High-order multirate time integration for multiphysics PDE systems

Daniel R. Reynolds[1], Rujeko Chinomona[2], Vu Thai Luan[3]

reynolds@smu.edu, rujeko.chinomona@temple.edu, luan@math.msstate.edu

[1]Department of Mathematics, Southern Methodist University
[2]Department of Mathematics, Temple University
[3]Department of Mathematics and Statistics, Mississippi State University

SIAM Southeastern Atlantic Section Conference
18 September 2021

Outline

Outline

Multiphysics/Multirate Problems [Keyes et al. 2013]

"Multiphysics" problems couple models together either in the bulk (cosmology, combustion) or across interfaces (climate, tokamak fusion), challenging traditional numerical methods.

- "Multirate": processes evolve on different time scales, may not admit analytical reformulation.
- Existence of stiff components prohibits fully explicit methods.
- Nonlinearity and insufficient differentiability challenge fully implicit methods.
- Parallel scalability demands optimal algorithms – while robust/scalable algebraic solvers exist for some pieces (e.g., FMM for particles, multigrid for diffusion), none are optimal for the full problem.

Here we'll consider the prototypical problem

$$y'(t) = f^S(t,y) + f^F(t,y), \quad t \in [t_0, t_f], \quad y(t_0) = y_0 \in \mathbb{R}^n.$$

- $f^S(t,y)$ contains "slow" components that evolve with time scale H, and
- $f^F(t,y)$ contains "fast" components that evolve with time scale $h \ll H$.
- f^S or f^F may be further decomposed into stiff/nonstiff or fast/slow parts.

Legacy Multirate Approaches

Historical approaches for the time step $y_n \approx y(t_n) \to y_{n+1} \approx y(t_n + H)$ include first-order Lie–Trotter splittings and subcycling, e.g.,

1. Compute: $y_n^{(1)} = y_n + H f^S(t_n, y_n)$
2. Evolve: $v'(\tau) = f^F(t_n + \tau, v)$, for $\tau \in [0, H]$, $v(0) = y_n^{(1)}$, and let $y_{n+1} = v(H)$.

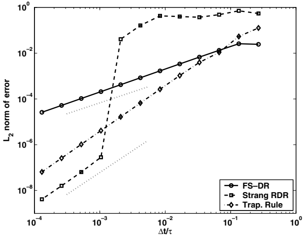
or potentially "Strang–Marchuk" splitting, e.g.,

1. Compute: $y_n^{(1)} = y_n + \frac{H}{4} f^S(t_n, y_n) + \frac{H}{4} f^S\left(t_n + \frac{H}{2}, y_n + \frac{H}{2} f^S(t_n, y_n)\right)$
2. Evolve: $v'(\tau) = f^F(t_n + \tau, v)$, for $\tau \in [0, H]$, $v(0) = y_n^{(1)}$ and let: $y_n^{(2)} = v(H)$
3. Compute: $y_{n+1} = y_n^{(2)} + \frac{H}{4} f^S\left(t_n + \frac{H}{2}, y_n^{(2)}\right) + \frac{H}{4} f^S\left(t_{n+1}, y_n^{(2)} + \frac{H}{2} f^S\left(t_n + \frac{H}{2}, y_n^{(2)}\right)\right)$

Legacy Multirate Approaches – Shortcomings

Low accuracy due to loose "initial condition" coupling:

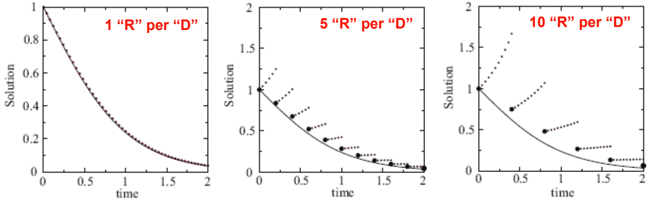
- Lie-Trotter is $\mathcal{O}(H)$ and Strang-Marchuk is $\mathcal{O}(H^2)$.
- Extrapolation can improve this but at significant cost.



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

Poor stability:

- Even with "stable" step sizes for each part, unstable modes may arise.



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

Multirate Improvements

In recent decades, improvements to accuracy and stability for multirate numerical methods have generally taken one of two forms:

A. Tighter slow $\leftrightarrow$ fast coupling[1]:

- \+ typically only require a single traversal of the step $[t_n, t_{n+1}]$ by each operator
- − typically only enable $\mathcal{O}(H^2)$ or $\mathcal{O}(H^3)$

B. Extrapolation / deferred correction techniques[2]:

- \+ potential for arbitrarily-high accuracy
- − require many traversals of the step $[t_n, t_{n+1}]$

In this work we pursue approach A.

[1] Gear & Wells 1984; Günther, Kværnø & Rentrop 1999-2002; Constantinescu & Sandu 2007-09; Fok 2016; Arnold, Galant, Knoth, Schlegel, Wensch & Wolke 2009-14

[2] Engstler, Hairer, Lubich, Ostermann 1990-97; Constantinescu & Sandu 2010-13, Bouzarth & Minion 2010

"Infinitesimal" Multirate Methods (MIS, RFSMR) [Knoth & Wolke 1998; Schlegel et al. 2009; ...]

The infinitesimal family of multirate methods allow up to $\mathcal{O}(H^3)$ accuracy, through more tightly coupling the fast and slow operators. Returning to our prototypical problem:

$$y'(t) = f^S(t,y) + f^F(t,y), \quad t \in [t_0, t_f], \quad y(t_0) = y_0 \in \mathbb{R}^n.$$

- the slow component is integrated using an explicit "outer" RK method, $T_O = \{A, b, c\}$, where $0 = c_1 \le c_2 \le \ldots \le c_s \le 1$;
- the fast component is assumed to *exactly* solve a modified IVP (next slide);
- practically, this fast solution is subcycled using an "inner" RK method.

MIS Algorithm

A single MIS step $y_n \to y_{n+1}$ has the form:

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

Here, the constants α_{ij} are uniquely defined from the "slow" Butcher table, T_O.

- When $c_i = c_{i-1}$, the IVP "solve" reduces to a standard RK update.
- The "fast" IVP in step 2b may be solved using any viable algorithm.

MIS Properties

MIS methods satisfy a number of desirable properties:

- $\mathcal{O}(H^2)$ if both inner/outer methods are at least $\mathcal{O}(H^2)$.
- $\mathcal{O}(H^3)$ if both inner/outer methods are at least $\mathcal{O}(H^3)$, and T_O satisfies

$$\sum_{i=2}^{s} (c_i - c_{i-1})(e_i + e_{i-1})^T Ac + (1 - c_s)\left(\frac{1}{2} + e_s^T Ac\right) = \frac{1}{3}.$$

- The inner method may be a subcycled T_O, enabling a *telescopic* multirate method (i.e., n-rate problems supported via recursion).
- Both inner/outer methods can utilize problem-specific tables (SSP, etc.).
- Highly efficient – only a single traversal of $[t_n, t_{n+1}]$ is required. To our knowledge, MIS are the most efficient $\mathcal{O}(H^3)$ multirate methods available.

Higher-order MIS-like methods [Sandu 2019; Bauer & Knoth 2019; Sexton & R. 2019]

In the last few years groups have worked to extend the MIS approach to $\mathcal{O}(H^4)$ accuracy:

- *MRI-GARK* modifies the fast IVP:

$$r \quad \to \quad r(\tau) = \sum_{j=1}^{i} \gamma_{i,j}\left(\frac{\tau}{(c_i - c_{i-1})H}\right) f^S\left(t_n + c_j H, z_j\right),$$

  and supports "solve-decoupled" implicit methods at the slow time scale (i.e., alternate between DIRK-like solves and fast IVP evolution).

- *extMIS* relaxes the MIS structure slightly, and then develops additional order conditions on T_O.

- *RMIS* computes y_{n+1} as a linear combination of $\left\{f^S(t_n + c_i H, z_i) + f^F(t_n + c_i H, z_i)\right\}_{i=1}^{s}$, additionally enabling conservation of linear invariants.

Outline

Multirate Exponential Runge–Kutta (MERK) & Rosenbrock (MERB) Methods [Luan, Chinomona & R., *SISC* 2020 & *arXiv* 2021]

We consider 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 $\mathcal{J}_n = \frac{\partial F}{\partial y}(t_n, u_n)$.
- "Slow" scale corresponds to $\mathcal{V}_n t + \mathcal{N}_n(t,y)$ with $\mathcal{V}_n = \frac{\partial F}{\partial t}(t_n, y_n)$ and $\mathcal{N}_n(t,y) = F(t,y) - \mathcal{J}_n y - \mathcal{V}_n t$.
- MERK, and MERB for autonomous systems, have $\mathcal{V}_n = 0$, allowing some simplifications.

This problem has the same structure assumed by ExpRK and ExpRB methods, that approximate y_{n+1} via:

$$z_i = y_n + c_i H \varphi_1(c_i H \mathcal{J}_n)\, F(t_n, y_n) + c_i^2 H^2 \varphi_2(c_i H \mathcal{J}_n) V_n + H \sum_{j=2}^{i-1} a_{ij}(H\mathcal{J}_n)\, D_{nj}, \quad 1 \le 2 \le s,$$

$$y_{n+1} = y_n + H\varphi_1(H\mathcal{J}_n)\, F(t_n, y_n) + H^2 \varphi_2(H\mathcal{J}_n) V_n + H \sum_{i=2}^{s} b_i(H\mathcal{J}_n)\, D_{ni},$$

where $D_{ni} = \mathcal{N}_n(t_n + c_i H, z_i) - \mathcal{N}_n(t_n, y_n)$, and $a_{ij}(z)$, $b_i(z)$ are typically linear combinations of $\varphi_k(c_i z)$ and $\varphi_k(z)$, with $\varphi_k(z) = \frac{1}{H^k} \int_0^H e^{(1-\tau/H)z} \frac{\tau^{k-1}}{(k-1)!}\, \mathrm{d}\tau, \; k \ge 1$.

MERK & MERB Construction

Theorem (Luan, Chinomona & R., 2020 & 2021)

Assuming that $a_{ij}(H\mathcal{J}_n)$, $b_i(H\mathcal{J}_n)$ are strictly linear combinations of $\varphi_k(c_iH\mathcal{J}_n)$ and $\varphi_k(H\mathcal{J}_n)$, respectively:

$$a_{ij}(H\mathcal{J}_n) = \sum_{k=1}^{l_{ij}} \alpha_{ij}^{(k)} \varphi_k(c_iH\mathcal{J}_n), \qquad b_i(H\mathcal{J}_n) = \sum_{k=1}^{m_i} \beta_i^{(k)} \varphi_k(H\mathcal{J}_n),$$

then z_i and y_{n+1} are the exact solutions of the "fast" IVPs

$$\begin{aligned} v_{n,i}'(\tau) &= \mathcal{J}_n v_{n,i}(\tau) + p_{ni}(\tau), \quad v_n(0) = y_n, \quad i = 1, \dots, s, \\ v_{n+1}'(\tau) &= \mathcal{J}_n v_{n+1}(\tau) + q_n(\tau), \quad v_{n+1}(0) = y_n \end{aligned}$$

at $\tau = c_iH$ and $\tau = H$, respectively, where

$$p_{n,i}(\tau) = \mathcal{N}_n(t_n, y_n) + (t_n + \tau)\mathcal{V}_n + \sum_{j=2}^{i-1} \left(\sum_{k=1}^{l_{ij}} \frac{\alpha_{ij}^{(k)}}{c_i^k H^{k-1}(k-1)!} \tau^{k-1} \right) D_{nj},$$

$$q_n(\tau) = \mathcal{N}_n(t_n, y_n) + (t_n + \tau)\mathcal{V}_n + \sum_{j=2}^{s} \left(\sum_{k=1}^{m_i} \frac{\beta_i^{(k)}}{H^{k-1}(k-1)!} \tau^{k-1} \right) D_{ni}.$$

MERK & MERB Convergence

Theorem (Luan, Chinomona & R., 2020 & 2021)

Assuming that a MERK or MERB method is constructed from an ExpRK or ExpRB method of global order p, with the associated "fast" IVPs integrated with a step $h = H/m$ using methods with global orders q and r, respectively, then:

$$\|y_n - y(t_n)\| \leq C_1 H^p + C_2 H h^q + C_3 h^r$$

on $t_0 \leq t_n = t_0 + nH \leq t_f$. Here C_1 depends on $t_f - t_0$ but is independent of n and H; C_2 and C_3 depend on the global order error constants of the chosen IVP solvers.

- Note the second term: for an order p method a "fast" solver for z_i need only have order $q = p - 1$.
- Evolve each stage over $[0, c_i H]$ – overall "fast traversal time" is $\left(1 + \sum_{i=1}^{s} c_i\right) H$ (typically $< 3H$).
- Since MERB methods may assume $\nabla \mathcal{N}_n = 0$ they have dramatically fewer order conditions, resulting in fewer fast solves and reduced fast traversal time.

MERK & MERB Methods

Comparison of MERK and MERB methods against Sandu's explicit MRI-GARK methods of comparable order.

- "Slow stages" corresponds to the number of evaluations of $\mathcal{N}_n$
- "Modified IVPs" corresponds to the number of fast IVP solves

Type	Order	Slow stages	Modified IVPs	Fast traversal time
MERK	3	3	3	$2.17H$
	4	6	4	$2.83H$
	5	10	5	$3.2H$
MERB	3	2	2	$1.5H$
	4	2	2	$1.75H$
	5	4	3	$2.08H$
	6	7	3	$1.25H$
MRI-GARK	3	3	3	H
	4	5	5	H

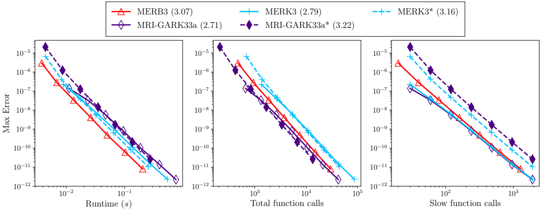
$\star\star\star$ *MERK5, MERB5 and MERB6 are the first-ever infinitesimal methods with order* > 4 $\star\star\star$

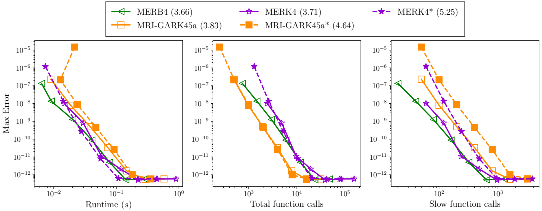
MERK & MERB Results – Reaction-Diffusion PDE test

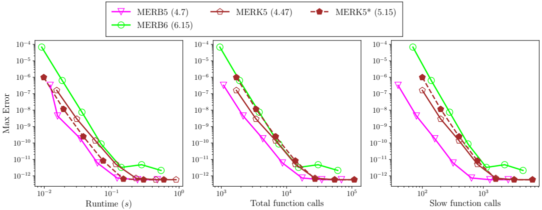
Problem: $u_t = \epsilon u_{xx} + \gamma u^2(1-u), \; x \in (0,5), \; t \in [0,5]$,
with $\gamma = 0.1$, $\epsilon = 10^{-2}$, $\lambda = \sqrt{5}$, $u(x,0) = (1+\exp(\lambda(x-1)))^{-1}$, and $u_x(0,t) = u_x(5,t) = 0$.

Efficiency plots (runtime, total 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.







Outline

Implicit-Explicit Multirate Infinitesimal GARK Methods [Chinomona & R., *SIAM J. Sci. Comput.*, 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:

$$y'(t) = f^I(t,y) + f^E(t,y) + f^F(t,y), \quad t \in [t_0, t_f], \quad y(t_0) = y_0 \in \mathbb{R}^n.$$

These follow the same basic approach as the previous MIS algorithm, but with

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

where $\Delta c_i = c_i - c_{i-1}$, $\gamma_{ij}(\theta) := \sum_{k=0}^{k_{max}} \gamma_{ij}^{\{k\}} \theta^k$ and $\omega_{ij}(\theta) := \sum_{k=0}^{k_{max}} \omega_{ij}^{\{k\}} \theta^k$.

- The coefficients $\Gamma^{\{k\}}, \Omega^{\{k\}} \in \mathbb{R}^{s \times s}$ are lower and strictly lower triangular, respectively.
- We provide order conditions up to $\mathcal{O}(H^4)$, relying on *GARK* framework [Sandu & Günther 2015].
- While theory supports "solve-coupled" methods; our current tables are solve-decoupled.

IMEX-MRI-GARK Construction

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, internal consistency, order conditions, and maximize "joint stability" [Zharovsky et al. SINUM 2015; Sandu SINUM 2019]:

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

$$\mathcal{S}^\sigma_\alpha := \left\{ z^\sigma \in \mathbb{C}^- \ : \ |\arg(z^\sigma) - \pi| \le \alpha \right\}$$

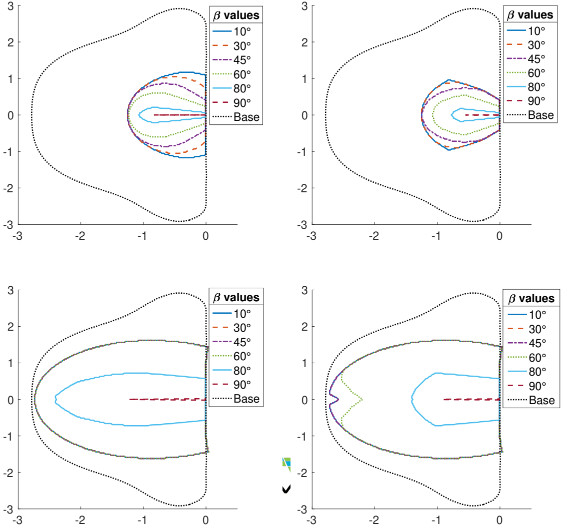
  - $\mathcal{O}(H^3)$: $s^2 + 2(k_{max}+1)s + 2$ algebraic conditions (plus stability opt.)
  - $\mathcal{O}(H^4)$: $s^2 + 2(k_{max}+1)s + 16$ algebraic conditions (plus stability opt.)

IMEX-MRI-GARK Stability – IMEX-MRI-GARK3a & IMEX-MRI-GARK3b (stab. opt.)

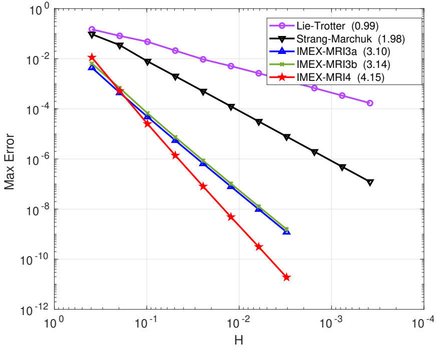
$\mathcal{J}_{\alpha,\beta}$ regions for various implicit sector angles β:

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

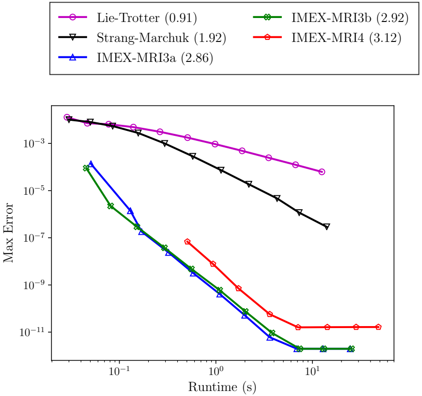
We have a simple $\mathcal{O}(H^4)$ IMEX-MRI-GARK4 table for convergence tests, but it lacks sufficient joint stability for general use.



IMEX-MRI-GARK Results



Nonlinear Kværnø-Prothero-Robinson test problem convergence.



Stiff brusselator PDE test runtime efficiency. $H = \{\frac{1}{40}, \frac{1}{80}\}$ runs were unstable for IMEX-MRI4.

IMEX-MRI-GARK Software

With David Gardner, Cody Balos & Carol Woodward at LLNL, we have implemented support for explicit and solve-decoupled implicit MIS and MRI-GARK methods within the `MRIStep` module of the ARKODE library from SUNDIALS:

- Built-in methods of orders 2, 3, and 4; user-provided MRI-GARK tables $\Gamma^{\{k\}}$ are supported.
- Slow time scale uses a user-defined H that can be varied between steps.
- Fast time scale is evolved using adaptive ARK-IMEX solver (`ARKStep`).

Future updates will include:

- Support for user-supplied fast time scale IVP solver [next minor release, October 2021].
- Support for IMEX-MRI-GARK methods [next major release, late 2021].
- Temporal adaptivity at both the slow and fast time scales [$\sim$1 year off].

Outline

Conclusions

Although simple operator-split subcycling remains standard, new & flexible methods are catching up.

- Problems allowing explicit slow treatment may benefit from higher-order approaches:
  - $\mathcal{O}(H^3)$: MIS, MRI-GARK, MERK, MERB
  - $\mathcal{O}(H^4)$: MRI-GARK, ExtMIS, RMIS, MERK, MERB
  - $\mathcal{O}(H^5)$: MERK, MERB
  - $\mathcal{O}(H^6)$: MERB
- To treat $f^S(t,y)$ implicitly, novel $\mathcal{O}(H^3)$ MIS & MRI-GARK, or $\mathcal{O}(H^4)$ MRI-GARK may be used.
- To split $f^S(t,y) = f^I(t,y) + f^E(t,y)$ we may use novel $\mathcal{O}(H^3)$ or $\mathcal{O}(H^4)$ IMEX-MRI-GARK.

All of these methods allow (a) flexibility for $f^F(t,y)$ via "infinitesimal" structure (explicit, implicit, IMEX, nested multirate), and (b) extension to allow temporal adaptivity of both H and h.

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

- the relative costs of $f^S(t,y)$ and $f^F(t,y)$,
- whether the problem admits a good multirate splitting,
- the desired solution accuracy, ...

Future Work

Much work remains to be done:

- Advanced algorithms for "solve-coupled" MRI-GARK and IMEX-MRI-GARK.
- Rigorous stability theory for additively-partitioned ODE systems (not just $y' = \sum_k \lambda_k y$, that assumes *simultaneous diagonalizability*).
- New $\Gamma^{(k)}$ and $\Omega^{(k)}$ tables (with embeddings) for $\mathcal{O}(H^4)$ MRI-GARK and IMEX-MRI-GARK methods (and order conditions for $\mathcal{O}(H^5)$ or higher).
- Improved support for MERK, MERB and other infinitesimal multirate methods within ARKODE's `MRIStep` module.

Thanks & Acknowledgements

Students & Collaborators:

- SMU: Alex Fish
- LLNL: David J. Gardner, Carol S. Woodward, Cody J. Balos, John J. Loffeld
- LBL: Jean M. Sexton, Andy Nonaka

Current Grant/Computing Support:

- DOE SciDAC & ECP Programs
- SMU Center for Scientific Computation

Software:

- SUNDIALS – https://computation.llnl.gov/projects/sundials

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.

References (all link to web versions)

- Keyes et al., *Int. J. High Perf. Comput. Appl.*, 2013.
- Strang, *SIAM J. Numer. Anal.*, 1968.
- Marchuk, *Aplikace Matematiky*, 1968.
- Ropp & Shadid, *J. Comput. Phys.*, 2005.
- Estep et al., *SINUM*, 2008.
- Gear & Wells, *BIT*, 1984.
- Günther et al., *BIT*, 2001.
- Constantinescu & Sandu, *J. Sci. Comput.*, 2007.
- Sandu & Constantinescu, *J. Sci. Comput.*, 2009.
- Fok, *J. Sci. Comput.*, 2016.
- Knoth & Wolke, *Appl. Numer. Math.*, 1998.
- Schlegel et al., *J. Comput. Appl. Math.*, 2009.
- Schlegel et al., *Appl. Numer. Math.*, 2012.

References (all link to web versions)

- Hairer & Ostermann, *Numer. Math.*, 1990.
- Engstler & Lubich, *Appl. Numer. Math.*, 1997.
- Constantinescu & Sandu, *SIAM J. Sci. Comput.*, 2010.
- Constantinescu & Sandu, *J. Sci. Comput.*, 2013.
- Bouzarth & Minion, *J. Comput. Phys.*, 2010.
- Sandu, *SIAM J. Numer. Anal.*, 2019.
- Bauer & Knoth, *J. Comput. Appl. Math.*, 2019.
- Sexton & Reynolds, *arXiv:1808.03718*, 2019.
- Luan, Chinomona & Reynolds, *SIAM J. Sci. Comput.*, 2020.
- Luan, Chinomona & Reynolds, *arXiv:2106.05385*, 2021.
- Sandu & Günther, *SIAM J. Numer. Anal.*, 2015.
- Chinomona & Reynolds, *SIAM J. Sci. Comput.*, 2021.
- Zharovsky et al., *SIAM J. Numer. Anal.*, 2015.