MAX-PLANCK-INSTITUT
FÜR PLASMAPHYSIK

Technical overview of the GENE-X code

P. Ulbl[1]

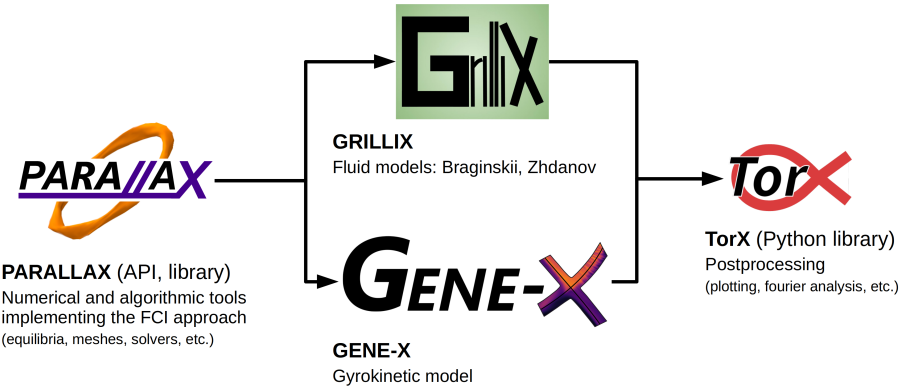
[1]Max Planck Institute for Plasma Physics,
Boltzmannstraße 2, 85748 Garching, Germany

philipp.ulbl@ipp.mpg.de

EUROfusion

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

The PARALLAX Code Family

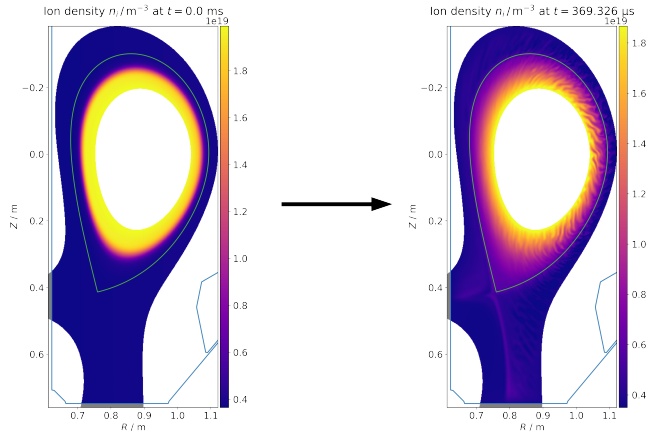


In a Nutshell

GENE-X is a gyrokinetic code to simulate plasma turbulence in **X-point geometries**.

Features:

- global
- non-linear
- full-f
- electromagnetic
- collisional



Equations solved by the code I

$$\frac{\partial f_\alpha}{\partial t} + v_{||}\frac{\mathbf{B}^*}{B^*_{||}}\cdot\nabla f_\alpha + \frac{c}{q_\alpha B^*_{||}}\mathbf{b}\times(\mu\nabla B + q_\alpha\nabla\phi_1)\cdot\nabla f_\alpha$$
$$- \frac{\mathbf{B}^*}{m_\alpha B^*_{||}}\cdot(\mu\nabla B + q_\alpha\nabla\phi_1)\frac{\partial f_\alpha}{\partial v_{||}} - \frac{q_\alpha}{m_\alpha c}\frac{\partial A_{1,||}}{\partial t}\frac{\partial f_\alpha}{\partial v_{||}} = C_\alpha(f_\alpha).$$

`timestep`
`operators/op_rhs_vlasov_static`
`operators/op_rhs_vlasov_dynamic`
`operators/collisions`

- $\alpha\ldots$ particle species. (electrons + ions)
- $v_{||}\ldots$ parallel velocity
- $\mu\ldots$ magnetic moment

Equations solved by the code II

$$-\nabla\cdot\left(\sum_\alpha \frac{m_\alpha c^2 n_{0,\alpha}}{B^2}\nabla_\perp\phi_1\right) = \sum_\alpha q_\alpha \int f_\alpha \,\mathrm{d}W,$$

$$-\Delta_\perp A_{1,||} = 4\pi\sum_\alpha \frac{q_\alpha}{c}\int v_{||} f_\alpha \,\mathrm{d}W,$$

$$-\left(\Delta_\perp + 4\pi\sum_\alpha \frac{q_\alpha^2}{m_\alpha c^2}\int v_{||}\frac{\partial f_\alpha}{\partial v_{||}}\,\mathrm{d}W\right)\frac{\partial A_{1,||}}{\partial t} = 4\pi\sum_\alpha \frac{q_\alpha}{c}\int v_{||}\left(\frac{\partial f_\alpha}{\partial t}\right)^\star \mathrm{d}W.$$

operators/field_solve_operators/op_solve_qn_eq

operators/op_mom_maxwells_eq

operators/field_solve_operators/op_solve_amps_law

operators/field_solve_operators/op_solve_ohms_law

operators/op_mom_ohms_law

Data structure

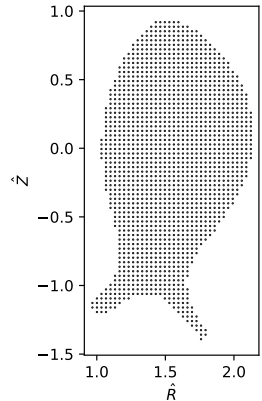
- The distribution function f_α is stored on a 5D+species grid $(RZ, \varphi, v_{||}, \mu, \alpha)$
- The electrostatic potential ϕ_1, parallel electromagnetic vector potential $A_{1,||}$ and parallel electric field $E_{1,||} = \partial A_{1,||}/\partial t$ are stored on a 3D grid (RZ, φ)
- Also moments (results of $\mathrm{d}W$ integrals) are 3D
- We have implemented types that contain this data + handle the pack/unpack to/from 1D for MPI exchange:
  `datastructures/data_storage_5d`
  `datastructures/data_storage_2d`
- All relevant data of the simulation is stored in `state_vector`
- All operations done on data are implemented in `operators`
- The state vector keeps its data encapsulated to prevent global access. Operators are applied via e.g. `state_vector%apply(op_rhs_vlasov_static)`

Dimensions

- RZ - Cartesian but unstructured 2D grid (see figure right)
- φ - angle dimension - uniform
- $v_{||}, \mu$ - velocity space - uniform 2D Cartesian
- α - species - parameter

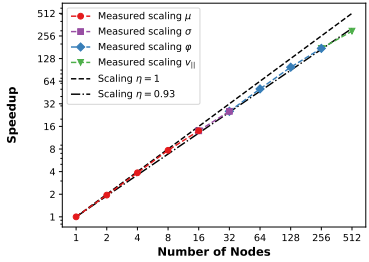
Conventions in the code:

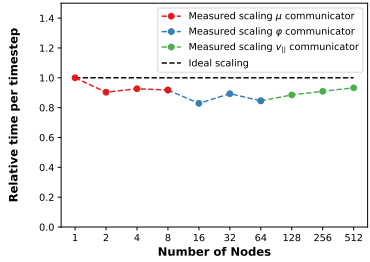
- Loop indices `i, k, l, m, n` for $RZ, \varphi, v_{||}, \mu, \alpha$
- `lb` and `ub` - 5D arrays with lower and upper bound - **with MPI ghosts**
- `lb_stripped` and `ub_stripped` - 5D arrays with lower and upper bound - **no MPI ghosts**



Parallelization

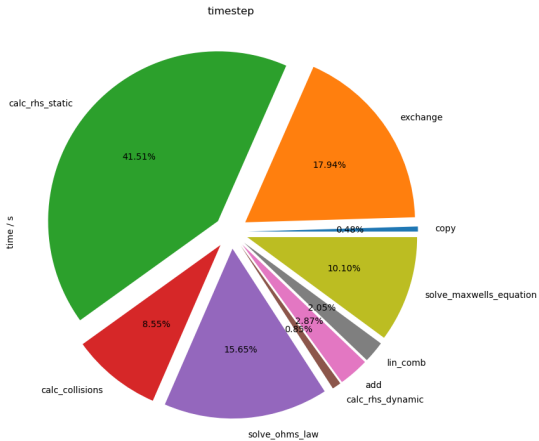
- Hybrid MPI+OMP. MPI in $\varphi, v_{||}, \mu, \alpha$. OMP in RZ
- Typical problem size approx. $(RZ, \varphi, v_{||}, \mu, \alpha)$ = $(200000, 32, 80, 20, 2)$
- Latest production runs on Raven supercomputer (MPCDF)
  320 nodes - 72 cores. 640 MPI procs, 36 cores for OMP.
  MPI decomposition e.g. $(RZ, \varphi, v_{||}, \mu, \alpha)$ = $(1, 16, 2, 10, 2)$.
- Good strong (left) + weak (right) scaling





Performance

Example for recent production run with $(RZ, \varphi, v_{||}, \mu, \alpha) = (200000, 32, 80, 20, 2)$



Wall-clock time: approx 2.6 s
(configuration given on last slide)

Intensity ranking:

1. `op_rhs_vlasov_static`
2. `field_solve_operators`
3. MPI `exchange`
4. `collisions`

Interface to PARALLAX

- `mesh/mesh_5d` contains all calls to PARALLAX library routines
- Important tasks: create equilibrium (magnetic field), create 2D RZ mesh (with multigrid), create elliptic solver (mGMRES)
- PARALLAX has 3 types of points: inner + boundary + ghost
- Currently GENE-X has 2 types: inner + ghost (contains PARALLAX boundary)
- Currently **no** MPI decomposition in RZ

Re-ordering of the unstructured mesh

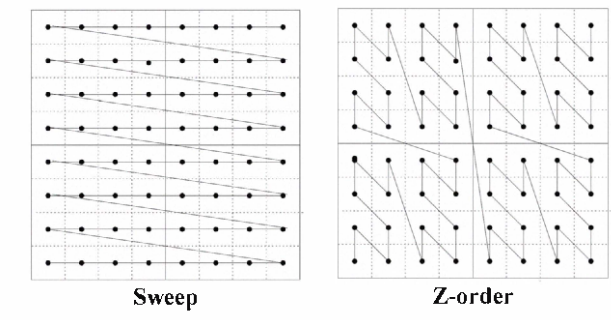
- Implemented in PARALLAX

```
call this%multigrid_3d(k)%init(..., reorder_size=X)
```

- However, we use `X=0` - **no** reordering so far
- Reordering would change the order of inner + boundary + ghost points

What is reorder?

https://en.wikipedia.org/wiki/Z-order_curve



May increase performance for stencil heavy operations.

Why reorder?

- Performance limited by memory access - bottleneck `op_rhs_vlasov_static`
- Finite difference stencils in RZ - because dimension is unstructured, neighbor search is needed
- Finite difference stencils in φ require CSR matrix-vector dot product - involves stencil in RZ due to interpolation
- Effect of reordering on PARALLAX elliptic solvers expected (may dig out details)
- Question: do all code parts **profit** from reordering?

Code Infrastructure

- Program for code benchmarks of individual operators is available `benchmarks/benchmark_operators` (field solvers not included so far)
- Workflow: protected master on Gitlab - branch out for new feature, merge request ASAP when ready to avoid conflicts (CI/CD)
- Unit tests are performed automatically when pushing to Gitlab. These may take a while due to the slurm queue. Best run tests manually before pushing to identify errors earlier
- New functions, subroutines require new tests if not covered by existing tests or if isolated feature
- Code review before merge to double check bugs
- PARALLAX coding standards available

Open questions: Code access? Runners for unit tests?

Summary

Current Status

- Bottleneck is memory access
- `op_rhs_vlasov_static` is most intense part of the code
- RZ dimension is unstructured
- Morton Z-reordering implemented in PARALLAX for RZ dimension
- Benchmark program for individual code parts available

Contact

philipp.ulbl@ipp.mpg.de

Todos

- Main goal: performance optimization of `op_rhs_vlasov_static`
- Enable reordering by fixing handling of boundary/ghost cells
- Identify other changes needed to enable reorder
- Benchmark effect on `op_rhs_vlasov_static` and other code parts (maybe add field solver benchmark?)