For given \( m \times n \) matrix \( A \), with \( m > n \), QR factorization has form

\[
A = QR
\]

where matrix \( Q \) is \( m \times m \) and orthogonal, and \( R \) is \( n \times n \) and upper triangular.

QR factorization can be used to solve linear systems, least squares problems, etc.

As with Gaussian elimination, zeros are introduced successively into matrix \( A \), eventually reaching upper triangular form, but using orthogonal transformations instead of elementary eliminators.

**Householder Transformations** have form

\[
H = I - 2\frac{vv^T}{v^Tv}
\]

where \( v \) is a nonzero vector.

- From definition, \( H = H^T = H^{-1} \), so \( H \) is both orthogonal and symmetric.
- For given vector \( a \), choose \( v \) so that

\[
Hv = \begin{bmatrix}
\alpha \\
0 \\
\vdots \\
0
\end{bmatrix}
\]

where \( \alpha = \alpha_k e_1 \), with sign chosen to avoid cancellation.

To find \( \alpha_k \), we can take

\[
\alpha_k = -\text{sign}(a_{kk}) \sqrt{a_{kk}^2 + \cdots + a_{mk}^2}
\]

\[
v_k = \begin{bmatrix}
0 \\
\cdots \\
a_{kk} \\
\cdots \\
0
\end{bmatrix}
\]

If \( \beta_k = 0 \), continue with next \( k \).

**Parallel Householder QR**

- Householder QR factorization is similar to Gaussian elimination for LU factorization.
- Forming Householder vector \( v_k \) is analogous to computing multipliers in Gaussian elimination.
- Subsequent updating of remaining unreduced portion of matrix is also analogous to Gaussian elimination.
- Thus, parallel implementation is similar to parallel LU, but with Householder vectors broadcast horizontally instead of multipliers.
- For this reason, we will not go into details.
Givens Rotations

- **Givens rotation** operates on pair of rows to introduce single zero

- For given 2-vector \( \mathbf{a} = [a_1, a_2]^T \), if
  \[
  c = \frac{a_1}{\sqrt{a_1^2 + a_2^2}}, \quad s = \frac{a_2}{\sqrt{a_1^2 + a_2^2}}
  \]
  then
  \[
  G\mathbf{a} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} a_1 \\ s \end{bmatrix}
  \]

- Scalars \( c \) and \( s \) are cosine and sine of angle of rotation, and \( c^2 + s^2 = 1 \), so \( G \) is orthogonal

Parallel Givens QR Factorization

- With 1-D partitioning of \( A \) by columns, parallel implementation of Givens QR factorization is similar to parallel Householder QR factorization, with cosines and sines broadcast horizontally and each task updating its portion of relevant rows

- With 1-D partitioning of \( A \) by rows, broadcast of cosines and sines is unnecessary, but there is no parallelism unless multiple pairs of rows are processed simultaneously

- Fortunately, it is possible to process multiple pairs of rows simultaneously without interfering with each other

References


