

# TRANSP framework discussion

Johan Carlsson & Jai Sachdev

Princeton Plasma Physics Laboratory

05/05/2017

Two-part discussion:

- 1 A more modular TRANSP
- 2 TRANSP as a framework & TRANSP as a component in other frameworks

We see (1) as a prerequisite for (2)

# The need for a more modular TRANSP

- FES has asked us to *"improve and modernize the code architecture"*
- The current C-shell based build system has limitations
  - No conditional build (build only the parts that you need)
  - Difficult to learn to use, understand and maintain
  - C shell is deprecated
- LEVGEO=7 (use ESC equilibrium solver) has been "indefinitely disabled" since March 26, 2008
- But I still had to fix a multiply-defined macro in ESC to make TRANSP build eight years later!
- We are working on a standard `make`-based system with conditional build
- `make --enable-ESC` to reverse disabled by default
- It is a bit painful to switch after forty years
- Could codes that provide heat and particle sources be external?

# TRANSP in other workflow managers / frameworks

- Framework: "software that connects codes and lets us run them as a single code"
- Workflow manager: "software that automates data processing done by a sequence of codes"
- Considerable overlap between them
- A workflow manager can be used as a framework
  - The Iter framework will probably be based on the workflow manager Kepler
  - But how about communication performance, and scalability for large simulations?
- What are OMFIT's plans?
- How do we make TRANSP integrate better in these (and other?) workflow managers / frameworks?
- Are their requirements different from Simulink's?

# TRANSP as a framework

- TRANSP is a framework to some extent:
  - "ls \$XE", several of those executables (mpi\_cql3d\_transp, etc.) are basically components executed from the TRANSP framework
  - But no generic interfaces. . .
- Uses source code from SWIM/IPS, including Plasma State
- When modular build system is in place, we will finalize plans for also making execution more modular
- Particle and source terms are natural candidates for componentization
- Solvers, however, need tightest possible coupling for numerical stability

# Donate some regression test cases!

- TRANSP has an automatic regression test system, where a fresh build runs a handful of cases every day and the correctness of the output is checked
- A summary from the latest regression test is on <http://w3.pppl.gov/~xshare/machine/TranspMachine.html>
- However, we need more diverse cases to minimize the risk of breaking less frequently used parts of the code
- Consider donating a few representative cases
- Users know best what parts of the code are critical for their purposes
- Will help keep TRANSP fully functional as we modernize the code