Demonstration of the Integration of BOUT++ into the OMFT Framework

by
T.M. Wilks

with
O. Meneghini, S. Smith, C. Moynihan, O. Izacard, X. Q. Xu, J. W. Hughes

1MIT-PSFC
2General Atomics
3PPPL
4UIUC
5LLNL

Presented at the
2018 BOUT++ Workshop
Livermore, CA
August 14, 2018
Overview of Tutorial

- OMFIT overview
- OMFIT setup options
- BOUT++ module overview
- Grid generation with Hypnotoad
- Running BOUT++
- Post-processing tools
Overview of Tutorial

- OMFIT overview
- OMFIT setup options
- BOUT++ module overview
- Grid generation with Hypnotoad
- Running BOUT++
- Post-processing tools

Active Participation

- NERSC user ID and password (train##)
- NoMachine or some Xming software
Overview of Tutorial

- OMFIT overview
- OMFIT setup options
- BOUT++ module overview
- Grid generation with Hypnotoad
- Running BOUT++
- Post-processing tools
- One Modeling Framework for Integrated Tasks

Separation of framework, modules, and projects allows each to evolve independently

- **Framework**
  - Physics independent API to manage data and run codes
  - Defines common set of functionality and capabilities

- **Physics module**
  - Collection of scripts for the execution of physics codes and workflows
  - Re-used for different physics studies

- **User project**
  - Collection of modules and data for specific physics studies
Overview of Tutorial

- **OMFIT overview**
- **OMFIT setup options**
- **BOUT++ module overview**
- **Grid generation with Hypnotoad**
- **Running BOUT++**
- **Post-processing tools**
Launch OMFIT on NERSC (Edison)

>> cd /global/project/projectdirs/atom/atom-install-edison/bin
>> ./omfit-dev

- Use suggested file location and id_rsa filename
- Press “enter” three times to leave password blank and continue launching OMFIT
Welcome to OMFIT

Accept license:
(emails Sterling and Orso to help keep track of usage)
Add your email and user names into preferences

This helps connect you to:
- the OMFIT community for help and error reports
- Other accounts like github and remote servers
Overview of Tutorial

- OMFIT overview
- OMFIT setup options
- BOUT++ module overview
- Grid generation with Hypnotoad
- Running BOUT++
- Post-processing tools
OMFIT allows easy setup and data management for BOUT++

- **Goal is to reduce the learning curve for running complex codes like BOUT++**
  - OMFIT handles details like input file setup, remote server connections, and data management
  - Allows user to focus on physics inputs and results and minimize user errors in setup

**OMFIT Framework**
- **BOUT++ Module**
  - Grid generation
  - BOUT++ input handling
  - Post-processing

**BOUT++ Code**
- BOUT.inp
- Bout.grid.nc
- Bout.exe
  (restart.nc)
- Bout.dmp.nc
- Bout.restart.nc
- Bout.log

Server independent
Import BOUT++ Module

File ➔ Import module …
Double Check Scratch Folder is Set Correctly

In “Terminal” Tab:
>echo $SCRATCH

OMFIT[‘MainSettings’][‘SERVER’][‘edison’][‘workdir’] = ‘/scratch#/scratchdirs/username/….’

Right click → edit tree entry
Open main GUI

Double click on "BOUT++" to open GUI

Can explore available scripts, settings, and data in python tree
Load example input files into project

Shared folder:
/global/homes/t/twilks/Share/BOUTworkshop18/

Load cxx and BOUT.inp “from file” in share folder
Compile the cxx file

Compile!
Overview of Tutorial

- OMFIT overview
- OMFIT setup options
- BOUT++ module overview
- Grid generation with Hypnootoad
- Running BOUT++
- Post-processing tools
Generate BOUT++ gridfile with Hypnotoad

- **Change Grid File (create new)**
  - Hypnotoad Grid generator

- **Load gfile and pfile from share folder**

- **Run hypnotoad**

Shared folder:
/global/homes/t/twilks/Share/BOUTworkshop18/
Generate Mesh with Hypnootoad

- Read geqdsk (will be in the same folder)
- Set number of radial and poloidal points
- Generate mesh
Output Mesh and load into OMFIT tree

- **Need to interface with the terminal to exit out of Hynutoad**
  
  Profile option: 1
  Temperature (eV): 1000
  Is this ok? Yes
  IDL>exit

- **This ends your OMFIT command that is running IDL and brings the gridfile back into the OMFIT tree**

  `root['GRIDGEN']['OUTPUTS']['GRID']`
Prepare gridfile to be passed to BOUT++

Write profiles from pfile to the grid
Overview of Tutorial

- OMFIT overview
- OMFIT setup options
- BOUT++ module overview
- Grid generation with Hypnotoad
- Running BOUT++
- Post-processing tools
Prepare BOUT++ simulation inputs

- “Parameters” tab handles BOUT++ input file options for physics assumptions
- “Server Settings” tab handles batch run settings
- “Run Settings” tab handles data fetching and convergence studies (still under development)
Run BOUT++

Console prints run status
OMFIT fetches the BOUT++ data back to the python tree

Data, restart, and log files are fetched into `root[OUTPUTS]`

Possible to fetch a trimmed number of time slices for analysis

With data available in "OUTPUTS", the "Analysis" tab becomes available
Overview of Tutorial

- OMFIT overview
- OMFIT setup options
- BOUT++ module overview
- Grid generation with Hypnotoad
- Running BOUT++
- Post-processing tools
Several simple analysis tools available

- Visualization of evolved quantities (e.g. pressure perturbation)
- Growth rate calculation
- Mode structure visualization
OMFIT can be used to simplify the process of running BOUT++

- Has potential to be coupled to entire pedestal workflow
  - Profile fitting
  - Kinetic EFITs
  - Varyped/ELITE/EPED

- GRIDGEN module has potential to be coupled with other codes
  - UEDGE
  - SOLPS

OMFIT Framework

- BOUT++ Module
  - Grid generation
  - BOUT++ input handling
  - Post-processing

BOUT++ Code

- Server independent
Always improving: BOUT++ module still under development

- This summer, SULI student Cody Moynihan developed
  - Saving (large amounts of) data into easily accessible tree structures for analysis of different scanned parameters
  - Quantitative checks on convergence with scripts to rerun if criteria is not met

- Users (you guys) can add analysis tools for the OMFIT BOUT++ community!
Always improving: BOUT++ module still under development

- **This summer, SULI student Cody Moynihan developed**
  - Saving (large amounts of) data into easily accessible tree structures for analysis of different scanned parameters
  - Quantitative checks on convergence with scripts to rerun if criteria is not met

- **Users (you guys) can add analysis tools for the OMFIT BOUT++ community!**
Questions/ Comments

?