Welcome to Ma221! Lecture 16, Sep 29

Courses announcement: class projects related to LAPACK

How do we pick best permutation $P$ 

i.e. order of row & columns to do Cholesky on $PA^T$? Goal: minimize flops + memory

(backward stable for any $P$)

Use language of graph theory:

vertices: rows and columns
edges: locations of nonzeros $(v_1, v_2)$ \(\rightarrow\) nonzero in $A_{v_1,v_2}$
weights: values of nonzeros

Last Time: RCM: Reverse Cuthill McKee

Breadth First Search: order vertices by distance from starting vertex

\(\Rightarrow\) "band" matrix

(block tri-diagonal)

Minimum Degree (MD)

Def: degree($v$) = # edges touching $v$

\(\Rightarrow\) # nonzeros in row & column $v$

Fact: If I pick $v$ as pivot:

\(\Rightarrow\) # flops = $d^2$ muls + add, $d$ = degree($v$)
\[ \Rightarrow \text{fill-in (new nonzeros) } \leq d^2 \]

\[ \cdot \cdot \cdot \quad \text{degree}(v) = 3 \]

Greedy Alg: at each step pick 
v with minimum \( \text{deg}(v) \)

\[ \text{degree}(v) = 3 \]

update graph to show nonzeros of next step
again pick minimum degree(v)

Matlab: amd, symamd, colamd

if \( A \neq A^T \) apply to \( A + A^T \)

ND = Nested Dissection: good if 
vertices only connected to
"nearest neighbors"

\[ V = \text{vertices} = V_1 \cup V_2 \cup V_3 \text{ disjoint} \]

\[ 1. \quad |V_1| \approx |V_2| \]

\[ 2. \quad |V_3| \text{ much smaller} \]

\[ 3. \quad \text{no edges connecting } V_1 \text{ and } V_2 \]

Order vertices \( V_1 \) first, then \( V_2 \), then \( V_3 \)
\[ A = \begin{bmatrix} v_1 & v_2 & v_3 \\ v_2 & A_{11} & 0 \\ v_3 & A_{21} & A_{22} \end{bmatrix} \quad \Rightarrow \quad L = \begin{bmatrix} L_{11} & & \\ & L_{22} & \\ & & L_{33} \end{bmatrix} \]

Apply recursively to \(A_{11}\) and \(A_{22}\)

(plots of 7x7 mesh)

Thm (George, Hoffman (M.1), Rose, Gilbert, Tarjan 70s-80s)

Any ordering for Cholesky on \(n \times n\) mesh does at least \(\Omega(n^3)\) ops

= dense Cholesky on last separator, a dense \(n \times n\) matrix which costs \(O(n^3)\)

attained by ND

Applies to planar graphs (draw it on paper with no edge crossings)
(ex: NASA airfoil) (type "load airfoil" in Matlab)

Thm (Ballard, D., Hall, Schwartz 2009)

\(\Omega(\#\text{flops}/IM)\) \#flops depends
on sparsity pattern

Thm: (David, D., Grigor, Peyroud, 2010)
attainable by ND, done “carefully”
bottleneck is dense nxn matrix
for \( V_3 \)

Contrast with BandSolver:

\[ \# \text{flops} = O(b^2 \cdot \text{dimension}) = n^4 \] for nxn mesh

(How to pick \( V_3 \) if not simple mesh:
lots of algo/Software, see CS267
METIS, ParMETIS)

What about 3D meshes \( n \times n \times n \)

\[ \begin{tiny} V_1 \end{tiny} \quad \begin{tiny} V_2 \end{tiny} \quad \begin{tiny} ND \end{tiny} \] still good

\[ \text{dimension} = n^3 \]

dense Chol: costs \( O(n^3) \)

band Chol: cost \( O(n^2) \)

\( ND \) : costs \( O(n^6) \)

Steps of sparse Cholesky:

Choose ordering (RCM, MD, ND, ...)
Build data Structures for \( A, L \)
Perform factorization

Contrast with GE with Partial Pivoting

where data structure dynamic
Ways to deal with Sparse general $A$

1. *Threshold pivoting*: among pivot choices at each step, pick one with least fill in, within a factor of 2 or 3 of largest (i.e. MD with some stability)

2. *Static Pivoting*: (SuperLU)

   1. reorder and scale $A$ to make diagonal as large as possible
   
   Thm: For any nonsingular $A$
   
   Exist Perm $P$ and diagonal $D_1, D_2$
   
   s.t. $B = P \cdot AP \cdot D_2$ satisfies
   
   $|B(i,i)| = 1$ and $|B(i,j)| \leq 1$
   
   $\Rightarrow$ like Cholesky, choose pivots from diagonal

   2. reorder rows and cols of $B$

   using same techniques as for Cholesky, build data structures for $B, L, U$

3. During factorization, if a prospective pivot too small, make it bigger (rare) — low rank perturbation of $A$

   $\Rightarrow$ use Sherman-Morrison or
   
   use perturbed $A$ as preconditioner in iterative algorithm (GMRES)

Lots of algs, SW... (see "updated survey" by Xiaoye Li)

on class webpage