



**SECAD Phase II Overview**  
**and VizSchema (if time permits**  
**and is interesting)**

Svetlana Shasharina



# Logistics of SECAD

- Phase 2
  - SECAD (a Schema-based Environment for Configuring, Analyzing and Documenting Integrated Fusion Simulations)
  - Starts now for 2 years
  - Sveta, Marc Durant and Dave Alexander
- Collaborators
  - SWIM
  - FACETS
  - Tech-X
  - Paratools
- Goal is to
  - Assist SWIM and FACETS
  - Merge towards FSP?

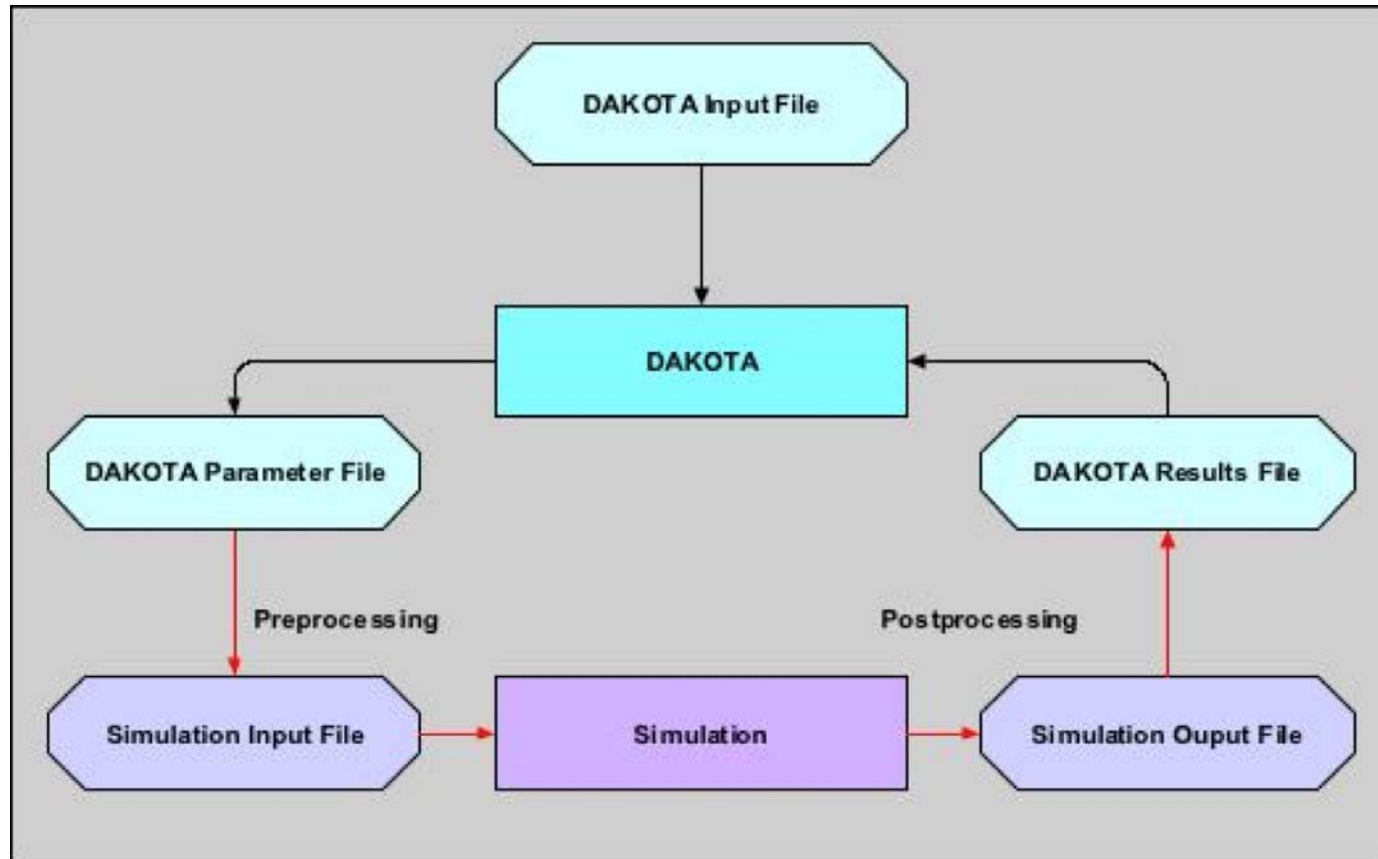


## Official Tasks of Phase 2

- Parameter and Optimization Studies for IPS and FACETS using DAKOTA
- GUIs (David Alexander will talk about this)
  - FACETS
  - IPS
  - DAKOTA
- Performance Instrumentation (Paratools)



# DAKOTA Uses its data files: need translation from parameter to IPS configuration file and from resulting file (NetCDF) to results file





# In Dakota Configuration file, one specifies the type of study, what it needs, and the driver (ips\_wrapper.sh)

```
strategy,  
  single_method  
  tabular_graphics_data  
method,  
  vector_parameter_study  
  final_point = 4  
  num_steps = 3  
variables,  
  continuous_design = 1  
  initial_point  1  
  descriptors    'TiO'  
interface,  
  system  
  analysis_driver = './ips_wrapper.sh'  
  parameters_file = 'params.in'  
  results_file = 'results.out'  
  file_tag  
  aprepro  
responses,  
  num_objective_functions = 1  
  no_gradients  
  no_hessians
```



# Phase I IPS+DAKOTA

- Parameter study using model components in a model simulation created by Don Batchelor
  - dakota\_wrapper.sh:
    - With two arguments (DAKOTA and IPS configuration files)
    - Runs once
    - Parses IPS configuration files and extracts and exports env vars
    - Parses DAKOTA, finds the variable name and values, creates working directory for each value, and runs DAKOTA using ips\_wrapper.sh as a driver
  - ips\_wrapper.sh
    - iCreates initial files for each value of parameter (configuration and NAMELIST)
    - Runs ips
    - Returns results from namelist file into DAKOTA results file (needed for optimization, not for parameter studies, though)
  - Helping Python scripts for parsing and substituting values in files
  - Look it up in ips/utilities/dakota\_studies/parameter\_study and run it on franklin



# Particular things to do in task 1 for SWIM

- Identify a set of “typical” IPS simulations
- Figure out the initialization dependencies:
  - If I want to use X in an independent variable in a parameter or optimization study, which initial files should be modified (namelists, plasma state?) when X is varied?
  - If we want  $F(x_0, \dots, x_1)$  as an objective function, from which output files I get  $x_0$  etc to evaluate the function
- For each simulation, identify studies of interest (which parameters to vary, what to minimize, what study to perform)
- Create templates for DAKOTA files and helping scripts
  - Set up and export environment variables (DAKOTA\_ROOT, SIMULATION\_ROOT etc)
  - Create working directories
  - Preprocess DAKOTA input file and generate IPS configuration and consistent input files
  - Invoke the simulation (as many times as needed)
  - Postprocess IPS results and feed into DAKOTA (from, for example, NAMELIST or NetCDF)
- Learn how do parallel (do we run DAKOTA on front node which will launch qsubs as drivers?)



## Scenarios

- SWIM (TSC) simulation varying initial conditions and simulated plasma response for minimizing forces on vacuum vessel
  - Do we have TSC on franklin?
- FACETS core-edge simulation varying plasma shape and source deposition for maximizing fusion gain  $Q$ 
  - Need to implement the objective function ( $Q$ )
- Optimization of start-up and shutdown of ITER
  - Seems to be more advanced
- Do we have simple examples to start???



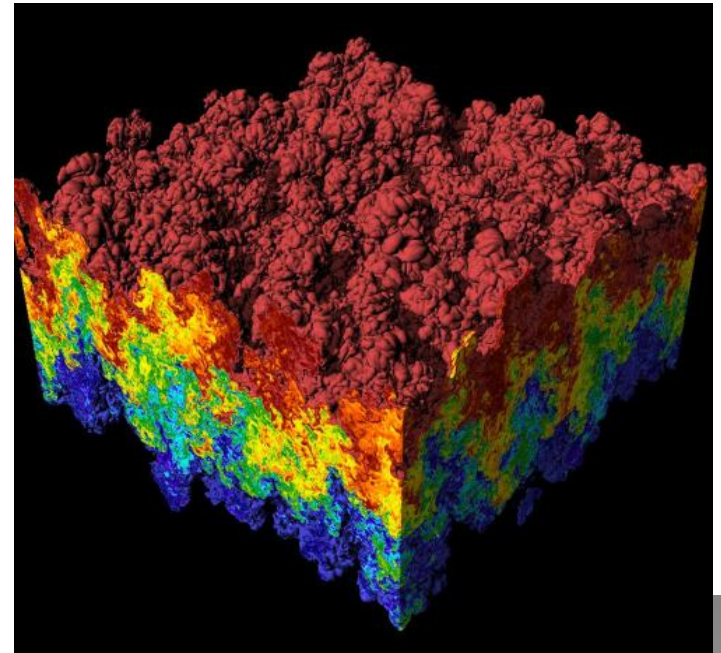
# Performance Instrumentation (Paratools)

- Components to instrument (?):
  - TSC
  - TORIC
  - CQL3D
  - IPS+DAKOTA
  - FACETS+DAKOTA
- Instrumentation means:
  - Performance measurement
  - Identification of bad load balancing, bottlenecks
  - Typically requires access to source code
- Can we give Paratools guys access to IPS and sources for these codes?
- So far Paratools instrumented:
  - NUBEAM (improved)
  - FACETS Core Solver: found load imbalance
  - FACETS with GYRO (topological settings for Intrepid)
  - UEDGE (nor finished)
  - AORSA?

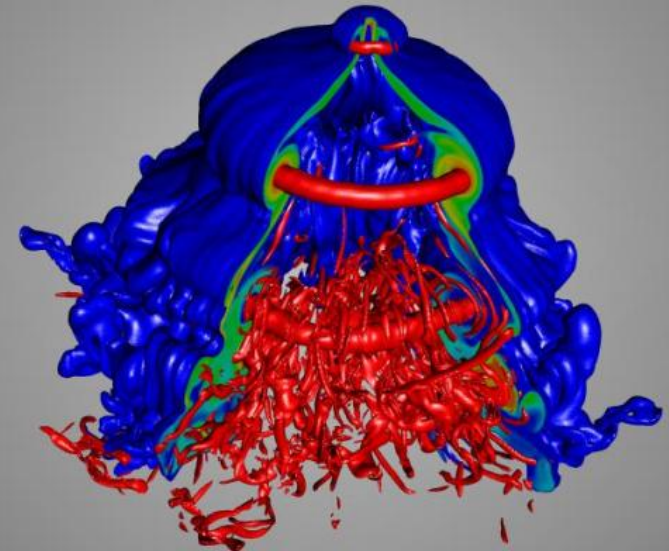


# Simulation Data and VizTools are Diverse

- Data
  - ASCII
  - All kinds of home grown binary format
  - Lately moving to self-described data formats
    - NetCDF
    - **HDF5 (our current choice)**
- VizTools
  - IDL
  - Matlab
  - AVS/Express
  - Lately moving to open standard tools
    - **VisIt (our current choice)**
    - ParaView



- VisIt – supported by VACET SciDAC
  - Free
  - Powerful
  - Great team to work with!
- Has internal data structures
- One needs to write a **reader** for each type of data format to transform into what VisIt expects





# One Cannot Write One Reader For All HDF5 data

- HDF5 consists of groups (like directories), datasets (like files – end leaves) and attributes (for small data and metadata)
- One can organize data in many ways using HDF5 constructs
- One could use any kinds of names and no metadata at all
- How one can understand what is what?
  - What is supposed to be visualized?
  - What order is used (row-major etc)?
  - Where is the mesh of the data?
  - What are the components of the mesh?



# VizSchema Is An Attempt to Standardize HDF5 for Viz

- Based on experience working with
  - VORPAL (uniform and unstructured meshes and particles)
  - NIMROD (structured meshes)
  - UEDGE (structured, multi-domain meshes)
  - TXFLUID (unstructured meshes)
  - FACETS (combination of the above)
- VizSchema
  - Data Model (agreement about data organization and metadata)
  - C++ reader (independent of Viz tool) of HDF5 data into in memory viz objects
  - VisIt plugin
- Funded by FSML grant (DOE SBIR, FES), FACETS grant (DOE SciDAC) and Tech-X Corporation

# VizSchema: Minimalistic but Enough for Viz

- Viz entities (based on our experience):
  - Variables (live on external mesh)
  - Variables with meshes (spatial info is mixed in)
  - Meshes
  - Derived variables
  - Multiple-domain Variables
- Metadata (internal to files vs XML, for example)
  - Identifying the entities and specifying their kinds (if any)
  - Providing information needed for minimal viz
- Principles
  - Metadata is minimal (and results of many friendly discussions)
  - Metadata is in attributes starting with “vs”
  - Groups and datasets names are not regulated



## How To Adopt VizSchema?

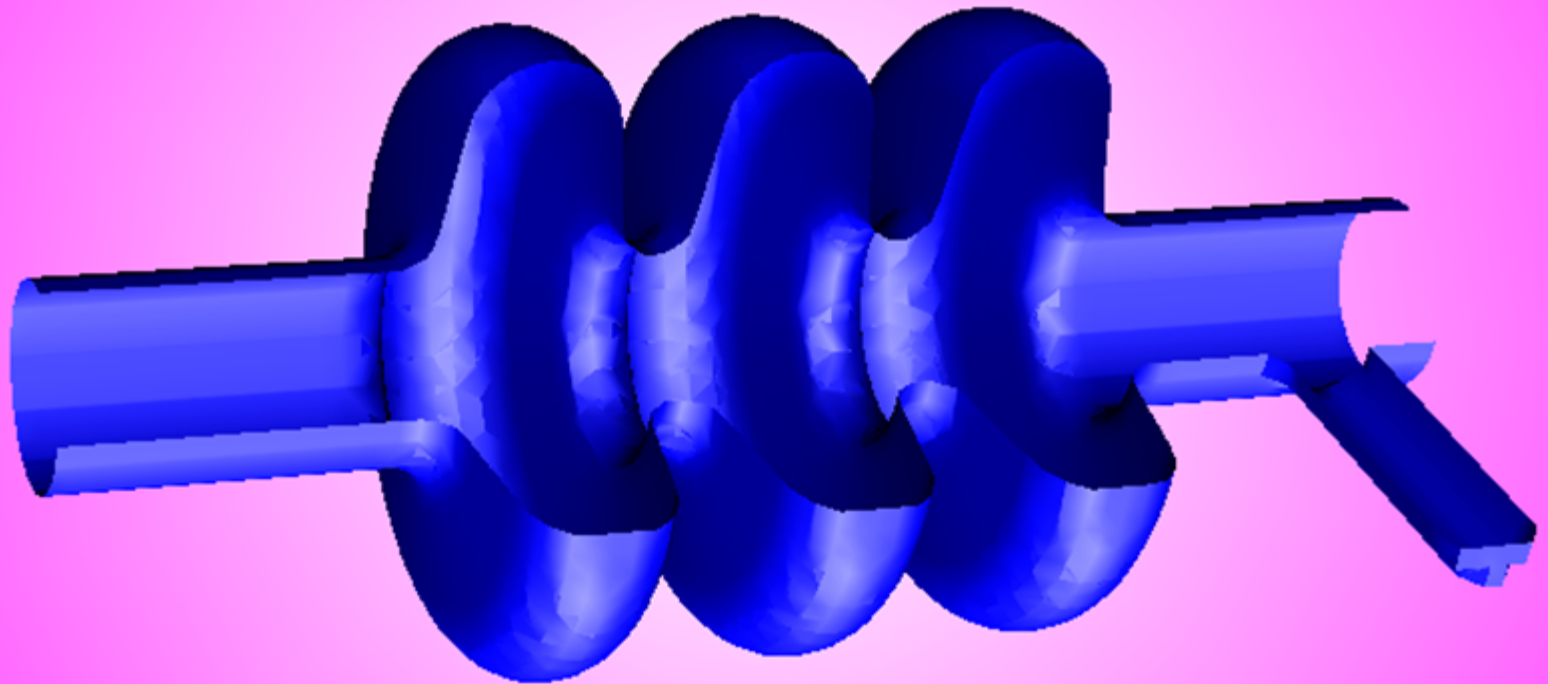
- Change your I/O
  - VORPAL
  - FACETS (Fusion SciDAC)
  - NIMROD (MHD)
  - PolySwift++ (nanotech)
- Or change your files using PyTables (very easy interface to modify and add attributes)

```
h5file = tables.openFile(fileName, mode='a')
dataSet = h5file.getNode("/") + dataSetName)
dataSet.attrs.vsType = "variable"
h5file.close()
```

  - Changing old VORPAL outputs to fit evolving schema
  - Changing SYNERGIA output
- First step is look at “h5dump –A” command and send to us...



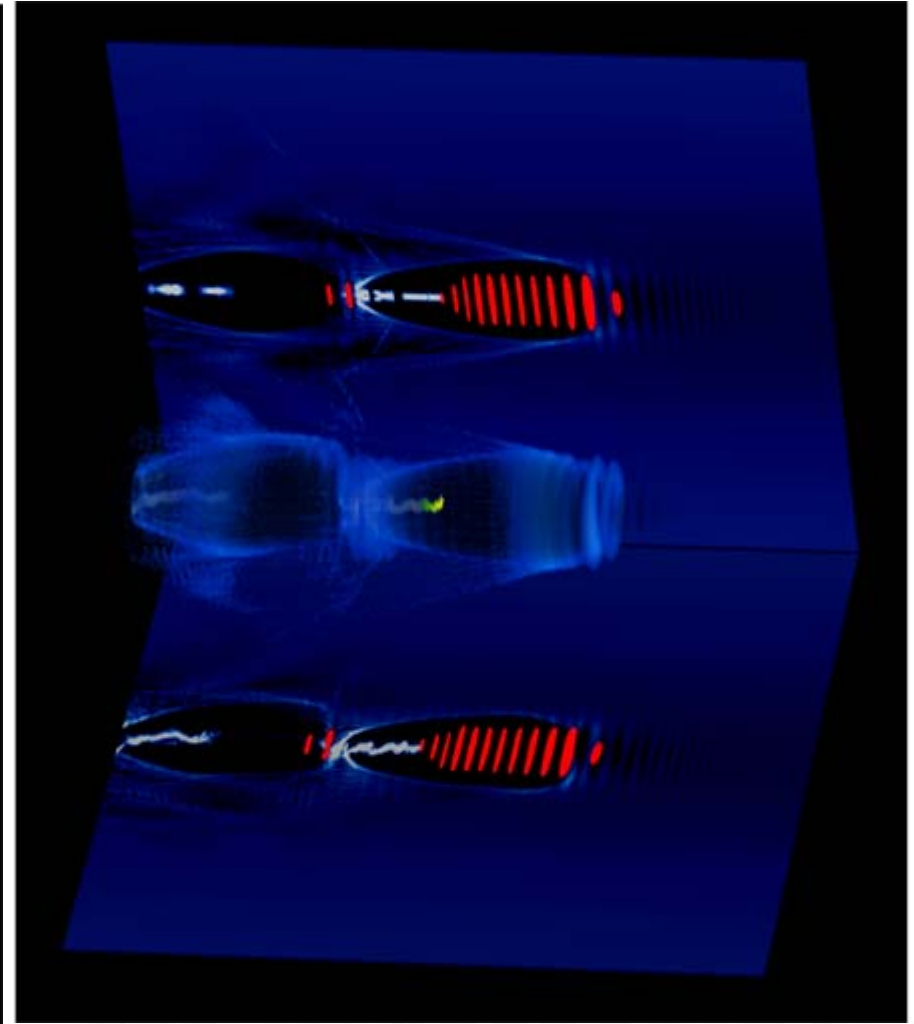
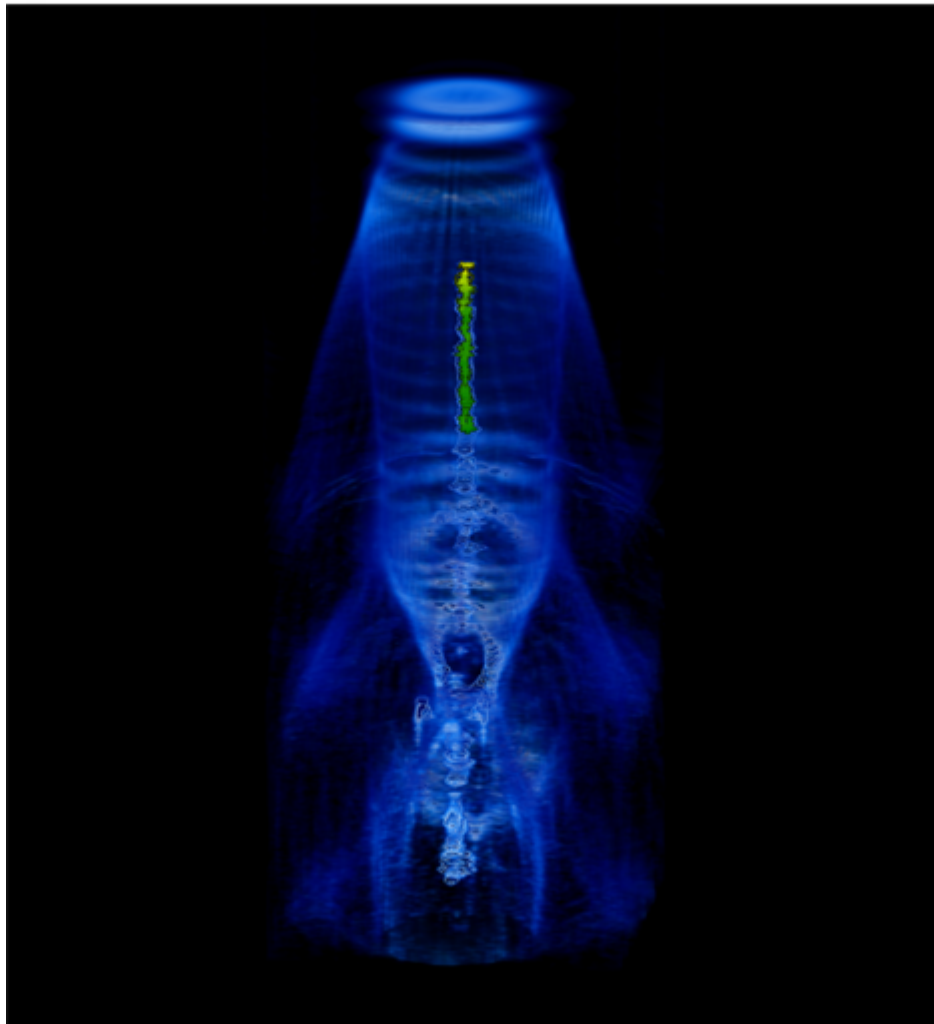
# RF Cavity Represented As Unstructured Mesh of Polygons (VORPAL)





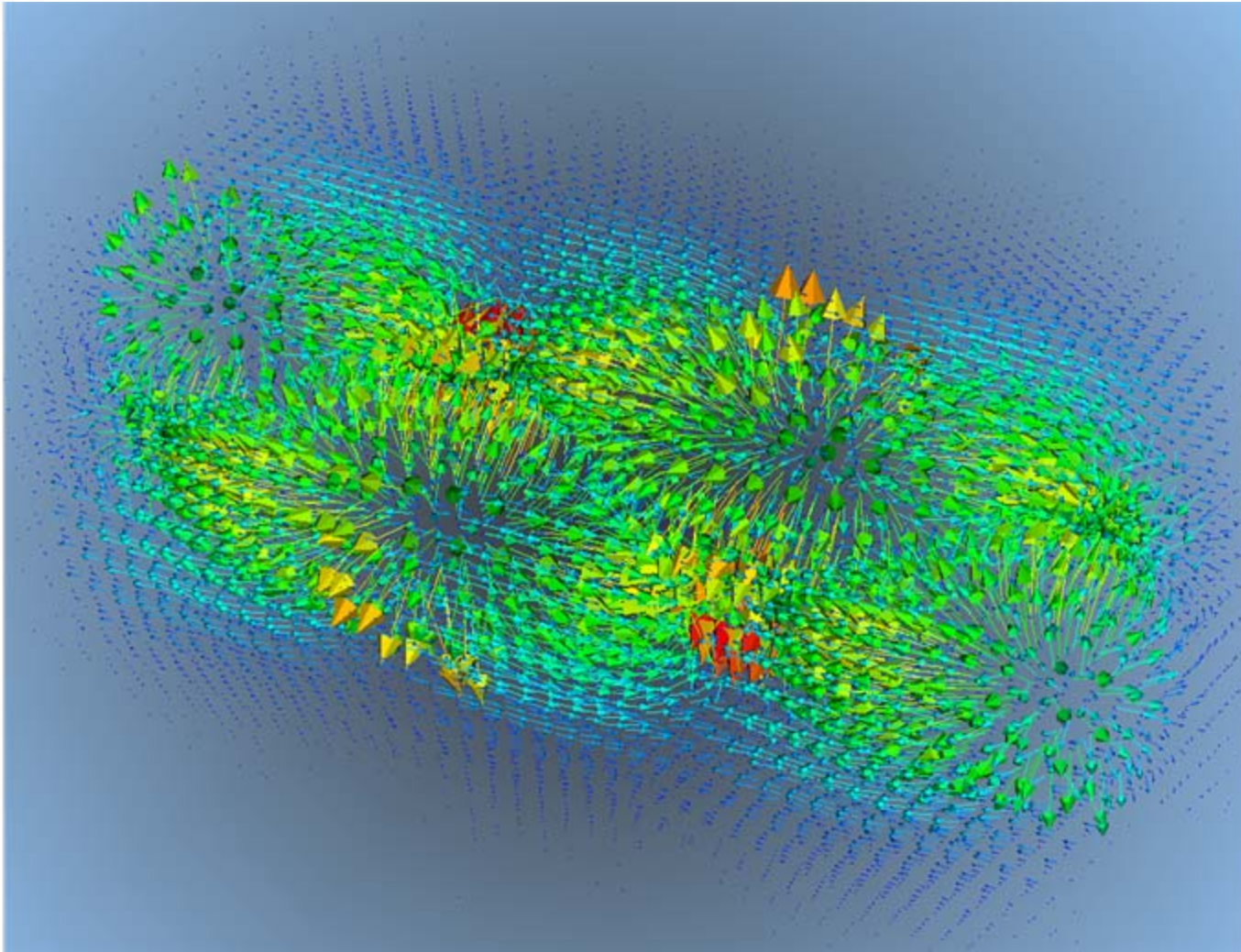
# Laser-Plasma Accelerator Simulation

(VORPAL). Left: plasma bubble and generated beam. Right: Laser field. Images - courtesy of C. Geddes and G. Webber.



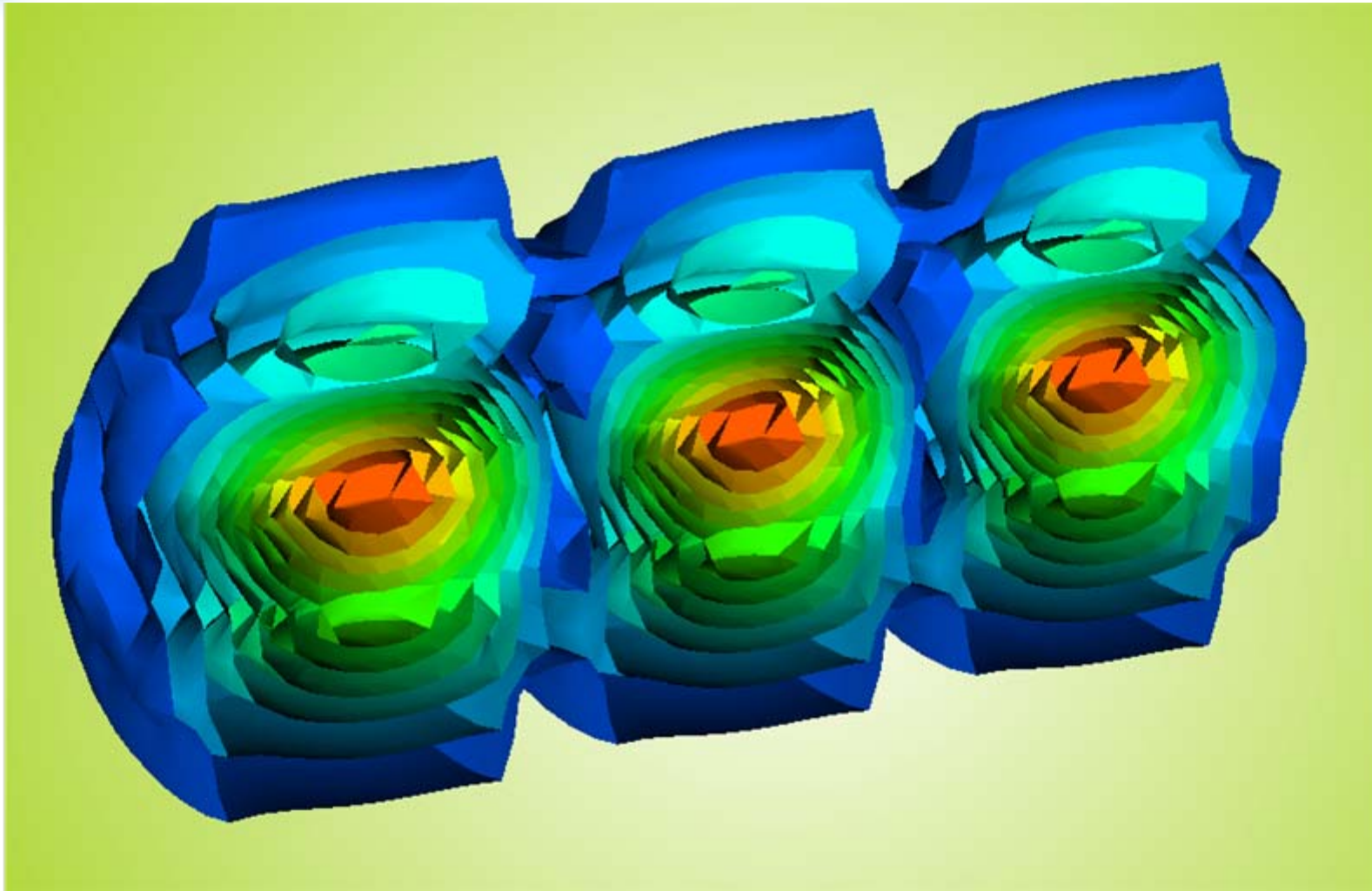


# Electric Field (Vector) in A 3 Cell Crab Cavity (VORPAL)



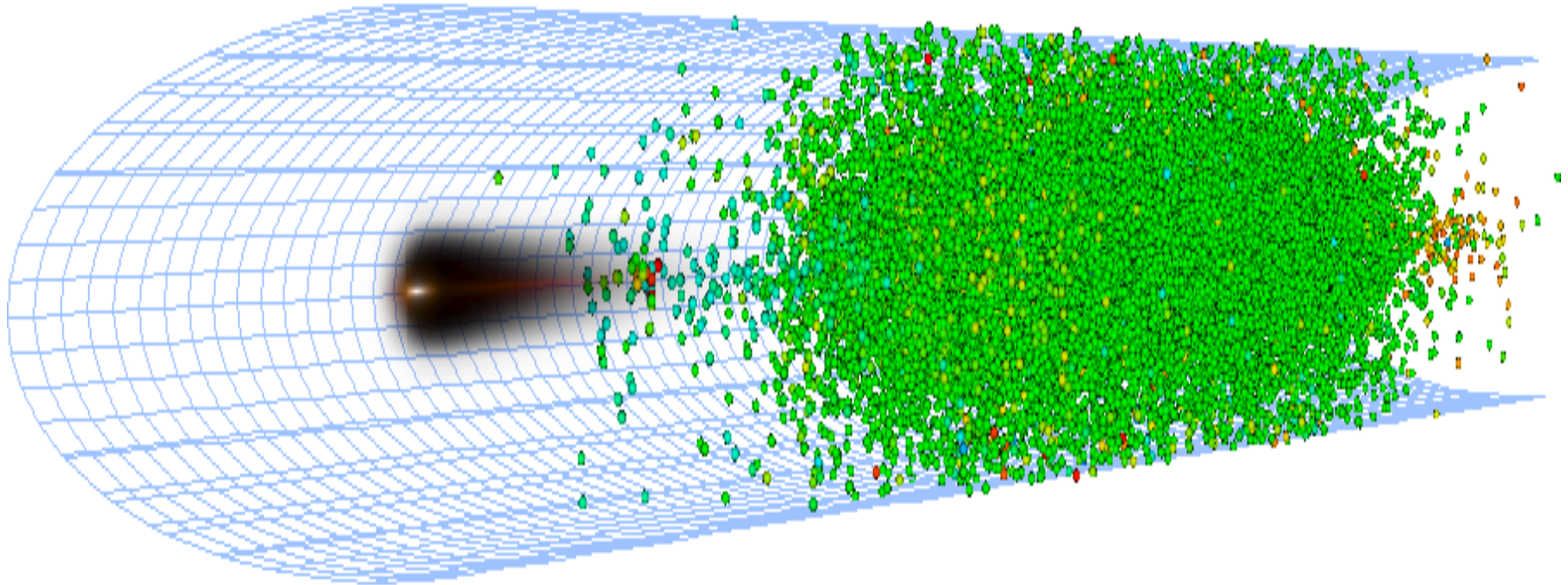


# Magnetic Field (isosurfaces) in A Cavity (VORPAL)

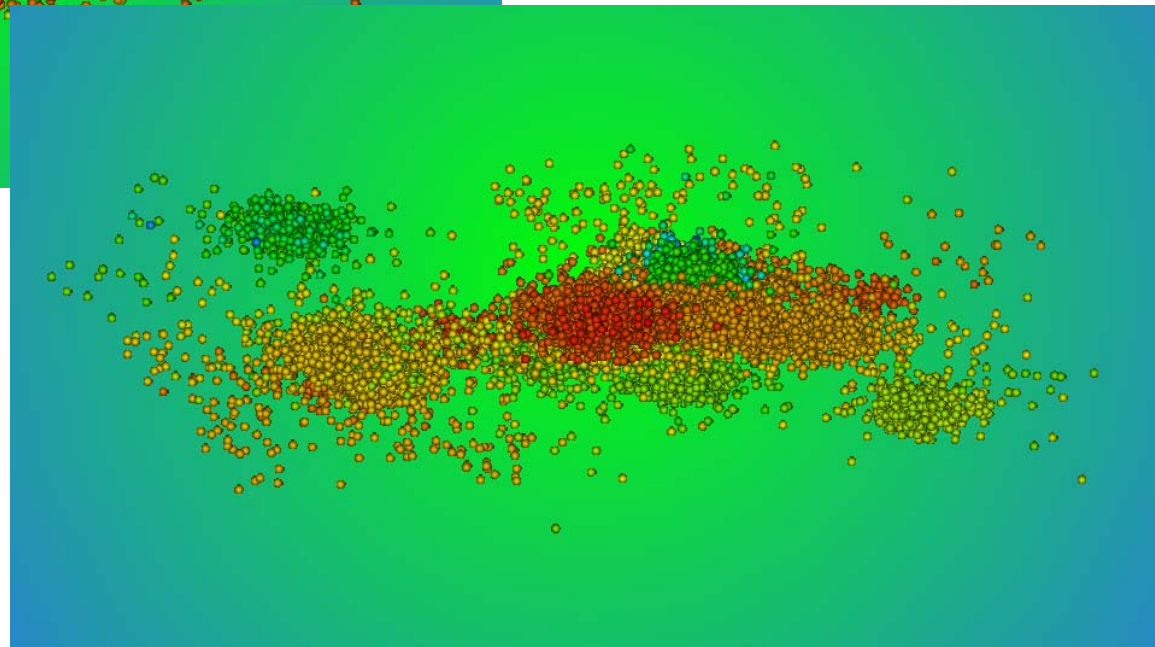
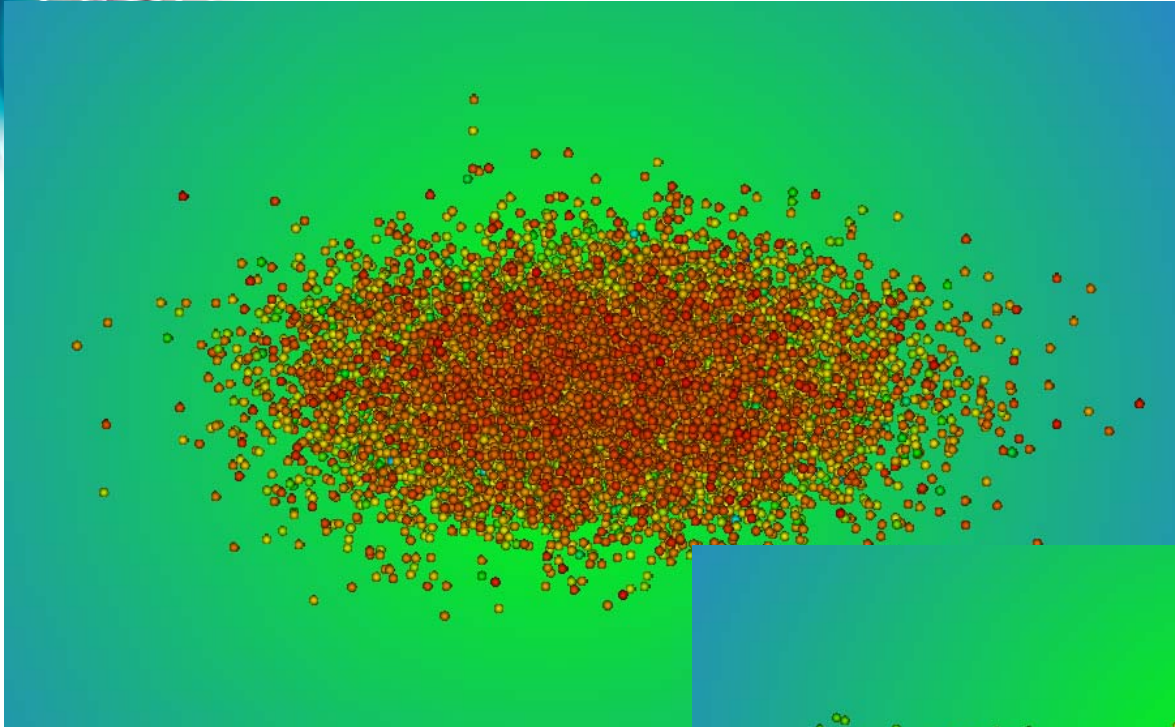




# Electron Cloud and Beam Current (VORPAL)

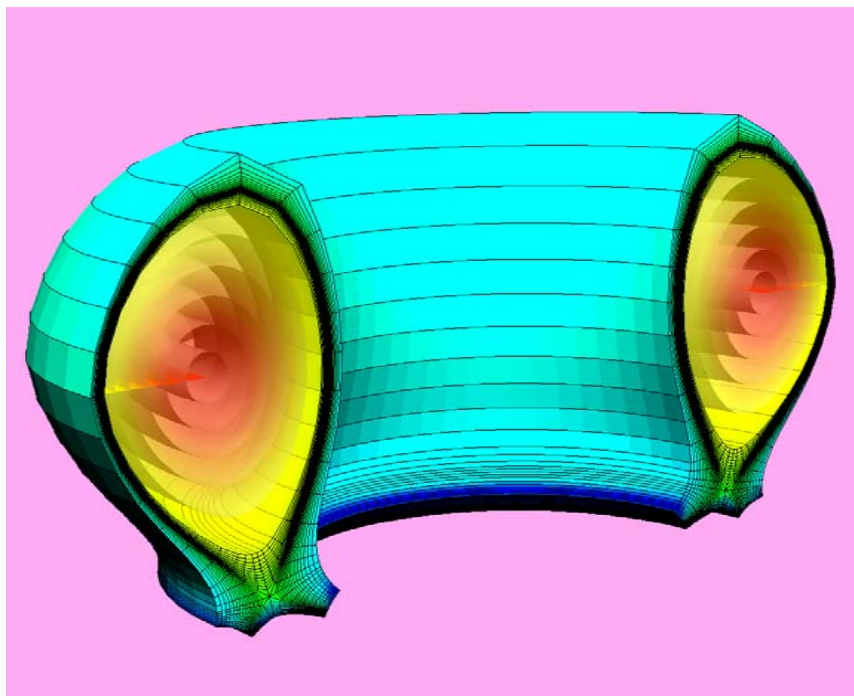


# Beam Particles ( $p^{**2}$ in SYNERGIA)

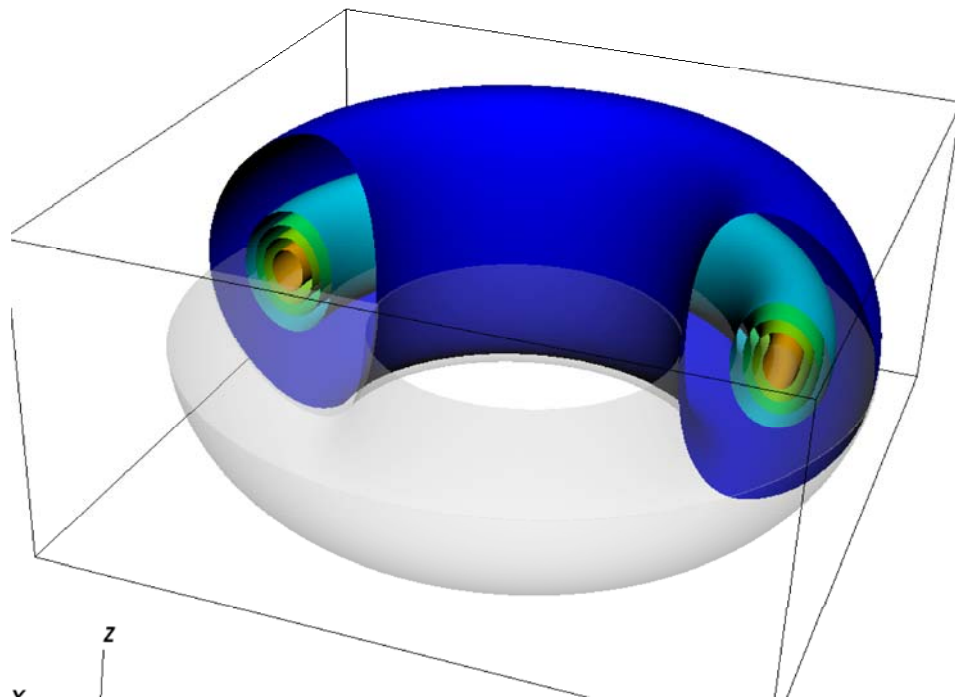




# Fuion Viz Requires Non-Trivial Knowledge of VisIt:



FACETS MD Viz



NIMROD Viz



## Information

- VizSchema information and download:  
<https://ice.txcorp.com/trac/vizschema/wiki/WikiStart>
- Soon will be come as one of standard VisIt plugins
- [sveta@txcorp.com](mailto:sveta@txcorp.com)



## Some ideas for discussion

- In general, it will be nice to understand the role of IPS and FACETS frameworks in FSP: what do they cover?
- Any other areas we could help with (new deadline for DOE/SBIR is Nov 20)
- SECAD could result in having a VisIt-based Portal for SWIM and FACETS if
  - We go with VisIt
  - Implement remote access
- Any comments on how (if at all) reshape SECAD?