On the Construction of Preconditioners for Systems of PDEs

Kent-Andre Mardal

Center for Biomedical Computing

11th March 2011

Outline

- Abstract framework for preconditioning
- A simple example: Stokes problem
- Related code
- More advanced examples (time dependent Stokes problem and an inverse problem)
- Related code
- Conclusion

Abstract Framework based on Functional Analysis

Let us consider the abstract problem:
Find $u \in V$ such that for $f \in V^*$

$$Au = f,$$

where A is a linear operator.

This problem is well-posed if A is an isomorphism mapping V to V^*, i.e.,

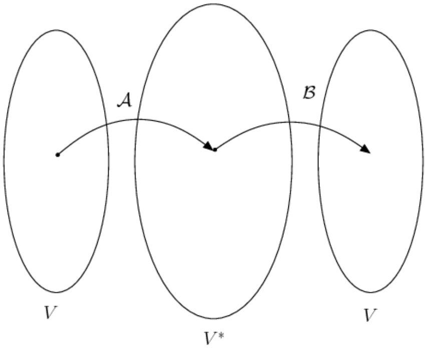
$$\|A\|_{L(V,V^*)} \leqslant C_1 \text{ and } \|A^{-1}\|_{L(V^*,V)} \leqslant C_2$$

Note that A has an unbounded spectrum and this causes problems for iterative solvers (both in the continuous and discrete cases).

Abstract Framework based on Functional Analysis

From a mathematical point of view the Riesz mapping $B : V^* \to V$ is the perfect preconditioner, since

$$\|B\|_{L(V^*,V)} = 1 \text{ and } \|B^{-1}\|_{L(V,V^*)} = 1$$



Abstract Framework based on Functional Analysis

Since

$$\|B\|_{L(V^*,V)} = 1 \text{ and } \|B^{-1}\|_{L(V,V^*)} = 1$$

the condition number of the preconditioned system is bounded as,

$$\|BA\|_{L(V,V)} \leqslant C_1 \text{ and } \|(BA)^{-1}\|_{L(V,V)} \leqslant C_2$$

Multigrid and domain decomposition techniques produce spectrally equivalent and efficient representations of Riesz mappings in most common spaces!

[Mardal, Winther NLAA 2011,
Hiptmair Comp. & Math. with Appl. 2006, Kirby SIAM Review 2011]

Example: Stokes problem

Consider the Stokes problem: Find $u, p \in H_0^1 \times L_0^2$ such that for $f \in H^{-1}$

$$A \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} -\Delta & -\nabla \\ \nabla\cdot & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}$$

The Riesz mapping B taking $H^{-1} \times L_0^2 \rightarrow H_0^1 \times L_0^2$ is

$$B = \begin{bmatrix} -\Delta^{-1} & 0 \\ 0 & I \end{bmatrix}$$

The spectrum of BA is bounded (even though B is very different from A)!

It is easy to construct spectrally equivalent and efficient versions of B with multigrid and domain decomposition techniques.

Corresponding code in FEniCS

```
V = VectorFunctionSpace(mesh, "CG", 2)
Q = FunctionSpace(mesh, "CG", 1)
W = V * Q

(v, q) = TestFunctions(W)
(u, p) = TrialFunctions(W)

a = (inner(grad(v), grad(u)) + div(v)*p +
    q*div(u))*dx
p = (inner(grad(v), grad(u)) + p*q)*dx

A, b = assemble_system(a, ...)
P, c = assemble_system(b, ...)

solver = KrylovSolver("tfqmr", "hypre_amg")
solver.set_operators(A, P)
solver.solve()
```

Extensions for parameter dependent problems: Weighted Sobolev spaces

Consider the problem: Find $u \in H^1_0$, for $f \in H^{-1}$

$$A_\alpha u = u - \alpha^2 \Delta u = f$$

Here, $\alpha > 0$

$$\|A_\alpha^{-1}\|_{L(H^{-1}, H^1_0)} \to \infty \text{ as } \alpha \to 0$$

If we consider A_α in $V = L_2 \cap \alpha H^1_0$ with inner product

$$(u, v)_{L_2 \cap \alpha H^1_0} = (u, v)_{L_2} + \alpha^2 (\nabla u, \nabla v)$$

Then

$$\|A_\alpha\|_{L(V, V^*)} = 1 \text{ and } \|A_\alpha^{-1}\|_{L(V^*, V)} = 1$$

Hence, A_α is the Riesz mapping between these weighted spaces.

(Bergh and Löfström, Interpolation Spaces, 1976)

Time-dependent Stokes problem

$$\begin{aligned} u - k\Delta u - \nabla p &= f \quad \text{in } \Omega, \\ \nabla \cdot u &= 0 \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega. \end{aligned}$$

Here k is the time stepping parameter.

This operator is an isomorphism mapping
$L^2 \cap k^{1/2} H_0^1 \times H^1 \cap L_0^2 + k^{-1/2} L_0^2$ to its dual space.
Furthermore, the bounds are uniform in k.

[Mardal-Winther, Numer. Math. 2004]

A preconditioner for the time-dependent Stokes problem

A preconditioner can be defined as

$$B = \begin{pmatrix} K^{-1} & 0 \\ 0 & L^{-1} + M^{-1} \end{pmatrix}.$$

where

$$\begin{aligned} (Ku, v) &= \int_\Omega u \cdot v + k^2 \nabla u : \nabla v \, dx, && (1) \\ (Lp, q) &= \int_\Omega k^{-2} pq \, dx, && (2) \\ (Mp, q) &= \int_\Omega \nabla p \cdot \nabla q \, dx. && (3) \end{aligned}$$

Code for constructing the preconditioner

```
a = dot(u, v)*dx + k*dot(grad(u),grad(v))*dx
  + div(v)*p*dx  + q*div(u)*dx

# the three components of the preconditioner
k = dot(u, v)*dx + k*inner(grad(u), grad(v))*dx
l = kinv*p*q*dx
m = dot(grad(p),grad(q))*dx

# right hand sides
L0 = dot(f, v)*dx; L1 = q*g*dx

# Assemble the various matrices
A, b  = assemble_system(a, L0, bc)
K, b0 = assemble_system(k, L0, bc)
L, b1 = assemble_system(l, L1)
M, b1 = assemble_system(m, L1)
```

Snippet from Python implementation of MinRes

```
while sqrt(rho) > tolerance and iter < maxit:
    ...

    # apply prec
    uo    = B*qo

    alpha = inner(s,q)
    x    += alpha*p
    s    -= alpha*u

    # compute residual
    r = b - A*x
    rho = fabs(rho)
    ...
```

A parameter identification problem

$$\min_{v \in L^2(\Omega)} \left\{ \frac{1}{2} \|u - d\|^2_{L^2(\Omega)} + \frac{1}{2}\alpha \|v\|^2_{L^2(\Omega)} \right\}$$

subject to

$$\begin{aligned} -\Delta u &= v + g \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega. \end{aligned}$$

[Schöberl, Zulehner SIAM J. Matrix Anal. 2007,
Nielsen and Mardal SIAM J. Control Optim., 2010]

A one-shot formulation of this problem gives a saddle point problem

- $$\begin{bmatrix} \alpha M & 0 & M \\ 0 & M & A \\ M & A & 0 \end{bmatrix} \begin{bmatrix} v \\ u \\ w \end{bmatrix} = \begin{bmatrix} \alpha M v \\ M d \\ g \end{bmatrix}$$
- M is mass matrix and A is stiffness matrix
- Typically ill-posed for $\alpha = 0$
- What are suitable norms for this system ?

Proper norms for this problem

Where is A an isomorphism?

- $A = \begin{bmatrix} \alpha M & 0 & M \\ 0 & M & A \\ M & A & 0 \end{bmatrix} : X \times Y \to (X \times Y)^*$

- $\|(v, u)\|_X^2 = \alpha \|v\|_0^2 + \alpha \|u\|_1^2 + \|u\|_0^2$
- $\|w\|_Y^2 = \frac{1}{\alpha} \|w\|_1^2 + \frac{1}{\alpha}^2 \|w\|_0^2$

Preconditioning

- Preconditioner should be an isomorphism

$$B : (X \times Y)^* \to X \times Y$$

- For example

$$B^{-1} = \begin{bmatrix} \alpha M & 0 & 0 \\ 0 & \alpha A + M & 0 \\ 0 & 0 & \frac{1}{\alpha} A + \frac{1}{\alpha}^2 M \end{bmatrix}$$

(in practice we use multigrid preconditioners)

Code for parameter identification problem

```
UU = Function(VV); U, V, W  = split(UU)

minimization   = ((U-d)**2 + alpha*V**2)*dx
state_eq    = (dot(grad(U),grad(W))-V*W-g*W)*dx

L = minimization + state_equation

norms = (alpha*V*V +
    alpha*(dot(grad(U),grad(U))) + U*U
  + (1/alpha)*dot(grad(W), grad(W))
  + (1/(alpha*alpha))*W*W)*dx

u = TrialFunction(VV)
v = TestFunction(VV)

prec = derivative(derivative(norms,UU,v),UU,u)
a  = derivative(derivative(L,UU,v),UU,u)
```

Further reading:

Logg, Mardal, Wells, Automated Scientific Computing, Springer, 2011 (URL:https://launchpad.net/fenics-book)

Mardal and Winther, NLAA, 2011

Nielsen and Mardal, SIAM J. Control Optim., 2010

Questions?