Flexible and High Order Time Integration for Multiphysics PDEs

Daniel R. Reynolds

Along with many collaborators, whose contributions will be noted in-line.

Department of Mathematics & Statistics, University of Maryland Baltimore County

North American High Order Methods Conference (NAHOMCon)
Santa Fe, New Mexico
2 June 2026

Outline

Outline

Multiphysics Simulations

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

Climate:

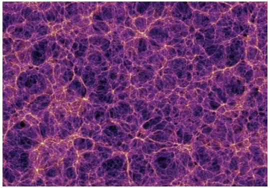
- Atmospheric simulations combine fluid dynamics with local "physics" models for chemistry, condensation, ...
- Atmosphere is coupled at interfaces to myriad other processes (ocean, land/sea ice, . . .), each using distinct models.

Above: https://e3sm.org.

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.

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



"Standard Practice"

Scientific simulations for evolving processes are typically posed as 1st-order systems of differential equations, e.g.

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

where $y \in \mathbb{R}^N$ (or $\mathbb{C}^N$) contains discrete solution values/weights throughout the computational domain.

Numerical methods typically compute approximations $y_n \approx y(t_n)$ at discrete times, $t_0 < t_1 < \cdots < t_{N_t} = t_f$, "marching" from one step to the next using a prescribed update formula, Φ.

Denoting $h_n := t_{n+1} - t_n$ and $y_n \approx y(t_n)$, "textbook" approaches cluster into two categories:

- Explicit methods: $y_{n+1} = \Phi(t_n, t_{n+1}, y_n, y_{n-1}, \ldots)$.
  To ensure numerical stability, these generally require that $h_n < h_{stab}$, where h_{stab} depends on f.
- Implicit methods: $y_{n+1} = \Phi(t_n, t_{n+1}, y_{n+1}, y_n, y_{n-1}, \ldots)$.
  While these may not have stability limitations, they generally require a nonlinear algebraic solver of the form $0 = \mathbf{F}(y) := y - \Phi\left(t_n, t_{n+1}, y, y_n, y_{n-1}, \ldots\right)$.
- If h_{acc} would give an accurate solution, then we call a simulation "stiff" if $h_{stab} \ll h_{acc}$.

Multiphysics Challenges

[Keyes et al., 2013]

Multiphysics simulations can challenge these textbook methods:

- "Multirate" processes evolve on different time scales but prohibit analytical reformulation.
- Stiff components disallow fully explicit methods.
- Nonlinearity and low differentiability challenge fully implicit methods.
- Parallel scalability demands optimal algorithms – while robust/scalable algebraic solvers exist for parts (e.g., FMM for particles, MG 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.

Legacy Multiphysics Method 1: Lie–Trotter

"Operator-splitting" approaches have historically been used for multiphysics applications.

Lie–Trotter computes $y_n \to y_{n+1}$ via

$$\begin{aligned}
\dot{y}^{\{1\}}(t) &= f^{\{1\}}\left(t, y^{\{1\}}\right), & t &\in [t_n, t_{n+1}], & y^{\{1\}}(t_n) &= y_n, \\
\dot{y}^{\{2\}}(t) &= f^{\{2\}}\left(t, y^{\{2\}}\right), & t &\in [t_n, t_{n+1}], & y^{\{2\}}(t_n) &= y^{\{1\}}(t_{n+1}), \\
& & &\vdots & & \\
\dot{y}^{\{M\}}(t) &= f^{\{M\}}\left(t, y^{\{M\}}\right), & t &\in [t_n, t_{n+1}], & y^{\{M\}}(t_n) &= y^{\{M-1\}}(t_{n+1}),
\end{aligned}$$

and sets $y_{n+1} = y^{\{M\}}(t_{n+1})$. Each IVP is tackled independently using different "standard" approaches (e.g., implicit Euler, ERK-4, subcycling, ...).

Legacy Multiphysics Method 2: Strang–Marchuk

$$\dot{y}^{\{1\}}(t) = f^{\{1\}}\left(t, y^{\{1\}}\right), \qquad t \in \left[t_n, t_n + \tfrac{h_n}{2}\right], \qquad y^{\{1\}}(t_n) = y_n,$$

$$\vdots$$

$$\dot{y}^{\{M-1\}}(t) = f^{\{M-1\}}\left(t, y^{\{M-1\}}\right), \qquad t \in \left[t_n, t_n + \tfrac{h_n}{2}\right], \qquad y^{\{M-1\}}(t_n) = y^{\{M-2\}}\left(t_n + \tfrac{h_n}{2}\right),$$

$$\dot{y}^{\{M\}}(t) = f^{\{M\}}\left(t, y^{\{M\}}\right), \qquad t \in [t_n, t_{n+1}], \qquad y^{\{M\}}(t_n) = y^{\{M-1\}}\left(t_n + \tfrac{h_n}{2}\right),$$

$$\dot{y}^{\{M-1\}}(t) = f^{\{M-1\}}\left(t, y^{\{M-1\}}\right), \qquad t \in \left[t_n + \tfrac{h_n}{2}, t_{n+1}\right], \qquad y^{\{M-1\}}\left(t_n + \tfrac{h_n}{2}\right) = y^{\{M\}}(t_{n+1}),$$

$$\vdots$$

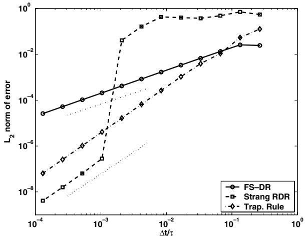
$$\dot{y}^{\{1\}}(t) = f^{\{1\}}\left(t, y^{\{1\}}\right), \qquad t \in \left[t_n + \tfrac{h_n}{2}, t_{n+1}\right], \qquad y^{\{1\}}\left(t_n + \tfrac{h_n}{2}\right) = y^{\{2\}}(t_{n+1}),$$

$$y_{n+1} = y^{\{1\}}(t_{n+1}).$$

Shorcomings of loose "initial condition" coupling

Generally poor accuracy:

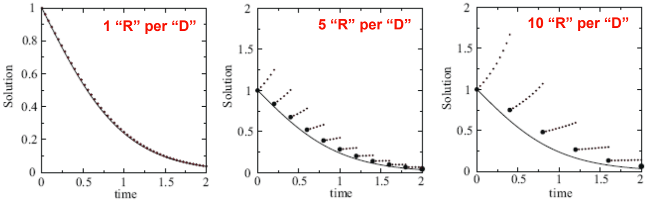
- *No matter the accuracy of each sub-solver,* Lie–Trotter is at best $\mathcal{O}(h_n)$ and Strang–Marchuk is $\mathcal{O}(h_n^2)$.
- Higher-order "fractional-step" methods are possible, but requre backwards-in-time integration [Goldman & Kaper, 1996].
- Extrapolation or deferred correction can also improve accuracy but at significant cost.



Convergence of splitting approaches (brusselator) [Ropp & Shadid, 2005].

Poor stability:

- Even "stable" step sizes for each part can result in unstable modes.



Subcycling stability (reaction-diffusion) [Estep et al., 2008].

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; . . .]

At each stage ARK methods must solve a root-finding problem:

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

- If $f^I(t, y) = J(t)y$ (i.e., f^I is *linear* in y) then this is a large-scale linear system for each z_i:

$$\left(I - h_n a^I_{i,i} J(t_{n,i})\right) z_i = rhs_i.$$

- Else this requires an iterative solver (e.g., Newton, accelerated fixed-point, or problem-specific), that itself may require solution of multiple linear systems.

- All operators in $f^E(t, y)$ are treated explicitly (do not affect algebraic solver convergence).

ImEx-ARK methods are 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, stability, . . .

Multirate Infinitesimal Step (MIS) methods

[Knoth & Wolke, 1998; Schlegel et al., 2009]

MIS methods provide a flexible approach for higher-order "subcycling" in multirate applications:

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

- $f^S(t,y)$ contains the "slow" dynamics, evolved with large step H_n.
- $f^F(t,y)$ contains the "fast" dynamics, evolved with small steps $h_{n,m} \ll H_n$.
- The slow component is similar to an ERK method, while the fast component is advanced between slow stages by solving a modified IVP with a subcycled "inner" method.
- Extremely efficient – achieves second order using *a single traversal* of $[t_n, t_{n+1}]$ for all suitable ERK tables (order 2, and **sorted** $\mathbf{0 = c_1 < \cdots < c_s = 1}$); achieves $\mathcal{O}\left(h_n^3\right)$ if the table satisfies

$$\sum_{i=2}^{s} (c_i - c_{i-1})(\mathbf{e}_i + \mathbf{e}_{i-1})^T Ac + (1 - c_s)\left(\frac{1}{2} + \mathbf{e}_s^T Ac\right) = \frac{1}{3}.$$

MIS Algorithm Outline

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: $t_{n,i} = t_n + c_i H_n$, $\Delta c_i = c_i - c_{i-1}$, and $r_i = \frac{1}{\Delta c_i} \sum_{j=1}^{i-1} \left(a_{i,j} - a_{i-1,j}\right) f_j^S$.
   b) Evolve: $\dot{v}_i(\tau) = f^F(\tau, v_i) + r_i$, for $\tau \in [t_{n,i-1}, t_{n,i}]$, $v(t_{n,i-1}) = z_{i-1}$.
   c) Let: $z_i = v(t_{n,i})$, and $f_i^S = f^S(t_{n,i}, z_i)$.
3. Let: $y_{n+1} = z_s$.

Notes:

- Step 2b may use any sufficiently accurate solver (including another MIS method), hence the "flexibility" mentioned earlier.
- Improved accuracy results from including r_i in solving the fast IVP, naturally serving to propagate information from the slow to fast time scale.

Outline

1. Background
2. Unifying Theory
3. Method development
4. Open-source scientific software
5. Conclusions, Etc.

Generalized Structure Additive Runge–Kutta Theory [Sandu & Günther, 2015]

In 2015, Adrian Sandu and Michael Günther introduced a theory to analyze all of the aforementioned multiphysics methods. Assigning a separate set of internal stages, $\left\{z_i^{\{q\}}\right\}_{i=1}^{s^{\{q\}}}$, for each RHS partition $\left\{f^{\{q\}}\right\}_{q=1}^{M}$, they write a single step $y_n \to y_{n+1}$ as

$$z_i^{\{q\}} = y_n + h_n \sum_{m=1}^{M} \sum_{j=1}^{s^{\{m\}}} a_{i,j}^{\{q,m\}} f^{\{m\}}\left(z_j^{\{m\}}\right), \quad i = 1, \ldots, s^{\{q\}}, \quad q = 1, \ldots, M,$$

$$y_{n+1} = y_n + h_n \sum_{q=1}^{M} \sum_{j=1}^{s^{\{q\}}} b_j^{\{q\}} f^{\{q\}}\left(z_j^{\{q\}}\right).$$

The coefficients form a *generalized Butcher tableau*

$$\begin{array}{cccc} A^{\{1,1\}} & A^{\{1,2\}} & \cdots & A^{\{1,M\}} \\ A^{\{2,1\}} & A^{\{2,2\}} & \cdots & A^{\{2,M\}} \\ \vdots & \vdots & & \vdots \\ A^{\{M,1\}} & A^{\{M,2\}} & \cdots & A^{\{M,M\}} \\ \hline b^{\{1\}} & b^{\{2\}} & \cdots & b^{\{M\}} \end{array}$$

GARK Order Conditions

[Sandu & Günther, 2015]

Assuming internal consistency between tables, $c^{\{q\}} := A^{\{q,1\}}\mathbf{1}^{\{1\}} = \cdots = A^{\{q,M\}}\mathbf{1}^{\{M\}}$ for $q = 1, \ldots, M$, and leveraging NB-tree theory [Araujo et al., 1997], they derived conditions up to order 4:

$$
\begin{aligned}
b^{\{\sigma\}}\mathbf{1}^{\{\sigma\}} &= 1 && \forall\sigma && \text{(order 1)},\\
b^{\{\sigma\}}c^{\{\sigma\}} &= \tfrac{1}{2} && \forall\sigma && \text{(order 2)},\\
b^{\{\sigma\}}C^{\{\sigma\}}c^{\{\sigma\}} &= \tfrac{1}{3} && \forall\sigma && \text{(order 3)},\\
b^{\{\sigma\}}A^{\{\sigma,\nu\}}c^{\{\nu\}} &= \tfrac{1}{6} && \forall\sigma,\nu && \text{(order 3)},\\
b^{\{\sigma\}}C^{\{\sigma\}}C^{\{\sigma\}}c^{\{\sigma\}} &= \tfrac{1}{4} && \forall\sigma && \text{(order 4)},\\
b^{\{\sigma\}}C^{\{\sigma\}}A^{\{\sigma,\nu\}}c^{\{\nu\}} &= \tfrac{1}{8} && \forall\sigma,\nu && \text{(order 4)},\\
b^{\{\sigma\}}A^{\{\sigma,\nu\}}C^{\{\nu\}}c^{\{\nu\}} &= \tfrac{1}{12} && \forall\sigma,\nu && \text{(order 4)},\\
b^{\{\sigma\}}A^{\{\sigma,\nu\}}A^{\{\nu,\mu\}}c^{\{\mu\}} &= \tfrac{1}{24} && \forall\sigma,\nu,\mu && \text{(order 4)},
\end{aligned}
$$

where $\mathbf{1}^{\{\nu\}} \in \mathbb{R}^{s^{\{\nu\}}}$ is a vector of all ones, and $C^{\{\sigma\}} = \operatorname{diag}\left(c^{\{\sigma\}}\right)$.

GARK Linear Stability

[Sandu & Günther, 2015]

Linear stability is more challenging – assuming simultaneous diagonalizability, they consider the Dahlquist-like test problem

$$\dot{y}(t) = \sum_{\ell=1}^{M} \lambda^{\{\ell\}} y, \quad y(0) = 1, \quad \Re\left(\lambda^{\{\ell\}}\right) < 0.$$

Denoting $z^{\{\ell\}} = h\lambda^{\{\ell\}}$, the GARK stability function may be written as

$$\mathcal{R}\left(z^{\{1\}}, \ldots, z^{\{M\}}\right) = 1 + \mathbf{b}Z\left(I_{s\times s} - AZ\right)^{-1}\mathbf{1}_{s\times 1},$$

where $s = \sum_{q=1}^{M} s^{\{q\}}, \quad \mathbf{b} = \begin{bmatrix} b^{\{1\}} & \cdots & b^{\{M\}} \end{bmatrix},$

$$\mathbf{A} = \begin{bmatrix} A^{\{1,1\}} & \cdots & A^{\{1,M\}} \\ & \ddots & \\ A^{\{M,1\}} & \cdots & A^{\{M,M\}} \end{bmatrix}, \quad \text{and} \quad Z = \begin{bmatrix} z^{\{1\}} \otimes I_{s^{\{1\}}\times s^{\{1\}}} & & \\ & \ddots & \\ & & z^{\{M\}} \otimes I_{s^{\{M\}}\times s^{\{M\}}} \end{bmatrix}.$$

However, plotting the stability region $\left\{\mathbf{z} \in \mathbb{C}^M \;:\; |\mathcal{R}(\mathbf{z})| < 1\right\}$ is considerably less straightforward than with traditional non-partitioned methods.

Outline

Multirate Infinitesimal GARK (MRI-GARK) methods [Sandu, 2019]

Soon after inventing GARK theory, Sandu extended MIS methods to implicit and higher-order accuracy. The only change from the MIS algorithm is in his definition of the forcing function,

$$r_i(\tau) = \frac{1}{\Delta c_i} \sum_{j=1}^{i} \gamma_{i,j}(\tau) f_j^S, \quad \text{where} \quad \gamma_{i,j}(\tau) = \sum_{\ell=1}^{n_\Gamma} \Gamma_{i,j}^{\{\ell\}} \tau^{\ell-1}.$$

Notes:

- Fourth order is attainable, again with *only a single traversal* of $[t_n, t_{n+1}]$.
- The coefficients $\left\{\Gamma^{\{\ell\}}\right\}_{\ell=1}^{n_\Gamma}$ satisfy order conditions derived from GARK theory *(next slide)*.
- Implicitness depends on $\gamma_{i,i}(\tau) \neq 0$. It is only allowed when $\Delta c_i = 0$ (i.e., "solve-decoupled"), in which case the "fast IVP" becomes a DIRK-like implicit solve

$$0 = F_i(z) := \left[z - h_n \left(\sum_{\ell=1}^{n_\Gamma} \frac{\Gamma_{i,i}^{\{\ell\}}}{\ell}\right) f^S\left(t_{n,i}, z\right)\right] - \left[z_{i-1} + h_n \sum_{j=1}^{i-1} \left(\sum_{\ell=1}^{n_\Gamma} \frac{\Gamma_{i,j}^{\{\ell\}}}{\ell}\right) f_j^S\right].$$

- **As with MIS, derivation requires sorted abscissae: $\mathbf{0 = c_1 \leq c_2 \leq \cdots \leq c_s = 1}$.**

Infinitesimal "order conditions"

MIS/MRI methods of order $\mathcal{O}(h_n^p)$ couple time scales via the forcing function $r_i(\tau) = \frac{1}{\Delta c_i}\sum_j \gamma_{ij}(\tau) f_j^S$,

that is used in solving the fast IVP

$$\dot{v}_i(\tau) = f^F(\tau, v_i) + r_i(\tau), \quad \tau \in [\tau_{i,0}, \tau_{i,f}], \quad v_i(\tau_0) = v_{i,0}$$

$$\Leftrightarrow$$

$$v_i(\tau_{i,f}) = \Phi(\tau_{i,0}, \tau_{i,f}, v_{i,0}).$$

MIS/MRI methods do not care how Φ operates, only that it also has global accuracy $\mathcal{O}(h_n^p)$. Thus to $\mathcal{O}\left(h_n^{p+1}\right)$ it is equivalent to an $s^{\{f\}}$-stage RK method $\{A^{\{f,f\}}, b^{\{f\}}, c^{\{f\}}\}$ that satisfies

$$\begin{aligned} b^{\{f\}}\mathbf{1}^{\{f\}} &= 1 && \text{(order 1)}, \\ b^{\{f\}}c^{\{f\}} &= \tfrac{1}{2} && \text{(order 2)}, \\ b^{\{f\}}C^{\{f\}}c^{\{f\}} &= \tfrac{1}{3} && \text{(order 3)}, \\ b^{\{f\}}A^{\{f\}}c^{\{f\}} &= \tfrac{1}{6} && \text{(order 3)}, \\ &\vdots \end{aligned}$$

These are then combined with the GARK theory to derive conditions on $\gamma_{ij}(\tau)$, and thus $\Gamma_{ij}^{\{\ell\}}$.

Implicit-Explicit Multirate Infinitesimal GARK Methods

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

My student Rujeko Chinomona extended Sandu's MRI-GARK methods to support implicit-explicit treatment of the slow time scale:

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

These define an ImEx forcing function

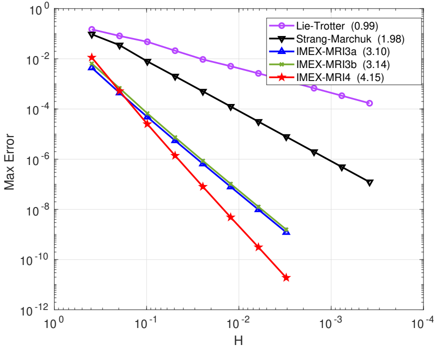
$$r_i(\tau) = \frac{1}{\Delta c_i} \sum_{j=1}^{i} \gamma_{i,j}(\tau) f_j^I + \frac{1}{\Delta c_i} \sum_{j=1}^{i-1} \omega_{i,j}(\tau) f_j^E,$$

where $\gamma_{i,j}(\tau)$ and $\omega_{i,j}(\tau)$ are defined similarly as for MRI-GARK.

- Order conditions up to $\mathcal{O}(h_n^4)$ again leverage the GARK framework.
- Derived $\mathcal{O}(h_n^3)$ methods with optimized linear stability, and a candidate $\mathcal{O}(h_n^4)$ method.
- *However, due to sorted abscissae requirement, we were unable to derive embedded $\mathcal{O}(h_n^3)$ ImEx-MRI-GARK methods, and struggled to optimize linear stability at $\mathcal{O}(h_n^4)$.*

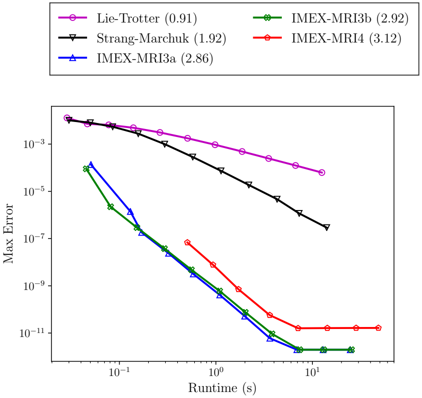
ImEx-MRI-GARK Convergence/Efficiency

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



Nonlinear Kværnø-Prothero-Robinson ODE test problem convergence.

All methods exhibit expected convergence rate; ImEx-MRI are far more accurate than OS.



1D stiff Brusselator PDE test runtime efficiency.

Similar story for stiff PDE test, but the largest two step sizes were unstable for ImEx-MRI4.

Implicit-Explicit Multirate Infinitesimal Stage-Restart Methods

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

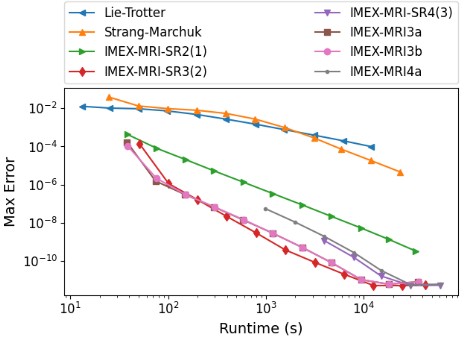

To circumvent the sorted abscissae requirement, my student Alex Fish, collaborator Steven Roberts, and I developed ImEx-MRI-SR methods by assuming a simpler structure for the step $y_n \to y_{n+1}$:

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

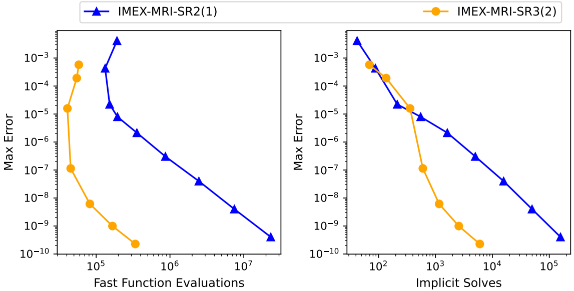
- Structure is independent of $\Delta c_i = 0$; no "padding" is required to derive ImEx-MRI-SR from ImEx-ARK.
- The embedding has an identical structure as the last stage, z_s.
- Derived embedded methods from $\mathcal{O}(h_n^2) - \mathcal{O}(h_n^4)$; linear stability at $\mathcal{O}(h_n^4)$ again challenging to optimize.

ImEx-MRI-SR Convergence/Efficiency – 1D Stiff Brusselator PDE [Fish, R., & Roberts, *JCAM*, 2024]

1D Stiff Brusselator PDE results: fixed-step runtime efficiency (left), and adaptive-step efficiency measured via f^F evaluations or f^I solves.



ImEx-MRI-SR have similar accuracy to ImEx-MRI-GARK (far better than OS); reduced linear stability at $\mathcal{O}(h_n^4)$ again visible..

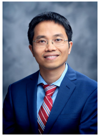


Both methods can adaptively track multirate dynamics (more on adaptivity in a few slides);
$\mathcal{O}(h_n^3)$ more efficient at tighter accuracy;
$\mathcal{O}(h_n^2)$ competive wrt implicit solves at loose accuracy.

Multirate Exponential Runge–Kutta (MERK) and Rosenbrock (MERB) Methods

[Luan, Chinomona & R., *SISC*, 2020; Luan, Chinomona & R., *SISC*, 2022]

To circumvent the explosion in GARK order conditions, with collaborator Vu Thai Luan, Rujeko and I leveraged exponential method theory to replace the action of the φ_j functions with "infinitesimal" IVP solves *(next slide)*.



We consider multirate IVP of the form

$$y'(t) = F(t,y) = \mathcal{J}_n y + f^S(t,y), \quad t \in [t_n, t_n + h_n], \quad y(t_n) = y_n \in \mathbb{R}^n.$$

- "Fast" scale corresponds to $\mathcal{J}_n y$: for MERK $\mathcal{J}_n$ may be arbitrary, for MERB $\mathcal{J}_n = \frac{\partial F}{\partial y}(t_n, y_n)$.
- "Slow" scale corresponds to $f^S(t,y)$; must be treated explicitly in MERK and MERB methods.
- Their implementations are nearly identical to ImEx-MRI-SR, albeit without any implicit solves.
- Provide embedded MERK up to $\mathcal{O}(h_n^5)$, and non-embedded MERB up to $\mathcal{O}(h_n^6)$.

Multirate Exponential "Idea"

[Luan, Chinomona & R., *SISC*, 2020]

A single exponential Euler time step $y_n \to y_{n+1}$ has the form

$$y_{n+1} = \varphi_0(h_n \mathcal{J}_n) y_n + h_n \varphi_1(h_n \mathcal{J}_n) f^S(t_n, y_n),$$

where $\varphi_0(z) = e^z$, and $\varphi_1(z) = \frac{e^z - 1}{z}$.

Instead of approximating these matrix functions, we note that this method exactly solves a *modified IVP*

$$\dot{v}(\tau) = \mathcal{J}_n v + r, \quad \tau \in [0, h_n], \quad v(0) = y_n, \quad r := f^S(t_n, y_n).$$

Thus, to $\mathcal{O}(h_n^2)$, we can replace the action of $\varphi_j(h_n \mathcal{J}_n)$ with an "infinitesimal" IVP solve.

Higher-order MERK and MERB methods are derived similarly, so long as the internal stages and time step solution of the exponential method may be written as linear combinations of φ_k functions,

$$\sum_{k=1}^{\ell_i} \alpha_{i,j}^{(k)} \varphi_k(c_i h_n \mathcal{J}_n),$$

where c_i is the abscissa of the i-th stage, and $\alpha_{i,j}^{(k)}$ are method-specific coefficients.

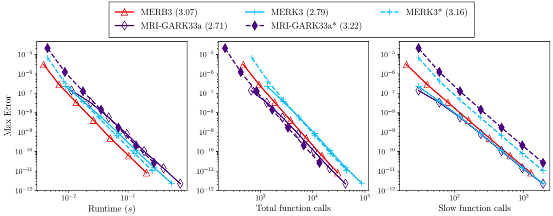
MERK/MERB Efficiency – 1D Reaction-Diffusion PDE

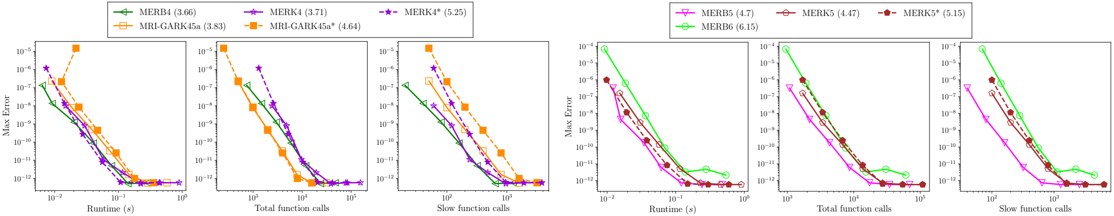
[Luan, Chinomona & R., *SISC*, 2022]

Problem: $u_t = \epsilon u_{xx} + \gamma u^2(1-u),\ x \in (0,5),\ t \in [0,5]$,
with $\gamma = 0.1$, $\epsilon = 10^{-2}$, $\lambda = \sqrt{5}$, $u(x,0) = (1+\exp(\lambda(x-1)))^{-1}$, and $u_x(0,t) = u_x(5,t) = 0$.

Efficiency plots (runtime, RHS calls, f^S calls):

- Top-right: $\mathcal{O}(H^3)$ methods
- Bottom-left: $\mathcal{O}(H^4)$ methods
- Bottom-right: $\mathcal{O}(H^5)$ and $\mathcal{O}(H^6)$ methods
- Methods* use a "natural" splitting; others use dynamic linearization.



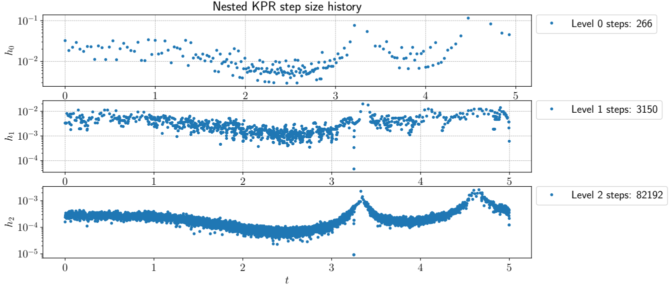


MERK/MERB more efficient than MRI-GARK in both runtime and f^S cost; MRI-GARK more efficient in f^F.

MRI time step adaptivity

[Fish & R., *SISC*, 2023; R., Amihere, Mitchell, & Luan, *JCAM*, 2026]

- For single-rate problems, it is well-known that method robustness, accuracy, and efficiency hinge on appropriate selection of h_n.
- While decades of work have optimized single-rate "controllers", we were the first to establish rigorous methods for time step control of MRI methods.
- In [Fish & R., 2023] we simultaneously select both H_n and h_n for a given time step $y_n \to y_{n+1}$, based on separate estimates of the local truncation error in both the MRI method and the inner integrator.
- In [R. et al., 2026] we instead select both H_n and an inner solver tolerance scaling factor $r_{tol,n}$, using similar local error estimates. These allow the inner solver to adapt *within* the MRI step $y_n \to y_{n+1}$, and enable temporal adaptivity with nested MRI methods.



Outline

SUNDIALS [Hindmarsh et al., *TOMS*, 2005; Balos et al., *ParComp*, 2021; Gardner et al., *TOMS*, 2022; Roberts et al., *TOMS*, 2026]

A primary goal of my research is to impact other fields – I feel can best achieve this through dissemination of modern methods through high-quality open-source software.

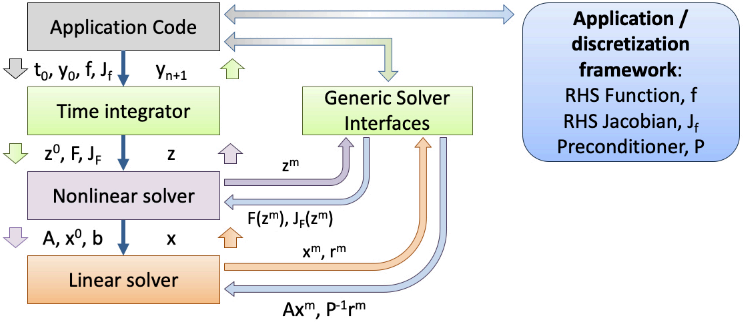
My collaborators and I develop and maintain *SUNDIALS* – the SUite of Nonlinear and DIfferential/ALgebraic Solvers

- Adaptive time integrators for ODEs and DAEs and efficient nonlinear solvers, used throughout research and industry.
- Written in C/C++, natively supports parallel computing:
    - GPUs via CUDA, HIP, SYCL, RAJA and Kokkos
    - Clusters/supercomputers via MPI
    - Threading via OpenMP and Pthreads
- BSD license; available from GitHub or Spack.
- Over 100,000 downloads/year since 2016.
- Runs anywhere from laptops to supercomputers.
- Embedded in MATLAB, Mathematica, Python (scikit), DifferentialEquations.jl, OpenModelica, . . .

- CVODE(S): linear multistep methods for IVPs, fwd/adj sensitivities
- IDA(S): linear multistep methods for DAEs, fwd/adj sensitivities
- ARKODE: multi-stage methods for IVPs
- KINSOL: nonlinear solver

SUNDIALS' Modular Design & Control Inversion [Gardner et al., 2022]

Control passes between integrator, solvers, and application code as the integration progresses:



Time integrators are agnostic as to the vector data layout and algebraic solvers used, leveraging application-specific implementations where possible, and providing native modules if desired.

SUNDIALS' ARKODE solver

[R. et al., *TOMS*, 2023; Roberts et al., *TOMS*, 2026]

ARKODE has been released as part of SUNDIALS since 2014, providing adaptive ImEx-ARK methods. Since then we have enhanced ARKODE to include a variety of "steppers":

- `ARKStep`: adaptive ARK, ERK, DIRK methods [all]; includes an interface to XBraid for parallel-in-time [D. Gardner]
- `ERKStep`: adaptive explicit RK methods [all]; includes fixed-step discrete FSA/ASA [C. Balos]
- `MRIStep`: adaptive infinitesimal multirate methods (MIS, MRI-GARK, ImEx-MRI-GARK, ImEx-MRI-SR, MERK) [R. Chinomona, D. Gardner]
- `SPRKStep`: fixed-step symplectic RK methods for partitioned Hamiltonian systems [C. Balos]
- `SplittingStep` and `ForcingStep`: fixed-step operator splitting methods (including very high order) [S. Roberts]
- `LSRKStep`: adaptive explicit RK methods with low-storage structure (strong-stability preserving, RKC and RKL "super time stepping") [M. Aggul]

Outline

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 simple operator-spliting remains standard, new & flexible methods are catching up, supporting time adaptivity, high order accuracy (up to $\mathcal{O}(h_n^6)$), 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, ...

Current projects in my group (all are in collaboration with external partners)

- Construct embedded ImEx and split-explicit super-time-stepping methods [M. Aggul, S. Amihere]
- Construct embedded ImEx-ARK methods with SSP properties [S. Amihere]
- Explore performance of novel ImEx and MRI algorithms for simulating instabilities in magnetic fusion plasmas [M. Aggul, S. Amihere, Y. Hu]
- Examine approaches to leverage data-driven machine learning techniques to accelerate first-principles predictive simulations [D. Mitchell]
- Examine approaches for temporal error estimation and control for application-specific quantities of interest

Future directions (for my group or others)

- Devise constraint-preserving MRI methods – while these naturally conserve linear invariants, are there extensions for energy conservation, asymptotic preserving, SSP?
- Can we extend these "flexible integration" ideas approaches (ImEx & MRI) to large-scale systems of DAEs?
- Can we devise robust and automated approaches for splitting an IVP into stiff/nonstiff or fast/slow partitions, $f(t,y) = \sum_k f^{\{k\}}(t,y)$?
- Can we come up with a more rigorous stability theory for additively-partitioned ODE systems (not just $\dot{y} = \sum_k \lambda_k y$)?
- ?

Thank you for your time and attention!

- For more information on any of our new methods research, please see my webpage: https://drreynolds.github.io.
- For more information on SUNDIALS, please see our
  - Github: https://github.com/llnl/sundials
  - Documentation: https://sundials.readthedocs.io
- For anything else, talk to me at this meeting or send me an email (dreynolds@umbc.edu)

Funding acknowledgments

This work was funded 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

- Frameworks, Algorithms and Software Technologies for Mathematics (FASTMath) Institute, under DOE award DE-SC0021354,
- Computational Evaluation and Design of Actuators for Core-Edge Integration (CEDA) project, under DOE award DE-SC0021354, and
- FRONTIERS in Leadership Gyrokinetic Simulation project, under DOE award DE-SC0024425.
- Developing Multiscale Simulation of Boundary Plasma Dynamics (ABOUND) project, under DOE award DE-SC0021354.

This work was also funded in part by the U.S. Department of Energy, Office of Science, Office of Fusion Energy Sciences, under the FIRE Collaborative: Advanced Profile Prediction for Fusion Pilot Plant Design (APP–FPP) project, under DOE award DE-SC0025853.