Implicit-Explicit Time Integration for Multiscale Physics

Geophysical and Astrophysical Fluid Dynamics Seminar,
Department of Applied Mathematics & Statistics, UC Santa Cruz

Debojyoti Ghosh
Center for Applied Scientific Computing

December 6, 2017

LLNL-PRES-742278
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

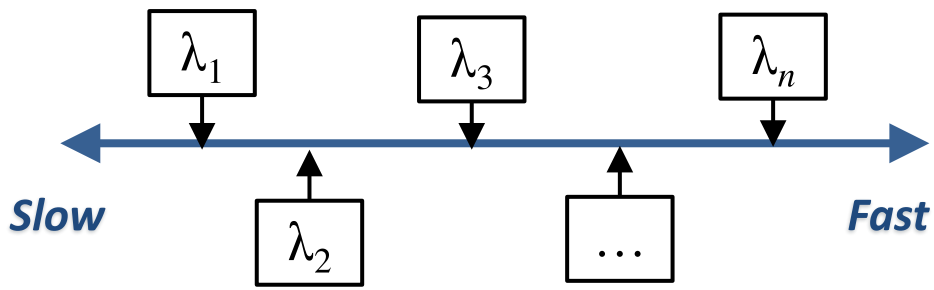
Time Scales and Numerical Time Integration

Motivation for Implicit-Explicit (IMEX) Approach

Complex physics are characterized by a **large range of temporal scales**

Model ODE in time

$$\frac{dy}{dt} = \lambda_1 y + \lambda_2 y + \cdots + \lambda_n y; \qquad \lambda_i \in \mathbb{Z}, \quad \lambda_1 < \lambda_2 < \cdots < \lambda_n$$



Explicit time-integration constrained by *fastest time scale in the model*

- *Inefficient when resolving slow dynamics*
- ***Split-Explicit methods*** *(atmospheric flow simulations)*

Implicit time-integration requires solution to *nonlinear system of equations*

- *Unconditional stability*
- *Why pay for inverting terms we want to resolve?*

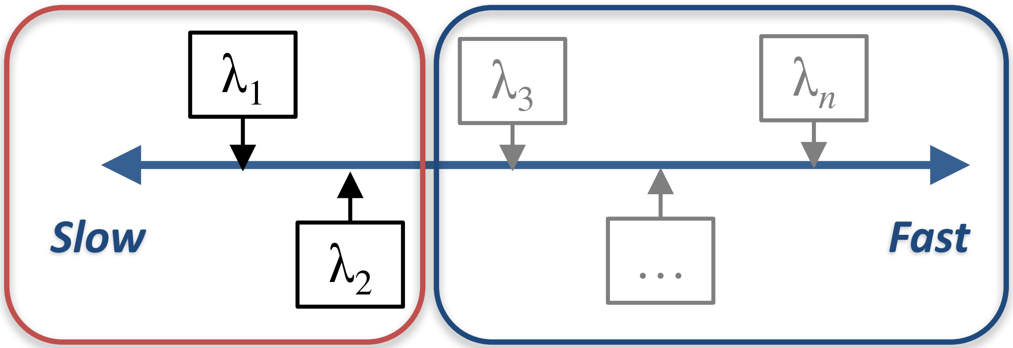
Which time scales do we want to resolve? *(Usually, some of them)*

Implicit-Explicit (IMEX) Time Integration

Resolve scales of interest; Treat implicitly faster scales

Time scales of interest
(nonstiff terms)

Explicit time integration



Faster time scales
(stiff terms)

Implicit time integration

ODE in time

Resulting from spatial discretization of PDE

$$\frac{d\mathbf{y}}{dt} = \mathcal{R}\left(\mathbf{y}\right)$$

IMEX time integration: *partition RHS*

$$\mathcal{R}\left(\mathbf{y}\right) = \mathcal{R}_{\text{stiff}}\left(\mathbf{y}\right) + \mathcal{R}_{\text{nonstiff}}\left(\mathbf{y}\right)$$

Linear stability constraint on time step

$$\Delta t \left(\lambda \left[\frac{d\mathcal{R}_{\text{nonstiff}}\left(\mathbf{y}\right)}{d\mathbf{y}} \right] \right) \in \{ z : |R(z)| \leq 1 \}$$

Time step constrained by eigenvalues (time scales) of ***nonstiff component of RHS***

Additive Runge-Kutta (ARK) Time Integrators
Multistage, high-order, conservative IMEX methods

Butcher tableaux representation

Explicit RK

$$\begin{array}{c|ccccc} 0 & 0 & & & & \\ c_2 & a_{21} & 0 & & & \\ \vdots & \vdots & \ddots & 0 & & \\ c_s & a_{s1} & \cdots & a_{s,s-1} & 0 \\ \hline & b_1 & \cdots & \cdots & b_s \end{array}$$

$+$

DIRK

$$\begin{array}{c|ccccc} 0 & 0 & & & & \\ \tilde{c}_2 & \tilde{a}_{21} & \gamma & & & \\ \vdots & \vdots & \ddots & \gamma & & \\ \tilde{c}_s & \tilde{a}_{s1} & \cdots & \tilde{a}_{s,s-1} & \gamma \\ \hline & b_1 & \cdots & \cdots & b_s \end{array}$$

Time step: From t_n to $t_{n+1} = t_n + \Delta t$ $\quad s \rightarrow$ number of stages

Stage solutions

$$\mathbf{y}^{(i)} = \mathbf{y}_n + \Delta t \sum_{j=1}^{i-1} a_{ij} \mathcal{R}_{\text{nonstiff}} \left(\mathbf{y}^{(j)} \right) + \Delta t \sum_{j=1}^{i} \tilde{a}_{ij} \mathcal{R}_{\text{stiff}} \left(\mathbf{y}^{(j)} \right), \ i = 1, \cdots, s$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \sum_{i=1}^{s} b_i \mathcal{R} \left(\mathbf{y}^{(i)} \right)$$ *Step completion*

Kennedy & Carpenter, J. Comput. Phys., 2003

Implicit Stage Solution

Requires solving nonlinear system of equations

Rearranging the stage solution expression:

$$\underbrace{\frac{1}{\Delta t \tilde{a}_{ii}}\mathbf{y}^{(i)} - \mathcal{R}_{\text{stiff}}\left(\mathbf{y}^{(i)}\right) - \left[\mathbf{y}_n + \Delta t \sum_{j=1}^{i-1}\left\{a_{ij}\mathcal{R}_{\text{nonstiff}}\left(\mathbf{y}^{(j)}\right) + \tilde{a}_{ij}\mathcal{R}_{\text{stiff}}\left(\mathbf{y}^{(j)}\right)\right\}\right] = 0}_{\mathcal{F}(y) = 0}$$

Jacobian-free Newton-Krylov method (*Knoll & Keyes, J. Comput. Phys., 2004*):

Initial guess: $y_0 \equiv \mathbf{y}_0^{(i)} = \mathbf{y}^{(i-1)}$

Newton update: $y_{k+1} = y_k - \mathcal{J}(y_k)^{-1}\mathcal{F}(y_k)$

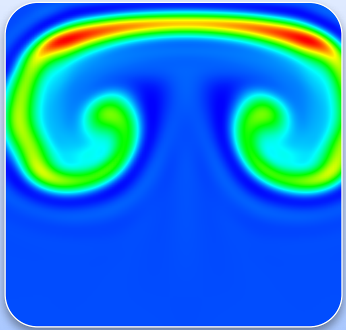
GMRES solver (preconditioned)

$$\mathcal{J}(y_k)\,\Delta y = \mathcal{F}(y_k)$$

Action of the Jacobian on a vector approximated by *directional derivative*

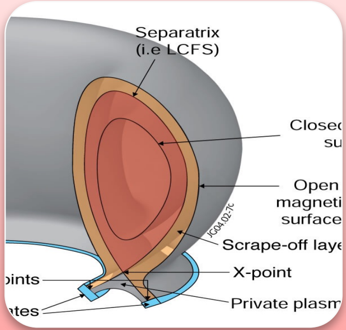
$$\mathcal{J}(y_k)\,x = \left.\frac{d\mathcal{F}(y)}{dy}\right|_{y_k} x \approx \frac{1}{\epsilon}\left[\mathcal{F}(y_k + \epsilon x) - \mathcal{F}(y_k)\right]$$

Applications



Atmospheric Flows

- Speed of sound much faster than dynamics of interest
- At ANL (with *Emil Constantinescu*)
- 2013 – 2015



Tokamak Edge Plasma Dynamics

- Multiscale dynamics at the edge region
- At LLNL (with *Milo Dorr, Mikhail Dorf, Jeff Hittinger*)
- 2015 – Present

Challenges in Atmospheric Flow Simulations

Limited-area and ***mesoscale*** simulations require a ***nonhydrostatic model***

Nonhydrostatic model introduces the **acoustic mode**

- Sound waves *much faster than flow velocities*
- ***Insignificant effect*** on atmospheric phenomena

Multiscale time integration

IMEX time integrators have been **applied to atmospheric flows**

Horizontal-Explicit, Vertical-Implicit Methods

- Simulation domains are much larger horizontally than vertically
- Grids are typically *much finer* along the vertical (z) axis
- Terms with ***z-derivatives integrated implicitly***, remaining terms integrated explicitly

Flux-Partitioned Methods

- Right-hand-side partitioned into linear stiff and nonlinear nonstiff components
- Formulation based on perturbations to the hydrostatic balance
- ***First-order perturbations treated implicitly***; higher-order perturbations treated explicitly.

Objectives

Develop a conservative atmospheric flow solver

Compressible Euler equations *(mass, momentum, energy)*

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ e\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ (e+p)u\end{bmatrix}+\frac{\partial}{\partial y}\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ (e+p)v\end{bmatrix}=\begin{bmatrix}0\\ \rho\mathbf{g}\cdot\hat{\mathbf{i}}\\ \rho\mathbf{g}\cdot\hat{\mathbf{j}}\\ \rho u\mathbf{g}\cdot\hat{\mathbf{i}}+\rho v\mathbf{g}\cdot\hat{\mathbf{j}}\end{bmatrix}$$

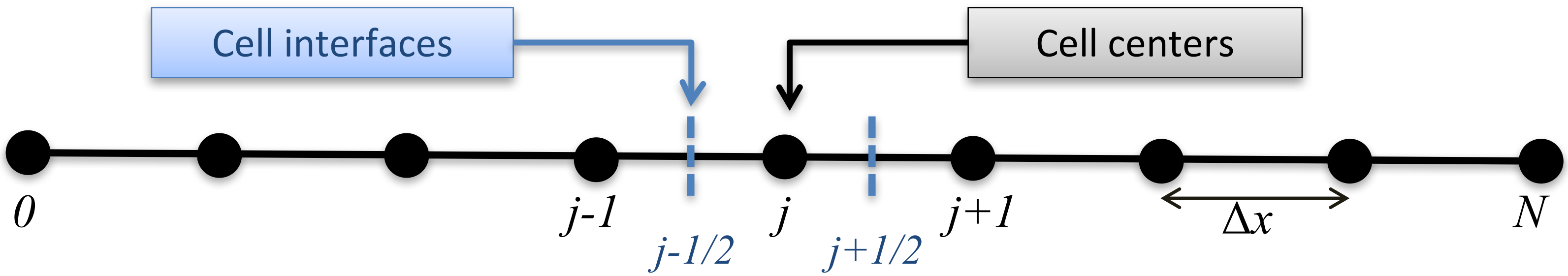
Forms of the governing equations in the literature:

- Expressed in terms of **Exner pressure** and **potential temperature**
  - Mass, momentum, energy not conserved
  - Examples: COAMPS – US Navy, NMM – NCEP, MM5 – NCAR/PSU).
- Conservation of mass and momentum; energy equation expressed as **conservation of potential temperature** (adiabatic assumption)
  - Energy not conserved to machine precision
  - True viscous terms cannot be prescribed if needed
  - Examples: WRF – NCAR, NUMA – NPS.

Slow-fast flux partitioning exist for these formulations

Derive a characteristic-based flux-partitioning for the Euler equations

Conservative Finite-Difference Schemes



Conservative finite-difference discretization of a 1D hyperbolic conservation law:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{1}{\Delta x}\left(\mathbf{h}_{j+\frac{1}{2}} - \mathbf{h}_{j-\frac{1}{2}}\right) = 0 \qquad \mathbf{f}(\mathbf{u}(x)) = \frac{1}{\Delta x}\int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} \mathbf{h}(\mathbf{u}(\xi))\, d\xi$$

$$\hat{\mathbf{f}}_{j+\frac{1}{2}} = \mathbf{h}\left(\mathbf{u}\left(x_{j+\frac{1}{2}}\right)\right) + \mathcal{O}\left(\Delta x^{p}\right)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{1}{\Delta x}\left(\hat{\mathbf{f}}_{j+\frac{1}{2}} - \hat{\mathbf{f}}_{j-\frac{1}{2}}\right) = 0$$

Spatially-discretized ODE in time

5th order WENO
(*Jiang & Shu, J. Comput. Phys., 1996*)

5th order CRWENO
(*Ghosh & Baeder, SIAM J. Sci. Comput., 2012*)

Characteristic-based Flux Partitioning (1)

1D Euler equations

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{u})}{\partial x} = 0$$

Spatial discretization

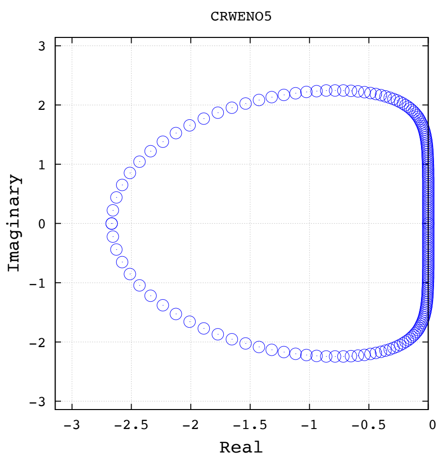
Semi-discrete ODE in time

$$\frac{\partial \mathbf{u}}{\partial t} = \hat{\mathbf{F}}(\mathbf{u}) = \left[\mathcal{D} \otimes \mathcal{A}(u)\right] \mathbf{u}$$

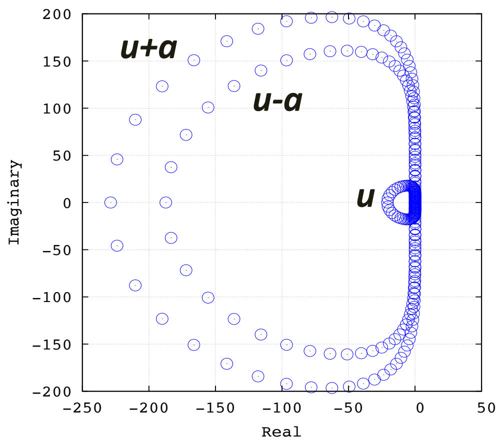
Discretization operator (e.g.:WENO5, CRWENO5)

Flux Jacobian

Example: Periodic density sine wave on a unit domain discretized by *N*=80 points.



Eigenvalues of the CRWENO5 discretization



Eigenvalues of the right-hand-side operator
(u=0.1, a=1.0, dx=0.0125)

$$\text{eig}\left[\frac{\partial \hat{\mathbf{F}}}{\partial \mathbf{u}}\right] = \text{eig}\left[\mathcal{D}\right] \times \text{eig}\left[\mathcal{A}(\mathbf{u})\right]$$

Time step size limit for linear stability

Eigenvalues of the right-hand-side of the ODE are the **eigenvalues of the discretization operator** times the **characteristic speeds** of the physical system

Characteristic-based Flux Partitioning (2)

Splitting of the **flux Jacobian** based on its eigenvalues

$$\frac{\partial \mathbf{u}}{\partial t} = \hat{\mathbf{F}}(\mathbf{u}) = \left[\mathcal{D} \otimes \mathcal{A}(u)\right]\mathbf{u}$$
$$= \left[\mathcal{D} \otimes \mathcal{A}_S(u) + \mathcal{D} \otimes \mathcal{A}_F(u)\right]\mathbf{u}$$
$$= \hat{\mathbf{F}}_S(\mathbf{u}) + \hat{\mathbf{F}}_F(\mathbf{u})$$

"Slow" flux **"Fast" Flux**

$$\mathcal{A}(\mathbf{u}) = \mathcal{R}\Lambda\mathcal{L}$$
$$= \mathcal{R}\Lambda_S\mathcal{L} + \mathcal{R}\Lambda_F\mathcal{L}$$
$$= \mathcal{A}_S(\mathbf{u}) + \mathcal{A}_{\mathcal{F}}(\mathbf{u})$$

$$\Lambda_S = \begin{bmatrix} u & & \\ & 0 & \\ & & 0 \end{bmatrix}$$

$$\Lambda_F = \begin{bmatrix} 0 & & \\ & u+a & \\ & & u-a \end{bmatrix}$$

$$\mathbf{f}_S(\mathbf{u}) = \begin{bmatrix} \left(\frac{\gamma-1}{\gamma}\right)\rho u \\ \left(\frac{\gamma-1}{\gamma}\right)\rho u^2 \\ \frac{1}{2}\left(\frac{\gamma-1}{\gamma}\right)\rho u^3 \end{bmatrix}$$

Convective flux (slow)

Acoustic flux (fast)

$$\mathbf{f}_F(\mathbf{u}) = \begin{bmatrix} \left(\frac{1}{\gamma}\right)\rho u \\ \left(\frac{1}{\gamma}\right)\rho u^2 + p \\ (e+p)u - \frac{1}{2}\left(\frac{\gamma-1}{\gamma}\right)\rho u^3 \end{bmatrix}$$

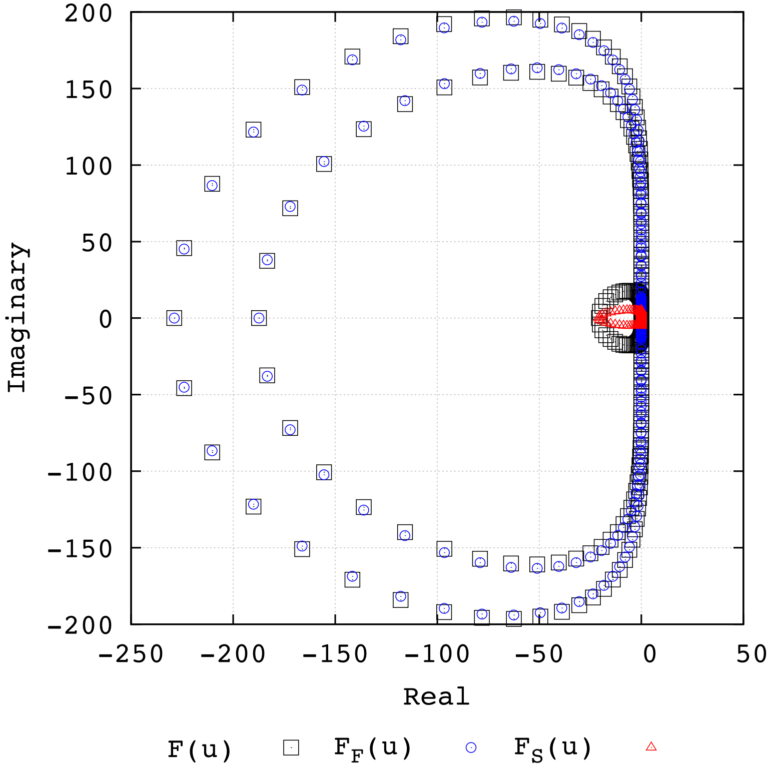
Characteristic-based Flux Partitioning (3)

Example: Periodic density sine wave on a unit domain discretized by *N*=80 points (CRWENO5).

$$\frac{\partial \mathbf{F}_{S,F}(\mathbf{u})}{\partial \mathbf{u}} \neq [\mathcal{A}_{S,F}]$$

Small difference between the eigenvalues of the complete operator and the split operator.

(Not an error)



$$\text{eig}\left[\frac{\partial \hat{\mathbf{F}}_S}{\partial \mathbf{u}}\right] \approx u \times \text{eig}\left[\mathcal{D}\right] \quad \text{eig}\left[\frac{\partial \hat{\mathbf{F}}_F}{\partial \mathbf{u}}\right] \approx \{u \pm a\} \times \text{eig}\left[\mathcal{D}\right]$$

IMEX Time Integration with Characteristic-based Flux Partitioning (1)

Apply **Additive Runge-Kutta (ARK)** time-integrators to the split form

Stage values (*s* stages)

$$\mathbf{U}^{(i)} = \mathbf{u}_n + \Delta t \sum_{j=1}^{i-1} a_{ij} \hat{\mathbf{F}}_S \left(\mathbf{U}^{(j)} \right) + \Delta t \sum_{j=1}^{i} \tilde{a}_{ij} \hat{\mathbf{F}}_F \left(\mathbf{U}^{(j)} \right)$$

$$i = 1, \cdots, s$$

Step completion

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t \sum_{i=1}^{s} b_i \hat{\mathbf{F}}_S \left(\mathbf{U}^{(i)} \right) + \Delta t \sum_{i=1}^{s} \tilde{b}_i \hat{\mathbf{F}}_F \left(\mathbf{U}^{(i)} \right)$$

Non-linear system of equations

$$\hat{\mathbf{F}}_F (\mathbf{u}) = \left[\mathcal{D}(\omega) \otimes \mathcal{A}_F (\mathbf{u}) \right] \mathbf{u}$$

Solution-dependent weights for the WENO5/CRWENO5 scheme

$$\omega = \omega \left[\mathbf{F}(\mathbf{u}) \right]$$

Nonlinear flux

Linearization of Flux Partitioning

Redefine the splitting as

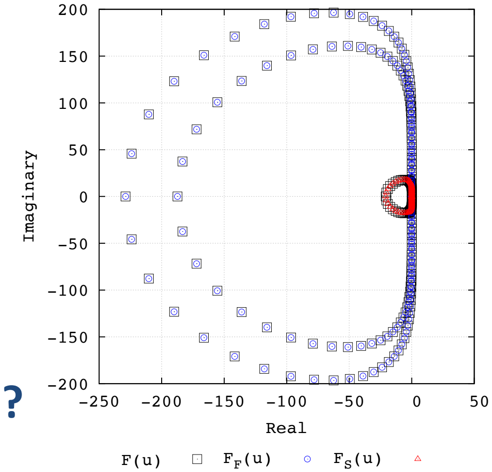
$$\mathbf{F}_F(\mathbf{u}) = [\mathcal{A}_F(\mathbf{u}_n)]\,\mathbf{u}$$
$$\mathbf{F}_S(\mathbf{u}) = \mathbf{F}(\mathbf{u}) - \mathbf{F}_F(\mathbf{u})$$

Note: Introduces **no error** in the governing equation.

At the beginning of a time step:-

$$\text{eig}\left[\frac{\partial \hat{\mathbf{F}}_S}{\partial \mathbf{u}}\right] = u \times \text{eig}\left[\mathcal{D}\right]$$

$$\text{eig}\left[\frac{\partial \hat{\mathbf{F}}_F}{\partial \mathbf{u}}\right] = \{u \pm a\} \times \text{eig}\left[\mathcal{D}\right]$$



Is F_F a good approximation at each stage?

Linearization of the WENO/CRWENO discretization:

$$\omega\left[\mathbf{F}(\mathbf{u}_n)\right]$$
$$\mathbf{U}^{(1)} = \mathbf{u}^n$$
$$\omega\left[\mathbf{F}\left(\mathbf{U}^{(1)}\right)\right]$$
$$\mathbf{U}^{(2)} = \mathbf{u}^n + \Delta t \tilde{a}_{21} \hat{\mathbf{F}}_F\left(\mathbf{U}^{(1)}\right) + \Delta t \tilde{a}_{22} \hat{\mathbf{F}}_F\left(\mathbf{U}^{(2)}\right) + \Delta t a_{21} \hat{\mathbf{F}}_S\left(\mathbf{U}^{(1)}\right)$$
$$\omega\left[\mathbf{F}\left(\mathbf{U}^{(2)}\right)\right]$$
$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \tilde{b}_1 \hat{\mathbf{F}}_F\left(\mathbf{U}^{(1)}\right) + \Delta t \tilde{b}_2 \hat{\mathbf{F}}_F\left(\mathbf{U}^{(2)}\right) + \Delta t b_1 \hat{\mathbf{F}}_S\left(\mathbf{U}^{(1)}\right) + \Delta t b_2 \hat{\mathbf{F}}_S\left(\mathbf{U}^{(2)}\right)$$

Within a stage, the non-linear weights are kept fixed.

Example: 2-stage ARK method

IMEX Time Integration with Characteristic-based Flux Partitioning (2)

Linear system of equations for implicit stages:

$$\left[\mathcal{I} - \Delta t \tilde{a}_{ii} \mathcal{D} \otimes \mathcal{A}_F(\mathbf{u_n})\right] \mathbf{U}^{(i)} = \mathbf{u}_n + \Delta t \sum_{j=1}^{i-1} a_{ij} \hat{\mathbf{F}}_S\left(\mathbf{U}^{(j)}\right) + \Delta t \left[\mathcal{D} \otimes \mathcal{A}_F(\mathbf{u_n})\right] \sum_{j=1}^{i-1} \tilde{a}_{ij} \mathbf{U}^{(j)},$$

$$i = 1, \cdots, s$$

Preconditioning (Preliminary attempts)

$$\mathcal{P} = \left[\mathcal{I} - \Delta t \tilde{a}_{ii} \mathcal{D}^{(1)} \otimes \mathcal{A}_F(\mathbf{u_n})\right] \approx \left[\mathcal{I} - \Delta t \tilde{a}_{ii} \mathcal{D} \otimes \mathcal{A}_F(\mathbf{u_n})\right]$$

First order upwind discretization
Periodic boundaries ignored

Block n-diagonal matrices
- Block tri-diagonal (1D)
- Block penta-diagonal (2D)
- Block septa-diagonal (3D)

- **Jacobian-free approach** → Linear Jacobian defined as a function describing its action on a vector
- **Preconditioning matrix** → Stored as a sparse matrix

ARK Methods (PETSc)

ARKIMEX 2c
- 2nd order accurate
- 3 stage (1 explicit, 2 implicit)
- L-Stable implicit part
- Large real stability of explicit part

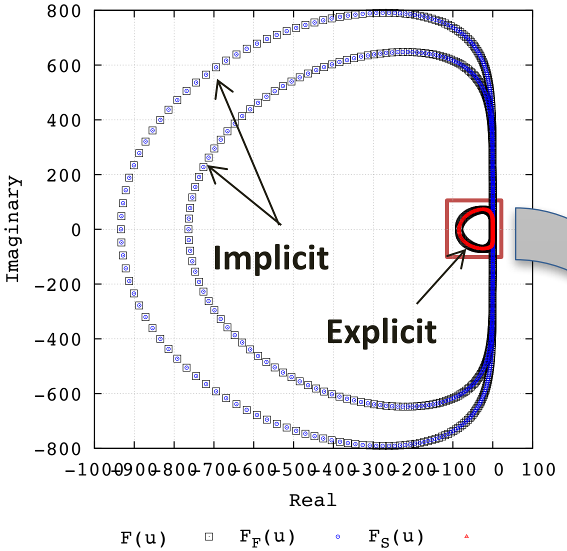
ARKIMEX 2e
- 2nd order accurate
- 3 stage (1 explicit, 2 implicit)
- L-Stable implicit part

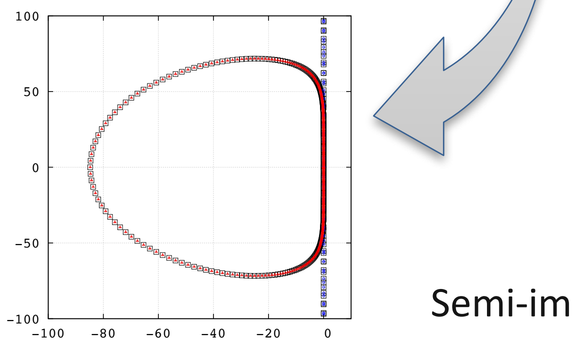
ARKIMEX 3
- 3rd order accurate
- 4 stage (1 explicit, 3 implicit)
- L-Stable implicit part

ARKIMEX 4
- 4th order accurate
- 5 stage (1 explicit, 4 implicit)
- L-Stable implicit part

Example: 1D Density Wave Advection ($M_\infty = 0.1$)

Eigenvalues



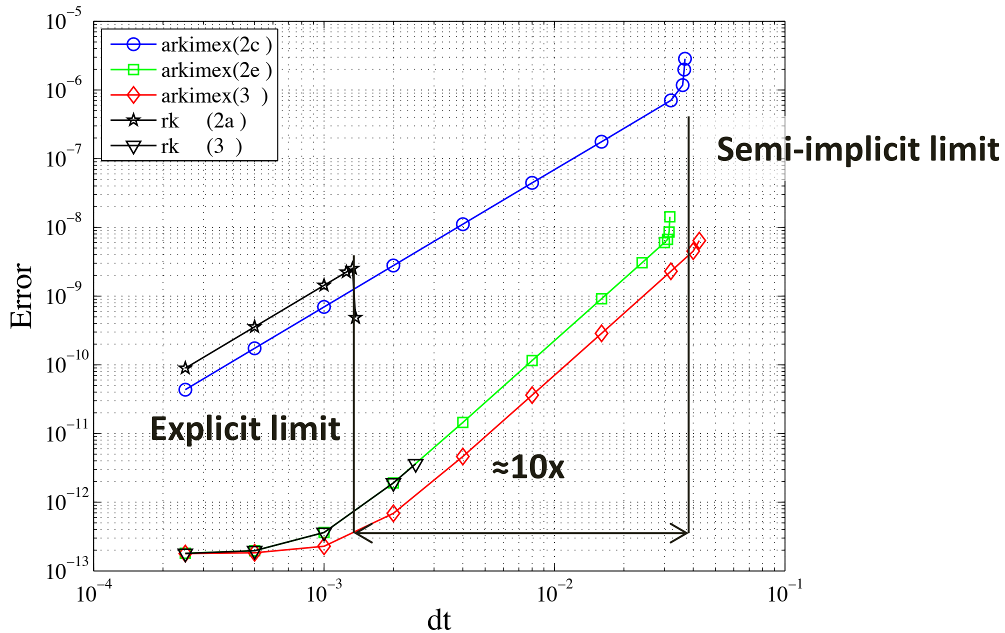


Initial solution

$$0 \le x \le 1$$

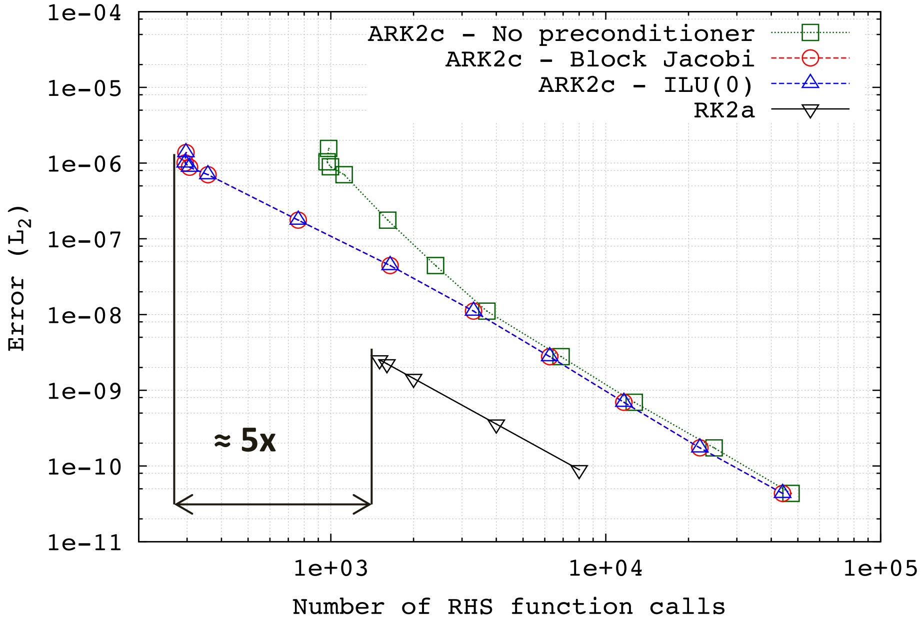
$$\rho = \rho_\infty + \hat{\rho} \sin(2\pi x)$$
$$u = u_\infty, p = p_\infty$$

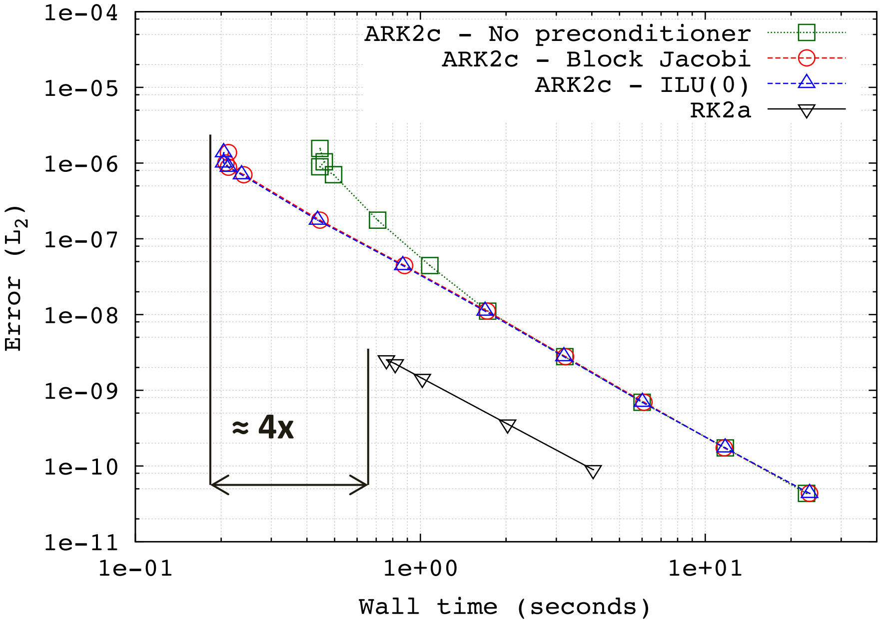
CRWENO5, 320 grid points



Semi-implicit time step size limit **$1/M_\infty$** than explicit time step size limit

Example: 1D Density Wave Advection ($M_\infty = 0.1$) Computational Cost





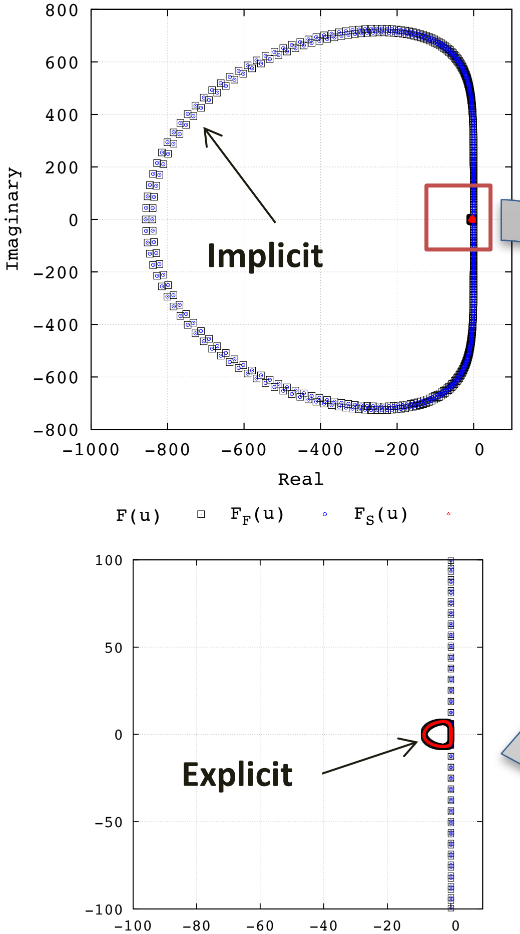
Number of function calls

Wall time

Number of function calls = (Number of time steps × number of stages) + Number of GMRES iterations (does not reflect cost of constructing preconditioning matrix and inverting it)

Example: 1D Density Wave Advection ($M_\infty = 0.01$)

Eigenvalues



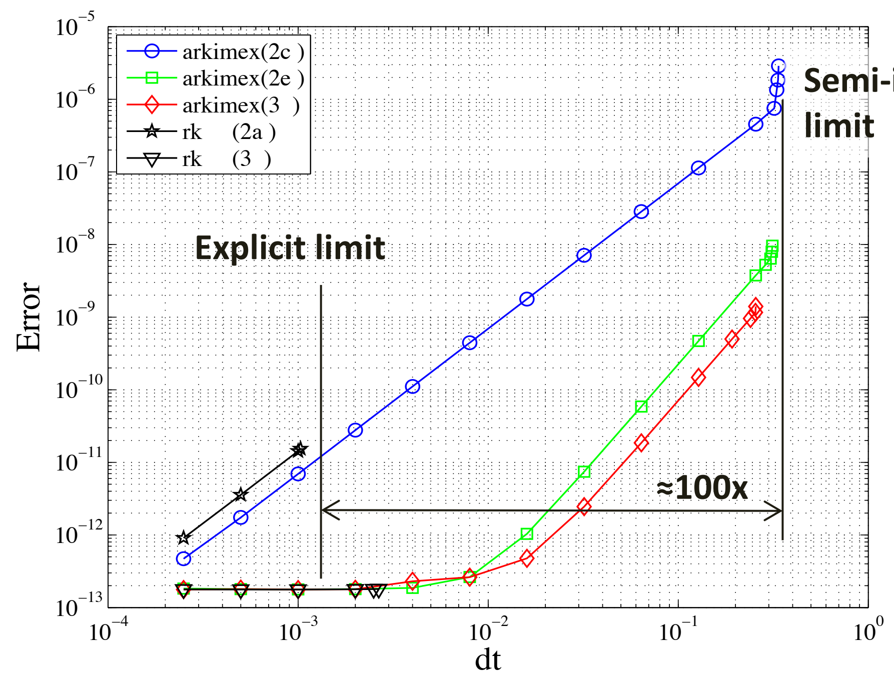
Initial solution

$$0 \leq x \leq 1$$

$$\rho = \rho_\infty + \hat{\rho}\sin(2\pi x)$$

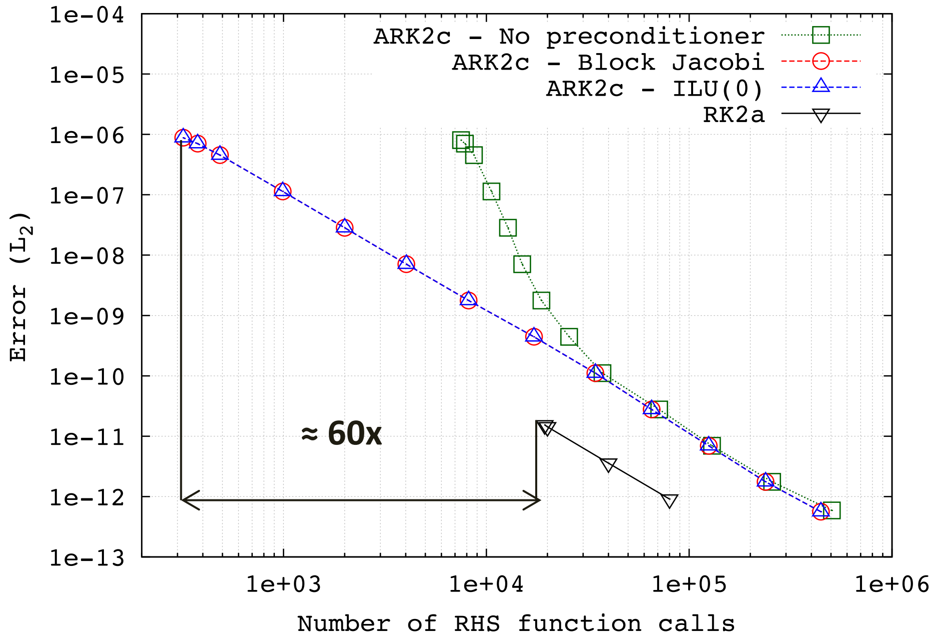
$$u = u_\infty, p = p_\infty$$

CRWENO5, 320 grid points

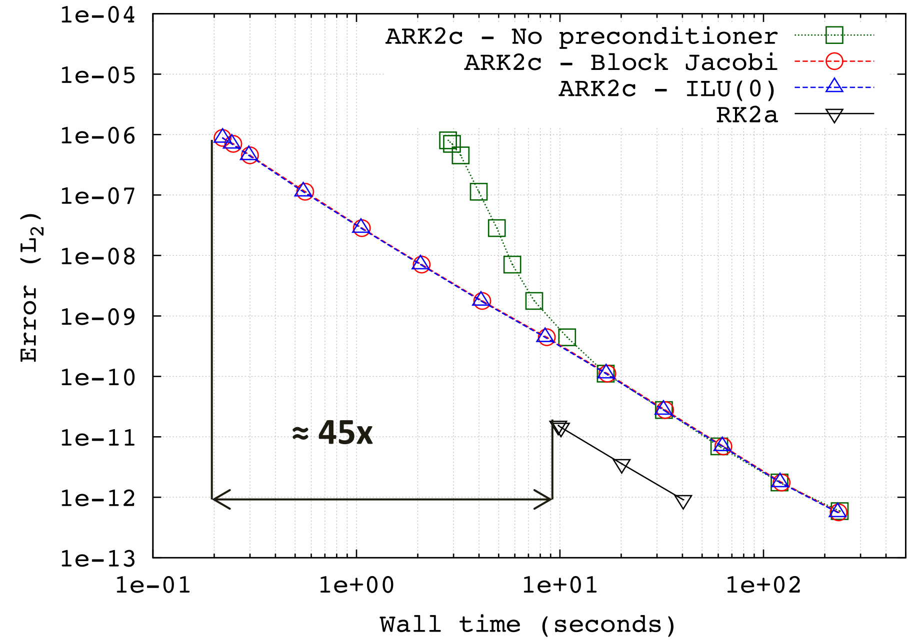


Semi-implicit time step size limit **1/ M_∞** than explicit time step size limit

Example: 1D Density Wave Advection ($M_\infty = 0.01$) Computational Cost



Number of function calls



Wall time

Number of function calls = (Number of time steps × number of stages) + Number of GMRES iterations (does not reflect cost of constructing preconditioning matrix and inverting it)

Example: 2D Low Mach Isentropic Vortex Convection

Freestream flow

$$\left.\begin{aligned} \rho_\infty &= 1 \\ p_\infty &= 1 \\ u_\infty &= 0.1 \\ v_\infty &= 0 \end{aligned}\right\} M_\infty \approx 0.08$$

Vortex (Strength *b = 0.5*)

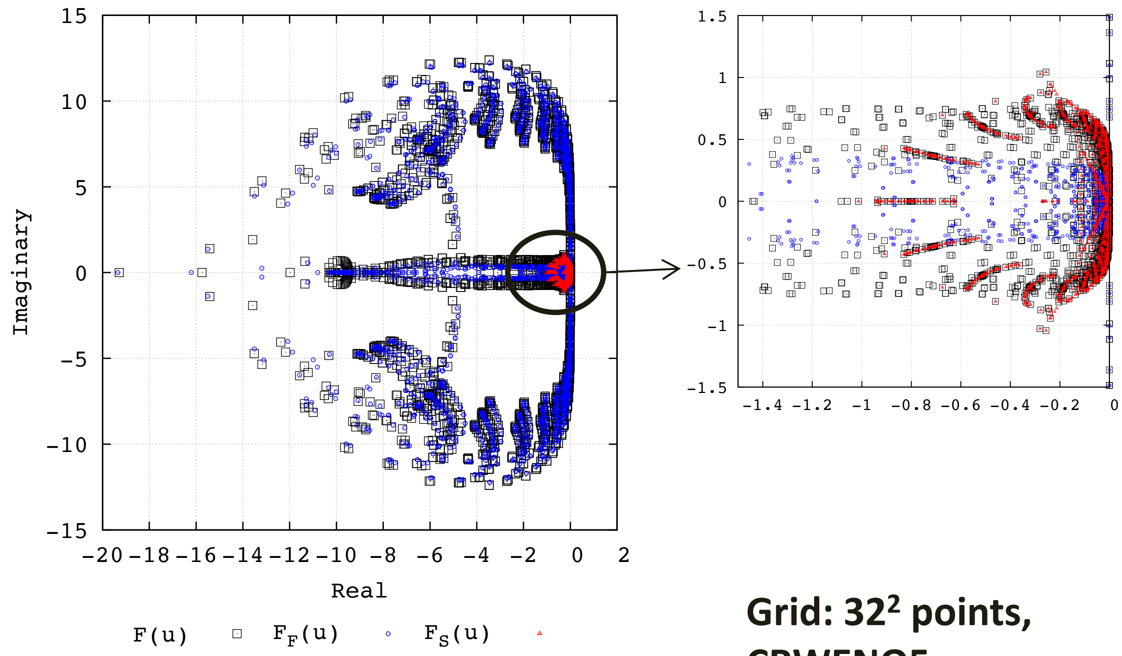
$$\rho = \left[1.0 - \frac{(\gamma-1)\,b^2}{8\gamma\pi^2}\exp\left(1-r^2\right)\right]^{\frac{1}{\gamma-1}}$$

$$p = \left[1.0 - \frac{(\gamma-1)\,b^2}{8\gamma\pi^2}\exp\left(1-r^2\right)\right]^{\frac{\gamma}{\gamma-1}}$$

$$u = u_\infty - \frac{b}{2\pi}\exp\left(\frac{1-r^2}{2}\right)(y-y_c)$$

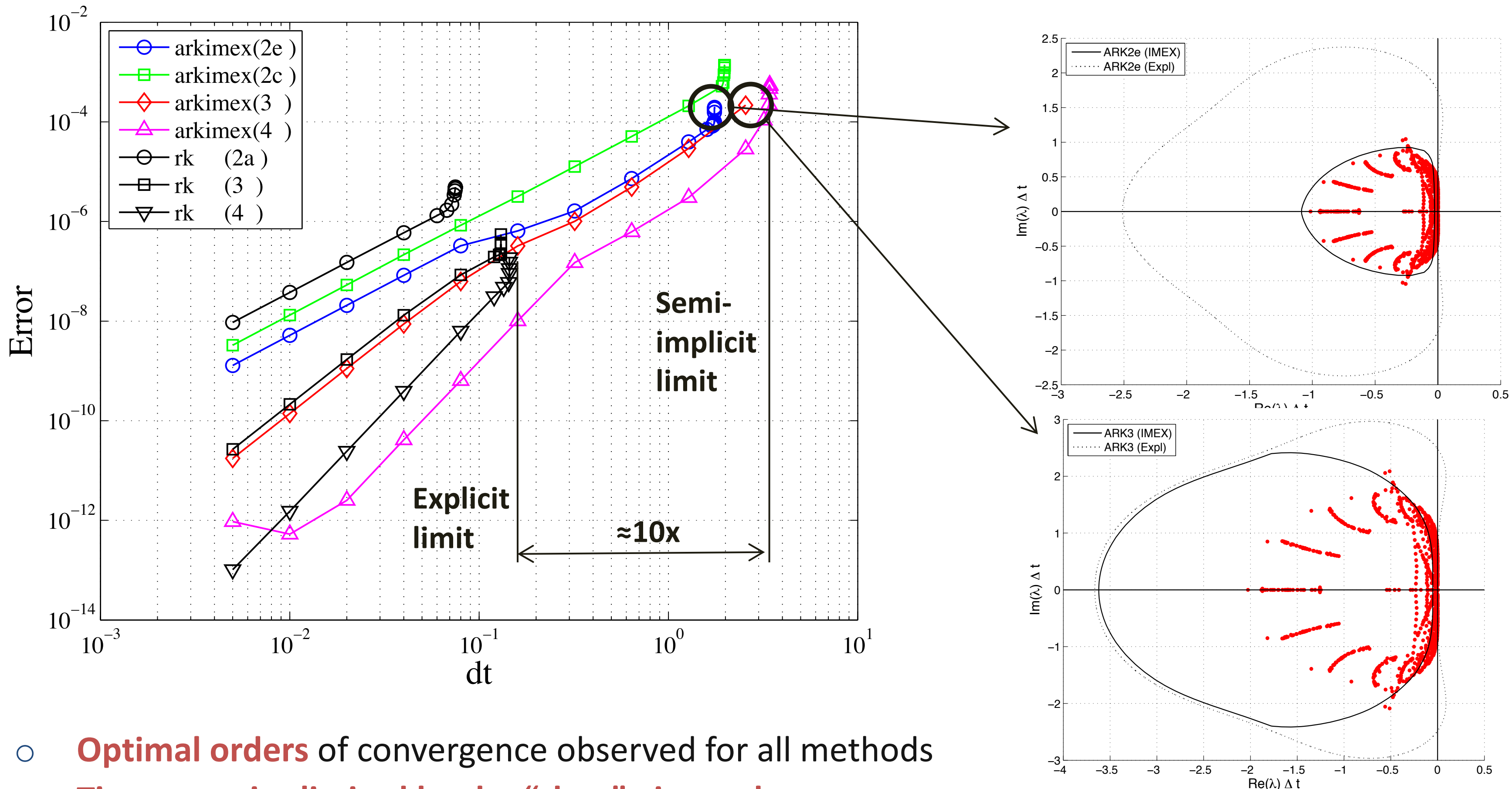
$$v = v_\infty + \frac{b}{2\pi}\exp\left(\frac{1-r^2}{2}\right)(x-x_c)$$

Eigenvalues of the right-hand-side operators



Grid: 32^2 points, CRWENO5

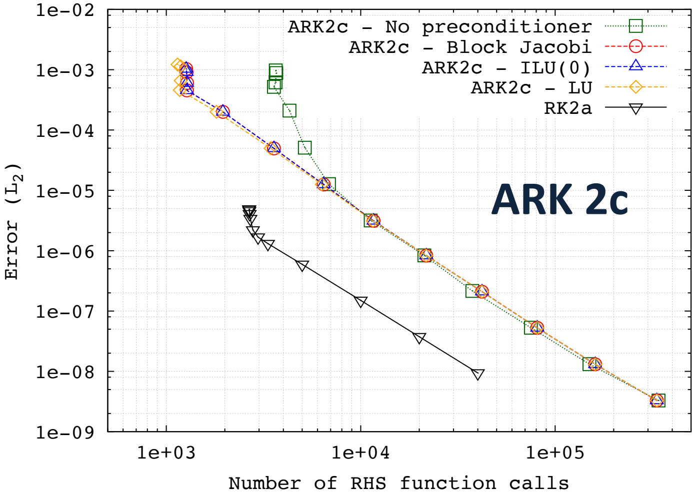
Example: 2D Low Mach Isentropic Vortex Convection

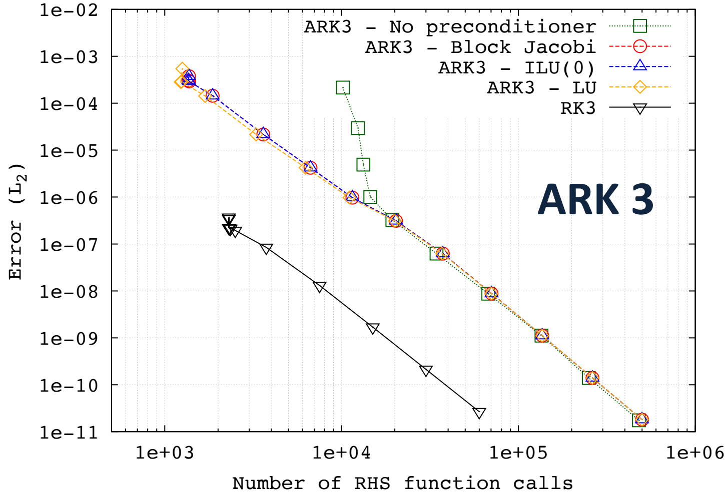


- **Optimal orders** of convergence observed for all methods
- **Time step size limited by the "slow" eigenvalues.**

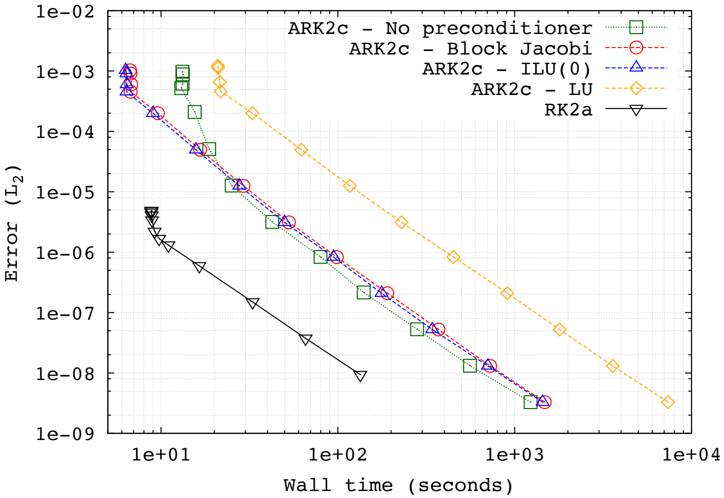
Example: Vortex Convection (Computational Cost)

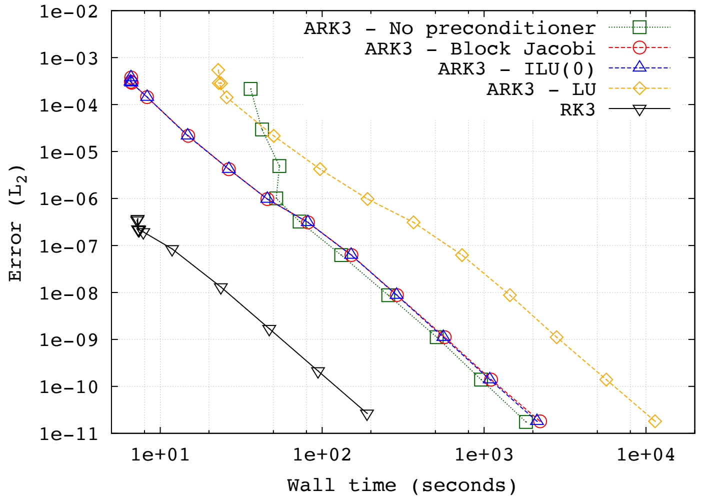
Number of function calls





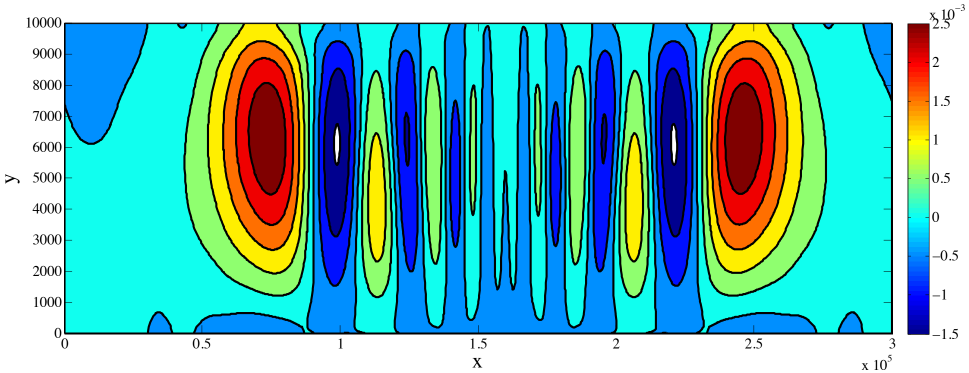
Wall time





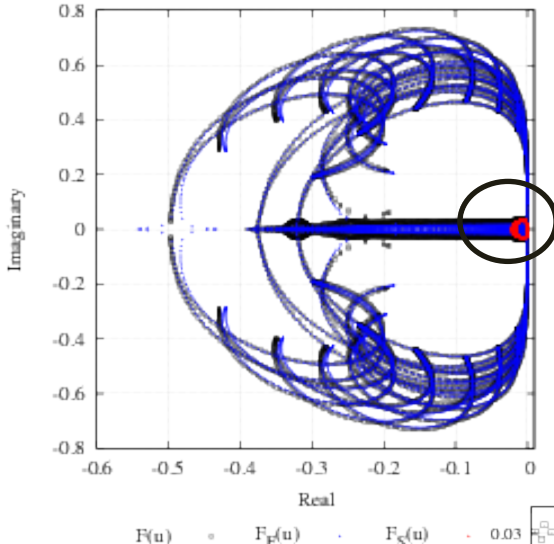
Example: Inertia – Gravity Wave

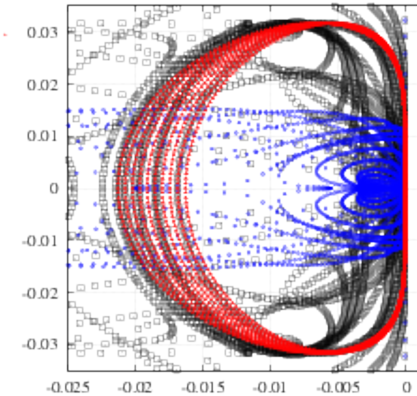
- **Periodic channel** – 300 km x 10 km
- **No-flux boundary conditions** at top and bottom boundaries
- **Mean horizontal velocity** of 20 m/s in a uniformly stratified atmosphere **($M_\infty \approx 0.06$)**
- Initial solution – **Potential temperature perturbation**



Potential temperature perturbations at 3000 seconds (Solution obtained with **WENO5** and **ARKIMEX 2e**, 1200x50 grid points)

Eigenvalues of the right-hand-side operators

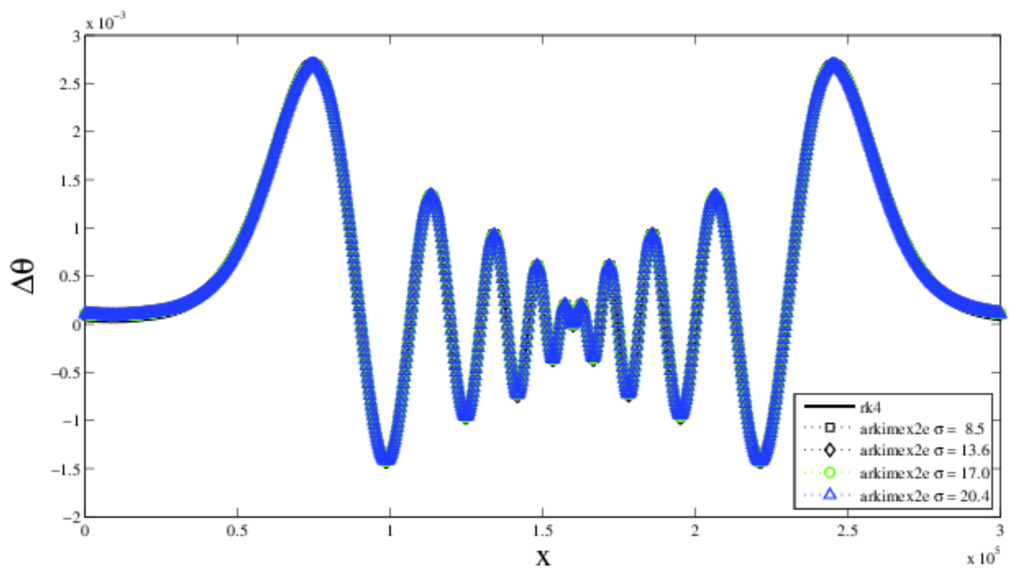




Grid: 300x10 points, CRWENO5

Example: Inertia – Gravity Wave

CFL	Wall time		Function counts	
	Absolute (s)	Normalized (/RK4)	Absolute	Normalized (/RK4)
8.5	6,149	1.14	24,800	1.03
13.6	4,118	0.76	17,457	0.73
17.0	3,492	0.65	14,820	0.62
20.4	**2,934**	**0.54**	**12,895**	**0.54**



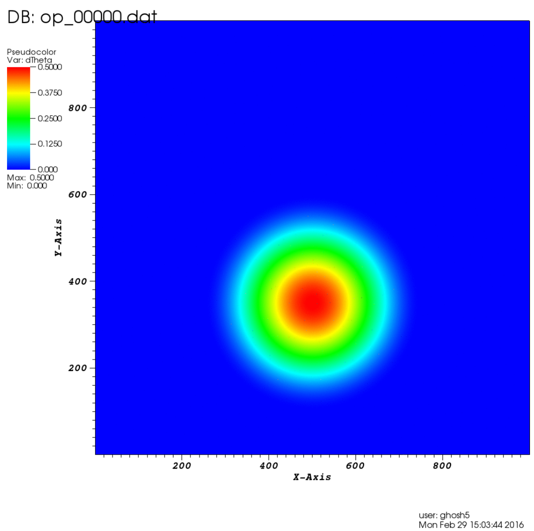
Fastest RK4

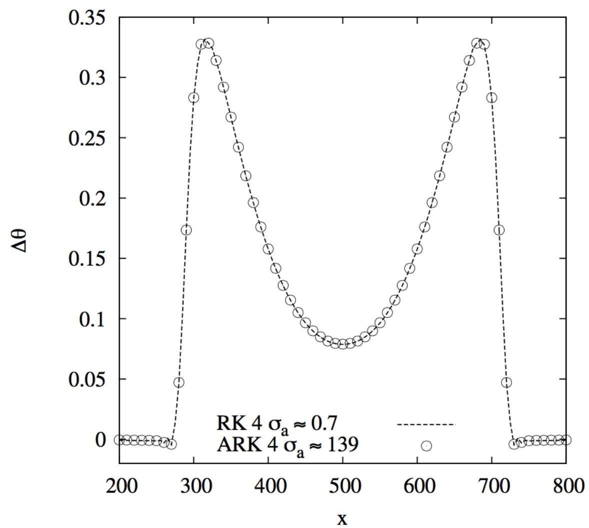
CFL ~ **1.0,** Wall time: **5400 s**
Function counts: **24000**

Cross-sectional potential temperature perturbations at 3000 seconds (y = 5 km) at CFL numbers **0.2 – 13.6**

Example: Rising Thermal Bubble

CFL	Wall time		Function counts	
	Absolute (s)	Normalized (/RK4)	Absolute	Normalized (/RK4)
6.9	73,111	2.42	360,016	2.25
34.7	22,104	0.73	111,824	0.70
138.9	**8,569**	**0.28**	**45,969**	**0.29**





Fastest RK4

CFL ~ **0.7,** Wall time: **30,154 s**

Function counts: **160,000**

Summary (Atmospheric Flows)

Characteristic-based flux splitting:

- Partitioning of flux **separates the acoustic and entropy modes** → Allows **larger time step sizes** (determined by flow velocity, not speed of sound).
- **Comparison** to alternatives
  - **Vs. explicit time integration:** Larger time steps → More efficient algorithm
  - **Vs. implicit time integration:** Semi-implicit solves a linear system **without any approximations** to the overall governing equations (as opposed to: solve non-linear system of equations or linearize governing equations in a time step).

Future work:

- **Improve efficiency** of the **linear solve**
  - Better preconditioning of the linear system
- Extend to **3D flow problems**

Tokamak-Edge Plasma Dynamics

Slow ← *Parallel Ion Transport* · *Ion-Ion Collisions* · *Electrostatic Sound Wave* · *Parallel Electron Transport* · *Electrostatic Alfven Waves* · *Plasma Frequency* → *Fast*

Inner edge: Adjacent to the core

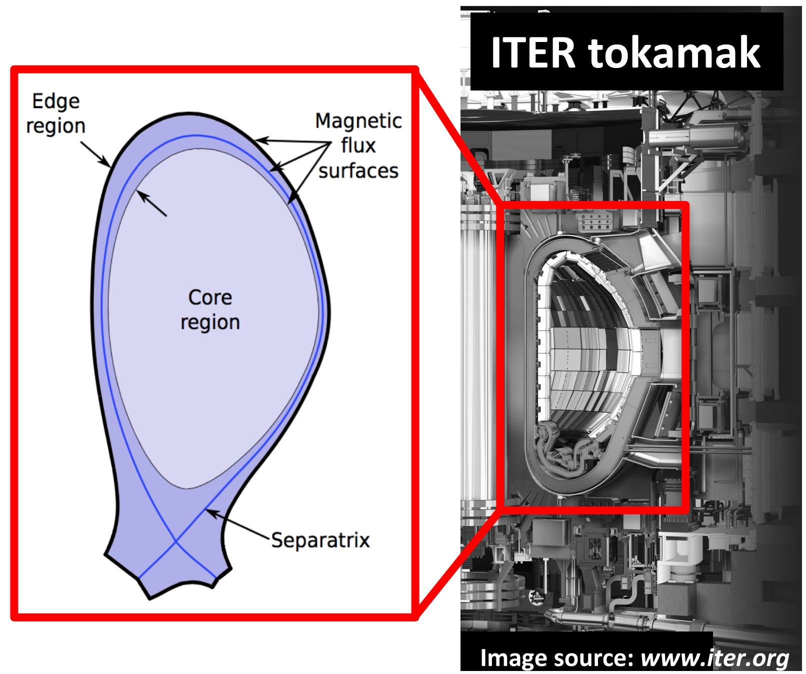
- High temperature and density; Mean free paths comparable to density/temperature gradients
- Weakly collisional

Requires kinetic simulation with collision model

Outer edge: Near tokamak wall

- Low temperature and density; Short mean free paths compared to density and temperature gradients
- Strongly collisional

Introduces very small time scales



Plasma dynamics in the edge region is characterized by a large range of temporal scales

Governing Equations

Full-*f* gyrokinetic Vlasov equation for each ion species

$$\frac{\partial B^*_{\parallel,\alpha} f_\alpha}{\partial t} + \underbrace{\nabla_{\mathbf{R}} \cdot \left(\dot{\mathbf{R}}_\alpha B^*_{\parallel,\alpha} f_\alpha \right) + \frac{\partial}{\partial v_\parallel} \left(\dot{v}_{\parallel,\alpha} B^*_{\parallel,\alpha} f_\alpha \right)}_{\text{Vlasov}} = \underbrace{\sum_\beta c\left[f_\alpha, f_\beta\right]}_{\text{Collisions}}$$

where

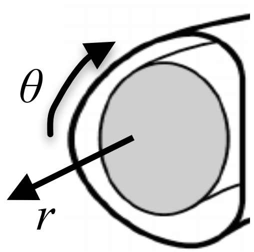
$$\dot{\mathbf{R}}_\alpha \equiv \dot{\mathbf{R}}_\alpha \left(\mathbf{R}, v_\parallel, \mu, t\right) = \frac{1}{B^*_{\parallel,\alpha}} \left[v_\parallel \mathbf{B}^*_\alpha + \frac{1}{Z_\alpha e} \hat{\mathbf{b}} \times \left(Z_\alpha e \mathbf{E} + \mu \nabla_{\mathbf{R}} B \right) \right] \quad \text{Velocity}$$

$$\dot{v}_{\parallel,\alpha} \equiv \dot{v}_{\parallel,\alpha} \left(\mathbf{R}, v_\parallel, \mu, t\right) = -\frac{1}{m_\alpha B^*_{\parallel,\alpha}} \mathbf{B}^*_\alpha \cdot \left(Z_\alpha e \mathbf{E} + \mu \nabla_{\mathbf{R}} B \right) \quad \text{Acceleration}$$

4D (2D-2V) phase space

$$\mathbf{R} \equiv \{r, \theta\}$$

$$v_\parallel, \;\; \mu = \frac{1}{2} \frac{m_\alpha v_\perp^2}{B}$$



Electric field **E** can be specified or computed from f_α using the Poisson equation for electrostatic potential

We consider *single-species cases* in this study.

Fokker-Planck Collision Model

Fokker-Planck-Rosenbluth equation

$$c\left[f_\alpha, f_\beta\right] = \lambda_c \left(\frac{4\pi Z_\alpha Z_\beta e^2}{m_\alpha}\right)^2 \nabla_{(v_\parallel,\mu)} \cdot \left[\vec{\gamma}_\beta f_\alpha + \overleftrightarrow{\tau}_\beta \nabla_{(v_\parallel,\mu)} f_\alpha\right]$$

where the advective and diffusive coefficients are given by

$$\vec{\gamma}_\beta = \begin{bmatrix} \frac{\partial \varphi_\beta}{\partial v_\parallel} & 2\mu \frac{m_\beta}{B} \frac{\partial \varphi_\beta}{\partial \mu} \end{bmatrix}, \quad \overleftrightarrow{\tau}_\beta = \begin{bmatrix} -\frac{\partial^2 \varrho_\beta}{\partial v_\parallel^2} & -2\mu \frac{m_\beta}{B} \frac{\partial^2 \varrho_\beta}{\partial v_\parallel \mu} \\ -2\mu \frac{m_\beta}{B} \frac{\partial^2 \varrho_\beta}{\partial v_\parallel \mu} & -2\mu \left(\frac{m_\beta}{B}\right)^2 \left\{2\mu \frac{\partial^2 \varrho_\beta}{\partial \mu^2} + \frac{\partial \varrho_\beta}{\partial \mu}\right\} \end{bmatrix}$$

Rosenbluth potentials are related to f_β by the Poisson equations

$$\frac{\partial^2 \varphi_\beta}{\partial v_\parallel^2} + \frac{m_\beta}{B} \frac{\partial}{\partial \mu} \left(2\mu \frac{\partial \varphi_\beta}{\partial \mu}\right) = f_\beta$$

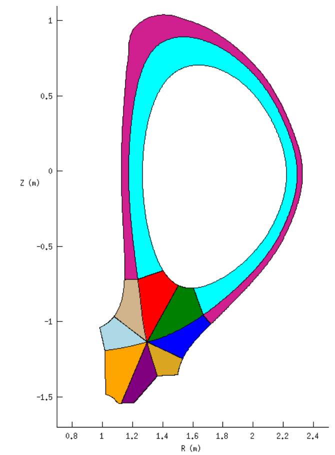
$$\frac{\partial^2 \varrho_\beta}{\partial v_\parallel^2} + \frac{m_\beta}{B} \frac{\partial}{\partial \mu} \left(2\mu \frac{\partial \varrho_\beta}{\partial \mu}\right) = \varphi_\beta$$

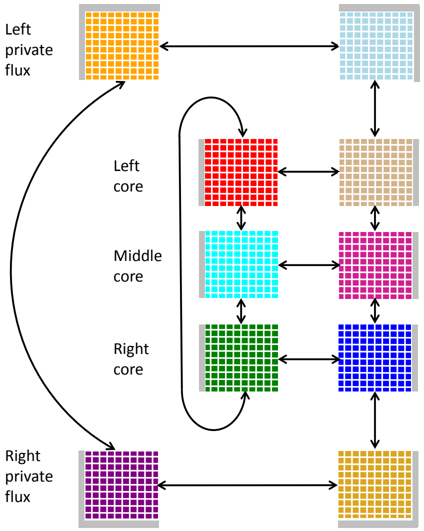
Non-linear, integro-differential term

Each evaluation of the Fokker-Planck term requires Poisson solve in the velocity space

COGENT: Continuum Gyrokinetic Edge New Technology

- *Governing equations*: 4D (2D 2V) Eulerian gyrokinetic Vlasov-Poisson system with imposed magnetic field with collision models
- *Domain*: Tokamak edge region (from core across the separatrix to the scrape-off layer)
- *Discretization*: 4^{th} order finite-volume method over mapped, multi-block grids
- Collaborative effort between *Center for Applied Scientific Computing* at LLNL and *Applied Numerical Algorithms Group* (ANAG) at LBL
- Based on *CHOMBO* (Finite-volume AMR package developed at LBL)
- Open-source, released under BSD license: https://github.com/LLNL/COGENT



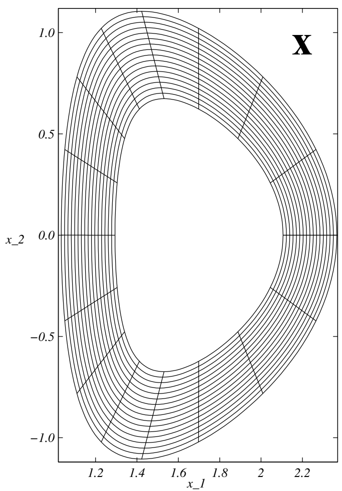


Current areas of research

- *IMEX methods* (this paper)
- *Electron models*
- *Extension to 5D (3D 2V)*

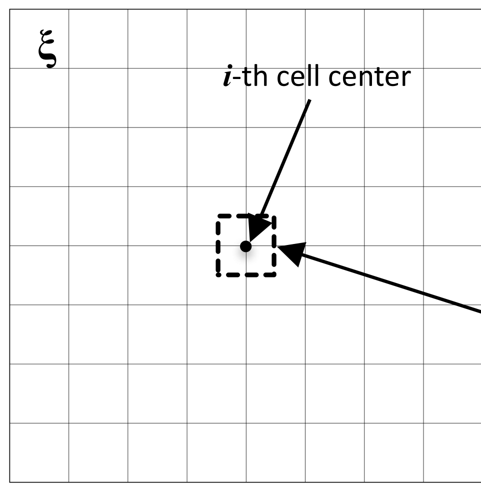
Spatial Discretization (1)

Finite-volume discretization on mapped grids



$$\mathbf{x} = \mathbf{X}(\boldsymbol{\xi})$$

$$\boldsymbol{\xi} = \mathbf{X}^{-1}(\mathbf{x})$$



Physical domain **Computational domain**

Hypercube of unit length

$$\Omega \equiv \{\boldsymbol{\xi} : 0 \le \boldsymbol{\xi} \cdot \mathbf{e}_d \le 1, 1 \le d \le 4\}$$

discretized into computational cells

$$\omega_{\mathbf{i}} = \prod_{d=1}^{4} \left[\left(\mathbf{i} - \frac{1}{2}\mathbf{e}_d \right) h, \left(\mathbf{i} + \frac{1}{2}\mathbf{e}_d \right) h \right]$$

$\mathbf{i}$: 4-dimensional integer index (i, j, k, l)
h: grid spacing

Integral form of the governing equations

$$\frac{\partial}{\partial t}\left(\int_{\mathbf{X}(\omega_{\mathbf{i}})} f d\mathbf{x} \right) = \int_{\partial\mathbf{X}(\omega_{\mathbf{i}})} \mathbf{V} f d\mathbf{x} + \int_{\partial\mathbf{X}(\omega_{\mathbf{i}})} \mathbf{C} f d\mathbf{x}$$

Both the Vlasov and the collision terms can be written in divergence form

Spatially-discretized ODE in time

$$\frac{\partial \bar{f}_{\mathbf{i}}}{\partial t} = \frac{1}{h}\sum_{d=1}^{4}\left[\left(\left\langle \hat{V}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} \right\rangle - \left\langle \hat{V}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d} \right\rangle \right) + \left(\left\langle \hat{C}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} \right\rangle - \left\langle \hat{C}_{\mathbf{i}-\frac{1}{2}\mathbf{e}_d} \right\rangle \right) \right]$$

Cell-averaged solution

Face-averaged Vlasov fluxes

Face-averaged collision fluxes

Spatial Discretization (2)

Face-averaged values are computed from *face-centered values* to 4th order using corrections

$$\langle u \rangle_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} = u_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} + \frac{h^2}{24} \sum_{\substack{d'=1 \\ d'\neq d}}^{4} \left. \frac{\partial^2 u}{\partial \xi_{d'}^2} \right|_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d} + \mathcal{O}\left(h^4\right)$$

Computed using 2nd order central differences since multiplied by h^2

$$\bar{u}_{\mathbf{i}} = u_{\mathbf{i}} + \frac{h^2}{24} \sum_{d=1}^{4} \left. \frac{\partial^2 u}{\partial \xi_d^2} \right|_{\mathbf{i}} + \mathcal{O}\left(h^4\right)$$

Cell-averaged values are computed from *cell-centered values* to 4th order using corrections

And vice-versa…

Spatial Discretization (3): Vlasov Flux

Face-averaged Vlasov flux

$$\left\langle \hat{V} \right\rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \left\langle \hat{\mathbf{V}} \right\rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} \cdot \mathbf{e}_d$$

The Vlasov flux vector is an *advective term*

$$\mathbf{V}(f) \equiv \begin{bmatrix} V^{(1)}(f) \\ \vdots \\ V^{(4)}(f) \end{bmatrix}$$

$$V^{(s)}(f) = a^{(s)} f;$$

$$a^{(s)} = \begin{cases} \left(\dot{\mathbf{R}} \cdot \hat{r}\right), & s = 1 \\ \left(\dot{\mathbf{R}} \cdot \hat{\theta}\right), & s = 1 \\ \dot{v}_{\parallel}, & s = 3, \\ 0, & s = 4 \end{cases}$$

The **face-averaged Vlasov flux** is computed as a 4^{th} *order accurate convolution*:

$$\left\langle \hat{V}^{(s)} \right\rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \left\langle a^{(s)} \right\rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} \left\langle \bar{\bar{f}} \right\rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} + \frac{h^2}{12} \sum_{\substack{d'=1 \\ d' \neq d}}^{4} \left(\frac{\partial a^{(s)}}{\partial \xi_{d'}} \frac{\partial \bar{\bar{f}}}{\partial \xi_{d'}} \right) + \mathcal{O}\left(h^4\right),$$

$$\bar{\bar{f}} = \frac{1}{\bar{\omega}_{\mathbf{i}}} \int_{\bar{\omega}_{\mathbf{i}}} f \, d\boldsymbol{\xi} = J_{\mathbf{i}}^{-1} \left[\bar{f}_{\mathbf{i}} - \frac{h^2}{12} \nabla_{\boldsymbol{\xi}} \bar{f} \cdot \nabla_{\boldsymbol{\xi}} J \right] + \mathcal{O}\left(h^4\right)$$

Computed using 2^{nd} order central differences since multiplied by h^2

$$\left\langle \bar{\bar{f}} \right\rangle_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d}$$

Computed from cell-averaged values using the 5^{th} *order WENO scheme*

Spatial Discretization (4): Collision Term

Face-centered collision flux

$$C^{(d)}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_d} = \begin{cases} \Gamma^{v_\parallel}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_3}, & d=3 \\ \Gamma^{\mu}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_4}, & d=4 \\ 0, & \text{otherwise} \end{cases}$$

The collision term *acts only on the velocity space* and the ***velocity grid is Cartesian***

Collision flux along $v_{||}$

$$\Gamma^{v_\parallel}_{\mathbf{i}+\frac{1}{2}\mathbf{e}_3} \equiv \Gamma^{v_\parallel}_{k+\frac{1}{2},l} = \left[\gamma_{v_\parallel} f + \tau_{v_\parallel v_\parallel}\frac{\partial f}{\partial v_\parallel} + \tau_{v_\parallel \mu}\frac{\partial f}{\partial \mu}\right]_{k+\frac{1}{2},l}$$

5th order upwind based on the sign of coefficient

4th order central

- The Poisson equations for the Rosenbluth potentials are solved using a 2nd order method (see *Dorf et. al, Contrib. Plasma Phys., 2014*)
- *Advection-diffusion coefficients computed from the Rosenbluth potentials using 2nd order central finite differences.*

Temporal Scales at Tokamak Edge

Eigenvalues (time scales) of the entire RHS, and the Vlasov and collision terms separately

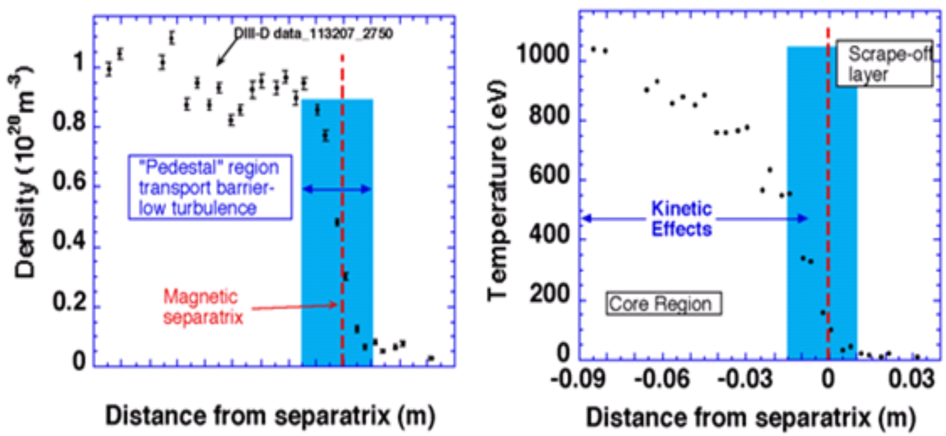
$$\lambda_R \equiv \lambda\left[\frac{d\mathcal{R}\left(\tilde{f}\right)}{d\tilde{f}}\right]$$

$$\lambda_V \equiv \lambda\left[\frac{d\mathcal{V}\left(\tilde{f}\right)}{d\tilde{f}}\right]$$

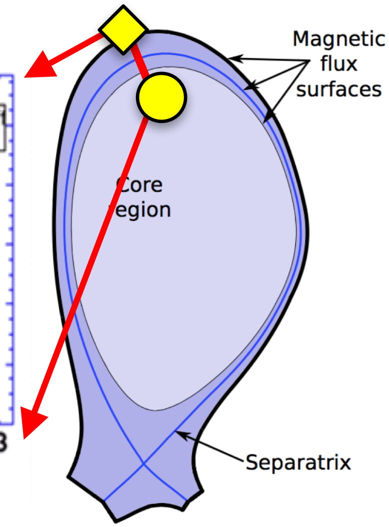
$$\lambda_C \equiv \lambda\left[\frac{d\mathcal{C}\left(\tilde{f}\right)}{d\tilde{f}}\right]$$

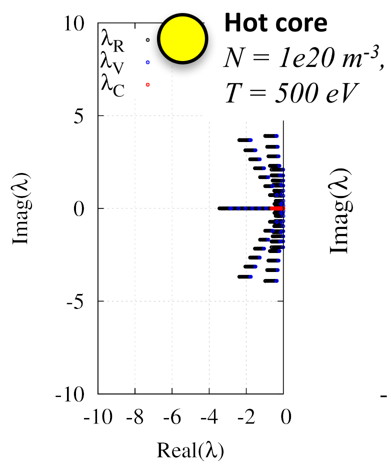
Jacobians computed using finite-differences on a very small grid

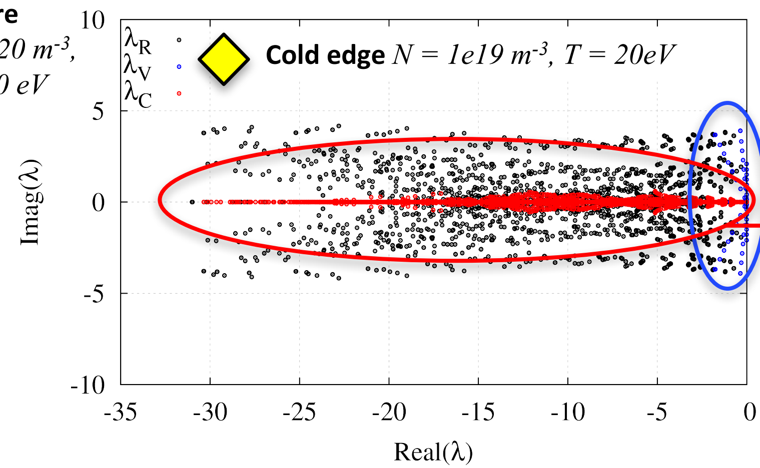
Tokamak edge density and temperature



Reference: *Porter, et al., Physics of Plasmas, 2000*







At the cold edge

$$|\lambda_V| \ll |\lambda_C|$$

$$\mathcal{R}_{\text{stiff}} \equiv \mathcal{C}\left(\tilde{f}\right)$$
$$\mathcal{R}_{\text{nonstiff}} \equiv \mathcal{V}\left(\tilde{f}\right)$$

Preconditioning

Preconditioner *(sparse matrix)* $\qquad$ **Exact Jacobian** *(never assembled)*

$$\left[\alpha\mathcal{I} - \frac{d\bar{\mathcal{C}}\left(\tilde{f}\right)}{d\tilde{f}}\right] \approx \left[\alpha\mathcal{I} - \frac{d\mathcal{C}\left(\tilde{f}\right)}{d\tilde{f}}\right]$$

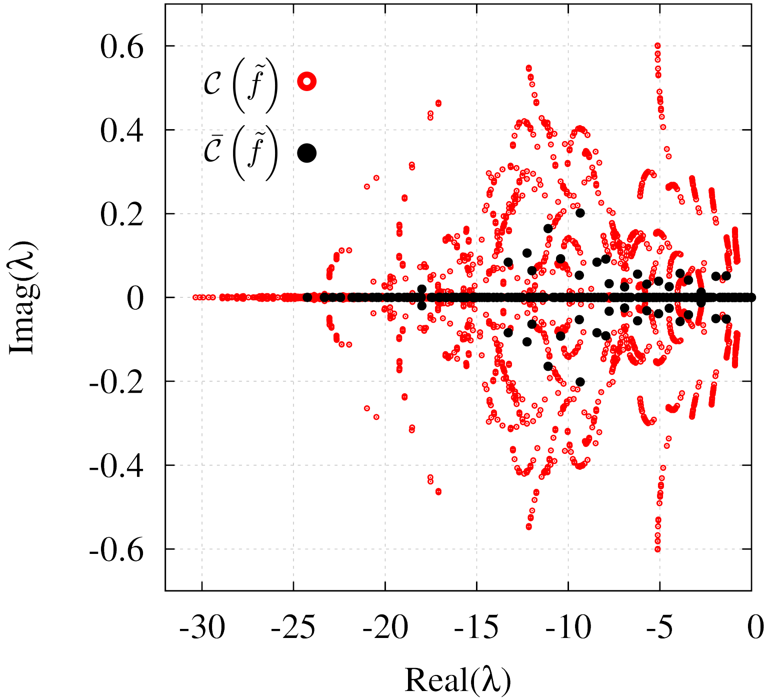
- Use lower order finite differences to construct the preconditioning matrix
- More sparse than the actual Jacobian
- Assembled and stored as a sparse matrix

$\mathcal{C}\left(\tilde{f}\right)$
- 5th order upwind for advective terms
- 4th order central for diffusion terms

$\bar{\mathcal{C}}\left(\tilde{f}\right)$
- 1st order upwind for advective terms
- 2nd order central for diffusion terms

→ Results in a 9-banded matrix

Eigenvalues of the Jacobian of the actual collisions term and the approximation for preconditioning



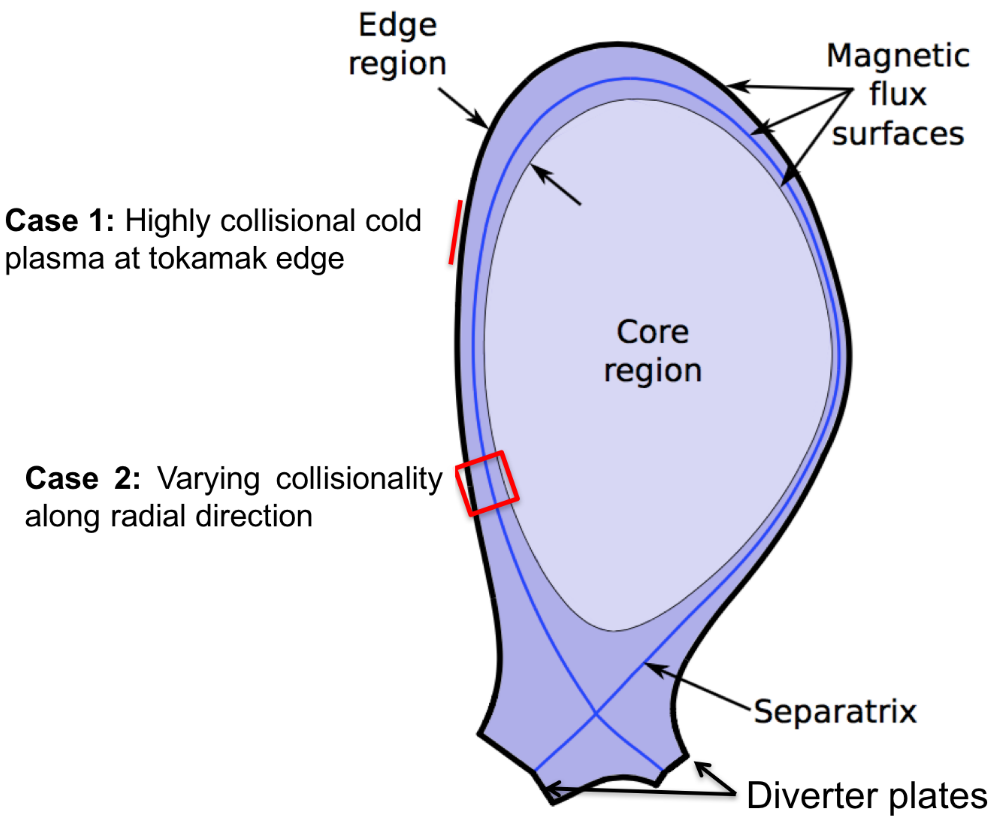
The preconditioner is inverted using the *Gauss-Seidel method* (computationally inexpensive)

Test Problems

Ion parallel heat transport on Cartesian domains

- **Case 1:** 1D dynamics of a strongly collisional plasma
- **Case 2:** 2D dynamics with varying collisionality; representative of a radial patch at the tokamak edge

Specified electrostatic potential

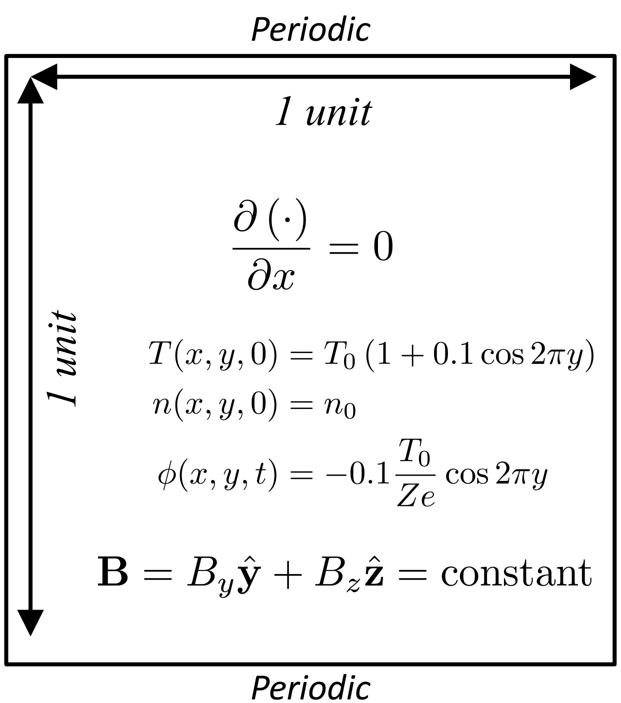


Test Problem 1: 1D Ion Parallel Heat Transport

A 2D slab (in configuration space), ***representative of cold edge***

$n = 10^{20}\ m^{-3}$
$T = 20\ eV$

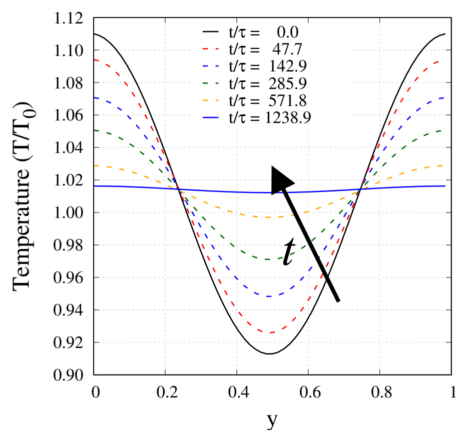
$k_{\|}\lambda = 0.065$ **Highly collisional**

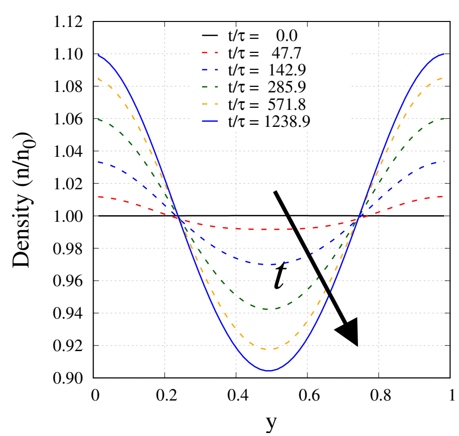


$$\frac{\partial(\cdot)}{\partial x} = 0$$

$$T(x, y, 0) = T_0\,(1 + 0.1\cos 2\pi y)$$
$$n(x, y, 0) = n_0$$
$$\phi(x, y, t) = -0.1\frac{T_0}{Ze}\cos 2\pi y$$

$$\mathbf{B} = B_y\hat{\mathbf{y}} + B_z\hat{\mathbf{z}} = \text{constant}$$



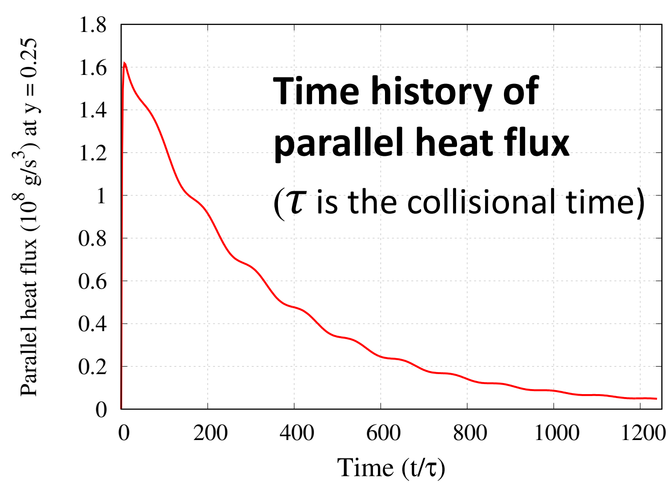


Transport time scale

- Temperature equilibrates to constant value
- Density assumes cosine shape to balance electrostatic potential

Collisional time scale

- Heat flux attains values consistent with temperature gradient



ARK4 vs. RK4 and Effect of Preconditioner

Computational cost of ARK4 with and without preconditioner (first 4 rows) and RK4 (last row)

Preconditioner for ARK4: *Gauss-Seidel solver with 80 iterations*

Vlasov CFL	Collision CFL	Number of Function Calls *			Wall time (seconds)		
		No PC	With PC	Ratio	No PC	With PC	Ratio
0.2	**2.4**	*388,408*	*397,023*	***1.02***	1.4×10^5	1.5×10^5	**1.07**
0.6	**6.1**	*156,935*	*142,297*	***0.91***	5.7×10^4	5.3×10^4	**0.93**
0.9	**9.7**	*103,801*	*75,249*	***0.72***	3.7×10^4	2.7×10^4	**0.73**
1.1	**12.1**	*89,544*	*61,298*	***0.68***	3.3×10^4	2.3×10^4	**0.70**
0.04	**0.5**	*260,000*			1.1×10^5		

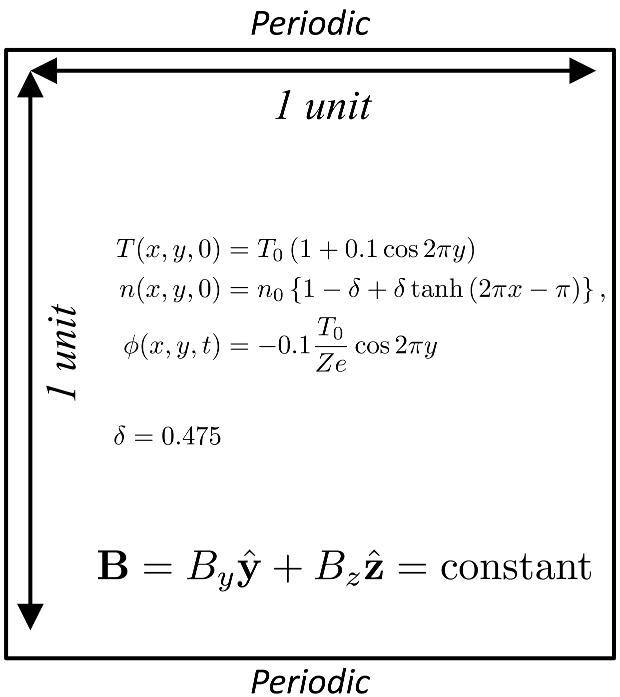
* Number of function calls = Calls from time integrator (time steps × stages) + number of Newton iterations + number of GMRES iterations

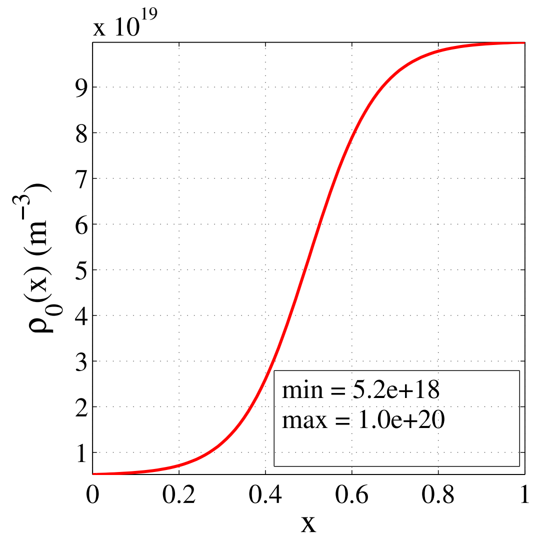
- Grid size: $6\ (x) \times 64\ (y) \times 36\ (v_{||}) \times 24\ (\mu)$, solved on 192 cores (2.6 GHz Intel Xeon)
- Preconditioner results in some speed-up at higher CFL numbers
- Overhead of assembling and inverting the preconditioning matrix is relatively small

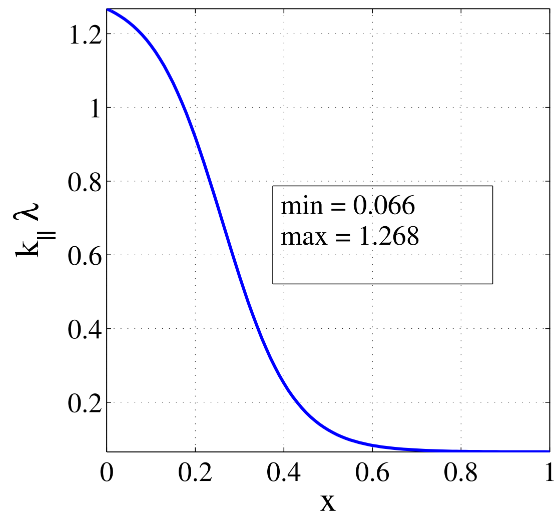
Test Problem 2: 2D Ion Parallel Heat Transport

A 2D slab (in configuration space), *representative of the **varying collisionality** in the edge region*

$$n = 10^{20}\ m^{-3}$$
$$T = 20\ eV$$

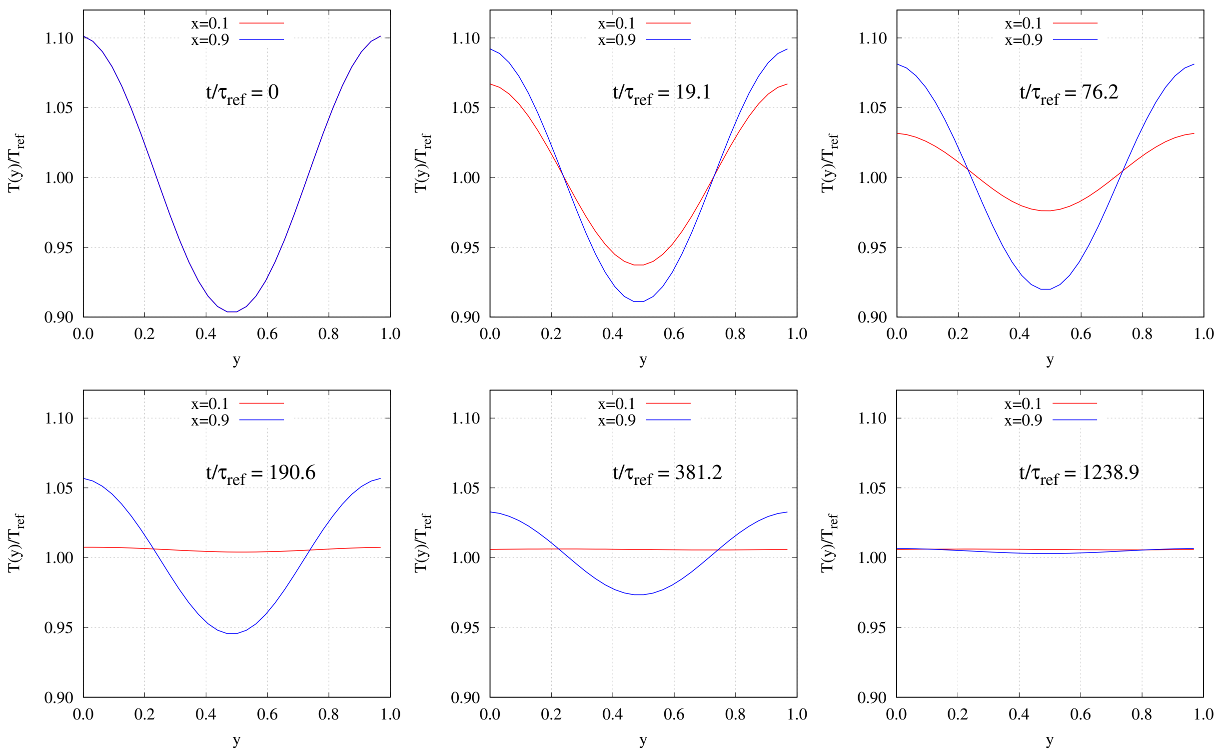


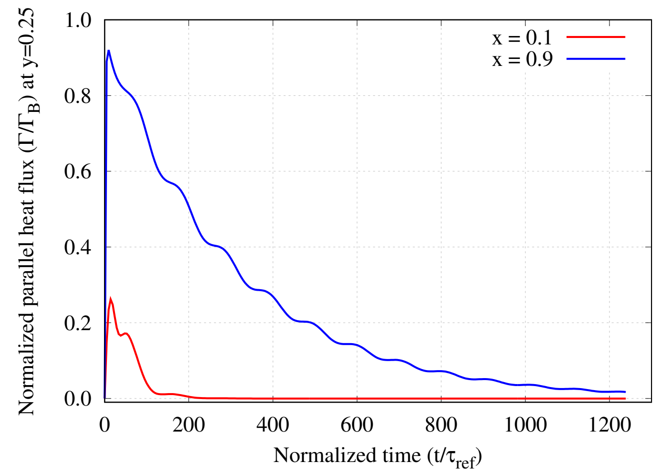




- As $x \rightarrow 0$, the plasma is *weakly collisional* due to low density: the *collisional mean free paths are comparable to temperature/density gradients*, and collisional time time scales are comparable to transit time scales.
- As $x \rightarrow 1$, the plasma is *strongly collisional* due to high density: the *collisional mean free paths are much smaller than temperature/density gradients*, and collisional time time scales are much faster to transit time scales.

Test Problem 2: 2D Ion Parallel Heat Transport





- Grid size: 32 (x) × 32 (y) × 36 $(v_{||})$ × 24 (μ)
- ***Fastest stable solution***: ARK4 is ~11x faster with ~50x larger time step.

Method	Vlasov CFL	Collision CFL	Number of Function Calls		Wall time (seconds)	
			No PC	With PC	No PC	With PC
ARK4	1.1	**25.8**	*52,551*	*30,852*	**2.5 ×10⁴**	**1.6 ×10⁴**
RK4	0.02	**0.52**	*260,000*		**1.7 ×10⁵**	

Summary and Future Work

- **IMEX approach for strongly-collisional tokamak edge plasma**
  - ✓ Collisions integrated in time implicitly while Vlasov term integrated in time explicitly
  - ✓ Wall time for fastest stable solution significantly reduced
  - ✓ Low order preconditioning results in lower computational cost at high collision CFL numbers

- **Future work**
  - More efficient solver for inverting the preconditioning matrix (Gauss-Seidel needs 80 iterations!)
  - Implement IMEX for other fast scales (*electrostatic Alfven waves, parallel electron transport, ion acoustic modes, parallel ion transport*)

Thank you.
Questions?

Lawrence Livermore National Laboratory