INCOMPLETE BLOCK $\mathbf{LD^{-1}L^T}$ FACTORIZATION: Constraint preconditioners

S. Lungten

Joint work with W.H.A. Schilders and J.M.L. Maubach.

Center for Analysis, Scientific Computing and Applications

Technische Universiteit
Eindhoven
University of Technology

International conference on preconditioning techniques for scientific and industrial applications 17 - 19 June 2015

Introduction and motivation

We focus at large and sparse $(n+m) \times (n+m)$ saddle point systems:

$$\underbrace{\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{\mathcal{A}}} \underbrace{\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix}}_{\boldsymbol{u}} = \underbrace{\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{g} \end{bmatrix}}_{\boldsymbol{d}}$$

where

- $\boldsymbol{A}$ is $n \times n$ symmetric positive definite (SPD)
- $\boldsymbol{B}$ is $m \times n$ matrix of full rank with $m < n$

Introduction and motivation (cont.)

Solution methods

- **Direct solvers** based on factorization of $\mathcal{A}$
  – time, memory, stability issues
- **Iterative solvers** based on Krylov subspace methods
  – lack robustness, convergence issue, esp. when $\mathcal{A}$ is ill-conditioned
- **Preconditioning techniques**
  – focus of this presentation

See e.g. Numerical solution of saddle point problems (Benzi, Golub and Liesen, 2005) and many others.

Introduction and motivation (cont.)

Preconditioning techniques

We focus at preconditioners of the form:

$$\boldsymbol{\mathcal{G}} = \begin{bmatrix} \boldsymbol{G} & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix}$$

where $\boldsymbol{G}$ approximates $\boldsymbol{A}$ and $\boldsymbol{G} \neq \boldsymbol{A}$

An attractive feature

Keller, Gould and Wathen (2000) showed that $\boldsymbol{\mathcal{G}}^{-1}\boldsymbol{\mathcal{A}}$ has

- an eigenvalue at 1 with multiplicity $2m$
- $n - m$ eigenvalues defined by $\boldsymbol{Z}^T\boldsymbol{A}\boldsymbol{Z}\boldsymbol{x}_{\mathsf{z}} = \lambda \boldsymbol{Z}^T\boldsymbol{G}\boldsymbol{Z}\boldsymbol{x}_{\mathsf{z}}$

Introduction and motivation (cont.)

- **Complexity: factorizing $\mathcal{G} \sim$ factorizing $\mathcal{A}$**
- **Alternative: $\mathcal{G}$ is formed implicitly**

Previous works on implicit form

- Incomplete Schilders' factorization with blocks of orders m and $n - m$ (Dollar and Wathen, 2006)

- Incomplete block factorization with blocks of orders 1 and 2 with only diagonal update (Schilders, 2009)

Current work

- Extend the idea of incomplete block factorization with blocks of orders 1 and 2 (Schilders) to more general form using fill-in strategies

It relies on the existence of SPD Schur complement reductions

- Postulate a stable incomplete Schur complement reductions of SPD matrices

Method

1. **Transformation** of $\mathcal{A}$ by transforming $\boldsymbol{B}$ into upper trapezoidal form:

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{B}_2 \end{bmatrix}$$

where

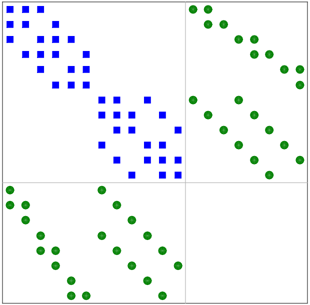
- $\boldsymbol{B}_1$ is $m \times m$ nonsingular upper triangular
- $\boldsymbol{B}_2$ is $m \times (n-m)$ remaining part

Obtain through:

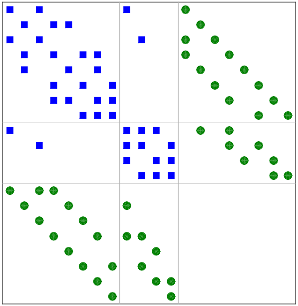
- permutation– special $\boldsymbol{B}$ (gradient, incidence matrices)
- sparse QR– general $\boldsymbol{B}$

Method (cont.)

Example: Transformation of 20×20 ($n = 12$, $m = 8$) $\mathcal{A}$ by permuting $\boldsymbol{B}$ into upper trapezoidal form



Original $\mathcal{A}$



Transformed $\mathcal{A}$

Method (cont.)

2 **Partitioning** into blocks of orders 1 and 2 such that

$$\boldsymbol{\pi}^T \boldsymbol{\mathcal{A}} \boldsymbol{\pi} = \left[\begin{array}{c|c} 2 \times 2 \text{ blocks} & 2 \times 1 \text{ blocks} \\ \hline 1 \times 2 \text{ blocks} & 1 \times 1 \text{ blocks} \end{array}\right]$$

where $\boldsymbol{\pi}$ is a permutation matrix of order $n + m$

3 **Block factorization**

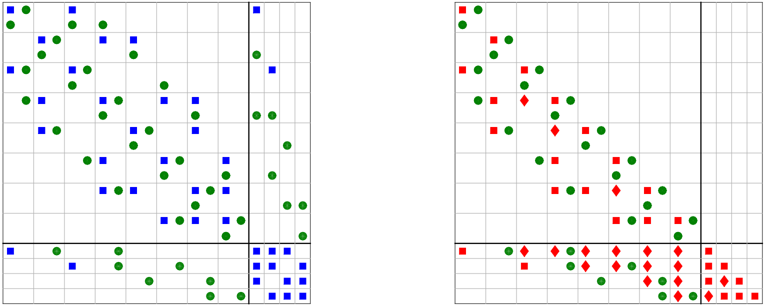
Exact: $\boldsymbol{\pi}^T \boldsymbol{\mathcal{A}} \boldsymbol{\pi} = \boldsymbol{L} \boldsymbol{D}^{-1} \boldsymbol{L}^T$

where

- $\boldsymbol{L}$ is block lower triangular with blocks of orders 1 and 2
- $\boldsymbol{D} = \text{diag}(\boldsymbol{L})$ is block diagonal part

Method (cont.)

Example: Factorization of n-by-n block $\boldsymbol{\pi}^T \boldsymbol{\mathcal{A}} \boldsymbol{\pi}$ with $n = 12$, $m = 8$



Partitioned $\boldsymbol{\pi}^T \boldsymbol{\mathcal{A}} \boldsymbol{\pi}$ — Exact $\boldsymbol{L}$

■ (blue) nonzero element of $\boldsymbol{A}$

● (green) nonzero element of $\boldsymbol{B}$

■ (red) updated nonzero element of $\boldsymbol{A}$

◆ (red) fill-in related to zero element of $\boldsymbol{A}$

Method (cont.)

Incomplete: $\boldsymbol{\pi}^T \boldsymbol{\mathcal{A}} \boldsymbol{\pi} = \hat{\boldsymbol{L}} \hat{\boldsymbol{D}}^{-1} \hat{\boldsymbol{L}}^T + \boldsymbol{E}$

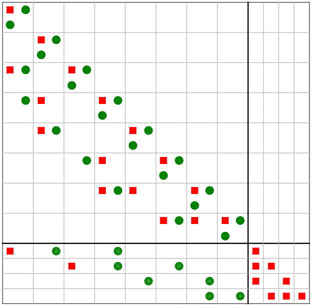
where

- $\hat{\boldsymbol{L}} \sim \boldsymbol{L}$
- $\mathrm{diag}(\hat{\boldsymbol{L}}) = \hat{\boldsymbol{D}} \sim \boldsymbol{D}$
- $\boldsymbol{E}$ is an error matrix

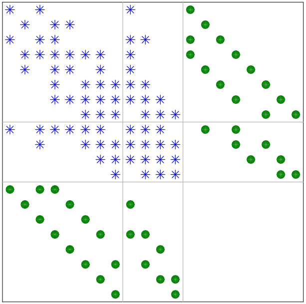
4 **Reconstruction** $\boldsymbol{\pi} \left(\hat{\boldsymbol{L}} \hat{\boldsymbol{D}}^{-1} \hat{\boldsymbol{L}}^T \right) \boldsymbol{\pi}^T$ forms a constraint preconditioner $\boldsymbol{\mathcal{G}}$
– required only for the proof of existence

Method (cont.)

Example: Set the $(i,j)^{\text{th}}$ element of $\hat{\boldsymbol{L}}$ to zero whenever $(i,\ j) \notin \mathbb{S} = \{(i,\ j):\ a_{ij} \neq 0\}$



Incomplete $\hat{\boldsymbol{L}}$



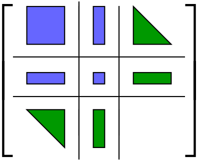
$$\mathcal{G} = \boldsymbol{\pi}\left(\hat{\boldsymbol{L}}\hat{\boldsymbol{D}}^{-1}\hat{\boldsymbol{L}}^{T}\right)\boldsymbol{\pi}^{T}$$

S. Lungten, W.H.A. Schilders, J.M.L. Maubach

Existence of block factorization

Assume the transformed saddle point matrix [Schilders, 2009]

$$\boldsymbol{\mathcal{A}} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} & \boldsymbol{B}_1^T \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} & \boldsymbol{B}_2^T \\ \boldsymbol{B}_1 & \boldsymbol{B}_2 & \boldsymbol{0} \end{bmatrix} =$$



and the permutation matrix $\boldsymbol{\pi}$ of order $n+m$ defined by

$$\boldsymbol{\pi} = \left[\boldsymbol{e}_1, \quad \boldsymbol{e}_{n+1}, \quad \boldsymbol{e}_2, \quad \cdots, \quad \boldsymbol{e}_m, \quad \boldsymbol{e}_{n+m}, \quad \boldsymbol{e}_{m+1}, \quad \cdots, \quad \boldsymbol{e}_n\right]$$

where

$\boldsymbol{e}_i$ is the i^{th} unit vector of length $n+m$

Existence of block factorization (cont.)

Let

$$\mathcal{A} = \left[\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B}^T \\ \hline \boldsymbol{B} & \boldsymbol{0} \end{array}\right] = \left[\begin{array}{cc|cc} a_{11} & \boldsymbol{a}^T & b_{11} & \boldsymbol{b}_{\mathsf{c}}^T \\ \boldsymbol{a} & \boldsymbol{A}_{\mathsf{s}} & \boldsymbol{b}_{\mathsf{r}} & \boldsymbol{B}_{\mathsf{s}}^T \\ \hline b_{11} & \boldsymbol{b}_{\mathsf{r}}^T & 0 & \boldsymbol{0} \\ \boldsymbol{b}_{\mathsf{c}} & \boldsymbol{B}_{\mathsf{s}} & \boldsymbol{0} & \boldsymbol{0} \end{array}\right]$$

where

- $\boldsymbol{A}_{\mathsf{s}}$ is $(n-1) \times (n-1)$ principal submatrix
- $\boldsymbol{B}_{\mathsf{s}}$ is $(m-1) \times (n-1)$ submatrix
- $\boldsymbol{b}_{\mathsf{c}} = \boldsymbol{0}$

Existence of block factorization (cont.)

Applying $\boldsymbol{\pi}_1$ to $\boldsymbol{\mathcal{A}}$ gives

$$\boldsymbol{\pi}_1^T \boldsymbol{\mathcal{A}} \boldsymbol{\pi}_1 = \boldsymbol{\pi}_1^{(1)^T} \boldsymbol{\mathcal{A}}^{(1)} \boldsymbol{\pi}_1^{(1)} = \left[\begin{array}{cc|cc} a_{11}^{(1)} & b_{11}^{(1)} & \boldsymbol{a}^{(1)^T} & \mathbf{0} \\ b_{11}^{(1)} & 0 & \boldsymbol{b}_{\mathsf{r}}^{(1)^T} & \mathbf{0} \\ \hline \boldsymbol{a}^{(1)} & \boldsymbol{b}_{\mathsf{r}}^{(1)} & \boldsymbol{A}_{\mathsf{s}}^{(1)} & \boldsymbol{B}_{\mathsf{s}}^{(1)^T} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{B}_{\mathsf{s}}^{(1)} & \mathbf{0} \end{array} \right]$$

where

- $\boldsymbol{A}^{(1)} = \boldsymbol{A}$
- $\boldsymbol{B}^{(1)} = \boldsymbol{B}$

Existence of block factorization (cont.)

Applying $\boldsymbol{\pi}_1$ to $\boldsymbol{\mathcal{A}}$ gives

$$\boldsymbol{\pi}_1^T \boldsymbol{\mathcal{A}} \boldsymbol{\pi}_1 = \boldsymbol{\pi}_1^{(1)^T} \boldsymbol{\mathcal{A}}^{(1)} \boldsymbol{\pi}_1^{(1)} = \left[\begin{array}{cc|cc} a_{11}^{(1)} & b_{11}^{(1)} & \boldsymbol{a}^{(1)^T} & \mathbf{0} \\ b_{11}^{(1)} & 0 & \boldsymbol{b}_{\mathsf{r}}^{(1)^T} & \mathbf{0} \\ \hline \boldsymbol{a}^{(1)} & \boldsymbol{b}_{\mathsf{r}}^{(1)} & \boldsymbol{A}_{\mathsf{s}}^{(1)} & \boldsymbol{B}_{\mathsf{s}}^{(1)^T} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{B}_{\mathsf{s}}^{(1)} & \mathbf{0} \end{array}\right]$$

where

- $\boldsymbol{A}^{(1)} = \boldsymbol{A}$
- $\boldsymbol{B}^{(1)} = \boldsymbol{B}$

1^{st} block-column of $\boldsymbol{L}$ = 1^{st} block-column of $\boldsymbol{\pi}_1^{(1)^T} \boldsymbol{\mathcal{A}}^{(1)} \boldsymbol{\pi}_1^{(1)}$

Existence of block factorization (cont.)

Schur complement of the 1^{st} 2×2 block yields:

$$\boldsymbol{A}^{(2)} = \boldsymbol{A}_{\mathsf{s}}^{(1)} - \frac{1}{b_{11}^{(1)^2}} \begin{bmatrix} \boldsymbol{a}^{(1)} & \boldsymbol{b}_{\mathsf{r}}^{(1)} \end{bmatrix} \begin{bmatrix} 0 & b_{11}^{(1)} \\ b_{11}^{(1)} & -a_{11}^{(1)} \end{bmatrix} \begin{bmatrix} \boldsymbol{a}^{(1)^T} \\ \boldsymbol{b}_{\mathsf{r}}^{(1)^T} \end{bmatrix}$$

$$\boldsymbol{B}^{(2)} = \boldsymbol{B}_{\mathsf{s}}^{(1)}$$

By induction, it can be shown that:

- $\boldsymbol{A}^{(k+1)}$ is SPD for $k = 1, \ldots, n-1$
- $\boldsymbol{B}^{(k+1)} = \boldsymbol{B}_{\mathsf{s}}^{(k)}$ for $k = 1, \ldots, m-1$

eventually leading to **exact** $\boldsymbol{\pi}^T \boldsymbol{\mathcal{A}} \boldsymbol{\pi} = \boldsymbol{L}\boldsymbol{D}^{-1}\boldsymbol{L}^T$

Existence of block factorization (cont.)

Incomplete form

- ❐ Updates or fill-ins occur only in $\boldsymbol{A}^{(k)}$

Considering a fill-in strategy based on sparsity of $\boldsymbol{A}$, we obtain

$$\boldsymbol{\pi}^T \boldsymbol{\mathcal{A}} \boldsymbol{\pi} = \hat{\boldsymbol{L}} \hat{\boldsymbol{D}}^{-1} \hat{\boldsymbol{L}}^T + \boldsymbol{E}$$

Back permutation yields:

$$\boldsymbol{\mathcal{A}} = \boldsymbol{\pi} \left(\hat{\boldsymbol{L}} \hat{\boldsymbol{D}}^{-1} \hat{\boldsymbol{L}}^T \right) \boldsymbol{\pi}^T + \boldsymbol{\pi} \boldsymbol{E} \boldsymbol{\pi}^T$$

$$\Rightarrow \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{G} & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{E}_{\mathsf{A}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$$

where $\boldsymbol{G} \sim \boldsymbol{A}$ and $\boldsymbol{E}_{\mathsf{A}}$ contains all the nonzeros of $\boldsymbol{E}$

- ❐ $\hat{L}$ is nonsingular only if incomplete forms, $\hat{\boldsymbol{A}}^{(k)}$ exist

Incomplete SCR of SPD matrices

Incomplete Schur complement reductions (SCR) of SPD matrices

Background

- **ICCG** for M-matrices (Meijerink and van der Vorst, 1977)
- **MILU** for matrices related to elliptic PDE problems (Gustafsson, 1978)
- **Incomplete shifted Cholesky factorization** for SPD matrices (Manteuffel, 1980)

Incomplete SCR of SPD matrices (cont.)

Let an $n \times n$ SPD matrix:

$$\boldsymbol{A} = \boldsymbol{A}^{(1)} = \left[\begin{array}{c|c} a_{11}^{(1)} & \boldsymbol{a}^{(1)^T} \\ \hline \boldsymbol{a}^{(1)} & \boldsymbol{A}_{\mathsf{s}}^{(1)} \end{array}\right]$$

where $\boldsymbol{A}_{\mathsf{s}}^{(1)}$ is principal submatrix of order $n-1$, then

$$\boldsymbol{A}^{(2)} = \boldsymbol{A}_{\mathsf{s}}^{(1)} - \frac{1}{a_{11}^{(1)}} \boldsymbol{a}^{(1)} \boldsymbol{a}^{(1)^T} \text{ is SPD.}$$

Incomplete SCR of SPD matrices (cont.)

Suppose, $\boldsymbol{A}^{(2)}$ is approximated by $\tilde{\boldsymbol{A}}^{(2)}$ based on $\mathbb{S} = \{(i,\ j) :\ a_{ij} \neq 0\}$.

Let $\boldsymbol{F}^{(1)} = \boldsymbol{a}^{(1)}\boldsymbol{a}^{(1)^T}$ (exact) and split:

$$\boldsymbol{F}^{(1)} = \underbrace{\boldsymbol{F}^{(1)}_{\text{in}}}_{\text{incomplete}} + \underbrace{\boldsymbol{F}^{(1)}_{\text{out}}}_{\text{dropped}} .$$

Then

$$\tilde{\boldsymbol{A}}^{(2)} = \boldsymbol{A}^{(1)}_{\text{s}} - \frac{1}{a^{(1)}_{11}} \boldsymbol{F}^{(1)}_{\text{in}}$$

or

$$\tilde{\boldsymbol{A}}^{(2)} = \boldsymbol{A}^{(2)} + \frac{1}{a^{(1)}_{11}} \boldsymbol{F}^{(1)}_{\text{out}}$$

which may turn out to be **indefinite** since $\boldsymbol{F}^{(1)}_{\text{out}}$ has **zero diagonal** and is **indefinite.**

Incomplete SCR of SPD matrices (cont.)

Choosing the **lumped semi-definite diagonal**

$$\boldsymbol{D}_{\text{add}}^{(1)}(i,i) = \sum_{j=1}^{n-1} \left| \boldsymbol{F}_{\text{out}}^{(1)}(i,j) \right|, \; i = 1, \ldots, n-1$$

and setting

$$\hat{\boldsymbol{A}}^{(2)} = \tilde{\boldsymbol{A}}^{(2)} + \frac{1}{a_{11}^{(1)}} \boldsymbol{D}_{\text{add}}^{(1)}$$

gives an SPD approximation of $\boldsymbol{A}^{(2)}$, since

$$\hat{\boldsymbol{A}}^{(2)} = \underbrace{\boldsymbol{A}^{(2)}}_{\text{SPD}} + \frac{1}{a_{11}^{(1)}} \underbrace{\left(\boldsymbol{F}_{\text{out}}^{(1)} + \boldsymbol{D}_{\text{add}}^{(1)} \right)}_{\text{symmetric semi-definite}} .$$

Incomplete SCR of SPD matrices (cont.)

By induction, we obtain

$$\boldsymbol{A} = \hat{\boldsymbol{L}}_{\mathsf{A}} \hat{\boldsymbol{D}}_{\mathsf{A}}^{-1} \hat{\boldsymbol{L}}_{\mathsf{A}}^{T} + \boldsymbol{R}$$

for SPD matrices, where $\hat{\boldsymbol{D}}_{\mathsf{A}} = \text{diag}(\hat{\boldsymbol{L}}_{\mathsf{A}}) > \boldsymbol{0}$ and $\boldsymbol{R}$ is an error matrix due to $\boldsymbol{F}_{\text{out}}^{(k)}$.

- Refer to it as **lump modified incomplete Cholesky (LMIC)**

Consequently, we obtain $\boldsymbol{\pi}^T \boldsymbol{\mathcal{A}} \boldsymbol{\pi} = \hat{\boldsymbol{L}} \hat{\boldsymbol{D}}^{-1} \hat{\boldsymbol{L}}^T + \boldsymbol{E}$ for saddle point matrices based on LMIC.

- Refer to it as **lump modified incomplete block Cholesky (LMIBC)**

Incomplete SCR of SPD matrices (cont.)

How different is LMIC from MILU ?

- MILU adds the row elements of dropped matrix to preserve the row sums
- LMIC adds the absolute row elements of dropped matrix to preserve SPD property

Example

For the SPD matrix

$$\boldsymbol{A} = \begin{bmatrix} 4 & -1 & -1 & 0 & 0 \\ -1 & 2 & 0 & -1 & 0 \\ -1 & 0 & 2 & -1 & -1 \\ 0 & -1 & -1 & 2 & 0 \\ 0 & 0 & -1 & 0 & 3 \end{bmatrix},$$

MILU gives $\hat{\boldsymbol{L}}(4,4) = 0$ while LMIC gives $\hat{\boldsymbol{L}}(4,4) = 1.5$

Implementation

Preconditioned conjugate gradient (PCG) method

- **Projected PCG method for the reduced system**
  related to the null space method (Gould, Hribar and Nocedal, 2001)
  – approximates $\boldsymbol{x}$, while $\boldsymbol{y}$ is solved from the equation $\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{y} = \boldsymbol{B}\left(\boldsymbol{f} - \boldsymbol{A}\boldsymbol{x}\right)$ (appears in null space method)

- **PCG method for the full system**
  – approximates $\boldsymbol{u} = \begin{bmatrix}\boldsymbol{x}^T & \boldsymbol{y}^T\end{bmatrix}^T$ in $\boldsymbol{u}_0 + \boldsymbol{\mathcal{K}}_N(\boldsymbol{\mathcal{G}}^{-1}\boldsymbol{\mathcal{A}},\ \boldsymbol{s})$, where $\boldsymbol{\mathcal{K}}_N$ is the N^{th} Krylov subspace generated by
  - $\boldsymbol{\mathcal{G}}^{-1}\boldsymbol{\mathcal{A}}$ and
  - preconditioned initial residual $\boldsymbol{s} = \boldsymbol{\mathcal{G}}^{-1}\boldsymbol{r}_0$, $\boldsymbol{r}_0 = \boldsymbol{d} - \boldsymbol{\mathcal{A}}\boldsymbol{u}_0$ for an initial guess $\boldsymbol{u}_0$

Implementation (cont.)

Necessary conditions (in both cases)

C1 Initial $\boldsymbol{u}_0 = \begin{bmatrix}\boldsymbol{x}^T & \boldsymbol{y}^T\end{bmatrix}^T$ is chosen such that $\boldsymbol{x}$ satisfies $\boldsymbol{B}\boldsymbol{x} = \boldsymbol{g}$.

C2 $\boldsymbol{\mathcal{G}}$ is chosen such that $(\boldsymbol{G}\boldsymbol{v},\ \boldsymbol{v}) \neq \boldsymbol{0}$ whenever $\boldsymbol{B}\boldsymbol{v} = \boldsymbol{0}$ for $\boldsymbol{0} \neq \boldsymbol{v} \in \mathbb{R}^n$.

Require to solve the preconditioned augmented system

$$\begin{bmatrix}\boldsymbol{G} & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{s}_1 \\ \boldsymbol{s}_2\end{bmatrix} = \begin{bmatrix}\boldsymbol{r}_1 \\ \boldsymbol{r}_2\end{bmatrix}$$

where $\boldsymbol{r}_0 = \begin{bmatrix}\boldsymbol{r}_1^T & \boldsymbol{r}_2^T\end{bmatrix}^T$ is the initial residual.

Condition C1 implies $\boldsymbol{r}_2 = \boldsymbol{0}$ and $(\cdot\,,\,\cdot)_{\boldsymbol{\mathcal{A}}} = (\cdot\,,\,\cdot)_{\boldsymbol{A}}$

Implementation (cont.)

Algorithm (PCG method for the full system)

1: *Choose an initial point* $\boldsymbol{x}$ *such that* $\boldsymbol{B}\boldsymbol{x} = \boldsymbol{g}$, *choose arbitrary* $\boldsymbol{y}$, *compute* $\boldsymbol{r}_1 = \boldsymbol{f} - \left(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}^T\boldsymbol{y}\right)$, *solve for* $\boldsymbol{s}_1$, $\boldsymbol{s}_2$ *and set* $\boldsymbol{p}_1 = \boldsymbol{s}_1$, $\boldsymbol{p}_2 = \boldsymbol{s}_2$

2: ***repeat***

3: $\alpha = (\boldsymbol{r}_1,\ \boldsymbol{s}_1) \,/\, (\boldsymbol{A}\boldsymbol{p}_1,\ \boldsymbol{p}_1)$

4: $\boldsymbol{x} \leftarrow \boldsymbol{x} + \alpha\boldsymbol{p}_1, \quad \boldsymbol{y} \leftarrow \boldsymbol{y} + \alpha\boldsymbol{p}_2$

5: $\boldsymbol{r}_1^+ \leftarrow \boldsymbol{r}_1 - \alpha\left(\boldsymbol{A}\boldsymbol{p}_1 + \boldsymbol{B}^T\boldsymbol{p}_2\right)$

6: *Solve for* $\boldsymbol{s}_1^+$, $\boldsymbol{s}_2^+$

7: $\beta = \left(\boldsymbol{r}_1^+,\ \boldsymbol{s}_1^+\right) \,/\, (\boldsymbol{r}_1,\ \boldsymbol{s}_1)$

8: $\boldsymbol{p}_1 \leftarrow \boldsymbol{s}_1^+ + \beta\boldsymbol{p}_1, \quad \boldsymbol{p}_2 \leftarrow \boldsymbol{s}_2^+ + \beta\boldsymbol{p}_2$

9: $\boldsymbol{s}_1 \leftarrow \boldsymbol{s}_1^+,\ \boldsymbol{s}_2 \leftarrow \boldsymbol{s}_2^+, \quad \boldsymbol{r}_1 \leftarrow \boldsymbol{r}_1^+$

10: ***until*** *the convergence test is satisfied*

For every iteration, $Bx = g$ is satisfied giving $r_2^+ = \mathbf{0}$.

Numerical experiments 1

Comparisons between ILU(0) and LMIC by using PCG method

- SPD system $\boldsymbol{Ax} = \boldsymbol{b}$
- preconditioned from left $\boldsymbol{P}^{-1}\boldsymbol{Ax} = \boldsymbol{P}^{-1}\boldsymbol{b}$ where $\boldsymbol{P}$ is obtained by ILU(0) or LMIC
- $\boldsymbol{A}$-inner product

Test matrices

SPD matrices from UF sparse matrix collection

Numerical experiments 1 (cont.)

Matrix $\boldsymbol{A}$	Source/application	size(n)	nnz	
			$\boldsymbol{A}$	$\hat{\boldsymbol{L}}_{\mathsf{A}}$
bcsstk13	CFD	$2,003$	$83,883$	$42,943$
nasa2910	NASA, Langley	$2,910$	$174,296$	$88,603$
bcsstk21	Clamped square plate	$3,600$	$26,600$	$15,100$
msc04515	MSC Nastran	$4,515$	$97,707$	$51,111$
s2rmq4m1	FEM	$5,489$	$263,351$	$134,420$
bcsstk17	Pressure vessel	$10,974$	$428,650$	$219,812$
nd6k	3D problem	$18,000$	$6,897,316$	$3,457,658$
smt	Thermal stress	$25,710$	$3,749,582$	$1,887,646$

Table 1: SPD matrices from UF sparse matrix collection with the number of nonzero elements 'nnz'.

Numerical experiments 1 (cont.)

Matrix	ILU(0)			LMIC	
	Iter	IterTime	$\hat{L}_A(i,i) < 0$	Iter	IterTime
bcsstk13	–	–	79	1030	0.51
nasa2910	1,054	0.80	16	281	0.30
bcsstk21	–	–	129	290	0.11
msc04515	–	–	346	938	0.51
s2rmq4m1	221	0.28	2	704	0.86
bcsstk17	–	–	37	607	2.40
nd6k	724	16.84	9	490	10.60
smt	–	–	36	2,116	29.08

Table 2: Comparisons between ILU(0) and LMIC using PCG: tolerance 10^{-10}; maximum iterations n.

- MILU failed to converge even to a small residual tolerance 10^{-2} for all these matrices

Numerical experiments 2

Implementation of LMIBC using:

(i) Projected PCG method for the reduced system

(ii) PCG method for the full system

Test matrices

- SPD matrices from UF sparse matrix collection
- Stokes 3D problems generated on $d \times d \times d$ grid (Wubs)

Numerical experiments 2 (cont.)

Matrix	size		nnz		
	n	m	$\mathring{\boldsymbol{\mathcal{A}}}$	$\boldsymbol{\mathcal{A}}$	$\hat{\boldsymbol{L}}$
sit100	$7,142$	$3,120$	$61,036$	$61,036$	$37,207$
cvxqp3	$10,000$	$7,500$	$114,962$	$114,958$	$67,226$
tuma1	$13,360$	$9,607$	$87,760$	$87,760$	$60,167$
mario001	$23,130$	$15,304$	$204,912$	$204,912$	$129,325$
Stokes12	$6,084$	$2,196$	$64,032$	$64,032$	$37,254$
Stokes15	$11,520$	$4,095$	$122,298$	$122,298$	$71,004$
Stokes17	$16,524$	$5,831$	$176,142$	$176,142$	$102,164$

Table 3: Saddle point matrices from UF collection and Stokes 3D problems. Original $\mathring{\boldsymbol{\mathcal{A}}}$, transformed $\boldsymbol{\mathcal{A}}$ and incomplete factor $\hat{\boldsymbol{L}}$.

- Nonzeros: nnz($\hat{\boldsymbol{L}}$) = nnz(block 'tril'($\boldsymbol{\pi}^T \boldsymbol{\mathcal{A}} \boldsymbol{\pi}$)).

Numerical experiments 2 (cont.)

Matrix	Projected PCG (reduced system)				PCG (full system)	
	Iter	IterTime	$\boldsymbol{y}$Time	TotTime	Iter	IterTime
sit100	$1,654$	8.45	0.24	8.69	$1,857$	9.60
cvxqp3	$1,209$	18.62	0.87	19.49	$1,170$	12.50
tuma1	$1,334$	12.70	1.38	14.08	$1,422$	13.40
mario001	962	18.36	3.54	21.9	991	18.52
Stokes12	543	2.30	0.16	2.46	554	2.42
Stokes15	784	6.7	0.53	7.23	805	6.76
Stokes17	972	12.00	1.09	13.09	992	12.30

Table 4: LMIBC implemented using PCG method for reduced and full systems with tolerance 10^{-8} and maximum iterations 2000.

Conclusions and future work

- Incomplete block Cholesky like factorization (LMIBC) is proposed for constraint preconditioning techniques. Numerical tests using PCG method confirms its applicability. It depends on LMIC of SPD matrices.
- Observed no significant time difference of PCG method between the reduced and full systems.
- Implementation of LMIBC using other iterative methods such as GMRES is essential to be discussed in future to compare with the methods like ILUT (Saad, 1994) and ILUC (Li and Saad, 2005).
- For general form of SPD matrices, LMIC is more advantageous over ILU(0) and MILU. It could be extended to semi-definite matrices in future.

Thank you!