Krylov-based Schur Complement Solvers via PETSc

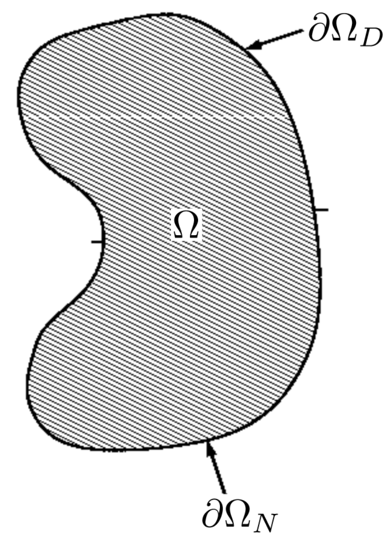
John W. Peterson

October 13, 2004

Problem Description

Consider the Stokes problem: find $\{\boldsymbol{u}, p\}$ s.t.

$$\begin{cases} -\Delta \boldsymbol{u} + \boldsymbol{\nabla} p = \boldsymbol{0} & \in \Omega \\ \boldsymbol{\nabla} \cdot \boldsymbol{u} = 0 & \in \Omega \\ \boldsymbol{u} = \boldsymbol{u_D} & \in \partial\Omega_D \\ \boldsymbol{\nabla u} \cdot \boldsymbol{n} = \boldsymbol{g}_N & \in \partial\Omega_N \end{cases}$$



Finite Element Formulation: find $\{\boldsymbol{u}_h, p_h\}$ s.t.

$$\int_\Omega (\boldsymbol{\nabla u}_h : \boldsymbol{\nabla v}_h - p_h \boldsymbol{\nabla} \cdot \boldsymbol{v}_h) dx = 0 \quad \forall\ \boldsymbol{v}_h \in \boldsymbol{V}_h \tag{1}$$

$$\int_\Omega (\boldsymbol{\nabla} \cdot \boldsymbol{u}_h) q dx = 0 \quad \forall\ q_h \in Q_h \tag{2}$$

Problem Definition

- Leads to a system of equations of the form

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \tag{3}$$

- A possible solution technique is the Schur complement solve: Given an initial guess for p,

$$u = A^{-1}(f - B^T p) \tag{4}$$

- Plugging into the second equation yields

$$BA^{-1}B^T p = BA^{-1}f - g \tag{5}$$

Algorithm

1. At step k

$$\begin{aligned} r_k &:= f - (Au_k + B^T p_k) \\ s_k &:= g - Bu_k \end{aligned}$$

2. Solve the Schur Complement system for δp_k:

$$BA^{-1}B^T \delta p_k = BA^{-1} r_k - s_k$$

3. Solve for δu_k using:

$$\delta u_k = A^{-1}(r_k - B^T \delta p_k)$$

4. Update:

$$\begin{aligned} u_{k+1} &:= u_k + \delta u_k \\ p_{k+1} &:= p_k + \delta p_k \end{aligned}$$

Difficulties

- The matrix A^{-1} is never explicitly formed.
- The Schur complement matrix $BA^{-1}B^T$ therefore cannot be used in the sense of a traditional iterative solver.
- Remark: By "traditional" iterative solver, we mean one in which the main computational task is to compute matrix-vector solves Ax such as e.g. CG, BCG, GMRES, etc.

Implementation

- PETSc provides routines to work with "shell" matrices, for example `MatCreateShell(...)` `MatShellSetOperation(...)`

- They allow the user to re-define what it means to compute a matrix-vector product, known as `MATOP_MULT` in PETSc.

- The Schur complement system requires a *linear solve* for each matrix-vector product (sometimes referred to as inner iteration)

Matrix-Vector Product

Problem Statement: Given matrices A, B, B^T, and input vector x, compute the result $y = BA^{-1}B^Tx$.

1. Compute $z := B^Tx$.

2. Solve* the linear system $Aw = z$ for w.

3. Compute $y = Bw$.

$*$: This solve is conducted with a few iterations of the PETSc built-in ILU preconditioner and GMRES solver.

Further Difficulties

- The Schur Complement system $BA^{-1}B^T\delta p_k = BA^{-1}r_k - s_k$ may require many (expensive) iterations if a suitable preconditioner is not used.
- Traditional preconditioners are not usable since the matrix $BA^{-1}B^T$ is not formed.
- PETSc routines to apply "shell" preconditioners available: `PCShellSetApply(...)`
- Work in Progress:
  - What preconditioner? (M_p^{-1}, A_p^{-1}, etc...)
  - Boundary conditions for preconditioner?