Development of data analysis platform in KSTAR

June 02, 2017

Jayhyun Kim\textsuperscript{1a}, Dong Keun Oh\textsuperscript{1b}, Insik Choi\textsuperscript{1}, Sang-Hee Hahn\textsuperscript{1}, Sehoon Ko\textsuperscript{1}, Seong-heon Seo\textsuperscript{1}, Jae-Min Kwon\textsuperscript{1}, Byung-ho Park\textsuperscript{1}, and Syun’ichi Shiraiwa\textsuperscript{2}

\textsuperscript{1}National Fusion Research Institute, Daejeon, Republic of Korea
\textsuperscript{2}Massachusetts Institute of Technology, Cambridge, MA, United States

\textsuperscript{a}Presenter: Jayhyun Kim to jayhyunkim@nfri.re.kr
\textsuperscript{b}Major contributor: Dong Keun Oh to spinhalf@nfri.re.kr
Fusion data processing requires multi-level analysis
- Real-time analysis during the discharge for feedback control
- Between-shot analysis for feedforward setting
  - In-situ analysis demands interactive platform for effective data processing.
- Post-shot analysis with intensive computing
- Statistical analysis or machine learning among multi-shots for revealing the underlying rules
- Integrated data handling platform is important: data integrity

Fusion data (1D profile, 2D image) typically consists of multi-channel data.
- However, multi-channel data demand high computing performance due to limited time.
- Parallel computing with GPU is adequate for handling the multi-channel data.
  - Python and pyCUDA are a good platform for GPU enhanced parallel computing.
- GPU enhanced data processing is applied to KSTAR experiments.
  - Microwave reflectometry for edge density profile measurement
  - Thermo-hydraulic analysis of KSTAR superconducting coil
Contents

• Interactive between-shot analysis
  • Equivalent handling of experimental and processed data

• Integrated data handling platform: \(\pi\)Scope
  • Interactive between-shot analysis
  • Pre-processor for post-shot analysis

• GPU enhanced computing for multi-channel/multi-time slice fusion data
  • Between-shot analysis
  • Real-time analysis (potential application)
Display and analyze time series data:
- Plot waveform(s) of measured data (time, data) and processed data derived from them
- Program user-specific functions with using popular computer language:
  - e.g. Python interpreter for obtaining spectrogram from Mirnov coil data
- Specify relevant data set by defining configuration file.
  - His/her own configuration files with user-specific analysis capability
- Extra features:
  - Off-site data management: e.g. MDS+ DB → HDF5 or NETCDF

Display and analyze profile data from multi-channel diagnostics:
- Plot profile(s) of measured data in certain time slice with error bar
- Link with equilibrium information for transforming into flux coordinate
- Extra features (under development or consideration):
  - Automated data analysis
  - Data binning during periodic events: inter ELM or sawteeth periods
Example of interactive between-shot analysis tool in KSTAR

- High performance computing enables more analysis even during between-shot.
- However, still, the more, the better → parallel computing

MP-driven ELM suppression in KSTAR #11341

Toroidal field
Channel selection by resonant condition
ECE radiometer
Mirnov coil data
Fast Fourier transform
Spectrogram of Mirnov coil
Unified analysis of experimental measurement and its processed data

EM load measurement during disruption:
sequential analysis

Mode locking by 3D fields:
multi-returns from single analysis

No difference in handling the data with processed levels

Locked mode coils (measurement)
Not displayed here
Needs of integrated data handling platform

• Many components of data
  - Diagnostic signals are basically multi-source and multi-channel.

• Usually, no routine work-flow – *Yes, that’s experiment!*
  - Hard to standardize, and easy to lose the integrity
    → *Procedures must be scriptable reducing error-prone steps in rewriting.*

• Numerical modeling – *a basic process for deep investigation*
  - Physical domains are tightly coupled – this needs parametric feed-back.
  - Many stand alone “single-physics” codes.. **NOT considering integration.**

• Importance of visualization and systematic data browsing
  - Perspective leads insight.

*Rule the procedure. Never be slaves.*
As a solution: πScope – python based analysis workbench

• Dedicated design for the experimental study in plasma physics lab
  → Developed by S. Shiraiwa with PSFC, MIT

• Scriptable and reusable by object-oriented feature
  → Programmable modules + data objects in the workbench

• Flexible and extendable
  → It’s written in python, accessing python scripts and python class itself.

• Powerful integrity
  → Project tree objects ↔ Scripting or visual access by users

• Rich facilities
  → Built-in MDS+ interface of background process and queuing
  → Launching and controlling the code in external computing servers via SSH
  → Many browsing and visualization functions for various formats
Interplay between data and numerical models I

- EFIT in KSTAR
- parameters for FEQ code
- output of FEQ code

• FEQ: Fix boundary Equilibrium solver

• Rebuilt as pyfeq library by f2py compiler

• Interactive workflow in GUI platform with project tree

• MDS+ interface ↔ script in GUI ↔ code in any external machine
Interplay between data and numerical models II

PEQ text (ELITE format) is imported after editing and fitting process...
Interplay between data and numerical models III

Modeling process with GYRO (Eulerian solver of gyrokinetic-Maxwell equations) in remote server.

This code requires its own format of equilibrium profile so that a pre-processor “profile_gen” has been developed to import the experimentally generated data...

GEQDSK (EFIT data) + PEQ text (profiles in ELITE format)

Controlling GYRO code
- Prepare remote directory
- Send input files for pre-processor
- Run pre-processor (profiles_gen)
- Run GYRO for post analysis
Fusion plasma research = a science project with big experimental data

- Needs of systematic data handling from multi-source diagnostics
  ➔ Demanding performance of interactive data processing tools

- Multi-channel feature fits the data parallel computing model.
  ➔ Data parallelism based CUDA programming model

- Python as a language for interactive framework
  ➔ Easy to integrate GPU code with pyCUDA library
Application of GPU enhanced data processing in KSTAR

Case #1. Edge density profile by microwave reflectometer

Case #2. Heat load of superconducting coils by AC loss

Layout of KSTAR diagnostics
Why GPU enhanced data processing is crucial in fusion data processing.

CASE #1. edge density profile by microwave reflectometer
- Max. 8000 floats/signal x $O(1000)$ signals/shot x $O(10)$ wavelet scales
  = ~ 1 Gbyte/shot (max. 30 shots/day)
- Existing analysis code (serial code) → 1.2 sec/signal → 1000 sec/shot → 8 hours/day
- It is impossible to analyze the data during between-shot (<20 min.)
- By GPU enhanced data processing (Nvidia® Tesla® K20m + pyCUDA)
  → x 500 times faster (~2 sec/shot = ~1 min./day)

CASE #2. heat load of superconducting coils by AC loss
- Analysis of the operation scenario before/after operation for superconducting coil safety
- Existing analysis code (Fortran) → 15 min. for 40 sec. scenario (impossible)
- By GPU enhanced data processing + algorithm improvement
  → x 300 times faster (< 3 sec. for 40 sec. scenario)
- It is suitable for real-time calculation with discharge control (13 ms for 100 ms run).
  - On-line feedforward adjustment for long pulse operation: KSTAR or ITER
Microwave reflectometry for measuring edge density profile

Tool-path in reflectometry (~1000 signals in parallel)

Detected signal

Spectrogram (Morlet wavelet)

Tracing the reflected wave (peak values)

Phase recovery for reflected wave

Cumulative phase

Profile reconstruction (B-C algorithm)
CUDA algorithms for microwave reflectometry

Wavelet transform (element-wise mult. + cuFFT)
Phase recovery (element-wise operations)

Cut-off finding

Reduction for the indices by bitwise operation: \(x \oplus 00 = x, 00 \oplus x = x, 10 \oplus 01 = 11, x1 \oplus 1y = xy\)

Cumulative phase

Work efficient parallel scan

Profile reconstruction: partially parallelizable for numerical integration

Threads for multiple data: ~1000 signals per routine

Partial fine-graining of the phase integral (reduction) for each step (256 threads)
GPU performance & result: microwave reflectometry

2.3 sec. computation time

-> Affordable for between-shot analysis

- For 1000 signals with Tesla® K20m (including initial loading time of GPU code)
  - 500 times faster than serial code!
  - Very easy to integrate into the data processing platform as a python class
Heat source = inductive loss $\propto (dB/dt)^2$

0-D model: no liquid helium flow
(= pessimistic analysis for safety)

$$\frac{d}{dt} \begin{pmatrix} T_c \\ T_i \\ T_j \end{pmatrix} = \begin{pmatrix} \frac{-k_{ic}+k_{cj}}{C_c} & \frac{k_{ic}}{C_c} & \frac{k_{cj}}{C_j} \\ \frac{k_{ic}}{C_i} & \frac{-k_{ij}}{C_i} & \frac{k_{ij}}{C_j} \\ \frac{k_{cj}}{C_j} & \frac{k_{ij}}{C_j} & \frac{-k_{cj}+k_{ij}}{C_j} \end{pmatrix} \begin{pmatrix} T_c \\ T_i \\ T_j \end{pmatrix} + \begin{pmatrix} \frac{\dot{q}_c / C_c}{0} \end{pmatrix}$$

Individual threads for 1824 coil spots
Applied techniques in 0-D model of KSTAR PF coils

Key features for the FASTER computation:

- Simplified ODE scheme
  - Linearized implicit scheme with adaptive time step

- B-field matrix and 2-D or 3-D tables of material properties

Efficient code developing framework:

A pilot project of real time application
- 1.6 sec. for 45 sec. evolution
- Existing FOTRAN code spends 10 min. for the same calculation – 300 times faster!
- Instantaneous analysis is possible for designed scenarios. – checking operation safety
- It is feasible to real-time application for plasma control.

→ But, no support of RTOS drivers yet
Summary

- Interactive between-shot analysis
  - Python-based tool is developed and will be implemented in $\pi$Scope.
  - User-defined libraries are being accumulated.

- Integrated data handling platform: $\pi$Scope
  - Interactive between-shot analysis: equilibrium and profile processing
  - Pre-processor for post-shot analysis: GYRO run within the platform

- GPU enhanced computing for multi-channel/multi-time slice fusion data
  - Between-shot analysis: e.g., edge reflectometry
  - Real-time analysis: e.g., AC loss analysis of PF coils
Future plan for real-time decision: disruption prediction

- Multiple precursors and faults
- Multiple responses: MGI and forced landing

Early warning?

Internal diagnostics

External diagnostics

Resulting signal

Temporal dropdown

Forced landing of discharge when locking occurs

• Multiple precursors and faults
• Multiple responses: MGI and forced landing
Future plan for real-time decision: disruption prediction II

- Disruption prediction in KSTAR currently adopts IF-THEN-RULES based on known criteria.
- It may be insufficient for increasing the reliability → artificial intelligence (neural networks)
Human's first and perhaps last victory for AlphaGo in 2016

AlphaGo resigns: The result “W+Resign” was added to the game information.