Code Structure and Data flow Management for Edge and SOL Integration

N Bisai and ITM Group

Institute for Plasma Research, Bhat, Gandhinagar-382428, India

June 9, 2011

Introduction to Integrated Tokamak Modeling

- We started Integrated Tokamak Modeling (ITM) in India
- We already have many separate codes (equilibrium, stability, etc..)
- Interconnecting (I/O exchange) of the codes had started
- Criteria:
  - Programming structures as simple as possible
  - Mainly FORTRAN-90 is used for programming

In this talk, only ITM related to edge and SOL integration will be presented.

SIMD for ITM

- We did not use MPI or openMP for programming
- We use single instruction multiple data (SIMD) programming using FORTRAN 90/95
  1. The programmer writes a single operation, "+" say, and the compiler causes it to be carried out on multiple pieces of data in as parallel a manner as the underlying hardware allows (SMP)
  2. It is SIMD in Multi cores (Quad-core, Hexa-core, etc.. with hyper-theading Technology) Workstation

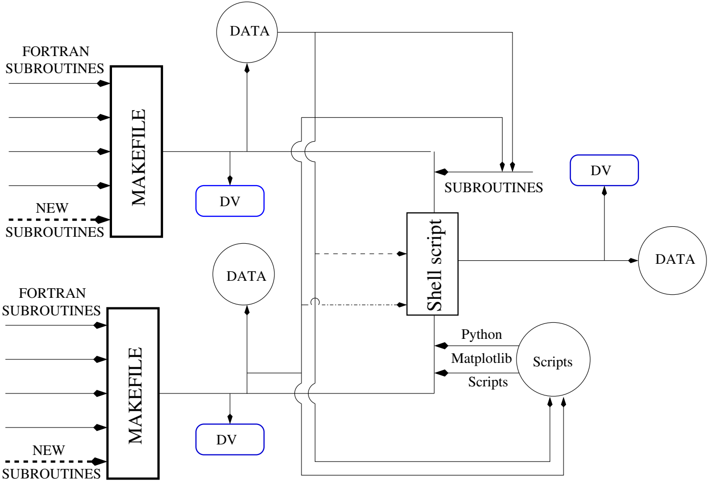
Softwares for ITM

Following softwares are used for the Integrated Tokamak Modeling

1. FORTRAN 90/95 (vectorization for SIMD)
2. NetCDF
3. NCARG (on fly data visualization)
4. Makefile
5. Shell Script
6. Python

Our ITM is based on Mainly OPEN SOURCE softwares (only non free software is FORTRAN 90/95 compiler). We used CLI rather than GUI.

Typical structure of ITM



DV=Data Visualization by NCARG or Python

Programming

Integrated Tokamak Modeling is based on extensive use of

- Main programming:
  1. Mainly Fortran 90 using vector programming in SIMD
  2. It also uses NETCDF (F90 API) library
  3. It also uses NCARG for visualization
  4. If a special diagnostic is needed we need to write new subroutines

NCARG for data visualization, programmed within main FORTRAN codes.

Programming (continue)

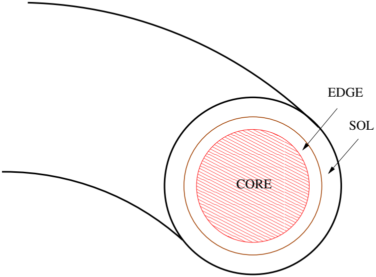
- Python programming: We use object oriented python programming for managing data flow

1. Numpy:
   ⇒Matrix manipulations for generating input data as well as for synthetic diagnostics. Statistical data description like probability distribution, nean, std, etc..
2. Scientific-python:
   ⇒python numerical library for short numerical computing as well for synthetic diagnostics
3. Python-matplotlib:
   ⇒2D & 3D Data visualizations.

Programming (continue)

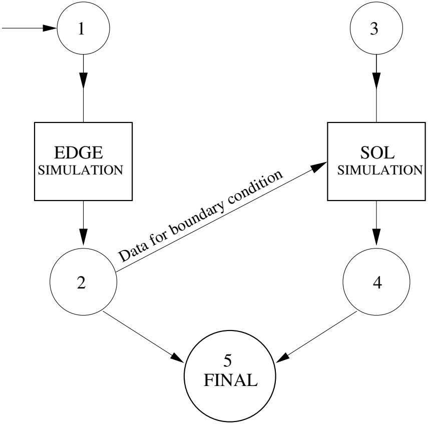
1. Makefile
2. Shell script (bash):
   - To run Fortran codes sequentially
   - Inter data communications
   - Data generation in fly and removal
   - To run python scripts
3. Python script:
   1. To run FORTRAN codes sequentially
   2. Inter connections among codes
   3. Data generation, analysis, synthetic data and visualization

ITM for Edge and SOL Coupling (Example)



- Edge region is governed by a set of equations
  Edge has influence from core
- SOL region is governed by another set of equations
  SOL has influence from edge

Data Flow for Edge and SOL



1. Input for edge
2. Output from edge (With Graphics)
3. Input for SOL
4. Output from SOL (With Graphics)
5. Edge+SOL data (With Graphics)

Summary & Conclusions

- Only we use only CLI. This is simple but robust for our purpose.
- We need to change only FORTRAN subroutines for different diagnostics purposes. Need recompiling and execution
- Data flow maintained mainly shell scripts and python scripts
- Data visualization by NCARG or matplotlib