BLOCK QR COLUMN PIVOTING

PRANAY SESHADRI*

1. Outline. The objective of this sketch is to offer some strategies for QR with column pivoting when the matrix $\boldsymbol{A}$ is too large to store in memory. The idea is that we can generate $r < n$ columns of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$.

1.1. Notes from Guintana-Orti et al. . The authors describe a new variant of the QRP algorithm that can employ Level 3 BLAS kernels. What seems to have kept the original QRP (Golub et al.) from using Level 3 BLAS kerneldes is the norm downdate scheme (step 4) – at every step we must downdate all column norms before we can select the next pivot column among the remaining ones. The formula for the norm downdate we used is not obviously numerically reliable, and G.W. Stewart developed a robust scehe,.

The normdowndate scheme has at least two noticeable features: (1) it makes the computation of column norms affordable and hence makes the column pivoting scheme practical, and (2) it governs the numerical aspects of the QRP procedure. Consulting Figure 1, we notice that in order to downdate the column norms after the jth step we need only to know the updated jth row. This allows us to choose the next pivot column, p. Next, to determine the next Householder transformation it is sufficient to apply the previous Householder transformations only to the pth column. The update of elements in other rows and columns can be delayed. This analysis underpins our block algorith: for every

---

Algorithm 1 Large least squares on problems involving kronecker products

---

Given square matrices `A1` and `A2` of sizes `order1` and `order2` respectively,

1. Compute the QR factorization with column pivoting for `A1` and `A2`: `[Q1,R1,P1] = qr(A1)`, `[Q2,R2,P2] = qr(A2)`;
2. Determine the size of `R1` and `R2`: `[rowsR2,colsR2] = size(R2)`, `[rowsR1,colsR1] = size(R1)`;
3. Reshape the model evaluations at the quadrature points: `B = reshape(b, [order2, order1])`;
4. Compute `K = Q2' * B * Q1`;
5. Determine the submatrix `K11 = K(1:rowsR2, 1:colsR1)`;
6. The least squares solution is then given by: `X = P2 * inv(R2) * K11 * inv(R1') * P1`.

---

*Research Associate, Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ, U.K., (`p.seshadri@eng.cam.ac.uk`)