Co-design of time stepping algorithms for large aerodynamic simulations

AFOSR BRI 12-2640-06

Adrian Sandu[1], Hong Zhang[1], Paul Tranquilli[1], Ross Glandon[1], Arash Sarshar[1], Mahesh Narayanamurthi[1]

[1] Computational Science Laboratory (CSL)
Department of Computer Science
Virginia Tech

December 19, 2014
AFOSR Workshop

CSL personnel

Hong Zhang
graduated (ANL)

Paul
Tranquilli

Ross
Glandon

Arash
Sarshar

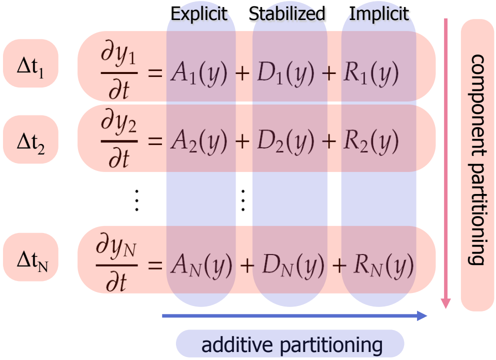
Mahesh
Narayanamurthi

Challenges to solving large evolutionary PDEs and co-design solution approaches I

Numerical approaches that allow the use of different strategies for different components are essential for multiphysics and multiscale systems.

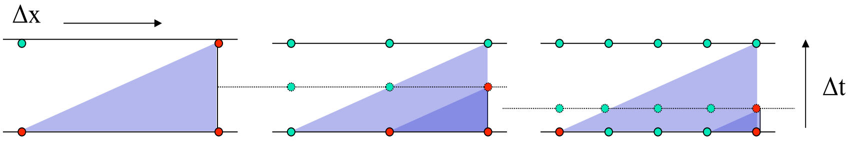
- **Multiphysics: additive partitioning**
  different physics have different dynamics and integrators with appropriate properties are required

- **Multiscale: component partitioning**
  adaptive mesh refinement and variable wave speed restrict the global time step



Challenges to solving large evolutionary PDEs and co-design solution approaches II

1. Explicit time stepping: simple, scalable, CFL bounded



2. Implicit time integration:
   - Unconditionally stable $\rightarrow$ step size determined by accuracy only
   - Huge nonlinear systems coupling all variables in the model at each time step
   - Error estimation and step size control lead to additional data dependencies
3. Our algorithmic co-design goals:
   - Identify and use minimal amount of implicitness
   - Use only operations that are scalable/amenable to acceleration

Implicit-explicit approach

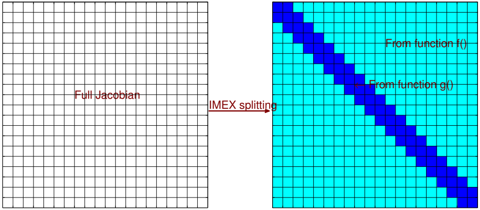


Figure: Consider $y' = f(t,y) + g(t,y)$.
Separate the stiff processes from the non-stiff processes and use implicitness to treat the stiff processes only: IMEX methods

The new K-methods perform implicit integration in a Krylov subspace meant to capture the stiff dynamics

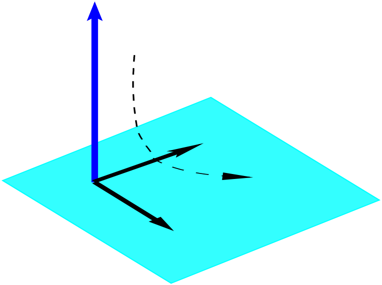


Figure: K-methods separate the small stiff subspace from the non-stiff subspace and use implicitness in the stiff subspace only: ROK, EXPK methods

Rosenbrock methods require (only) the solution of linear systems – in full space

- Initial value problem (semi-discrete PDE)

$$y'(t) = f(y), \quad y(t_0) = y_0, \quad t_0 \le t \le t_{\rm F}\,, \quad y(t), f(y) \in \mathbb{R}^N.$$

- Solution by an s-stage Rosenbrock method:

$$(\mathbf{I} - h\gamma \mathbf{J}_n)\, k_i = h\, f\left(y_n + \sum_{j=1}^{i-1} \alpha_{ij} k_j\right) + h\mathbf{J}_n \sum_{j=1}^{i-1} \gamma_{ij} k_j\,,$$

$$y_1 = y_0 + \sum_{j=1}^{s} b_i k_i\,.$$

- The Jacobian matrix, $\mathbf{J}_n = \partial f / \partial y\,|_{y=y_n}$ appears explicitly.
- Solves linear systems in full space .

Rosenbrock-Krylov methods solve linear systems in a reduced space

Arnoldi: compute $\mathbf{H}$ and $\mathbf{V}$ for $\mathcal{K}_M\left(\mathbf{J}_n, f_n\right)$

for i = 1 **to** s

$$F_i = f\left(y_n + \sum_{j=1}^{i-1} \alpha_{ij} k_j\right)$$

$$\psi_i = \mathbf{V}^T f_i$$

$$\lambda_i = \left(\mathbf{I}_{M\times M} - h\gamma\mathbf{H}\right)^{-1}\left(h\psi_i + h\mathbf{H}\sum_{j=1}^{i-1}\gamma_{ij}\lambda_j\right)$$

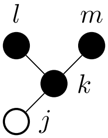
$$k_i = \mathbf{V}\lambda_i + h\left(F_i - \mathbf{V}\phi_i\right)$$

end for i

$$y_{n+1} = y_n + \sum_{i=1}^{s} b_i k_i$$

Accuracy of ROK methods

- Krylov approximation property reduces the set of relevant trees
- ROK conditions up to order three $\equiv$ ROS conditions
- There is one additional TK-tree and ROK condition for order four

	$A_{JK} f^K_{LM} f^L f^M$	$\sum b_j \gamma_{jk} \alpha_{km} \alpha_{kl} \quad = \quad 0$

Theorem (Type 1 order conditions)

A Rosenbrock-K method of type 1 has order p iff the underlying Krylov space has dimension $M \geq p$, and the following order conditions hold:

$$\sum_j b_j \, \phi_j(t) = \frac{1}{\gamma(t)} \quad \forall \, t \in T \;\; \textit{with } \rho(t) \leq p \,,$$

$$\sum_j b_j \, \phi_j(t) = 0 \quad \forall \, t \in TK \backslash T \;\; \textit{with } \rho(t) \leq p \,.$$

Stability and convergence of ROK methods

For accuracy:

- M is small and independent of problem size.

For stability:

- Intuitively M should be sufficiently large such that the Krylov space contains the stiff subspace of the underlying problem (see also Weiner et al)
- How to automatically choose M so that the method is stable is a topic of ongoing work.

Exponential-Krylov methods compute matrix exponential times vectors in small space

Arnoldi: compute $\mathbf{H}$ and $\mathbf{V}$ for $\mathcal{K}_M\left(\mathbf{J}_n, f_n\right)$

for i = 1 **to** s

$$F_i = f\left(y_n + \sum_{j=1}^{i-1} \alpha_{ij} k_j\right)$$

$$\psi_i = \mathbf{V}^T f_i$$

$$\lambda_i = \varphi(h\gamma\mathbf{H})\left(h\psi_i + h\mathbf{H}\sum_{j=1}^{i-1}\gamma_{ij}\lambda_j\right)$$

$$k_i = \mathbf{V}\lambda_i + h\left(F_i - \mathbf{V}\phi_i\right)$$

end for i

$$y_{n+1} = y_n + \sum_{i=1}^{s} b_i\, k_i$$

K-methods outperform traditional solvers on a two dimensional shallow water test problem

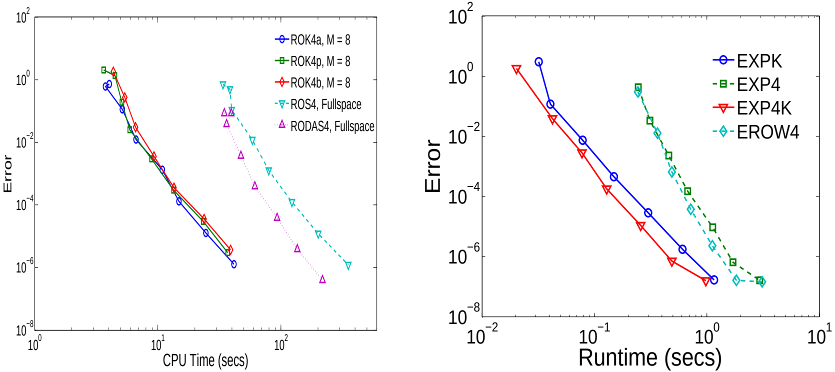
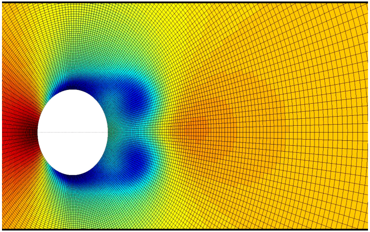
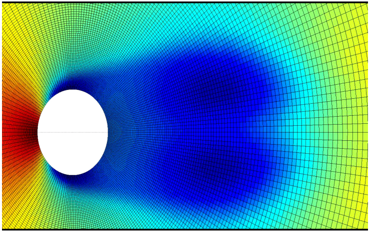
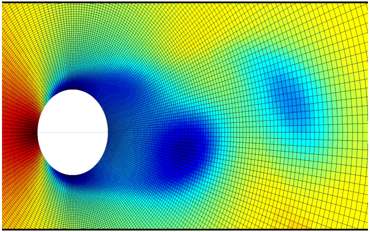


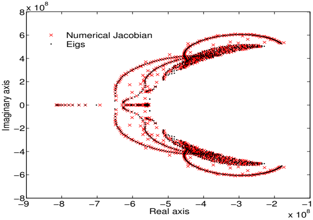
Figure: Performance comparison on shallow water equations using centered finite differences on a 32×32 cartesian grid, $N = 3072$.

Sensei-Lite experiment: flow development in vortex shedding cylinder test case









Time integration results on SENSEI-Lite

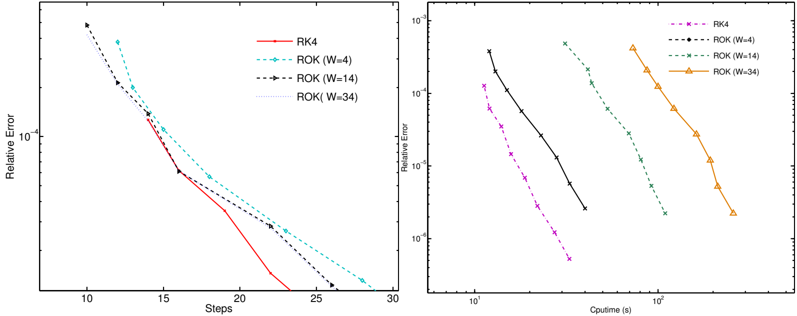


Figure: Performance of time integrators with adaptive time stepping and varying number of basis vectors on SENSEI-Lite

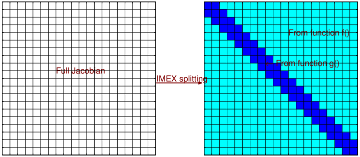
New development: block-orthogonal ROK/EXPK methods

- Construction of an orthogonal basis to the Krylov subspace is the primary efficiency bottleneck.
- Solution: block orthogonal basis – orthogonalize every block (only) against the first.

$$\mathbf{V} = [\mathbf{V}_1 \mathbf{V}_2 \ldots \mathbf{V}_k], \quad \mathbf{V}_i^T \mathbf{V}_1 = \delta_{1,i}\mathbf{I}, \quad \mathbf{V}_i^T \mathbf{V}_i = \mathbf{I}$$

- If we choose blocks of 4, the nth vector requires orthogonalization against at most 8 vectors, as opposed to $n-1$ vectors for a fully orthogonal basis.
- Implementation similar to a standard ROK method, except replacing $\mathbf{H}$ by $\mathbf{H}\mathbf{V}^T\mathbf{V}$.
- Computing $\mathbf{V}^T\mathbf{V}$ replaces much of the cost of orthogonalization, but can be much more easily parallelized since components can be computed in any order.
- Basis vectors do not have to come from a standard Arnoldi iteration. May reuse previous timesteps' basis $\mathbf{J}_n^k \mathbf{J}_{n-1}^k \cdots \mathbf{J}_1^k f_1$ to capture stiff modes inexpensively.

IMplicit-EXplicit time stepping schemes



- Partition the system into two part based on stiffness $y' = f(t, y) + g(t, y)$; treat stiff part implicitly while nonstiff part explicitly
- Existing IMEX families:
  - IMEX Linear Multistep Method (poor stability)
  - IMEX Runge-Kutta methods (order reduction)
- Goal: to develop new IMEX Methods with several properties:
  - no order reduction
  - good stability
  - ...

IMEX General Linear Methods

A two-way partitioned GLM: $(\widehat{\mathbf{A}}, \widehat{\mathbf{B}})$ implicit, $(\mathbf{A}, \mathbf{B})$ explicit

$$
\begin{aligned}
Y_i &= h \left(\sum_{j=1}^{i-1} a_{i,j}\, f(Y_j) + \sum_{j=1}^{i} \widehat{a}_{i,j}\, g(Y_j) \right) + y_i^{[n-1]}, \quad i = 1, \ldots, s\,, \\
y_i^{[n]} &= h \left(\sum_{j=1}^{s} b_{i,j}\, f(Y_j) + \sum_{j=1}^{s} \widehat{b}_{i,j}\, g(Y_j) \right) + \sum_{j=1}^{r} v_{i,j}\, y_j^{[n-1]}, \quad i = 1, \ldots, r\,.
\end{aligned}
$$

Derivation: Assume

$$
y = x + z\,, x' = \tilde{f}(x, z) = f(x+z)\,, z' = \tilde{g}(x, z) = g(x+z)\,,
$$

we do not need to know what x and z are. It works as if the combined state y is advanced through integration.

Starting procedure: Approximate $h^k x^{(k)}(t_0)$, $h^k z^{(k)}(t_0)$, using finite differences on small step solutions.

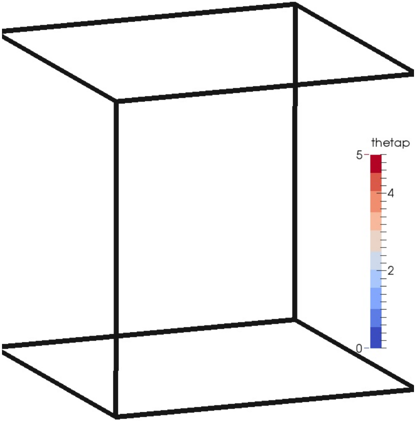
Properties of IMEX GLMs

- High stage order. Order p, stage order q, number of external stages r, number of internal stages s are related by $p = q = r = s$.
- Implicit part is L-stable and constrained explicit stability region is maximized using optimization technique. DIMSIMs are constructed with Runge-Kutta stability.
- No additional coupling condition.

Theorem (Zhang and Sandu)

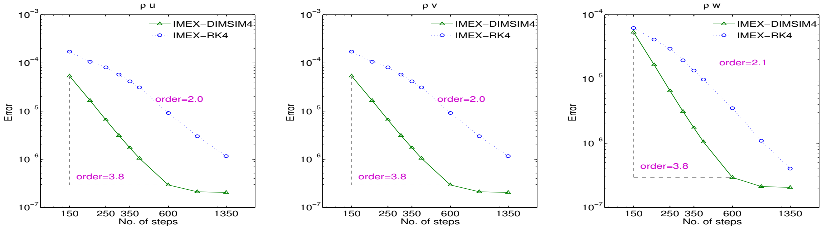
- *Partitioned GLM has order p and stage order $q = p$ $\Updownarrow$ each individual method has order p and stage order $q = p$.*
- *Partitioned GLM has order p and stage order $q = p - 1$ $\Updownarrow$ each constituent method has order p and stage order $q = p - 1$.*

IMEX GLMs on 3D compressible Euler equations I

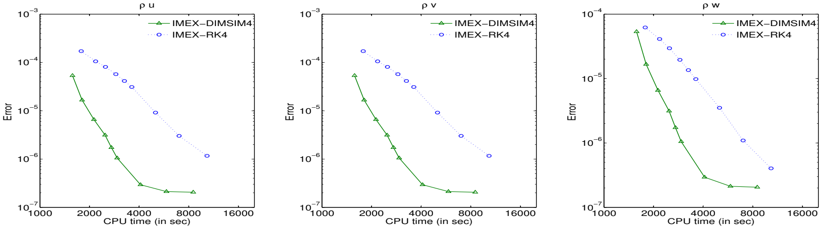


Evolution of the potential temperature for the 3D rising thermal bubble on the domain $[200, 800]^2 \times [-600, 0]$. GMSH-DG code (UCLouvain): discontinuous Galerkin method of order 3 on a uniform grid for spatial discretization. The system has approximately 7×10^4 variables.

IMEX GLMs on 3D compressible Euler equations II



(a) Convergence (3D rising bubble)



(b) Work-precision (3D rising bubble)

Linearly Implicit Runge-Kutta W (LIRK-W) methods I

- Split the initial value problem into linear and nonlinear parts

$$\frac{dy}{dt} = \mathbf{L}y + (F(t,y) - \mathbf{L}y) \ , \quad t_0 \le t \le t_F \ , \quad y(t_0) = y_n \ ;$$

$$y(t), F(t,y) \in \mathbf{R}^N \ , \quad \mathbf{L} \in \mathbf{R}^{N\times N}.$$

- Solution by an s-stage LIRK-W method:

$$(\mathbf{I} - h\,\gamma_{i,i}\,\mathbf{L})\ Y_i = y_n + h\sum_{j=1}^{i-1} a_{i,j}\,F(Y_j) + h\,\mathbf{L}\sum_{j=1}^{i-1}\gamma_{i,j}\ Y_j,$$

$$y_{n+1} = y_n + h\sum_{j=1}^{s} b_j\,F(Y_j) + h\,\mathbf{L}\sum_{j=1}^{s} g_i\,Y_j.$$

- $\mathbf{L} \sim \mathbf{J}_n$ can be arbitrary; for stability it should capture stiff dynamics.
- For discrete 2D Laplacian let $\mathbf{L}_x$ and $\mathbf{L}_y$ be directional derivatives.

Linearly Implicit Runge-Kutta W (LIRK-W) methods II

- Approximate the linear system as

$$\mathbf{I} - h\gamma_{i,i}\,\mathbf{L} := \left(\mathbf{I} - h\gamma_{i,i}\mathbf{L}_x\right)\left(\mathbf{I} - h\gamma_{i,i}\mathbf{L}_y\right).$$

- The products are independent and so can be inverted in parallel.
- LIRK-W maintains obtains full order under such an approximation.

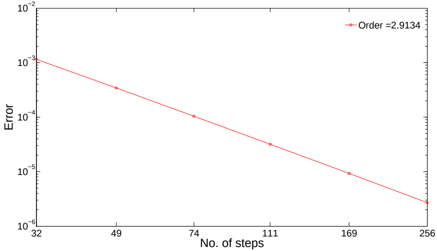
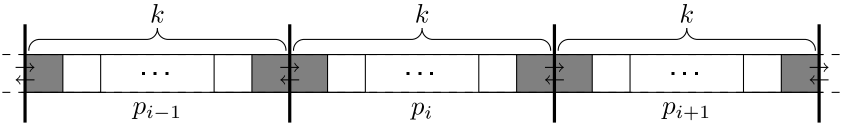


Figure: Convergence of a third order LIRK-W method, applied to Allen-Cahn with approximate matrix factorization

Parallelizing ROK methods

- Parallel Burgers ODE function (local computation):

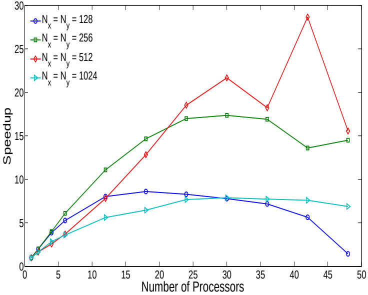
$$f_n^{i_p:j_p} = \frac{1}{2\Delta x} \begin{bmatrix} \left(\mathbf{y}_n^{i_p-1}\right)^2 - \left(\mathbf{y}_n^{i_p+1}\right)^2 \\ \left(\mathbf{y}_n^{i_p:j_p-2}\right)^2 - \left(\mathbf{y}_n^{i_p+2:j_p}\right)^2 \\ \left(\mathbf{y}_n^{j_p-1}\right)^2 - \left(\mathbf{y}_n^{j_p+1}\right)^2 \end{bmatrix}, \quad \mathbf{J}_n\, v \approx \frac{f(\mathbf{y}_n + \varepsilon v) - f(\mathbf{y}_n)}{\varepsilon}$$



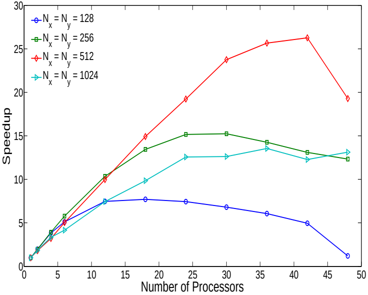
- Scalable Jacobian-vector product (local computation):

$$\left(\mathbf{J}_n\, v\right)^{i_p:j_p} = \frac{1}{\Delta x} \begin{bmatrix} \mathbf{y}_n^{i_p-1}\, v^{i_p-1} - \mathbf{y}_n^{i_p+1}\, v^{i_p+1} \\ \mathbf{y}_n^{i_p:j_p-2}\, v^{i_p:j_p-2} - \mathbf{y}_n^{i_p+2:j_p}\, v^{i_p+2:j_p} \\ \mathbf{y}_n^{j_p-1}\, v^{j_p-1} - \mathbf{y}_n^{j_p+1}\, v^{j_p+1} \end{bmatrix}$$

The scalability of Jacobian-vector products is similar to the scalability of the ODE function



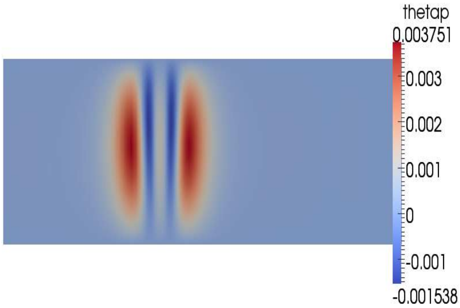
(a) Right-hand side function



(b) Jacobian-vector products

Figure: Speedups for evaluating the ODE function and Jacobian-vector products for shallow water equations. OpenMP parallelization of two-dimensional shallow water equations.

Slowdown for multicore parallel solvers on a two dimensional simulation of acoustic waves



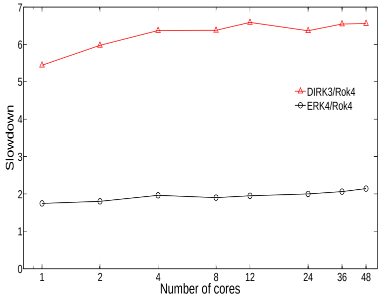


Figure: Slowdown of DIRK and ERK methods compared to the ROK solver. Tests performed on a quad socket machine using AMD Magny-Cours CPUs with a total of 48 cores.

Plans

- Test various methods in Sensei Lite (Chris' group)
- (Need Jacobian-vector products!)
- Run the winning methods on full Sensei (Chris' group)

- Automatic selection on the subspace dimension for stability (Eric)
- Automatic implementation of accelerated Jacobian-vector products (Wu's group)
- Start working on compressible flows (Chris, Danesh)