High-Order Semi-Implicit Time Integration Methods for Multiscale Advective Physics

Presented at University of Zaragoza (Online)

Debojyoti Ghosh
Center for Applied Scientific Computing

October 25, 2022

LLNL-PRES-841756
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

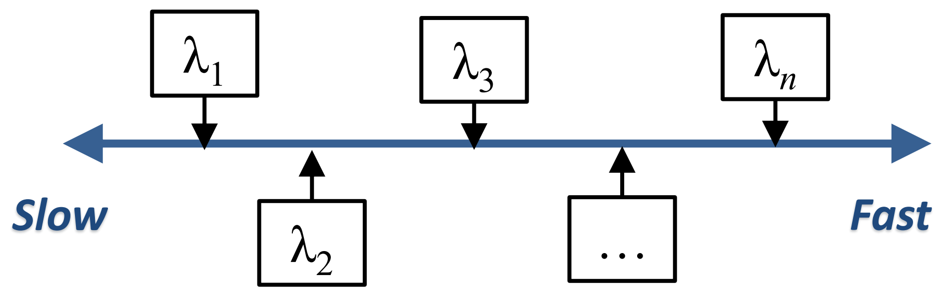
Lawrence Livermore National Laboratory

Time Scales in Complex Physics Simulations

Model ODE in time

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

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

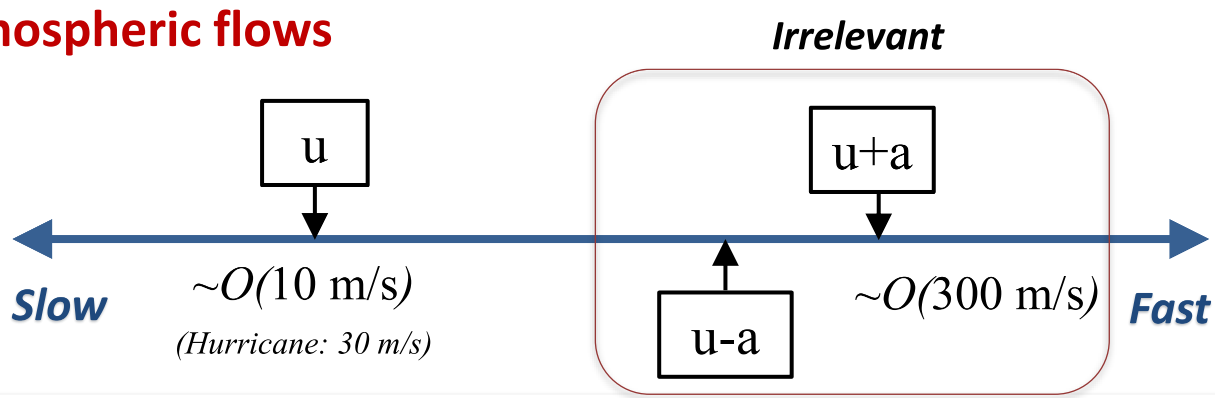


Which time scales do we want to resolve? *(Usually, some of them)*

In the context of atmospheric flows

u: *flow velocity*

a: *speed of sound*

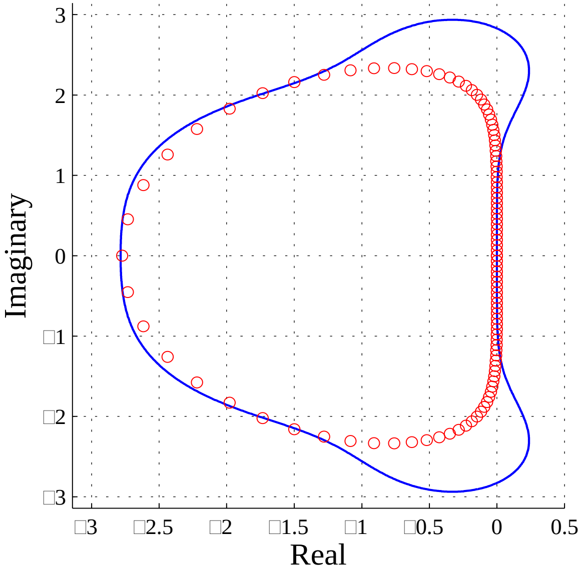


Explicit vs. Implicit Time Integration

Motivation for Implicit-Explicit (IMEX) Approach

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

- Simple to implement; very scalable
- Eigenvalues of RHS must lie within stability region of time integrator
- *Inefficient when resolving slow dynamics*



Implicit time-integration allows time steps determined by the physics

- *Unconditional stability*
- Requires solution to *nonlinear system of equations* (computational expense, scalability, preconditioning)
- *Why pay for inverting/solving the terms we want to resolve?*

IMEX: Best of both worlds?

Implicit-Explicit (IMEX) Time Integration

Resolve scales of interest; Treat implicitly faster scales

Semi-discrete ODE in time

Resulting from spatial discretization of PDE

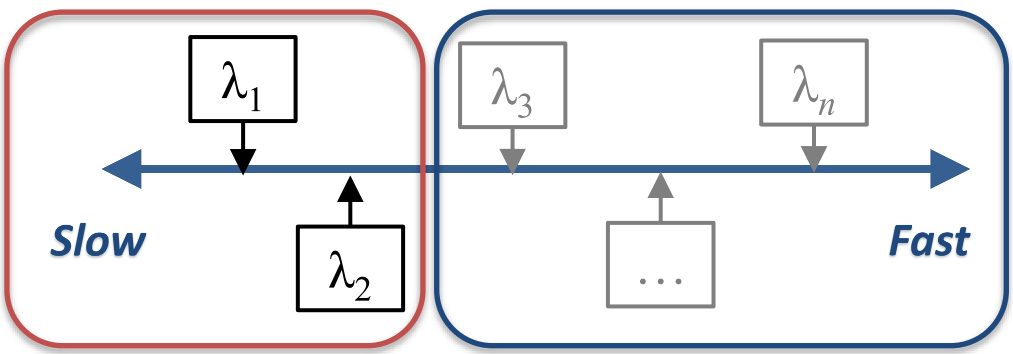
$$\frac{d\mathbf{y}}{dt} = \mathcal{R}\left(\mathbf{y}\right)$$

IMEX time integration: *partition RHS*

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

Time scales of interest
(nonstiff terms)

Explicit time integration



Faster time scales
(stiff terms)

Implicit time integration

Linear stability constraint on time step

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

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

Additive Runge-Kutta (ARK) Time Integrators

Multistage, high-order, conservative IMEX methods

$$\frac{d\mathbf{y}}{dt} = \mathcal{R}_{\text{nonstiff}}(\mathbf{y}) + \mathcal{R}_{\text{stiff}}(\mathbf{y})$$

Time step: From t_n to $t_{n+1} = t_n + \Delta t$

Stage solutions

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

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

Step completion

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

Butcher tableaux representation

Explicit RK

$$\begin{array}{c|ccccc} 0 & 0 & & & \\ c_2 & a_{21} & 0 & & \\ \vdots & \vdots & \ddots & 0 & \\ c_s & a_{s1} & \cdots & a_{s,s-1} & 0 \\ \hline & b_1 & \cdots & \cdots & b_s \end{array} \quad + \quad \begin{array}{c|ccccc} 0 & 0 & & & \\ \tilde{c}_2 & \tilde{a}_{21} & \gamma & & \\ \vdots & \vdots & \ddots & \gamma & \\ \tilde{c}_s & \tilde{a}_{s1} & \cdots & \tilde{a}_{s,s-1} & \gamma \\ \hline & b_1 & \cdots & \cdots & b_s \end{array}$$

Diagonally-Implicit RK (DIRK)

s is the number of stages

Note: *not any combination of an explicit RK and DIRK will work!*

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

Preconditioned GMRES

$$\mathcal{J}\mathcal{P}^{-1}\mathcal{P}\Delta y = \mathcal{F}(y_k)$$

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

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

Atmospheric Flows: Governing Equations

Exner Pressure and Potential Temperature

- COAMP5 (US Navy), MM5 (NCAR/PSU), NMM (NCEP)
- Does not conserve mass/momentum/energy

Mass, Momentum, and Potential Temperature

- WRF (NCAR), NUMA (NPS)
- Conserved mass and momentum, not energy
- Does not allow inclusion of true diffusion terms

Mass, Momentum, and Energy

- Examples?
- Conserves mass, momentum, and energy
- Allows inclusion of viscosity and thermal conduction

Atmospheric flows: **small perturbations** around hydrostatic balance

- Perturbation form of governing equations
- **Balanced formulation with full quantities**

Main Advantage? Allows the application of the vast number of CFD codes with minimal modifications

Fast Time Scales and Time Integration

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

Explicit Time Integration

- Time step size restricted by acoustic waves and/or vertical grid spacing
- Acoustic waves do not significantly impact any atmospheric phenomenon
- **Split-explicit methods**

Implicit Time Integration

- Unconditionally stable
- Requires solutions to non-linear system or linearized approximation

IMEX Time Integrators for Atmospheric Flows

IMEX Time Integration

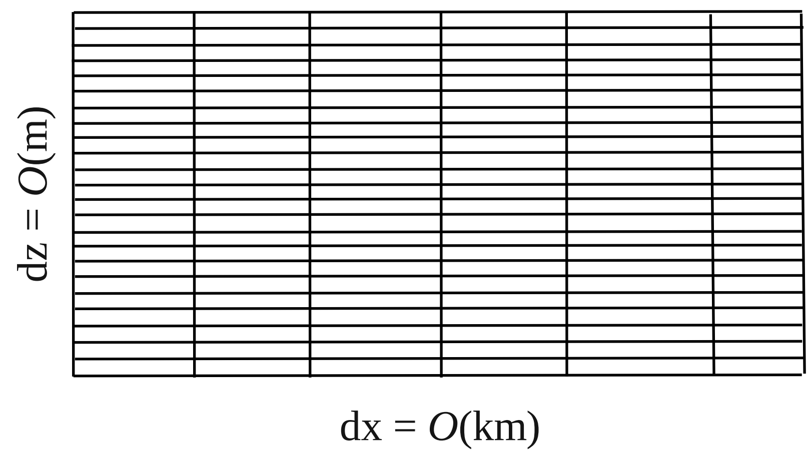
“Fast” waves implicitly
“Slow” waves explicitly

Horizontal-Explicit, Vertical-Implicit (HEVI)

Flux-Partitioned Methods

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



IMEX Time Integrators for Atmospheric Flows

IMEX Time Integration

"Fast" waves implicitly
"Slow" waves explicitly

Horizontal-Explicit, Vertical-Implicit (HEVI)

Flux-Partitioned Methods

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}) = \mathbf{S}$$

$$\downarrow$$

$$\nabla \cdot \mathbf{F} = \nabla \cdot \mathbf{F}_{\text{slow}} + \nabla \cdot \mathbf{F}_{\text{fast}}$$

Flux-Partitioned Methods

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

Developing an Atmospheric Flow Code

Collaborator: Emil Constantinescu (ANL) (2013-2016)

Develop a conservative, high-order finite-difference based on the compressible Euler equations *(conservation of mass, momentum, energy)*

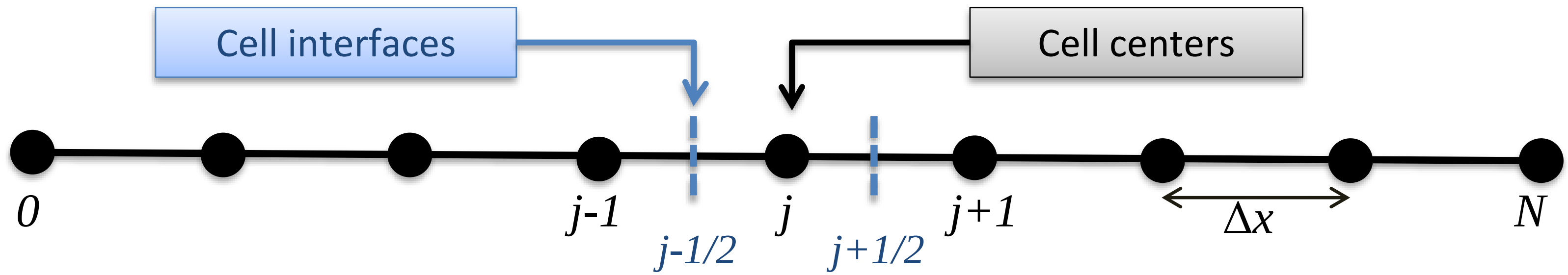
$$\frac{\partial}{\partial t}\begin{bmatrix}\rho \\ \rho u \\ \rho v \\ e\end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix}\rho u \\ \rho u^2 + p \\ \rho u v \\ (e+p)u\end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix}\rho v \\ \rho u v \\ \rho v^2 + p \\ (e+p)v\end{bmatrix} = \begin{bmatrix}0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

Balanced formulation for full quantities: Hydrostatic balance preserved to machine precision without writing equations in terms of perturbations

Flux-partitioning for IMEX time-integration: Isolate acoustic and gravity waves from convective mode

The work presented here is implemented in HyPar (http://hypar.github.io/) - C/C++ code for hyperbolic-parabolic PDEs.

Conservative Finite-Difference Schemes



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

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

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

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

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

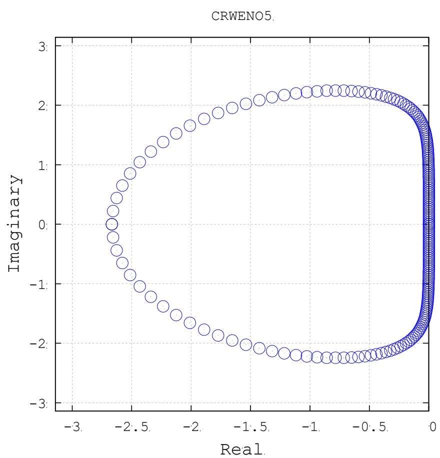
Semi-discrete ODE in time

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

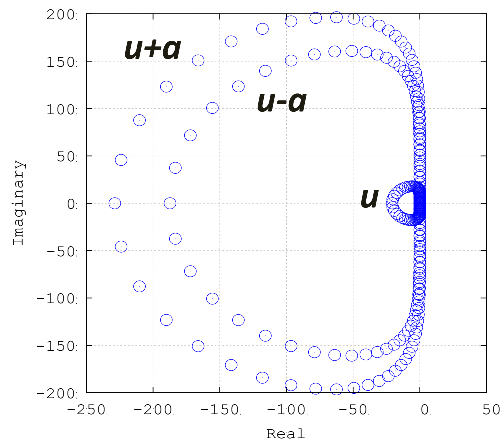
Discretization operator (e.g.:WENO5, CRWENO5)

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**

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

Convective flux (slow)

Acoustic flux (fast)

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

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

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

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

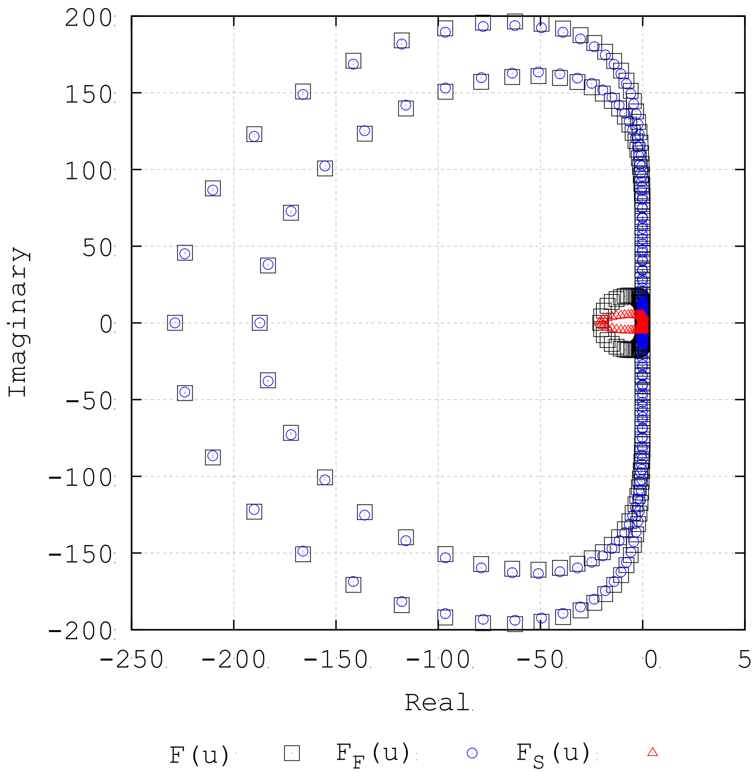
Characteristic-based Flux Partitioning (3)

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

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

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

(Not an error)



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

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

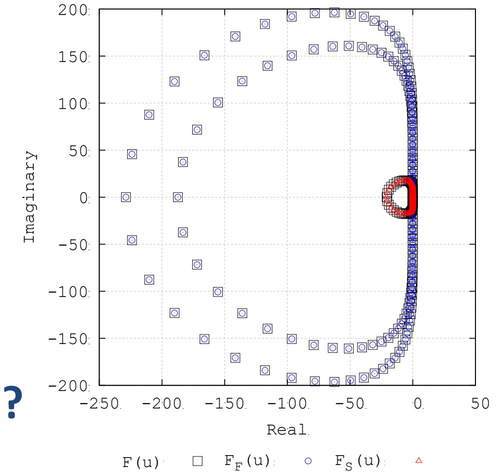
$$\mathbf{F}_F(\mathbf{u}) = [\mathcal{A}_F(\mathbf{u}_n)]\,\mathbf{u}$$
$$\mathbf{F}_S(\mathbf{u}) = \mathbf{F}(\mathbf{u}) - \mathbf{F}_F(\mathbf{u})$$

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)

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

$$\Downarrow$$

First order upwind discretization

Periodic boundaries ignored

$\Rightarrow$

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

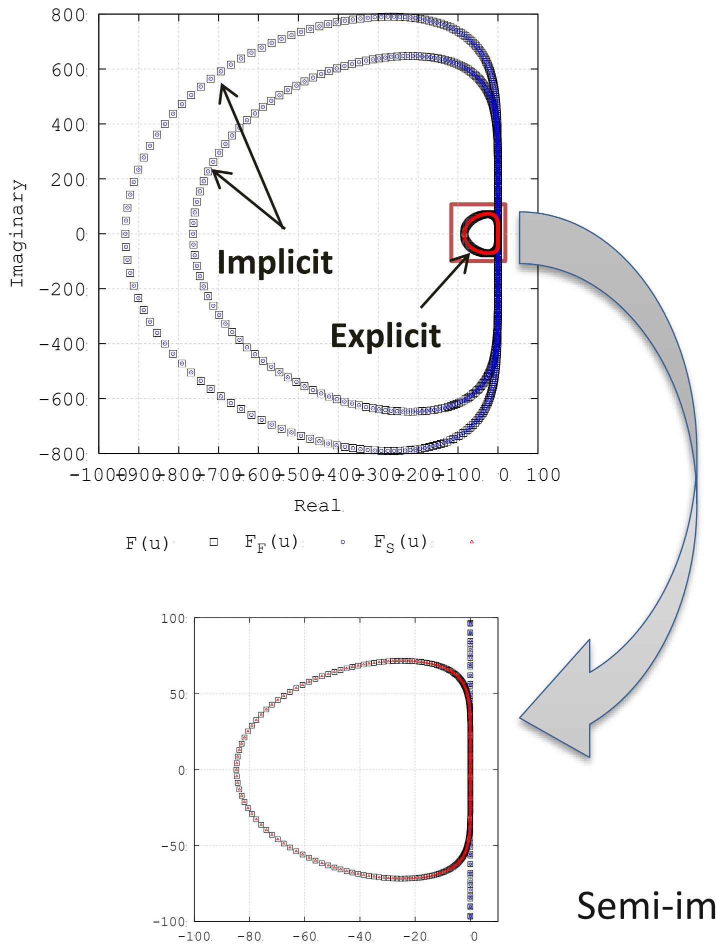
ARKIMEX 2e
- 2nd order accurate
- 3 stage (1 explicit, 2 implicit)
- L-Stable implicit part

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

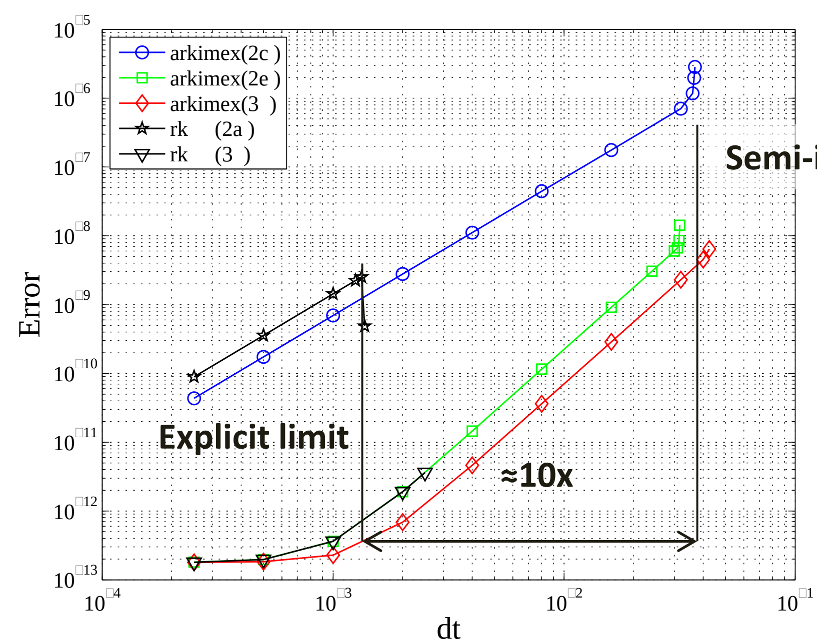


Initial solution

$$0 \leq x \leq 1$$

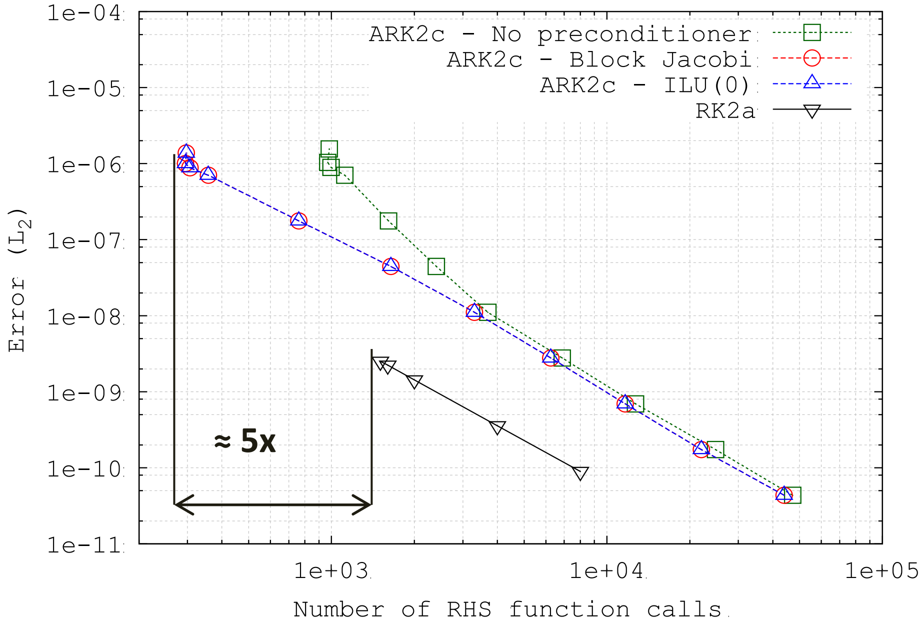
$$\rho = \rho_\infty + \hat{\rho}\sin(2\pi x)$$
$$u = u_\infty, p = p_\infty$$

CRWENO5, 320 grid points

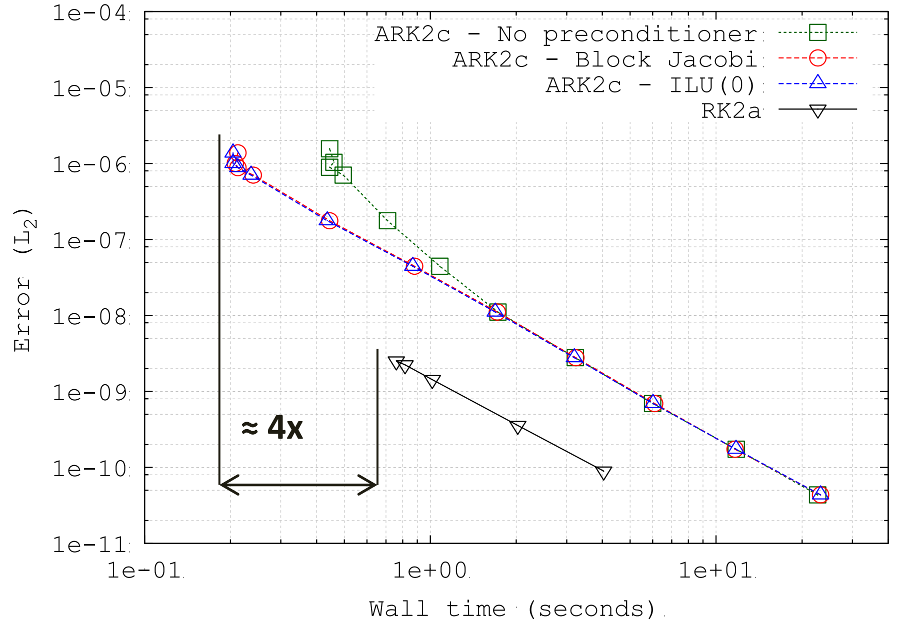


Semi-implicit time step size limit **1/ M_∞** than explicit time step size limit

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



Number of function calls

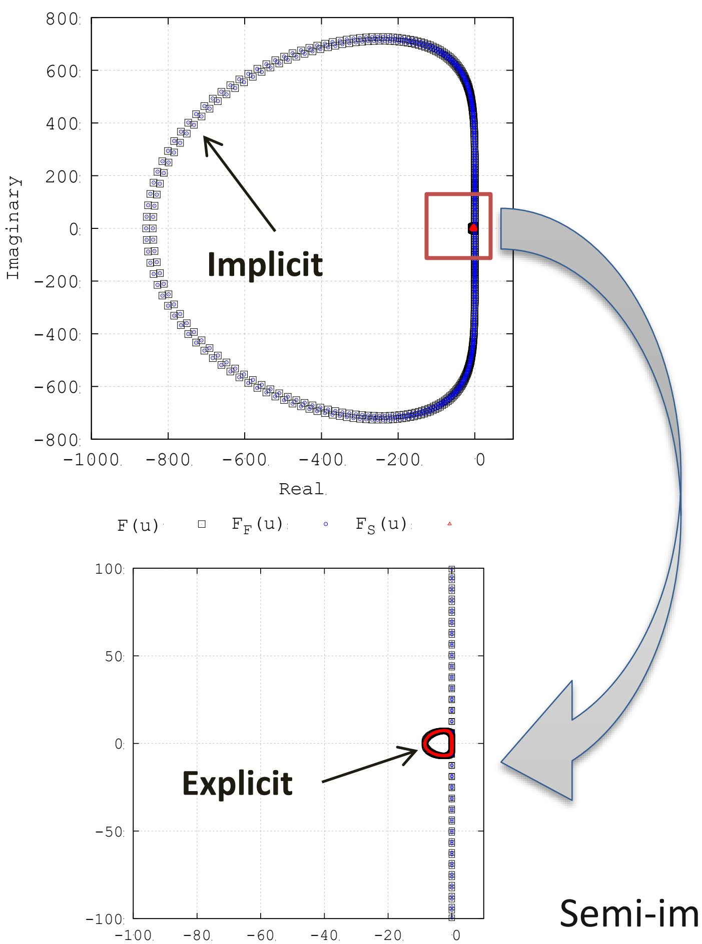


Wall time

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 ($M_\infty = 0.01$)

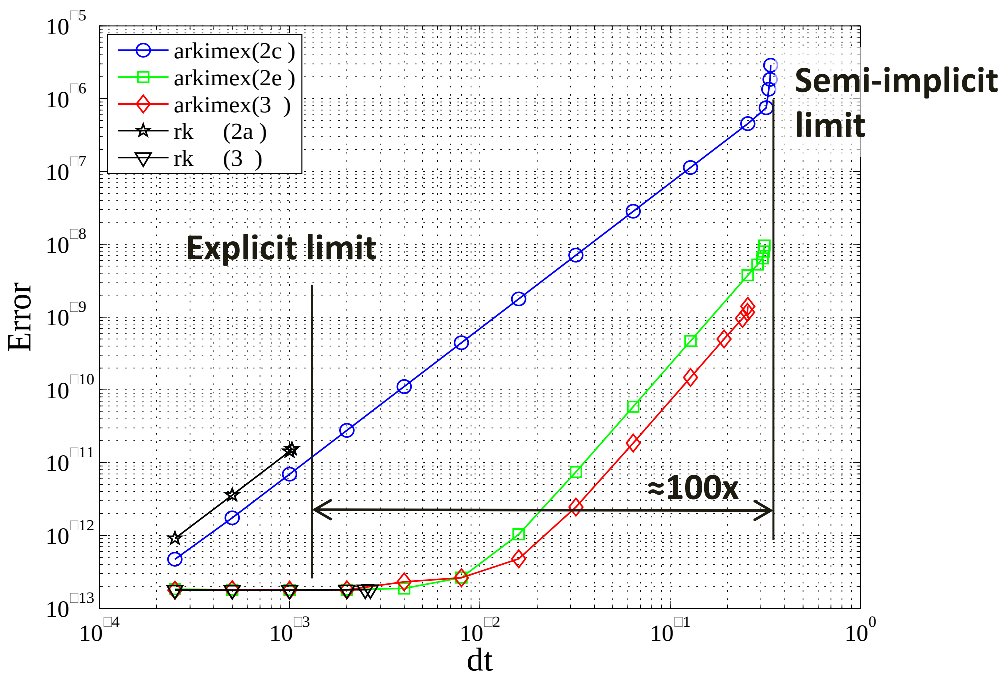
Eigenvalues



Initial solution

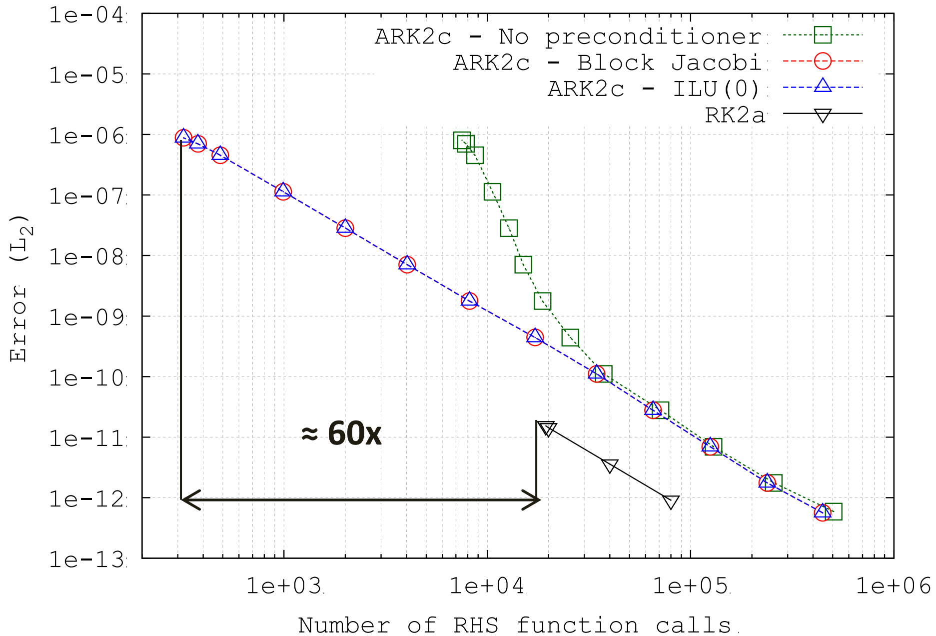
$$0 \leq x \leq 1 \qquad \rho = \rho_\infty + \hat{\rho}\sin(2\pi x)$$
$$u = u_\infty, p = p_\infty$$

CRWENO5, 320 grid points

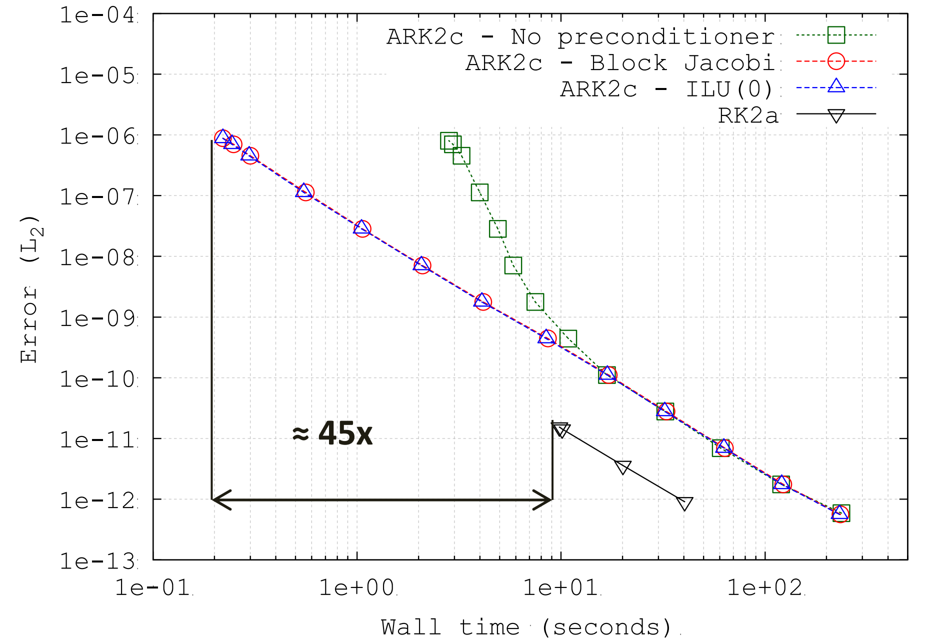


Semi-implicit time step size limit **1/ M_∞** than explicit time step size limit

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



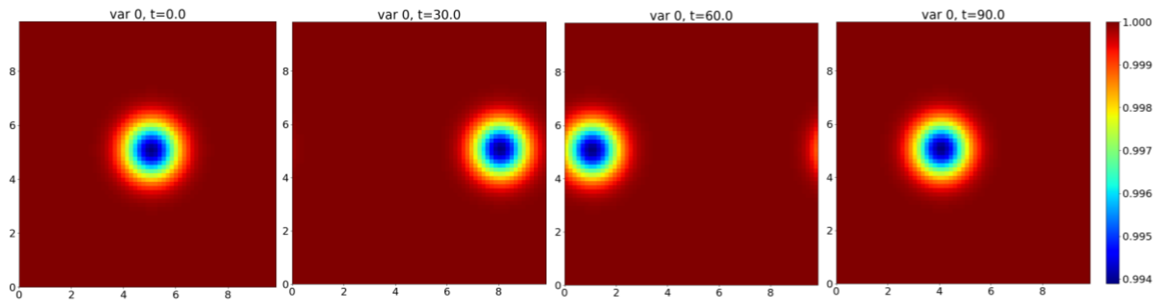
Number of function calls



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

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

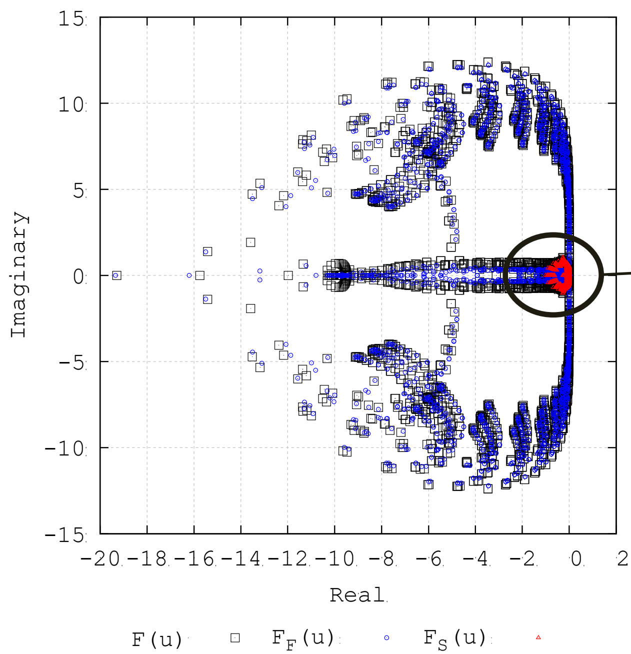
Vortex (Strength $b = 0.5$)

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

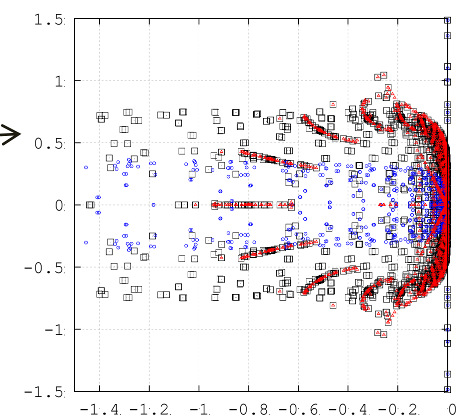
$$p = \left[1.0 - \frac{(\gamma-1)\,b^2}{8\gamma\pi^2}\exp\left(1-r^2\right)\right]^{\frac{\gamma}{\gamma-1}}$$

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

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

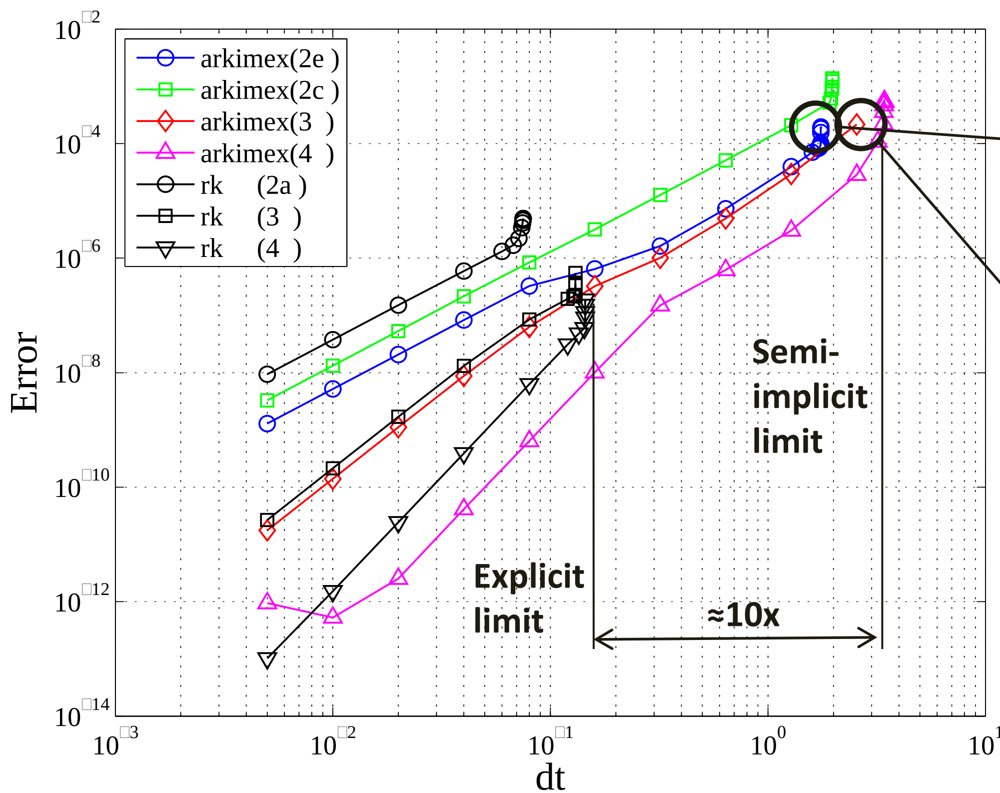


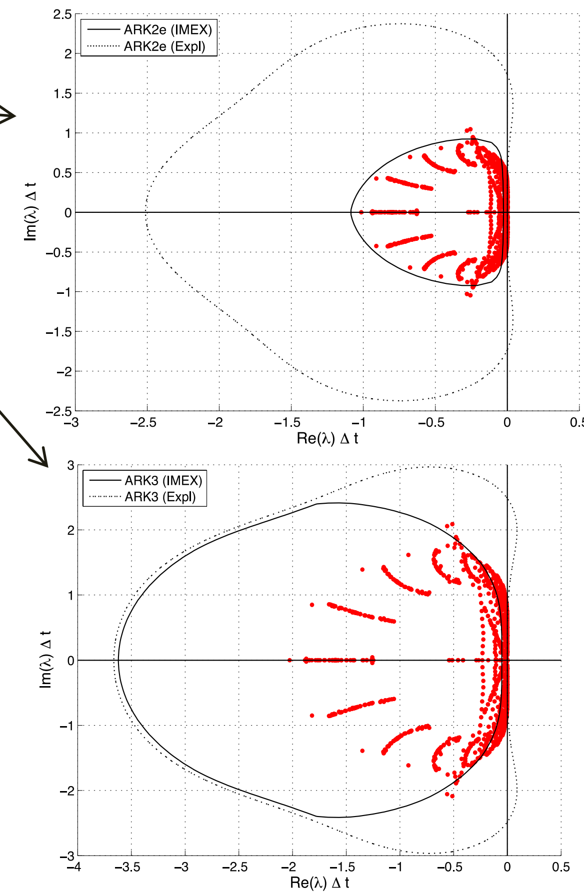
Eigenvalues of the right-hand-side operators



Grid: 32^2 points, CRWENO5

Example: 2D Low Mach Isentropic Vortex Convection



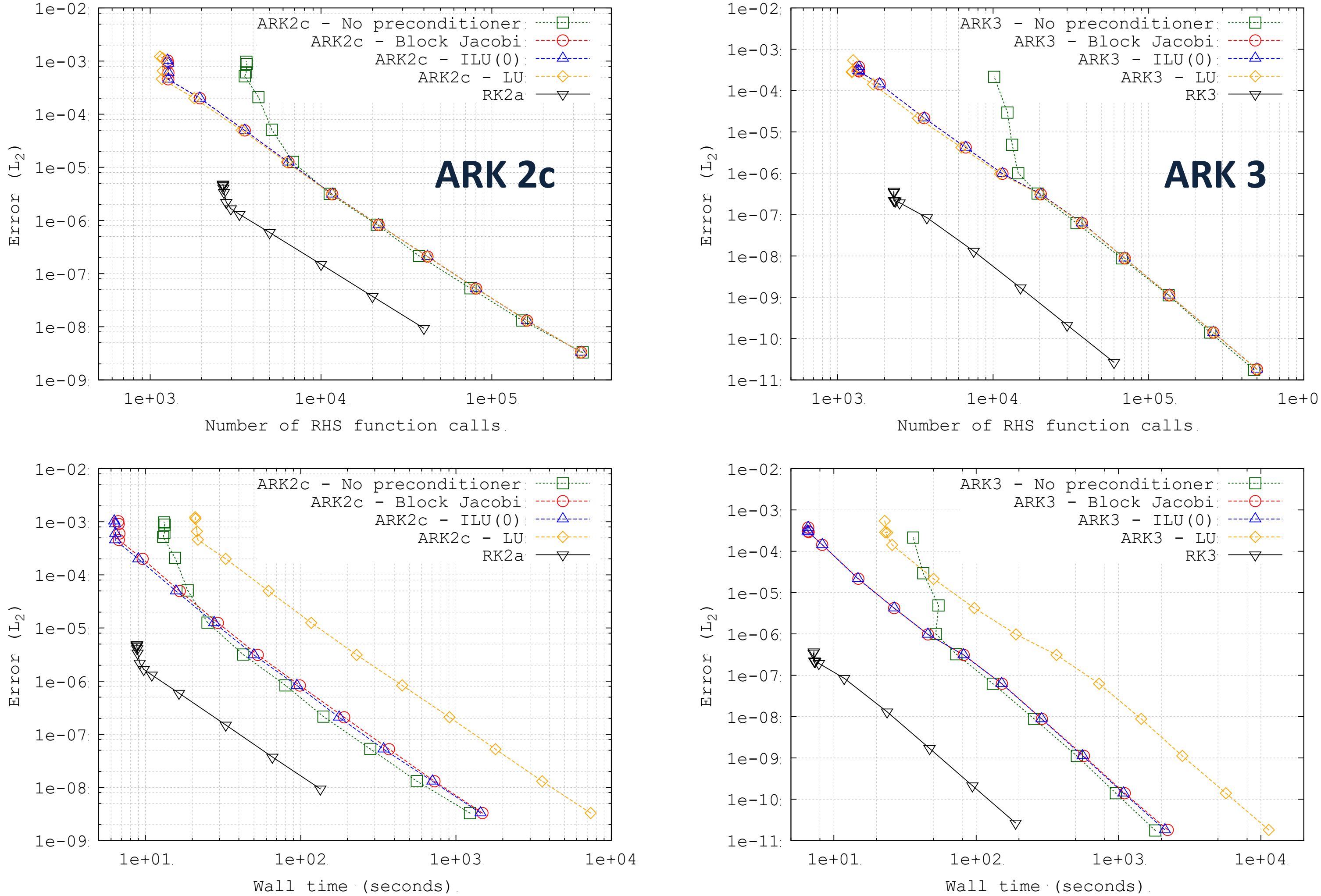


- **Optimal orders** of convergence observed for all methods
- **Time step size limited by the "slow" eigenvalues.**

Example: Vortex Convection (Computational Cost)

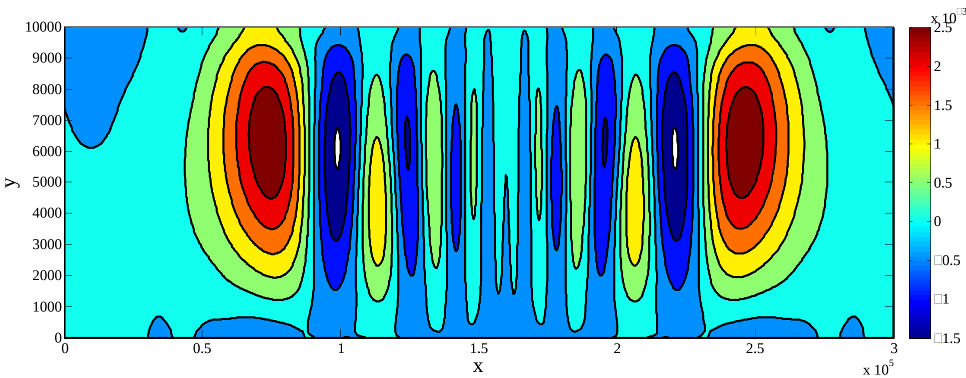
Number of function calls

Wall time



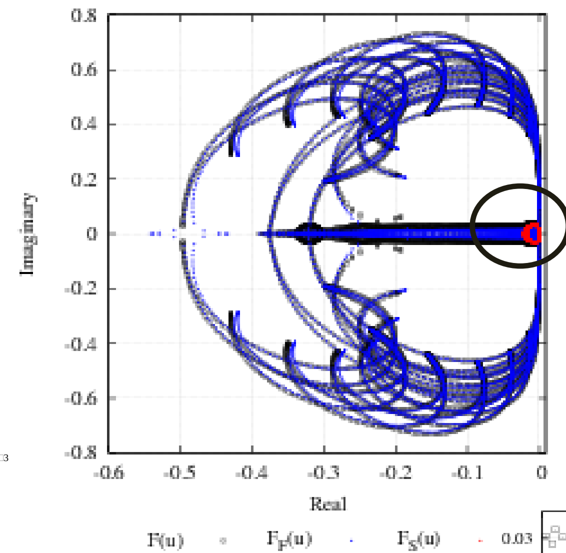
Example: Inertia – Gravity Wave

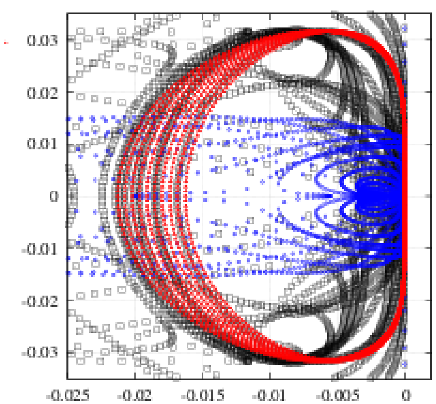
- **Periodic channel** – 300 km x 10 km
- **No-flux boundary conditions** at top and bottom boundaries
- **Mean horizontal velocity** of 20 m/s in a uniformly stratified atmosphere **($M_\infty \approx$ 0.06)**
- Initial solution – **Potential temperature perturbation**



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

Eigenvalues of the right-hand-side operators

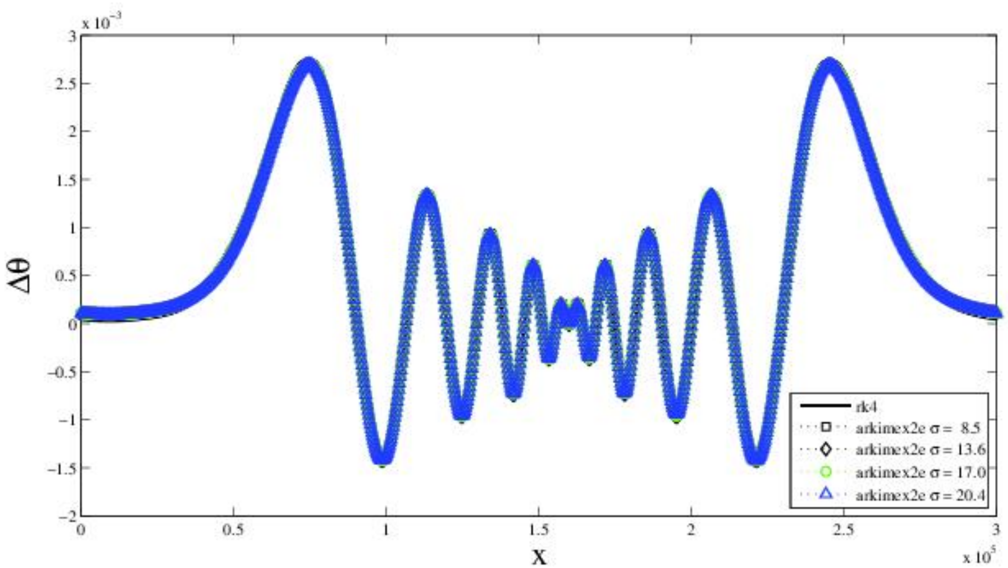




Grid: 300x10 points, CRWENO5

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

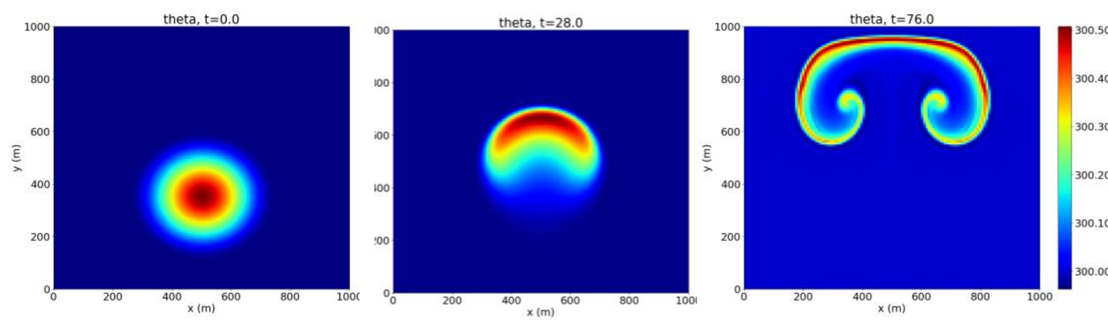
CFL ~ **1.0,** Wall time: **5400 s**
# of function calls: **24000**

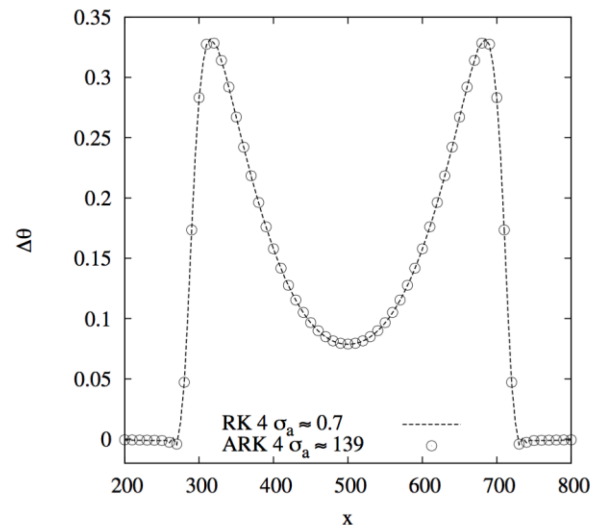
Cross-sectional potential temperature perturbations at 3000 seconds (y = 5 km) at CFL numbers **0.2 – 13.6**

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, # of** function calls: **160,000**





- **Box** – 1000 m^2; **No-flux boundary conditions** on all boundaries
- **Warm bubble initially at rest** rises in a still ambient atmosphere

Summary of Characteristic-Based Flux Partitioning for IMEX Methods

- 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 solving non-linear system of equations or linearize governing equations in a time step).
- **Disclaimer:** This work is quite old (~7 years); there has been more interesting work since then!

Multirate Time Integration for Atmospheric Flows
Background

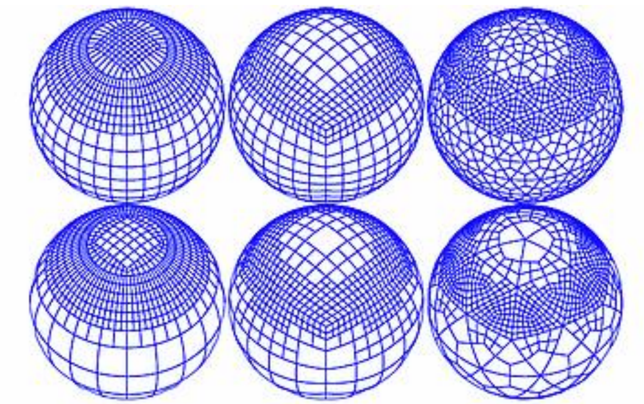


Image source:
http://frankgiraldo.wixsite.com/mysite/numa

NUMA – Nonhydrostatic Unified Model of the Atmosphere

- A scalable *high-order spectral-element code* for solving the Navier-Stokes and shallow water equations on a cube or a sphere (*part of the U.S. Navy NEPTUNE project*)
- **AMR-capable** (p4est/p6est library - http://p4est.github.io/)
- **Collaborators:** *Emil Constantinescu (ANL), Frank Giraldo (Naval Post. School), Michal Kopera (UC Santa Cruz)*

Time scales

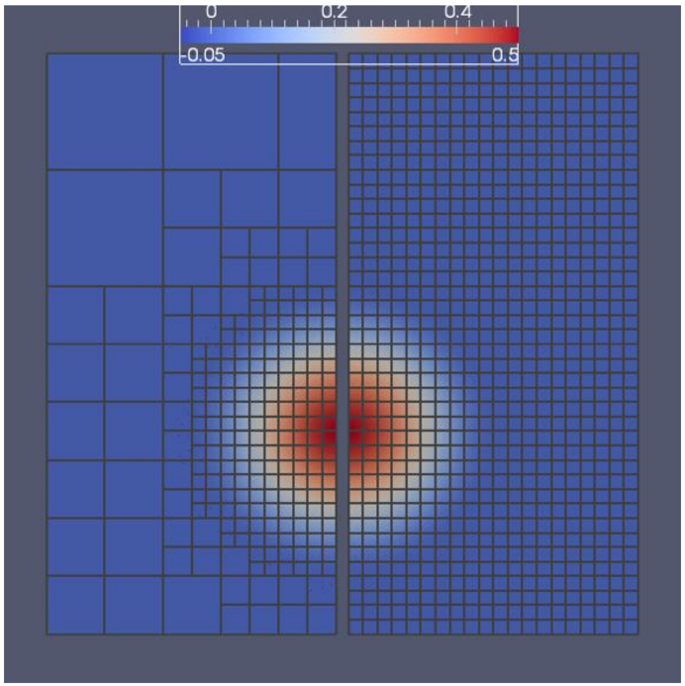
IMEX
Acoustic
Convective

Multirate time integration
Coarse grid cells/elements (slow)
Fine grid cells/elements (fast)

- **Implicit-Explicit (IMEX) Time Integration:** integrate acoustic terms implicitly, convective terms explicitly
- **Multirate Time Integration:** For AMR simulations, integrate refined elements with a “fast” method

Multirate Time Integration for Atmospheric Flows

Adaptive Mesh Refinement (AMR): Selective refining of the grid based on the solution



Single-rate time-integrator: Time step limited by refined elements

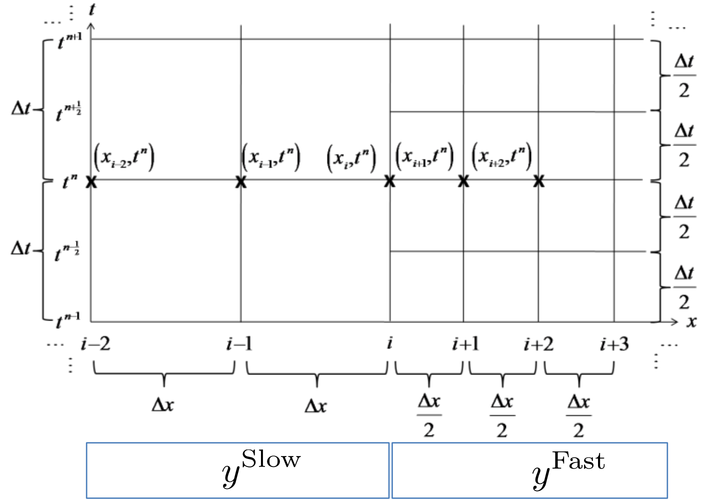


Illustration by: Emil Constantinescu (ANL)

Multirate allows each cell to use the time-step suitable with local grid spacing

$$\begin{bmatrix} \dot{y}^{\mathrm{Slow}} \\ \dot{y}^{\mathrm{Fast}} \end{bmatrix} = \begin{bmatrix} f_1(y^{\mathrm{Slow}}, y^{\mathrm{Fast}}) \\ f_2(y^{\mathrm{Slow}}, y^{\mathrm{Fast}}) \end{bmatrix}$$

Partitioning of the governing equation

$$\dot{y} = f(y)$$

$f^{\mathrm{Slow}}_{\mathrm{Nonstiff}}(y(t))$

Convection on coarse elements

$f^{\mathrm{Fast}}_{\mathrm{Nonstiff}}(y(t))$

Convection on refined elements

$f_{\mathrm{Stiff}}(y(t))$

Acoustic waves on all elements

Extrapolated Methods Overview
Building high-order methods from low-order method

$\mathcal{I}_{\mathrm{LO}}$

Low-order (computationally cheap) time integrator, for example, forward/backward Euler

$T_{11} \equiv y^{n+1}\left(\mathcal{I}_{\mathrm{LO}}\left[\Delta t\right]\right)$				
$T_{21} \equiv y^{n+1}\left(\mathcal{I}_{\mathrm{LO}}\left[\frac{\Delta t}{2}\right]\right)$	T_{22}			
$T_{31} \equiv y^{n+1}\left(\mathcal{I}_{\mathrm{LO}}\left[\frac{\Delta t}{3}\right]\right)$	T_{32}	T_{33}		
$T_{41} \equiv y^{n+1}\left(\mathcal{I}_{\mathrm{LO}}\left[\frac{\Delta t}{4}\right]\right)$	T_{42}	T_{43}	T_{44}	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$

Extrapolation formula: build higher order solutions

$$T_{j,k+1} = T_{j,k} + \frac{T_{j,k} - T_{j-1,k}}{j/\left(j-k\right) - 1}$$

High-order solutions: low order + (column index-1)

Extrapolated Multirate IMEX Methods

First-order IMEX Multirate Method

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f^{\text{Fast}}_{\text{Nonstiff}}(y(s))\,\mathrm{d}s \;\longrightarrow\; \text{EX}^{\text{Slow}}_{\text{Nonstiff}} := \begin{array}{c|cc} 0 & 0 & \\ 0 & 0 & 0 \\ \hline & 0 & 1 \end{array}$$

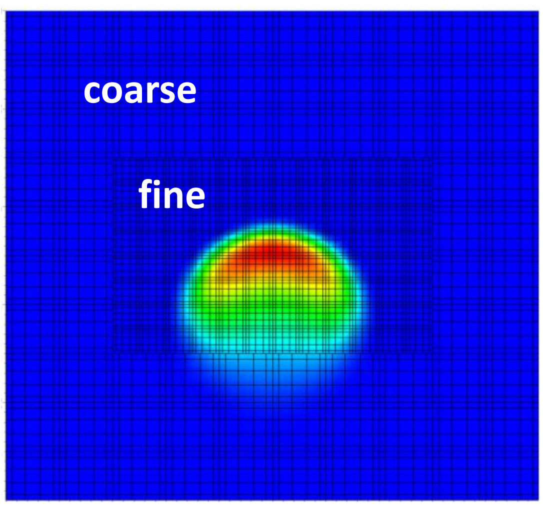
Forward Euler

$$+ \int_{t_n}^{t_{n+1}} f^{\text{Slow}}_{\text{Nonstiff}}(y(s))\,\mathrm{d}s \;\longrightarrow\; \text{EX}^{\text{Fast}}_{\text{Nonstiff}} := \begin{array}{c|cc} 0 & 0 & \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & 0 & 1 \end{array}$$

Two steps of forward Euler disguised as multistage RK

$$+ \int_{t_n}^{t_{n+1}} f_{\text{Stiff}}(y(s))\,\mathrm{d}s \;\longrightarrow\; \text{IM}_{\text{Stiff}} := \begin{array}{c|cc} 0 & 0 & \\ 1 & 0 & 1 \\ \hline & 0 & 1 \end{array}$$

Backward Euler



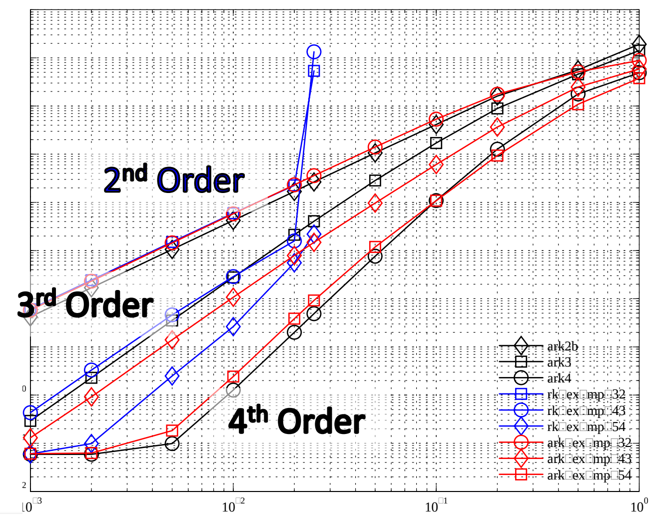
Extended to higher-orders using ***extrapolation***

Refinement levels: 0,1,2

Single-rate-IMEX

Multirate explicit

Multirate IMEX



Thank you.
Questions?

Lawrence Livermore National Laboratory

Balanced, Conservative Finite-Difference Formulation (1)

Governing Equations for 2D flows (gravity acting along –*y* axis)

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{u})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{u})}{\partial y} = \mathbf{s}(\mathbf{u})$$

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ (e+p)u \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ (e+p)v \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} 0 \\ 0 \\ -\rho g \\ -\rho v g \end{bmatrix}$$

Hydrostatically balanced equilibrium

Pressure gradient balanced by gravitational source

$$u = \text{constant}, v = 0, \ \rho = \rho_0 \varrho(y), \ p = p_0 \varphi(y)$$

$$\boxed{\frac{dp}{dy} = -\rho g}$$

Flow variables at reference altitude

$$\boxed{RT_0 \left[\varrho(y)\right]^{-1} \varphi'(y) = -g}$$

Balanced, Conservative Finite-Difference Formulation (2)

Extension of Xing & Shu's method (*J. Sci. Comput., 2013*)

$$\varrho(y) = \exp\left(-\frac{gy}{RT}\right)$$
$$\varphi(y) = \exp\left(-\frac{gy}{RT}\right)$$

Isothermal equilibrium

$$\varrho(y) = \left[1 - \frac{(\gamma-1)gy}{\gamma R\theta}\right]^{1/(\gamma-1)}$$
$$\varphi(y) = \left[1 - \frac{(\gamma-1)gy}{\gamma R\theta}\right]^{\gamma/(\gamma-1)}$$

Constant potential temperature
(Rising thermal bubble)

$$\varrho(y) = \exp\left(-\frac{\mathcal{N}^2}{g}y\right)\left[1 + \frac{(\gamma-1)g^2}{\gamma R T_0 \mathcal{N}^2}\left\{\exp\left(-\frac{\mathcal{N}^2}{g}y\right) - 1\right\}\right]^{1/(\gamma-1)}$$
$$\varphi(y) = \left[1 + \frac{(\gamma-1)g^2}{\gamma R T_0 \mathcal{N}^2}\left\{\exp\left(-\frac{\mathcal{N}^2}{g}y\right) - 1\right\}\right]^{\gamma/(\gamma-1)}$$

Stratified atmosphere with a specified Brunt-Vaïsälä frequency
(Inertia-gravity wave)

Balanced, Conservative Finite-Difference Formulation (3)

Differential Form

$$\frac{\partial \mathbf{G}(\mathbf{u})}{\partial y} = \mathbf{s}(\mathbf{u})$$

Discretized Form

$$\frac{1}{\Delta y}\mathcal{D}\left[\mathbf{G}(\mathbf{u})\right]_j = \left[\mathbf{s}(\mathbf{u})\right]_j$$

Well-Balanced Formulation:

Discretized equilibrium holds **not just for** $\Delta y \to 0$ but for any grid resolution.

Modified Governing Equations

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{u})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{u})}{\partial y} = \mathbf{s}^*(\mathbf{u}, y) \qquad \mathbf{s}^* = \begin{bmatrix} 0 \\ 0 \\ \rho R T_0 \left[\varrho(y)\right]^{-1} \varphi'(y) \\ \rho v R T_0 \left[\varrho(y)\right]^{-1} \varphi'(y) \end{bmatrix}$$

$$\boxed{R T_0 \left[\varrho(y)\right]^{-1} \varphi'(y) = -g \Rightarrow \mathbf{s}^* \equiv \mathbf{s}}$$

$$\mathcal{D}_{\mathbf{G}}\left[p\right] = \rho R T_0 \left\{\varrho(y)\right\}^{-1} \mathcal{D}_{\mathbf{s}^*}\left[\varphi(y)\right]$$

If flux and source discretized by **same operator**

$$\mathcal{D}_{\mathbf{G}} = \mathcal{D}_{\mathbf{s}^*} = \mathcal{D}$$

Finite-Difference Discretization

$$\left.\frac{\partial \phi}{\partial x}\right|_{x=x_j} \approx \mathcal{D}\left[\phi\right] \equiv \sum_{k=-m}^{n} \sigma_k^{\mathcal{D}} \phi_{j+k}$$

$$\Rightarrow \mathcal{D}\left[p - \rho R T_0 \left\{\varrho(y)\right\}^{-1} \varphi(y)\right] = \mathcal{D}\left[p_0 \varphi(y) - \rho_0 \varrho(y) R T_0 \left\{\varrho(y)\right\}^{-1} \varphi(y)\right] = 0$$

Balanced, Conservative Finite-Difference Formulation (4)

	Flux	Source
Interpolation	$\hat{\mathbf{G}}^{L,R}_{j+1/2} = \mathcal{R}^{L,R}_{\mathbf{G}}[\mathbf{G}] \equiv \sum_{k=-m}^{n} \hat{\sigma}_k \mathbf{G}_{j+k}$	$\hat{\varphi}^{L,R}_{j+1/2} = \mathcal{R}^{L,R}_{\mathbf{G}}[\varphi] \equiv \sum_{k=-m}^{n} \hat{\sigma}_k \varphi_{j+k}$
Upwinding (Rusanov)	$\hat{\mathbf{G}}_{j+1/2} = \frac{1}{2}\left[\hat{\mathbf{G}}^{L}_{j+1/2} + \hat{\mathbf{G}}^{R}_{j+1/2}\right] + \frac{1}{2}\max_{j,j+1} \nu_j \left(\hat{\mathbf{u}}^{L}_{j+1/2} - \hat{\mathbf{u}}^{R}_{j+1/2}\right)$	$\hat{\varphi}_{j+1/2} = \frac{1}{2}\left[\hat{\varphi}^{L}_{j+1/2} + \hat{\varphi}^{R}_{j+1/2}\right]$
Differencing	$\left.\frac{\partial \mathbf{G}}{\partial y}\right\vert_{y=y_j} \approx \frac{1}{\Delta y}\left[\hat{\mathbf{G}}_{j+1/2} - \hat{\mathbf{G}}_{j-1/2}\right]$	$\left.\frac{\partial \varphi}{\partial y}\right\vert_{y=y_j} \approx \frac{1}{\Delta y}\left[\hat{\varphi}_{j+1/2} - \hat{\varphi}_{j-1/2}\right]$

$$\mathcal{R}^{L,R}_{\mathbf{G}}[\phi] \equiv \sum_{k=-m}^{n} \hat{\sigma}_k \phi_{j+k}$$

Represents the **WENO/CRWENO** finite-difference operator with the non-linear weights computed based on **G(u)**

Diffusion term in upwinding must vanish for equilibrium solution

$$\hat{\mathbf{G}}_{j+1/2} = \frac{1}{2}\left[\hat{\mathbf{G}}^{L}_{j+1/2} + \hat{\mathbf{G}}^{R}_{j+1/2}\right] + \frac{1}{2}\max_{j,j+1} \nu_j \left(\hat{\mathbf{u}^*}^{L}_{j+1/2} - \hat{\mathbf{u}^*}^{R}_{j+1/2}\right)$$

$$\mathbf{u}^* = \begin{bmatrix} \rho\left\{\varrho(y)\right\}^{-1} \\ \rho u \left\{\varrho(y)\right\}^{-1} \\ \rho v \left\{\varrho(y)\right\}^{-1} \\ \frac{p\{\varphi(y)\}^{-1}}{\gamma-1} + \frac{1}{2}\rho\left\{\varrho(y)\right\}^{-1}\left(u^2+v^2\right) \end{bmatrix}$$

Constant at steady state

Verification of Balanced Formulation

Case 1: Isothermal equilibrium

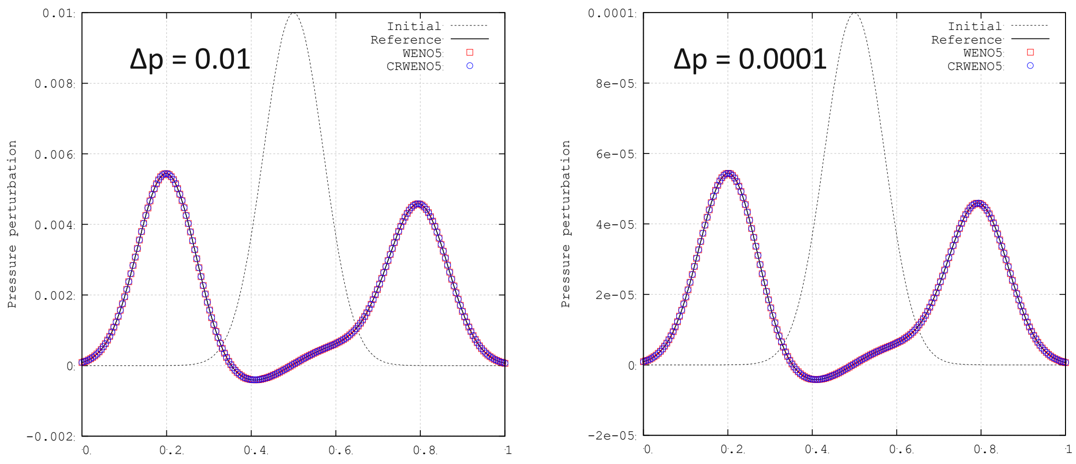
Case 2: Stratified atmosphere with constant potential temperature

Case 3: Stratified atmosphere with specified Brunt-Vaïsälä frequency

Difference of the final solution with the initial solution

(Verification that hydrostatic balance is preserved to machine precision)

Case	L_1	L_2	L_∞	L_1	L_2	L_∞
	WENO5			CRWENO5		
Case 1	2.46E-15	2.89E-15	3.91E-15	2.00E-14	1.71E-14	1.50E-14
Case 2	6.02E-15	7.11E-15	1.31E-14	1.50E-14	1.53E-14	2.09E-14
Case 3	3.63E-15	4.35E-15	8.15E-15	1.58E-14	1.83E-14	6.11E-14



Small perturbation to isothermal hydrostatic balance

- Algorithm is able to **preserve the hydrostatic balance** at any grid resolution and for any duration to machine precision.
- **Small perturbations** to the hydrostatic balance are **accurately resolved.**