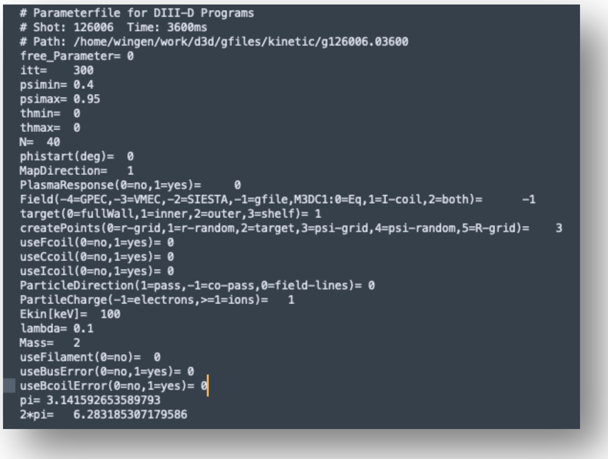
```
#>  dtfix _fix.dat 1 MyFileTag
```

or

```
#>  mpirun –np 4 dtlaminar_mpi _lam.dat MyFileTag
```

4. Control file

All tools need a control file, generated by the GUI. Here is an example:

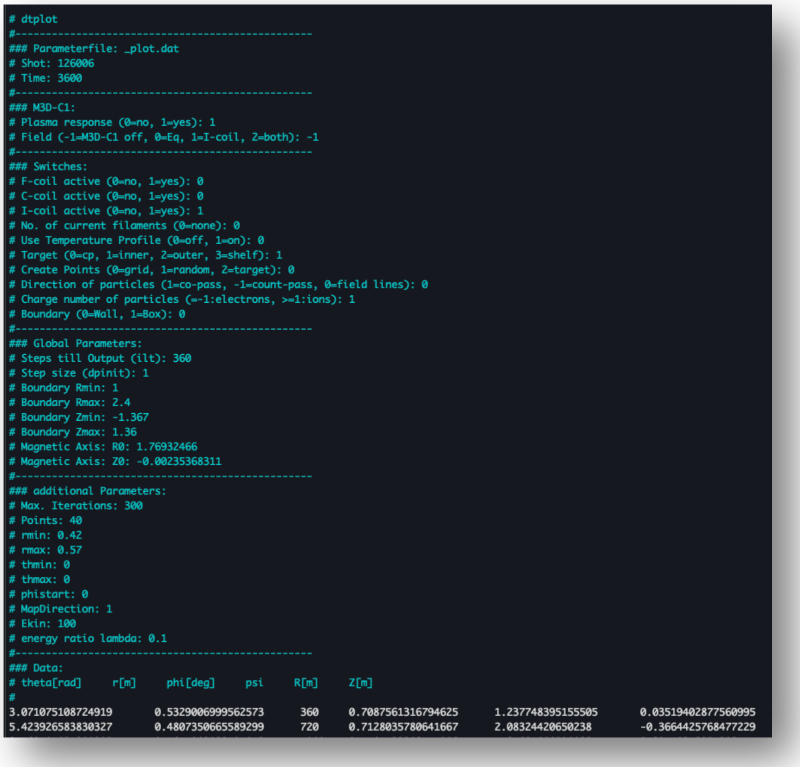


```
# Parameterfile for DIII-D Programs
# Shot: 126006  Time: 3600ms
# Path: /home/wingen/work/d3d/gfiles/kinetic/g126006.03600
free_Parameter= 0
itt=    300
psimin= 0.4
psimax= 0.95
thmin=  0
thmax=  0
N=  40
phistart(deg)=  0
MapDirection=   1
PlasmaResponse(0=no,1=yes)=     0
Field(-4=GPEC,-3=VMEC,-2=SIESTA,-1=gfile,M3DC1:0=Eq,1=I-coil,2=both)=      -1
target(0=fullWall,1=inner,2=outer,3=shelf)= 1
createPoints(0=r-grid,1=r-random,2=target,3=psi-grid,4=psi-random,5=R-grid)=    3
useFcoil(0=no,1=yes)= 0
useCcoil(0=no,1=yes)= 0
useIcoil(0=no,1=yes)= 0
ParticleDirection(1=pass,-1=co-pass,0=field-lines)= 0
PartileCharge(-1=electrons,>=1=ions)=   1
Ekin[keV]=  100
lambda= 0.1
Mass=   2
useFilament(0=no)=  0
useBusError(0=no,1=yes)= 0
useBcoilError(0=no,1=yes)= 0
pi= 3.141592653589793
2*pi=   6.283185307179586
```

You can skip the GUI at any point and edit the control file by hand. Then you have to know what you are doing! The name of any parameter can be arbitrary as long as no spaces are used. The name has no influence on the way the data is read. The data is read by its position in the file.

5. Output file structure

All output files have a header which includes all relevant parameters for the respective run.



```
# dtplot
#-------------------------------------------------
### Parameterfile: _plot.dat
# Shot: 126006
# Time: 3600
#-------------------------------------------------
### M3D-C1:
# Plasma response (0=no, 1=yes): 1
# Field (-1=M3D-C1 off, 0=Eq, 1=I-coil, 2=both): -1
#-------------------------------------------------
### Switches:
# F-coil active (0=no, 1=yes): 0
# C-coil active (0=no, 1=yes): 0
# I-coil active (0=no, 1=yes): 1
# No. of current filaments (0=none): 0
# Use Temperature Profile (0=off, 1=on): 0
# Target (0=cp, 1=inner, 2=outer, 3=shelf): 1
# Create Points (0=grid, 1=random, 2=target): 0
# Direction of particles (1=co-pass, -1=count-pass, 0=field lines): 0
# Charge number of particles (=-1:electrons, >=1:ions): 1
# Boundary (0=Wall, 1=Box): 0
#-------------------------------------------------
### Global Parameters:
# Steps till Output (ilt): 360
# Step size (dpinit): 1
# Boundary Rmin: 1
# Boundary Rmax: 2.4
# Boundary Zmin: -1.367
# Boundary Zmax: 1.36
# Magnetic Axis: R0: 1.76932466
# Magnetic Axis: Z0: -0.00235368311
#-------------------------------------------------
### additional Parameters:
# Max. Iterations: 300
# Points: 40
# rmin: 0.42
# rmax: 0.57
# thmin: 0
# thmax: 0
# phistart: 0
# MapDirection: 1
# Ekin: 100
# energy ratio lambda: 0.1
#-------------------------------------------------
### Data:
# theta[rad]     r[m]     phi[deg]     psi     R[m]     Z[m]
#
3.071075108724919       0.5329006999562573      360     0.7087561316794625      1.237748395155505       0.03519402877560995
5.423926583830327       0.4807350665589299      720     0.7128035780641667      2.08324420650238        -0.3664425768477229
```

- All comment lines start with '#'.
- The data in the columns is double precision, and the columns are separated by tabs.
- The data is labeled at the end of the header, just above the data.

All tools return log files, which store the screen output as well as list error messages. After your run terminates normally you can safely delete the log files. For openMPI codes, the *Master* log file protocols the communication and provides progress information; just look at its tail.

6. Plot MAFOT output

A Python tools is available to plot MAFOT output

d3dplot.py - Can plot MAFOT outputs.

To show the command line help, there are many options, and its syntax, use:

```
#> d3dplot.py -h
```

7. Xpand

xpand - **Calculates the B-field outside of the VMEC last closed flux surface**
It runs separate from all other MAFOT tools and does not need a control file or a g-file. It only needs a VMEC *wout*- and a *points*-file, which contains the list of points where you want to know the B-field

In the MAFOT/python folder there is a Python script that creates the points-file suitable for xpand and necessary for Fields = VMEC. But you can always make one yourself.
A points file looks like:

```
# 128x128x48 points 786432
1.0    0.0    -1.5
...
```

The first column is R[m], the second is phi[rad] and the third is Z[m]. R varies first, then Z, then phi. The first header line is just FYI and has here NRxNZxNphi and the total number of points (You need/are only allowed this one line of comments). xpand is parallel, so it always needs to be launched by mpirun.

To run xpand, look at the shell help by:

```
#> mpirun -n 4 xpand_mpi -h
```

The output is like:

```
# Xpand results from VMEC file wout.nc
# 16384 points
# R[m]        phi[rad]      Z[m]          BR[T]          Bphi[T]        BZ[T]
1      0      -1.5   0.2000096130793928  -3.329098836152724  -0.0142624523069116
...
```

A python tool is available to help with setup of an xpand run.
use_xpand.py - Provides a points file for xpand, loads the output, plots the output a and can "heal" bad points

use_xpand requires a python or ipython session. You have to import it, initialize the class and go from there. See the comments in the file.