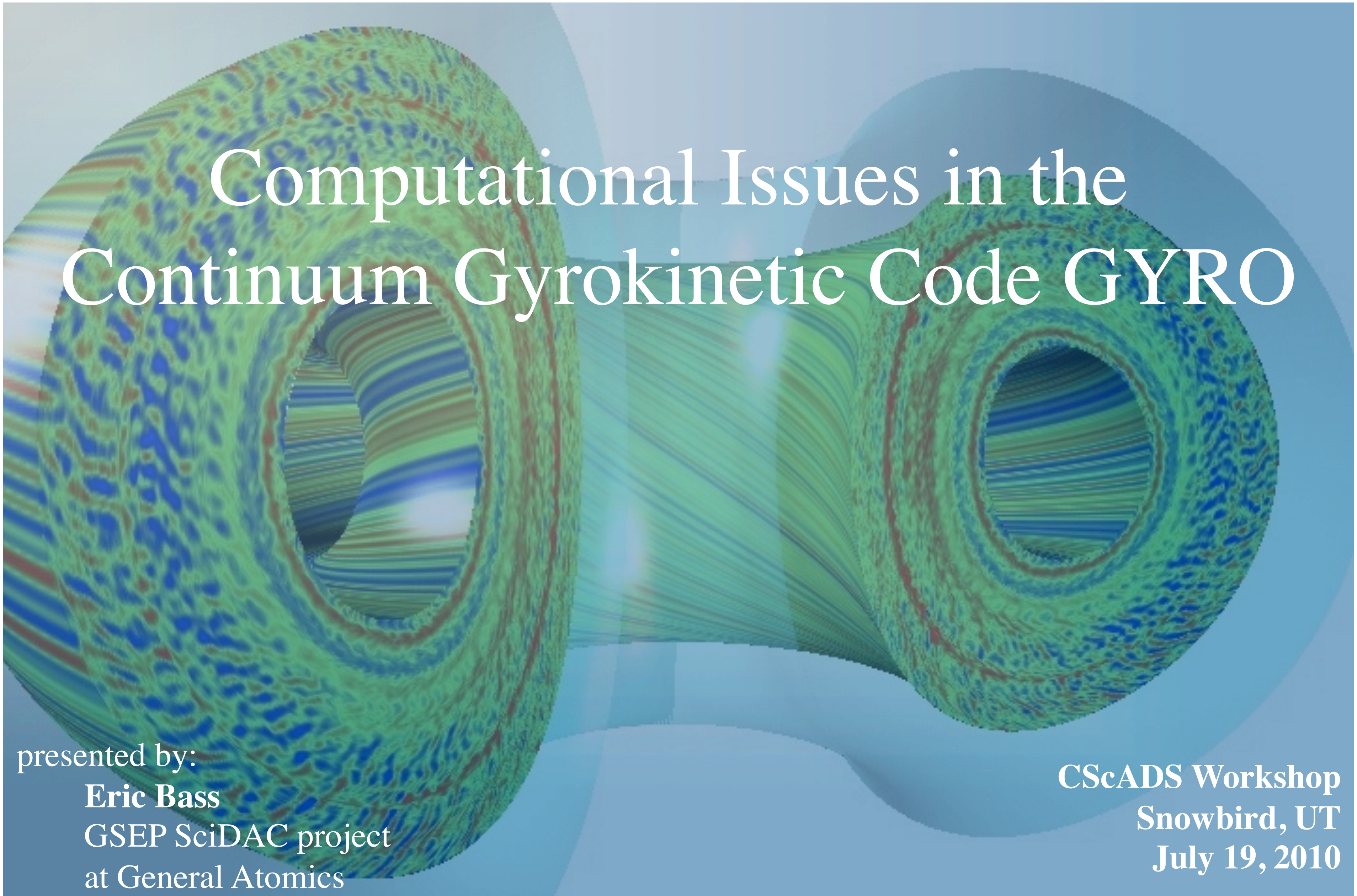


Computational Issues in the Continuum Gyrokinetic Code GYRO

presented by:

Eric Bass

GSEP SciDAC project

at General Atomics

CScADS Workshop

Snowbird, UT

July 19, 2010

GENERAL ATOMICS

About GYRO

Purpose: To predict transport driven by turbulence in toroidal nuclear fusion devices called tokamaks.

About GYRO

Purpose: To predict transport driven by turbulence in toroidal nuclear fusion devices called tokamaks.

Method: Solve the gyrokinetic equation on a five-dimensional phase-space grid.

- 3D real space: 2D grid + 1 spectral (toroidal *n* number) degree of freedom
- 2D velocity-space grid comes from averaging out fast cyclotron motion.

About GYRO

Purpose: To predict transport driven by turbulence in toroidal nuclear fusion devices called tokamaks.

Method: Solve the gyrokinetic equation on a five-dimensional phase-space grid.

- 3D real space: 2D grid + 1 spectral (toroidal *n* number) degree of freedom
- 2D velocity-space grid comes from averaging out fast cyclotron motion.

Parallelization: The 2D velocity-space grid and 1 spectral degree of freedom are distributed (separate MPI_COMM groups). Each processor follows a 2D real-space grid.

About GYRO

Purpose: To predict transport driven by turbulence in toroidal nuclear fusion devices called tokamaks.

Method: Solve the gyrokinetic equation on a five-dimensional phase-space grid.

- 3D real space: 2D grid + 1 spectral (toroidal *n* number) degree of freedom
- 2D velocity-space grid comes from averaging out fast cyclotron motion.

Parallelization: The 2D velocity-space grid and 1 spectral degree of freedom are distributed (separate MPI_COMM groups). Each processor follows a 2D real-space grid.

Multiple operating modes: GYRO can follow the time evolution from initial conditions or find the eigenvalues of the full system (GKEIGEN, SLEPc/PETSc) or the much-reduced Maxwell dispersion matrix (FIELDEIGEN, spatial degrees only).

Notes about the GYRO time step (initial-value mode)

• GYRO uses a hybrid implicit-explicit (IMEX) scheme with fixed time steps. It can also be set to use fully explicit Runge-Kutta time steps as well.

Notes about the GYRO time step (initial-value mode)

- GYRO uses a hybrid implicit-explicit (IMEX) scheme with fixed time steps. It can also be set to use fully explicit Runge-Kutta time steps as well.

- The (cyclotron averaged) fields are solved for on each processor using (usually sparse) linear algebra.

Notes about the GYRO time step (initial-value mode)

• GYRO uses a hybrid implicit-explicit (IMEX) scheme with fixed time steps. It can also be set to use fully explicit Runge-Kutta time steps as well.

• The (cyclotron averaged) fields are solved for on each processor using (usually sparse) linear algebra.

• An MPI_ALL_REDUCE produces the total field actually used in the equations of motion of the system (linear or non-linear operation).

Notes about the GYRO time step (initial-value mode)

• GYRO uses a hybrid implicit-explicit (IMEX) scheme with fixed time steps. It can also be set to use fully explicit Runge-Kutta time steps as well.

• The (cyclotron averaged) fields are solved for on each processor using (usually sparse) linear algebra.

• An MPI_ALL_REDUCE produces the total field actually used in the equations of motion of the system (linear or non-linear operation).

• In non-linear operation, MPI calls over the COMM representing the spectral degree of freedom are required to evaluate the non-linear term.

Notes about the GYRO time step (initial-value mode)

• GYRO uses a hybrid implicit-explicit (IMEX) scheme with fixed time steps. It can also be set to use fully explicit Runge-Kutta time steps as well.

• The (cyclotron averaged) fields are solved for on each processor using (usually sparse) linear algebra.

• An MPI_ALL_REDUCE produces the total field actually used in the equations of motion of the system (linear or non-linear operation).

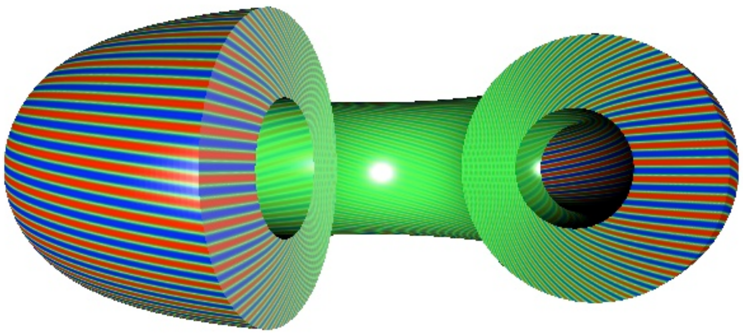
• In non-linear operation, MPI calls over the COMM representing the spectral degree of freedom are required to evaluate the non-linear term.

• Actual time advance operations for one entire spatial cross section (at one or more velocity or spectral grid points) are handled by each processor.

GYRO is versatile and scalable.

Local, linear simulations quickly give all features of the most-unstable mode even on a desktop machine.

1-32 cores , < 1 hour

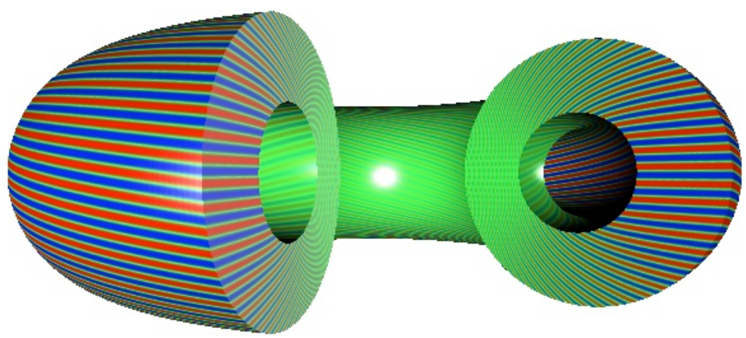


Waltz standard case flux tube

GYRO is versatile and scalable.

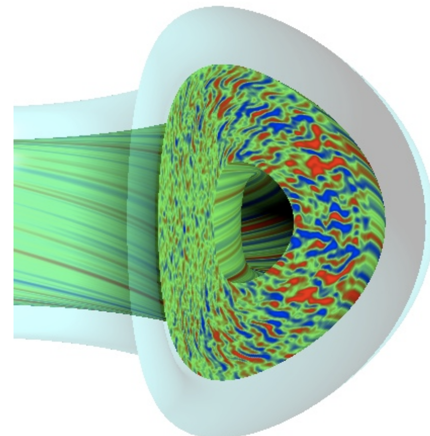
Local, linear simulations quickly give all features of the most-unstable mode even on a desktop machine.

1-32 cores , < 1 hour



Waltz standard case flux tube

Candy supertorus case



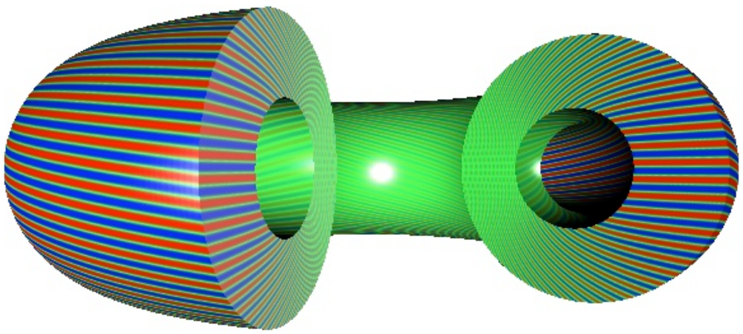
Non-linear simulations track multiple toroidal *n* numbers.

16-512 cores ,
1-48 hours

GYRO is versatile and scalable.

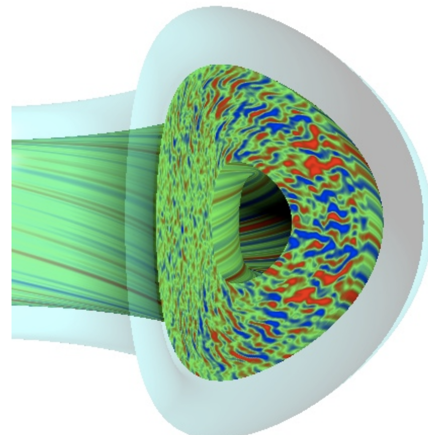
Local, linear simulations quickly give all features of the most-unstable mode even on a desktop machine.

1-32 cores , < 1 hour



Waltz standard case flux tube

Candy supertorus case

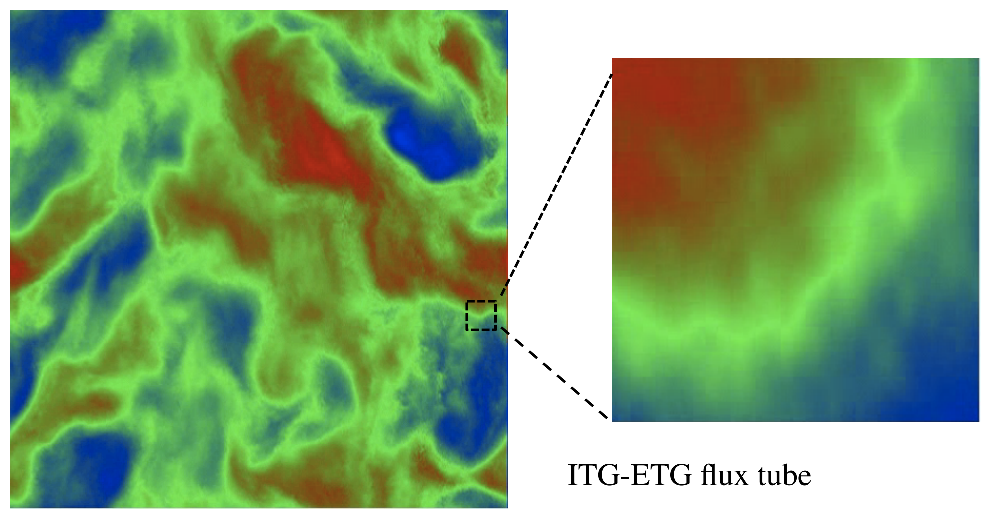


Non-linear simulations track multiple toroidal *n* numbers.

16-512 cores ,
1-48 hours

Massively multi-scale, non-linear simulations have disparate length and time scales and are only practical on terascale+ machines.

~1000 cores , 10 hours-7 days



ITG-ETG flux tube

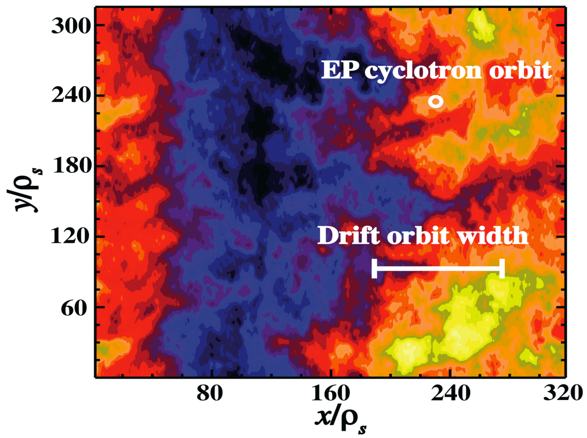
New GYRO simulations are pushing toward petascale territory

Multiple nonlinear instances managed in tGYRO can efficiently use 10,000+ cores
(talk by Luc Peterson)

New GYRO simulations are pushing toward petascale territory

Multiple nonlinear instances managed in tGYRO can efficiently use 10,000+ cores (talk by Luc Peterson)

Non-linear cases with high spectral resolution are highly parallelizable...

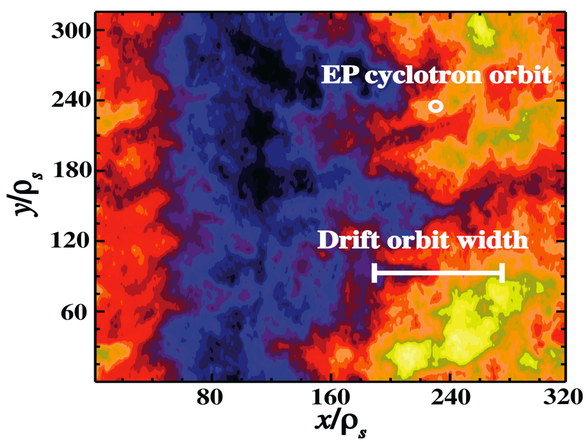


ϕ - <ϕ> in a nonlinear GYRO run with driven TAE/EPM turbulence.

New GYRO simulations are pushing toward petascale territory

Multiple nonlinear instances managed in tGYRO can efficiently use 10,000+ cores (talk by Luc Peterson)

Non-linear cases with high spectral resolution are highly parallelizable...



ϕ - <ϕ> in a nonlinear GYRO run with driven TAE/EPM turbulence.

max cores=2560

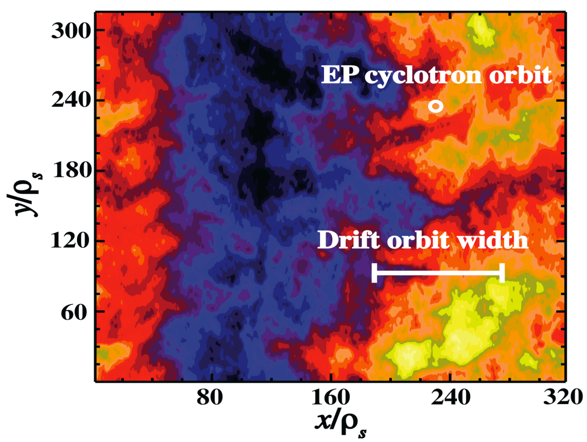
But, for 100% efficiency max cores=1280

...but maximum efficiency requires a corresponding increase in real-space grid density because spectral and spatial degrees of freedom are convoluted in the non-linear terms. Total calculation time is increased.

New GYRO simulations are pushing toward petascale territory

Multiple nonlinear instances managed in tGYRO can efficiently use 10,000+ cores (talk by Luc Peterson)

Non-linear cases with high spectral resolution are highly parallelizable...



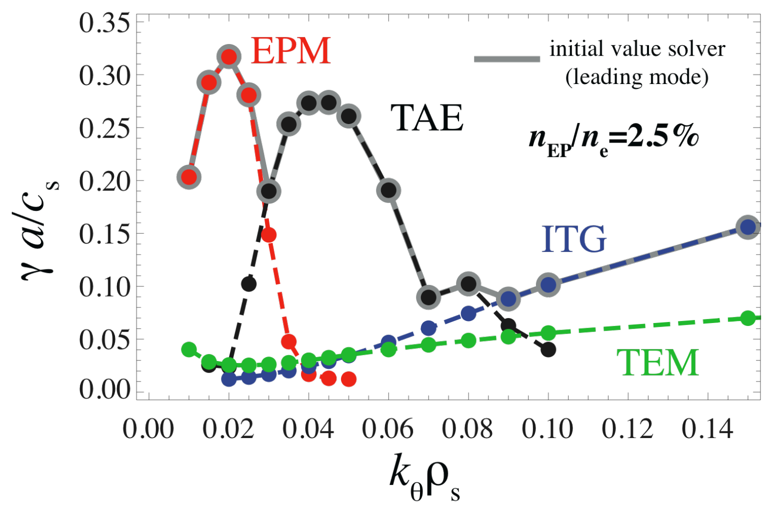
ϕ - <ϕ> in a nonlinear GYRO run with driven TAE/EPM turbulence.

max cores=2560

But, for 100% efficiency max cores=1280

...but maximum efficiency requires a corresponding increase in real-space grid density because spectral and spatial degrees of freedom are convoluted in the non-linear terms. Total calculation time is increased.

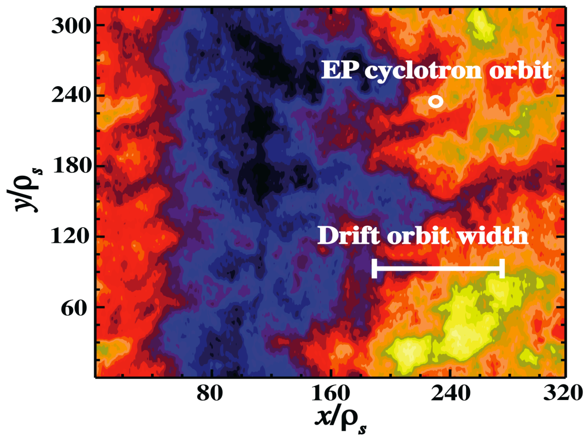
Even spatially small cases of GKEIGEN solve 25,000×25,000 matrices with parallel processing...



New GYRO simulations are pushing toward petascale territory

Multiple nonlinear instances managed in tGYRO can efficiently use 10,000+ cores (talk by Luc Peterson)

Non-linear cases with high spectral resolution are highly parallelizable...



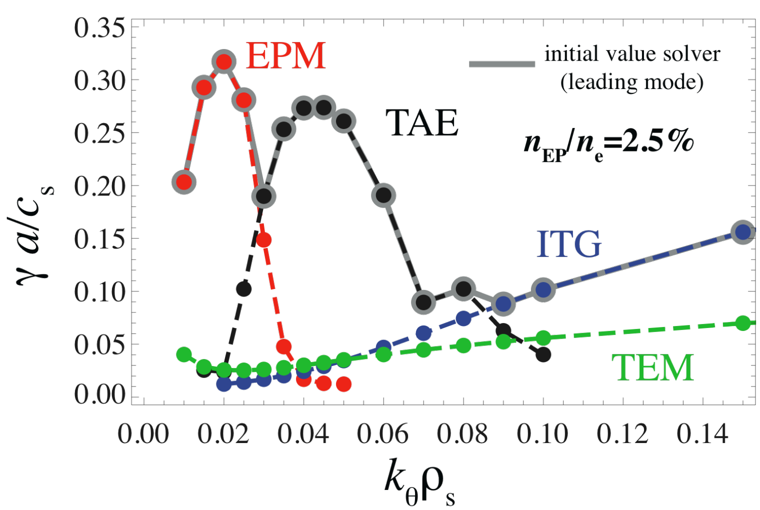
ϕ - <ϕ> in a nonlinear GYRO run with driven TAE/EPM turbulence.

max cores=2560

But, for 100% efficiency max cores=1280

...but maximum efficiency requires a corresponding increase in real-space grid density because spectral and spatial degrees of freedom are convoluted in the non-linear terms. Total calculation time is increased.

Even spatially small cases of GKEIGEN solve 25,000×25,000 matrices with parallel processing...



...but GKEIGEN cases are implemented now with GYRO's parallelization scheme. Total cores ≤ 64.

Some bottlenecks limit the largest simulations.

Cases involving an energetic particle (EP) species show such bottlenecks.

Some bottlenecks limit the largest simulations.

Cases involving an energetic particle (EP) species show such bottlenecks.

Stiff term in the gyrokinetic equation:

$$\frac{\partial h_{a,n}}{\partial t} - i\omega_\theta H_{a,n} - i\omega_d H_{a,n} - i\omega_E h_{a,n} - i\omega_* \Psi_{a,n} + \frac{q\rho_{s,\mathrm{unit}}}{rG_r} a\left\{\hat{h}_a, \hat{\Psi}_a\right\} = C_a^{CL}\left[H_{a,n}\right]$$

Curvature drift (ω_d) term is proportional to particle energy. EP requires small time steps.

Some bottlenecks limit the largest simulations.

Cases involving an energetic particle (EP) species show such bottlenecks.

Stiff term in the gyrokinetic equation:

$$\frac{\partial h_{a,n}}{\partial t} - i\omega_\theta H_{a,n} - i\omega_d H_{a,n} - i\omega_E h_{a,n} - i\omega_* \Psi_{a,n} + \frac{q\rho_{s,\text{unit}}}{rG_r} a\left\{\hat{h}_a, \hat{\Psi}_a\right\} = C_a^{CL}\left[H_{a,n}\right]$$

IMEX time-integration scheme used to mitigate the electron parallel advection (ω_q) term should probably be avoided for ω_d term for physics reasons.

Curvature drift (ω_d) term is proportional to particle energy. EP requires small time steps.

Some bottlenecks limit the largest simulations.

Cases involving an energetic particle (EP) species show such bottlenecks.

Stiff term in the gyrokinetic equation:

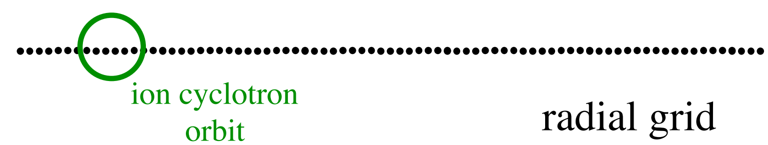
$$\frac{\partial h_{a,n}}{\partial t} - i\omega_\theta H_{a,n} - i\omega_d H_{a,n} - i\omega_E h_{a,n} - i\omega_* \Psi_{a,n} + \frac{q\rho_{s,\mathrm{unit}}}{rG_r} a\left\{\hat{h}_a, \hat{\Psi}_a\right\} = C_a^{CL}\left[H_{a,n}\right]$$

IMEX time-integration scheme used to mitigate the electron parallel advection (ω_q) term should probably be avoided for ω_d term for physics reasons.

Curvature drift (ω_d) term is proportional to particle energy. EP requires small time steps.

Expensive cyclotron average and field solve:

$\Psi_{a,n}$ is a cyclotron-averaged composite field variable.



Some bottlenecks limit the largest simulations.

Cases involving an energetic particle (EP) species show such bottlenecks.

Stiff term in the gyrokinetic equation:

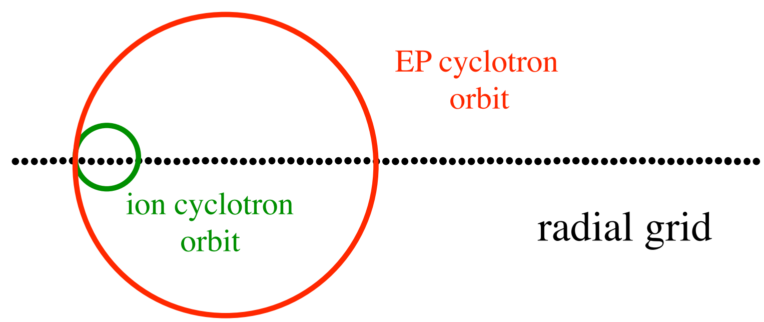
$$\frac{\partial h_{a,n}}{\partial t} - i\omega_\theta H_{a,n} - i\omega_d H_{a,n} - i\omega_E h_{a,n} - i\omega_* \Psi_{a,n} + \frac{q\rho_{s,\mathrm{unit}}}{rG_r} a\left\{\hat{h}_a, \hat{\Psi}_a\right\} = C_a^{CL}\left[H_{a,n}\right]$$

IMEX time-integration scheme used to mitigate the electron parallel advection (ω_q) term should probably be avoided for ω_d term for physics reasons.

Curvature drift (ω_d) term is proportional to particle energy. EP requires small time steps.

Expensive cyclotron average and field solve:

$\Psi_{a,n}$ is a cyclotron-averaged composite field variable. EPs have large cyclotron orbits.



Some bottlenecks limit the largest simulations.

Cases involving an energetic particle (EP) species show such bottlenecks.

Stiff term in the gyrokinetic equation:

$$\frac{\partial h_{a,n}}{\partial t} - i\omega_\theta H_{a,n} - i\omega_d H_{a,n} - i\omega_E h_{a,n} - i\omega_* \Psi_{a,n} + \frac{q\rho_{s,\mathrm{unit}}}{rG_r} a\left\{\hat{h}_a, \hat{\Psi}_a\right\} = C_a^{CL}\left[H_{a,n}\right]$$

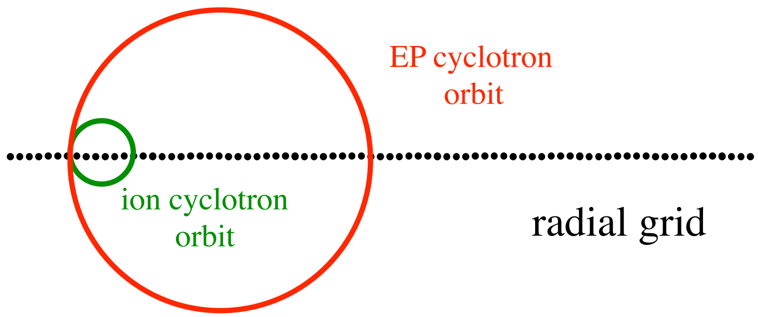
IMEX time-integration scheme used to mitigate the electron parallel advection (ω_q) term should probably be avoided for ω_d term for physics reasons.

Curvature drift (ω_d) term is proportional to particle energy. EP requires small time steps.

Expensive cyclotron average and field solve:

$\Psi_{a,n}$ is a cyclotron-averaged composite field variable. EPs have large cyclotron orbits.

Many more grid points are coupled and the $\Psi_{a,n}$ field-solve linear algebra becomes dense.



Some bottlenecks limit the largest simulations.

Cases involving an energetic particle (EP) species show such bottlenecks.

Stiff term in the gyrokinetic equation:

$$\frac{\partial h_{a,n}}{\partial t} - i\omega_\theta H_{a,n} - i\omega_d H_{a,n} - i\omega_E h_{a,n} - i\omega_* \Psi_{a,n} + \frac{q\rho_{s,\mathrm{unit}}}{rG_r} a\left\{\hat{h}_a, \hat{\Psi}_a\right\} = C_a^{CL}\left[H_{a,n}\right]$$

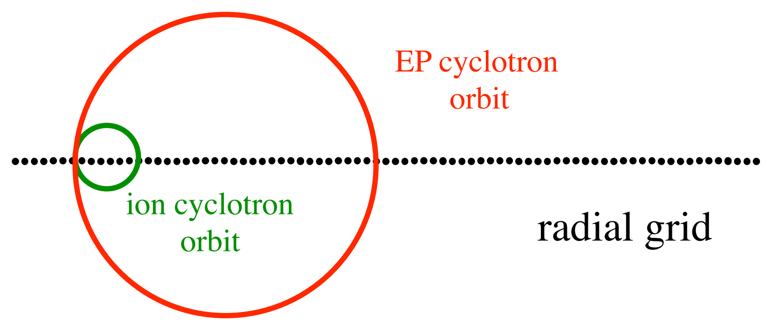
IMEX time-integration scheme used to mitigate the electron parallel advection (ω_q) term should probably be avoided for ω_d term for physics reasons.

Curvature drift (ω_d) term is proportional to particle energy. EP requires small time steps.

Expensive cyclotron average and field solve:

$\Psi_{a,n}$ is a cyclotron-averaged composite field variable. EPs have large cyclotron orbits.

Many more grid points are coupled and the $\Psi_{a,n}$ field-solve linear algebra becomes dense.



Dense field-solve linear algebra also occurs in multi-scale turbulence cases with a dense radial grid.

Workshop goals

Explore strategies to overcome speed bottlenecks:

Workshop goals

Explore strategies to overcome speed bottlenecks:

- Consider alternative to LAPACK linear algebra solvers for dense algebra

Workshop goals

Explore strategies to overcome speed bottlenecks:

- Consider alternative to LAPACK linear algebra solvers for dense algebra
- Develop a more clever way to set up the field-solve matrix in dense cases

Workshop goals

Explore strategies to overcome speed bottlenecks:

- Consider alternative to LAPACK linear algebra solvers for dense algebra
- Develop a more clever way to set up the field-solve matrix in dense cases
- **Possibly** implement a new IMEX time-integration scheme for new stiff terms

Workshop goals

Explore strategies to overcome speed bottlenecks:

- Consider alternative to LAPACK linear algebra solvers for dense algebra
- Develop a more clever way to set up the field-solve matrix in dense cases
- **Possibly** implement a new IMEX time-integration scheme for new stiff terms

Increase parallelizability:

Workshop goals

Explore strategies to overcome speed bottlenecks:

- Consider alternative to LAPACK linear algebra solvers for dense algebra
- Develop a more clever way to set up the field-solve matrix in dense cases
- **Possibly** implement a new IMEX time-integration scheme for new stiff terms

Increase parallelizability:

- Consider strategies to distribute the spatial grid (particularly the radial grid) or the separate kinetic species.

Workshop goals

Explore strategies to overcome speed bottlenecks:

- Consider alternative to LAPACK linear algebra solvers for dense algebra
- Develop a more clever way to set up the field-solve matrix in dense cases
- **Possibly** implement a new IMEX time-integration scheme for new stiff terms

Increase parallelizability:

- Consider strategies to distribute the spatial grid (particularly the radial grid) or the separate kinetic species.
- For eigenvalue-solving mode, distribute the time-evolution matrix over a greater number of processors (for use by SLEPc) than that suggested by the existing gyro parallelization scheme.