1 Introduction

1.1 Fully implicit Runge-Kutta

Consider the method-of-lines approach to solving partial differential equations (PDEs), where we discretize in space and arrive at a system of ODEs in time,

$$M\mathbf{u}'(t) = \mathcal{N}(\mathbf{u}, t) \quad \text{in } (0, T], \quad \mathbf{u}(0) = \mathbf{u}_0, \tag{1}$$ {{eq:problem}}

where M is a mass matrix, and $\mathcal{N} \in \mathbb{R}^{N\times N}$ a discrete, time-dependent, nonlinear operator depending on t and $\mathbf{u}$ (including potential forcing terms). Then consider time propagation using an s-stage Runge-Kutta scheme, characterized by the Butcher tableaux

$$\begin{array}{c|c} \mathbf{c}_0 & A_0 \\ \hline & \mathbf{b}_0^T \end{array},$$

with Runge-Kutta matrix $A_0 = (a_{ij})$, weight vector $\mathbf{b}_0^T = (b_1, \ldots, b_s)^T$, and nodes $\mathbf{c}_0 = (c_0, \ldots, c_s)$.

Runge-Kutta methods update the solution using a sum over stage vectors,

$$\begin{aligned} \mathbf{u}_{n+1} &= \mathbf{u}_n + \delta t \sum_{i=1}^{s} b_i \mathbf{k}_i, \\ M\mathbf{k}_i &= \mathcal{N}\left(\mathbf{u}_n + \delta t \sum_{i=1}^{s} b_i \mathbf{k}_i, t_n + \delta t c_i\right). \end{aligned}$$

For nonlinear PDEs, $\mathcal{N}$ is linearized using, for example, a Newton or a Picard linearization of the underlying PDE. Let us denote this linearization $\mathcal{L} \in \mathbb{R}^{N\times N}$ (or, in the case of a linear PDE, let $\mathcal{L} := \mathcal{N}$). Expanding, solving for the stages $\mathbf{k}$ as each step in a nonlinear iteration, or as the update to $\mathbf{u}$ for a linear PDE, can then be expressed as a block linear system,

$$\left(\begin{bmatrix} M & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & M \end{bmatrix} - \delta t \begin{bmatrix} a_{11}\mathcal{L}_1 & \ldots & a_{1s}\mathcal{L}_1 \\ \vdots & \ddots & \vdots \\ a_{s1}\mathcal{L}_s & \ldots & a_{ss}\mathcal{L}_s \end{bmatrix}\right) \begin{bmatrix} \mathbf{k}_1 \\ \vdots \\ \mathbf{k}_s \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_s \end{bmatrix}. \tag{2}$$ {{eq:k0}}

2 Inverting the IRK system

The RK stage system in (2) can be reformulated as [TODO: cite Will]

$$\left(A_0^{-1} \otimes M - \delta t \begin{bmatrix} \mathcal{L}_1 & & \\ & \ddots & \\ & & \mathcal{L}_s \end{bmatrix}\right)(A_0 \otimes I) \begin{bmatrix} \mathbf{k}_1 \\ \vdots \\ \mathbf{k}_s \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_s \end{bmatrix}.$$

For ease of notation, let us scale both sides of the system by a block-diagonal mass matrix and, excusing the slight abuse of notation, let $\mathcal{L}_i \mapsto \delta t M^{-1}\mathcal{L}_i$, $i = 1, .., s$. Note the time step is now included in $\mathcal{L}_i$. Because $\mathcal{L}_i$ is time-dependent, it is possible that δt is also time-dependent. Now let α_{ij} denote the ij-element of A_0^{-1} (assuming A_0 is invertible). Then, solving (2) can be effectively reduced to inverting the operator

$$\mathcal{M}_s := A_0^{-1} \otimes I - \begin{bmatrix} \mathcal{L}_1 & & \\ & \ddots & \\ & & \mathcal{L}_s \end{bmatrix} = \begin{bmatrix} \alpha_{11}I - \mathcal{L}_1 & \alpha_{12}I & \ldots & \alpha_{1s}I \\ \alpha_{21}I & \alpha_{22}I - \mathcal{L}_2 & & \alpha_{2s}I \\ \ddots & & \ddots & \vdots \\ \alpha_{s1}I & \ldots & \alpha_{s(s-1)}I & \alpha_{ss}I - \mathcal{L}_s \end{bmatrix}. \tag{3}$$ {{eq:k1}}

Note, there are a number of methods with one explicit stage preceded or followed by several fully implicit and coupled stages. In such cases, A_0 is not invertible, but the explicit stage can be eliminated from the system. The remaining operator can then be reformulated as above, and the inverse that must be applied takes the form of (3) but based on a principle submatrix of A_0.

Returning to the time propagation, the Runge-Kutta update can be written as

$$\begin{aligned}
\mathbf{u}_{n+1} &= \mathbf{u}_n + \delta t \sum_{i=1}^{s} b_i \mathbf{k}_i \\
&= \mathbf{u}_n + \delta t (\mathbf{b}_0^T A_0^{-1} \otimes I) \mathcal{M}_s^{-1} \mathbf{f}. \qquad (4) \quad \{\{\texttt{eq:update}\}\}
\end{aligned}$$

In the case of stiffly accurate RK schemes, $\mathbf{b}_0^T A_0 = [0, ..., 0, 1]$, and (4) takes the simplified form

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \delta t \begin{bmatrix} 0 & ... & I \end{bmatrix} \mathcal{M}_s^{-1} \mathbf{f}.$$

3 The time-dependent case

3.1 LU/Gauss decomposition

See Theorem 4.4 in "A Theory of Noncommutative Determinants and Characteristic Functions of Graphs." This proves the standard LU/Gauss decomposition for a matrix over a noncommutative division ring. It is based on quasideterminants. That paper gives a UL decomposition, but Theorem 4.9.7 in arxiv version of QUASIDETERMINANTS gives the LU decomposition. In practice I think we would want the LU, not UL, particularly for stiffly accurate because we don't need to compute the whole U^{-1}.

Observe that the block matrix $\mathcal{M}_s$ in (3) is a matrix defined over a noncommutative ring R. Let us further assume that R is a division ring, that is, every element in R has a multiplicative inverse. Note that it is easy to construct an example where R is not a division ring: simply let α_{12} be an eigenvalue of $(\alpha_{11} I - \mathcal{L})$, in which case $(\alpha_{11} I - \mathcal{L}) - \alpha_{12} I$ is singular and does not have a multiplicative inverse. However, this assumption is a technical tool and I believe RK coefficients make it okay in practice?

Moving forward, in matrix form a constant α_{ij} is typically implied to be $\alpha_{ij} I$, but the I is removed for ease of notation.

3.1.1 2x2

$$L = \begin{bmatrix} 1 & 0 \\ \alpha_{21}(\alpha_{11} - \mathcal{L}_1)^{-1} & I \end{bmatrix}, \qquad U = \begin{bmatrix} \alpha_{11} - \mathcal{L}_1 & \alpha_{12} \\ 0 & \alpha_{22} - \mathcal{L}_2 - \alpha_{12}\alpha_{21}(\alpha_{11} - \mathcal{L}_1)^{-1} \end{bmatrix}$$

3.1.2 3x3

$$\begin{aligned}
L &= \begin{bmatrix} I & 0 & 0 \\ \alpha_{31}(\alpha_{11} - \mathcal{L}_1)^{-1} & I & 0 \\ \alpha_{21}(\alpha_{11} - \mathcal{L}_1)^{-1} & (\alpha_{22} - \mathcal{L}_2 - \alpha_{12}\alpha_{21}(\alpha_{11} - \mathcal{L}_1)^{-1})(\alpha_{32} - \alpha_{12}\alpha_{31}(\alpha_{11} - \mathcal{L}_1)^{-1})^{-1} & I \end{bmatrix}, \\
U &= \begin{bmatrix} \alpha_{11} - \mathcal{L}_1 & \alpha_{12} & \alpha_{13} \\ 0 & \alpha_{32} - \alpha_{12}\alpha_{31}(\alpha_{11} - \mathcal{L}_1)^{-1} & \alpha_{33} - \mathcal{L}_3 - \alpha_{13}\alpha_{31}(\alpha_{11} - \mathcal{L}_1)^{-1} \\ 0 & 0 & Q \end{bmatrix} \\
Q &= \alpha_{23} - \alpha_{13}\alpha_{21}(\alpha_{11} - \mathcal{L}_1)^{-1} - (\alpha_{22} - \mathcal{L}_2 - \alpha_{12}\alpha_{21}(\alpha_{11} - \mathcal{L}_1)^{-1}) \\
&\quad (\alpha_{33} - \mathcal{L}_3 - \alpha_{13}\alpha_{31}(\alpha_{11} - \mathcal{L}_1)^{-1})(\alpha_{32} - \alpha_{12}\alpha_{31}(\alpha_{11} - \mathcal{L}_1)^{-1})^{-1} \\
&= \alpha_{23} - \alpha_{13}\alpha_{21}(\alpha_{11} - \mathcal{L}_1)^{-1} - (\alpha_{11} - \mathcal{L}_1)^{-1}((\alpha_{11} - \mathcal{L}_1)(\alpha_{22} - \mathcal{L}_2) - \alpha_{12}\alpha_{21}) \\
&\quad ((\alpha_{33} - \mathcal{L}_3)(\alpha_{11} - \mathcal{L}_1) - \alpha_{13}\alpha_{31})(\alpha_{32}(\alpha_{11} - \mathcal{L}_1) - \alpha_{12}\alpha_{31})^{-1} \\
&= (\alpha_{11} - \mathcal{L}_1)^{-1} \Bigg(\Big[\alpha_{23}(\alpha_{11} - \mathcal{L}_1) - \alpha_{13}\alpha_{21} \Big] \Big[\alpha_{32}(\alpha_{11} - \mathcal{L}_1) - \alpha_{12}\alpha_{31} \Big] - \\
&\quad \Big[(\alpha_{11} - \mathcal{L}_1)(\alpha_{22} - \mathcal{L}_2) - \alpha_{12}\alpha_{21} \Big] \Big[(\alpha_{33} - \mathcal{L}_3)(\alpha_{11} - \mathcal{L}_1) - \alpha_{13}\alpha_{31} \Big] \Bigg) \Big[\alpha_{32}(\alpha_{11} - \mathcal{L}_1) - \alpha_{12}\alpha_{31} \Big]^{-1}
\end{aligned}$$

Quasideterminant:

$$\Bigg[\alpha_{33} - \mathcal{L}_3 - \alpha_{32} \Bigg(\alpha_{13} \left(\alpha_{12} - (\alpha_{11} - \mathcal{L}_1) \frac{(\alpha_{22} - \mathcal{L}_2)}{\alpha_{21}} \right)^{-1} - \frac{\left(\alpha_{23} \left(\alpha_{12} - (\alpha_{11} - \mathcal{L}_1) \frac{(\alpha_{22} - \mathcal{L}_2)}{\alpha_{21}} \right)^{-1} (\alpha_{11} - \mathcal{L}_1) \right)}{\alpha_{21}} \Bigg) - \alpha_{31} \Bigg(- \Bigg(\frac{\Big(\alpha_{13}(\alpha_{22} - \mathcal{L}_2)(\alpha_{12} - (\alpha_{11} - \mathcal{L}_1)(\alpha_{22} - \mathcal{L}_2)/\alpha}{\alpha_{21}}$$

3.1.3 4x4

3.2 Direct inverse?

Consider

$$\mathcal{M}_3 := \begin{bmatrix} (\alpha_{11}I + \mathcal{L}) & \alpha_{12}I & \alpha_{13}I \\ \alpha_{21}I & (\alpha_{22}I + \mathcal{L}) & \alpha_{23}I \\ \alpha_{31}I & \alpha_{32}I & (\alpha_{33}I + \mathcal{L}) \end{bmatrix}. \tag{5}$$ {{eq:Mnt}}

For stiffly accurate problems, we only need the last row of $\mathcal{M}_3^{-1}$. Using the Noncommutative algebra package in mathematica, we can work this out as

$$\begin{aligned} P &:= \frac{1}{\alpha_{21}}\Big(\alpha_{13}\alpha_{21} - \alpha_{23}(\alpha_{11} - \mathcal{L}_1) + \Big[\alpha_{12}\alpha_{21} - (\alpha_{11} - \mathcal{L}_1)(\alpha_{22} - \mathcal{L}_2)\Big] \\ &\qquad \Big[\alpha_{32}\alpha_{21} - \alpha_{31}(\alpha_{22} - \mathcal{L}_2)\Big]^{-1}\Big[\alpha_{23}\alpha_{31} - \alpha_{21}(\alpha_{33} - \mathcal{L}_3)\Big]\Big), \\ r_1 &= P^{-1} \\ r_2 &= -\frac{1}{\alpha_{21}}P^{-1}\left((\alpha_{11} - \mathcal{L}_1) - \alpha_{31}\Big[\alpha_{12}\alpha_{21} - (\alpha_{11} - \mathcal{L}_1)(\alpha_{22} - \mathcal{L}_2)\Big]\Big[\alpha_{32}\alpha_{21} - \alpha_{31}(\alpha_{22} - \mathcal{L}_2)\Big]^{-1}\right), \\ r_3 &= -P^{-1}\Big[\alpha_{12}\alpha_{21} - (\alpha_{11} - \mathcal{L}_1)(\alpha_{22} - \mathcal{L}_2)\Big]\Big[\alpha_{32}\alpha_{21} - \alpha_{31}(\alpha_{22} - \mathcal{L}_2)\Big]^{-1} \end{aligned}$$

For purposes of preconditioning, let us assume commutation of $\mathcal{L}_1$ and $\mathcal{L}_2$ or $\mathcal{L}_2$ and $\mathcal{L}_3$ and factor out the inverse term in P. This results in the modified operator

$$\begin{aligned} \widehat{P} := \frac{1}{\alpha_{21}}\Big(&\Big[\alpha_{13}\alpha_{21} - \alpha_{23}(\alpha_{11} - \mathcal{L}_1)\Big]\Big[\alpha_{32}\alpha_{21} - \alpha_{31}(\alpha_{22} - \mathcal{L}_2)\Big] + \\ &\Big[\alpha_{12}\alpha_{21} - (\alpha_{11} - \mathcal{L}_1)(\alpha_{22} - \mathcal{L}_2)\Big]\Big[\alpha_{23}\alpha_{31} - \alpha_{21}(\alpha_{33} - \mathcal{L}_3)\Big]\Big) \end{aligned}$$

Interestingly, I think $\widehat{P}$ is exactly the determinant in (10).. Upside of that is maybe using the determinant/adjugate form would avoid having to apply the action of P (which includes an inverse) every iteration. Downside is it seems like that doesn't take advantage of the fact that we only need $[0, ..., 0, 1]\mathcal{M}_s^{-1}$.

3.3 Two stages

Consider the case of two fully implicit stages. Then we need to invert a block linear system $\mathcal{M}_2\mathbf{s} = \mathbf{r}$, given by

$$\begin{bmatrix} \alpha_{11}I - \mathcal{L}_1 & \alpha_{12}I \\ \alpha_{21}I & \alpha_{22}I - \mathcal{L}_2 \end{bmatrix}\begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{bmatrix}. \tag{6}$$ {{eq:Mnt}}

In the context of 2×2 block operators, the key to inverting the matrix is inverting one of the Schur complements. Define the matrix polynomials

$$\begin{aligned} P_{\mathcal{L}} &:= (\alpha_{11}I - \mathcal{L}_1)(\alpha_{22}I - \mathcal{L}_2) - \alpha_{12}\alpha_{21}I, \\ Q_{\mathcal{L}} &:= (\alpha_{22}I - \mathcal{L}_2)(\alpha_{11}I - \mathcal{L}_1) - \alpha_{12}\alpha_{21}I, \end{aligned}$$

and consider both Schur complements,

$$\begin{aligned} S_{22} &= \alpha_{22}I - \mathcal{L}_2 - \alpha_{12}\alpha_{21}(\alpha_{11}I - \mathcal{L})^{-1} \\ &= \left[(\alpha_{22}I - \mathcal{L}_2)(\alpha_{11}I - \mathcal{L}_1) - \alpha_{12}\alpha_{21}I\right](\alpha_{11}I - \mathcal{L}_1)^{-1} \\ &= Q_{\mathcal{L}}(\alpha_{11}I - \mathcal{L}_1)^{-1} \\ &= (\alpha_{11}I - \mathcal{L}_1)^{-1}P_{\mathcal{L}} \\ S_{11} &= P_{\mathcal{L}}(\alpha_{22}I - \mathcal{L}_2)^{-1} \\ &= (\alpha_{22}I - \mathcal{L}_2)^{-1}Q_{\mathcal{L}}. \end{aligned}$$

Appealing to the closed form inverse of 2×2 block matrices derived from a block LDU decomposition, we can then write $\mathcal{M}_2^{-1}$ in terms of the Schur complements:

$$\begin{aligned} \begin{bmatrix} \alpha_{11}I - \mathcal{L}_1 & \alpha_{12}I \\ \alpha_{21}I & \alpha_{22}I - \mathcal{L}_2 \end{bmatrix}^{-1} &= \begin{bmatrix} (\alpha_{22}I - \mathcal{L}_2)P_{\mathcal{L}}^{-1} & -\alpha_{12}Q_{\mathcal{L}}^{-1} \\ -\alpha_{21}P_{\mathcal{L}}^{-1} & (\alpha_{11}I - \mathcal{L}_1)Q_{\mathcal{L}}^{-1} \end{bmatrix} \\ &= \begin{bmatrix} \alpha_{22}I - \mathcal{L}_2 & -\alpha_{12}I \\ -\alpha_{21}I & \alpha_{11}I - \mathcal{L}_1 \end{bmatrix} \begin{bmatrix} P_{\mathcal{L}}^{-1} & \mathbf{0} \\ \mathbf{0} & Q_{\mathcal{L}}^{-1} \end{bmatrix}. \end{aligned} \tag{7}$$ {{eq:Minv}}

Note, this more or less exactly takes the form of a scalar 2×2 matrix inverse, replacing the $1/det$ with the right scaling by the polynomial inverses.. We can also restructure to pull polynomials out of the left

3.3.1 Two stages

Returning to the case of two stages, by assuming $\mathcal{L}_1 = \mathcal{L}_2$ we have $P_{\mathcal{L}} = Q_{\mathcal{L}} := P_2(\mathcal{L})$. Then, $\mathcal{M}_2^{-1}$ (7) can be applied through two applications of $P_2(\mathcal{L})^{-1}$, along with some additional mat-vecs and vector addition. Note that $P_2(\mathcal{L})$ is a quadratic polynomial in $\mathcal{L}$, which can be solved in two steps by factoring (**??**) and applying the successive inverses $(\lambda_1 I - \mathcal{L})^{-1}(\lambda_2 I - \mathcal{L})^{-1}$, where λ_1, λ_2 are the eigenvalues of $\boldsymbol{\alpha} = A_0^{-1}$. Note, in the 2×2 case these have a relatively simple algebraic structure,

$$\lambda_1, \lambda_2 = \frac{1}{2}\left(\alpha_{11} + \alpha_{22} \pm \sqrt{(\alpha_{11} + \alpha_{22})^2 - 4(\alpha_{11}\alpha_{22} - \alpha_{12}\alpha_{21})}\right).$$

3.3.2 Three stages

For nonsingular scalar 3×3 matrix A, the inverse is given by

$$A^{-1} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}^{-1} = \det(A)^{-1} \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}, \tag{8}$$ {{eq:3inv}}

with elements defined by

$$\begin{aligned} A_{11} &= \ \ (a_{22}a_{33} - a_{23}a_{32}), & A_{12} &= -(a_{12}a_{33} - a_{13}a_{32}), & A_{13} &= \ \ (a_{12}a_{23} - a_{13}a_{22}), \\ A_{21} &= -(a_{21}a_{33} - a_{23}a_{31}), & A_{22} &= \ \ (a_{11}a_{33} - a_{13}a_{31}), & A_{23} &= -(a_{11}a_{23} - a_{13}a_{21}), \\ A_{31} &= \ \ (a_{21}a_{32} - a_{22}a_{31}), & A_{32} &= -(a_{11}a_{32} - a_{12}a_{31}), & A_{33} &= \ \ (a_{11}a_{22} - a_{12}a_{21}), \end{aligned}$$

and determinant $\det(A) = a_{11}A_{11} + a_{12}A_{21} + a_{13}A_{31}$.

In our case, consider

$$\mathcal{M}_3 := \begin{bmatrix} (\alpha_{11}I + \mathcal{L}) & \alpha_{12}I & \alpha_{13}I \\ \alpha_{21}I & (\alpha_{22}I + \mathcal{L}) & \alpha_{23}I \\ \alpha_{31}I & \alpha_{32}I & (\alpha_{33}I + \mathcal{L}) \end{bmatrix}. \tag{9}$$ {{eq:Mnt}}

Define $\mathcal{N}_3$ as a block 3×3 matrix with entries of A^{-1} as in (8), excluding the determinant. Plugging in, we have entries of $\mathcal{N}_3$ given by

$$\begin{aligned} A_{11} &= (\alpha_{22}I - \mathcal{L})(\alpha_{33}I - \mathcal{L}) - \alpha_{23}\alpha_{32}I, \\ A_{12} &= -\alpha_{12}(\alpha_{33}I - \mathcal{L}) + \alpha_{13}\alpha_{32}I, \\ A_{13} &= \alpha_{12}\alpha_{23}I - \alpha_{13}(\alpha_{22}I - \mathcal{L}), \\ A_{21} &= -\alpha_{21}(\alpha_{33}I - \mathcal{L}) + \alpha_{23}\alpha_{31}I, \\ A_{22} &= (\alpha_{11}I - \mathcal{L})(\alpha_{33}I - \mathcal{L}) - \alpha_{13}\alpha_{31}I, \\ A_{23} &= -\alpha_{23}(\alpha_{11}I - \mathcal{L}) + \alpha_{13}\alpha_{21}I, \\ A_{31} &= \alpha_{21}\alpha_{32}I - \alpha_{31}(\alpha_{22}I - \mathcal{L}), \\ A_{32} &= -\alpha_{32}(\alpha_{11}I - \mathcal{L}) + \alpha_{12}\alpha_{31}I, \\ A_{33} &= (\alpha_{11}I - \mathcal{L})(\alpha_{22}I - \mathcal{L}) - \alpha_{12}\alpha_{21}I. \end{aligned}$$

Working through the details, it is striaghtforward to confirm that $\mathcal{N}_3\mathcal{M}_3$ is a block-diagonal matrix, with diagonal blocks given by the (block) determinant of $\mathcal{M}_3$,

$$\begin{aligned} D = (\alpha_{11}I - \mathcal{L})(\alpha_{22}I - \mathcal{L})(\alpha_{33}I - \mathcal{L}) - \alpha_{23}\alpha_{32}(\alpha_{11}I - \mathcal{L}) - \\ \alpha_{13}\alpha_{31}(\alpha_{22}I - \mathcal{L}) - \alpha_{12}\alpha_{21}(\alpha_{33}I - \mathcal{L}) + (\alpha_{13}\alpha_{32}\alpha_{21} + \alpha_{12}\alpha_{23}\alpha_{31})I. \end{aligned} \tag{10}$$ {{eq:det}}

Similar to the 2×2 case (albeit algebraically more complicated), this is a cubic polynomial in $\mathcal{L}$. By computing the roots of this polynomial, we can construct error propagation of a three-stage fixed-point iteration that produces the exact inverse of $\mathcal{D}$.

A few more thoughts:

- Working through the algebra, the cancellation does not fully happen if $\mathcal{L}_i$ is time-dependent. However, a lot of it does. There will be some off-diagonal terms that take the form, for example,

$$(\alpha_{11}I - \mathcal{L}_1)(\alpha_{22}I - \mathcal{L}_2) - (\alpha_{22}I - \mathcal{L}_2)(\alpha_{11}I - \mathcal{L}_1) = \mathcal{L}_1\mathcal{L}_2 - \mathcal{L}_2\mathcal{L}_1.$$

  This provides a nice theoretical tool to analyze what is going on. It is possible these are often quite small. Moreover, we may be able to choose an ordering where these terms only occur on, say, the strictly lower triangular part, in which case a block-triangular preconditioning would also be exact.

4 Parallel in time

- For pAIR or MGRiT, can we do a p-multigrid approach, where we solve order p MGRIT via a set of p MGRiT/pAIR solves for order 1?
- Another (simpler) possibility is use global Krylov only applied to the solution vector, and precondition a high-order time integrator with pAIR built on a low-order time integrator. This could also be done as a two-grid method, where we do high-order stage integration as a relaxation coupled with a pAIR solve with a low-order time integrator. This could couple well if I come up with a good way to do fully implicit IRK solves, we could use that on the finest grid. Or, we could just use full implicit IRK as the operator, relax w/ high order SDIRK, then coarse-grid correct with pAIR on low-order SDIRK (is this SDC or PFAST?)

4.1 Precondition HO with LO

Let Φ denote a single step of our high-order (fine-grid) Runge-Kutta operator and Ψ a single step of our low-order (coarse-grid) operator. Then error propagation for preconditioning a global time-stepping scheme base don Φ with a low-order scheme based on Ψ is given by

$$I - B_\Delta^{-1}A = \begin{bmatrix} \mathbf{0} & & & & \\ I & \mathbf{0} & & & \\ \Psi & I & \mathbf{0} & & \\ \vdots & \vdots & \ddots & \ddots & \\ \Psi^{N_c-2} & \Psi^{N_c-3} & \dots & I & \mathbf{0} \end{bmatrix} \mathrm{diag}(\Psi - \Phi).$$

Here we have assumed Φ and Ψ to be linear and independent of time, but that is not necessary. The (linearized) time-dependent setting has the same form as above, but with products of Ψ evaluated at successive (discrete) times.

Note this is analogous to two-level convergence of MGRiT. Necessary and sufficient conditions for convergence are given by the temporal approximation property,

Definition 1 (Temporal approximation property)**.** *Let Φ denote a fine-grid time-stepping operator and Ψ denote a coarse-grid time-stepping operator, for all time points, with coarsening factor k. Then, Φ satisfies an F-relaxation temporal approximation property with power p (F-TAP$_p$), with respect to Ψ, with constant $\varphi_{F,p}$, if, for all vectors $\mathbf{v}$,*

$$\|(\Psi - \Phi^k)^p\mathbf{v}\| \leq \varphi_{F,p}\left[\min_{x\in[0,2\pi]} \left\|(I - e^{\mathrm{i}x}\Psi)^p\mathbf{v}\right\|\right]. \quad (11) \quad \{\{eq:tap\_f\}\}$$

Similarly, Φ satisfies an FCF-relaxation temporal approximation property with power p (FCF-TAP$_p$), with respect to Ψ, with constant $\varphi_{FCF,p}$, if, for all vectors $\mathbf{v}$,

$$\|(\Psi - \Phi^k)^p\mathbf{v}\| \leq \varphi_{FCF,p}\left[\min_{x\in[0,2\pi]} \left\|(\Phi^{-k}(I - e^{\mathrm{i}x}\Psi))^p\mathbf{v}\right\|\right]. \quad (12) \quad \{\{eq:tap\_fcf\}\}$$

For the F-relaxation case, we have $k = 1$. The analogous FCF context here would presumably correspond to doing a complete fine-grid solve on processor. The downside is this is kind of expensive.

However, if we treat Φ as an explicit scheme, with the stages and implicit solves included, just the action of the operator will be expensive to compute. I think it will be important to somehow consider the RK scheme expanded, so that when we apply a coarse-grid preconditioner, computing the action of the operator can be done explicitly and doesn't require computing a bunch of inverses...

L-stability appears to be important here, and is not perfect. But, with just F-relaxation, then we don't even need to solve the fine grid implicit discretization. Need to check how this could work with fully implicit fine grid discretization, preconditioned with BE.. Would be cool if that could be effective.

4.2 A multigrid or block preconditioning perspective

Let A and $\mathbf{b}$ correspond to Butcher tableaux,

$$\implies \qquad \mathbf{u}_{i+1} = \mathbf{u}_i + \delta t \sum_{i=1}^{s} b_i \mathbf{k}_i,$$

and let $\hat{\mathbf{u}} = [\mathbf{u}_0, \mathbf{u}_1, ...]$ denote the space-time solution at times $t_0, t_1, ...$ Then the full space-time linear system takes the form $A\hat{\mathbf{u}} = \hat{\mathbf{g}}$, where

$$A = \begin{array}{c} \begin{array}{cccccc} \underline{t_0} & \text{(stages)} & \underline{t_1} & \text{(stages)} & \underline{t_2} & \cdots \end{array} \\ \begin{pmatrix} I & & & & & \\ -\delta t(\mathbf{1}_n \otimes \mathcal{L}) & I - \delta t A \otimes \mathcal{L} & & & & \\ -I & -\mathbf{b} \otimes I & I & & & \\ & & -\delta t(\mathbf{1}_n \otimes \mathcal{L}) & I - \delta t A \otimes \mathcal{L} & & \\ & & -I & -\mathbf{b} \otimes I & I & \\ & & & \ddots & \ddots & \ddots \end{pmatrix} \end{array}.$$

Note, for ease of notation here, we have assumed that $\mathcal{L}$ is independent of time and the stage matrix can be written as a Kronecker product. However, for time-dependent problems, the space-time matrix takes a similar structure, but the stage-matrix must be written explicitly as a block $s \times s$ matrix. Now, let us denote all intermediate RK stages as F-points and time-points as C-points. Eliminating the F-points in a Schur complement sense yields the reduced MGRiT-style system,

$$A = \begin{array}{c} \begin{array}{ccccc} \underline{t_0} & \underline{t_1} & \cdots & \cdots & \underline{t_N} \end{array} \\ \begin{pmatrix} I & & & & \\ -\Phi & I & & & \\ & -\Phi & I & & \\ & & \ddots & \ddots & \\ & & & -\Phi & I \end{pmatrix} \end{array},$$

where

$$\Phi = I + \delta t \mathbf{b}_0^T \otimes I \left(I - \delta t A_0 \otimes \mathcal{L}\right)^{-1} (\mathbf{1}_n \otimes \mathcal{L}). \tag{13}$$ {{eq:mgrit}}

As previously, this takes a more complicated but similar form if $\mathcal{L}$ is time-dependent.

I see two nice ways to think about this:

1. The simplest is a block-preconditioning approach. Reordering the matrix by stages and then time-points, We will use a block lower-triangular preconditioner. If we exactly invert the F-block (solve all RK stages exactly) followed by inverting the Schur complement, we converge in two iterations. Of course in this case, that is just sequential time stepping. But, we could approximate our Schur complement with a space-time solve using a low-order integrator (BE?) and couple this with an F-relaxation (some inexact solution to the stage matrices). Note, theory and practice have generally indicated that if one solve is not exact, the other also does not need to be, so we probably shouldn't solve the stages exactly. This makes me think we might be able to let the true RK scheme be a high-order fully implicit RK scheme, and precondition this with an SDIRK scheme on stage integration.

2. Above can also be thought of in some sense as a non-Galerkin multigrid method. But, we developed a decent two-grid theoretical framework for AIR. I wonder if we can construct R and P so that approximating high-order Φ on the coarse-grid with backward Euler is actually a Petrov-Galerkin multigrid method, that is, RAP directly leads to a matrix analogous to (13) but with $\Phi \sim$ backward Euler (or some other one-stage method)?