Visualization of Rocket Thruster Models and Experiments
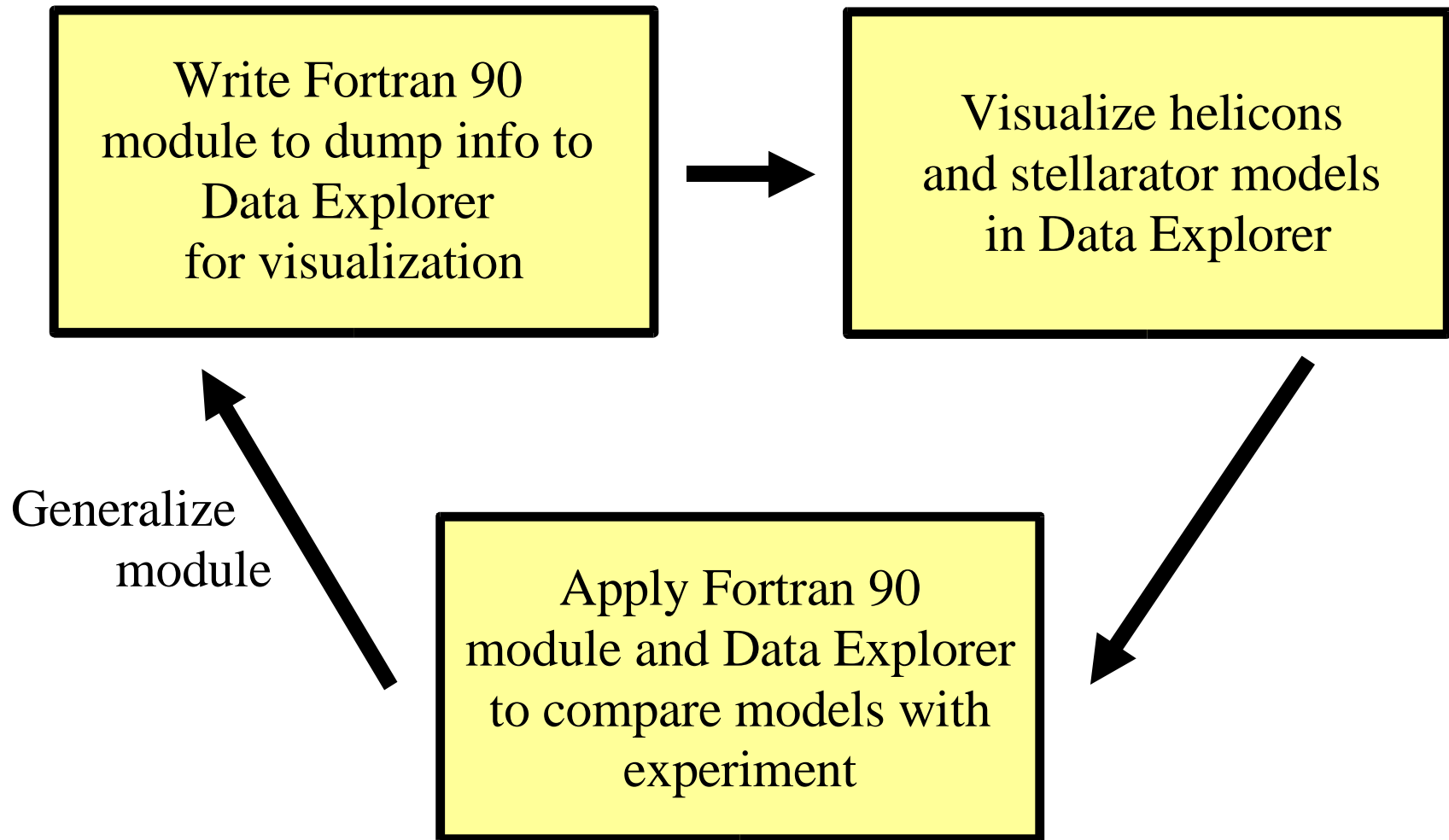
Kate White
University of Tennessee, Knoxville

Mentor: Mark D. Carter
Fusion Energy Division
Oak Ridge National Laboratory

Outline

- Summer ERULF project goals

- Intro to VASIMR

- Examples of images created with Data Explorer

- Tools:

  - Emir and AORSA codes and
    Fortran 90 Module dxdump.f

  - IBM Data Explorer

- Future work with module and Data Explorer

Goals for Summer ERULF Project

Write Fortran 90 module to dump info to Data Explorer for visualization

Visualize helicons and stellarator models in Data Explorer

Apply Fortran 90 module and Data Explorer to compare models with experiment

Generalize module

VASIMR: Variable Specific Impulse Magnetoplasma Rocket

The VASIMR rocket uses a plasma propulsion system allowing it to travel a greater distance in a shorter amount of time. The use of hydrogen gas as propellent greatly reduces the payload. Shorter flight duration is a necessity for astronauts as the human body cannot withstand long periods within the vacuum of space.

To produce thrust, hydrogen gas is injected into the system. Then a helicon antenna ionizes the gas producing plasma. The plasma is contained within an magnetic field produced by superconducting electromagnets. The plasma is then shot out to the end of the rocket producing a forward thrust.

The main experiment of the VASIMR concept, VX−10, is located in Houston, Texas at NASA's Advanced Space Propulsion Laboratory. The sister experiment at ORNL, mini−RFTF, is located in the Fusion Energy building at the Y−12 plant.

Visualizations of the VASIMR concept were created using the outputs of the EMIR codes.
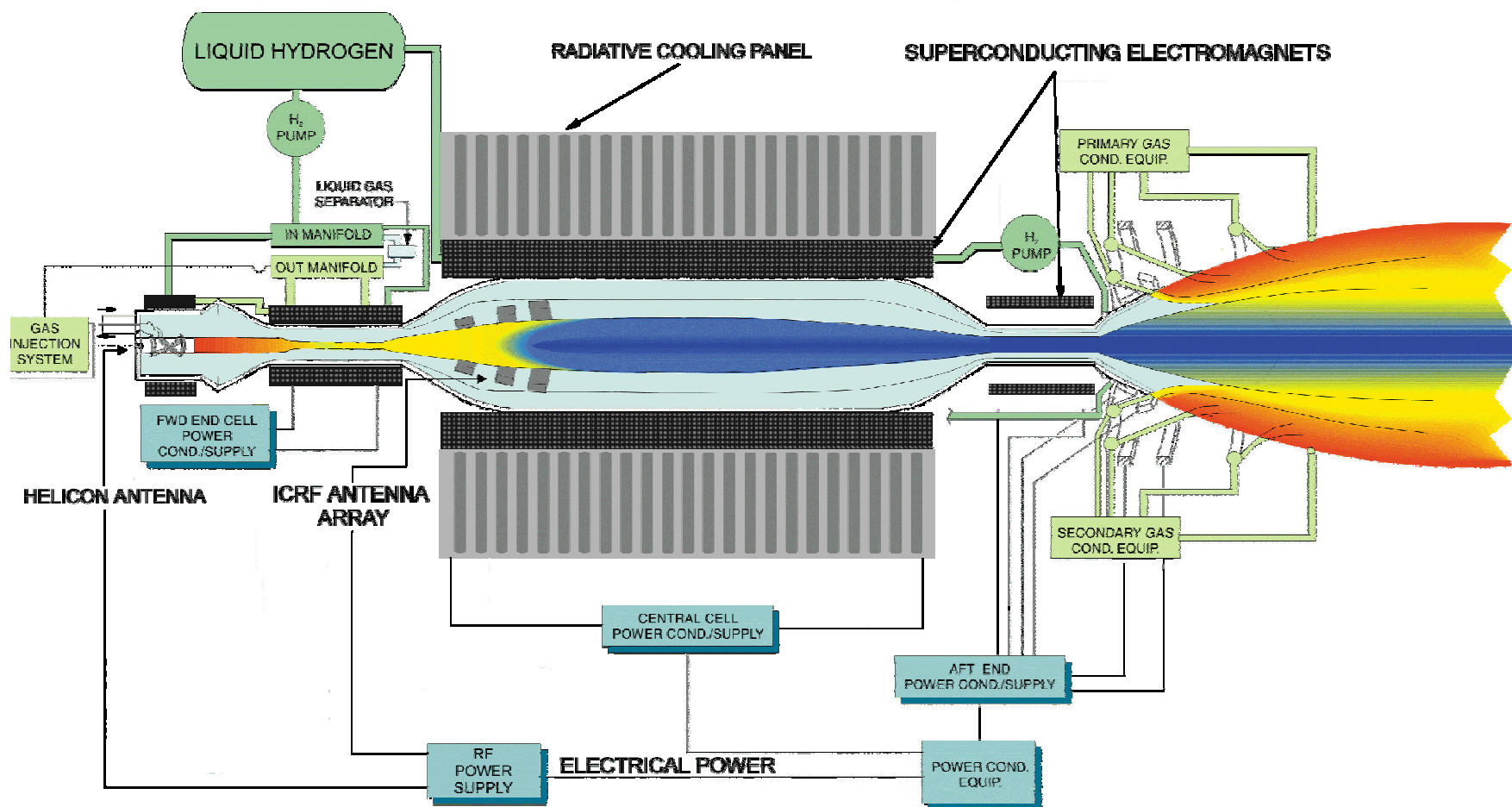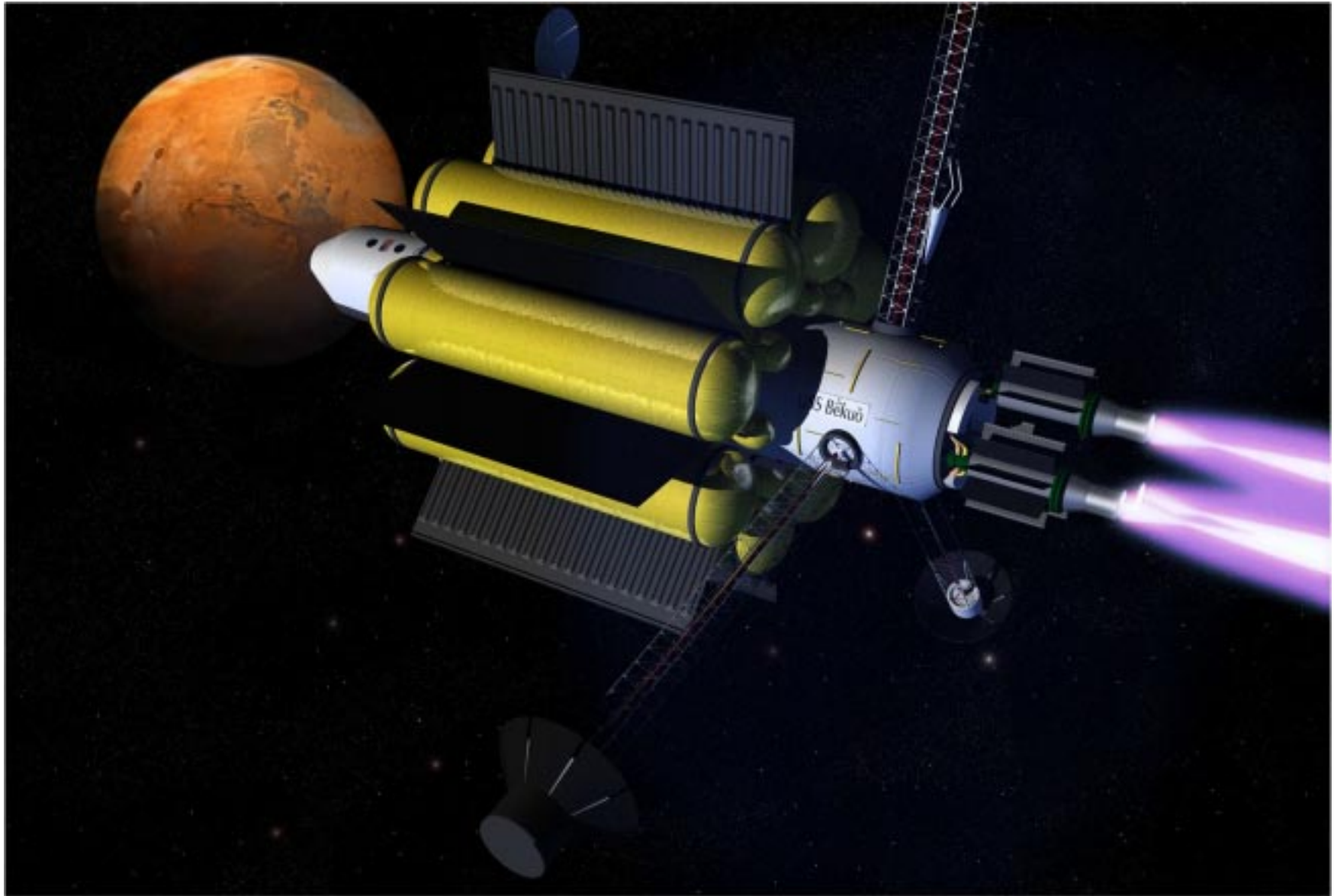
VASIMR SYSTEM

# VASIMR concept of a manned Mars mission

Mini–RFTF model of VASIMR concept

The mini–RFTF experiment at Oak Ridge has one long solenoidal coil around the helicon antenna, and a smaller mirror coil around the ICRF antenna.

The use of the opacity variable in DX is shown in this picture. By setting the opacity of the orange colored coils to .21, one can see through the coils to the antenna and flux surfaces inside the rocket.
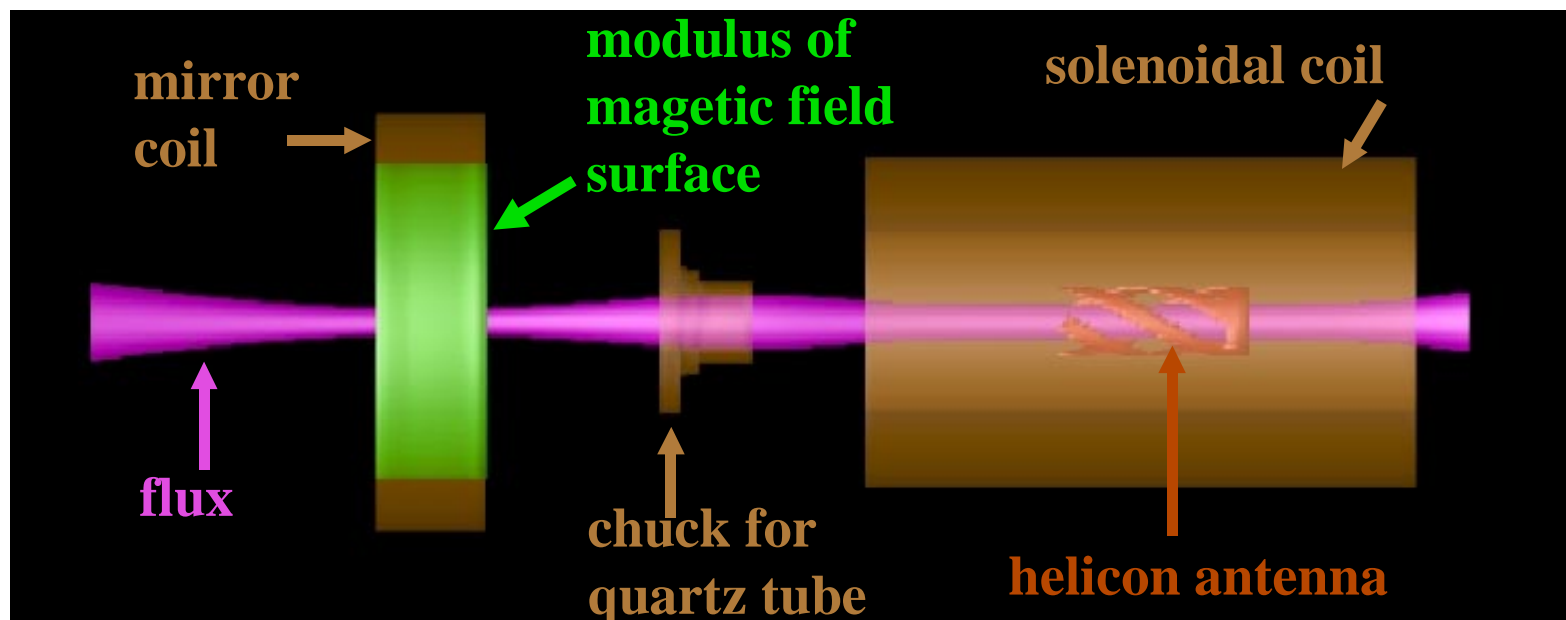
ASPL VX–10 experiment model
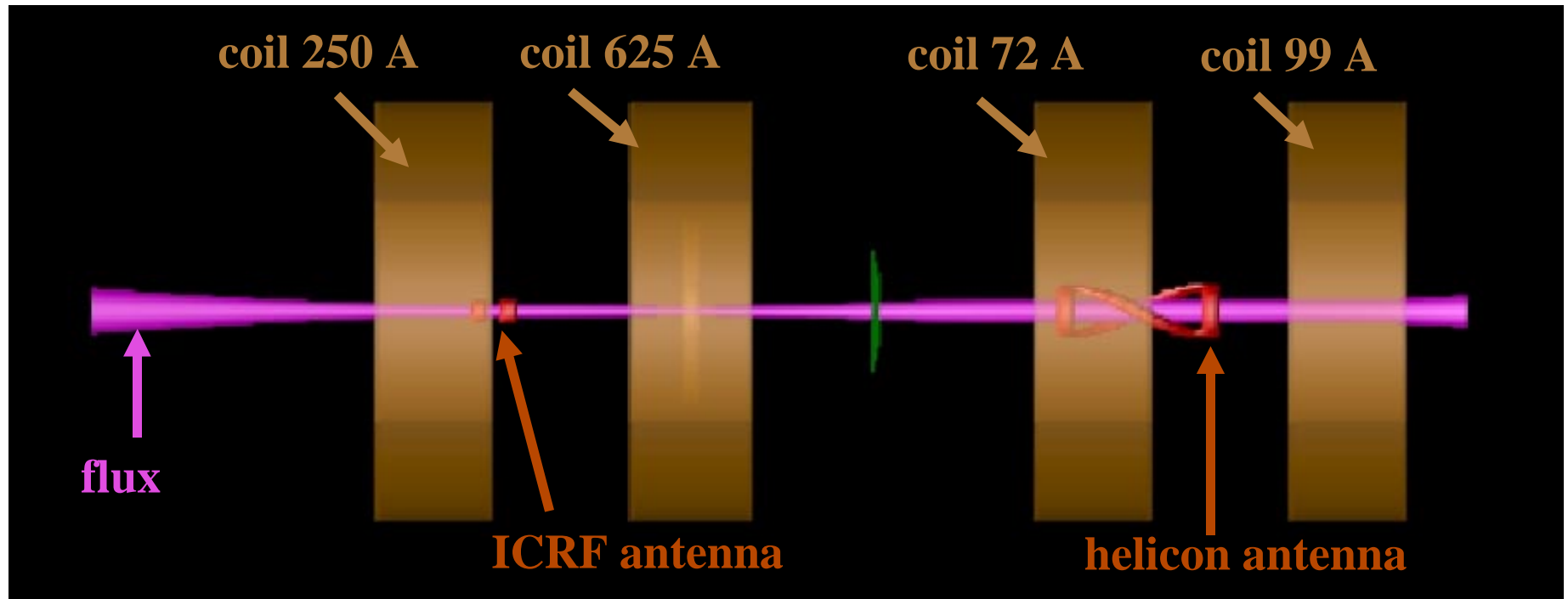
The ASPL experiment has four coils. The helicon antenna is located between the two coils on the right and the ICRF antenna is located between the two coils on the left. The chuck for the quartz tube is located inside the third coil from the right.

Mini–RFTF continued

This is an isosurface (a 3–dimensional contour plot) of the magnitude of the electric field in mini–RFTF.
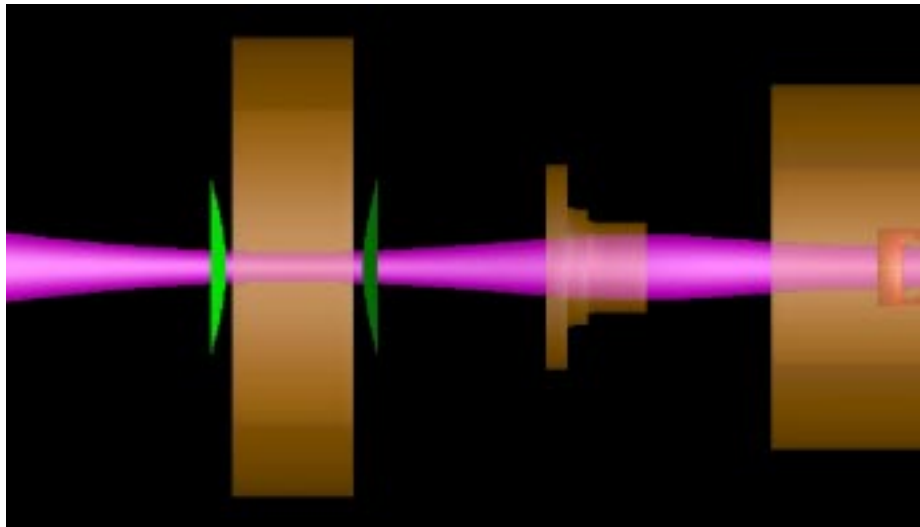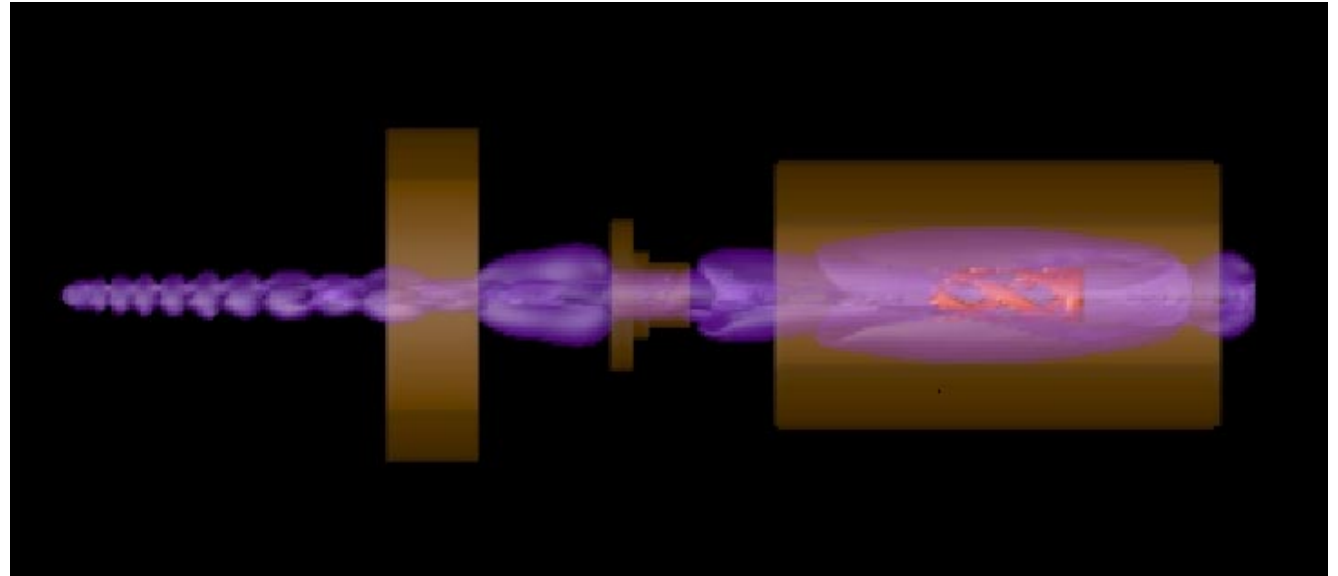

This is a different data set for the mini–RFTF model. The zoom function of DX has been used to allow a closer look at the last flux surface that will fit through the chuck for the quartz tube.

AORSA model for stellarator shows power deposition

The Fortran module and DX software were also applied to stellarator models using the outputs of the AORSA code.

The power deposition slab can be animated and an mpeg movie can be created using DX and ImageMagick.
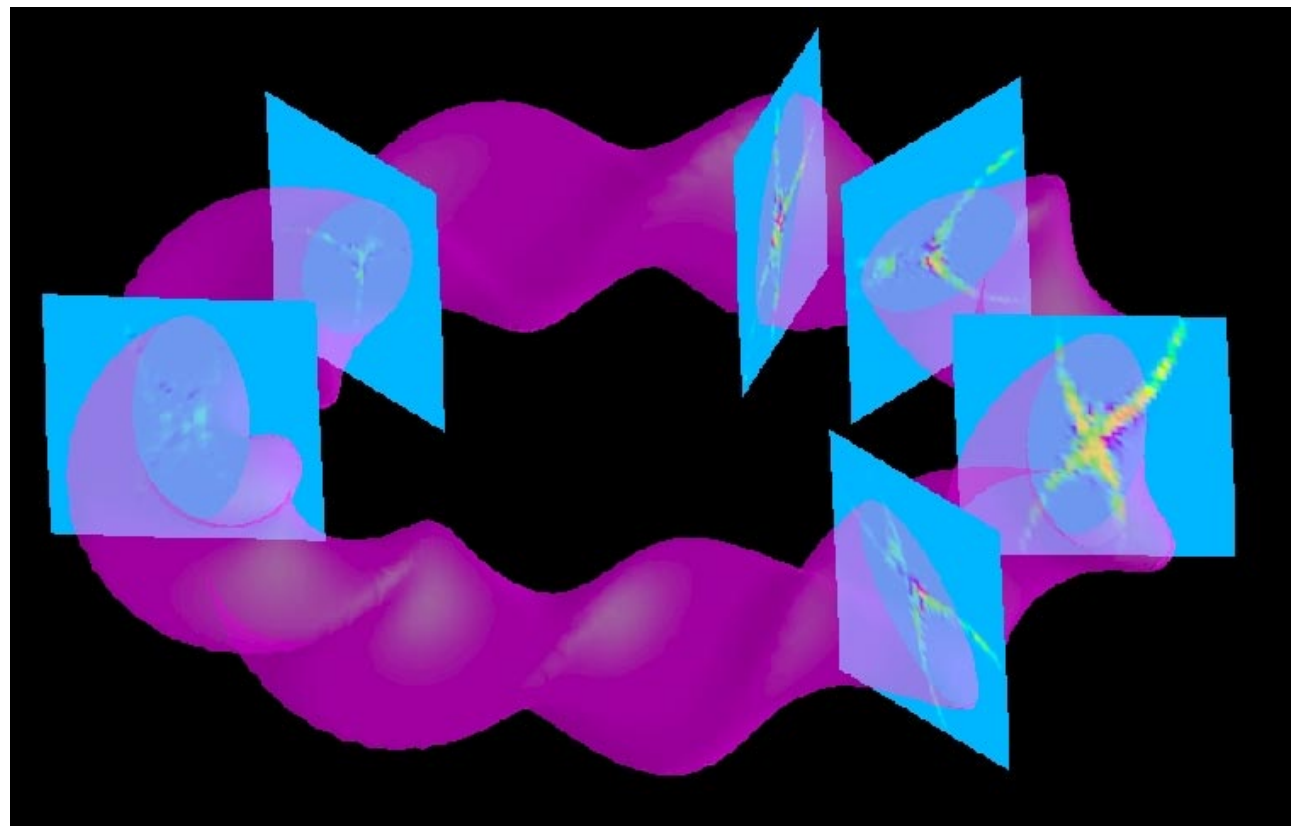
EMIR and AORSA codes

The EMIR and AORSA codes are fortran codes written to model plasma experiments. The EMIR codes solve Maxwell's equations for linearized cold plasma in a rocket geometry, and the AORSA codes solve Maxwell's equations for hot plasma in a stellarator geometry. To produce a set of model data, the user must input the number of amps running through the coils, the phase, the loading on the helicon antenna, and many other variables. The codes then solve the equations:

$$\nabla \times \vec{E} = i\omega\vec{B} \quad ; \quad \nabla \times \vec{B} = \mu_0 \vec{J}_{ext} - i\omega\epsilon_0\mu_0 \widetilde{K} \cdot \vec{E}$$

$\vec{E}$ and $\vec{B}$ represent the electric and magnetic fields respectively. $\widetilde{K}$ is the dielectric tensor operator. $\vec{J}_{ext}$ is the external antenna current. ϵ_0 and μ_0 are the permittivity and permeability of free space respectively. ω is the RF frequency in radians per second, and i is the square root of negative one.

How do the outputs of EMIR and AORSA get visualized in DX?

The main purpose of this internship was to write a Fortran 90 module to output native Data Explorer files so that the EMIR outputs could be visualized in Data Explorer. The module was written in Fortran 90 because both the EMIR and the AORSA codes were written in Fortran. The Fortran 90 module dxdump.f calls a subroutine to write a dx file which contains references to a data file and positions file. The user need only choose between calling the function dxPolar or dxCartesian because the eight subroutines have been overloaded.

dxCartesian	1–dimensional	3–dimensional
single precision	dxCartesian_1d_sp	dxCartesian_3d_sp
double precision	dxCartesian_1d_dp	dxCartesian_3d_dp

dxPolar	1–dimensional	3–dimensional
single precision	dxPolar_1d_sp	dxPolar_3d_sp
double precision	dxPolar_1d_dp	dxPolar_3d_dp

The Beauty of the Module:
add 2–3 lines of code and POOF!

```
program flux

use dxdump
integer, parameter :: nx=30, ny=40, nz=50
real :: x(nx), y(ny), z(nz)
real :: vx(nx,ny,nz), vy(nx,ny,nz), vz(nx,ny,nz)
real*8 :: flux(nx,ny,nz)
...
...
call dxCartesian(vx, vy, vz, x, y, z, 'test')
call dxCartesian(flux, x, y, z, 'flux_test')
...
...
end
```

The program flux calls the function dxCartesian twice. In the first call, because the arguments are single precision vectors, the subroutine dxCartesian_3d_sp will execute. In the second call dxCartesian_1d_dp will execute.

The subroutines will output the files test.dx, test.dat, test.pos, flux_test.dx, flux_test.dat, and flux_test.pos. The *.dx files are the native DX format files which are read in during import. These *.dx files call the *.pos files to read in positions, and the *.dat files to get the data values at those positions.

What does a Data Explorer Network do?

IMPORT DATA

The import module reads in data in native DX format. The native DX format contains four objects: positions, connections (regular or irregular), data, and the class field connecting the previous three objects.

REALIZE DATA

The type of realization module used depends on the type of data the user wants to visualize. For two dimensional data the construct module can be used to create a plot. For three dimensional scalar data, an isosurface may be used. Vector data may be realized with glyphs. There are many options.

RENDER DATA

Rendering allows the user to see the image displayed to the screen. This may be done simply with an image module, or the user may specify specifics with a camera module and render module.
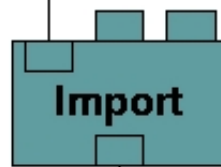
FileSelector

Import

Scalar

Isosurface

Image

Color
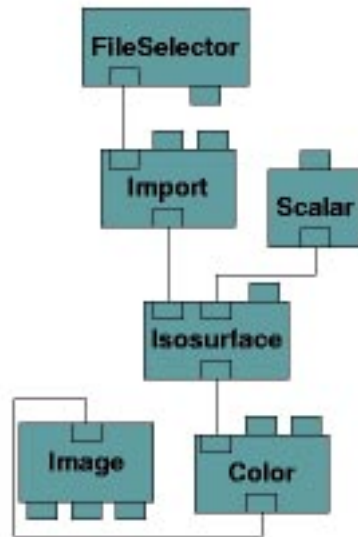
Simple DX Network

File Selector
opens a browser for
the user to select file
to import

Import
imports the file selected
in above file selector

Image
renders the isosurface
with a camera and
displays it to the
screen

Scalar
allows user to specify the
value at which to realize the
isosurface
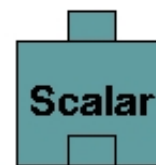
Isosurface
creates a surface at all
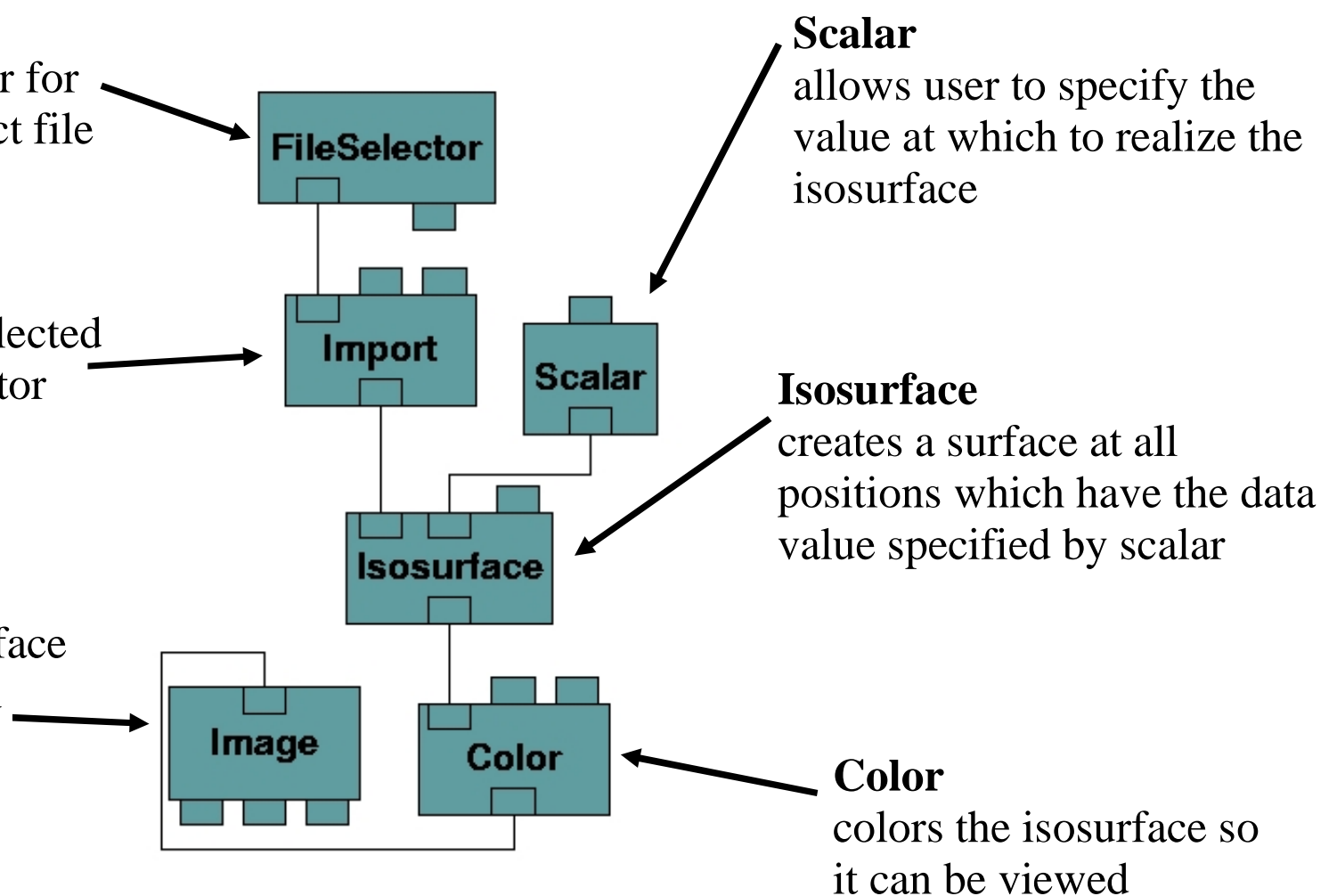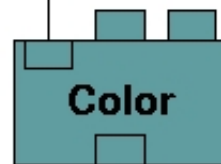positions which have the data
value specified by scalar
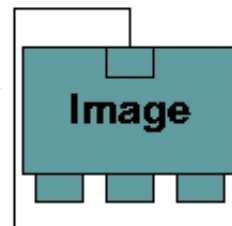
Color
colors the isosurface so
it can be viewed

FileSelector

Import

Scalar

Isosurface

Image

Color

IBM Data Explorer has advantages over other visualization packages "shameless plug"

- **FREE!!!** Data Explorer is open source code.
  - download at http://www.opendx.org
- requires **inexpensive** hardware to run DX
- data manipulation within the network
- rotation and navigation in view mode
- can manipulate opacity of objects
- can create animations
- networking capabilities

Future use of the Fortran 90 module and Data Explorer

- documentation has been produced explaining the function and application of the subroutines and the module dxdump.f, making DX more user friendly for "newbies"

- after extensive testing has been completed, the module may be available for public use

- a set of general use networks for DX has been created and is available for use within the Fusion Energy Division

- additional documentation of the DX networks can be created

Acknowledgements