Parallel-in-Time Solver for the All-at-Once Runge–Kutta Discretization

Luca Bergamaschi
Department of Civil Environmental and Architectural Engineering
University of Padua
e-mail:luca.bergamaschi@unipd.it, webpage: www.dmsa.unipd.it/~berga

Joint work with
Santolo Leveque, Scuola Normale Superiore, Pisa,
Ángeles Martinez, Dept. of Maths, Informatics & Geosciences, University of Trieste,
John Pearson, School of Maths, University of Edinburgh.

Exploiting Algebraic and Geometric Structure in Time-Integration Methods
April 3–5, 2024 Pisa

Outline

- All-at-once approach for the Heat Equation with high order implicit Runge-Kutta schemes.
- Preconditioner for the resulting 2×2 block linear system.
- Optimality of the preconditioner. Computational issues. Parallelizability.

Outline

- All-at-once approach for the Heat Equation with high order implicit Runge-Kutta schemes.
- Preconditioner for the resulting 2×2 block linear system.
- Optimality of the preconditioner. Computational issues. Parallelizability.

- All-at-once approach for the Stokes equation.
- Block preconditioner

Outline

- All-at-once approach for the Heat Equation with high order implicit Runge-Kutta schemes.
- Preconditioner for the resulting 2×2 block linear system.
- Optimality of the preconditioner. Computational issues. Parallelizability.

- All-at-once approach for the Stokes equation.
- Block preconditioner

- Numerical results, sequential

- Numerical results, parallel (Up to more than 1000 processors)

- Conclusions

Heat equation

$$\begin{cases} \dfrac{\partial v}{\partial t} - \nabla^2 v = f(\mathrm{x}, t) & \text{in } \Omega \times (0, t_f), \\ v(\mathrm{x}, t) = g(\mathrm{x}, t) & \text{on } \partial\Omega \times (0, t_f), \\ v(\mathrm{x}, 0) = v_0(\mathrm{x}) & \text{in } \Omega, \end{cases}$$

Finite Elements in space + Runge-Kutta time discretizations

$$M\mathrm{v}_{n+1} = M\mathrm{v}_n + \tau M \sum_{i=1}^{s} b_i \mathrm{k}_{in} \quad n = 0, \ldots, n_t - 1,$$

The stages k_{in} are defined as follows:

$$M\mathrm{k}_{in} + K\mathrm{v}_n + \tau K \textstyle\sum_{j=1}^{s} a_{ij}\mathrm{k}_{jn} = \mathrm{f}_{in}, \qquad i = 1, \ldots, s, \ n = 0, \ldots, n_t - 1,$$

where

$$(\mathrm{f}_{in})_m = \int_\Omega f(\mathrm{x}, t_n + c_i\tau)\phi_m \, \mathrm{d}\Omega, \quad i = 1, \ldots, s.$$

K and M are the *stiffness* and *mass* matrix respectively.

Coefficients $a_{ij} \to A_{\mathrm{RK}}, \quad b_i \to b_{\mathrm{RK}}$ form the **Butcher tableau**.

All-at-once matrix formulation

$$\underbrace{\begin{bmatrix} \Phi & \Psi_1 \\ \Psi_2 & \Theta_H \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \mathbf{v} \\ \mathbf{k} \end{bmatrix} = \begin{bmatrix} \mathsf{r}_1 \\ \mathsf{r}_2 \end{bmatrix},$$

where

$$\Phi = \begin{bmatrix} M & & & \\ -M & \ddots & & \\ & \ddots & \ddots & \\ & & -M & M \end{bmatrix} \qquad \Psi_1 = -\begin{bmatrix} 0 & & \\ \tau \mathsf{b}_{\mathrm{RK}}^\top \otimes M & & \\ & \ddots & \\ & & \tau \mathsf{b}_{\mathrm{RK}}^\top \otimes M \end{bmatrix}$$

$$\Psi_2 = \begin{bmatrix} \mathsf{e} \otimes K & & \\ & \ddots & \\ & & \mathsf{e} \otimes K \quad 0 \end{bmatrix}$$

$$\begin{aligned} \Theta_H &= I_{n_t} \otimes \hat{\Theta}_H, \\ \hat{\Theta}_H &= I_s \otimes M + \tau A_{\mathrm{RK}} \otimes K \end{aligned}$$

Preconditioner

We consider as a preconditioner for $\mathbf{A} = \begin{bmatrix} \Phi & \Psi_1 \\ \Psi_2 & \Theta_H \end{bmatrix}$

the matrix $\mathbf{P} = \begin{bmatrix} S & \Psi_1 \\ 0 & \Theta_H \end{bmatrix}$,

where $S = \Phi - \Psi_1 \Theta_H^{-1} \Psi_2$ is the Schur complement.

Specifically we have

$$S = \begin{bmatrix} \mathsf{M} & & \\ & \ddots & \\ & & \mathsf{M} \end{bmatrix} \underbrace{\begin{bmatrix} I_{n_x} & & & \\ -I_{n_x} + \widehat{X} & \ddots & & \\ & \ddots & \ddots & \\ & & -I_{n_x} + \widehat{X} & I_{n_x} \end{bmatrix}}_{\widehat{S}},$$

where

$$\widehat{X} \quad = \quad \tau \begin{bmatrix} b_1 I_{n_x} & \dots & b_s I_{n_x} \end{bmatrix} \underbrace{[I_s \otimes \mathsf{M} + \tau A_{\mathrm{RK}} \otimes \mathsf{K}]}_{\widehat{\Theta}_H}^{-1} [\mathsf{e} \otimes \mathsf{K}] .$$

Practical approximation of $\mathcal{P}$

P is optimal.

In fact, supposing that both Θ_H and S are invertible,

1. $\lambda(\mathbf{P}^{-1}\mathbf{A}) = \{1\}$
2. the minimal polynomial of the $\mathbf{P}^{-1}\mathbf{A}$ has degree 2 (GMRES will converge in at most 2 iterations)

WARNING! We can not explicitly form the Schur complement S due to the large dimension of the system.

Practical approximation of **P**

- Application of block diagonal matrix Θ_H needs solution of n_t linear systems with matrix $I_s \otimes \mathrm{M} + \tau A_{\mathrm{RK}} \otimes \mathrm{K}$. Preconditioner needed.
- A parallel solve for $\widehat{S}$ is performed by MGRIT routine [Falgout *et al*, SISC 2014] employing the XBraid v3.0.0 routine.

Falgout, Friedhoff, Kolev, MacLachlan and Schröder

Parallel time integration with multigrid

SISC, 2014

Preconditioner for the stages

We need an optimal preconditioner for the system of the stages

$$\underbrace{(I_s \otimes \mathsf{M} + \tau A_{\mathrm{RK}} \otimes \mathsf{K})}_{\widehat{\Theta}_H} \mathbf{k} = \mathbf{b}$$

The idea is to compute a (real) SVD of A_{RK}: $A_{\mathrm{RK}} = U\Sigma V^\top$. Hence

$$\begin{aligned} I_s \otimes \mathsf{M} + \tau A_{\mathrm{RK}} \otimes \mathsf{K} &= I_s \otimes \mathsf{M} + \tau U\Sigma V^\top \otimes \mathsf{K} \\ &= (U \otimes I_{n_x})[(U^\top V) \otimes \mathsf{M} + \tau\Sigma \otimes \mathsf{K}](V^\top \otimes I_{n_x}). \end{aligned}$$

The eigenvalues of $U^\top V$ lie all on the unit circle centered at the origin of the complex plane, and its eigenvectors are mutually orthogonal.

Main idea. Since $|\lambda(U^T V)| = 1$, we approximate $U^\top V \approx I_s$ and propose the following preconditioner:

$$\mathbf{P}_{\mathrm{RK}} := [U \otimes I_{n_x}]\,[I_s \otimes \mathsf{M} + \tau\Sigma \otimes \mathsf{K}]\left[V^\top \otimes I_{n_x}\right] \approx \widehat{\Theta}_H.$$

Note that now the systems to be solved (with $I_s \otimes \mathsf{M} + \tau\Sigma \otimes \mathsf{K}$) are decoupled due to diagonal matrix Σ.

Preconditioner application

To (approximately) solve for $\mathbf{P} = \begin{bmatrix} S & \Psi_1 \\ 0 & \Theta_H \end{bmatrix}$, we employ the (inner) GMRES method to solve for $\widehat{\Theta}_H$ with preconditioner

$$\mathbf{P}_{\mathrm{RK}} = (U \otimes I_{n_x})(I_s \otimes M + \tau\Sigma \otimes K)(V^\top \otimes I_{n_x}).$$

Theorem

If the real part of Rayleigh quotient $\dfrac{x^*(U^\top V)x}{x^*x}$ *is positive, for any* $x \in \mathbb{C}^s \setminus \{0\}$

then the eigenvalues of the matrix $\mathrm{P}_{\mathrm{RK}}^{-1}\widehat{\Theta}_H$ *all lie in the right-half of the unit circle centered at the origin of the Gauss plane. If in addition* $1 \in \sigma(U^T V)$ *then* $1 \in \sigma(\mathrm{P}_{\mathrm{RK}}^{-1}\widehat{\Theta}_H)$ *with multiplicity* n_x.

- Assumption $\mathbf{Re}(\sigma(U^T V)) > 0$ holds for all RK methods we tried, up to order 7.

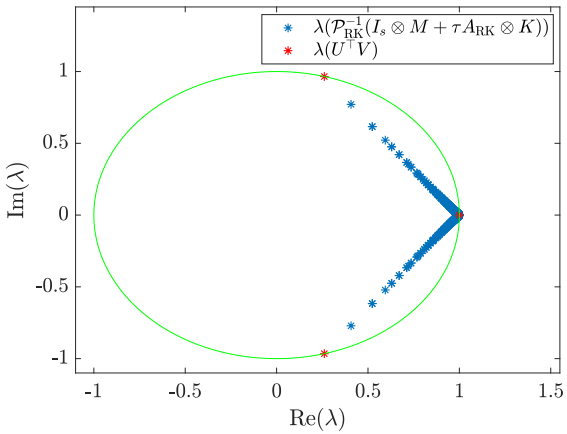
5-stage Radau method (order 9) works fine in practice but it has a pair of eigenvalues with real part equal to -5×10^{-4}.

- for odd s eigenvalue 1 always belongs to $\sigma(U^T V)$.

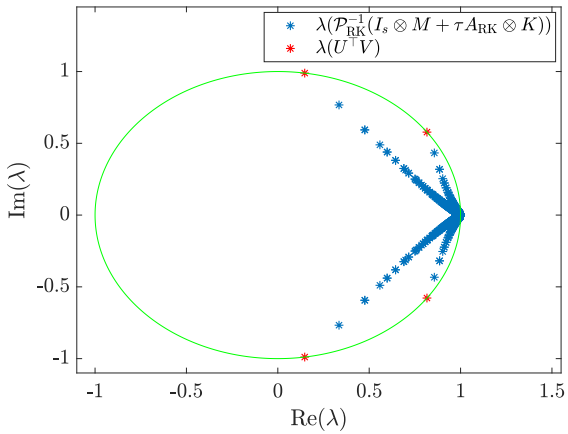
Eigenvalues of $\mathbf{P}_{\mathrm{RK}}^{-1}\widehat{\Theta}$

Q_2 elements, $\ell = 4$ with $h = 2^{-\ell}$, $\tau = 0.2$. Unit circle in green.

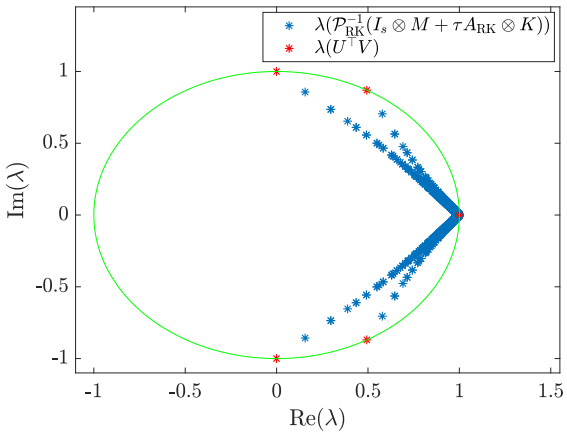
3-stages Gauss



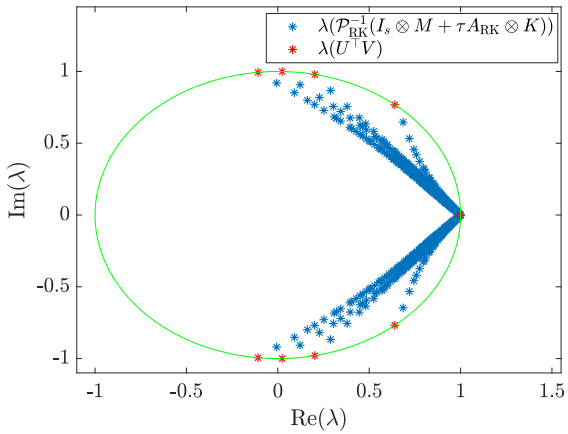
4-stages Lobatto IIIC



5-stages Radau IIA



9-stages Radau IIA



Properties of the SVD preconditioner

Even when the hypotheses of the Theorem are not satisfied: the preconditioner performs well:

GMRES residual after 5 and 10 inner iterations (Lobatto RK method) is weakly influenced by the number of stages s.

s	$\|r_5\|$	$\|r_{10}\|$
3	2.8228e-03	4.0014e-06
5	5.3229e-02	1.7263e-03
7	5.2125e-02	6.1990e-03
9	5.6382e-02	8.1128e-03
11	6.6290e-02	1.2204e-02
13	7.2945e-02	1.8589e-02
15	7.8344e-02	1.8545e-02

Construction of RK methods satisfying the hypotheses

We recall the W-transformation (Theorem 5.1, p. 71, Hairer & Wanner book, [HW]) to construct a stable implicit RK method of a given order. Given an integer $s \geq 2$,

let $P_s(x)$ a shifted and scaled Legendre polynomial. The nodes c_i are the roots of

$$\widehat{P}(x) = P_s(x) + \alpha_1 P_{s-1}(x) + \alpha_2 P_{s-2}(x).$$

The weights satisfy the usual (interpolatory) condition $B(s-2)$:

$$\sum_{i=1}^{s} b_i c_i^{q-1} = \frac{1}{q}, \text{ for } q = 1, \ldots, s-2.$$

Define matrix W as $W = (w_{ij})$, and $w_{ij} = P_{j-1}(c_i)$, $\varepsilon_k = \frac{1}{2\sqrt{4k^2-1}}$, and

$$X = \begin{bmatrix} 0.5 & -\varepsilon_1 & & \\ \varepsilon_1 & 0 & \ddots & \\ & \ddots & 0 & \beta_{s-1} \\ & & \varepsilon_{s-1} & \beta_s \end{bmatrix}.$$

Then $A_{\mathrm{RK}} = W^{-1} X W$ corresponds to an RK method of order $2s-2$.

Construction of RK methods satisfying the hypotheses

We are (almost) free to vary four parameters $\alpha_1, \alpha_2, \beta_{s-1}, \beta_s$ to **maximize the minimum of the real parts of the eigenvalues of** $U^\top V$.

Constraints on the parameters:

$$\begin{aligned} \alpha_2 &< \frac{s-1}{s}\frac{\sqrt{2s+1}}{2s-3} && \text{to have real } c_i\text{'s} \\ \beta_{s-1} < 0 \quad &\wedge \quad \beta_s \geq 0, && \text{for A-stability, see (5.45) of [HW].} \end{aligned}$$

Then run Matlab function `fminconc` for the constrained optimization problem

$$\overline{\mu} = \max_{\alpha_1, \alpha_2, \beta_{s-1}, \beta_s} \min\{\mathbf{Re}(\lambda) : \lambda \in \sigma(U^\top V)\}$$

Aim: $\overline{\mu} > 0$.

Construction of RK methods satisfying the hypotheses

Compare with Lobatto(s) method

$$\mu_{\min} = \min\{\mathbf{Re}(\lambda) : \lambda \in \sigma(U^\top V)\}$$

s	$\overline{\mu}$	$\mu_{\min}$	β_{s-1}	β_s	α_1	α_2
3	0.6329	0.3855	−0.0005	0.6138	−2.2912	−2.9759
4	0.3371	0.1472	−2.9977	1.0152	−2.9955	0.1492
5	0.2354	0.0155	−1.4125	0.4791	−1.6364	−2.3049
6	0.1131	−0.0431	−0.4872	0.2597	−2.5456	−1.6340
7	0.0476	−0.0701	−0.4240	0.2304	−2.4701	−1.3328
8	0.0034	−0.0892	−0.3376	0.1950	−2.0680	−1.1077
9	−0.0223	−0.1036	−0.3371	0.1985	−1.7087	−0.9326
10	−0.0392	−0.1143	−2.0388	1.1091	−1.4238	−0.8077

Up to $s = 8$ (order 14) RK matrices satisfy the hypotheses of the Theorem.

Eigenvalues are shifted towards the right part of the complex plane by the optimization procedure.

Are the eigenvalues of the preconditioned $\widehat{\Theta}_k$ bounded away from zero?

Heuristic says: Yes!

The eigenvalues of $\mathcal{P}_{\mathrm{RK}}^{-1}\widehat{\Theta}_k$ solve the following generalized eigenvalue problem:

$$(U^\top V + \tau\bar{\lambda}\Sigma)\mathrm{x} = \lambda(I_s + \tau\bar{\lambda}\Sigma)\mathrm{x},$$

where $\bar{\lambda}$ is an eigenvalue of $M^{-\frac{1}{2}}KM^{-\frac{1}{2}}$. This can be seen as a perturbation of

$$\left(\Lambda + \tau\bar{\lambda}\Sigma\right)\mathrm{x} = \lambda\left(I_s + \tau\bar{\lambda}\Sigma\right)\mathrm{x}, \qquad \Lambda = \mathrm{diag}(\mu_1, \ldots, \mu_s).$$

For every j, setting $\mu_j \equiv a + \mathbf{i}b$, $c = \tau\bar{\lambda}\sigma_j$, with $a \geq \mu_{\min} > 0$, we have

$$\lambda = \frac{a + \mathbf{i}b + c}{1 + c} = \frac{a + c}{1 + c} + \mathbf{i}\frac{b}{1 + c}, \qquad |\lambda|^2 = \frac{(a + c)^2 + b^2}{(1 + c)^2}.$$

Whence

$$|\lambda|^2 = \frac{1 + 2ac + c^2}{(1 + c)^2} = 1 + 2(a-1)\frac{c}{(1 + c)^2} \equiv \varphi(c) \geq \varphi(1) = \frac{1 + a}{2} \geq \frac{1 + \mu_{\min}}{2} \geq \frac{1}{2},$$

which shows that the eigenvalues are outside the circle of center 0 and radius $\frac{\sqrt{2}}{2}$.

Alternative preconditioners for $\widehat{\Theta}_k$ (comparisons underway)

L and U: triangular factor in the LU decomposition of A_{RK}^{-1} (with $\mathrm{diag}(U) = I_s$)

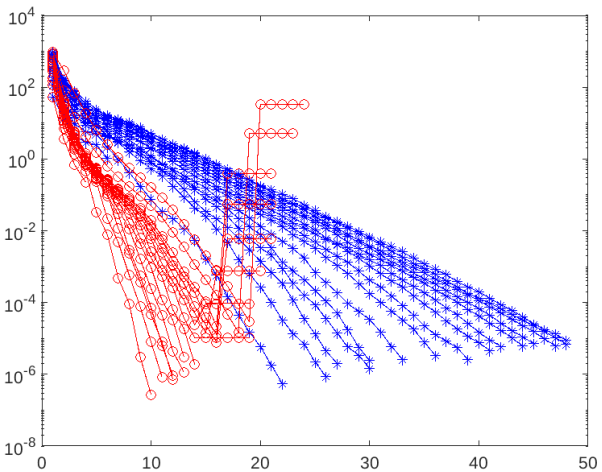
Spectral decomposition of L as

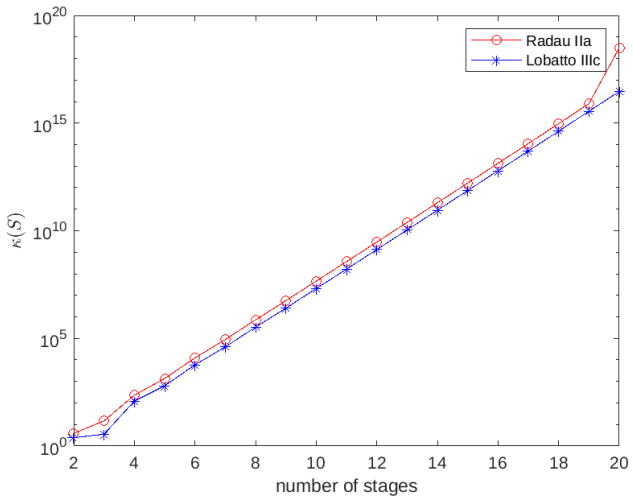
$$L = S\Lambda S^{-1}.$$

Munch-et-al's preconditioner is defined as

$$\mathbf{P}_{\mathrm{RK}}^{-1} = (S \otimes I_n)\left(\Lambda \otimes M + \tau I_s \otimes K\right)^{-1}\left(S^{-1} \otimes I_n\right)$$

- **PRO**: Eigenvalue distribution of $\mathbf{P}_{\mathrm{RK}}^{-1}\widehat{\Theta}_k$ more favorable.
- **CON**: Matrix S exponentially ill-conditioned with the number of stages s.





Munch, Dravins, Kronbichler, Neytcheva
Stage-parallel fully implicit Runge-Kutta implementations with optimal multilevel preconditioners at the scaling limit
https://doi.org/10.48550/arXiv.2209.06700

Approximate inversion of the Schur complement

Recalling the expression for S.

$$S = \begin{bmatrix} \mathsf{M} & & \\ & \ddots & \\ & & \mathsf{M} \end{bmatrix} \underbrace{\begin{bmatrix} I_{n_x} & & & \\ -I_{n_x} + \widehat{X} & \ddots & & \\ & \ddots & \ddots & \\ & & -I_{n_x} + \widehat{X} & I_{n_x} \end{bmatrix}}_{\widehat{S}},$$

Solving then for S requires multiplying by matrix

$$\widehat{X} = \tau \begin{bmatrix} b_1 I_{n_x} & \dots & b_s I_{n_x} \end{bmatrix} \left[I_s \otimes \mathsf{M} + \tau A_{\mathrm{RK}} \otimes \mathsf{K} \right]^{-1} \left[\mathsf{e} \otimes \mathsf{K} \right]$$

which in its turn calls for (a GMRES) solution of n_t systems with $\widehat{\Theta}_H$.

Stokes equations

$$\begin{cases} \dfrac{\partial \vec{v}}{\partial t} - \nabla^2 \vec{v} + \nabla p = \vec{f}(\mathrm{x}, t) & \text{in } \Omega \times (0, t_f), \\ \nabla \cdot \vec{v} = 0 & \text{in } \Omega \times (0, t_f), \\ \vec{v}(\mathrm{x}, t) = \vec{g}(\mathrm{x}, t) & \text{on } \partial\Omega \times (0, t_f), \\ \vec{v}(\mathrm{x}, 0) = \vec{v}_0(\mathrm{x}) & \text{in } \Omega. \end{cases}$$

click to go to the test

After dividing the time interval $[0, t_f]$ into n_t subintervals, the discretization of the Stokes equation by a Runge–Kutta method reads:

$$\begin{aligned} M_v \mathrm{v}_{n+1} &= M_v \mathrm{v}_n + \tau M_v \textstyle\sum_{i=1}^{s} b_i \mathrm{k}_{in}^v \qquad n = 0, \dots, n_t - 1, \\ M_p \mathrm{p}_{n+1} &= M_p \mathrm{p}_n + \tau M_p \textstyle\sum_{i=1}^{s} b_i \mathrm{k}_{in}^p \qquad n = 0, \dots, n_t - 1, \end{aligned}$$

The stages k_{in}^v and k_{in}^p are defined as:

$$\begin{aligned} & M_v \mathrm{k}_{in}^v + K_v \mathrm{v}_n + \tau K_v \textstyle\sum_{j=1}^{s} a_{ij} \mathrm{k}_{jn}^v + B^\top \mathrm{p}_n + \tau B^\top \sum_{j=1}^{s} a_{ij} \mathrm{k}_{jn}^p = \mathrm{f}_{in}, \\ & B \mathrm{v}_n + \tau B \textstyle\sum_{j=1}^{s} a_{ij} \mathrm{k}_{jn}^v = 0, \end{aligned}$$

- f_{jn} accounts for the discretization of the source term,
- K_v and M_v (resp., K_p and M_p) are the *vector–* (resp. *pressure–*) stiffness and mass matrices.
- B (B^T) is the discrete divergence (gradient) operator.

All-at-once RK-Stokes

In matrix form, the system is, as before,

$$\underbrace{\begin{bmatrix} \Phi & \Psi_1 \\ \Psi_2 & \Theta_S \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \mathbf{v} \\ \mathbf{k} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{bmatrix}, \quad \text{with}$$

$$\Phi = \begin{bmatrix} 1 & & & \\ -1 & \cdots & & \\ & \cdots & \cdots & \\ & & -1 & 1 \end{bmatrix} \otimes M, \qquad \Psi_2 = I_{n_t} \otimes \widehat{\Psi}_2$$

$$\Psi_1 = \begin{bmatrix} 0 & \\ & I_{n_t} \end{bmatrix} \otimes (\tau \mathrm{b}_{\mathrm{RK}}^{\top} \otimes \mathrm{M}), \qquad \Theta_S = I_{n_t} \otimes \widehat{\Theta}_S$$

where the blocks defined for each time-step are $M = \begin{bmatrix} M_v & 0 \\ 0 & M_p \end{bmatrix}$ and

$$\widehat{\Psi}_2 = \begin{bmatrix} \mathrm{e} \otimes K_v & \mathrm{e} \otimes (B^{\top}) \\ \mathrm{e} \otimes B & 0 \end{bmatrix} \quad \widehat{\Theta}_S = \begin{bmatrix} I_s \otimes M_v + \tau A_{\mathrm{RK}} \otimes K_v & \tau A_{\mathrm{RK}} \otimes B^{\top} \\ \tau A_{\mathrm{RK}} \otimes B & 0 \end{bmatrix}$$

Preconditioner

Applying preconditioner $\mathbf{P} = \begin{bmatrix} S & \Psi_1 \\ 0 & \Theta_S \end{bmatrix}$ rests on efficiently approximating block $\widehat{\Theta}_S$

$$\widehat{\Theta}_S = \begin{bmatrix} I_s \otimes M_v + \tau A_{\mathrm{RK}} \otimes K_v & \tau A_{\mathrm{RK}} \otimes B^\top \\ \tau A_{\mathrm{RK}} \otimes B & 0 \end{bmatrix} = \begin{bmatrix} \widehat{\Theta}_H & \tau A_{\mathrm{RK}} \otimes B^\top \\ \tau A_{\mathrm{RK}} \otimes B & 0 \end{bmatrix}$$

In order to solve for this matrix, we employ as a preconditioner

$$\mathbf{P}_{\mathrm{RK}} = \begin{bmatrix} I_s \otimes M_v + \tau A_{\mathrm{RK}} \otimes K_v & 0 \\ \tau A_{\mathrm{RK}} \otimes B & S_{\mathrm{RK}} \end{bmatrix} \equiv \begin{bmatrix} \widehat{\Theta}_H & 0 \\ \tau A_{\mathrm{RK}} \otimes B & S_{\mathrm{RK}} \end{bmatrix}$$

where

$$S_{\mathrm{RK}} = -\tau^2 (A_{\mathrm{RK}} \otimes B) \widehat{\Theta}_H^{-1} (A_{\mathrm{RK}} \otimes B^\top).$$

Again we need an efficient strategy to solve a system with $\mathbf{P}_{\mathrm{RK}}$.

The $(1,1)$ block in $\mathbf{P}_{\mathrm{RK}}$ is preconditioned as described before for the Heat equation.

Application of S_{RK}

$$S_{\mathrm{RK}} = -\tau^2 (A_{\mathrm{RK}} \otimes I_{n_p}) S_{\mathrm{int}} (A_{\mathrm{RK}} \otimes I_{n_p}).$$

where, using properties of the Kronecker product, we have defined

$$S_{\mathrm{int}} = (I_s \otimes B) \widehat{\Theta}_H^{-1} (I_s \otimes B^\top).$$

We approximate (precondition) S_{int} with

$$\widetilde{S}_{\mathrm{int}} := (I_s \otimes K_p)(I_s \otimes M_p + \tau A_{\mathrm{RK}} \otimes K_p)^{-1} (I_s \otimes M_p).$$

employing the block-commutator argument derived independently in

Leveque, Pearson

Parameter-robust preconditioning for Oseen iteration applied to stationary and instationary Navier–Stokes control
SISC, 2022

Danieli, Southworth, Wathen

Space-time block preconditioning for incompressible flow
SISC, 2022

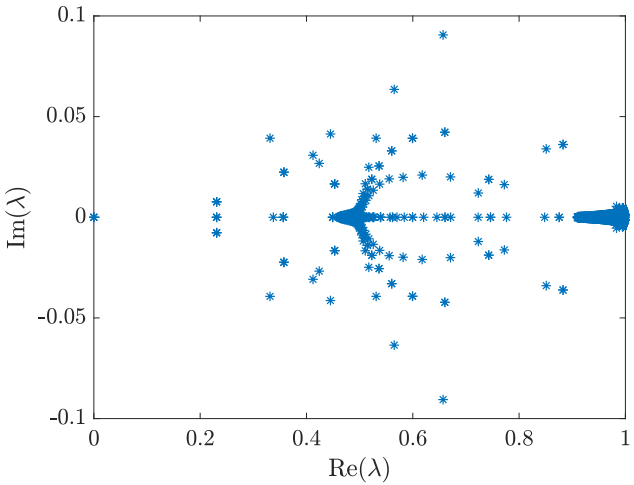
Then, our approximation of the Schur complement is given by

$$\widetilde{S}_{\mathrm{RK}} = -\tau^2 (A_{\mathrm{RK}} \otimes I_{n_p}) \widetilde{S}_{\mathrm{int}} (A_{\mathrm{RK}} \otimes I_{n_p}).$$

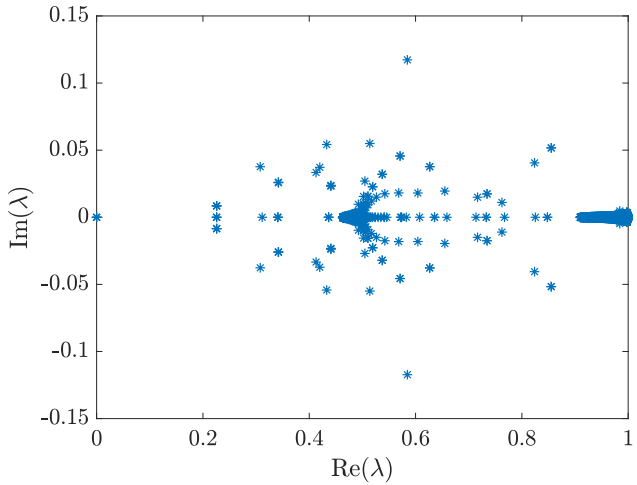
How good is $\widetilde{S}_{\mathrm{RK}}$ as a preconditioner for S_{RK}?

Eigenvalues of $\widetilde{S}_{\mathrm{RK}}^{-1} S_{\mathrm{RK}}$

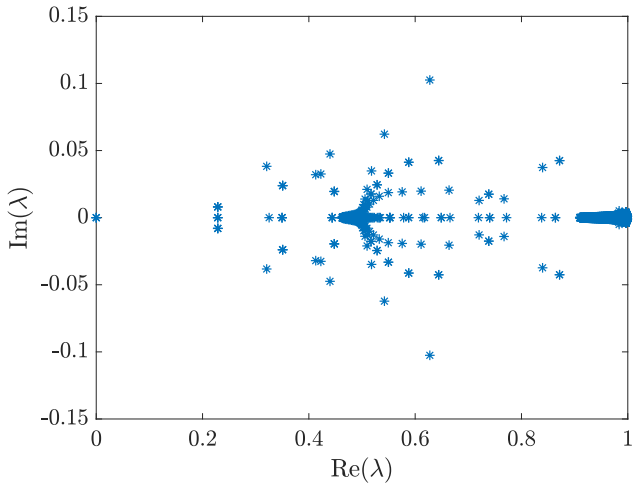
3-stages Gauss.



3-stages Lobatto IIIC.



3-stages Radau IIA.



Summarizing

$$\widetilde{\mathbf{P}}_{\mathrm{RK}} = \begin{bmatrix} U \otimes I_{n_v} & 0 \\ 0 & U \otimes I_{n_p} \end{bmatrix} \widetilde{\mathbf{P}}_{\mathrm{int}} \begin{bmatrix} V^\top \otimes I_{n_v} & 0 \\ 0 & V^\top \otimes I_{n_p} \end{bmatrix},$$

with

$$\widetilde{\mathbf{P}}_{\mathrm{int}} = \begin{bmatrix} I_s \otimes M + \tau \Sigma \otimes K & 0 \\ \tau \Sigma \otimes B & -\tau^2 \left((\Sigma V^\top) \otimes I_{n_p} \right) \widetilde{S}_{\mathrm{int}} \left((U \Sigma) \otimes I_{n_p} \right) \end{bmatrix}.$$

$$\widetilde{S}_{\mathrm{int}} := (I_s \otimes K_p)(I_s \otimes M_p + \tau A_{\mathrm{RK}} \otimes K_p)^{-1}(I_s \otimes M_p).$$

Solution of a system with $\widetilde{\mathbf{P}}_{\mathrm{RK}}$ involves solution of a system with $\widehat{\Theta}_H$ and "inversion" of $\widetilde{S}_{\mathrm{RK}}$.

$$\widetilde{S}_{\mathrm{RK}}^{-1} = -\tau^{-2} \left((U\Sigma) \otimes I_{n_p} \right)^{-1} (I_s \otimes M_p^{-1})(I_s \otimes M_p + \tau A_{\mathrm{RK}} \otimes K_p)(I_s \otimes K_p^{-1}) \left((\Sigma V^\top) \otimes I_{n_p} \right)^{-1}$$

whose application to a vector is carried on by iterative solution of systems with M_p and K_p.

Stokes equations: parallel all-at-once solve

Problem: time-dependent version of the *lid-driven cavity*

Initial condition $\vec{v}(\mathrm{x}, 0) = \vec{0}$

Flow described by the Stokes equations with $\vec{f}(\mathrm{x}, t) = \vec{0}$ and the following boundary conditions:

$$\vec{g}(\mathrm{x}, t) = \begin{cases} [t, 0]^\top & \text{on } \partial\Omega_1 \times (0, 1), \\ [1, 0]^\top & \text{on } \partial\Omega_1 \times [1, t_f), \\ [0, 0]^\top & \text{on } (\partial\Omega \setminus \partial\Omega_1) \times (0, t_f). \end{cases}$$

We set $\partial\Omega_1 := (-1, 1) \times \{1\}$.

Simulations up to more than 200 time units.

Numerical results. Sequential

- 2D examples.
- Discretizations: Q_1 and Q_2 FE for the heat equation; inf–sup stable Taylor–Hood Q_2–Q_1 for the Stokes equations.
- ℓ denotes the level of refinement: $h = 2^{-\ell-1}$ (Q_1), $h = 2^{-\ell}$ (Q_2)
- Solution to mass matrices: 20 steps of Chebyshev semi-iteration
- Solution to stiffness/related matrices: Multigrid with 2 V-cycles of the `HSL_MI20` solver.
- FGMRES as the outer solver (10^{-8} tolerance on the relative residual)
- Linear systems with Θ_H are solved with GMRES and maximum number of iterations 5 (Heat equation) 10 (Stokes equation).
- MATLAB R2018b, (1.70GHz Intel quad-core i5 processor with 8 GB RAM)

Heat equation: All-at-once solver with Lobatto IIIC methods

Test case with exact solution known.

it = outer FGMRES iterations

s	ℓ	Q_1 DoF	Q_1 it	Q_1 CPU	Q_1 v_{error}	Q_2 DoF	Q_2 it	Q_2 CPU	Q_2 v_{error}
2	4	11,025	6	1.7	3.56e-03	133,579	7	16	3.25e-04
	5	93,217	7	8.2	1.05e-03	1,528,065	8	162	4.56e-05
	6	766,017	7	58	2.71e-04	17,580,610	8	1979	4.64e-06
3	4	5625	6	1.1	1.28e-03	39,401	7	4.9	1.58e-05
	5	31,713	6	3.3	3.63e-04	257,985	7	27	2.59e-06
	6	194,481	7	16	8.27e-05	1,758,061	8	231	2.58e-07
4	4	4725	8	1.1	1.54e-03	29,791	9	7	1.57e-06
	5	29,791	10	4.7	3.53e-04	162,729	10	26	7.26e-08
	6	142,884	9	16.0	8.40e-05	983,869	9	157	3.71e-07
5	4	5625	9	1.7	1.54e-03	29,791	9	5.5	1.17e-06
	5	24,025	9	3.6	4.01e-04	146,853	10	25	5.30e-08
	6	123,039	9	15.0	9.64e-05	790,321	9	137	2.18e-07

Heat equation: All-at-once solver with Radau IIA methods

it = outer FGMRES iterations

s	ℓ	Q_1 DoF	Q_1 it	Q_1 CPU	Q_1 v_{error}	Q_2 DoF	Q_2 it	Q_2 CPU	Q_2 v_{error}
2	4	5625	6	0.9	1.35e-03	47,089	7	5.2	2.51e-05
	5	38,440	6	3.0	3.46e-04	384,993	8	40	3.41e-06
	6	254,016	7	19	8.35e-05	3,112,897	8	351	5.71e-07
3	4	4725	7	0.9	1.46e-03	27,869	7	3.4	2.28e-06
	5	27,869	7	2.9	3.30e-04	178,605	8	21	2.24e-07
	6	130,977	7	11	1.03e-04	1,048,385	8	136	2.68e-08
4	4	4725	9	1.4	1.54e-03	24,986	10	4.7	1.12e-06
	5	24,986	10	4.0	3.77e-04	142,884	12	27	7.14e-08
	6	123,039	10	15	9.00e-05	741,934	12	159	1.34e-08
5	4	5625	10	1.9	1.54e-03	24,025	10	4.8	1.20e-06
	5	24,025	10	4.1	4.01e-04	146,853	9	22	3.15e-07
	6	123,039	10	17	9.65e-05	693,547	9	119	1.82e-07

Comment: satisfactory scalability with respect to both meshsize parameter ℓ and number of stages.

Comparisons with ParaDiag. Sequential

All-at-once solve of the Heat equation.

ParaDiag vs 5-stage Radau IIA methods – Q2 finite elements.

s	ParaDiag				5-stage Radau IIA			
	DOF	it	CPU	v_{error}	DOF	it	CPU	v_{error}
3	3825	4	0.1	2.26e-04	5625	10	1.1	1.92e-05
4	45,167	4	1.1	2.68e-05	24,025	9	3.7	1.21e-06
5	512,001	3	14	3.41e-06	146,853	10	32	7.59e-08
6	5,870,956	3	179	4.26e-07	693,547	10	176	1.08e-07
7	†	–	–	–	3,186,225	8	761	6.47e-07

For fine meshes the proposed solver outperforms ParaDIAG.

This is mainly due to the high order RK discretization which allows reduction of number of time-steps and hence the overall size of the system.

Gander, Liu, Wu, Yue, Zhou

ParaDiag: parallel-in-time algorithms based on the diagonalization technique
https://doi.org/10.48550/arXiv.2005.09158, 2021

Parallel machine and efficiency indicators

The parallel code is written in Fortran 95 with pure MPI as the message passing protocol, and compiled with the -O5 option.

Computing machine: Marconi 100: IBM Power AC922 (located at CINECA, Bologna) with 980 computing nodes and 2x16 cores at 2.6(3.1) GHz on each node.

T_{n_p} = CPU seconds when running the code on n_p processors.

Parallel efficiency measures: **Speedup** S_{n_p} and **Efficiency** E_{n_p}:

$$S_{n_p} = \frac{T_1}{T_{n_p}}, \qquad E_{n_p} = \frac{S_{n_p}}{n_p} = \frac{T_1}{T_{n_p} n_p}.$$

$T^{\Theta}_{n_p}$ and $T^{S}_{n_p}$ $\rightarrow$ approximate inversion of Θ and S preconditioner blocks.

We approximate (from below) T_1 time by setting $T_1 := 2 n_p T^{\Theta}_{n_p}$ since

$$T_1 \lesssim 2 T^{\Theta}_1 \lesssim 2 n_p T^{\Theta}_{n_p}$$

Thus we **underestimate** T_1 and also both S_{n_p} and E_{n_p}.

Parallelization

Uniform distribution of the data: each processor manages n_t/n_p timesteps.

To solve the system with $\widehat{S}$ at every FGMRES iteration we employed the Parallel MGRIT: Xbraid with the following parameters

- Coarsening factor: `cfactor` $= 4$,
- Minimum coarse grid size: `min_coarse` $= 2$,
- Maximum number of iterations: `max_its` $= 1$.

Test cases (last column: number of FGMRES outer iterations)

#	Eq.	RK method	T_f	order FE	order RK	n_t	ℓ	s	Total Ndof	Its
1	Heat	Radau IIA	255.9	3	7	2048	8	4	2 674 140 160	**11**
2	Heat	Lobatto IIIC	2047.9	3	6	16384	7	4	5 326 913 024	**9**
3	Stokes	Radau IIA	386.8	2	5	2048	7	3	1 201 839 360	**13**
4	Stokes	Lobatto IIIC	510.7	2	6	2048	7	4	1 502 262 528	**19**

Heat equation

Test case # 1

p	$T^{\Theta}_{n_p}$	$T^{S}_{n_p}$	T_{n_p}	S_{n_p}	E_{n_p}
64	14471	17536	32065	58	90%
128	7254	11717	19000	98	76%
256	3639	9511	13166	141	55%
512	1888	8419	10320	180	35%

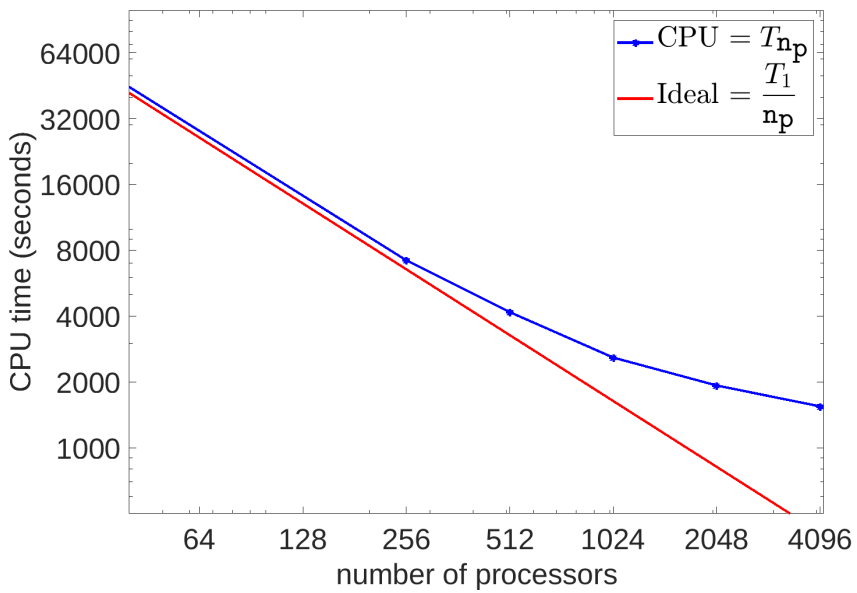
Test case # 2

256	3061	4103	7187	224	87%
512	1586	2568	4168	386	75%
1024	784	1785	2583	622	61%
2048	427	1485	1926	834	41%
4096	205	1334	1545	1040	25%

CPUs vs n_p for test case #2.

Blue line: real CPUs

Red line: ideal CPUs assuming 100% efficiency.



Parallel run with $n_p = 4096$	1545 seconds	=	25 minutes
Sequential run (projected CPU)	288381 seconds	=	3 days 8 hours

Results for the Stokes equation

Simulation of the Lid-Driven Cavity.

Each block of the preconditoner is approximately inverted with 10 GMRES (inner) iterations

(Outer) FGMRES is restarted after 20 iterations, with tolerance of 10^{-6}.

CPUs, speedups and parallel efficiencies for test case #3 (left) and #4 (right).

p	$T^{\Theta}_{n_p}$	$T^{S}_{n_p}$	T_{n_p}	S_{n_p}	E_{n_p}	$T^{\Theta}_{n_p}$	$T^{S}_{n_p}$	T_{n_p}	S_{n_p}	E_{n_p}
64	10345	13540	23925	55	87%	21314	26696	48084	57	89%
128	5180	8898	14100	94	73%	10685	17521	28247	97	76%
256	2624	7228	9870	134	52%	5257	13921	19207	142	56%
512	1309	6288	7606	174	34%	2695	12375	15085	181	35%
	Radau IIIA					Lobatto IIC				

State-of-the-art and Next steps

What we have achieved:

- Block triangular preconditioner based on the SVD of A_{RK} (no complex arithmetics!)
- In the Stokes problem this is combined with the block-commutator preconditioner.
- Robustness wrt space discretization parameter h and to the number of RK stages.
- Parallel solution of the block bidiagonal system successfully performed by MGRIT.
- High speedups and efficiencies up to a number of processors roughly $n_t/4$.

Future directions:

- Variable timestep implementation
- (Non trivial) extension to problems with nonsymmetric operators. Advection-diffusion, Navier-Stokes?
- Parallelization also in space (to handle finer space discretization).

Essential Bibliography

- Our paper

Leveque, Bergamaschi, Martinez and Pearson

Parallel-in-Time Solver for the All-at-Once Runge–Kutta Discretization

https://arxiv.org/abs/2303.02090, 2023

- Alternative preconditioner for the stages

Munch, Dravins, Kronbichler, and Neytcheva

Stage-parallel fully implicit Runge-Kutta implementations with optimal multilevel preconditioners at the scaling limit

https://doi.org/10.48550/arXiv.2209.06700, 2023

- Xbraid MGRIT solver

Falgout, Friedhoff, Kolev, MacLachlan and Schröder

Parallel time integration with multigrid

SISC, 2014

- ParaDiag implementation used for comparisons

Gander, Liu, Wu, Yue, Zhou

ParaDiag: parallel-in-time algorithms based on the diagonalization technique

https://doi.org/10.48550/arXiv.2005.09158, 2021

- First paper on IRK coupled with Multigrid for PDEs

Van Lent and Vandewalle

Multigrid Methods for Implicit Runge–Kutta and BV Method Discretizations of Parabolic PDEs

SISC, 2005