A New Class of High-Order, Flexible, IMEX Multirate Integrators for Multiphysics Applications

R. Chinomona, Dr. V.T. Luan, Dr. D.R. Reynolds

SMU

February 26, 2019

Multirate Solvers

Multirate solvers involve the use of multiple time steps in evolving different components of a system of ordinary differential equations.

Problems that require multirate solvers arise from multiphysics process (for example in climate modeling) that:

- Have different components that evolve at different rates.
- Mix stiff and nonstiff components.

Multirate Solvers

Multirate solvers involve the use of multiple time steps in evolving different components of a system of ordinary differential equations.

Problems that require multirate solvers arise from multiphysics process (for example in climate modeling) that:

- Have different components that evolve at different rates.
- Mix stiff and nonstiff components.

Focus of talk: The numerical implementation of Multirate Exponential Runge Kutta Methods (MERK).

Multirate Solvers

Multirate solvers involve the use of multiple time steps in evolving different components of a system of ordinary differential equations.

Problems that require multirate solvers arise from multiphysics process (for example in climate modeling) that:

- Have different components that evolve at different rates.
- Mix stiff and nonstiff components.

Focus of talk: The numerical implementation of Multirate Exponential Runge Kutta Methods (MERK).

Main question in the assessment of multirate solvers:
How do we pick test problems in a way that offers meaningful comparison between solvers?

Motivations for multirate solvers

Stiff Problems

- Do not want to use an implicit solver.
- Concern is not on the accuracy of the fast time scale but the stability.
- The fast time step can be fixed to satisfy stability conditions.

Multirate Problems

- The fast time scale contributes significantly to the slow dynamics.
- Capture coupling between slow and fast time scales accurately.
- Investigate what the optimal time scale separation is.

Problem definition & assumptions

Consider the following system:

$$
\begin{aligned}
u'(t) &= F(t, u(t)) \\
&= Au(t) + g(t, u(t)), \\
u(t_0) &= u_0
\end{aligned}
$$

on $t_0 \leq t \leq T$.

- Vector field $F(t, u(t))$.
- $F(t, u(t))$ has a natural splitting into :
  - $Au(t)$ linear (stiff) fast part - cheap
  - $g(t, u(t))$ nonlinear (nonstiff) slow part - expensive

.

Multirate Exponential Runge Kutta (MERK) Integrators

Consider the s-stage explicit one-step exponential Runge-Kutta method:

$$U_{n,i} = \mathrm{e}^{c_i hA} u_n + h \sum_{j=1}^{i-1} a_{ij}(hA) g(t_n + c_j h, U_{n,j}), \quad 1 \leq i \leq s,$$

$$u_{n+1} = \mathrm{e}^{hA} u_n + h \sum_{i=1}^{s} b_i(hA) g(t_n + c_i h, U_{n,i}).$$

where $u_{n+1} \approx u(t_{n+1}) = u(t_n + h)$ and $U_{n,i} \approx u(t_n + c_i h)$ are the internal stages.
Then $u(t_{n+1})$ and $u(t_n + c_i h)$ are exact solutions of:

$$v'(\tau) = Av(\tau) + g(t_n + \tau, u(t_n + \tau)), \quad v(0) = u(t_n),$$

at $\tau = h$ and $\tau = c_i h$ respectively.

Multirate Exponential Runge Kutta (MERK) Integrators

Consider the s-stage explicit one-step exponential Runge-Kutta method:

$$U_{n,i} = \mathrm{e}^{c_i hA} u_n + h \sum_{j=1}^{i-1} a_{ij}(hA) g(t_n + c_j h, U_{n,j}), \quad 1 \le i \le s,$$

$$u_{n+1} = \mathrm{e}^{hA} u_n + h \sum_{i=1}^{s} b_i(hA) g(t_n + c_i h, U_{n,i}).$$

where $u_{n+1} \approx u(t_{n+1}) = u(t_n + h)$ and $U_{n,i} \approx u(t_n + c_i h)$ are the internal stages.
Then $u(t_{n+1})$ and $u(t_n + c_i h)$ are exact solutions of:

$$v'(\tau) = Av(\tau) + g(t_n + \tau, u(t_n + \tau)), \quad v(0) = u(t_n),$$

at $\tau = h$ and $\tau = c_i h$ respectively.
MERK methods: Find modified ODEs whose exact solutions at $\tau = c_i h$ and $\tau = h$ are $U_{n,i}$ and u_{n+1} and solve them numerically to get approximations $\hat{U}_{n,i}$ and $\hat{u}_{n+1}$.

MERK Strategy

1. Set $\hat{U}_{n,1} = \hat{u}_n \approx u_n$.
2. $\hat{U}_{n,1}$ is now known. Evaluate $\hat{p}_{n,2}(\tau)$ and solve modified ODE:

$$\hat{y}'_{n,2}(\tau) = Ay_{n,2}(\tau) + \hat{p}_{n,2}(\tau), \quad \hat{y}_{n,2}(0) = \hat{u}_n$$

   on $[0, c_2 h]$ to obtain $\hat{U}_{n,2} \approx \hat{y}_{n,2}(c_2 h)$.
3. $\hat{U}_{n,1}, \hat{U}_{n,2}$ are now known. Evaluate $\hat{p}_{n,3}(\tau)$ and solve modified ODE to obtain $\hat{U}_{n,3} \approx \hat{y}_{n,3}(c_3 h)$.

   $\vdots$
4. $\hat{U}_{n,1}, \cdots \hat{U}_{n,s-1}$ are now known. Evaluate $\hat{p}_{n,s}(\tau)$ and solve modified ODE to obtain $\hat{U}_{n,s} \approx \hat{y}_{n,s}(c_s h)$.
5. Knowing all $\hat{U}_{n,i}$ we find $\hat{q}_n(\tau)$ and solve:

$$y'_n(\tau) = Ay_n(\tau) + \hat{q}_n(\tau), \quad y_n(0) = \hat{u}_n$$

   on $[0, h]$ to find $\hat{u}_{n+1}$.

MERK Algorithms

For in-depth discussion, please see:

- "*On the Derivation of a New Class of Multirate Methods Based on Exponential Integrators*", Vu Thai Luan
  MS390: Friday 11:30 -11:50am

MERK highlights

- MERK methods expand on the idea of using a modified ODE to evolve the fast time scale from one slow stage to another.
- MERK methods do not invlove matrix function evaluations.
- Currently, MERK methods up to fifth order have been generated though in theory, arbitrary order is possible.

MERK Implementation

- We have a macro time-step H and a micro time-step h is used in evaluating the modified ODEs.
- The slow and fast time scales are separated by a factor of m.
- We consider three MERK methods:
  - MERK3 - 3 stages, 3^{rd} order.
  - MERK4 - 6 stages, 4^{th} order
    Note : $U_{n,3}$ and $U_{n,4}$ share the same modified ODE. So does $U_{n,5}$ and $U_{n,6}$.
  - MERK5 - 10 stages, 5^{th} order.
    Note : $U_{n,3}$ and $U_{n,4}$ share the same modified ODE.
    So does $U_{n,5}, U_{n,6}$ and $U_{n,7}$;
    $U_{n,8}, U_{n,9}$ and $U_{n,10}$.
- Run comparisons with Knoth & Wolke's Multirate Infinitesimal Step 3^{rd} order method MIS-KW3 [Knoth & Wolke 1998, Schlegel et al. 2009].
  Same concept: Evolve fast time scale using modified ODEs.

Choice of test problems

Due to the varied nature of multirate problems we test MERK methods on two different categories of test problems.

Category I	**Category II**
Stiff fast part	Stiff or non-stiff
Temporal error mostly from slow part	Fast error contributes significantly to overall temporal error
Micro time-step h constant	Time scale separation factor m fixed
Brusselator problem, Reaction Diffusion	One-way coupling and Bidirectional coupling

- For each of the test problems, we show convergence and efficiency plots.
- Efficiency is evaluated using number of function calls (slow, total).
- More emphasis on slow function calls.

Reaction and Diffusion Problem [Savcenco,Hundsdorfer & Verwer 2007]

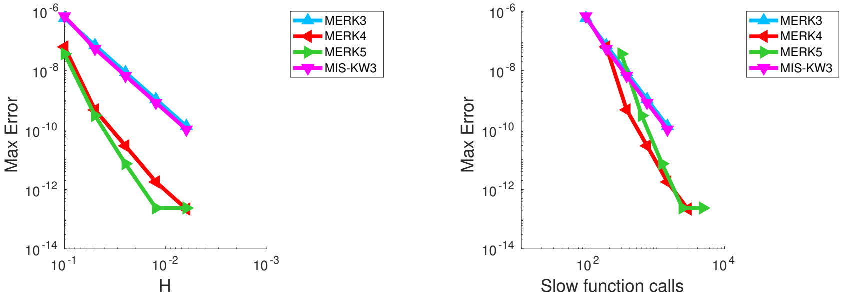
$$u_t = \frac{1}{\epsilon \cdot 10^4} u_{xx} + u^2(1-u),$$

for $0 < x < L, 0 < t \leq T$. Initial and boundary conditions are given by

$$u_x(0,t) = u_x(L,t) = 0, \qquad u(x,0) = (1 + e^{\lambda(x-1)})^{-1},$$

where $\lambda = \frac{1}{2}\sqrt{2\epsilon \cdot 10^4}$.

Reaction and Diffusion Results



- Convergence stagnates at $\sim 10^{-13}$, indicating reference solution accuracy.
- Best-fit convergence rates:
  `MERK3` - 3.03, `MERK4` - 4.93, `MERK5` - 5.71 and `MIS-KW3` - 3.20.
- For slow function calls `MERK4` is the most efficient.

Brusselator Problem (stiff version of [Hairer, Nørsett & Wanner 1993])

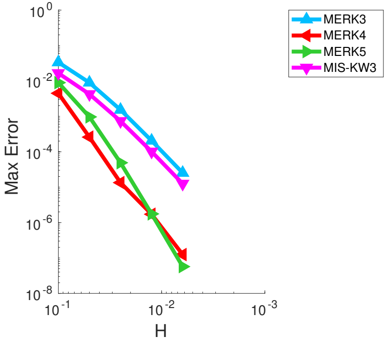
$$\begin{bmatrix} u \\ v \\ w \end{bmatrix}' = \begin{bmatrix} a - (w+1)u + u^2 v \\ wu - u^2 v \\ \frac{b-w}{\epsilon} - uw \end{bmatrix},$$

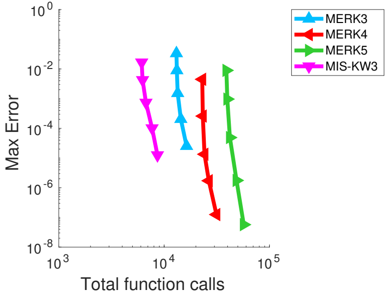
$$\mathbf{u}(0) = [1.2, 3.1, 3]^T$$

on interval $[0, 2]$ with $a = 1, b = 3.5$ and $\frac{1}{\epsilon} = 100$.

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{-1}{\epsilon} & 0 & 0 \end{bmatrix}, \qquad g(t,u) = \begin{bmatrix} a - (w+1)u + u^2 v \\ wu - u^2 v \\ \frac{b}{\epsilon} - uw \end{bmatrix}.$$

Brusselator Results





- Best-fit convergence rates:
  MERK3 - 2.62, MERK4- 3.75, MERK5 - 4.36, MIS-KW3 - 2.61.
- Total number of function calls remains almost constant as error decreases since we held the micro time-step constant.

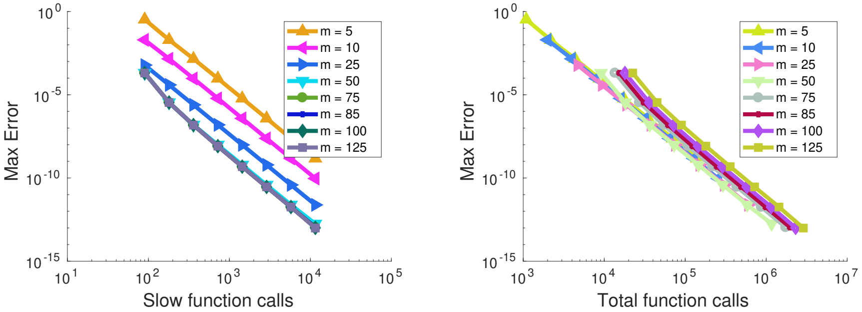
One-way coupling [Estep, Ginting & Tavener 2012]

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix}' = \begin{bmatrix} 0 & -50 & 0 \\ 50 & 0 & 0 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix},$$
$$\mathbf{u}(0) = [1, 0, 2]^T$$

solved on $[0, 1]$.

$$A = \begin{bmatrix} 0 & -50 & 0 \\ 50 & 0 & 0 \\ 1 & -1 & 0 \end{bmatrix}, \qquad g(t, \mathbf{u}) = \begin{bmatrix} 0 \\ 0 \\ -w \end{bmatrix}.$$

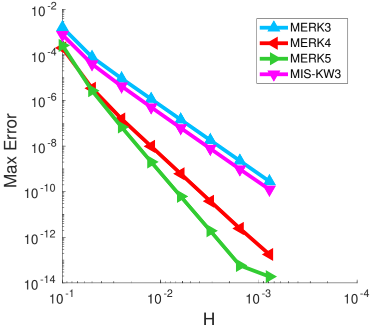
Optimal m for MERK4

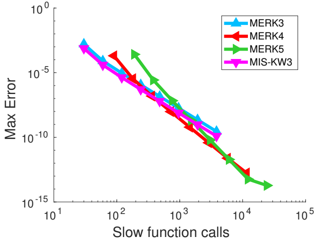


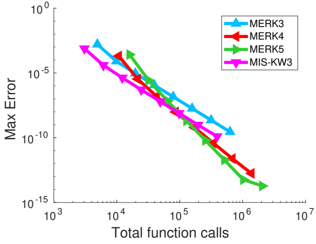
- Slow function calls
  - Smallest m : any increase in m results in the same error for the same work.
  - Smallest of the m values for which lines lie on top of each other.
- Total function calls
  - Largest of the m values for which lines lie on top of each other.

One-way coupling results

- Best-fit convergence rates:
  MERK3 $(m = 75)$ - 3.16,
  MERK4 $(m = 50)$ - 4.28,
  MERK5 $(m = 25)$ - 5.26,
  MIS-KW3 $(m = 75)$ - 3.20.
- MERK4 and MERK5 are eventually most efficient.







Bidirectional coupling

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix}' = \begin{bmatrix} 0 & 100 & 1 \\ -100 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix},$$

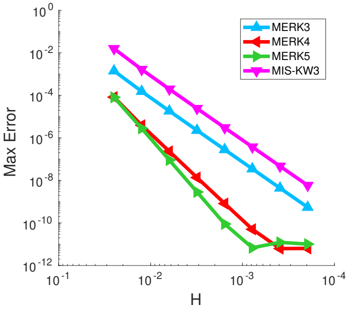
$$\mathbf{u}(0) = \left[\frac{9001}{10001}, \frac{10^5}{10001}, 1000\right]^T$$

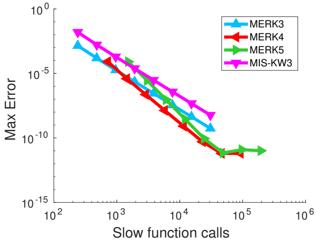
solved on $[0, 2]$.

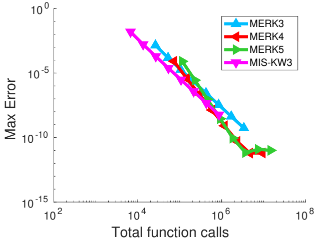
$$A = \begin{bmatrix} 0 & 100 & 0 \\ -100 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \qquad g(t, \mathbf{u}) = \begin{bmatrix} w \\ 0 \\ -w \end{bmatrix}.$$

Bidirectional coupling results

- Convergence stagnates at $\sim 10^{-11}$, indicating reference solution accuracy.
- Best-fit convergence rates:
  MERK3 $(m = 50)$ - 3.07,
  MERK4 $(m = 50)$ - 4.14,
  MERK5 $(m = 25)$ - 4.75,
  MIS-KW3 $(m = 25)$ - 3.08.

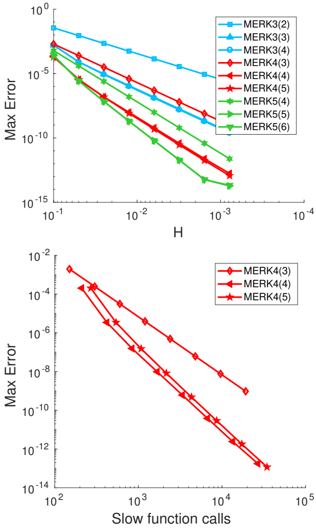




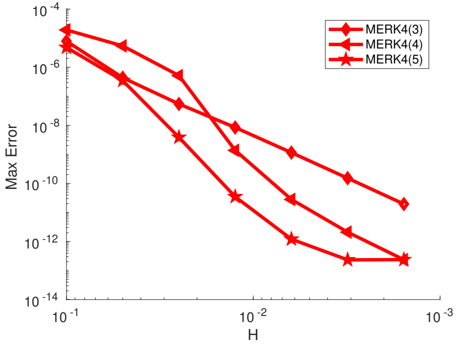


Choice of inner solver [One-way coupling]

- Compare using inner methods of varied accuracy to check for the most effective use of the algorithms.
- Best-fit convergence rates:
  - MERK3(2) - 2.00 , MERK3(3) - 3.16 , MERK3(4) - 3.20.
  - MERK4(3) - 3.00 , MERK4(4) - 4.37 , MERK4(5) - 4.37.
  - MERK5(4) - 4.74 , MERK5(5) - 5.12 , MERK5(6) - 5.23.
- Lower order inner method results in overall low order. Higher order inner method results in overall decrease in efficiency.



Implicit inner method [Reaction-Diffusion problem]



- MERK4 is tested with inner implicit methods of the same, lower and higher orders of convergence.
- Modified ODEs are solved using a single time step.
- Best-fit convergence rates:
  MERK4 (3) - 3.02, MERK4(4) - 4.84 and MERK4(5) - 4.48.

Conclusion & Future Work

Recap

- Investigated the characteristics of a new class of algorithms based on exponential Runge-Kutta methods.
- Discussed a number of test problems to which we can apply the methods, how we apply them and why.
- Confirmed convergence rates.
- Ran comparisons with another multirate method.
- Determined the importance of choosing an appropriate inner method.

Future Considerations

- Extend methods to include a nonlinear fast part.
- Develop higher order methods.
- Investigate time adaptivity.

Polynomials

$$\hat{p}_{n,i}(\tau) = \sum_{j=1}^{i-1} \left(\sum_{k=1}^{l_{ij}} \frac{\alpha_{ij}^{(k)}}{c_i^k h^{k-1}(k-1)!} \tau^{k-1} \right) g(t_n + c_j h, \hat{U}_{n,j}),$$

$$\hat{q}_{n,s}(\tau) = \sum_{i=1}^{s} \left(\sum_{k=1}^{m_i} \frac{\beta_i^{(k)}}{h^{k-1}(k-1)!} \tau^{k-1} \right) g(t_n + c_i h, \hat{U}_{n,i}),$$

MERK4 Characteristics

Conditions to be satisfied: $c_3 \neq c_4, c_5 \neq c_6, c_6 \neq \frac{2}{3}, c_5 = \frac{4c_6-3}{6c_6-4}$.
Two sets of c values:

- $c_2 = \frac{1}{2} = c_3 = c_5; c_4 = \frac{1}{3}; c_6 = 1.$
- $c_2 = c_3 = \frac{1}{2}; c_4 = c_6 = \frac{1}{3}; c_5 = \frac{5}{6}.$

Comparison with MRI-GARK methods [Sandu, arxiv 2018]

- Comparison ran on test problem with one-way coupling.
- Best-fit convergence rates:
  MERK3 $(m = 75)$ - 3.16,
  MERK4 $(m = 50)$ - 4.28,
  MERK5 $(m = 25)$ - 5.26,
  MRI-GARK33 $(m = 25)$ - 3.13
  MRI-GARK45a $(m = 10)$ - 4.20.