Lecture # 8
The Givens Q–R Factorizations and Updates

Givens Q–R Factorization

Based upon 2×2 transformations.

$$\begin{aligned}
\mathbf{x} &= \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \\
\begin{pmatrix} c & s \\ -s & c \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} &= \begin{pmatrix} \rho \\ 0 \end{pmatrix} \\
c^2 + s^2 &= 1, \quad c = \cos\,\theta, s = \sin\,\theta \\
c &= x_1/\rho, s = x_2/\rho, \quad \rho = \pm\| \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \|_2 .
\end{aligned}$$

Geometrically, it rotates a vector through an angle θ.

You are free to choose the sign of ρ any way you wish, but the BLAS implementation (and the MATLAB function Givens) choose ρ so that

$$\rho = \text{sign}(x_{max})\| \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \|_2$$

where x_{max} is the one of x_1 and x_2 of greatest magnitude. A proper implementation of this is as follows.

```
$absx1 = |x_1|$; $absx2 = |x_2|$;
if $absx1 \geq absx2$
   $alpha = \sqrt{1 + (x_2/x_1)^2}$;
   $c = 1/\alpha$; $s = x_2 * c/x_1$; $\rho = x_1 * alpha$;
else
   $alpha = \sqrt{1 + (x_1/x_2)^2}$;
   $s = 1/\alpha$; $c = x_1 * s/x_2$; $\rho = x_2 * alpha$;
end;
```

This formulation is also resistant to underflow and overflow.

A rotation on rows i and j has the form

$$G = \begin{array}{c} \\ i-1 \\ 1 \\ j-i-1 \\ 1 \\ n-j \end{array} \!\!\begin{array}{c} \begin{array}{ccccc} i-1 & 1 & j-i-1 & 1 & n-j \end{array} \\ \left(\begin{array}{ccccc} I_{i-1} & 0 & 0 & 0 & 0 \\ 0 & c & 0 & s & 0 \\ 0 & 0 & I_{j-i-1} & 0 & 0 \\ 0 & -s & 0 & c & 0 \\ 0 & 0 & 0 & 0 & I_{n-j} \end{array} \right) \end{array}.$$

If

$$B = GA$$

and

$$A = \begin{pmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_n^T \end{pmatrix}, \quad B = \begin{pmatrix} \mathbf{b}_1^T \\ \mathbf{b}_2^T \\ \vdots \\ \mathbf{b}_n^T \end{pmatrix},$$

then

$$\begin{aligned} \mathbf{b}_i &= c\mathbf{a}_i + s\mathbf{a}_j \\ \mathbf{b}_j &= -s\mathbf{a}_i + c\mathbf{a}_j \end{aligned}$$

We now show one way to do a Givens Q–R decomposition of a 4 × 3 matrix.

$$\begin{array}{cccccc} G_{21} & \rightarrow & x & x & x \\ & \rightarrow & x & x & x \\ & & x & x & x \\ & & x & x & x \end{array} \quad \begin{array}{cccccc} G_{31} & \rightarrow & x & x & x \\ & & 0 & x & x \\ & \rightarrow & x & x & x \\ & & x & x & x \end{array} \quad \begin{array}{cccccc} G_{41} & \rightarrow & x & x & x \\ & & 0 & x & x \\ & & 0 & x & x \\ & \rightarrow & x & x & x \end{array}$$

$$\begin{array}{ccccc} & & x & x & x \\ G_{32} & \rightarrow & & x & x \\ & \rightarrow & & x & x \\ & & & x & x \end{array} \quad \begin{array}{ccccc} & & x & x & x \\ G_{42} & \rightarrow & & x & x \\ & & & 0 & x \\ & \rightarrow & & x & x \end{array} \quad \begin{array}{ccccc} & & x & x & x \\ G_{42} & & & x & x \\ G_{43} & \rightarrow & & & x \\ & \rightarrow & & & x \end{array}$$

$$\begin{array}{ccc} x & x & x \\ & x & x \\ & & x \end{array}$$

We have that

$$\begin{aligned} Q^T &= G_{43}G_{42}G_{32}G_{41}G_{31}G_{21} \\ Q &= G_{21}^T G_{31}^T G_{41}^T G_{32}^T G_{42}^T G_{43}^T \end{aligned}$$

Properties

- Same orthogonality and error analysis properties as Householder, but twice the number of multiplications.
- Various tricks such as "fast Givens" roatations to reduce the operation count – not that effective in practice.
- Used mostly for special cases, sparse matrices, structured problems, etc.

In fact, we note that (from your first assignment)

$$H = \begin{pmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{pmatrix}$$

is a Householder transformation and can be substituted for Givens in any of the algorithms given in these note. However, note that if we take the limit as $\theta \to 0$ (no rotation), then

$$\lim_{\theta\to 0} G(\theta) = \lim_{\theta\to 0} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

whereas the corresponding set of Householder transformations has the limit

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

In some eigenvalue computations, the Givens limiting property is more useful!

One special use is adding a row to a Q–R factoriztion. Let

$$X = Q \begin{pmatrix} R \\ 0 \end{pmatrix} = Q_1 R$$

and let

$$\hat{X} = \begin{pmatrix} X \\ \mathbf{x}_{m+1}^T \end{pmatrix}.$$

Then

$$\hat{X} = \begin{pmatrix} Q_1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R \\ \mathbf{x}_{m+1}^T \end{pmatrix}.$$

Again this is best illustrated with a picture.

$$
\begin{array}{cccccc}
G_1 & \rightarrow & r & r & r & r \\
 & & & r & r & r \\
 & & & & r & r \\
 & & & & & r \\
 & \rightarrow & x & x & x & x
\end{array}
\qquad
\begin{array}{cccccc}
 & & r & r & r & r \\
G_2 & \rightarrow & & r & r & r \\
 & & & & r & r \\
 & & & & & r \\
 & \rightarrow & 0 & x & x & x
\end{array}
\qquad
\begin{array}{cccccc}
 & & r & r & r & r \\
 & & & r & r & r \\
G_3 & \rightarrow & & & r & r \\
 & & & & & r \\
 & \rightarrow & 0 & 0 & x & x
\end{array}
$$

$$
\begin{array}{cccccc}
 & & r & r & r & r \\
 & & & r & r & r \\
 & & & & r & r \\
G_4 & \rightarrow & & & & r \\
 & \rightarrow & 0 & 0 & 0 & x
\end{array}
$$

$$
\begin{array}{cccc}
r & r & r & r \\
 & r & r & r \\
 & & r & r \\
 & & & r
\end{array}
$$

Thus

$$
\begin{aligned}
\begin{pmatrix} \hat{R} \\ 0 \end{pmatrix} &= G_n \cdots G_1 \begin{pmatrix} R \\ \mathbf{x}_{m+1}^T \end{pmatrix} \\
\hat{Q}_1 &= \begin{pmatrix} Q_1 & 0 \\ 0 & 1 \end{pmatrix} G_1^T \cdots G_n^T
\end{aligned}
$$

produces the updated factorization

$$\hat{X} = \hat{Q}_1 \hat{R}.$$

Notice that explicit storage of Q_1 is necessary for this for this scheme to work.

Adding a column to Q–R factorization (equivalent to adding a variable to a regression) is similar. Let

$$\hat{X} = \begin{pmatrix} X & \mathbf{x}_{n+1} \end{pmatrix}.$$

There are different ways depending upon whether Householder or Gram–Schmidt is used to construct the factorization. If it is Householder, then

$$Q = H_1 \cdots H_n$$

and

$$Q^T\hat{X} = \begin{pmatrix} R & \mathbf{f}_1 \\ 0 & \mathbf{f}_2 \end{pmatrix}.$$

We choose $\tilde{H}_{n+1}$ such that

$$\begin{aligned}
\tilde{H}_{n+1}\mathbf{f}_2 &= r_{n+1,n+1}\mathbf{e}_1 \\
H_{n+1} &= \begin{pmatrix} I_n & 0 \\ 0 & \tilde{H}_{n+1} \end{pmatrix} \\
H_{n+1}Q^T X &= \begin{array}{c} \\ n \\ 1 \\ 0 \end{array}\!\!\begin{array}{c} \begin{array}{cc} \;\;n\;\; & \;\;\;1\;\;\; \end{array} \\ \begin{pmatrix} R & \mathbf{f}_1 \\ 0 & r_{n+1,n+1} \\ 0 & 0 \end{pmatrix} \end{array} = \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix} \\
\hat{Q} &= QH_{n+1} = H_1 \cdots H_{n+1}
\end{aligned}$$

In other words, it is just like adding an extra column (as long as $m \geq n+1$).

Modified Gram–Schmidt We have that

$$\hat{X} = \begin{pmatrix} Q_1 R & \mathbf{x}_{n+1} \end{pmatrix}$$

We treat $\mathbf{x}_{n+1}$ much the same as a right hand side. We do the extra procedure

```
$\mathbf{s} = \mathbf{x}_{n+1}$;
for$j = 1{:}\,n$
    $r_{j,n+1} = \mathbf{q}_j^T\mathbf{s}$;
    $\mathbf{s} = \mathbf{s} - r_{j,n+1}\mathbf{q}_j$;
end;
$r_{n+1,n+1} = \|\mathbf{y}\|_2$; $\mathbf{q}_{n+1} = \mathbf{y}/r_{n+1,n+1}$;
$\hat{X} = \hat{Q}_1\hat{R}$;
$\hat{Q} = \begin{pmatrix} Q_1 & \mathbf{q}_{n+1} \end{pmatrix} \begin{pmatrix} R & \mathbf{f}_1 \\ 0 & r_{n+1,n+1} \end{pmatrix}$;
$\mathbf{f}_1 = (r_{1,n+1}, \ldots, r_{n,n+1})^T$;
```

If we delete a column (get rid of a variable), we also use Givens rotations!! For that, suppose we have a Gram–Schmidt factorization of a 5×4 matrix X and we decide to delete the $3rd$ column of X.

Then

$$X = Q_1 R = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5)$$

and

$$\hat{X} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_5) = Q_1 \bar{R}$$

where $\bar{R}$ has the shape

$$\begin{array}{cccc} r & r & r & r \\ & r & r & r \\ & & r & r \\ & & r & r \\ & & & r \end{array}$$

Two Givens rotations will give us an upper triangular matrix again.

$$\begin{array}{cc} & \\ & \\ G_3 & \rightarrow \\ & \rightarrow \\ & \end{array} \quad \begin{array}{cccc} r & r & r & r \\ & r & r & r \\ & & r & r \\ & & r & r \\ & & & r \end{array} \quad \begin{array}{cc} & \\ & \\ & \\ G_4 & \rightarrow \\ & \rightarrow \end{array} \quad \begin{array}{cccc} r & r & r & r \\ & r & r & r \\ & & r & r \\ & & & r \\ & & & r \end{array} \quad \begin{array}{cccc} r & r & r & r \\ & r & r & r \\ & & r & r \\ & & & r \end{array}.$$

Thus deleting column j generates the operations

$$G_{n-1} \cdots G_j \bar{R} = \begin{array}{c} n-1 \\ 1 \end{array} \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix}.$$

The matrix Q_1 is updated by

$$\bar{Q}_1 = Q_1 G_j^T \cdots G_{n-1}^T.$$

The new factorzation is

$$\hat{X} = \bar{Q}_1 \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix} = \hat{Q}_1 \hat{R}$$

where $\hat{Q}_1 = \bar{Q}_1(:, 1\colon n-1)$.

Deleting a row is very subtle and very delicate numerically. It is a topic that is suited for a graduate seminar. Last year, your instructor got a paper out of it. If you are a glutton for punishment, here it is.

J.L. Barlow, A. Smoktunowicz, and H. Erbay, Improved Gram–Schmidt Downdating, BIT, (2005) 45:259–285.

But, yes, it involves Givens rotations!!