Parameter-robust preconditioning for unsteady Stokes control problems

Santolo Leveque and John W. Pearson

1 School of Mathematics, The University of Edinburgh, The King’s Buildings, Edinburgh, EH9 3FD, UK

We propose a saddle-point preconditioner for an optimization problem constrained by the time-dependent Stokes equations, discretized using the backward Euler method in time. The key ingredients are an inner iteration for the (1, 1)-block, accelerated by a known preconditioner for the heat control problem, and an approximation of the Schur complement involving a commutator argument applied to a block matrix. Numerical results demonstrate the efficacy and robustness of this approach.

1 Problem statement and resulting linear system

The rapid iterative solution of time-dependent partial (PDE) [9] or fractional (FDE) [10] differential equation constrained optimization problems is a key challenge in numerical linear algebra. Here, we consider the unsteady Stokes control problem:

\[
\min_{\tilde{v}, \tilde{u}} \frac{1}{2} \int_0^T \left| \frac{d}{dt} \tilde{v} - \tilde{b} \right|^2 \, dx \, dt + \frac{\beta}{2} \int_0^T \left| \frac{d}{dt} \tilde{u} \right|^2 \, dx \, dt \quad \text{s.t.} \quad \begin{cases} \frac{d}{dt} \tilde{v} - \nabla \cdot \tilde{u} = 0 & \text{in } Q, \\ \nabla \cdot \tilde{v} = 0 & \text{in } Q, \quad \tilde{v} = \tilde{g} \quad \text{at } t = 0. \end{cases}
\]

The state variables \((\tilde{v}, \tilde{p})\) denote velocity and pressure in space–time domain \(Q := \Omega \times (0, T)\), with \(\tilde{g}\) equipped with boundary data on \(\partial \Omega\) and an initial condition, \(\tilde{u}\) and \(\tilde{u}_i\) denote the control variable and desired state, and \(\beta\) is a regularization parameter.

We discretize the PDE constraints above and the corresponding adjoint equations (with adjoint variables \(\tilde{\lambda}, \mu \) to \(\tilde{v}, \tilde{p}\)), using the gradient equation \(\beta \tilde{u} - \tilde{\lambda} = 0\) to eliminate \(\tilde{u}\). Applying backward Euler in time leads to the ‘saddle-point type’ system:

\[
\begin{pmatrix}
\mathbf{A} & \mathbf{B}^T \\
\mathbf{B} & 0
\end{pmatrix}
\begin{pmatrix}
\mathbf{v} \\
\lambda
\end{pmatrix}
- \begin{pmatrix}
\mathbf{L} & \frac{1}{2} \mathbf{M}\lambda \\
\mathbf{B}_v & 0
\end{pmatrix}
\begin{pmatrix}
\mathbf{b}_1 \\
b_2
\end{pmatrix}
- \begin{pmatrix}
\mathbf{B}_v^T \\
0
\end{pmatrix}
\begin{pmatrix}
\mathbf{v} \\
\lambda
\end{pmatrix}
= \begin{pmatrix}
\mathbf{v}_d \\
b_3
\end{pmatrix}.
\]

The vectors \(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\) arise from the boundary data, \(\mathbf{v}_d\) corresponds to the desired state, and

\[
\mathbf{L} = \begin{pmatrix}
\mathbf{M} + \tau \mathbf{K} & \mathbf{M} + \tau \mathbf{K} \\
-\mathbf{M} & \mathbf{M} + \tau \mathbf{K}
\end{pmatrix},
\mathbf{M}_v = \text{blkdiag}(\tau \mathbf{M}, \tau \mathbf{M}, ..., \tau \mathbf{M}, 0),
\mathbf{M}_\lambda = \text{blkdiag}(0, \tau \mathbf{M}, ..., \tau \mathbf{M}, \tau \mathbf{M}),
\mathbf{B}_v = \text{blkdiag}(\tau \mathbf{B}, \tau \mathbf{B}, ..., \tau \mathbf{B}, \tau \mathbf{B}),
\]

with \(\mathbf{K}\) discretizing the (negative) vector Laplacian, \(\mathbf{M}\) the identity operator in a vectorial sense, and \(\mathbf{B}\) the (negative) divergence. We take a time-step size \(\tau = T/n_t\), with \(n_t\) time-steps. Each block matrix in (1) contains \(n_t + 1\) blocks, with the MATLAB notation blkdiag defining a block diagonal matrix. For the spatial discretization, we elect to use Taylor–Hood \((Q_2-Q_1)\) finite elements. We now present a preconditioner for (1), based on work in [6] for Navier–Stokes control problems.

2 Preconditioned iterative method for linear system

To solve the linear system (1) we employ a block-triangular preconditioner \(\mathcal{P}\) which may be summarized as follows:

\[
\mathcal{P} = \begin{pmatrix}
\tilde{\mathbf{A}} & 0 \\
\mathbf{B} & -\tilde{\mathbf{S}}
\end{pmatrix},
\tilde{\mathbf{A}} \approx \begin{pmatrix}
\tilde{\mathbf{M}}_v & 0 \\
\mathbf{L} & -\tilde{\mathbf{S}}_\lambda
\end{pmatrix},
\tilde{\mathbf{M}}_v = \text{blkdiag}(\tau \mathbf{M}, \tau \mathbf{M}, ..., \tau \mathbf{M}, \epsilon \tau \mathbf{M}),
\tilde{\mathbf{S}}_\lambda = \left(\mathbf{L} + \frac{1}{\sqrt{\epsilon}} \tilde{\mathbf{M}}_\lambda \right) \tilde{\mathbf{M}}_v^{-1} \left(\mathbf{L} + \frac{1}{\sqrt{\epsilon}} \tilde{\mathbf{M}}_\lambda \right)^T,
\]

which may be applied within a Krylov subspace method such as GMRES [11]. The matrix \(\tilde{\mathbf{A}}\) is related to a heat control problem, solved over each spatial dimension, and \(\tilde{\mathbf{A}}\) is known to be a potent and parameter-robust preconditioner for \(\mathbf{A}\) [9]. Here, \(\epsilon > 0\) is a ‘perturbation parameter’ which guarantees invertibility of \(\tilde{\mathbf{M}}_v\), and \(\mathbf{M} = \text{blkdiag}(0, \tau \mathbf{M}, ..., \tau \mathbf{M}, \sqrt{\epsilon} \tau \mathbf{M})\) ensures that \(\tilde{\mathbf{M}}_v \tilde{\mathbf{M}}_v^{-1} \tilde{\mathbf{M}} = \mathbf{M}\). To ensure a sufficiently accurate approximation of \(\tilde{\mathbf{A}}\) within the Stokes control preconditioner \(\mathcal{P}\), we apply \(\tilde{\mathbf{A}}^{-1}\) within an inner iterative method: either GMRES, or an Uzawa iteration (see [2]).

* Corresponding author: e-mail j.pearson@ed.ac.uk

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2021 The Authors. Proceedings in Applied Mathematics & Mechanics published by Wiley-VCH GmbH.
The matrix \( \hat{S} \) approximates the (negative) Schur complement \( S = B A^{-1} B^T \) of the matrix in (1). We apply a commutator argument [3, Ch. 9], which allows one to build approximations of the form \( B L^{-1} B^T \approx K_p L^{-1} M_p \), where \( L \) is a matrix derived on the velocity space, with \( K_p, M_p, L_p \) analogous operators on the pressure space to \( K, M, L \). Based on this we take

\[
\hat{S} = \tau^2 \left( \begin{array}{cc} K_p & 0 \\ 0 & K_p \end{array} \right) \left( \begin{array}{cc} \mathcal{L} & \mathcal{L}^T \\ -\frac{1}{\beta} \mathcal{L}^T \end{array} \right)_{p}^{-1} \left( \begin{array}{cc} M_p & 0 \\ 0 & M_p \end{array} \right), \quad K_p = \text{blkdiag}(K_p, K_p, \ldots, K_p, K_p), \quad M_p = \text{blkdiag}(M_p, M_p, \ldots, M_p, M_p).
\]

See [6] for a complete description of this approach, which involves applying the commutator argument to each block of \( \mathcal{A} \).

### 3 Numerical results for two solution strategies

We test our preconditioner on a problem posed on the space–time domain \((-1,1)^2 \times (0,2)\), with spatial coordinates \((x_1, x_2)\), the boundary condition \(f(x, t) = [\min\{t, 1\}, 0]^T\) on \(x_2 = 1\) with \(f(x, t) = 0\) elsewhere, the initial condition \(g(x) = 0\), and \(v_0(x, t) = \begin{cases} h_1(x) \cos(\frac{\pi t}{2}) \left[ (\frac{100}{49})^2 x_2, -(\frac{100}{49})^2 (x_1 - \frac{1}{2}) \right]^T & \text{if } h_1(x) := 1 - \sqrt{\left(\frac{100}{49} (x_1 - \frac{1}{2})\right)^2 + \left(\frac{100}{49} x_2\right)^2} \geq 0, \\ h_2(x) \cos(\frac{\pi t}{2}) \left[ -(\frac{100}{49})^2 x_2, (\frac{100}{49})^2 (x_1 + \frac{1}{2}) \right]^T & \text{if } h_2(x) := 1 - \sqrt{\left(\frac{100}{49} (x_1 + \frac{1}{2})\right)^2 + \left(\frac{100}{49} x_2\right)^2} \geq 0, \\ [0, 0]^T & \text{otherwise.} \end{cases}\)

Table 1 shows results with flexible GMRES as the outer method, and an inner GMRES solver for the matrix \(\mathcal{A}\).

Table 2: Outer iterations and CPU times for flexible GMRES method, with GMRES as inner solver for \(\mathcal{A}\), \(n_t = 40\), and a range of \(\ell, \beta\).

| \(n_t = 40\) | \(\ell\) | \(\beta = 1\) | \(\beta = 10^{-1}\) | \(\beta = 10^{-2}\) | \(\beta = 10^{-3}\) | \(\beta = 10^{-4}\) | \(\beta = 10^{-5}\) | \(\beta = 10^{-6}\) | DoF |
|---|---|---|---|---|---|---|---|---|---|
| \(n_t\) | \(\ell\) | it | CPU | it | CPU | it | CPU | it | CPU |
| 2 | 15 | 5.79 | 16 | 6.10 | 18 | 6.87 | 17 | 6.44 | 15 | 5.67 | 14 | 5.34 | 20 | 7.61 | 10,086 |
| 3 | 16 | 14.0 | 18 | 16.0 | 19 | 15.3 | 16 | 13.8 | 16 | 12.6 | 16 | 6.07 | 22 | 13.6 | 43,542 |
| 4 | 16 | 36.7 | 19 | 43.5 | 19 | 42.0 | 17 | 44.7 | 17 | 43.7 | 17 | 41.1 | 22 | 49.1 | 181,302 |
| 5 | 16 | 155 | 22 | 211 | 20 | 191 | 17 | 160 | 17 | 150 | 17 | 150 | 21 | 156 | 740,214 |
| 6 | 25 | 1123 | 24 | 1080 | 23 | 1027 | 17 | 754 | 17 | 709 | 17 | 725 | 22 | 932 | 2,991,606 |

Table 2 shows results with flexible GMRES as the outer method, and an inner Uzawa method for the matrix \(\mathcal{A}\).

Table 2: Outer iterations and CPU times for flexible GMRES method, with Uzawa as inner solver for \(\mathcal{A}\), \(\ell = 5\), and a range of \(n_t, \beta\).

| \(\ell = 5\) | \(n_t\) | \(\beta = 1\) | \(\beta = 10^{-1}\) | \(\beta = 10^{-2}\) | \(\beta = 10^{-3}\) | \(\beta = 10^{-4}\) | \(\beta = 10^{-5}\) | \(\beta = 10^{-6}\) | DoF |
|---|---|---|---|---|---|---|---|---|---|
| \(\ell\) | \(n_t\) | it | CPU | it | CPU | it | CPU | it | CPU |
| 10 | 14 | 26.0 | 15 | 28.3 | 17 | 31.6 | 18 | 31.5 | 21 | 36.6 | 40 | 60.1 | 198,594 |
| 12 | 14 | 51.0 | 15 | 54.0 | 17 | 60.8 | 17 | 60.5 | 18 | 56.9 | 18 | 59.3 | 29 | 80.8 | 379,134 |
| 15 | 15 | 103 | 15 | 103 | 16 | 110 | 17 | 115 | 17 | 107 | 18 | 113 | 22 | 116 | 740,214 |
| 20 | 15 | 201 | 15 | 202 | 16 | 213 | 17 | 223 | 17 | 213 | 17 | 211 | 18 | 187 | 1,462,374 |
| 40 | 15 | 375 | 15 | 377 | 16 | 400 | 17 | 421 | 17 | 426 | 17 | 411 | 18 | 370 | 2,906,694 |

Acknowledgements SL acknowledges financial support from a School of Mathematics PhD studentship at the University of Edinburgh. JWP acknowledges support from the Engineering and Physical Sciences Research Council (EPSRC) grant EP/S027785/1. Open access funding enabled and organized by Projekt DEAL.

© 2021 The Authors. Proceedings in Applied Mathematics & Mechanics published by Wiley-VCH GmbH. www.gamm-proceedings.com
References

[1] J. Boyle, M. Mihajlović, and J. Scott, Int. J. Numer. Meth. Eng. 82(1), 64–98 (2010).
[2] H. C. Elman and G. H. Golub, SIAM J. Numer. Anal. 31(6), 1645–1661 (1994).
[3] H. C. Elman, D. J. Silvester, and A. J. Wathen, Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics, Oxford University Press, 2nd Ed. (2014)
[4] G.H. Golub and R. S. Varga, Numer. Math. 3, 147–156 (1961).
[5] S. Leveque and J. W. Pearson, arXiv preprint arXiv:2007.08410 (2020).
[6] S. Leveque and J.W. Pearson, Parameter-robust preconditioning for Oseen iteration applied to stationary and instationary Navier–Stokes control, arXiv preprint arXiv:2108.00282 (2021).
[7] Y. Notay, AGMG software and documentation; see http://agmg.eu/index.html.
[8] I. V. Oseledets et al., TT-Toolbox software; see https://github.com/oseledets/TT-Toolbox.
[9] J. W. Pearson, M. Stoll, and A. J. Wathen, SIAM J. Matrix Anal. Appl. 33(4), 1126–1152 (2012).
[10] S. Pougkakiotis, J. W. Pearson, S. Leveque, and J. Gondzio, SIAM J. Matrix Anal. Appl. 41(3), 1443–1476 (2020).
[11] Y. Saad and M. H. Schultz, SIAM J. Sci. Stat. Comput. 7(3), 856–869 (1986).
[12] Y. Saad, SIAM J. Sci. Comput. 14(2), 461–469 (1993).