Parallel Solution of Symmetric Eigenvalue Problems

Zack

2/21/2014
• Typically, the eigenvalue problem is solved in three successive steps:
  – Reduction to tridiagonal form
  – Solution of the tridiagonal eigenproblem
  – Back transformation of the computed eigenvectors to those of the original matrix
• (2005) Compute all eigenpairs of a 15,000*15,000 matrix on 16 processor:
  – 546s for reduction
  – 22.2s for tridiagonal solution
  – 160s for backtransformation
Fig. 1. Algorithmic paths and modules for computing eigensystems of symmetric matrices.
Reduction to banded (full matrix)

- Householder transformation
- Original one-step tridiagonalization

- Reduction to Tridiagonal Form

\[
\begin{pmatrix}
  x & x & x & x & x \\
  x & x & x & x & x \\
  x & x & x & x & x \\
  x & x & x & x & x \\
  x & x & x & x & x \\
\end{pmatrix}
\rightarrow
\begin{pmatrix}
  x & x & x & x & x \\
  x & x & x & x & x \\
  x & x & x & x & x \\
  x & x & x & x & x \\
  x & x & x & x & x \\
\end{pmatrix}
\rightarrow
\begin{pmatrix}
  x & x \\
  x & x & x & x & x \\
  x & x & x & x & x \\
  x & x & x & x & x \\
  x & x & x & x & x \\
\end{pmatrix}
\]

with Householder rotation, we can reduce A to a tridiagonal matrix. This is called Householder Tridiagonalization Algorithm.
If we only reduce the matrix to banded form, the update can be done with a block orthogonal transformation, such as WY representation $Q = I + WY^T$ with width-$b$ matrices $W$ and $Y$, then the vast majority of the operations can be done with highly efficient BLAS-3 routines instead of memory bandwidth-limited BLAS-2 or BLAS-1.

**Fig. 2.** Partitioning and first stage in the blocked reduction to banded form. The diagonal blocks $D_k$ are $b \times b$. 
Band-to-tridiagonal

\[ Q_1^{(1)} \]

Step 0
Band-to-tridiagonal

Step 1

$\Gamma$

$Q_2^{(1)}$
Band-to-tridiagonal
Band-to-tridiagonal

Step 3
Band-to-tridiagonal

Step 4

$Q_1^{(3)}$

$Q_3^{(2)}$
Band-to-tridiagonal

Step 5

$Q_2^{(3)}$

$Q_4^{(2)}$

$\Gamma$

$\Gamma$
• Algorithm

1: wait for $Q_i$ and last column of $B_i$
2: compute $A_{i+1,i} Q_i$
3: generate and send $Q_{i+1}$
4: compute $Q_i^T A_{i,i} Q_i$
5: send first column of $B_i$
6: compute $Q_{i+1}^T A_{i+1,i}$
Reduction to banded (sparse matrix)

- Cuthill-McKee (CMK) ordering

*Fig. 8.* Nonzero pattern of the original `rap_rail_1357` matrix (left) and after reordering according to the Cuthill–McKee scheme (right).
With CMK method, we can reduce the band without flop computation
if some node has large degree, it is hard to reduce band to a small number with permutation.
• A possible way to deal with the nodes which has large degree:

1. move these nodes to bottom;
2. solve the eigen-update

\[
\begin{bmatrix}
A & b \\
b^T & c
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix} = \lambda
\begin{bmatrix}
x \\
y
\end{bmatrix}
\]

where A is banded matrix, which is easy to be decomposed, b is dense vector.
\[ \begin{bmatrix} A & b \\ b^T & c \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} x \\ y \end{bmatrix} \]

\[ \Rightarrow \begin{cases} Ax + by = \lambda x \\ b^T x + cy = \lambda y \end{cases} \]

\[ \Rightarrow Ax + \frac{bb^T}{\lambda - c} x = \lambda x \]

\[ \Rightarrow \left( A + \frac{bb^T}{\lambda - c} - \lambda I \right) x = 0 \]
• \[ \left( A + \frac{bb^T}{\lambda - c} - \lambda I \right) x = 0 \]

Suppose that \( A = EDE^T \)

\[
\text{det} \left( D + \frac{ZZ^T}{\lambda - c} - \lambda I \right) = 0
\]
• \( \det \left( D + \frac{zz^T}{\lambda - c} - \lambda I \right) = 0 \)

• Secular equation:

\[
1 + \sum_i \frac{z_i^2}{(d_i - \lambda)(\lambda - c)} = 0
\]

which is similar to rank-1 update
Band-to-tridiagonal (non-uniform)

Fig. 9. Non-uniform block structure for a $12 \times 12$ matrix.
Table 1
Flop estimates and (serial) tridiagonalization times for the uniform and non-uniform block structures with the Cuthill–McKee and reverse Cuthill–McKee orderings on one core of a 3 GHz Intel Core2 Quad Q9650 system with 4 GB of main memory.

<table>
<thead>
<tr>
<th>Method</th>
<th>Matrix size</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1357</td>
<td>5177</td>
<td>20,209</td>
<td>79,841</td>
</tr>
<tr>
<td><strong>Uniform block structure</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMK ordering and</td>
<td>Gflop</td>
<td>0.79</td>
<td>22.3</td>
<td>676</td>
<td>21,000</td>
</tr>
<tr>
<td>RCMK ordering</td>
<td>Time [s]</td>
<td>0.31</td>
<td>12.1</td>
<td>454</td>
<td>14,000</td>
</tr>
<tr>
<td><strong>Non-uniform block structure</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMK ordering</td>
<td>Gflop</td>
<td>0.26</td>
<td>7.2</td>
<td>214</td>
<td>6600</td>
</tr>
<tr>
<td></td>
<td>Time [s]</td>
<td>0.17</td>
<td>3.9</td>
<td>161</td>
<td>4800</td>
</tr>
<tr>
<td>RCMK ordering</td>
<td>Gflop</td>
<td>0.49</td>
<td>13.8</td>
<td>417</td>
<td>12,900</td>
</tr>
<tr>
<td></td>
<td>Time [s]</td>
<td>0.24</td>
<td>6.9</td>
<td>297</td>
<td>8900</td>
</tr>
</tbody>
</table>
# Tridiagonal eigensolver

<table>
<thead>
<tr>
<th>Method</th>
<th>Feature</th>
<th>Consequence</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combination of bisection and inverse iteration (B&amp;I)</td>
<td>Able to compute a subset of k eigenpairs at reduced cost</td>
<td>Lose orthogonality</td>
<td>$O(k^2n)$</td>
</tr>
<tr>
<td>MRRR</td>
<td></td>
<td></td>
<td>$O(kn)$</td>
</tr>
<tr>
<td>QR/QL</td>
<td>Designed to compute either all or no eigenvectors</td>
<td>Good eigensystem</td>
<td>$O(n^3)$</td>
</tr>
<tr>
<td>Divide-and-conquer (D&amp;C)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Divide-and-conquer algorithm

• Step 1, Split

the original tridiagonal matrix could be split into two half sized submatrices:

\[ T = \begin{pmatrix} T_1 \\ T_2 \end{pmatrix} + \rho w w^T, \]
• Step 2, solve subproblems
  when subproblems are small enough, call QR/QL method,
  
  \[ T_1 = Q_1 \Lambda_1 Q_1^T \quad T_2 = Q_2 \Lambda_2 Q_2^T \]

  then we have
  
  \[ T = Q_{\text{sub}} (\Lambda_{\text{sub}} + \rho zz^T) Q_{\text{sub}}^T \]

  where,
  
  \[ Q_{\text{sub}} = \text{diag} (Q_1, Q_2), \quad \Lambda_{\text{sub}} = \text{diag} (\Lambda_1, \Lambda_2), \quad \text{and} \quad z = Q_{\text{sub}}^T w \]
• Step 3, deflate

  (1) some components \( z_i \) of \( z \) are (almost) zero,

  (2) two elements in diagonal matrix are identical (or close), then zero elements can be generated in \( z \)

\[
T = Q_{\text{sub}}^{\text{RP}} \left( \Lambda_n^{\text{sub}} + \rho z_n z_n^T \begin{pmatrix} \Lambda_d^{\text{sub}} \\ \end{pmatrix} \right) (Q_{\text{sub}}^{\text{RP}})^T
\]
• Step 4, rank-1 update eigendecomposition
  
  – Solve secular equation $1 + \sigma \sum_{i=1}^{n} \frac{\zeta_i^2}{(d_i-\lambda)} = 0$

Fig. 1. Graph of the secular equation, $n = 5, \rho > 0$
Back transformation to eigenvectors

- Eigenvectors of $T$ $\rightarrow$ eigenvectors of $A$

- $E^{(A)} = H_1 H_2 \ldots H_p E^{(T)}$ $\quad$ --- $\quad$ $O(n^2 p)$
  - 1D data layout
  - 2D data layout
Fig. 6. 1D data layout (left) and 2D block data layout (right) for the back transformation of eigenvectors.
Fig. 1. Algorithmic paths and modules for computing eigensystems of symmetric matrices.
Thank you