MULTIFRONTAL METHOD

Manoj Gupta

CS497MH

March 25 1999

Road Travelled So Far

Graph Theory

$(Adj(Y),\ Deg(Y),\ Path\ etc.)$

$\downarrow$

Ordering (Phase 1)

Reverse Cuthill Mckee

Minimum degree

Nested Dissection

$\downarrow$

Graph Partioning

Inertial Partitioning

Breadth First Search

Kernighan-Lin

Spectral Partitioning

$\downarrow$

Symbolic Factorization (Phase 2)

Constructing Elimination Tree

Fill Characterization

$\downarrow$

Numeric Factorization (Phase 3)

Forms of Cholesky Factorization

Multifrontal Method

What is Multifrontal Method ?

One line definition:
A reorganization of sparse Cholesky factorization based on the structure of the Elimination tree by partial factorization of dense and smaller submatrices called Frontal matrices formed by using Update matrices from previous steps.

Something for the back of your head:

- Operations on dense matrices.
- Naturally parallelizable
- Good data locality
- Good for vectorization and loop unrolling

Notion of outer-product updates

- One step elimination using submatrix Cholesky.

$$A = \begin{pmatrix} d & v^t \\ v & C \end{pmatrix}$$

$$= \begin{pmatrix} \sqrt{d} & 0 \\ v/\sqrt{d} & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & C - vv^t/d \end{pmatrix} \begin{pmatrix} \sqrt{d} & v^t/\sqrt{d} \\ 0 & I \end{pmatrix}$$

$$= \begin{pmatrix} \sqrt{d} & 0 \\ v/\sqrt{d} & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & H \end{pmatrix} \begin{pmatrix} \sqrt{d} & v^t/\sqrt{d} \\ 0 & I \end{pmatrix}$$

$$= \begin{pmatrix} \sqrt{d} & 0 \\ v/\sqrt{d} & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & L_H \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & L_H^t \end{pmatrix} \begin{pmatrix} \sqrt{d} & v^t/\sqrt{d} \\ 0 & I \end{pmatrix}$$

$$= \begin{pmatrix} \sqrt{d} & 0 \\ v/\sqrt{d} & L_H \end{pmatrix} \begin{pmatrix} \sqrt{d} & v^t/\sqrt{d} \\ 0 & L_H^t \end{pmatrix}$$

$$= LL^t$$

- Assume submatrix B has been factored into $L_B L_B^t$.

$$A = \begin{pmatrix} B & V^t \\ V & C \end{pmatrix}$$

$$= \begin{pmatrix} L_B & 0 \\ VL_B^{-t} & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & C - VB^{-1}V^t \end{pmatrix} \begin{pmatrix} L_B^t & L_B^{-1}V^t \\ 0 & I \end{pmatrix}$$

- The submatrix $-VB^{-1}V^t$ represents the outer-product updates from the <u>appropriate portions</u> of the first $j-1$ columns to the submatrix C.

$$-VB^{-1}V^t = -(VL_B^{-t})(L_B^{-1}V^t)$$

$$= \sum_{k=1}^{j-1} \begin{pmatrix} \ell_{j,k} \\ . \\ . \\ . \\ \ell_{n,k} \end{pmatrix} \begin{pmatrix} \ell_{j,k} & . & . & . & \ell_{n,k} \end{pmatrix}$$

Elimination Trees and Assembly Trees

Theorem: If node k is a descendent of j in the elimination tree, then the structure of the vector $(\ell_{jk}, ..., \ell_{nk})^t$ is contained in the structure $(\ell_{jj}, ..., \ell_{nj})^t$.

Structure of a column is the set of row subscripts of the nonzeros in the column.

Only descendent columns in the elimination affect the proper ancestor columns. The update data donation is upward and not cross.

Hey, I can think of a different tree which can do the same thing!
Sure...

Assembly Tree :
A tree of the n nodes $\{1, ..., n\}$ is an assembly tree if for any node j with parent node p, the off-diagonal structure of L_{*j} is a subset of the structure of L_{*p} and $p > j$.

Example 1

$$A = \begin{array}{c} 1\\ 2\\ 3\\ 4\\ 5\\ 6\\ 7\\ 8\\ 9 \end{array} \begin{pmatrix} a^1 & & & & & & \bullet & \bullet & \bullet \\ & b^2 & & \bullet & & \bullet & & & \\ & & c^3 & & \bullet & & & \bullet & \\ & \bullet & & d^4 & & & & \bullet & \bullet \\ & & \bullet & & e^5 & \bullet & & \bullet & \\ & \bullet & & & \bullet & f^6 & & & \bullet \\ \bullet & & & & & & g^7 & \bullet & \bullet \\ \bullet & & \bullet & \bullet & \bullet & & \bullet & h^8 & \\ \bullet & & & \bullet & & \bullet & \bullet & & i^9 \end{pmatrix}$$

$$L = \begin{array}{c} 1\\ 2\\ 3\\ 4\\ 5\\ 6\\ 7\\ 8\\ 9 \end{array} \begin{pmatrix} a^1 & & & & & & & & \\ & b^2 & & & & & & & \\ & & c^3 & & & & & & \\ & \bullet & & d^4 & & & & & \\ & & \bullet & & e^5 & & & & \\ & \bullet & & \circ & \bullet & f^6 & & & \\ \bullet & & & & & & g^7 & & \\ \bullet & & \bullet & \bullet & \bullet & \circ & \bullet & h^8 & \\ \bullet & & & \bullet & & \bullet & \bullet & \circ & i^9 \end{pmatrix}$$

Elimination Tree

Assembly Tree

Frontal and Update Matrices

• The subtree update matrix:

Let $i_0, i_1, ..., i_r$ be the row subscripts of nonzeros in L_{*j} with $i_0 = j$.

$$\overline{U}_j = -\sum_{k\in T[j]-j} \begin{pmatrix} \ell_{j,k} \\ \ell_{i_1,k} \\ . \\ . \\ . \\ \ell_{i_r,k} \end{pmatrix} \begin{pmatrix} \ell_{j,k} & \ell_{i_1,k}. & . & . & \ell_{i_r,k} \end{pmatrix}$$

It contains <u>all</u> the outer-product contributions from proper descendents of j.

For column 6:

$$\overline{U}_6 = -\begin{pmatrix} \ell_{62} \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} \ell_{62} & 0 & 0 \end{pmatrix} - \begin{pmatrix} 0 \\ \ell_{83} \\ 0 \end{pmatrix} \begin{pmatrix} 0 & \ell_{83} & 0 \end{pmatrix}$$
$$- \begin{pmatrix} \ell_{64} \\ \ell_{84} \\ \ell_{94} \end{pmatrix} \begin{pmatrix} \ell_{64} & \ell_{84} & \ell_{94} \end{pmatrix} - \begin{pmatrix} \ell_{65} \\ \ell_{85} \\ 0 \end{pmatrix} \begin{pmatrix} \ell_{65} & \ell_{85} & 0 \end{pmatrix}$$

jth frontal matrix F_j for A:

$$F_j = \overbrace{\begin{pmatrix} a_{j,j} & a_{j,i_1} & . & . & . & a_{j,i_r} \\ a_{i_1,j} & & & & & \\ . & & & 0 & & \\ . & & & & & \\ . & & & & & \\ a_{i_r,j} & & & & & \end{pmatrix}}^{r+1} + \overline{U}_j$$

Do one step of submatrix-Cholesky elimination on the column j.

$$F_j =$$

$$\begin{pmatrix} \ell_{j,j} & 0 \\ \ell_{i_1,j} & \\ . & I \\ . & \\ \ell_{i_r,j} & \end{pmatrix} \begin{pmatrix} 1 & 0 \\ & \\ & \\ & \\ 0 & U_j \end{pmatrix} \begin{pmatrix} \ell_{j,j} & \ell_{i_1,j} & .. & \ell_{i_r,j} \\ & & & \\ & & & \\ & & & I \\ 0 & & & \end{pmatrix}$$

$+$ is an extend-add operator

The Flow

- The sparsity blesses the independence of different fronts from fronts in the different family.
- The flow is not sequentially consecutive but parallelly hierarchal.
- The essence is advancing fronts while creating and merging update matrices and factorizing atleast one column when the front advances.
- The flow is not like that of a one long river but like that of multiple streams flowing, merging and advancing.

Multifrontal Algorithm

for column $j = 1$ to n do

- Let $j, i_1, ..., i_r$ be the locations of nonzeros in L_{*j};
- Let $c_1, ..., c_s$ be the children of j in the elimination tree;
- Form the update matrix $\overline{U} = U_{c_1} + ... + U_{c_s}$;

$$F_j = \overbrace{\begin{pmatrix} a_{j,j} & a_{j,i_1} & . & . & . & a_{j,i_r} \\ a_{i_1,j} & & & & & \\ . & & & 0 & & \\ . & & & & & \\ . & & & & & \\ a_{i_r,j} & & & & & \end{pmatrix}}^{r+1} + \overline{U}_j$$

Factor F_j into

$$\begin{pmatrix} \ell_{j,j} & 0 \\ \ell_{i_1,j} & \\ . & I \\ . & \\ \ell_{i_r,j} & \end{pmatrix} \begin{pmatrix} 1 & 0 \\ & \\ & \\ & \\ 0 & U_j \end{pmatrix} \begin{pmatrix} \ell_{j,j} & \ell_{i_1,j} & .. & \ell_{i_r,j} \\ & & & \\ & & & \\ & & & I \\ 0 & & & \end{pmatrix}$$

end for

Example 2

$\overline{U}_1, \overline{U}_2, \overline{U}_3 = 0$. Since $T[4] - \{4\} = \{2\}$, then

$$\overline{U}_4 = \begin{pmatrix} \ell_{42} \\ \ell_{62} \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} \ell_{42} & \ell_{62} & 0 & 0 \end{pmatrix}$$

$$F_4 = \begin{pmatrix} a_{44} & 0 & a_{48} & a_{49} \\ 0 & 0 & 0 & 0 \\ a_{84} & 0 & 0 & 0 \\ a_{94} & 0 & 0 & 0 \end{pmatrix} + \overline{U}_4$$

$$= \begin{array}{c} 4 \\ 6 \\ 8 \\ 9 \end{array} \begin{pmatrix} a_{44} & 0 & a_{48} & a_{49} \\ 0 & 0 & 0 & 0 \\ a_{84} & 0 & 0 & 0 \\ a_{94} & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} \ell_{42} \\ \ell_{62} \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} \ell_{42} & \ell_{62} & 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} a_{44} - \ell_{42}^2 & -\ell_{42}\ell_{62} & a_{48} & a_{49} \\ -\ell_{42}\ell_{62} & \ell_{62}^2 & 0 & 0 \\ a_{84} & 0 & 0 & 0 \\ a_{94} & 0 & 0 & 0 \end{pmatrix}$$

Example 2 Continues

$$U_4 = \begin{pmatrix} -\ell_{62}^2 - \ell_{64}^2 & -\ell_{64}\ell_{84} & -\ell_{64}\ell_{94} \\ -\ell_{64}\ell_{84} & -\ell_{84}^2 & -\ell_{84}\ell_{94} \\ -\ell_{64}\ell_{94} & -\ell_{84}\ell_{94} & -\ell_{94}^2 \end{pmatrix}$$

U_4 as the outer-product updates from column 2 and 4.

$$= \begin{pmatrix} \ell_{62} \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} \ell_{62} & 0 & 0 \end{pmatrix} - \begin{pmatrix} \ell_{64} \\ \ell_{84} \\ \ell_{94} \end{pmatrix} \begin{pmatrix} \ell_{64} & \ell_{84} & \ell_{94} \end{pmatrix}$$

U_4 will now be send up to the parent node 6 where it will merge with U_5 and consequently update the column 6.

Example 3

For column 6:

$$\overline{U}_6 = -\begin{pmatrix} \ell_{62} \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} \ell_{62} & 0 & 0 \end{pmatrix} - \begin{pmatrix} 0 \\ \ell_{83} \\ 0 \end{pmatrix} \begin{pmatrix} 0 & \ell_{83} & 0 \end{pmatrix}$$

$$- \begin{pmatrix} \ell_{64} \\ \ell_{84} \\ \ell_{94} \end{pmatrix} \begin{pmatrix} \ell_{64} & \ell_{84} & \ell_{94} \end{pmatrix} - \begin{pmatrix} \ell_{65} \\ \ell_{85} \\ 0 \end{pmatrix} \begin{pmatrix} \ell_{65} & \ell_{85} & 0 \end{pmatrix}$$

Row subscript set of $\overline{U}_j$ is $\{6, 8, 9\}$

$$F_6 = \begin{array}{c} 6 \\ 8 \\ 9 \end{array} \begin{pmatrix} a_{66} & 0 & a_{69} \\ 0 & 0 & 0 \\ a_{96} & 0 & 0 \end{pmatrix} + \overline{U}_6 =$$

$$\begin{pmatrix} a_{66} - \ell_{62}^2 - \ell_{64}^2 - \ell_{65}^2 & -\ell_{64}\ell_{84} - \ell_{65}\ell_{85} & a_{69} - \ell_{64}\ell_{94} \\ -\ell_{64}\ell_{84} - \ell_{65}\ell_{85} & -\ell_{83}^2 - \ell_{84}^2 - \ell_{85}^2 & -\ell_{84}\ell_{94} \\ a_{69} - \ell_{64}\ell_{94} & -\ell_{84}\ell_{94} & -\ell_{94}^2 \end{pmatrix}$$

Example 3 continues

One step of elimination on F_6 will give factor column for 6 and U_6.

$T[6] = \{2, 3, 4, 5, 6\}$ and off-diagonal non-zero subscripts are $\{8, 9\}$.

$$U_6 = -\begin{array}{c}8\\9\end{array}\begin{pmatrix}\ell_{83}\\0\end{pmatrix}\begin{pmatrix}\ell_{83} & 0\end{pmatrix} - \begin{pmatrix}\ell_{84}\\\ell_{94}\end{pmatrix}\begin{pmatrix}\ell_{84} & \ell_{94}\end{pmatrix}$$

$$-\begin{pmatrix}\ell_{85}\\0\end{pmatrix}\begin{pmatrix}\ell_{85} & 0\end{pmatrix} - \begin{pmatrix}\ell_{86}\\\ell_{96}\end{pmatrix}\begin{pmatrix}\ell_{86} & \ell_{96}\end{pmatrix}$$

U_j is called the update-matrix. It is to be send up the tree as the final contribution from the subtree $T[j]$ to the ancestors.

$$U_j = -\sum_{k \in T[j]} \begin{pmatrix} \ell_{i_1,k} \\ . \\ . \\ . \\ \ell_{i_r,k} \end{pmatrix} \begin{pmatrix} \ell_{i_1,k}. & . & . & \ell_{i_r,k} \end{pmatrix}$$

Don't get confused with $\overline{U}_j$ and U_j

- $\overline{U}_j$ is a basket containing U_c s from all the children of node j where c is a children node
- U_c relates to $T[c]$ while $\overline{U}_j$ relates to $T[j] - \{j\}$.

Postordering

How to manage the storage and generation of update matrices such that there is less storage overhead ?

Generation sequence in our previous elimination tree:

U_1
$U_1 \rightarrow U_2$
$U_1 \rightarrow U_2 \rightarrow U_3$
$U_1 \rightarrow U_3 \rightarrow U_4$
U_2 consumed by F_4

- Two matrix orderings are equivalent if the reordered matrices have identical filled graphs.
- A postordering is an equivalent ordering such that a node is numbered before its parent and nodes in every subtree are numbered consecutively.
- It is possible to have a LIFO stack of update matrices by doing postordering.
- Minimize the maximum stack height by more optimal postordering (depends on matrix sparsity).
- Further reduction in working storage by tree restructuring.

Supernodal Version

A supernode is a maximal set of contiguous nodes $\{j+1, ..., j+t\}$ such that

$$Adj_G(T[j]) = \{j+1, ..., j+t\} \cup Adj_G(T[j+t])$$

• A maximal block of contiguous columns with identical off-diagonal column structures.

Six supernodes for our example

$$\{1,2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7,8,9\}$$

• A supernodal elimination tree is constructed and equivalent Frontal and Update matrices are defined.

Advantages Revisited

- The partial factorization is done in full matrix form.
- Vector processing on dense frontal matrix.
- Good data locality.
- Efficient out of core scheme.
- Natural adaptability to parallelism.

Conclusion

We know Multifrontal method because:

- We know it is just a reorganization of submatrix Cholesky.
- We know that we achieve this by partial factorization of some matrix called Frontal matrix.
- We know that unlike submatrix Cholesky, in this the updates are collected in a matrix called Update Matrix and applied only when needed.
- We know that Elimination tree guides our method by flowing, merging and creating frontal and update matrices.
- We know what are the advantages of Multifrontal method like no indexing overhead, parallelization, out-of-core etc.