flair v2 status

Vasilis.Vlachoudis@cern.ch

flair v2

Flair V2 was released on May 2014 and used for the first time in the 15th Beginners FLUKA course in Thessaloniki Greece.

Currently few resistant to changes persons still use V1

- Major user interface relooking
- Major code restructuring: resulted in less lines with more functionality :) and hopefully more robust
- Many changes towards python v3
- More intuitive actions and behaviour from the program (e.g. USR-1D less options providing the same functionality, renaming, copy&paste in geometry editor...)
- Run-->Data, user customizable commands to process the data
- Keyboard bindings for accessing the ribbon
- Volume calculations (still in development)
- Copy&Paste of plots from project to project

flair v2[cont]

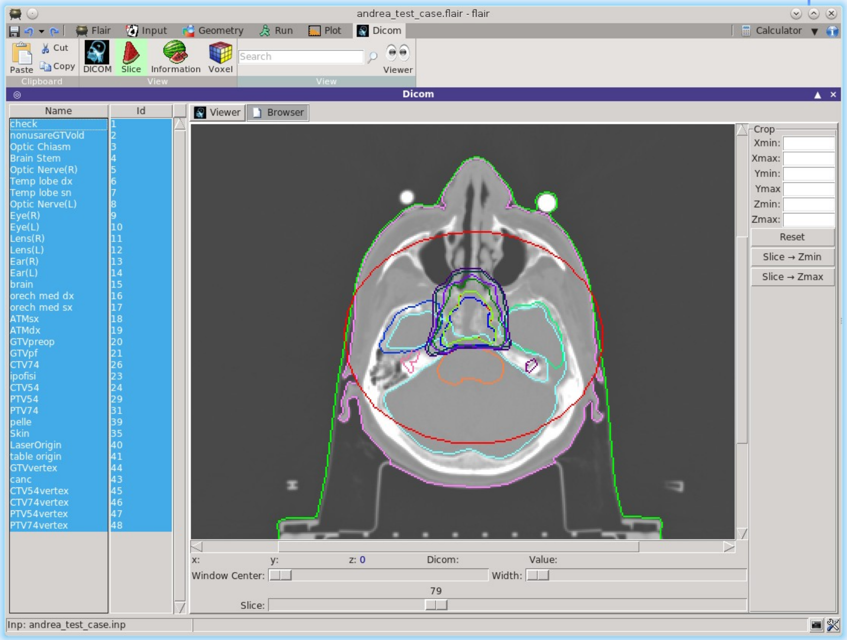
- DXF library to import 2D DXF drawings, converting to connected paths
- Matplotlib bindings (started)
- Strict order checking of $start_transform, translat, expand
- Run job submission and monitoring decoupled from GUI (GA)
- Several optimization
  - Directory caching for faster accessing folders with a lot of files
  - Scanning for current event is done manually
- install.sh for easier update to the svn version
- Manual.py can export the manual in .html format
- More robust threadpool for parallel processing

flair v2[cont2]

- Import from GDML improved (PolyCones and material importing)
- Export to OpenSCAD format
- Lock & Freeze buttons on geometry
- FOV in the 3D plots
- OpenGL bindings in the geometry editor for possibly using it as hardware accelerated viewer
- VOXEL 3D tracking improved (x3) by simplifying the inner loop using only additions and comparisons
- Auto correcting overlapping regions in 2D
- Multiple bugs fixes

DICOM

- .flair format change to accommodate multiple DICOM
- Rewritten DICOM using the numpy, with a remarkable performance improvement in the DICOM processing.
- Tree based dicom browser
- Reading and Plotting of RTSTRUCT
- Inclusion of the RSTRUCT inside the VOXEL file, using a mapping dictionary to describe all structures per each voxel.
- Offline DVH histogram calculations
- Added magnetic field flag in the VOXEL generation



Name	Id
check	1
nonusareGTVold	2
Optic Chiasm	3
Brain Stem	4
Optic Nerve(R)	5
Temp lobe dx	6
Temp lobe sn	7
Optic Nerve(L)	8
Eye(R)	9
Eye(L)	10
Lens(R)	11
Lens(L)	12
Ear(R)	13
Ear(L)	14
brain	15
orech med dx	16
orech med sx	17
ATMsx	18
ATMdx	19
GTVpreop	20
GTVpf	21
CTV74	26
ipofisi	23
CTV54	24
PTV54	29
PTV74	31
pelle	39
Skin	35
LaserOrigin	40
table origin	41
GTVvertex	44
canc	43
CTV54vertex	45
CTV74vertex	46
PTV54vertex	47
PTV74vertex	48

Genetic algorithms

Genetic algorithms optimization python library

For the moment not accessible through the graphical interface

In the field of artificial intelligence, a genetic algorithm (GA) is a search that mimics the process of natural selection. It is commonly used to optimize problems using techniques inspired by natural evolution

Description:

- Each individual is described by a chromosome = a list of alleles (genes)

A	B	C	D	E	F	G	H	I	J

  In flair is a list of #define variables, where each one contains an *int*, *float*, or *list of choices* (e.g. materials)

- Population = a group of individual
  Initially the population is built randomly

Genetic algorithms (cont)

- Fitting function: the value to minimize/maximize, a user supplied function that calculates a single value for each individual

  In flair the fitting function is the outcome of one or many estimators after a FLUKA run

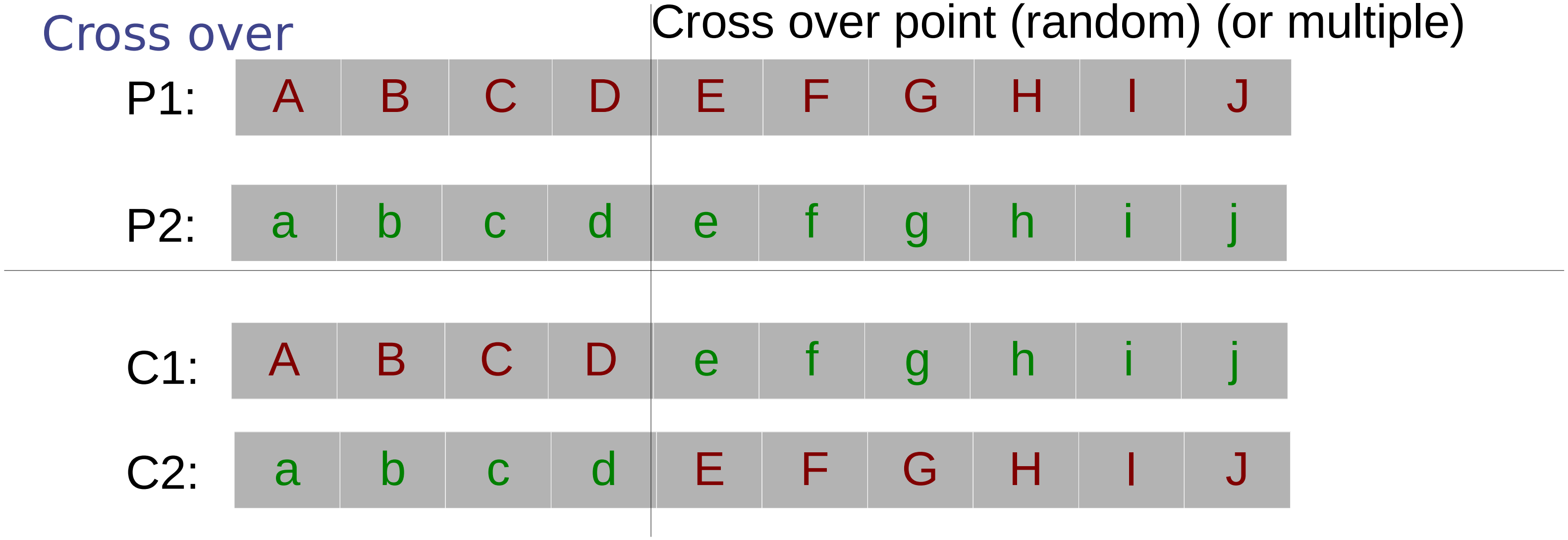
  This is the most complicated step:

  1) To define a proper fitting function
  2) To write the code to extract it
  3) When to stop?

- On each step=generation, the whole population is evaluated (with the fitting function) and sorted
- A new generation (child) population is generated using the following operations: Selection, Cross Over, Mutation

Genetic algorithms (cont)

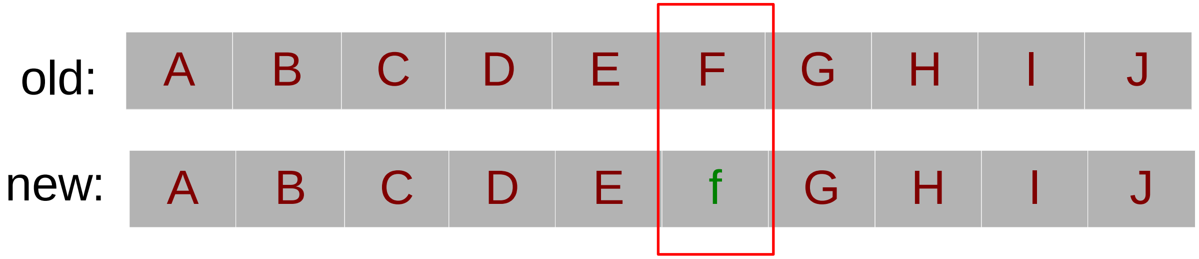
- Selection: selecting randomly the fittest parents
- Cross over



Cross over point (random) (or multiple)

P1:	A	B	C	D	E	F	G	H	I	J
P2:	a	b	c	d	e	f	g	h	i	j
C1:	A	B	C	D	e	f	g	h	i	j
C2:	a	b	c	d	E	F	G	H	I	J

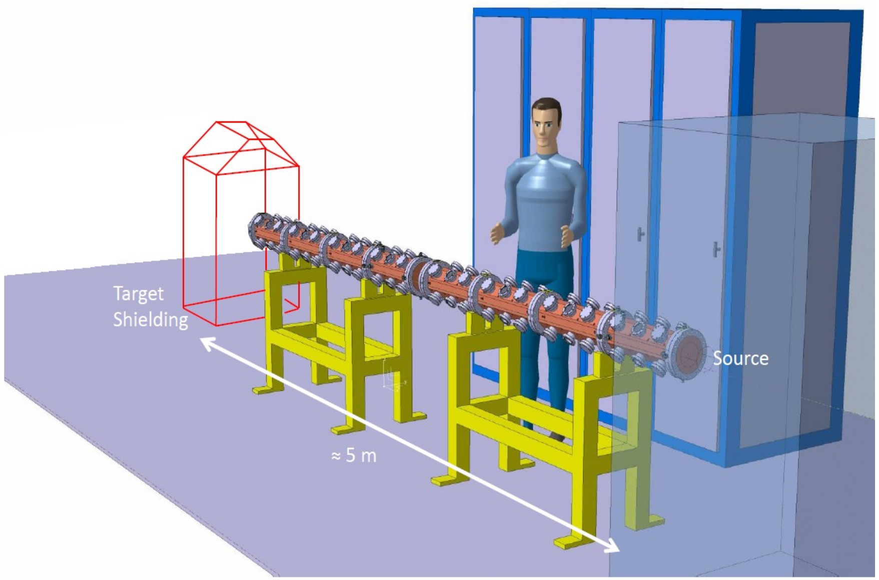
- Mutation (very low probability 0.2%)

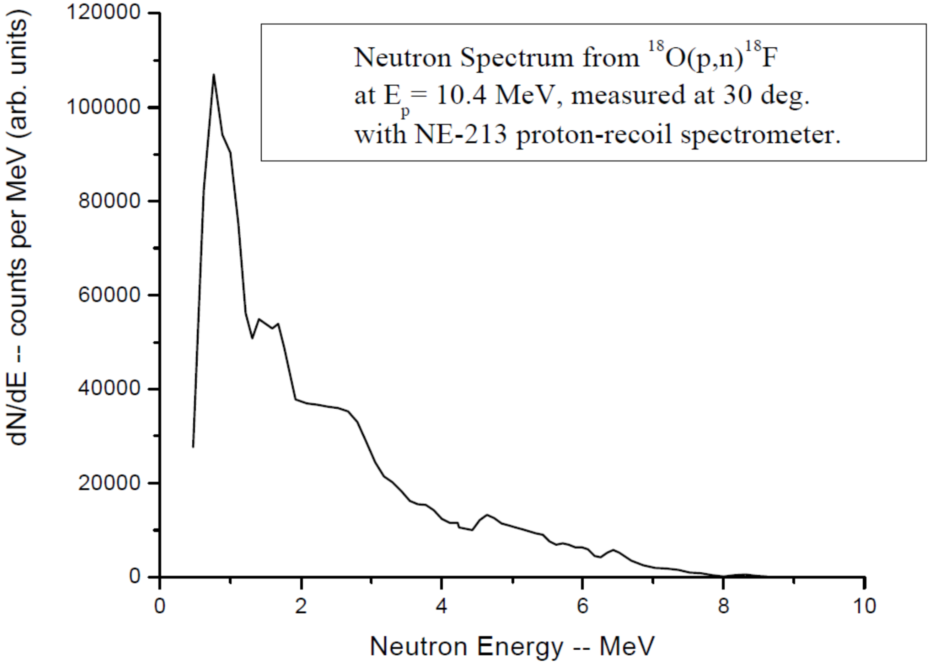


old:	A	B	C	D	E	F	G	H	I	J
new:	A	B	C	D	E	f	G	H	I	J

Example: HF RFQ optimization

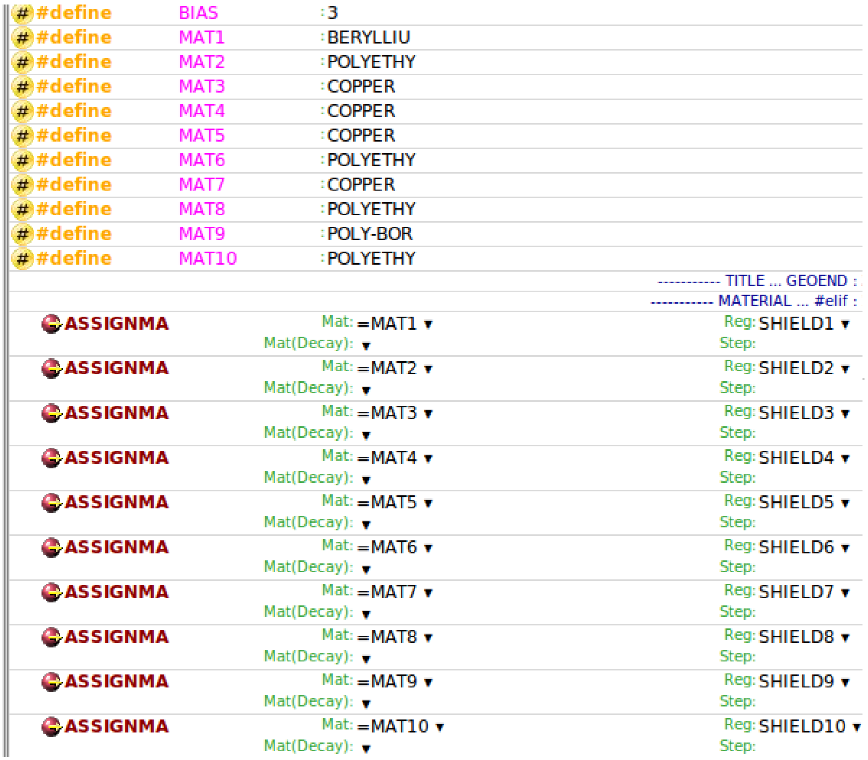
- Shielding calculations for the HF-RFQ from the neutrons out of the $^{18}O(p,n)^{18}F$
- For 20uA p the saturated value gives 1.26 10^{11}n/s
- Goal to find the best configuration to minimize the outside dose



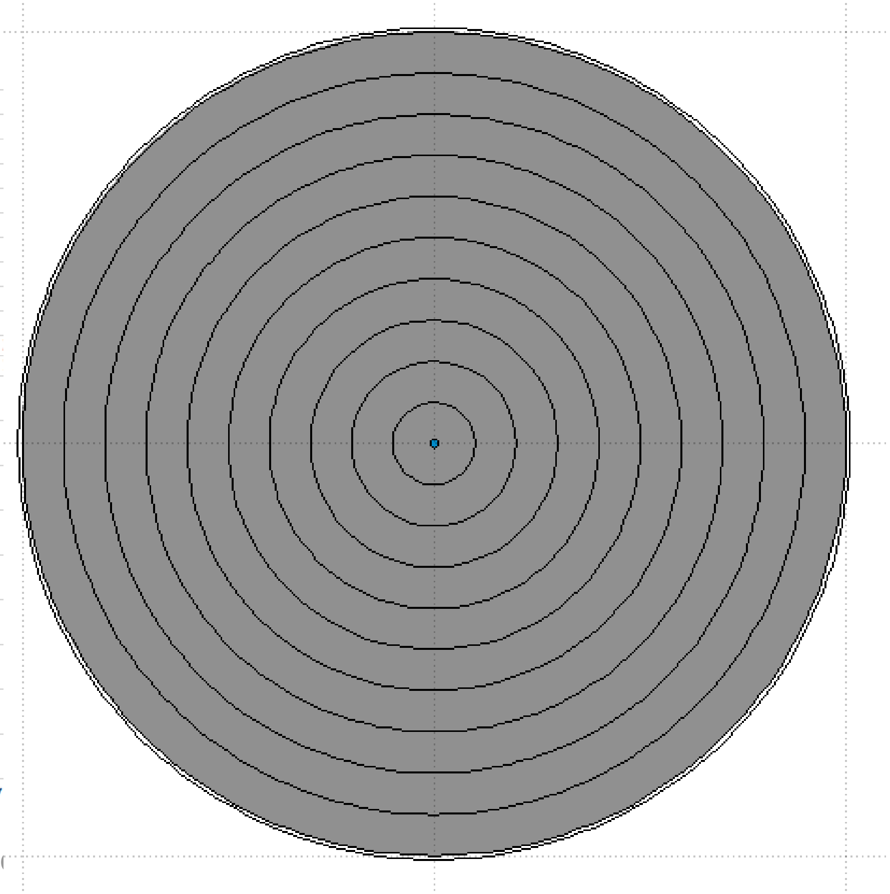


HF RFQ Shielding

- Onion like, spherical geometry with #defined material per 10cm layer
- Fitting function: Minimum dose on an external thin layer
- Materials considered: PE, B-PE, Fe, Cu, W, Concrete,…

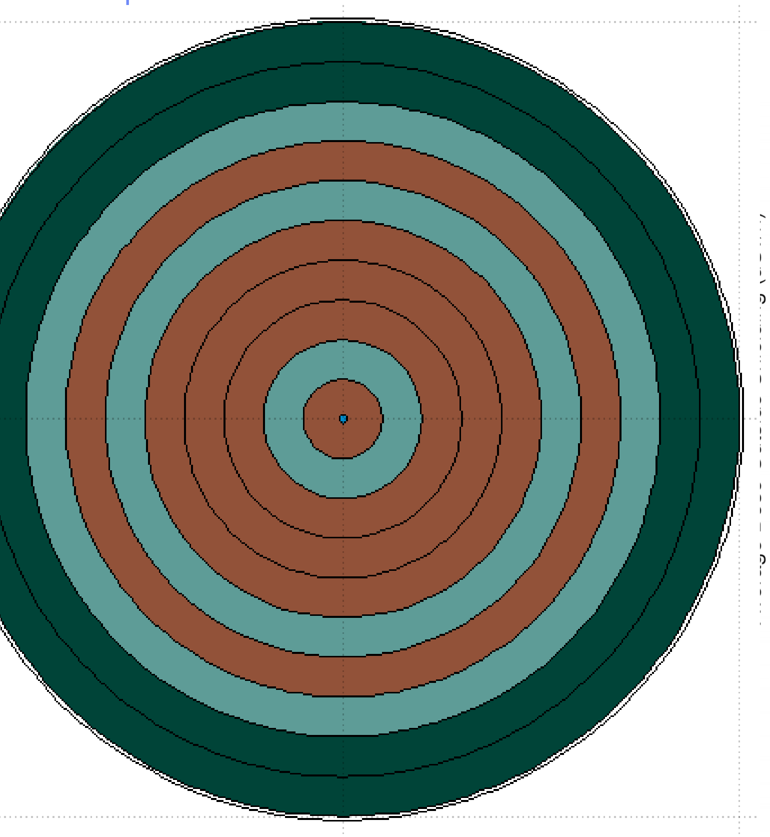


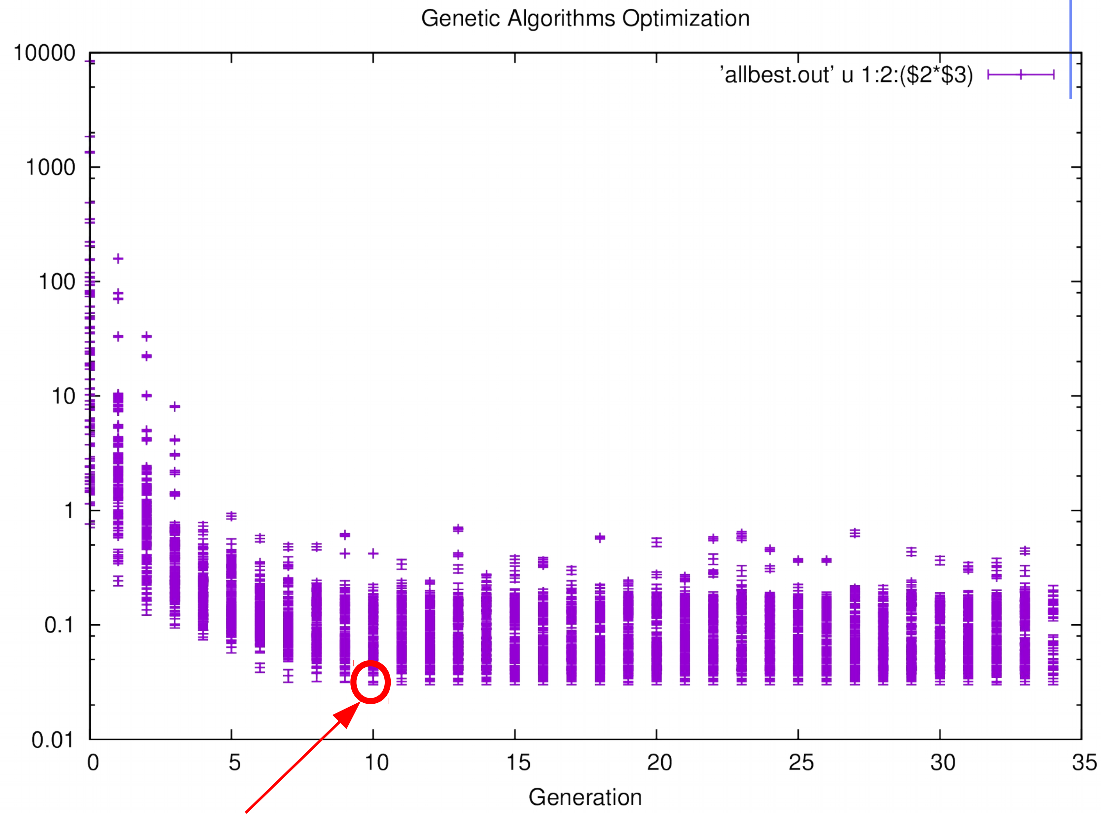
```
#define    BIAS     :3
#define    MAT1     :BERYLLIU
#define    MAT2     :POLYETHY
#define    MAT3     :COPPER
#define    MAT4     :COPPER
#define    MAT5     :COPPER
#define    MAT6     :POLYETHY
#define    MAT7     :COPPER
#define    MAT8     :POLYETHY
#define    MAT9     :POLY-BOR
#define    MAT10    :POLYETHY
                                          ----------- TITLE ... GEOEND :
                                          ----------- MATERIAL ... #elif :
  ASSIGNMA           Mat: =MAT1 ▾                  Reg: SHIELD1 ▾
             Mat(Decay): ▾                         Step:
  ASSIGNMA           Mat: =MAT2 ▾                  Reg: SHIELD2 ▾
             Mat(Decay): ▾                         Step:
  ASSIGNMA           Mat: =MAT3 ▾                  Reg: SHIELD3 ▾
             Mat(Decay): ▾                         Step:
  ASSIGNMA           Mat: =MAT4 ▾                  Reg: SHIELD4 ▾
             Mat(Decay): ▾                         Step:
  ASSIGNMA           Mat: =MAT5 ▾                  Reg: SHIELD5 ▾
             Mat(Decay): ▾                         Step:
  ASSIGNMA           Mat: =MAT6 ▾                  Reg: SHIELD6 ▾
             Mat(Decay): ▾                         Step:
  ASSIGNMA           Mat: =MAT7 ▾                  Reg: SHIELD7 ▾
             Mat(Decay): ▾                         Step:
  ASSIGNMA           Mat: =MAT8 ▾                  Reg: SHIELD8 ▾
             Mat(Decay): ▾                         Step:
  ASSIGNMA           Mat: =MAT9 ▾                  Reg: SHIELD9 ▾
             Mat(Decay): ▾                         Step:
  ASSIGNMA           Mat: =MAT10 ▾                 Reg: SHIELD10 ▾
             Mat(Decay): ▾                         Step:
```



HF RFQ: Results

- Running 40 generations of 60 individuals = 2400 FLUKA runs





Fittest individual
from 10th generation

Ongoing

- Medical RTPLAN, RTSTRUCT…
- GA integration in the interface
- MatPlotLib integration
- Geometrical optimization of CSG using multiple 2D slices
- Input validator (to remove completely the auto arranging of cards during saving)
- Fitting CAD geometries with quadratic primitives
- Spline addition as flair object, for calculating doses along a path or movie creation
- Plotting of USRDUMP in geometry editor
- Plots graph wizard

Ongoing

- Medical RTPLAN, RTSTRUCT…
- GA dialog integration
- MatPlotLib integration
- Geometrical optimization of CSG using multiple 2D slices
- Input validator (to remove completely the auto arranging of cards during saving)
- Fitting CAD geometries with quadratic primitives
- Spline addition as flair object, for calculating doses along a path or movie creation
- Plotting of USRDUMP in geometry editor
- Plots graph wizard