3.3. Block Tridiagonal Systems

- Example 2.4 produced the block tridiagonal system of the form

\[ Ax = b \]  \hspace{1cm} (1a)

or

\[
\begin{bmatrix}
A_1 & B_1 \\
C_2 & A_2 & B_2 \\
& \ddots & \ddots \\
& & C_n & A_n
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_n
\end{bmatrix}
= 
\begin{bmatrix}
b_1 \\
b_2 \\
\vdots \\
b_n
\end{bmatrix} \hspace{1cm} (1b)
\]

- \( A_k, C_k, B_k, k = 1 : n, \) are \( q \times q \)
- The blocks of Example 2.4 were symmetric

\[ A_j = 
\begin{bmatrix}
4 & -1 \\
-1 & 4 & -1 \\
& \ddots & \ddots \\
& & -1 & 4
\end{bmatrix}
\]

\[ B_j = C_j = 
\begin{bmatrix}
-1 \\
-1 \\
\ddots \\
-1
\end{bmatrix}
\]

- Do not assume symmetry \((B_j \neq C_j + 1)\)
  * But do not pivot (dangerous)
The Block Tridiagonal algorithm

- Assume a factorization of the form

\[
L = \begin{bmatrix}
I \\
L_2 & I \\
& \ddots \\
& & L_n & I \\
\end{bmatrix}, \quad U = \begin{bmatrix}
U_1 & B_1 \\
& U_2 & B_2 \\
& & \ddots & \ddots \\
& & & U_n \\
\end{bmatrix}
\]  

(2)

- Multiply

\[
U_1 = A_1
\]  

(3a)

\[
L_i U_{i-1} = C_i, \quad i = 2 : n
\]  

(3b)

\[
L_i B_{i-1} + U_i = A_i, \quad i = 2 : n
\]  

(3c)

- Factorization

\[
U_1 = A_1;
\]

for i = 2:n

Factor \( U_{i-1}^T \) and determine the rows of \( L_i \) from

\[
U_{i-1}^T L_i^T = C_i^T \]

by forward and backward substitution;

\[
U_i = A_i - L_i b_{i-1};
\]

end

- Forward and backward substitution: from (1, 2)

\[
y_1 = b_1
\]  

(4a)

\[
y_i = b_i - L_i y_{i-1}, \quad i = 2 : n
\]  

(4b)

\[
U_n x_n = y_n
\]  

(4c)

\[
U_i x_i = y_i - b_i x_{i+1}, \quad i = n - 1 : -1 : 1
\]  

(4d)

- Save \( U_i \) in factored form from (3b)
The Tridiagonal Algorithm

- Operation count
  - $2q^3/3$ FLOPs to factor $U_{i-1}$ in (3b)
  - $2q^3$ FLOPs to determine $L_i$ in (3b)
  - $2q^3$ FLOPs to determine $U_i$ in (3c)
  - Summing on $i$, we have $14nq^3/3$ multiplications to factor
  - The forward and backward substitution takes $O(nq^2)$ FLOPs
  - $U_i$ and $L_i$ are not tridiagonal even if $A_i$ is

- Band vs Block
  - Depends on the sizes of $p$ (the bandwidth) and $q$ (the block size)
  - For the 2D Poisson equation on an $N \times N$ grid (Example 2.4), $p \approx q$
    * Banded factorization has approximately $2N^4$ FLOPs
    * Block factorization has $n \approx N^2$ and $q \approx N$ for approximately $14N^5/3$ FLOPs
    * Some reductions in the block operation count are possible by accounting for the sparsity of $B_i$ and $C_i$
Two-Dimensional Poisson Equation

- Example 1. Consider Poisson’s equation on an \( N \times N \) grid (cf. Example 2.4)
  - Finite differencing for a \( 4 \times 4 \) problem
  - Row-by-row numbering with a single index

\[
\begin{bmatrix}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
1 & X & X & X & & & & & \\
2 & X & X & X & X & & & & \\
3 & X & X & X & X & & & & \\
4 & X & X & X & X & & & & \\
5 & X & X & X & X & & & & \\
6 & X & X & X & X & & & & \\
7 & X & X & X & & & & & \\
8 & X & X & X & & & & & \\
9 & X & X & X & & & & & \\
\end{bmatrix}
\]

- Write equations at the odd points first
  - Order the odd unknowns first

\[
\begin{bmatrix}
1 & 3 & 5 & 7 & 9 & 2 & 4 & 6 & 8 \\
1 & X & & & & X & & & \\
3 & X & & & & X & & & \\
5 & X & & & & X & & & \\
7 & X & & & & X & & & \\
9 & X & & & & X & & & \\
2 & X & X & X & & & & & \\
4 & X & X & X & & & & & \\
6 & X & X & X & & & & & \\
8 & X & X & X & & & & & \\
\end{bmatrix}
\]
Odd-Even Ordering

- The structure of the matrix with odd-even ordering is

\[
A = \begin{bmatrix}
D_1 & V_1 \\
V_2 & D_2
\end{bmatrix}
\]

- To solve

\[
\begin{bmatrix}
D_1 & V_1 \\
V_2 & D_2
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix} = \begin{bmatrix}
b_1 \\
b_2
\end{bmatrix}
\]

  - Factor

\[
\begin{bmatrix}
D_1 & V_1 \\
V_2 & D_2
\end{bmatrix} = \begin{bmatrix}
I & 0 \\
L_2 & I
\end{bmatrix}
\begin{bmatrix}
U_1 & B_1 \\
0 & U_2
\end{bmatrix}
\]

  - Expand the factorization

\[
U_1 = D_1, \quad B_1 = V_1 \\
L_2 = V_2 U_1^{-1}, \quad U_2 = D_2 - L_2 V_1
\]

- Forward and backward substitution

\[
y_1 = b_1, \quad y_2 = b_2 - V_2 U_1^{-1} y_1 \\
U_2 x_2 = y_2, \quad D_1 x_1 = y_1 - V_1 x_2
\]

- \( U_2 \) is penta-diagonal
Domain Decomposition

- “Dissection,” “substructuring,” and “domain decomposition”
  - Partition $A$ into three pieces so that
    * The unknowns $x_1$ are only connected to $x_3$
    * The unknowns $x_2$ are only connected to $x_3$
  - Occurs “naturally” in finite difference and finite element applications

- The structure of the system is
  \[
  Ax = \begin{bmatrix}
  A_{11} & 0 & A_{13} \\
  0 & A_{22} & A_{23} \\
  A_{13}^T & A_{23}^T & A_{33}
  \end{bmatrix}
  \begin{bmatrix}
  x_1 \\
  x_2 \\
  x_3
  \end{bmatrix}
  = \begin{bmatrix}
  b_1 \\
  b_2 \\
  b_3
  \end{bmatrix}
  \]  
  (5)

- Suppose $A$ is symmetric
  - Factor $A$ into $LDL^T$
    \[
    L = \begin{bmatrix}
    I & I \\
    \omega_{13}^T & \omega_{23}^T & I
    \end{bmatrix},
    \quad
    D = \begin{bmatrix}
    D_1 \\
    D_2 \\
    D_3
    \end{bmatrix}
    \]
Domain Decomposition

- Factorization procedure

function \([L, D] = \text{dd}(A)\)
% \text{dd}: block factorization of a matrix \(A\) into \(LDL^T\)

for \(k = 1:2\)
    \(D_k = A_{kk};\)
    Solve \(D_k \omega_k = A_k\) for \(\omega_k;\)
end
\(D_3 = A_{33} - \omega_{13}^T D_1 \omega_{13} - \omega_{23}^T D_2 \omega_{23};\)

- Note:
  
  i. \(D_1\) and \(D_2\) can be factored without knowledge of each other or \(D_3, \omega_{13},\) and \(\omega_{23}\)
  
  ii. The factorizations of \(D_1\) and \(D_2\) can be done in parallel
  
  iii. Design modifications only change those matrices affected
  
  iv. The procedure can be done recursively
  
  v. \(D_3\) is called the *Schur complement* of \(A\)

  - For \(N \times N\) grid problems, George (1973) \(^1\) shows that factorization takes \(O(N^3)\) FLOPs
  
  - Banded and profile techniques require \(O(N^4)\) FLOPs
  
  - Nested dissection has the optimal order of operations

---

Nested Dissection

- Two-dimensional nested dissection
  - Bisect the domain
  - Number unknowns on the fine level first
  - Number unknowns on the interface between two regions next
  - Number unknowns at the juncture of four regions last

- Nested ordering of a $4 \times 4$ mesh

```
1  2  3  4  5  6  7  8  9
1 X   X   X
2   X   X   X
3       X   X   X
4               X   X
5   X   X   X   X
6   X   X   X   X
7       X   X   X   X
8               X   X
9                        X   X
```