Flexible Time Integration Methods for Multiphysics PDE Systems

Daniel R. Reynolds[1]

Collaborators: Alex Fish[1], Rujeko Chinomona[2], Vu Thai Luan[3],
Steven Roberts[4], David Gardner[4], Carol Woodward[4], Cody Balos[4]

[1]Department of Mathematics, Southern Methodist University
[2]Department of Mathematics, Temple University
[3]Department of Mathematics, Mississippi State University
[4]Center for Applied Scientific Computing, Lawrence Livermore National Laboratory

SIAM Central States Computational and Applied Mathematics Forum
13 November 2023

Outline

1. Background
2. IMEX-MRI Methods
3. MRI Time Adaptivity
4. Software
5. Conclusions, etc.

Outline

1. Background
2. IMEX-MRI Methods
3. MRI Time Adaptivity
4. Software
5. Conclusions, etc.

Multiphysics Simulations

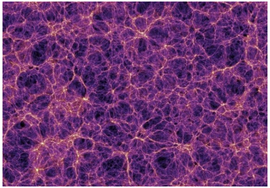
Multiphysics simulations couple different models either in the bulk or across interfaces.

Climate:

- Atmospheric simulations combine fluid dynamics with local "physics" models for chemistry, condensation, ...
- Atmosphere is coupled at interfaces to myriad other processes (ocean, land/sea ice, . . .), each using distinct models.

Above: https://e3sm.org.

Below: http://svs.gsfc.nasa.gov.



Astrophysics/cosmology:

- Dark matter modeled using particles that give rise to large-scale gravitational structures (at right).
- Baryonic matter modeled by combining fluid dynamics, gravity, radiation transport, and reaction networks for chemical ionization states.

Multiphysics Challenges

[Keyes et al., 2013]

These model combinations can challenge traditional numerical methods:

- "Multirate" processes evolve on different time scales but prohibit analytical reformulation.
- Stiff components disallow fully explicit methods.
- Nonlinearity and insufficient differentiability challenge fully implicit methods.
- Parallel scalability demands optimal algorithms – while robust/scalable algebraic solvers exist for parts (e.g., FMM for particles, multigrid for diffusion), none are optimal for the whole.

We may consider a prototypical problem as having m coupled evolutionary processes:

$$\dot{y}(t) = f^{\{1\}}(t, y) + \cdots + f^{\{m\}}(t, y), \quad t \in [t_0, t_f], \quad y(t_0) = y_0.$$

Each component $f^{\{k\}}(t, y)$:

- may act on all of y (in the bulk), or on only a subset of y (within a subdomain),
- may evolve on a different characteristic time scale,
- may be "stiff" or "nonstiff," thereby desiring implicit or explicit treatment.

Implicit-Explicit Additive Runge–Kutta Methods

[Ascher et al., 1997; Kennedy & Carpenter, 2003; ...]

IMEX-ARK methods allow high-order adaptive ImEx time integration for additively-split *single rate* simulations:

$$\dot{y}(t) = f^E(t,y) + f^I(t,y), \quad t \in [t_0, t_f], \quad y(t_0) = y_0,$$

- $f^E(t,y)$ contains the nonstiff terms to be treated explicitly,
- $f^I(t,y)$ contains the stiff terms to be treated implicitly.

Combine two s-stage RK methods; denoting $h_n = t_{n+1} - t_n$, $t^E_{n,j} = t_n + c^E_j h_n$, $t^I_{n,j} = t_n + c^I_j h_n$:

$$z_i = y_n + h_n \sum_{j=1}^{i-1} a^E_{i,j} f^E(t^E_{n,j}, z_j) + h_n \sum_{j=1}^{i} a^I_{i,j} f^I(t^I_{n,j}, z_j), \quad i = 1, \ldots, s,$$

$$y_{n+1} = y_n + h_n \sum_{j=1}^{s} \left[b^E_j f^E(t^E_{n,j}, z_j) + b^I_j f^I(t^I_{n,j}, z_j) \right] \quad \text{(solution)}$$

$$\tilde{y}_{n+1} = y_n + h_n \sum_{j=1}^{s} \left[\tilde{b}^E_j f^E(t^E_{n,j}, z_j) + \tilde{b}^I_j f^I(t^I_{n,j}, z_j) \right] \quad \text{(embedding)}$$

Solving each stage z_i, $i = 1, \ldots, s$ [Ascher et al., 1997; Kennedy & Carpenter, 2003; ...]

Per-stage cost is commensurate with implicit Euler for $\dot{y}(t) = f^I(t, y)$ – solve a root-finding problem:

$$0 = G_i(z) = \left[z - h_n a^I_{i,i} f^I(t^I_{n,i}, z)\right] - \left[y_n + h_n \sum_{j=1}^{i-1} \left(a^E_{i,j} f^E(t^E_{n,j}, z_j) + a^I_{i,j} f^I(t^I_{n,j}, z_j)\right)\right]$$

- If $f^I(t, y)$ is *linear* in y then this is a large-scale linear system for each z_i.
- Else this requires an iterative solver (e.g., Newton, accelerated fixed-point, or problem-specific).
- All operators in $f^E(t, y)$ are treated explicitly (do not affect algebraic solver convergence).

Defined by compatible *explicit* $\left\{c^E, A^E, b^E, \tilde{b}^E\right\}$ and *implicit* $\left\{c^I, A^I, b^I, \tilde{b}^I\right\}$ tables. These are derived in unison to satisfy order conditions arising from NB-trees (along with stability, high stage order, ...).

Multirate Infinitesimal (MRI) methods

[Schlegel et al., 2009; Sandu, 2019; . . .]

MRI methods provide a flexible approach to "subcycling" and support up to $\mathcal{O}(H^6)$ for multirate problems:

$$\dot{y}(t) = f^S(t,y) + f^F(t,y), \quad t \in [t_0, t_f], \quad y(t_0) = y_0.$$

- $f^S(t,y)$ contains the "slow" dynamics, evolved with time step H.
- $f^F(t,y)$ contains the "fast" dynamics, evolved with time steps $h \ll H$.
- The slow component is defined by an "outer" RK method, while the fast component is advanced between slow stages by solving a modified IVP with a subcycled "inner" RK method.
- Extremely efficient – fourth order is attainable with *only a single traversal* of $[t_n, t_{n+1}]$.

MRI Algorithm Outline

[Schlegel et al., 2009; Sandu, 2019; . . .]

Denoting $y_n \approx y(t_n)$ and $H = t_{n+1} - t_n$, a single step $y_n \to y_{n+1}$ proceeds as follows:

1. Let: $z_1 = y_n$.
2. For each slow stage z_i, $i = 2, \ldots, s$:
   a) Define: $r_i(\tau) = \sum_{j=1}^{i} \gamma_{i,j} \left(\frac{\tau}{(c_i - c_{i-1})H}\right) f^S\left(t_n + c_j H, z_j\right)$.
   b) Evolve: $\dot{v}(\tau) = f^F\left(t_n + \tau, v\right) + r_i(\tau)$, for $\tau \in [c_{i-1}H, c_i H]$, $v(c_{i-1}H) = z_{i-1}$.
   c) Let: $z_i = v(c_i H)$.
3. Let: $y_{n+1} = z_s$.

- MIS: $\gamma_{i,j}(\theta)$ is independent of θ, with coefficients computed from an "outer" RK method [Schlegel et al., 2009].
- MRI: $\gamma_{i,j}(\theta)$ is polynomial in θ, coefficients satisfy GARK-based order conditions [Sandu, 2019].
- Step 2b may use any applicable algorithm of sufficient accuracy (including another MRI method).
- When $c_i = c_{i-1}$, step 2b reduces to a standard ERK/DIRK Runge–Kutta stage update.
- Implicitness at the slow scale depends on $\gamma_{i,i}(\theta) \neq 0$, only used when $c_i = c_{i-1}$ ("solve-decoupled").

Outline

1. Background
2. IMEX-MRI Methods
3. MRI Time Adaptivity
4. Software
5. Conclusions, etc.

Implicit-Explicit Multirate Infinitesimal GARK Methods

[Chinomona & R., *SISC*, 2021]

To better support the flexibility needs of multiphysics problems, we have extended Sandu's MRI-GARK methods to support implicit-explicit treatment of the slow time scale, for problems of the form:

$$\dot{y}(t) = f^I(t,y) + f^E(t,y) + f^F(t,y), \quad t \in [t_0, t_f], \quad y(t_0) = y_0.$$

These follow the same basic approach as the previous MRI algorithm, but with in ImEx forcing function

$$r_i(\tau) = \sum_{j=1}^{i} \gamma_{i,j}\left(\tfrac{\tau}{(c_i - c_{i-1})H}\right) f^I(t_n + c_j H, z_j) + \sum_{j=1}^{i-1} \omega_{i,j}\left(\tfrac{\tau}{(c_i - c_{i-1})H}\right) f^E(t_n + c_j H, z_j),$$

where $\gamma_{i,j}(\theta) := \sum_{k=0}^{k_{max}} \gamma_{i,j}^{\{k\}} \theta^k$ and $\omega_{i,j}(\theta) := \sum_{k=0}^{k_{max}} \omega_{i,j}^{\{k\}} \theta^k$.

- Coefficients matrices $\Gamma^{\{k\}}, \Omega^{\{k\}} \in \mathbb{R}^{s\times s}$ are lower and strictly lower triangular, respectively.
- Order conditions up to $\mathcal{O}(H^4)$ leverage GARK framework.

IMEX-MRI-GARK Construction

[Chinomona & R., *SISC*, 2021]

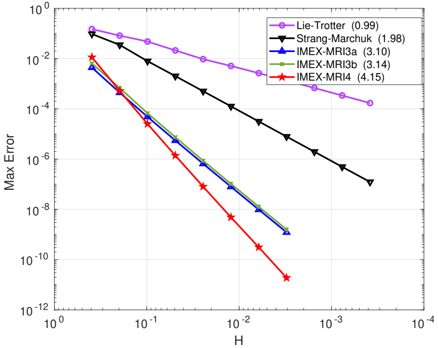
Begin with an IMEX-ARK pair $\{A^I, b^I, c^I; A^E, b^E, c^E\}$ where $c^I = c^E \equiv c$ with $0 = c_1 \le \cdots \le c_{\tilde{s}} \le 1$.

- Convert to solve-decoupled form: insert redundant stages such that $\Delta c_i A^I_{ii} = 0$ for $i = 1, \ldots, s$.
- Extend A^I, A^E and c to ensure "stiffly-accurate" condition: $c_s = 1$, $A^I_{s,:} = b^I$, $A^E_{s,:} = b^E$.
- Generate $\Gamma^{(k)}$ and $\Omega^{(k)}$ for $k = 0, \ldots, k_{max}$, to satisfy ARK consistency (s^2 conditions), internal consistency ($2s(k_{max}+1)$ conditions), plus order conditions:
  - $\mathcal{O}(H^1)$ and $\mathcal{O}(H^2)$: no additional order conditions,
  - $\mathcal{O}(H^3)$: 2 additional order conditions,
  - $\mathcal{O}(H^4)$: 16 additional order conditions.
- With any additional degrees of freedom, we maximized "joint linear stability".

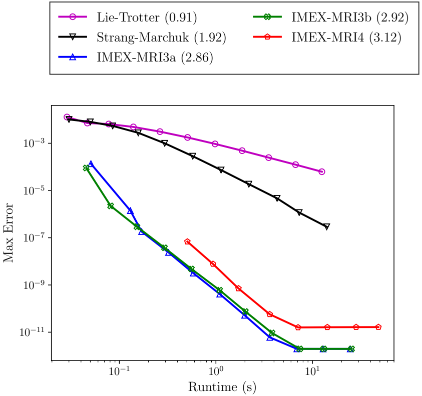
Note: we found it challenging to construct embedded IMEX-MRI-GARK methods, largely due to our reliance on IMEX-ARK base methods and the "sorted" abscissa requirement.

IMEX-MRI-GARK Convergence/Efficiency

[Chinomona & R., *SISC*, 2021]



Nonlinear Kværnø-Prothero-Robinson test problem convergence.



Stiff brusselator PDE test runtime efficiency.

$H = \left\{\frac{1}{40}, \frac{1}{80}\right\}$ runs were unstable for IMEX-MRI4.

Implicit-Explicit Multirate Infinitesimal Stage-Restart Methods

[Fish, R., & Roberts, *JCAM*, 2024]

To circumvent the constraints required for IMEX-MRI-GARK construction, we developed IMEX-MRI-SR methods by assuming a simpler structure for the step $y_n \to y_{n+1}$:

1. Let: $z_1 = y_n$.
2. For each slow stage z_i, $i = 2, \ldots, s$:
   a) Define: $r_i(\tau) = \frac{1}{c_i} \sum_{j=1}^{i-1} \omega_{i,j}\left(\frac{\tau}{c_i H}\right)\left(f_j^E + f_j^I\right)$, with $\omega_{i,j}(\theta) = \sum_{k=0}^{n_\Omega - 1} \omega_{i,j}^{\{k\}} \theta^k$.
   b) Evolve: $\dot{v}(\tau) = f^F\left(t_n + \tau, v\right) + r_i(\tau)$, for $\tau \in [0, c_i H]$, $v(0) = y_n$.
   c) Solve: $z_i = v(c_i H) + H \sum_{j=1}^{i} \gamma_{i,j} f^I(t_n + c_j H, z_j)$.
3. Let: $y_{n+1} = z_s$.

- For brevity above, we denote $f_j^E := f^E(t_n + c_j H, z_j)$ and $f_j^I := f^I(t_n + c_j H, z_j)$.
- The embedding has an identical structure as the last stage, z_s.
- There is no "hidden" dependence on $\Delta c_i = 0$ for the algorithm structure, and no "padding" is required when deriving IMEX-MRI-SR methods from IMEX-ARK.

IMEX-MRI-SR Construction

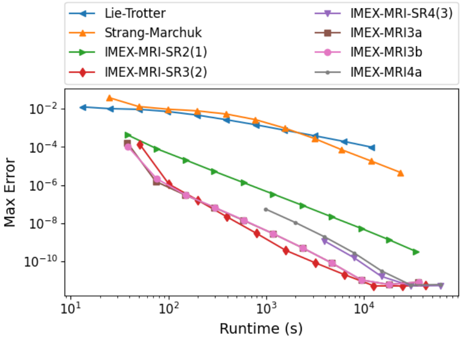
[Fish, R., & Roberts, *JCAM*, 2024]

Again take an an IMEX-ARK pair $\{A^I, b^I, c^I; A^E, b^E, c^E\}$ where $c^I = c^E \equiv c$ (not necessarily sorted).

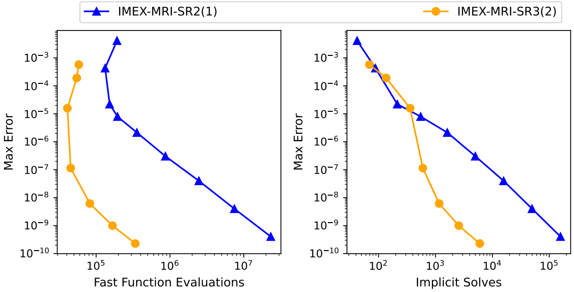
- Extend A^I, A^E and c to ensure "stiffly-accurate" condition: $c_s = 1$, $A^I_{s,:} = b^I$, $A^E_{s,:} = b^E$.
- Generate Γ and $\Omega^{(k)}$ for $k = 0, \ldots, n_\Omega$, to satisfy IMEX-ARK consistency (s^2 conditions), internal consistency ($s(2 + n_\Omega)$ conditions), plus order conditions:
  - $\mathcal{O}(H^1)$ and $\mathcal{O}(H^2)$: no additional order conditions,
  - $\mathcal{O}(H^3)$: 1 additional order condition,
  - $\mathcal{O}(H^4)$: 6 additional order conditions.
- With remaining degrees of freedom, maximize joint linear stability for the method and minimize leading order error for embedding.

IMEX-MRI-SR Convergence/Efficiency – “Stiff” Brusselator PDE [Fish, R., & Roberts, *JCAM*, 2024]

Runtime efficiency of IMEX-MRI-SR, IMEX-MRI-GARK, and IMEX-MRI versions of Lie–Trotter and Strang–Marchuk splittings:



Modified problem with time-dependent advection, diffusion and reaction coefficients. We explore adaptive IMEX-MRI-SR efficiency using tolerances 10^{-k} with $k = 1, \ldots, 9$ (more on MRI adaptivity in a moment):



Outline

1. Background
2. IMEX-MRI Methods
3. MRI Time Adaptivity
4. Software
5. Conclusions, etc.

Multirate Infinitesimal Time Step Adaptivity

[Fish & R., *SISC*, 2023]

As with single-rate IVPs, robustness, accuracy, and efficiency hinge on appropriate selection of time step sizes. In the MRI setting, this is complicated:

- We now have separate control parameters at each time scale (H and h):
- The overall solution error is not simply the sum of errors at fast and slow time scales, since errors may propagate between them.
- With two parameters, we need separate estimates of temporal errors that arise at each scale.
- Although significant work has been performed on single-rate controllers, multirate control has received little investigation (particularly higher-order controller methods).

Multirate control

[Fish & R., *SISC*, 2023]

Denoting the overall error in an MRI time step solution as ε_{n+1}, we estimate

$$\begin{aligned}\varepsilon_{n+1} &= \|y(t_{n+1}) - y_{n+1}\| \\ &\le \|y(t_{n+1}) - y^*_{n+1}\| + \|y^*_{n+1} - y_{n+1}\| \\ &= \varepsilon^s_{n+1} + \varepsilon^f_{n+1} \\ &= \left(\phi^s_n H_n^P + \mathcal{O}\left(H_n^{P+1}\right)\right) + \left(\phi^f_n \left(\frac{H_n}{M_n}\right)^p H_n + \mathcal{O}\left(\left(\frac{H_n}{M_n}\right)^{p+1} H_n\right)\right),\end{aligned}$$

where

- y^*_{n+1} is the imagined solution wherein each fast IVP is solved exactly,
- P and p are the global orders of accuracy for the MRI method embedding and fast solver embedding, resp.,
- ϕ^s and ϕ^f are the principal error functions for each scale (these depend on method and IVP, but not on H_n or M_n).

Multirate controllers

[Fish & R., *SISC*, 2023]

Extending the single-rate approach for controller derivation by Gustafsson [*ACM TOMS*, 1994], we:

- set the desired fast and slow errors to separate tolerances $\varepsilon^f_{n+1} \approx \mathtt{TOL}^f$ and $\varepsilon^s_{n+1} \approx \mathtt{TOL}^s$,
- solve the previous asymptotic estimates for $\log(H_n)$ and $\log(M_n)$, and
- approximate the principal error functions $\log(\phi^f_n)$ and $\log(\phi^s_n)$ using piecewise polynomials,

to derive the *constant-constant* controller

$$H_{n+1} = H_n \left(\frac{\mathtt{TOL}^s}{\varepsilon^s_{n+1}}\right)^{\alpha}, \qquad M_{n+1} = M_n \left(\frac{\mathtt{TOL}^s}{\varepsilon^s_{n+1}}\right)^{\beta_1} \left(\frac{\mathtt{TOL}^f}{\varepsilon^f_{n+1}}\right)^{\beta_2},$$

and the *linear-linear* controller

$$H_{n+1} = H_n \left(\frac{H_n}{H_{n-1}}\right) \left(\frac{\mathtt{TOL}^s}{\varepsilon^s_{n+1}}\right)^{\alpha_1} \left(\frac{\mathtt{TOL}^s}{\varepsilon^s_{n}}\right)^{\alpha_2},$$

$$M_{n+1} = M_n \left(\frac{M_n}{M_{n-1}}\right) \left(\frac{\mathtt{TOL}^s}{\varepsilon^s_{n+1}}\right)^{\beta_{11}} \left(\frac{\mathtt{TOL}^s}{\varepsilon^s_{n}}\right)^{\beta_{12}} \left(\frac{\mathtt{TOL}^f}{\varepsilon^f_{n+1}}\right)^{\beta_{21}} \left(\frac{\mathtt{TOL}^f}{\varepsilon^f_{n}}\right)^{\beta_{22}}.$$

Multirate controllers (continued)

[Fish & R., *SISC*, 2023]

We additionally propose two additional controllers:

- *PIMR* is a multirate extension of the PI single-rate controller:

$$H_{n+1} = H_n \left(\frac{\mathtt{TOL}^s}{\varepsilon^s_{n+1}}\right)^{\alpha_1} \left(\frac{\mathtt{TOL}^s}{\varepsilon^s_{n}}\right)^{\alpha_2},$$

$$M_{n+1} = M_n \left(\frac{\mathtt{TOL}^s}{\varepsilon^s_{n+1}}\right)^{\beta_{11}} \left(\frac{\mathtt{TOL}^s}{\varepsilon^s_{n}}\right)^{\beta_{12}} \left(\frac{\mathtt{TOL}^f}{\varepsilon^f_{n+1}}\right)^{\beta_{21}} \left(\frac{\mathtt{TOL}^f}{\varepsilon^f_{n}}\right)^{\beta_{22}}.$$

- *PIDMR* is a multirate extension of the PID single-rate controller:

$$H_{n+1} = H_n \left(\frac{\mathtt{TOL}^s}{\varepsilon^s_{n+1}}\right)^{\alpha_1} \left(\frac{\mathtt{TOL}^s}{\varepsilon^s_{n}}\right)^{\alpha_2} \left(\frac{\mathtt{TOL}^s}{\varepsilon^s_{n-1}}\right)^{\alpha_3},$$

$$M_{n+1} = M_n \left(\frac{\mathtt{TOL}^s}{\varepsilon^s_{n+1}}\right)^{\beta_{11}} \left(\frac{\mathtt{TOL}^s}{\varepsilon^s_{n}}\right)^{\beta_{12}} \left(\frac{\mathtt{TOL}^s}{\varepsilon^s_{n-1}}\right)^{\beta_{13}} \left(\frac{\mathtt{TOL}^f}{\varepsilon^f_{n+1}}\right)^{\beta_{21}} \left(\frac{\mathtt{TOL}^f}{\varepsilon^f_{n}}\right)^{\beta_{22}} \left(\frac{\mathtt{TOL}^f}{\varepsilon^f_{n-1}}\right)^{\beta_{23}}.$$

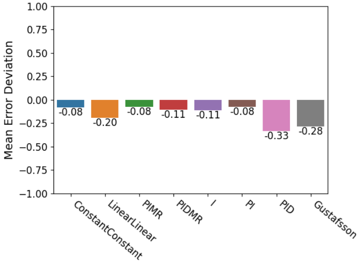
[Approaches for estimating ε^f_n and ε^s_n (time-permitting)]

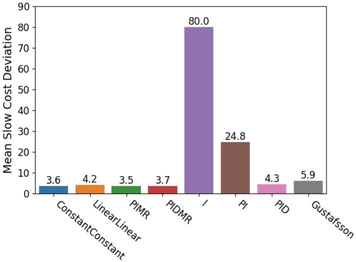
MRI controller performance

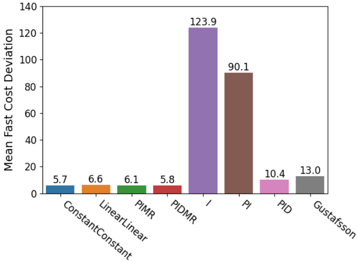
[Fish & R., *SISC*, 2023]

Tested 4 MRI controllers along with 4 single-rate H controllers (each used a fixed $M = 10$), across a test suite of 7 test problems, 4 IVP methods, and 3 tolerances.

- Left: overall controller ability to achieve desired tolerance ($0 \Rightarrow$ perfect, $< 0 \Rightarrow$ overly accurate)
- Center: overall controller f^S cost as multiple of "best possible" (i.e., $1 \Rightarrow$ perfect)
- Right: overall controller f^F cost as multiple of "best possible"







Outline

1 Background

2 IMEX-MRI Methods

3 MRI Time Adaptivity

4 **Software**

5 Conclusions, etc.

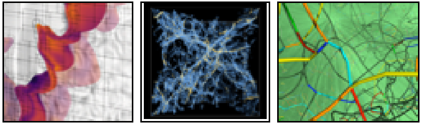
Open Source Software

[Gardner et al., *TOMS*, 2022; Balos et al., *ParComp*, 2021; Hindmarsh et al., *TOMS*, 2005]

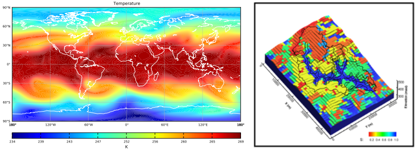
I believe that the importance of applied math research lies in its *impact* on other fields. My collaborators and I achieve impact both through publications and high-quality open-source software.

We develop and maintain *SUNDIALS* – the SUite of Nonlinear and DIfferential/ALgebraic Solvers

- Adaptive time integrators for ODEs and DAEs and efficient nonlinear solvers, used in applications worldwide throughout research and industry.
- Written in C/C++, natively supports parallel computing:
  - Clusters/supercomputers via MPI
  - GPUs via CUDA, HIP, SYCL, RAJA and Kokkos
  - Threading via OpenMP and Pthreads
- BSD license; available from GitHub or Spack.
- Over 160,000 downloads in 2022.
- Runs anywhere from laptops to the largest supercomputers in the world.



Combustion (Pele), Cosmology (Nyx), Materials science (ParaDiS)



Climate (Tempest), Subsurface flow (ParFlow)

Awarded the 2023 SIAM/ACM Prize in Computational Science & Engineering

Software: ARKODE

[R. et al., *ACM TOMS*, 2023]

Our "flexible" integrators reside in ARKODE, initially released within SUNDIALS in 2014 with adaptive IMEX-ARK methods. Since then we have enhanced it to include a variety of "steppers":

- `ARKStep`: supports all of ARKODE's original functionality (adaptive ARK, ERK, DIRK methods); includes an interface to XBraid for PinT (work by D. Gardner).
- `ERKStep`: tuned for highly efficient explicit Runge–Kutta methods.
- `MRIStep`: multirate infinitesimal time stepping module.
  - Includes explicit MIS methods $\mathcal{O}(H^3)$, explicit or implicit MRI-GARK methods of $\mathcal{O}(H^2)$ to $\mathcal{O}(H^4)$, IMEX-MRI-GARK methods of $\mathcal{O}(H^3)$ and $\mathcal{O}(H^4)$.
  - Supports user-provided MRI-GARK tables $\Gamma^{\{k\}}$ or IMEX-MRI-GARK tables $\{\Gamma^{\{k\}}, \Omega^{\{k\}}\}$.
  - Currently requires a user-defined H that can be varied between steps. Fast time scale evolved using `ARKStep` or any viable user-supplied IVP solver.
  - *Will soon include embedded IMEX-MRI-SR methods of $\mathcal{O}(H^2)$ to $\mathcal{O}(H^4)$, and multirate time adaptivity controllers.*

[Large-scale multirate reacting flow results (time permitting)]

Outline

1 Background

2 IMEX-MRI Methods

3 MRI Time Adaptivity

4 Software

5 Conclusions, etc.

Conclusions

Large-scale multiphysics problems:

- Nonlinear, interacting models pose key challenges to stable, accurate and scalable simulation.
- Large data requirements require scalable solvers; while individual processes admit "optimal" algorithms & time scales, these rarely agree.
- Most classical methods derived for idealized problems perform poorly on "real world" applications.

Although operator-spliting remains standard, new & flexible methods are catching up, supporting high order accuracy (even up to $\mathcal{O}(H^6)$) and multirate/ImEx flexibility.

The optimal choice of method depends on a variety of factors:

- whether the problem admits a natural and effective ImEx and/or multirate splitting,
- relative costs of $f^S(t,y)$ and $f^F(t,y)$ for multirate; availability of optimal algebraic solvers for $f^I(t,y)$,
- desired solution accuracy, ...

Future Work

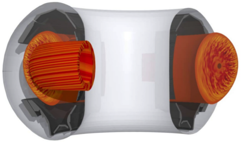
Much work remains to be done:

- Improved [embedded] IMEX-MRI-GARK and IMEX-MRI-SR methods (particulary for $\mathcal{O}(H^4)$).
- Robust temporal controllers for nested multirating, $h_1 > h_2 > \cdots > h_m$.
- Extension of flexible integration approaches to large-scale systems of DAEs.
- Robust (automated?) approaches for determining additive splittings $f(t,y) \to \sum_k f^{\{k\}}(t,y)$.
- Support for a broad range of adaptive MRI methods within open-source software libraries.
- Close collaboration with applications scientists to incorporate novel algorithms within their codes.
- Suggestions?

Postdoctoral Positions in Numerical Methods for Fusion Energy

I'm recruiting two postdocs to work on the development and implementation of advanced time integration methods for large-scale simulations in magnetic fusion energy.

- Looking for candidates with expertise in one or more of:
  - high-performance computing,
  - numerical analysis, and
  - simulation of differential equations.
- Competitive salary (including benefits).
- Initial appointment is for 1 year (renewable annually up to 4).
- Funded by DOE SciDAC partnership program (ASCR & FES).
- Contact me at reynolds@smu.edu with any questions or interest.
- Information and application at MathJobs.org.



We're also hiring two data science postdocs as part of our department's RTG (also on MathJobs.org).

Funding & Computing Support

This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) Program through the FASTMath Institute, under Lawrence Livermore National Laboratory subcontract B626484 and DOE award DE-SC0021354.

This research was supported in part by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

Outline

6 Appendix

IMEX-MRI-GARK Joint Linear Stability

[Chinomona & R., *SISC*, 2021]

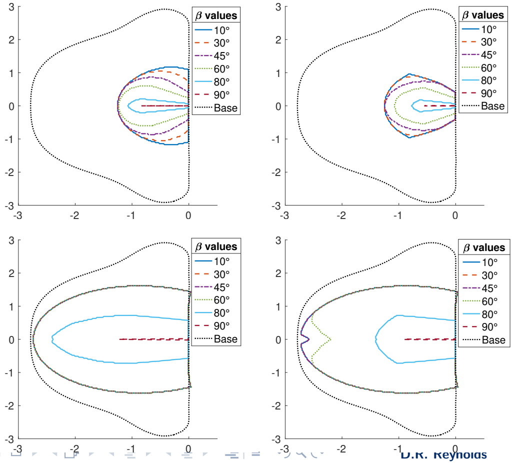
Multirate method stability is currently difficult to analyze. Examining "joint stability" [Zharovsky et al., 2015] for the Dahlquist-like test problem $\dot{y} = \lambda^I y + \lambda^E y + \lambda^F y$:

$$\mathcal{J}_{\alpha,\beta} = \left\{ z^E \in \mathbb{C}^- : \left| R\left(z^F, z^E, z^I\right) \right| \le 1, \forall z^F \in \mathcal{S}^F_\alpha, \forall z^I \in \mathcal{S}^I_\beta \right\}, \quad \mathcal{S}^\sigma_\alpha = \left\{ z^\sigma \in \mathbb{C}^- : |\arg(z^\sigma) - \pi| \le \alpha \right\}$$

$\mathcal{J}_{\alpha,\beta}$ regions for various implicit sector angles β:

- IMEX-MRI-GARK3a ($\uparrow$)
- IMEX-MRI-GARK3b ($\downarrow$)
- fast $\alpha = 10^o$ ($\leftarrow$)
- fast $\alpha = 45^o$ ($\rightarrow$)

We have an $\mathcal{O}(H^4)$ IMEX-MRI-GARK4 table for convergence verification, though it has poor joint stability.



[Back to IMEX-MRI-GARK construction]

IMEX-MRI-SR Joint Linear Stability

[Fish, R., & Roberts, *JCAM*, 2024]

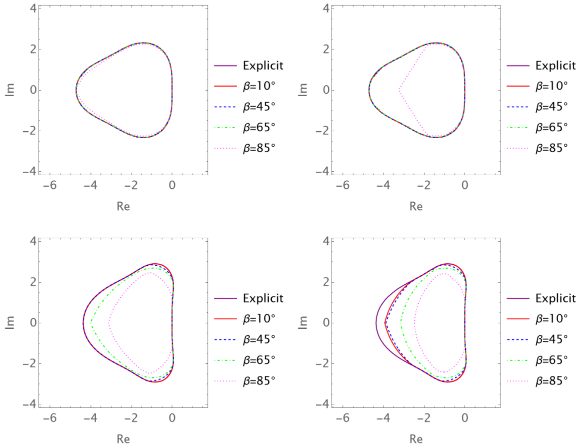
We again analyze joint linear stability for the Dahlquist-like test problem $\dot{y} = \lambda^I y + \lambda^E y + \lambda^F y$:

$$\mathcal{J}_{\alpha,\beta} = \left\{ z^E \in \mathbb{C}^- : \left| R\left(z^F, z^E, z^I\right) \right| \le 1, \forall z^F \in \mathcal{S}^F_\alpha, \forall z^I \in \mathcal{S}^I_\beta \right\}, \quad \mathcal{S}^\sigma_\alpha = \left\{ z^\sigma \in \mathbb{C}^- : |\arg(z^\sigma) - \pi| \le \alpha \right\}$$

$\mathcal{J}_{\alpha,\beta}$ regions for various implicit sector angles β:

- IMEX-MRI-SR2(1) ($\uparrow$)
- IMEX-MRI-SR3(2) ($\downarrow$)
- fast $\alpha = 10^o$ ($\leftarrow$)
- fast $\alpha = 45^o$ ($\rightarrow$)

We have an $\mathcal{O}(H^4)$ IMEX-MRI-SR4(3) table for convergence verification, though it again has relatively poor joint stability.



[Back to IMEX-MRI-SR construction]

MRI error estimation

[Fish & R., *SISC*, 2023]

All controllers require accurate/cheap estimates for ε_n^s and ε_n^f. Assuming the MRI method provides an embedding, $\tilde{y}_n$, then $\varepsilon_n^s \approx \|y_n - \tilde{y}_n\|$. However, estimation of $\varepsilon_n^f = \|y_n^* - y_n\|$ is less obvious.

We tested a variety of strategies:

- *Full-Step (FS)* – compute each fast solve twice using fast integrators of different orders, with forcing functions $r_i(\tau)$ that use separate $f^S(t,y)$ evaluations, to obtain $\varepsilon_n^f = \|y_n - \hat{y}_n\|$.
- *Stage-Aggregate (SA)* – compute each fast solve twice using fast integrators of different orders, but with forcing functions $r_i(\tau)$ that use shared $f^S(t,y)$ evaluations, and aggregate stage differences to obtain $\varepsilon_n^f = \text{aggregate}\left(\|z_i - \hat{z}_i\|, i = 2, \ldots, s\right)$.
- *Local-Accumulation-Stage-Aggregate (LASA)* – compute each fast solve once using an embedded method, and accumulate sub-step error estimates $d_{i,j}$ into an overall estimate $\varepsilon_n^f = \text{aggregate}\left(\sum_{j=1}^{M} d_{i,j}, i = 2, \ldots, s\right)$.

In the end, the "LASA" strategies proved sufficiently accurate (with the least expense).

[Back to MRI controllers]

Multirate reacting flow demonstration problem

[R. et al., *ACM TOMS*, 2023]

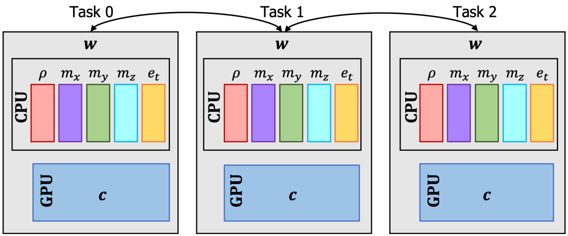
3D nonlinear compressible Euler equations combined with stiff chemical reactions for a low-density primordial gas (molecular & ionization states of H and He, free electrons, and internal gas energy), present in models of the early universe.

$$\partial_t \mathbf{w} = -\nabla \cdot \mathbf{F}(\mathbf{w}) + \mathbf{R}(\mathbf{w}), \quad \mathbf{w}(t_0) = \mathbf{w}_0,$$

$\mathbf{w}$: density, momenta, total energy, and chemical densities (10)
$\mathbf{F}$: advective fluxes (nonstiff/slow); and $\mathbf{R}$: reaction network (stiff/fast)

$\mathbf{w}$ is stored as an `MPIManyVector`:

- Software layer treating collection of vector objects as a single cohesive vector.
- Does not touch any vector data directly.
- Simplifies partitioning of data among computational resources (e.g., CPU vs GPU).
- May also combine distinct MPI intracommunicators together in a multiphysics simulation.



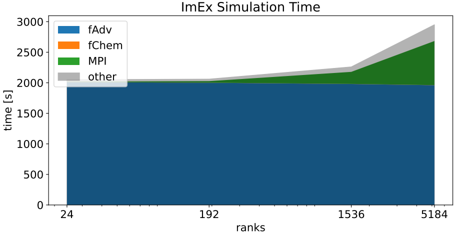
Fluid species (density, momenta, total energy) are stored in main memory, while chemical densities are stored in GPU memory.

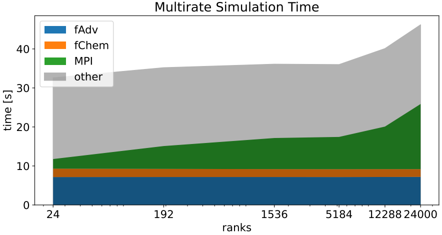
Multirate reacting flow solver strategy

[R. et al., *ACM TOMS*, 2023]

- Method of lines: $(X,t) \in \Omega \times (t_0, t_f]$, with $\Omega = [x_l, x_r] \times [y_l, y_r] \times [z_l, z_r]$.
- Regular $n_x \times n_y \times n_z$ grid for Ω, parallelized using standard 3D MPI domain decomposition.
- $\mathcal{O}\left(\Delta x^5\right)$ FD-WENO flux reconstruction for $\mathbf{F}(\mathbf{w})$ [Shu, 2003].
- Resulting IVP system: $\dot{\mathbf{w}}(t) = f_1(\mathbf{w}) + f_2(\mathbf{w})$, $\mathbf{w}(t_0) = \mathbf{w}_0$, where $f_1(\mathbf{w})$ contains $-\nabla \cdot \mathbf{F}(\mathbf{w})$ and is evaluated on the CPU, while $f_2(\mathbf{w})$ contains spatially-local reaction network $\mathbf{R}(\mathbf{w})$ and is evaluted on the GPU.
- Compare two forms of temporal evolution:
  (a) Temporally-adaptive, $\mathcal{O}\left(H^3\right)$ IMEX-ARK method from `ARKStep`: f_1 explicit and f_2 implicit,
  (b) Fixed-step, $\mathcal{O}\left(H^3\right)$ MRI-GARK method from `MRIStep` (temporally-adaptive fast step h): f_1 slow/explicit and f_2 fast/DIRK.

Multirate reacting flow – ImEx and multirate results using hybrid CPU+GPU on OLCF Summit.





- Weak scaling runs with 1 MPI rank per GPU.
- Both use robust, GPU-enabled MAGMA batched linear solver.
- Multirate H chosen proportional to CFL condition on f_1.
- Both ImEx & MRI show excellent algorithmic scalability, good MRI runtime scaling up to 86% of Summit.
- Huge reduction in f_1 evaluations allows ImEx $\rightarrow$ MRI speedup of $\sim 70\times$.
- GPU synchronization more severly hinders ImEx scalability, due to increased frequency (fast vs slow stages).

[Return to ARKODE]