Preconditioned iterative methods for problems arising in PDE-constrained optimization

Performance study

Owe Axelsson, Shiraz Farouq, Maya Neytcheva
Uppsala University

Umeå, August 26-27, 2015

Outline of the talk

PDE-constrained optimization

Discretized optimal control problems (FEM) and arising algebraic structures

Benchmark problems

Preconditioning

Numerical results, performance comparison

Take-away message

Significant scientific interest in solution methods for PDE-constrained optimization

J. Schöberl, W. Zulehner, Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems. SIAM J. Matrix Anal. Appl. **2007**

T. Rees, M. Stoll, Block-triangular preconditioners for PDE-constrained optimization. Numer. Linear Algebra Appl. **2010**

T. Rees, M. Stoll, A Wathen, All-at-once preconditioning in PDE-constrained optimization. Kybernetika (Prague) **2010**

T. Rees, H.S. Dollar, A. Wathen, Optimal solvers for PDE-constrained optimization. SIAM J. Sci. Comput. **2010**

M. Benzi, E. Haber, L. Taralli, A preconditioning technique for a class of PDE-constrained optimization problems. Adv. Comput. Math. **2011**

Significant scientific interest, cont.

Z.-Z. Bai, Block preconditioners for elliptic PDE-constrained optimization problems. Computing 2011

J. Pearson, M. Stoll, A. Wathen, Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems. SIAM J. Matrix Anal. Appl. 2012

J. Pearson, A. Wathen, A new approximation of the Schur complement in preconditioners for PDE-constrained optimization. Numer. Linear Algebra Appl. 2012

J. Pearson, Fast iterative solvers for PDE-constrained optimization problems. Thesis (D.Phil.), University of Oxford (United Kingdom). 2013

J. Pearson, A radial basis function method for solving PDE-constrained optimization problems. Numer. Algorithms 2013

G. Zhang, Z. Zheng, Block-symmetric and block-lower-triangular preconditioners for PDE-constrained optimization problems. J. Comput. Math. 2013

Significant scientific interest, cont.

W. Zulehner, Efficient solvers for saddle point problems with applications to PDE-constrained optimization. Advanced finite element methods and applications, LNACM, 66, Springer, Heidelberg, 2013

M. Stoll, One-shot solution of a time-dependent time-periodic PDE-constrained optimization problem. IMA J. Numer. Anal. 2014

M. Stoll, T. Breiten, Low-rank in time approach to PDE-constrained optimization. SIAM J. Sci. Comput. 2015

Z. Xiaoying Zhang Yumei Huang, On block preconditioners for PDE-constrained optimization problems, Journal of Computational Mathematics, in press.

PDE-constraint optimization

- General formulation of a PDE-constrained optimization problem:

$$\min_{y,u} \mathcal{J}(y,u)$$
$$\text{subject to} \quad \underbrace{\mathcal{L}(y,u)=0}_{\text{the state equation}},$$

$\mathcal{J}$ represents the cost functional,
$\mathcal{L}$ is a PDE-constraint,
y is the state variable,
u is the decision/control/design or parameter identification variable.

Dealing with optimization problems

We construct the so-called Lagrangian:

$$\mathbf{L}(y, u, \lambda) = \mathcal{J}(y, u) + \lambda \cdot \mathcal{L}(y, u)$$

with λ - the Lagrange variable (the dual or adjoint state variable). Most often finding the solution of PDE-constrained minimization problem is through the *first order optimality* conditions, also known as the Karush-Kuhn-Tucker (KKT) conditions:

$$\frac{\partial \mathbf{L}}{\partial y} = 0, \quad \frac{\partial \mathbf{L}}{\partial u} = 0, \quad \underbrace{\frac{\partial \mathbf{L}}{\partial \lambda} = 0}_{\mathcal{L}(y,u)=0}.$$

Dealing with the KKT system numerically

Numerically dealing with the PDE-constrained optimization problems requires two steps: **discretization** and **optimization**.

Two possible approaches:

- Optimize then discretize
  - Formulate the Lagrangian and its corresponding first order optimality conditions, discretize them, and form the algebraic system.
- Discretize then optimize
  - Discretize the objective function, formulate its Lagrangian and the corresponding first order optimality conditions, and then form an algebraic system.

• For some PDE-constraint optimization problems, especially when the PDE is not *self-adjoint*, the two approaches lead to different algebraic systems.

PDE-constraint optimization, cont.

- Consider a heating source u applied to the surface of a domain Ω to control the temperature y. The temperature distribution y in Ω is described by $-\Delta y = u$ (or $-\nabla \cdot a \nabla y = u$) Ω.
- Control the heating source u, such that Ω acquires a temperature y, as close to the target $\hat{y}$ as possible.

- This task takes the form of an optimization problem:

$$\min_{y,u} \mathcal{J}(y,u) = \frac{1}{2}\|y - \hat{y}\|^2_{L2(\Omega)} + \frac{1}{2}\beta\|u\|^2_{L^2(\Omega)}$$

subject to $\underbrace{-\Delta y = u \text{ in } \Omega}_{\text{distributed control}}$ or $\underbrace{-\Delta y = f, y = u \text{ on } \partial\Omega}_{\text{boundary control}}$

The term $\frac{1}{2}\beta\|u\|^2_{L_2(\Omega)}$ is added to make the solution well-defined.
$\beta > 0 \Longrightarrow$ the regularization parameter.

Benchmarking distrib. opt. control problems

Task:

compare the performance of different numerical solution techniques and preconditioners

using the same software, on one and the same computer.

- the distributed optimal control of the Poisson equation.
- the distributed optimal control of the convection-diffusion equation.

Benchmarking distrib. opt. control problems

$$\min_{y,u} \frac{1}{2}\|y-\hat{y}\|^2_{L^2(\Omega)} + \frac{1}{2}\beta\|u\|^2_{L^2(\Omega)}$$

s.t.

$-\Delta y = u$ in Ω, **Poisson control**

$y = \hat{y}|_{\partial\Omega}$ on $\partial\Omega$

$-\varepsilon\Delta y + (\vec{w}\cdot\nabla)y + cy = u$ in Ω **Conv.Diff. control**

$y = \hat{y}|_{\partial\Omega}$ on $\partial\Omega$

$\Omega = [0,1]^2$ defines the domain with boundary $\partial\Omega$.

$\hat{y}$ is the desired state given by

Poisson:
$$\hat{y} = \begin{cases} (2x_1 - 1)^2(2x_2 - 1)^2 & \text{if } \mathbf{x} \in \left[0, \frac{1}{2}\right]^2 \\ 0 & \text{otherwise.} \end{cases}$$

Conv.Diff.: same $\hat{y}$, $\vec{w} = [cos\theta, sin\theta]$ for $\theta = \frac{\pi}{4}$,

with $\varphi(z) = (1 - cos(0.8\pi z))(1 - z)^2$.

Distrb.opt. control, Poisson equation, KKT

$$\mathcal{A}\begin{bmatrix}\mathbf{y}\\ \mathbf{u}\\ \boldsymbol{\lambda}\end{bmatrix} \equiv \begin{bmatrix} M & 0 & K^T \\ 0 & \beta M & -M \\ K & -M & 0 \end{bmatrix}\begin{bmatrix}\mathbf{y}\\ \mathbf{u}\\ \boldsymbol{\lambda}\end{bmatrix} = \begin{bmatrix}\mathbf{b}\\ \mathbf{0}\\ \mathbf{d}\end{bmatrix}.$$

If the state $\mathbf{y}$, the control $\mathbf{u}$ and the adjoint $\boldsymbol{\lambda}$ are discretized using the same finite element spaces, then we can eliminate the control $\mathbf{u} = \frac{1}{\beta}\boldsymbol{\lambda}$, and reduce the system:

$$\mathcal{A}\begin{bmatrix}\mathbf{y}\\ \boldsymbol{\lambda}\end{bmatrix} \equiv \begin{bmatrix} M & K^T \\ K & -\frac{1}{\beta}M \end{bmatrix}\begin{bmatrix}\mathbf{y}\\ \boldsymbol{\lambda}\end{bmatrix} = \begin{bmatrix}\mathbf{b}\\ \mathbf{d}\end{bmatrix} \quad (K = K^T).$$

Conv.Diff.: Stabilization using Local Projection schemes, KKT

Local projection stabilization, Becker and Vexler (2007), leads to:

$$\mathcal{A}\begin{bmatrix}\mathbf{y}\\ \mathbf{u}\\ \boldsymbol{\lambda}\end{bmatrix} \equiv \begin{bmatrix} M & 0 & F^T\\ 0 & \beta M & -M\\ F & -M & 0\end{bmatrix}\begin{bmatrix}\mathbf{y}\\ \mathbf{u}\\ \boldsymbol{\lambda}\end{bmatrix} = \begin{bmatrix}\mathbf{b}\\ \mathbf{0}\\ \mathbf{d}\end{bmatrix}$$

and the corresponding reduced system is given by

$$\mathcal{A}\begin{bmatrix}\mathbf{y}\\ \boldsymbol{\lambda}\end{bmatrix} \equiv \begin{bmatrix} M & F^T\\ F & -\frac{1}{\beta}M\end{bmatrix}\begin{bmatrix}\mathbf{y}\\ \boldsymbol{\lambda}\end{bmatrix} = \begin{bmatrix}\mathbf{b}\\ \mathbf{d}\end{bmatrix}$$

The scheme leads to a $F - F^T$ system with an optimal error convergence order.

The arising matrices have a very rich structure.

Saddle-point systems naturally arise

The finite element discretization plus KKT conditions lead to a saddle-point system

$$\mathcal{A}x = \begin{bmatrix} A & B_1 \\ B_2 & -C \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}$$

where $f \in \mathbb{R}^n$, $g \in \mathbb{R}^m$, $A \in \mathbb{R}^{n\times n}$, B_1, $B_2^T \in \mathbb{R}^{m\times n}$, $C \in \mathbb{R}^{m\times m}$, $m \leq n$.
$\mathcal{A}$ is large, sparse and indefinite.

Thus, to solve $\mathcal{A}$ efficiently ... Krylov iterative methods.... preconditioning ...

General preconditioners for saddle point matrices

Recall the general form of a saddle point matrix:

$$\mathcal{A} = \begin{bmatrix} A & B_1 \\ B_2 & -C \end{bmatrix}.$$

A preconditioner can have a block-diagonal or a block lower-triangular structure, i.e.,

$$\mathcal{P}_{bd} = \begin{bmatrix} A & 0 \\ 0 & -S \end{bmatrix}, \quad \mathcal{P}_{bt} = \begin{bmatrix} A & 0 \\ B_2 & -S \end{bmatrix}, \quad \mathcal{P}_{bt} = \begin{bmatrix} [A] & 0 \\ B_2 & -[S] \end{bmatrix}.$$

Here S is the (negative) Schur complements of $\mathcal{A}$,

$$S = C + B_2 A^{-1} B_1.$$

Iterative solvers and preconditioning

Preconditioned Krylov subspace methods: MINRES, (F)GMRES
Desirable properties: parameter-independent solvers (h, β, $\mathbf{w}$ etc.)

- Techniques used to construct preconditioners for saddle point systems arising from distributed optimal control problems:
  - Schur complement approximation
  - Operator preconditioning with standard and non-standard norms
  - Structure-utilizing factorization.

Preconditioners: Poisson equation (full system)

- Block-diagonal preconditioner

$$\widehat{\mathcal{P}}_{bd} = \begin{bmatrix} \widehat{M} & 0 & 0 \\ 0 & \beta\widehat{M} & 0 \\ 0 & 0 & (K + \frac{1}{\sqrt{\beta}}M)M^{-1}(K + \frac{1}{\sqrt{\beta}}M)^T \end{bmatrix}.$$

- Lower block-triangular preconditioner

$$\widehat{\mathcal{P}}_{lbt} = \begin{bmatrix} \widehat{M} & 0 & 0 \\ 0 & \beta\widehat{M} & 0 \\ K & -M & -\{(K + \frac{1}{\sqrt{\beta}}M)M^{-1}(K + \frac{1}{\sqrt{\beta}}M)^T\} \end{bmatrix}.$$

Preconditioners: Poisson equation (reduced)

- Block-diagonal preconditioner

$$\widehat{\mathcal{P}}_{bd_1} = \begin{bmatrix} \widehat{M} & 0 \\ 0 & (K + \frac{1}{\sqrt{\beta}}M)M^{-1}(K + \frac{1}{\sqrt{\beta}}M)^T \end{bmatrix}$$

$$eigs\left(((K + \frac{1}{\sqrt{\beta}}M)M^{-1}(K + \frac{1}{\sqrt{\beta}}M)^T)^{-1}S\right) \in [0.5,\, 1].$$

- Block-diagonal preconditioner, nonstandard norms

$$\widehat{\mathcal{P}}_{bd_2} = \begin{bmatrix} M + \sqrt{\beta}K & 0 \\ 0 & \frac{1}{\beta}(M + \sqrt{\beta}K) \end{bmatrix}, \quad \varkappa(\widehat{\mathcal{P}}_{bd_2}^{-1}\mathcal{A}) \leq \sqrt{2}.$$

Preconditioners: Poisson equation (reduced)

Structure-utilizing technique

$$\mathcal{A} = \begin{bmatrix} M & -\beta K^T \\ \alpha K & M \end{bmatrix}.$$

$$\mathcal{P}_{UU} = \begin{bmatrix} M & -\beta K^T \\ \alpha K & M + \sqrt{\alpha\beta}(K + K^T) \end{bmatrix}.$$

$$eigs(\mathcal{P}_{UU}^{-1}\mathcal{A}) \in [0.5, 1]$$

M, pos. def., $K + K^T$ pos. semi-definite, $ker(M) \cup ker(K) = \{0\}$, $ker(M) \cup ker(K^T) = \{0\}$

Preconditioners: Poisson equation (reduced)

- An efficient algorithm to solve

$$\mathcal{P}_{UU}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} M & -\beta K \\ \alpha K & M + 2\sqrt{\alpha\beta}K \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}.$$

based on the exact form of the inverse of $\mathcal{P}_{UU}$

Let $H_i = M + \sqrt{\alpha\beta}\,K_i$, $i = 1, 2$ be nonsingular. Then

$$\mathcal{P}_{UU}^{-1} = \begin{bmatrix} H_1^{-1} + H_2^{-1} - H_2^{-1}MH_1^{-1} & \sqrt{\frac{\beta}{\alpha}}\,(I - H_2^{-1}M)H_1^{-1} \\ -\sqrt{\frac{\alpha}{\beta}}\,H_2^{-1}(I - MH_1^{-1}) & H_2^{-1}MH_1^{-1} \end{bmatrix}.$$

Efficient algorithms for the action of $\mathcal{P}_{UU}^{-1}$

Algorithm: The action of $\mathcal{P}_{UU}^{-1}$ on a vector

1: *Compute* $b_1 = \frac{\sqrt{\alpha}}{\sqrt{\beta}} f + g$

2: *Solve* $(M + \sqrt{\alpha\beta} K_1) s_1 = b_1$

3: *Compute* $b_2 = M s_1 - \frac{\sqrt{\alpha}}{\sqrt{\beta}} f$

4: *Solve* $(M + \sqrt{\alpha\beta} K_2) y = b_2$

5: *Compute* $x = \frac{\sqrt{\beta}}{\sqrt{\alpha}} (s_1 - y)$

Preconditioners: conv.-diff.

$$\begin{bmatrix} \widehat{M} & 0 & 0 \\ 0 & \beta\widehat{M} & 0 \\ 0 & 0 & (F + \frac{1}{\sqrt{\beta}}M)M^{-1}(F + \frac{1}{\sqrt{\beta}}M)^T \end{bmatrix}$$

BIT

Preconditioners: conv.-diff.

$$\begin{bmatrix} \widehat{M} & 0 & 0 \\ 0 & \beta\widehat{M} & 0 \\ 0 & 0 & (F + \frac{1}{\sqrt{\beta}} \end{bmatrix}$$

$$\begin{bmatrix} \widehat{M} & 0 & 0 \\ 0 & \beta\widehat{M} & 0 \\ F & -M & -\{(F + \frac{1}{\sqrt{\beta}}M)M^{-1}(F + \frac{1}{\sqrt{\beta}}M)^T\} \end{bmatrix}$$

Preconditioners: conv.-diff.

$$\begin{bmatrix} \widehat{M} & 0 & 0 \\ 0 & \beta\widehat{M} & 0 \\ 0 & 0 & (F + \frac{1}{\sqrt{\beta}} \end{bmatrix}$$

$$\begin{bmatrix} \widehat{M} & 0 & 0 \\ 0 & \beta\widehat{M} & 0 \\ F & -M & -\{(F + \frac{1}{\sqrt{\beta}}M)M^{-1}(F + \frac{1}{\sqrt{\beta}}M)^T\} \end{bmatrix}$$

$$\begin{bmatrix} \widehat{M} & 0 \\ 0 & (F + \frac{1}{\sqrt{\beta}}M)M^{-1}(F + \frac{1}{\sqrt{\beta}}M)^T \end{bmatrix}$$

BIT

Preconditioners: conv.-diff.

$$\begin{bmatrix} \widehat{M} & 0 & 0 \\ 0 & \beta\widehat{M} & 0 \\ 0 & 0 & (F + \frac{1}{\sqrt{\beta}} \cdots \end{bmatrix}$$

$$\begin{bmatrix} \widehat{M} & 0 & 0 \\ 0 & \beta\widehat{M} & 0 \\ F & -M & -\{(F + \frac{1}{\sqrt{\beta}}M)M^{-1}(F + \frac{1}{\sqrt{\beta}}M)^T\} \end{bmatrix}$$

$$\begin{bmatrix} \widehat{M} & 0 \\ 0 & (F + \frac{1}{\sqrt{\beta}}M)M^{-1}(F + \frac{1}{\sqrt{\beta}}M)^T \end{bmatrix}$$

$$\begin{bmatrix} M + \sqrt{\beta}F & 0 \\ 0 & \frac{1}{\beta}(M + \sqrt{\beta}F) \end{bmatrix}$$

Preconditioners: conv.-diff.

$$\begin{bmatrix} \widehat{M} & 0 & 0 \\ 0 & \beta\widehat{M} & 0 \\ 0 & 0 & (F + \frac{1}{\sqrt{\beta}} \end{bmatrix}$$

$$\begin{bmatrix} \widehat{M} & 0 & 0 \\ 0 & \beta\widehat{M} & 0 \\ F & -M & -\{(F + \frac{1}{\sqrt{\beta}}M)M^{-1}(F + \frac{1}{\sqrt{\beta}}M)^T\} \end{bmatrix}$$

$$\begin{bmatrix} \widehat{M} & 0 \\ 0 & (F + \frac{1}{\sqrt{\beta}}M)M^{-1}(F + \frac{1}{\sqrt{\beta}}M)^T \end{bmatrix}$$

$$\begin{bmatrix} M + \sqrt{\beta}F & 0 \\ 0 & \dfrac{1}{\beta}(M + \sqrt{\beta}F) \end{bmatrix}$$

$$\begin{bmatrix} M & -\beta F^T \\ F & M + \sqrt{\beta}(F + F^T) \end{bmatrix}$$

BIT

Software and solvers used:

- All preconditioners are tested and compared within the same environment:
  - C++ implementation using the open source package DEAL.II and
  - open source libraries such as Trillinos.

To our best of knowledge such comparisons have not been performed yet.

Software and solvers used:

- Solutions with M
  replaced by 20 Chebyshev semi-iterations.
- Solutions with K, $M + \sqrt{\beta}K$, $(K + \frac{1}{\sqrt{\beta}}M)$, $M + \sqrt{\beta}F$, $(F + \frac{1}{\sqrt{\beta}}M)$
  replaced by 1 iteration of V-cycle Algebric Multigrid (AMG) solver with 2 pre-smoothing and 2 post-smoothing steps by symmetric Gauss-Seidel smoother.

Poisson equation:

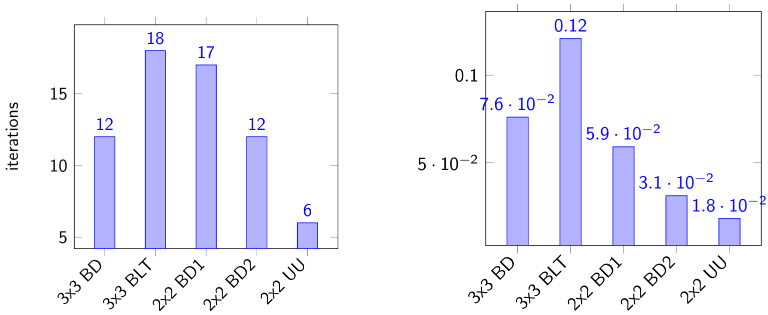


Figure : Mesh size $h = 2^{-6}$ and $\beta = 10^{-6}$, comparison across different preconditioners.

Convection-diffusion:

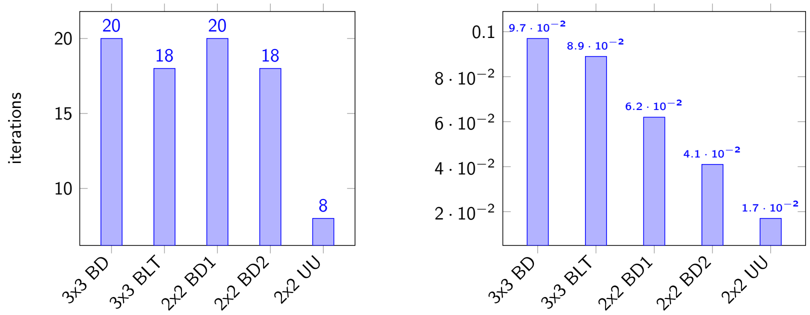


Figure : Mesh size $h = 2^{-6}$ and $\beta = 10^{-6}$

BIT

Outcome

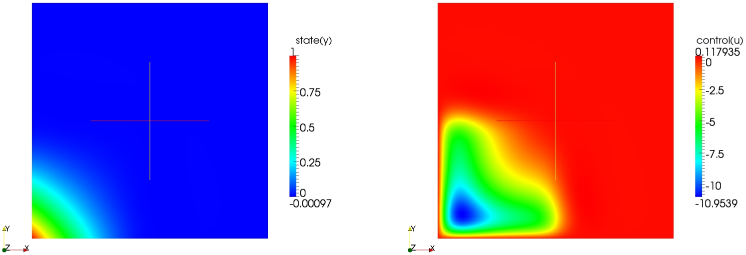
- Reducing the discrete system when such a possibility is available leads to better performance both in terms of computational time and the iteration count.
- Full utilization of the structure helps.
- When reporting numerical experiments, think about:

reproducibility of the numerical results and
fair comparison with other methods.
This will make the paper very useful also for applied scientists and practitioners, that we hope to reach with our work.

- The UU framework has been tuned also for full Stokes control problem, showing analogous behaviour.

Thank you for your attention!

Example: Temperature control



(a) $\mathbf{y}, \beta = 2 \times 10^{-6}$ (b) $\mathbf{u}, \beta = 2 \times 10^{-6}$

Figure : State (y) (temperature) and control (u) (heat) distribution