Outline

Theory

Load balancing

Cholesky

Symbolic computations

Practical considerations

Implementations

Literature

Direct methods for linear systems

Imre Pólik, PhD

McMaster University
School of Computational Engineering and Science

January 14, 2008

Outline

Theory

Load balancing

Cholesky

Symbolic computations

Practical considerations

Implementations

Literature

Outline

1. Theoretical background
2. Example: Better load balancing
3. Cholesky factorization
4. Symbolic computations
5. Practical considerations
6. Implementations

Outline

Theory
Factorizations

Load balancing

Cholesky

Symbolic computations

Practical considerations

Implementations

Literature

Factorizations

- Goal: simplify the structure of the matrix
  - triangular, (orthogonal)
- LU factorization
  - $A = LU$
  - Gaussian elimination
- Cholesky (symmetric LU) factorization
  - $A = LL^T$ or $A = LDL^T$
- Factors are unique
- They can be sparse for sparse A

Outline

Theory

Load balancing

Tridiagonal

Cyclic reduction

Cholesky

Symbolic computations

Practical considerations

Implementations

Literature

Tridiagonal systems

$$A = \begin{pmatrix} a_1 & c_1 & & & \\ c_1 & a_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-2} & a_{n-1} & c_{n-1} \\ & & & c_{n-1} & a_n \end{pmatrix}, L = \begin{pmatrix} 1 & & & & \\ \ell_1 & 1 & & & \\ & \ddots & \ddots & & \\ & & \ell_{n-2} & 1 & \\ & & & \ell_{n-1} & 1 \end{pmatrix}$$

$$\begin{aligned} d_1 &= a_1, \ell_1 = c_1/d_1 \\ i = 2, \dots, n & \\ d_i &= a_i - \ell_{i-1} c_{i-1} \\ \ell_i &= c_i/d_i \end{aligned}$$

- Total work: $3n$ (factorization) $+ 5n$ (solution)
- Parallelization
  - $\ell_1 \to \ell_2 \to \cdots \to \ell_{n-1}$ (critical path)
  - at least $n - 1$ steps, poor parallel performance

Cyclic reduction

- Eliminate the odd numbered variables in parallel

$$x_1 = \frac{b_1 - c_1 x_2}{a_1}$$

$$x_3 = \frac{b_3 - c_3 x_4 - c_2 x_2}{a_3}$$

$$\vdots$$

- Remaining system: tridiagonal, $\lceil n/2 \rceil$ variables
- Total work: $16n$ (factorization and solution)
- Critical path: $\log n$ (instead of $n - 1$)
- Favourable for larger n
- Generalizes to band matrices

Outline

Theory

Load balancing

Cholesky

Right-looking

Left-looking

Multifrontal

Symbolic computations

Practical considerations

Implementations

Literature

1. Theoretical background
2. Example: Better load balancing
3. Cholesky factorization
   - Right-looking
   - Left-looking
   - Multifrontal
4. Symbolic computations
5. Practical considerations
6. Implementations

Outline

Theory

Load balancing

Cholesky

Right-looking

Left-looking

Multifrontal

Symbolic computations

Practical considerations

Implementations

Literature

Right-looking Cholesky factorization

$$
\begin{aligned}
A &= \begin{pmatrix} d_1 & v_1^T \\ v_1 & H_1 \end{pmatrix} \\
&= \begin{pmatrix} \sqrt{d_1} & 0 \\ \frac{v_1}{\sqrt{d_1}} & I_{n-1} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & H_1 - \frac{v_1 v_1^T}{d_1} \end{pmatrix} \begin{pmatrix} \sqrt{d_1} & \frac{v_1}{\sqrt{d_1}} \\ 0 & I_{n-1} \end{pmatrix} \\
&= L_1 A_1 L_1^T \\
&\vdots \\
A_{n-1} &= L_n I_n L_n^T \\
L &= L_1 L_2 \dots L_n = L_1 + L_2 + \dots + L_n - (n-1) I_n
\end{aligned}
$$

- Cost:
  - sparse: $\mathcal{O}\left(\sum_{i=1}^{n} \operatorname{nnz}(L_{:,i})^2\right)$
  - dense: $\mathcal{O}(n^3)$
- Fill: l_{ij} becomes nonzero if $k < j$ and $l_{ik} \neq 0, l_{jk} \neq 0$

Outline

Theory

Load balancing

Cholesky

Right-looking

Left-looking

Multifrontal

Symbolic computations

Practical considerations

Implementations

Literature

Left-looking Cholesky factorization

- Solve $A = LL^T$ for elements of L

$$
\begin{aligned}
j &= 1, \dots, n \\
&\quad k = 1, \dots, j-1 \\
&\qquad \begin{pmatrix} a_{jj} \\ \vdots \\ a_{nj} \end{pmatrix} \leftarrow \begin{pmatrix} a_{jj} \\ \vdots \\ a_{nj} \end{pmatrix} - l_{jk} \begin{pmatrix} l_{jk} \\ \vdots \\ l_{nk} \end{pmatrix} \\
&\quad l_{jj} \leftarrow \sqrt{a_{jj}} \\
&\quad \begin{pmatrix} l_{j+1,j} \\ l_{j+2,j} \\ \vdots \\ l_{nj} \end{pmatrix} \leftarrow l_{jj}^{-1} \begin{pmatrix} a_{j+1,j} \\ a_{j+2,j} \\ \vdots \\ a_{nj} \end{pmatrix}
\end{aligned}
$$

- Same cost, same fill-in

Outline

Theory

Load balancing

Cholesky

Right-looking

Left-looking

Multifrontal

Symbolic computations

Practical considerations

Implementations

Literature

Multifrontal Cholesky factorization

$$
\begin{aligned}
A &= \begin{pmatrix} G & V^T \\ V & H \end{pmatrix} \\
&= \begin{pmatrix} L_G & 0 \\ VL_G^{-T} & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & H - VG^{-1}V^T \end{pmatrix} \begin{pmatrix} L_G & L_G^{-1}V^T \\ 0 & I \end{pmatrix}
\end{aligned}
$$

- Factorize G to get L_G
- Get $VG^{-1}V^T = (VL_G^{-T})(L_G^{-1}V^T)$
- Continue with $H - VG^{-1}V^T$
- Very efficient in practice

Outline

Theory

Load balancing

Cholesky

Symbolic computations

Symbolic factorization

Orderings

Practical considerations

Implementations

Literature

Symbolic factorization

- Find the structure of the factor
- Run the algorithms without the computation
- Quick algorithms
- Defines the storage

Outline
Theory
Load balancing
Cholesky
Symbolic computations
Symbolic factorization
Orderings
Practical considerations
Implementations
Literature

Fill reducing orderings

- Reorder the matrix to have sparse factors
- Best ordering is NP-hard to find
- Heuristics
- Approximate Minimum Degree ordering (Davis)
  - AMD: default in MA41, CPLEX
  - COLAMD: SuperLU
  - part of Matlab
- COLPERM: increasing nonzero count (LU)
- SYMRCM: reverse Cuthill-McKee, reduces bandwidth (LU and Cholesky)
  See Matlab's `lu` and `spparms` for details.

Outline

Theory

Load balancing

Cholesky

Symbolic computations

Practical considerations

Elimination trees, supernodes

Numerical problems

General solution scheme

Refinement

Implementations

Literature

1. Theoretical background
2. Example: Better load balancing
3. Cholesky factorization
4. Symbolic computations
5. Practical considerations
   - Elimination trees, supernodes
   - Numerical problems
   - General solution scheme
   - How to get a better solution?
6. Implementations

Outline

Theory

Load balancing

Cholesky

Symbolic computations

Practical considerations

Elimination trees, supernodes

Numerical problems

General solution scheme

Refinement

Implementations

Literature

Elimination trees, supernodes

- Elimination tree
  - The structure of L drives the algorithm
  - The order of elimination is not fixed
- Supernode
  - Columns with identical sparsity pattern
  - Treated together to
    - gain speedup from BLAS
    - reduce storage
  - The speedup is practical, not theoretical!

Outline

Theory

Load balancing

Cholesky

Symbolic computations

Practical considerations

Elimination trees, supernodes

Numerical problems

General solution scheme

Refinement

Implementations

Literature

Numerical problems

- Factorization of ill-conditioned/close to singular matrices
- Avoid division if the result is too small/large
  - global pivot shifting: work on $A + \alpha I$
  - local pivot shifting: work on $A + \alpha E_i$
  - pivot skipping
- Reduces performance
- Limits parallelism

Outline

Theory

Load balancing

Cholesky

Symbolic computations

Practical considerations

Elimination trees, supernodes

Numerical problems

General solution scheme

Refinement

Implementations

Literature

How to solve $Ax = b$ with factorization?

1. Choose an appropriate ordering
   - based on factorization algorithm
2. Determine the structure of the factor
   - set up storage for the factor
3. Factorize
   - costliest part
4. Solve the triangular systems

Outline

Theory

Load balancing

Cholesky

Symbolic computations

Practical considerations

Elimination trees, supernodes

Numerical problems

General solution scheme

Refinement

Implementations

Literature

Iterative refinement

- Approximate solution

$$Ax \approx b$$

- Correction

$$A\Delta x = b - Ax$$

- Easy to solve is A is already factorized! $\mathcal{O}(n^2)$ extra work

Direct solvers for sparse systems

- Serial

Code	Type	Scope	By
MA47	MF	SPD	HSL
Sparspak	Left	SPD	George
SuperLU	Left	Gen	Li
UMFPACK	MF	Gen	Davis

- Parallel

Code	Type	Scope	By
DMF	MF	Sym	Lucas
MA41	MF	Sym-pat	HSL
PARDISO	Left	Sym-pat	Schenk
SuperLU MT	Left	Gen	Li
WSMP	MF	Sym	IBM

J. Demmel, P. Koev, and X. Li.

A brief survey of direct linear solvers.

In Zhaojun Bai, James Demmel, Jack Dongarra, Axel Ruhe, and Henk van der Vorst, editors, *Templates for the Solution of Algebraic Eigenvalue Problems: a Practical Guide*. SIAM, 2000.

Alan Edelman.

MIT 18.337: Applied parallel computing.

Lecture notes, 2004.

Chapter 4 and 5.

Alan George and Joseph W. Liu.

Computer Solutions of Large Sparse Positive Definite Systems.

Prentice Hall, 1981.