Flexible Time Integration Methods for Multiphysics PDE Systems

Daniel R. Reynolds

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

Department of Mathematics, Southern Methodist University,
Department of Mathematics & Statistics, University of Maryland Baltimore County

VIASM Workshop on Advanced Numerical Methods for
Deterministic and Stochastic Differential Equations
13 June 2025

Outline

Outline

Multiphysics Simulations

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

Climate:

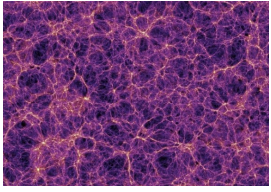
- Atmospheric simulations combine fluid dynamics with local "physics" models for chemistry, condensation, ...
- Atmosphere is coupled at interfaces to myriad other processes (ocean, land/sea ice, ...), each using distinct models.

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.



Multiphysics Challenges

[Keyes et al., 2013]

Since multiphysics simulations combine multiple physical processes, they 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, multigrid for diffusion), none are optimal for the whole.

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

$$y'(t) = f^{\{1\}}(t,y) + \cdots + f^{\{m\}}(t,y), \quad t \in [t_0, t_f], \quad y(t_0) = y_0.$$

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; ...]

At each stage we 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 (MRI) methods

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

MRI methods provide a rigorous and highly accurate approach to "subcycling" 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, naturally evolved with large steps H.
- $f^F(t,y)$ contains the "fast" dynamics, that evolves with small steps $h \ll H$.
- It is generally assumed that $f^S(t,y)$ is *considerably more costly to evaluate* than $f^F(t,y)$, and thus a method which evaluates $f^S(t,y)$ infrequently is desirable.
- Extremely efficient – fourth order is attainable with *only a single traversal* of $[t_n, t_{n+1}]$; fifth order is attainable with only slightly higher cost.

MRI Algorithm Outline

Denoting $y_n \approx y(t_n)$, a single explicit MRI step $t_n \to t_n + H$ proceeds as:

1. Let: $z_1 = y_n$
2. For $i = 2, ..., s$:
   a. Solve: $v_i'(t) = f^F(t, v_i(t)) + r_i(t)$, for $t \in [t_{n,i-1}, t_{n,i}]$ with $v_i(t_{n,i-1}) = z_{i-1}$.
   b. Let: $z_i = v_i(t_{n,i})$.
3. Solve: $\tilde{v}_s'(t) = f^F(t, \tilde{v}_s(t)) + \tilde{r}_s(t)$, for $t \in [t_{n,s-1}, t_{n+1}]$ with $\tilde{v}_s(t_{n,s-1}) = z_{s-1}$.
4. Let: $y_{n+1} = z_s$ (solution) and $\tilde{y}_{n+1} = \tilde{v}_s(t_{n+1})$ (embedding).

- MRI methods are uniquely defined by: stage times $t_{n,i} = t_n + c_i H$ and forcing functions $r_i(t)$, $\tilde{r}_s(t)$, that are linear combinations of $\{f^S(t_{n,j}, z_j)\}$ and propagate information from slow to fast scales.
- Implicit versions are possible, that replace step 2a with a DIRK-like solve during implicit stages.
- The original MIS methods [Schlegel et al., 2009] used constant $r_i(t)$ and achieved up to $\mathcal{O}(H^3)$.
- Sandu's MRI-GARK [2019] extended these to $\mathcal{O}(H^4)$ via polynomial $r_i(t)$, and introduced embeddings.

Outline

Implicit-Explicit Multirate Infinitesimal GARK Methods

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

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

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

- Order conditions up to $\mathcal{O}(H^4)$ leverage the GARK framework [Sandu & Günther, *SINUM*, 2015].
- *However, these inherited the MRI-GARK requirement for sorted abscissae ($0 = c_1 \le c_2 \le \cdots \le c_s = 1$), thus we were unable to derive embedded IMEX-MRI-GARK methods, and struggled to derive 4th order methods with good stability.*
- Information on their construction and computational results are available in the (Appendix).

Implicit-Explicit Multirate Infinitesimal Stage-Restart Methods

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

To circumvent the sorted abscissae requirement, 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: $r_i(t) = \sum_{j=1}^{i-1} \omega_{i,j}\left(\frac{t-t_n}{c_i H}\right)\left(f_j^E + f_j^I\right)$, with $\omega_{i,j}(\theta) = \frac{1}{c_i}\sum_{k=0}^{n_\Omega - 1} \omega_{i,j}^{\{k\}}\theta^k$.
   b) Evolve: $\dot{v}(t) = f^F(t,v) + r_i(t)$, for $t \in [t_n, t_{n,i}],\ v(t_n) = y_n$.
   c) Solve: $z_i = v(t_{n,i}) + H\sum_{j=1}^{i}\gamma_{i,j} f^I(t_{n,j}, z_j)$.
3. Let: $y_{n+1} = z_s$.

- For brevity above, we denote $f_j^E := f^E(t_{n,j}, z_j)$ and $f_j^I := f^I(t_{n,j}, z_j)$.
- The embedding has an identical structure as the last stage, z_s.
- There is no "hidden" dependence on $\Delta c_i = 0$ for the stage structure, thus no "padding" is required to derive IMEX-MRI-SR methods from IMEX-ARK.
- Information on their construction and computational results are available in the (Appendix).

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

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

To circumvent the explosion in GARK order conditions, we leveraged exponential method theory, replacing the action of the φ_j fcns. with "infinitesimal" IVP solves.

We apply multirate IVP splittings of the form

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

- "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 $\mathcal{V}_n t + \mathcal{N}_n$ with $\mathcal{V}_n = \frac{\partial F}{\partial t}(t_n, y_n)$ and $\mathcal{N}_n = F - \mathcal{J}_n y - \mathcal{V}_n t$.
- Due to their exponential structure, MERK and MERB only support explicit treatment of f^S.
- Their implementations are nearly identical to IMEX-MRI-SR, albeit without any implicit solves.
- We have embedded MERK up to $\mathcal{O}(H^5)$, and non-embedded MERB up to $\mathcal{O}(H^6)$.
- Computational results are available in the (Appendix).

Outline

Single rate control

[Gustafsson, 1991; Söderlind, 2006]

- Traditional adaptivity controls *local error*, $\ell_n := y(t_n + H_n) - y_{n+1}$, assuming y_n is exact, adapting H_n to ensure $\varepsilon_n := \|\ell_n\| \leq 1$ where the norm incorporates the user tolerances, among other objectives.
- A "controller" $\mathcal{C}$ typically depends on a few (H_k, ε_k) pairs, i.e.,

$$\tilde{H} = \mathcal{C}(H_n, \varepsilon_n, H_{n-1}, \varepsilon_{n-1}, \ldots, p),$$

  where p is the global method order, i.e., $\varepsilon_n \leq c(t)\, H_n^{p+1}$, for some $c(t)$ independent of H_n.
- The simple *I controller* assumes equality above, and a piecewise constant $c = \frac{\varepsilon_n}{H_n^{p+1}}$, to predict $\tilde{H}$:

$$1 = c\tilde{H}^{p+1} = \varepsilon_n \frac{\tilde{H}^{p+1}}{H_n^{p+1}} \quad \Leftrightarrow \quad \tilde{H} = \frac{H_n}{\varepsilon_n^{1/(p+1)}}.$$

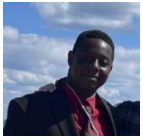
- More advanced options exist, that typically use additional (H_k, ε_k) values to build higher-degree piecewise polynomial approximations of the principal error function.

For multirate control, we thus require both a strategy to estimate local temporal errors ε_n^S and ε_n^F (Appendix), and algorithms for selecting step sizes H and h.

MRI time step control – Decoupled (Dec) controllers

The simplest approach to MRI adaptivity uses decoupled single-rate controllers:

$$\tilde{H} = \mathcal{C}^S(H_n, \varepsilon_n^S, H_{n-1}, \varepsilon_{n-1}^S, \ldots, P),$$
$$\tilde{h} = \mathcal{C}^F(h_{n,m}, \varepsilon_{n,m}^F, h_{n,m-1}, \varepsilon_{n,m-1}^F, \ldots, p),$$



where (H_k, ε_k^S) are the stepsize and local error estimates for time step k at the slow time scale, and $(h_{k,l}, \varepsilon_{k,l}^F)$ are the stepsize and local error estimates for the fast substep l within the slow step k.

- $\mathcal{C}^S$ and $\mathcal{C}^F$ are independent of one another, so selection of $\tilde{H}$ and $\tilde{h}$ occurs independently.
- We expect this to work well for problems with weakly coupled time scales.
- Due to its decoupled nature, this trivially extends to an arbitrary number of time scales, allowing adaptivity for so-called "telescopic" MRI methods.

MRI time step control – Step-tolerance (H-Tol) controllers

For problems with more strongly coupled scales, we may want tighter accuracy from the inner solver. When called over fast intervals $[\tau_{0,i}, \tau_{F,i}]$, $\tau_{F,i} - \tau_{0,i} \le H$, we assume its accumulated error satisfies

$$\varepsilon_i^F = \chi(t) H_n \left(\text{reltol}_n^F\right),$$

where reltol_n^F is the requested relative tolerance for the inner solver, and $\chi(t)$ is independent of reltol_n^F.

This matches the single-rate controller assumption $\varepsilon_n = c(t)h^{p+1}$, where $\chi(t_n)H_n$ is $c(t)$, reltol_n^F is h, and $p = 0$. Thus a single-rate controller could adjust reltol_n^F between slow step attempts.

Thus to construct an "H-Tol" controller, we leverage three separate single-rate controllers:

- $\mathcal{C}^{S,H}$ – single-rate controller to adapt H_n within the slow integrator.
- $\mathcal{C}^{S,Tol}$ – single-rate controller to adapt reltol_n^F above.
- $\mathcal{C}^F$ – single rate controller to adapt $h_{n,m}$ within the fast integrator to achieve reltol_n^F.

This class of controllers also support telescopic multirate methods.

MRI time step control – Coupled (H-h) controllers

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

These extend the single-rate derivations of [Gustafsson, 1994] by approximating both slow and fast principal error functions using piecewise polynomials. Four MRI controllers were proposed that simultaneously adapt H_n and $M_n = H_n/h_n$:

- *constant-constant*: $H_{n+1} = H_n \left(\varepsilon^S_{n+1}\right)^{\alpha}, \quad M_{n+1} = M_n \left(\varepsilon^S_{n+1}\right)^{\beta_1} \left(\varepsilon^F_{n+1}\right)^{\beta_2},$
- *linear-linear*:
$$H_{n+1} = H_n \left(\frac{H_n}{H_{n-1}}\right) \left(\varepsilon^S_{n+1}\right)^{\alpha_1} \left(\varepsilon^S_{n}\right)^{\alpha_2},$$
$$M_{n+1} = M_n \left(\frac{M_n}{M_{n-1}}\right) \left(\varepsilon^S_{n+1}\right)^{\beta_{11}} \left(\varepsilon^S_{n}\right)^{\beta_{12}} \left(\varepsilon^F_{n+1}\right)^{\beta_{21}} \left(\varepsilon^F_{n}\right)^{\beta_{22}}.$$
- *PIMR* (a multirate extension of the PI single-rate controller):
$$H_{n+1} = H_n \left(\varepsilon^S_{n+1}\right)^{\alpha_1} \left(\varepsilon^S_{n}\right)^{\alpha_2}, \quad M_{n+1} = M_n \left(\varepsilon^S_{n+1}\right)^{\beta_{11}} \left(\varepsilon^S_{n}\right)^{\beta_{12}} \left(\varepsilon^F_{n+1}\right)^{\beta_{21}} \left(\varepsilon^F_{n}\right)^{\beta_{22}}.$$
- *PIDMR* (a multirate extension of the PID single-rate controller):
$$H_{n+1} = H_n \left(\varepsilon^S_{n+1}\right)^{\alpha_1} \left(\varepsilon^S_{n}\right)^{\alpha_2} \left(\varepsilon^S_{n-1}\right)^{\alpha_3},$$
$$M_{n+1} = M_n \left(\varepsilon^S_{n+1}\right)^{\beta_{11}} \left(\varepsilon^S_{n}\right)^{\beta_{12}} \left(\varepsilon^S_{n-1}\right)^{\beta_{13}} \left(\varepsilon^F_{n+1}\right)^{\beta_{21}} \left(\varepsilon^F_{n}\right)^{\beta_{22}} \left(\varepsilon^F_{n-1}\right)^{\beta_{23}}.$$

Outline

Kværno-Prothero-Robinson (KPR) test problem

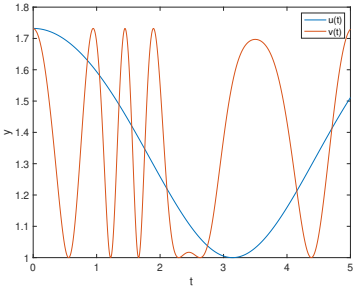
$$\begin{pmatrix} u'(t) \\ v'(t) \end{pmatrix} = \begin{bmatrix} G & e \\ e & -1 \end{bmatrix} \begin{pmatrix} \left(u^2 - p - 2\right)/(2u) \\ \left(v^2 - q - 2\right)/(2v) \end{pmatrix} + \begin{pmatrix} p'(t)/(2u) \\ q'(t)/(2v) \end{pmatrix},$$

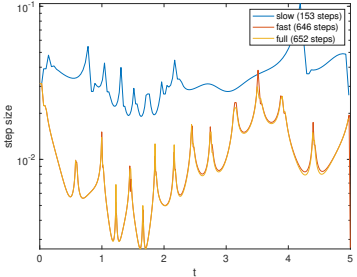
over $0 \le t \le 5$, where $p(t) = \cos(t)$ and $q(t) = \cos(\omega t(1 + e^{-(t-2)^2}))$.

The analytical solution is $u(t) = \sqrt{2 + p(t)}$ and $v(t) = \sqrt{2 + q(t)}$.

- e that determines the strength of coupling between the time scales,
- $G < 0$ determines the stiffness at slow time scale,
- w that determines the time-scale separation factor.

Top right: analytical solutions with $G = -10$, $e = 1/10$, $\omega = 5$.
Bottom right: internal single-rate time steps.





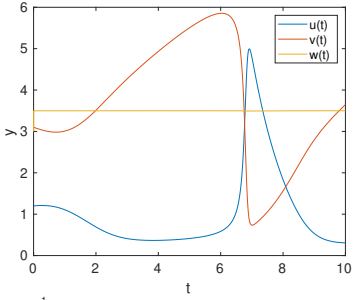
Stiff Brusselator test problem

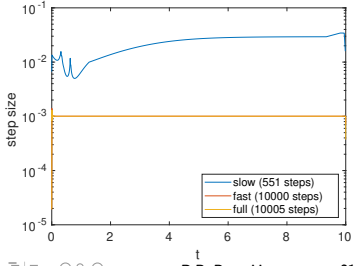
$$\begin{pmatrix} u'(t) \\ v'(t) \\ w'(t) \end{pmatrix} = \begin{pmatrix} a-(w+1)u+vu^2 \\ wu-vu^2 \\ \frac{b-w}{\epsilon}-wu \end{pmatrix}, \quad 0 \le t \le 10.$$

Initial condition $\begin{pmatrix} 1.2 & 3.1 & 3 \end{pmatrix}^T$; parameters $a = 1$, $b = 3.5$.

- ϵ is the stiffness parameter. MRI methods may be used to circumvent implicit solves using explicit substepping.
- Sharp solution transition at $t \approx 6.5$ stresses temporal adaptivity (particularly at the slow time scale).

Top right: example solution with $\epsilon = 1/2500$.
Bottom right: internal single-rate time steps.





Embedded MRI methods

We test MRI adaptivity using 15 embedded MRI methods of varying order and type:

- MRI-GARK methods from [Sandu, 2019]:
  - Explicit `MRI_ERK22a`, `MRI_ERK22b`, `MRI_ERK33a`, and `MRI_ERK45a` methods (orders 2, 2, 3, 4);
  - Implicit `MRI_IRK21a`, `MRI_ESDIRK34a`, and `MRI_ESDIRK46a` methods (orders 2, 3, 4);
- `MRI_RALSTON2` explicit MRI-GARK method (order 2) from [Roberts, 2022];
- `MRISR21`, `MRISR32`, and `MRISR43` IMEX-MRI-SR methods (orders 2, 3, 4) from [Fish et al., 2024];
- `MERK21`, `MERK32`, `MERK43`, and `MERK54` explicit MERK methods (orders 2, 3, 4, 5) from [Luan et al., 2020] (custom embeddings).

Each of the above methods include an embedding with order of accuracy one lower.

MRI adaptive accuracy – KPR

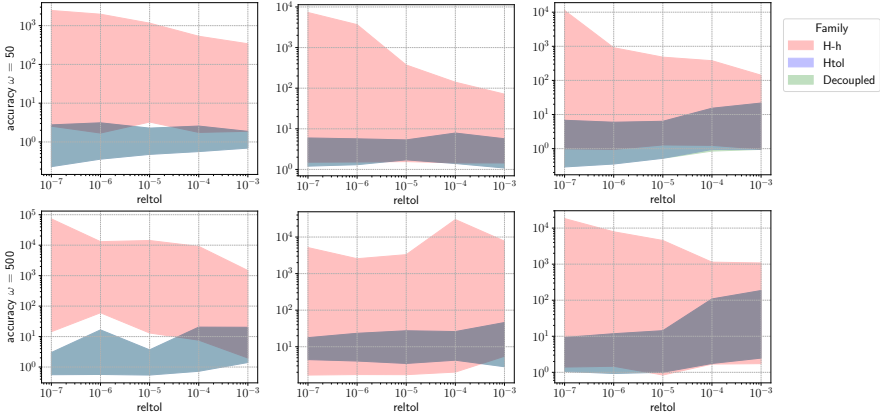
We compare the ability of each method to achieve the target accuracy over all components l and time steps n for the KPR test: accuracy $= \max_{n,l} \left| \dfrac{y_{n,l} - y_{ref,l}(t_n)}{\text{abstol} + \text{reltol}\,|y_{ref,l}(t_n)|} \right|$.

Left order 2, Center order 3, Right orders 4-5

Shading includes all results from a family.

H-Tol and Dec robust across all tests.

H-h accuracy varies wildly across methods.

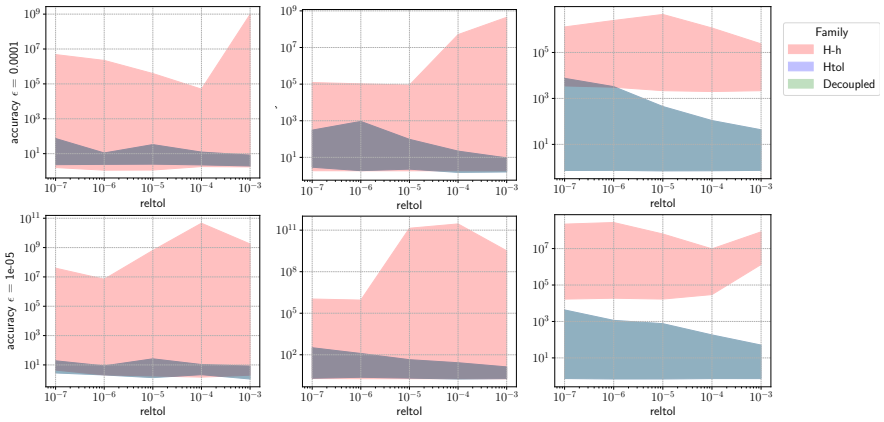


MRI adaptive accuracy – Stiff Brusselator

We repeat the previous experiment for the stiff Brusselator test:

H-Tol and Dec robust for all but `MRI_ESDIRK34a`, `MRI_ESDIRK46a`

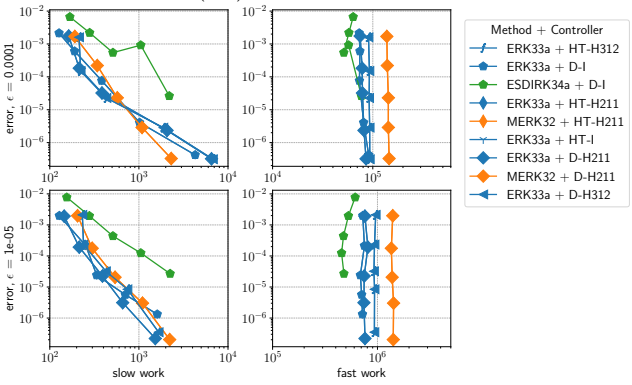
H-h fails for all but `MERK21` and `MERK32`.



MRI adaptive efficiency

We ran 4200 tests varying problem setup, MRI method, adaptivity controller, and tolerance, to determine the most "efficient" combinations. We tabulated their average rank for each problem and method order, computing their z-scores in comparison to competing methods (negative is good):

Below: best $\mathcal{O}(H^3)$ methods for stiff Brusselator.



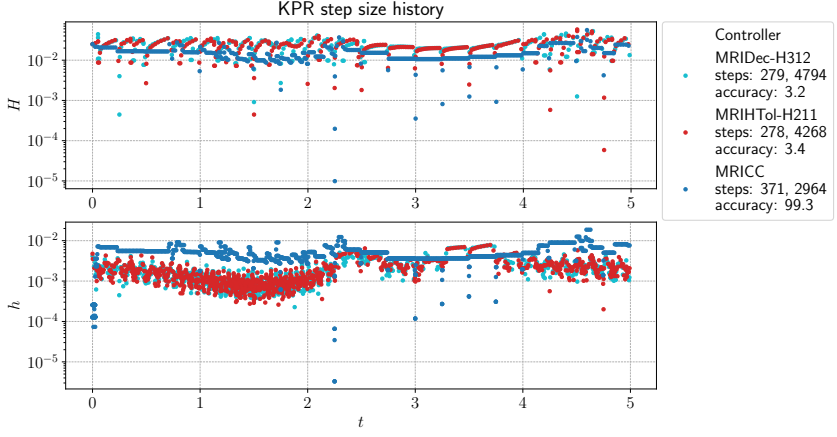
Overall conclusions:

- The top MRI adaptivity methods (z-score): MRIDec-I (-0.4), MRIHTol-I (-0.25), and MRIHTol-H312 (-0.21).
- The top $\mathcal{O}(H^2)$ MRI methods: `MRI_IRK21a` (-0.87), `MRI_RALSTON2` (-0.18), and `MRI_ERK22b` (-0.17)
- The top $\mathcal{O}(H^3)$ MRI methods: `MRI_ERK33a` (-0.67) and `MERK32` (0.05)
- The top $\mathcal{O}(H^4)$ MRI methods: `MRI_ERK45a` (-0.52), `MRI_ESDIRK46a` (-0.12), and `MERK43` (0.02)

MRI adaptive step histories – KPR

We plot the slow and fast step size histories for a few adaptivity controllers using `MRI_ERK33a` at reltol $= 10^{-5}$, listing the total numbers of slow and fast time steps, and accuracy ratio.

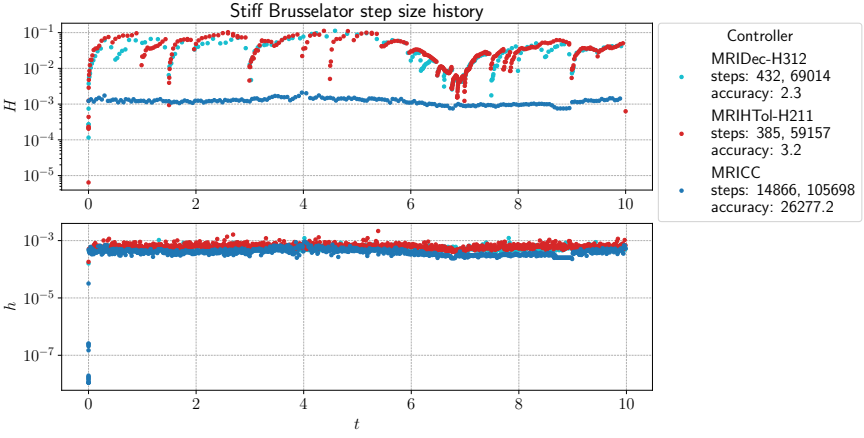
H-Tol is slightly more efficient than Dec, but MRICC does not allow sufficient time scale separation.



MRI adaptive step histories – stiff Brusselator

We repeat the previous experiment for the stiff Brusselator problem.

$H\text{-}Tol$ is again slightly more efficient than Dec, while MRICC is wildly inaccurate.



Outline

Conclusions and future work

- To match today's rich ecosystem of scientific applications, we require an equally rich "toolkit" of time integration methods.
- I aim to expand this toolkit to higher order, increased flexibility, and robust temporal adaptivity.
- With respect to multirate adaptivity:
  - Both the Dec and $H\text{-}Tol$ families show robust adaptive control across a wide range of problem types, tolerances, and MRI methods.
  - Of the two, $H\text{-}Tol$ shows slightly better computational efficiency.
  - $H\text{-}h$, however, struggled significantly, since it seems to artificially constrain the step size ratio $M_n = H_n/h_n$.
- All MRI methods in this talk (except MERB), along with Dec an $H\text{-}Tol$ families, are available within the ARKODE solver from the SUNDIALS library.
- We are currently incorporating these methods within large-scale application codes, including *Perturbo* (solid state physics) and *BOUT++* (fusion).

Outline

IMEX-MRI-GARK Construction

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

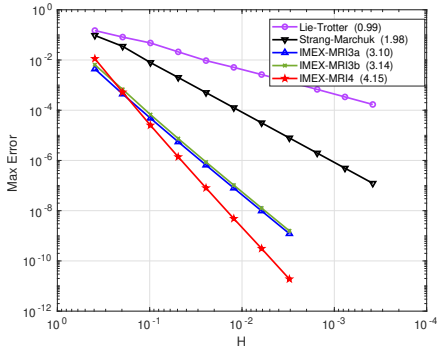
Begin with an IMEX-ARK pair $\{A^I, b^I, c^I; A^E, b^E, c^E\}$ where $c^I = c^E \equiv c$ with $0 = c_1 \le \cdots \le c_{\tilde{s}} \le 1$.

- 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".

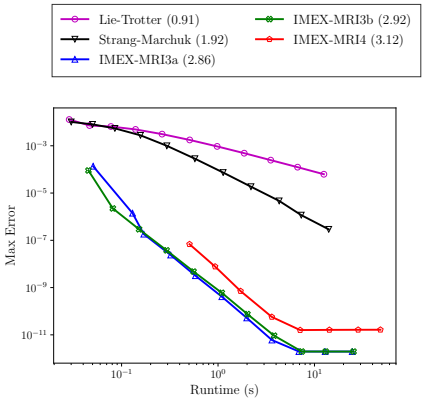
Note: we found it challenging to construct embedded IMEX-MRI-GARK methods, largely due to our reliance on IMEX-ARK base methods and the "sorted" abscissa requirement.

IMEX-MRI-GARK Convergence/Efficiency

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



Nonlinear Kværnø-Prothero-Robinson test problem convergence.



Stiff Brusselator PDE test runtime efficiency. Largest two step sizes were unstable for IMEX-MRI4.

IMEX-MRI-SR Construction

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

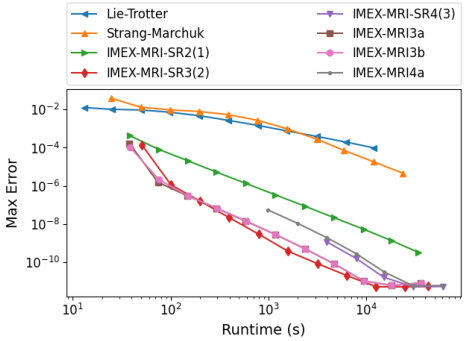
Again take an an IMEX-ARK pair $\{A^I, b^I, c^I; A^E, b^E, c^E\}$ where $c^I = c^E \equiv c$ (not necessarily sorted).

- 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.

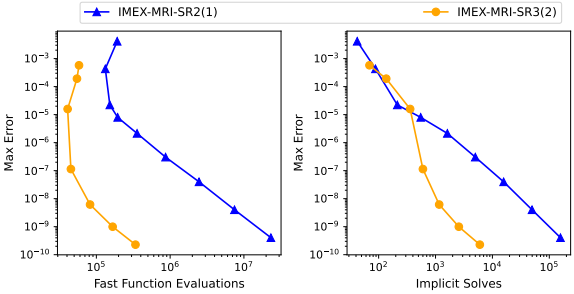
IMEX-MRI-SR Convergence/Efficiency – Stiff Brusselator PDE

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

Fixed step size runtime efficiency: IMEX-MRI-SR, IMEX-MRI-GARK, and IMEX-MRI versions of Lie–Trotter and Strang–Marchuk methods.



Adaptive IMEX-MRI-SR efficiency: modified problem with time-dependent advection, diffusion and reaction coefficients, using tolerances 10^{-k} with $k = 1, \ldots, 9$:



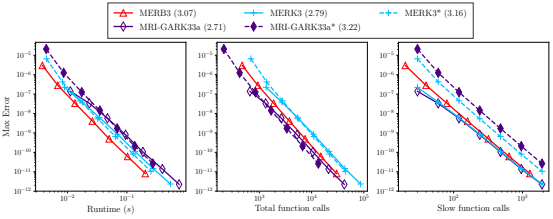
MERK and MERB Convergence/Efficiency – Reaction-Diffusion PDE

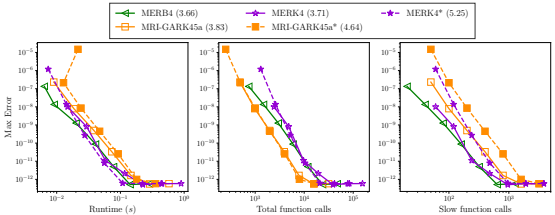
[Luan, Chinomona & R., *SISC*, 2020; Luan, Chinomona & R., *J. Sci. Comput.*, 2022]]

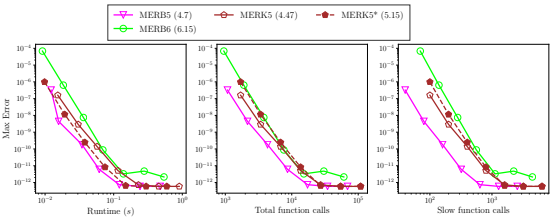
Problem: $u_t = \epsilon u_{xx} + \gamma u^2(1-u),\ x \in (0,5),\ t \in [0,5]$,
with $\gamma = 0.1$, $\epsilon = 10^{-2}$, $\lambda = \sqrt{5}$, $u(x,0) = (1+\exp(\lambda(x-1)))^{-1}$, and $u_x(0,t) = u_x(5,t) = 0$.

Efficiency plots (runtime, RHS calls, $\mathcal{N}_n$ 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.







MRI temporal error estimation – ε_n^S and ε_n^F

Slow error may be estimated as usual for embedded methods, $\varepsilon_n^S = \|y_n - \tilde{y}_n\|$, but non-intrusive estimates for ε^F are more challenging. We consider four strategies:

- Three assume that at each sub-step $t_{n,m}$ the fast integrator computes a local error estimate, $\varepsilon_{n,m}^F$, and itself is temporally adaptive with relative tolerance, reltol^F. We accumulate these via:

$$\varepsilon_{n,max}^F = \text{reltol}^F \max_{m\in M} \varepsilon_{n,m}^F \qquad \text{"Maximum accumulation,"}$$

$$\varepsilon_{n,add}^F = \text{reltol}^F \sum_{m\in M} \varepsilon_{n,m}^F \qquad \text{"Additive accumulation," or}$$

$$\varepsilon_{n,avg}^F = \varepsilon_{n,add}^F/|M| \qquad \text{"Average accumulation,"}$$

  where M is the set of all steps since the fast error accumulator has been reset.

- The fourth uses fixed fast steps h and kh to compute y_h^F and y_{kh}^F, estimating

$$\varepsilon_{n,dbl}^F = \frac{\|y_h^F - y_{kh}^F\|}{|k^p - 1|} \qquad \text{"Double-step accumulation,"}$$

  where p is the global order of accuracy for the fast method.

Accumulated error comparisons

We partition time into 20 subintervals. We evolve over each $[t_k, t_{k+1}]$ with the initial condition reset to the reference $y(t_k) = y_{ref}(t_k)$. We compare the estimated ε_X^F against the "true" error, $\varepsilon_{ref}^F(t_{k+1})$ via

$$\text{ratio}_X(t_{k+1}) = \frac{\varepsilon_{ref}^F(t_{k+1})}{\varepsilon_X^F(t_{k+1})},$$

- Left adaptive, right fixed-step.
- Top KPR, bottom stiff Brusselator.
- Adaptive: both the Average and Maximum accumulators work well for all test problems and tolerances.
- Fixed-step: no accumulator works well across all tests.