Lecture 8

QR factorization

- Read 3.4.3 and 5.6.1 of the text.

- Definition 3.1 A matrix $A \in \mathbb{R}_{m\times n}$ with $m \geq n$ admits a QR factorization if there exists an orthogonal matrix $Q \in \mathbb{R}_{m\times m}$ and an upper trapezoidal matrix $R \in \mathbb{R}_{m\times n}$ with zero rows from the $(n+1)$-st row on such that

$$A = QR.$$

  This factorization can be constructed by three methods:

  1. Gram-Schmidt
  2. Householder
  3. Givens

- Property 3.3 (Reduced QR) Suppose the rank of $A \in \mathbb{R}_{m\times n}$ is n for which $A = QR$ is known. Then

$$A = \tilde{Q}\tilde{R}$$

  where $\tilde{Q}$ and $\tilde{R}$ are submatrices of Q and R given respectively by

$$\tilde{Q} = Q = Q(1:m, 1:n), \qquad \tilde{R} = R(1:n, 1:n).$$

  Moreover $\tilde{Q}$ has orthonormal columns and $\tilde{R}$ is upper triangular and coincides with the Cholesky factor H of the positive definite matrix A^TA, that is, $A^TA = \tilde{R}^T\tilde{R}$.

- Gram-Schmidt:

  Let $A = [a_1|a_2|\cdots|a_n] \in \mathbb{R}_{m\times n}$ where the columns are linearly independent. Set

$$\tilde{q}_1 = a_1/\|a_1\|_2$$

  and for $k = 1, \ldots, n-1$,

$$q_{k+1} = a_{k+1} - \sum_{j=1}^{k} (\tilde{q}_j^T a_{k+1})\tilde{q}_j \tag{1}$$

  and set

$$\tilde{q}_{k+1} = q_{k+1}/\|q_{k+1}\|_2.$$

  To recover $\tilde{Q}$ and $\tilde{R}$, rewrite (1) as

$$a_{k+1} = \|q_{k+1}\|_2 \tilde{q}_{k+1} + \sum_{j=1}^{k} (\tilde{q}_j^T a_{k+1})\tilde{q}_j$$

  So

$$\tilde{Q} = [\tilde{q}_1|\tilde{q}_2|\cdots|\tilde{q}_n]$$

  and $\tilde{R} \in \mathbb{R}_{n\times n}$ is upper triangular where

$$\tilde{r}_{j,k+1} = \tilde{q}_j^T a_{k+1}, \quad j = 1, \ldots, k,$$

  and

$$\tilde{r}_{k+1,k+1} = \|q_{k+1}\|_2, \quad \tilde{r}_{11} = \|a_1\|_2$$

- Gram-Schmidt as Triangular Orthogonalization

  The above algorithm means after all the steps, we get a product of triangular matrices

$$AR_1R_2\cdots R_n = \widetilde{Q}$$

  Set $\widetilde{R} = (R_1R_2\cdots R_n)^{-1}$.

- Disadvantage of (classical) Gram-Schmidt:

  Sensitive to rounding error (orthogonality of the computed vectors can be lost quickly or may even be completely lost) $\longrightarrow$ modified Gram-Schmidt.

  Example:

$$A = \begin{bmatrix} 1+\epsilon & 1 & 1 \\ 1 & 1+\epsilon & 1 \\ 1 & 1 & 1+\epsilon \end{bmatrix}$$

  with very small ϵ such that $3+2\epsilon$ will be computed accurately but $3+2\epsilon+\epsilon^2$ will be computed as $3+2\epsilon$. Then

$$Q \approx \begin{bmatrix} \frac{1+\epsilon}{\sqrt{3+2\epsilon}} & \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{3+2\epsilon}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{3+2\epsilon}} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}$$

  and $\cos\theta_{12} = \cos\theta_{13} \approx \pi/2$ but $\cos\theta_{23} \approx \pi/3$.

- Modified Gram-Schmidt

  The $k+1$st step (1) of CGS is replaced by a number of steps

$$
\begin{aligned}
a_{k+1}^{(1)} &= a_{k+1} - (\tilde{q}_1^T a_{k+1})\tilde{a}_1 \\
a_{k+1}^{(i+1)} &= a_{k+1}^{(i)} - (\tilde{q}_{i+1}^T a_{k+1}^{(i)})\tilde{q}_{i+1}, \quad i = 1, \dots, k-1.
\end{aligned}
$$

  Theoretically

$$a_{k+1}^{(k)} = q_{k+1}.$$

- flop count is about mn^2 for CGS, $2mn^2$ for MGS.

- From a numerical point of view, both these CGS and MGS may produce a set of vectors which is far from orthogonal and sometimes the orthogonality can be lost completely.

  Loss of orthogonality:

$$
\begin{aligned}
\|I - \hat{Q}^T\hat{Q}\| &\propto K(A)u \qquad \text{(for MGS)} \\
\|I - \hat{Q}^T\hat{Q}\| &\propto K^2(A)u \qquad \text{(for CGS)}
\end{aligned}
$$

  provided that the matrix A^TA is numerically nonsingular

- For a numerically nonsingular matrix A the loss of orthogonality in MGS occurs in a predictable way and it can be bounded by a term proportional to the condition number $K(A)$ and to the roundoff unit u. Therefore, the loss of orthogonality of computed vectors is close to roundoff unit level only for well-conditioned matrices, while for ill-conditioned matrices it can be much larger leading to complete loss (the loss of linear independence) for numerically singular or rank-deficient problems.

- MGS method can be used to solve least squares problems and that the algorithm is backward-stable.

- CGS resurfaces in some recent articles, especially regarding its usefulness because it takes advantage of BLAS2. Practically a better candidate for parallel implementation than MGS.

- mod_grams (Modified Gram-Schmidt method)

```
function [Q,R] = mod_grams(A)
[m,n]=size(A);
Q=zeros(m,n); Q(1:m,1) = A(1:m,1); R=zeros(n); R(1,1)=1;
for k = 1:n
 R(k,k) = norm (A(1:m,k)); Q(1:m,k) = A(1:m,k)/R(k,k);
 for j=k+1:n
    R (k,j) = Q (1:m,k)' * A(1:m,j);
    A (1:m,j) = A (1:m,j) - Q(1:m,k)*R(k,j);
 end
end
```

- Impossible to overwrite QR factorization on A. The matrix $\tilde{R}$ is overwritten on A and $\tilde{Q}$ is stored separately.

- Compare CGS and MGS for the vectors $a_1 = (1, \epsilon, 0, 0)^T$, $a_2 = (1, 0, \epsilon, 0)^T$, $a_3 = (1, 0, 0, \epsilon)^T$, where ϵ is so small $1 + \epsilon^2 \approx 1$.

Householder reflections and Givens rotations

Householder QR

- Householder QR = Orthogonal triangularization

$$\underset{A}{\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}} \overset{P_{(1)}}{\longrightarrow} \underset{P_{(1)}A}{\begin{bmatrix} \mathtt{x} & \mathtt{x} & \mathtt{x} \\ 0 & \mathtt{x} & \mathtt{x} \\ 0 & \mathtt{x} & \mathtt{x} \\ 0 & \mathtt{x} & \mathtt{x} \\ 0 & \mathtt{x} & \mathtt{x} \end{bmatrix}} \overset{P_{(2)}}{\longrightarrow} \underset{P_{(2)}P_{(1)}A}{\begin{bmatrix} \times & \times & \times \\ & \mathbf{x} & \mathbf{x} \\ & 0 & \mathbf{x} \\ & 0 & \mathbf{x} \\ & 0 & \mathbf{x} \end{bmatrix}} \rightarrow \cdots$$

- After all the steps,

$$P_{(n)} \cdots P_{(2)} P_{(1)} A = R \text{ if } m > n$$

and

$$P_{(n-1)} \cdots P_{(2)} P_{(1)} A = R \text{ if } m = n$$

Then

$$Q = (P_{(n)} \cdots P_{(2)} P_{(1)})^{-1} = P_{(1)}^{-1} P_{(2)}^{-1} \cdots P_{(n)}^{-1} = P_{(1)} \cdots P_{(n)} \text{ if } m > n$$

and

$$Q = (P_{(n-1)} \cdots P_{(2)} P_{(1)})^{-1} = P_{(1)}^{-1} P_{(2)}^{-1} \cdots P_{(n-1)}^{-1} = P_{(1)} \cdots P_{(n-1)} \text{ if } m = n$$

since $P_{(i)}^{-1} = P_{(i)}^T = P_{(i)}$ as each $P_{(i)}$ is a Householder reflection matrix.

Householder reflections

- The Householder reflection
$$P = I - 2vv^T/\|v\|_2^2$$
sends x to $y = Px$ which is the reflection of x with respect to the hyperplane span $v^{\perp}$: $Pv = -v$, $Pu = u$ whenever $u \perp v$.

- $P^2 = P$, $P^T = P$

- Householder reflection can be used to set to zero a block of components of a given $x \in \mathbb{R}^n$: Set
$$v = x \pm \|x\|_2 e_m$$
where $e_m = (0, \ldots, 1, 0, \ldots, 0)^T \in \mathbb{R}^n$ in which the 1 appears in the mth component. Then
$$Px = \pm\|x\|_2 e_m$$

- Let $P_{(k)}$ be the form
$$P_{(k)} = \begin{bmatrix} I_{k-1} & 0 \\ 0 & R_{n-k} \end{bmatrix} \in \mathbb{R}_{n\times n}$$
where
$$R_{n-k}x^{(n-k)} = \begin{bmatrix} \|x^{(n-k)}\| \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \|x^{(n-k)}\| e_1^{(n-k)},$$
where $x^{(n-k)} \in \mathbb{R}^{n-k}$ is the vector formed by the last $n-k$ compoenents of x and $e_1^{(n-k)}$ is the first standard unit vector of $\mathbb{R}^{n-k}$.

- $$R_{n-k} = I_{n-k} - \frac{2w^{(k)}(w^{(k)})^T}{\|w^{(k)}\|_2^2}, \quad w^{(k)} = x^{(n-k)} \pm \|x^{(n-k)}\|_2 e_1^{(n-k)}$$

- $Q = P_{(n)}P_{(n-1)} \cdots P_{(1)}$ if $m > n$ and $Q = P_{(n-1)} \cdots P_{(1)}$ if $m = n$.

- Read p.208-209

Choice of Householder reflection

- It is convenient to choose the minus sign in $w^{(k)} = x^{(n-k)} \pm \|x^{(n-k)}\|_2 e_1^{(n-k)}$ so that $R_{n-k}x^{(n-k)}$ is a positive multiple of $e_1^{(n-k)}$.

- If $x_{k+1} > 0$ where $x^{(n-k)} = (x_{k+1}, \ldots, x_n)^T$, in order to avoid numerical cancellations, rationalization is used:

$$w_1^{(k)} = \frac{x_{k+1}^2 - \|x^{(n-k)}\|_2}{x_{k+1} + \|x^{(n-k)}\|_2} = \frac{-\sum_{j=k+2}^{n} x_j^2}{x_{k+1} + \|x^{(n-k)}\|_2}$$

- Program 32 `vhouse`: Construction of the Householder vector

```
function [v,beta]=vhouse(x)
n=length(x);
x=x/norm(x);
s=x(2:n)'*x(2:n);
v=[1; x(2:n)];
if (s==0), beta=0;
else
   mu=sqrt(x(1)^2+s);
   if (x(1) <= 0)
      v(1)=x(1)-mu;
   else
      v(1)=-s/(x(1)+mu);
   end
   beta=2*v(1)^2/(s+v(1)^2);
   v=v/v(1);
end
return
```

- Read p.216-217

Givens rotations

- Alternative to Householder reflections

- A Givens rotation is simply a rotation

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

  rotates $x \in \mathbb{R}^2$ by θ.

- We can choose $\theta \in \mathbb{R}$ so that

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_i \\ x_j \end{bmatrix} = \begin{bmatrix} \sqrt{x_i^2 + x_j^2} \\ 0 \end{bmatrix},$$

$$\cos\theta = \frac{x_i}{\sqrt{x_i^2 + x_j^2}}, \quad \sin\theta = \frac{-x_j}{\sqrt{x_i^2 + x_j^2}}.$$

- Read p.209-230

Givens QR

- Zero things bottom-up and left-right.

$$\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \overset{(3,4)}{\longrightarrow} \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} \end{bmatrix} \overset{(2,3)}{\longrightarrow} \begin{bmatrix} \times & \times & \times \\ \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} \\ & \times & \times \end{bmatrix} \overset{(1,2)}{\longrightarrow}$$

$$\begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} \\ & \times & \times \\ & \times & \times \end{bmatrix} \overset{(3,4)}{\longrightarrow} \begin{bmatrix} \times & \times & \times \\ & \times & \times \\ & \mathbf{x} & \mathbf{x} \\ & 0 & \mathbf{x} \end{bmatrix} \overset{(2,3)}{\longrightarrow} \begin{bmatrix} \times & \times & \times \\ & \mathbf{x} & \mathbf{x} \\ & 0 & \mathbf{x} \\ & & \times \end{bmatrix} \overset{(3,4)}{\longrightarrow} R$$

- flop count $3nm^2 - m^3$ (about 50% more than Householder QR)

Stability

- $A \mapsto QA$ where Q is Householder reflection or Givens rotation:

$$fl(QA) = Q(A + \delta A)$$

  where $\|\delta A\| / \|A\|_2$ is tiny. Thus the computation of QA is normwise backward stable.

- MATLAB's command `[Q,R]=qr(A,0)` which uses Householder reflections.

- Read p.213