October 11, 2013

# Preconditioning

- right, left, split

or you can think of them as

- "algebraic" preconditioners $\longrightarrow$ ILU
- problem-tailored " $\longrightarrow$ eg. if the problem comes from PDEs, try to find a preconditioner that "knows" what underlying differential operator the linear system features

ex. if

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \rightsquigarrow \text{stokes}$$

$$-\Delta \vec{u} + \nabla p = f$$
$$\nabla \cdot u = 0$$

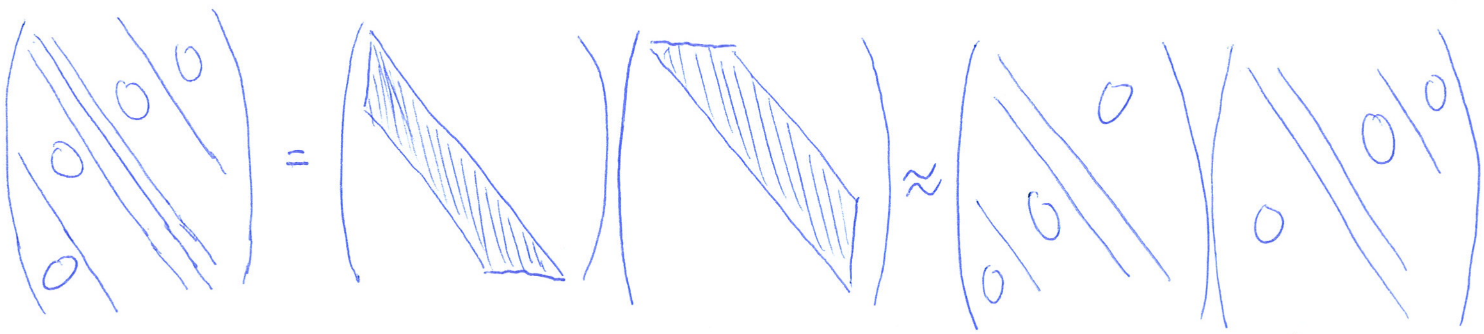
$\hookrightarrow$ Schur complement : $BA^{-1}B^T$

$$\nabla \cdot \nabla \equiv \Delta \quad \text{so} \quad \overset{\shortparallel}{\nabla \cdot \Delta^{-1} \nabla} \approx \underbrace{\nabla \cdot \nabla \Delta^{-1}}_{\text{Identity}}$$

$\hookrightarrow$ mass matrix preconditioner

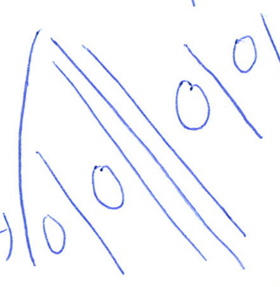
(related to inf-sup, LBB spectral equivalence).

ILU:



$(ILU)_{ij} = A_{ij}$ wherever $A_{ij} \neq 0$ =

(5 diagonals are exact)

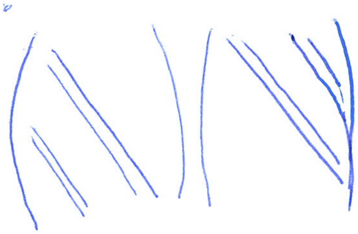


$\|A - LU\|$ not small, but $(LU)^{-1}A$ has many eig $\simeq 1$

ILU is $\sim O(n)$

For each $(i,j) \in P$, set $a_{ij} = 0$
For $k = 1, \dots, n-1$ pg 289 of Saad
for $i = k+1 : n$ and if $(i,k) \notin P$ ← static pattern (diagonal of A is always connected)
do $a_{ik} = a_{ik} / a_{kk}$
for $j = k+1 : n$ $(i,j) \in P$
$a_{ij} := a_{ij} - a_{ik} a_{kj}$
end
end
end

ILU(0): no fill → good only for easy problems
ILU(1):



Dynamic pattern:
- what we really want is to drop elements that are small enough
  ↳ ILUT: ILU w/ a threshold
  ILUTP: ILUT w/ pivoting

replace the condition on P (static pattern) by a more sophisticated dropping rule.
- $\tau$: drop an element below a value $\tau \cdot \text{norm(row)}$ ← to deal w/ scaling
- p: want to keep only upto p largest elements in a row.
- to allow pivoting, use a row-oriented version of G.E., called KIJ.

MILU: modified ILU

- make $LU$ and $A$ have the same row-sums, typically by modifying the diagonal

  $LUe = Ae$ $\quad$ e-vector of all 1's

- significantly better than ILU as a stationary scheme (for our usual model problem)

  $M = LU$

  $LU x_{k+1} = (LU - A)x_k + b$

Who Cares?! → but we want to use it as a preconditioner