Executions of a Fusion Drift Kinetic Equation solver on Grid

A J. Rubio-Montero\textsuperscript{1}, L. A. Flores\textsuperscript{1}, F. Castejón\textsuperscript{1,2}, E. Montes\textsuperscript{1}, M. Rodríguez-Pascual\textsuperscript{1} and R. Mayo\textsuperscript{1}.

\textsuperscript{1}CIEMAT(Spain)
\textsuperscript{2}Lab. Nacional de Fusión (Spain)

18\textsuperscript{th} Euromicro International Conference on Parallel, Distributed and Network-Based Computing
Several Grid-related projects at CIEMAT:

- EGEE. Large Grid resources (fusion VO ~ 25,000 CPUs).
- EUFORIA: Link with HPCs.
- EELA: Latin-American Grid.
- DKESG: Example of application running on EELA.
Outline

- Neoclassical transport and its simulation
- Variational DKE solver code
- Porting DKES2 to Grid

- Test performed
  - Plasma results for TJ-II
  - Performance results

- State of the work
Neoclassical (NC) transport is

- a fundamental step of the complete simulation cycle of the behaviour of plasmas inside Fusion reactors.
- caused by magnetic field inhomogeneities and by collisions
- always present in all Fusion devices (Relevant in stellarators and 3D tokamaks).
- It is a low boundary of total transport
- usually used to study the efficiency of a certain coil configuration before it is implemented.
- Interesting for 3D ITER.

NC transport can be simulated by Drift Kinetic Equation Solver (DKES) or Monte Carlo (MC) methods

- DKES advantage over MC: can calculate all transport coefficients, MC only can estimate the diagonal ones.
- DKES drawback under MC:
  - high computation time and memory consumption, being usually executed on shared memory computers.
  - MC methods are easily deployed on Grid.
DKES (Drift Kinetic Equation Solver): STANDARD NC Transport Tool.

\[
\begin{pmatrix}
\Gamma \\
Q \\
j_{||}
\end{pmatrix} = -
\begin{pmatrix}
L_{11} & L_{12} & L_{13} \\
L_{12} & L_{22} & L_{23} \\
L_{13} & L_{23} & L_{33}
\end{pmatrix}
\begin{pmatrix}
\nabla n \\
\nabla T \\
\nabla \phi
\end{pmatrix}
\]

For Instance:

\[
\Gamma^s = -L_{11}^s \nabla n - L_{12}^s \nabla T - L_{13}^s \nabla \phi
\]
Variational DKES version (W.I. van Rij & S.P. Hirshman)
- Obtain upper and lower bounds for the NC diffusion coefficients of a prescribed toroidal plasma equilibrium.
- Widely accepted by the Fusion Community.
- Applied to stellarators such as TJ-II, HSX, CHS, LHD, ATF, W7-AS; and to projected ones: W7-X, NCSX, QPS.
- We plan also apply it to 3D tokamaks like ITER.

Available code
- Developed in Fortran77 in 1989
- Implemented for obsolete shared memory hosts (CRAY-1, CRAY-X-MP)
- Modified (~2001) to run on SGI Origin 3000 MIPS machines (IRIX64).
- Parametric and sequential nature
Every Transport coefficient at a given radial position is written as an integral of a mono-energetic coefficient.

\[
L_{ij}^s(n,T,\phi) = \frac{2n}{\pi} \int_0^\infty dK \sqrt{K} e^{-K} g_i(K)g_j(K) D_{ij}(\frac{\nu}{\sqrt{K}}, \frac{\nabla \phi}{\sqrt{K}})
\]

\[\nu = v_0 n T^{-3/2}\]

Independent processes to estimate the monoenergetic coeff.

Number of processes=
#collisionalities \(10^2\) x #fields \(10^2\) x #energies \(10^3\) x #radial positions \(10^2\) x #species \(2\).
How to work with Variational DKES

- Requires continuously manual configuring, recompiling code and collection of high number of output data files
- Magnetic configuration is described by Fourier series
- Energy distribution function of particles by a series of Legendre polynomials.
- Parameter sweep of normalized collisionality and electric field

>10^9 combinations x reactor

www.eu-eela.eu
All code in a nutshell

- Algorithms from DKES2 software up-to date and extended (new DKES3 binary)
- Ported and optimized to Linux x86-32/64
- No software must be installed in resources

Grid-enabled:

- Automatic generation of input files
- Unattended management of jobs to Grid through GridWay
- Configurable performance monitor
- Output checking and result compilation.
- Run on Globus and gLite Grids
- Tested on EGEE-III, EELA-2 and regional infrastructures.
Each DKES3 instance is submitted with its input to Grid via GridWay.

A performance monitor is executed jointly the DKES3 job.

Resubmit DKES3 task if output is incorrect.

Checks the output.

Var. range (vV, Er/vV) Reactor (Rmax, |B| file, Plasma State)

DKESG-Diffusion

Normalized Diffusion coefficients

Results combination & post-processing

Normalized Diffusion Coefficients
Execution tests for TJ-II (July 2009)

- **Applied to TJ-II Flexible Heliac**
  - Sited at CIEMAT Madrid (Spain)
  - Began fully operational in 1997
  - 1.5 m major plasma radius

- **Inputs generated**
  - Normalized collisionality [0.1 ... 100] with 0.5 step
  - 5 electrostatic potentials (assumed constant on every magnetic surface)
  - Fixed: 4cm plasma radius, 100 Legendre polynomials, 343 Fourier coeff.

- **Total**:
  - 2500 pairs processed within 500 jobs
  - Distributed among 12 EELA-2 Grid sites.
  - 1 job ~ 4 mins in a core Xeon 5160(3.0Ghz) → 500 jobs ~ 33 hours
  - **Total time spent:** ~53 mins.
  - CPU time: ~ 45 hours (middleware & file transfer overhead not included)
Results: NC diffusion coefficients

NC diffusion coefficients for \(T=1\) KeV and \(M=1u\)

- Resistivity Enhancement (\(D_{33}\))
- Bootstrap current (\(D_{13}\))
- Temperature distribution (\(D_{11}\))

Is necessary lower granularity

>10^6 CMUL – efield combinations
Performance measurements and discussion

Execution statistics:
Only sites with more than 25% of successfully terminated jobs has been included (6/12).

Accumulated time for whole test:
- Real time for DKES calculus: ~ 45 hours
- Intrinsic Overhead (Middleware & file transfers): ~ 61.5%
- Fail overhead (total spent time by failed jobs): ~92 hours

Increasing the number of DKES calculus by Grid job will reduce the intrinsic overhead percentage

Note: It is necessary >10^6 DKES task by selected radius for obtaining accurate results

- It is a specific fitting for the certain employed infrastructure.
- Its influence is similar for all applications.
- It is a task for the GridWay administrator.
Create a new framework: DKEsG (Drift Kinetic Equation solver for Grids) to

- Easily retrieve all neoclassical transport coefficients:

  \[
  L_{ij}(n,T,\phi) = \frac{2n}{\pi} \int_0^\infty dK \sqrt{K} e^{-K} g_i(K) g_j(K) D_{ij}(\frac{\nabla}{\sqrt{K}}, \frac{\nabla\phi}{\sqrt{K}})
  \]

- **High level implementation on java DRMAA standard**
  - To also run on local clusters through PBS/SGE/LSF
  - To make code more solid, extensible and reusable.

- **Filling a broad database with the complete [configuration - transport matrix – state] for several Fusion reactors to:**
  - Bring results to Fusion Community
  - Avoid performing the same simulation twice and easy re-analysis

- **Easy interaction with other Fusion applications to build complex workflows to**
  - Perform complete simulations of plasma (FAFNER2/MaRaTra – DKEsG)
  - Find more optimized configurations of both stellarators and 3D tokamkas (ITER). (VMEC – DKEsG)
Module has been implemented to calculate NC transport coefficients.

Real simulations is being performed for TJ-II in EELA-2 infrastructure and a preliminary coefficients database is being filled out.

First fusion application workflows is being prepared for testing.
Thanks for you attention