MATH 676

–

Finite element methods in scientific computing

Wolfgang Bangerth, Texas A&M University

Lecture 37:

What preconditioner to use

Part 4: Simple preconditioners for complex problems

Part 4:
Constructing simple preconditioners
for non-scalar, non-trivial problems

(1980s – 1990s)

Non-scalar problems

Non-scalar (vector-valued) problems:

- Stokes:

$$\begin{aligned} -\mu \Delta u + \nabla p &= f \\ \nabla \cdot u &= 0 \end{aligned}$$

- Navier-Stokes:

$$\begin{aligned} -\mu \Delta u + u \cdot \nabla u + \nabla p &= f \\ \nabla \cdot u &= 0 \end{aligned}$$

- Coupled physics (e.g. thermo-elasticity):

$$\begin{aligned} -k \Delta T &= q \\ -\lambda \nabla (\nabla \cdot \vec{u}) - \mu \nabla \cdot (\nabla \vec{u} + \nabla \vec{u}^T) &= \beta \nabla T \end{aligned}$$

- Mixed form of the biharmonic equation:

$$\begin{aligned} -\Delta u - v &= 0 \\ -\Delta v &= f \end{aligned}$$

Non-scalar problems

Such problems induce block-structured matrices:

- Stokes:

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}$$

- Navier-Stokes:

$$\begin{pmatrix} A+N & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}$$

- Coupled physics (e.g. thermo-elasticity):

$$\begin{pmatrix} A & 0 \\ -B & C \end{pmatrix} \begin{pmatrix} T \\ U \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}$$

- Mixed form of the biharmonic equation:

$$\begin{pmatrix} A & -M \\ 0 & A \end{pmatrix} \begin{pmatrix} U \\ V \end{pmatrix} = \begin{pmatrix} 0 \\ F \end{pmatrix}$$

Non-scalar problems

Problem: In many coupled problems, there is a zero block on the diagonal.

For example, for Stokes:

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}$$

Consequence: Jacobi, Gauss-Seidel, SOR, SSOR can no longer be used because they divide by diagonal entries!

Non-scalar problems

Idea 1 (1980s, early 1990s): Generalize what we did for scalar problems.

Example: For Jacobi, we multiply with the inverse of diagonal elements

$$P^{-1} = D^{-1}$$

Interpretation:

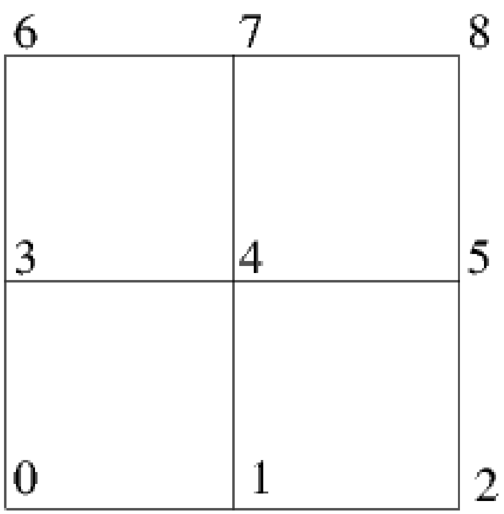
- For scalar elements, there is one degree of freedom per "node"
- Jacobi preconditioning then walks over all "nodes" and updates their values one-by-one
- Gauss-Seidel, SOR, SSOR work similarly

Consequence: We sometimes call them "point solvers".

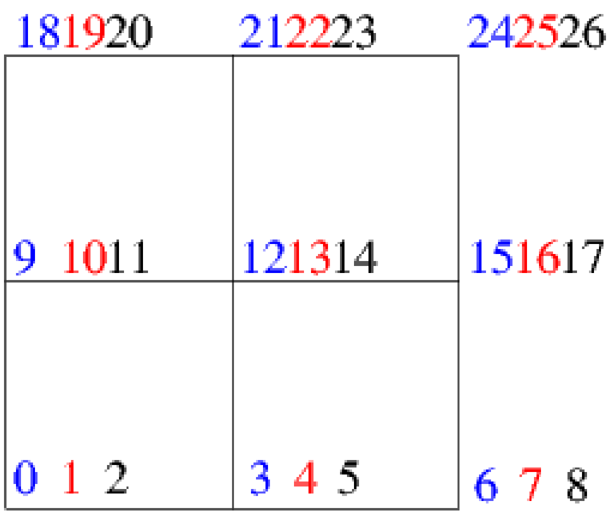
Non-scalar problems

Idea 1 (1980s, early 1990s): Generalize what we did for scalar problems.

Approach: Assume for the moment a Q1/Q1 element



Vertices



DoF indices: u, v, p

Non-scalar problems

Idea 1 (1980s, early 1990s): Generalize what we did for scalar problems.

Approach: Assume for the moment a Q1/Q1 element

- Order DoFs by node, group matrix into blocks

$$A = \begin{pmatrix} a_{11}^{uu} & a_{12}^{uv} & a_{13}^{up} & a_{14}^{uu} & a_{15}^{uv} & a_{16}^{up} & \cdots \\ a_{21}^{vu} & a_{22}^{vv} & a_{23}^{vp} & a_{24}^{vu} & a_{25}^{vv} & a_{26}^{vp} & \cdots \\ a_{31}^{pu} & a_{32}^{pv} & 0 & a_{34}^{pu} & a_{35}^{pv} & 0 & \cdots \\ a_{41}^{uu} & a_{42}^{uv} & a_{43}^{up} & a_{44}^{uu} & a_{45}^{uv} & a_{46}^{up} & \cdots \\ a_{51}^{vu} & a_{52}^{vv} & a_{53}^{vp} & a_{54}^{vu} & a_{55}^{vv} & a_{56}^{vp} & \cdots \\ a_{61}^{pu} & a_{62}^{pv} & 0 & a_{64}^{pu} & a_{65}^{pv} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \qquad U = \begin{pmatrix} u_1 \\ v_1 \\ p_1 \\ u_2 \\ v_2 \\ p_2 \\ \vdots \end{pmatrix}$$

Non-scalar problems

Idea 1 (1980s, early 1990s): Generalize what we did for scalar problems.

Approach: Assume for the moment a Q1/Q1 element
- Order DoFs by node, group matrix into blocks
- Define a "block Jacobi" preconditioner:

$$P^{-1} := D^{-1} = \begin{pmatrix} a_{11}^{uu} & a_{12}^{uv} & a_{13}^{up} & 0 & 0 & 0 & \cdots \\ a_{21}^{vu} & a_{22}^{vv} & a_{23}^{vp} & 0 & 0 & 0 & \cdots \\ a_{31}^{pu} & a_{32}^{pv} & 0 & 0 & 0 & 0 & \cdots \\ & & & & & & \\ 0 & 0 & 0 & a_{44}^{uu} & a_{45}^{uv} & a_{46}^{up} & \cdots \\ 0 & 0 & 0 & a_{54}^{vu} & a_{55}^{vv} & a_{56}^{vp} & \cdots \\ 0 & 0 & 0 & a_{64}^{pu} & a_{65}^{pv} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}^{-1}$$

Non-scalar problems

Idea 1 (1980s, early 1990s): Generalize what we did for scalar problems.

Approach: Assume for the moment a Q1/Q1 element

- Order DoFs by node, group matrix into blocks
- Define a “block Jacobi” preconditioner

- Cheap to compute/apply: Only 3x3 matrix inverses
- Still called “point solver”

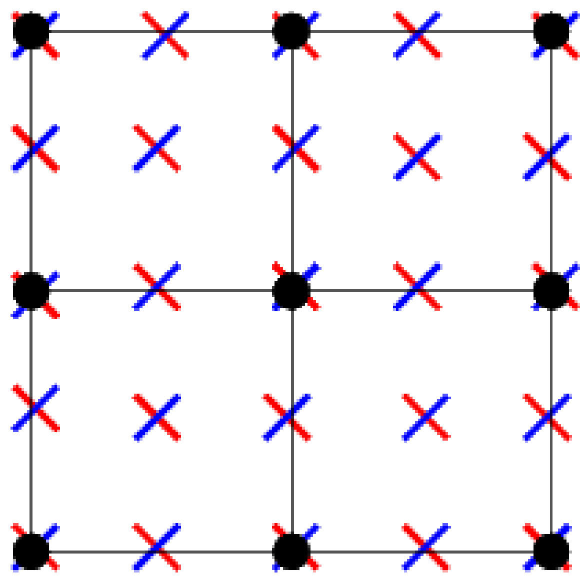
- Can be generalized to block-Gauss-Seidel, block-SSOR, etc

Non-scalar problems

Idea 1 (1980s, early 1990s): Generalize what we did for scalar problems.

Problem: What if it is a Q2/Q1 element?

- Not all DoFs defined at vertices:



u, v, p

Non-scalar problems

Idea 1 (1980s, early 1990s): Generalize what we did for scalar problems.

Problem: What if it is a Q2/Q1 element?

- Not all DoFs defined at vertices
- Can't define “point local” systems

Non-scalar problems

Idea 1 (1980s, early 1990s): Generalize what we did for scalar problems.

Problem: What if it is a Q2/Q1 element?

- Not all DoFs defined at vertices
- Can't define “point local” systems

Solution (Vanka, 1986):

- Define local problems for each pressure DoF
- Solve one after the other in the spirit of Jacobi or SSOR

- Can be generalized to other saddle point problems

Non-scalar problems

Idea 1 (1980s, early 1990s): Generalize what we did for scalar problems.

Experience:

- Vanka preconditioners do work for general saddle-point problems
- However, not overly well
  – Shares the same problems as Jacobi/SSOR/...
  – In particular, slow speed of information propagation
- Has fallen out of favor since the late 1990s

Non-scalar problems

Idea 2 (1980s, early 1990s): Use ILU/IC.

Experience:

- Works just fine
- Limited by memory consumption/CPU time

- Limited use for large problems

Non-scalar problems

Idea 3 (1990s, 2000s): Generalize multigrid.

Observation:

- Many coupled problems have "elliptic character" (e.g. Stokes), so why not use multigrid on the whole system?

Experience:

- It's not that easy
- Smoothers have to have very particular properties that are difficult to achieve

- "Direct multigrid" for coupled problems has fallen out of favor

Non-scalar problems

Summary:

Finding good but simple preconditioners
for coupled problems is difficult!

- Jacobi/SSOR do not work
- Generalizations such as Vanka work, but not well
- Multigrid
  – could work in theory
  – but is difficult to make well in practice

MATH 676

–

Finite element methods in scientific computing

Wolfgang Bangerth, Texas A&M University