Fast Solution of Fully Implicit Runge-Kutta and Discontinuous Galerkin in Time for Numerical PDEs, Part I: The Linear Setting

B. S. Southworth, O. Krzysik, W. Pazner, and H. De Sterk (Editor in Chief at SISC)

Part II covers nonlinear problems and DAEs

7 Nov 2025

Motivating Implicit (Runge-Kutta) Methods

1D Heat Equation

Consider our favorite (time-dependent) PDE:

$$\partial_t u = \partial_{xx} u \quad \text{on} \quad \mathbb{R}_+ \times (0, 2\pi) \quad \text{and} \quad u(0) = u_0$$

with periodic BCs. Let's also assume $\int_0^{2\pi} u_0 \,\mathrm{d}x = 0$.

Using a Fourier series, we obtain an infinite system of ODEs:

$$u(t,x) = \sum_{k=-\infty}^{\infty} \hat{u}_k(t) e^{\imath kx} \implies \frac{d}{dt} u_k = -k^2 u_k, \quad k \in \mathbb{Z}.$$

The exact solution is $u_k(t) = e^{-k^2 t} u_k(0)$, so high frequency modes are damped very quickly.

Let's pretend we don't have the exact solution and we want to compute the solution.

Time-stepping the heat equation

Let's restrict $k \in [-N/2, N/2] \setminus \{0\}$, giving the diagonal $N \times N$ system

$$\frac{d}{dt}\vec{u}_N = D\vec{u}, \quad \text{where} \quad \vec{u}_N = \begin{bmatrix} u_{-N/2} \\ \vdots \\ u_{N/2} \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} -N^2/4 & & \\ & \ddots & \\ & & -N^2/4 \end{bmatrix}.$$

Note: As we refine in space (increase N), the largest eigenvalues of D blow up like $N^2/4$.

If we try to discretize with explicit Euler, we get

$$\vec{u}_N^{n+1} = (I + \Delta t D)\vec{u}_N^n \implies \vec{u}_N^n = (I + \Delta t D)^n \vec{u}_N^0.$$

Thus, $\vec{u}_N^n$ blows up as $n \to \infty$ unless all eigenvalues of $I + \Delta t D$ are less than 1:

$$\Delta t < \frac{4}{N^2} \qquad \text{(CFL condition)}.$$

Problem:

Time step is increasingly small for more accurate spatial discretizations.

Same problem happens for any explicit method.

Implicit Runge-Kutta Methods

We all know (and love) implicit Euler and Crank-Nicholson:

$$\vec{u}_N^{n+1} = (I - \Delta t D)^{-1} \vec{u}_N^n \quad \text{and} \quad \vec{u}_N^{n+1} = (2I - \Delta t D)^{-1}(2I + \Delta t D)\vec{u}_N^n.$$

Good news:

No time-step restriction if we only require that $\vec{u}_N$ remains bounded.

Question:

How can we systematically construct high-order implicit methods.

Runge Kutta Methods for $\vec{y}'(t) = \vec{f}(t, \vec{y}(t))$:

$$\vec{y}^{n+1} = \vec{y}^n + \Delta t \sum_{i=1}^{s} b_i \vec{k}_i,$$

$$\vec{k}_i := \vec{f}\left(t + c_i \Delta t, \vec{y}^n + \Delta t \sum_{j=1}^{s} a_{i,j} \vec{k}_j\right), \quad i = 1, \dots, s,$$

Butcher tableau:

$$\begin{array}{c|c} c & A \\ \hline & b^\top \end{array}$$

- s: number of stages
- b_i: "quadrature weights"
- c_i: "collocation" points
- $\vec{k}_i$: stage vectors

Implicit Runge-Kutta Methods

Runge Kutta Methods for $\vec{y}'(t) = \vec{f}(t, \vec{y}(t))$:

$$\vec{y}^{n+1} = \vec{y}^n + \Delta t \sum_{i=1}^{s} b_i \vec{k}_i,$$

$$\vec{k}_i := \vec{f}\left(t + c_i \Delta t, \vec{y}^n + \Delta t \sum_{j=1}^{s} a_{i,j} \vec{k}_j\right), \quad i = 1, \ldots, s,$$

Butcher tableau:

$$\begin{array}{c|c} c & A \\ \hline & b^\top \end{array}$$

- s: number of stages
- b_i: "quadrature weights"
- c_i: "collocation" points
- $\vec{k}_i$: stage vectors

There are many ways to construct implicit Runge-Kutta methods:

1. Collocation methods
   - Gauss-Legendre method (order $2s$)
   - Gauss-Radau IIA (order $2s - 1$)
   - Gauss-Lobatto IIIA (order $2s - 2$)
2. "Discontinuous" collocation methods
   - Gauss-Radau IA (order $2s - 1$)
   - Gauss-Lobatto IIIB (order $2s - 2$)
   - Gauss-Lobatto IIIC (order $2s - 2$)
3. "Perturbed" collocation methods – generate all RK methods

Other Motivations for Implicit Runge-Kutta Methods

Types of Implicit RK methods:

- Diagonally implicit (DIRK): A is lower triangular
- Singly diagonally implicit (SDIRK): DIRK with $\text{diag}(A) = [d, \ldots, d]$.
- Singly implicit (SIRK): A has one unique eigenvalue
- Multiply implicit (MIRK): A has s distinct eigenvalues
- ... insert a random acronym IRK and it probably exists
- Fully implicit (FIRK) or (IRK): no assumption on A.

They all affect the resulting structure of the (non)linear problem to be solved at each time step.

Why FIRK?

- DIRKs have lower accuracy for the same number of stages
- SDIRKs maximum order of $s + 1$ (vs $2s$ for IRK)
- A-stability (and other stability properties)
- Symplectic DIRK limited to 4th order (negative diagonal entries above 2nd order)

Implicit Runge-Kutta Methods for Linear Problems

Resulting Linear System

If we apply a FIRK to the ODE $Mu'(t) = \mathcal{L}u + f(t)$, we get the following linear system at each time step:

$$\left(\begin{bmatrix} M & & \\ & \ddots & \\ & & M \end{bmatrix} - \Delta t \begin{bmatrix} a_{11}\mathcal{L} & \cdots & a_{1s}\mathcal{L} \\ \vdots & \ddots & \vdots \\ a_{s1}\mathcal{L} & \cdots & a_{ss}\mathcal{L} \end{bmatrix} \right) \begin{bmatrix} k_1 \\ \vdots \\ k_s \end{bmatrix} = \begin{bmatrix} g_1 \\ \vdots \\ g_s \end{bmatrix},$$

where $g_i = f(t_n + (\Delta t)c_i) + \mathcal{L}u_n$. In Kronecker product notation, this reads

$$(I \otimes M - \Delta t A \otimes \mathcal{L})k = g.$$

Typically, we factor out A:

$$(A^{-1} \otimes M - \Delta t I \otimes \mathcal{L})(A \otimes I)k = g.$$

- Note that A is small compared to $\mathcal{L}$ – If A is $s \times s$, then the method is order $2s$, $2s - 1$, $2s - 2$, etc., so s is rarely larger than 5-6.

Central Question:

How can we invert $A^{-1} \otimes M - \Delta t I \otimes \mathcal{L}$ quickly?

Inverting $A^{-1} \otimes M - \Delta t I \otimes \mathcal{L}$

1. Some existing work tries to do something that looks like preconditioning A^{-1}; they have the flavor of "preconditioning a FIRK with a DIRK."
   - I think this is a bad idea
   - For a general FIRK, A does not seem to have any exploitable structure
   - Typically only consider SPD $\mathcal{L}$
2. Other existing work factors A^{-1} and then exploits this factorization to reduce $A^{-1} \otimes M - \Delta t I \otimes \mathcal{L}$ to smaller subproblems involving only 1-2 stages coupled together.
   - A is small, so computing the factorization of A is fine and can be done offline.

Red flag:

For some reason, Part I and Part II (nonlinear problems and DAEs) do not use the same method to handle A^{-1}, but both fall under the second category.

Actual Content of Southworth et al.

Setup and Assumptions

- Apply a FIRK to the ODE $Mu'(t) = \mathcal{L}u + f(t)$, M is a "mass matrix".
  - I guess this means that M is SPD, but this is never explicitly stated. Invertible is probably good enough.
- Need to invert the matrix $(A^{-1} \otimes M - \Delta t I \otimes \mathcal{L})$.
- Assume that all eigenvalues of A have positive real part. (Assumption 3)
  - Unclear if there is a proof of this for any arbitrary order FIRK
  - They use Gauss-Legendre, Gauss-Radau IIa, and Gauss-Lobatto IIIc, which numerically satisfy this property up to $s = 5$. I checked up to $s = 40$.
  - There's no Assumption 1 or 2 for some reason.
- Assume that the field of values $W(\mathcal{L}) := \{\langle \mathcal{L}x, x\rangle : \|x\| = 1\}$ is a subset of the left half (complex) plane (including imaginary axis). (Assumption 4)
  - This will ensure later that certain systems are invertible and is used in some condition number estimate proofs.
- (Implicitly) assume that we have efficient solvers for backward Euler steps: $\eta M - (\Delta t)\mathcal{L}$.

Inverting $A^{-1} \otimes M - \Delta t I \otimes \mathcal{L}$

- Define $\hat{\mathcal{L}} := \Delta t M^{-1}\mathcal{L}$ and factor out M:

$$A^{-1} \otimes M - \Delta t I \otimes \mathcal{L} = (I \otimes M)(A^{-1} \otimes I - I \otimes \hat{\mathcal{L}}) =: (I \otimes M)\mathcal{M}$$

- Lemma 5: Let $P(x)$ denote the characteristic polynomial of A^{-1}. If we view $\mathcal{M}$ as a matrix over the commutative ring $\{I, \hat{\mathcal{L}}\}$, then $\det \mathcal{M} = P(\hat{\mathcal{L}})$ and

$$\mathcal{M}^{-1} = (I \otimes P(\hat{\mathcal{L}}))^{-1} \operatorname{adj}(\mathcal{M}),$$

  where $\operatorname{adj}(\mathcal{M})$ is the adjugate of $\mathcal{M}$ (when viewed as a matrix over the commuative ring).

- The diagonal of $\operatorname{adj}(\mathcal{M})$ consists of monic polynomials of $\hat{\mathcal{L}}$ of degree $s-1$, off diagonal degree $s-2$.

- Southworth et al only implement $s \leq 5$, so they have mathematica provide a function to analytically compute the adjugate of a matrix.

- We factor the characteristic polynomial, combining conjugate pairs:

$$P(\hat{\mathcal{L}}) = \prod_{\lambda_k \in \mathbb{R}} (\lambda_k I_N - \hat{\mathcal{L}}) \prod_{\lambda_\ell^{\pm} = \eta_l \pm \imath \beta_l} \left\{(\eta_l I_N - \hat{\mathcal{L}})^2 + \beta_l^2 I_N\right\}$$

Taking a Time Step

Putting everything together, the update takes the form $u_{n+1} = u_n + (\Delta t)v$, where

$$v = \prod_{\lambda_k \in \mathbb{R}} (\lambda_k I_N - \hat{\mathcal{L}})^{-1} \prod_{\lambda_\ell^\pm = \eta_l \pm \imath \beta_l} \left\{(\eta_l I_N - \hat{\mathcal{L}})^2 + \beta_l^2 I_N\right\}^{-1} (b^\top A^{-1} \otimes I_N)\,\mathrm{adj}(\mathcal{M})(I_s \otimes M^{-1})g.$$

- $I_s \otimes M^{-1}$: s applications of M^{-1}.
- $(b^\top A^{-1} \otimes I_N)\,\mathrm{adj}(\mathcal{M})$: $(s-1)^2$ or $(s-1)s$ applications of M^{-1} depending on b.
- $\left\{(\eta_l I_N - \hat{\mathcal{L}})^2 + \beta_l^2 I_N\right\}^{-1}$: one solve with this mess for each complex valued eigenvalue pair of A, typically $\approx s/2$.
- $(\lambda_k I_N - \hat{\mathcal{L}})^{-1}$: Backward Euler solves for each real eigenvalue of A, typically 1.

The final ingredient is a simple preconditioner for $\left\{(\eta_l I_N - \hat{\mathcal{L}})^2 + \beta_l^2 I_N\right\}^{-1}$.

Concern:

$\mathcal{O}(s^2)$ applications of M^{-1} per time step seems like a lot, unless you use DG in space – using a Schur factorization of A^{-1} instead of this mess requires s applications of M^{-1}. Part II uses the Schur factorization $\Longrightarrow$ (?) they know this mess is not the best method.

Preconditioning $(\eta I_N - \hat{\mathcal{L}})^2 + \beta^2 I_N$

Idea: Precondition with two "backward Euler steps" $(\delta I_N - \hat{\mathcal{L}})(\gamma I_N - \hat{\mathcal{L}})$.

Corollary 7: The condition number of

$$(\delta I_N - \hat{\mathcal{L}})^{-1}(\gamma I_N - \hat{\mathcal{L}})^{-1}\{(\eta I_N - \hat{\mathcal{L}})^2 + \beta^2 I_N\}$$

over all $\hat{\mathcal{L}}$ satisfying Assumption 4 is minimized over all $\delta, \gamma \in (0, \infty)$ when $\delta = \gamma = \gamma_* := \sqrt{\eta^2 + \beta^2}$, and this condition number is bounded above by

$$\sqrt{1 + \frac{\beta^2}{\eta^2}},$$

which is achieved for some $\hat{\mathcal{L}}$.

Concerns:

Why do we care about condition number if $\mathcal{L}$ is not symmetric?

Basting & Bänsch '17 only consider $\delta = \gamma$ and showed that if $\mathcal{L}$ is symmetric negative semidefinite, then the condition number with $\delta = \gamma = \gamma_*$ is at most 2.

Basting & Bänsch '17 are cited, but their result is not stated until part II, where it is incorrectly stated. So, either Basting & Bänsch '17 have a mistake or Southworth et al. do not understand the result.

Solver algorithm for $(\eta I_N - \hat{\mathcal{L}})^2 + \beta^2 I_N$

Recall that $\hat{\mathcal{L}} = \Delta t M^{-1}\mathcal{L}$, which we do not want to form. Instead, we multiply by M, which gives systems that look like

$$((\eta M - \Delta t\mathcal{L})M^{-1}(\eta M - \Delta t\mathcal{L}) + \beta^2 M)x = Mb,$$

which we left precondition with

$$(\gamma M - \Delta t\mathcal{L})^{-1}M(\gamma M - \Delta t\mathcal{L})^{-1}, \qquad \gamma = \sqrt{\eta^2 + \beta^2}.$$

Each Krylov iteration requires

- an application of M^{-1} to apply the action of the matrix,
- two backward Euler solves to apply the preconditioner.

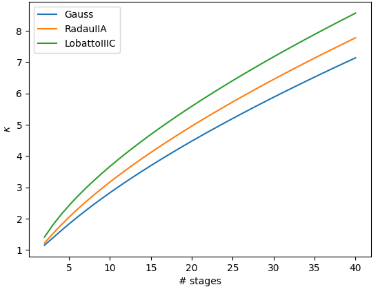
Southworth et al. use inner-iterations to apply the two backward Euler solves.

Concerns:

"Under quite general assumptions on the spatial discretization that yield stable time integration, the preconditioned operator is proven to have condition number bounded by a small, order-one constant, independent of the spatial mesh and time-step size, and with only weak dependence on number of stages/polynomial order; for example, the preconditioned operator for 10th-order Gauss IRK has condition number less than two, *independent of the spatial discretization and time step*."

Behavior of condition number

Condition number bound is $\sqrt{1 + \frac{\beta^2}{\eta^2}}$ for each of the complex eigenvalue pairs.



- "weak dependence on number of stages"
- If you only check up to $s = 5$, of course this is true, as would any statement be on the dependence on the number of stages
- Seems to be no theory for the behavior of the maximum of $\sqrt{1 + \frac{\beta^2}{\eta^2}}$ for any FIRK

Summary of the method

Putting everything together, the update takes the form $u_{n+1} = u_n + (\Delta t)v$, where

$$v = \prod_{\lambda_k \in \mathbb{R}} (\lambda_k I_N - \hat{\mathcal{L}})^{-1} \prod_{\lambda_\ell^\pm = \eta_l \pm \imath \beta_l} \left\{ (\eta_l I_N - \hat{\mathcal{L}})^2 + \beta_l^2 I_N \right\}^{-1} (b^\top A^{-1} \otimes I_N)\, \mathrm{adj}(\mathcal{M})(I_s \otimes M^{-1}) g.$$

- $I_s \otimes M^{-1}$: s applications of M^{-1}.
- $(b^\top A^{-1} \otimes I_N)\, \mathrm{adj}(\mathcal{M})$: $(s-1)^2$ or $(s-1)s$ applications of M^{-1} depending on b.
- $\left\{ (\eta_l I_N - \hat{\mathcal{L}})^2 + \beta_l^2 I_N \right\}^{-1}$: Multiply by M, solve systems that look like

  $$((\eta M - \Delta t \mathcal{L}) M^{-1} (\eta M - \Delta t \mathcal{L}) + \beta^2 M) x = Mb,$$

  which we left precondition with

  $$(\gamma M - \Delta t \mathcal{L})^{-1} M (\gamma M - \Delta t \mathcal{L})^{-1}, \qquad \gamma = \sqrt{\eta^2 + \beta^2},$$

  using inner iterations to compute backward Euler steps.
- $(\lambda_k I_N - \hat{\mathcal{L}})^{-1}$: Backward Euler solves for each real eigenvalue of A, typically 1.