STABILIZED ORTHOGONAL FACTORIZATIONS

1. Conditioning of linear algebra problems

Recall that the relative condition number of a mathematical problem $f: X \to Y$ characterizes the amplification by f of perturbations in its argument

$$\kappa = \lim_{\varepsilon \to 0} \sup_{\|\delta x\| \leqslant \varepsilon} \left(\frac{\|f(x+\delta x) - f(x)\|}{\|f(x)\|} / \frac{\|\delta x\|}{\|x\|} \right).$$

Linear combination. The basic operation of linear combination $\boldsymbol{A}\boldsymbol{x}, \boldsymbol{A} \in \mathbb{C}^{m \times n}$, seen as the problem $\mathbb{C}^n \xrightarrow{f} \mathbb{C}^m$ has the condition number

$$\kappa = \sup_{\delta x} \left(\frac{\|\boldsymbol{A}\,\delta\boldsymbol{x}\|}{\|\boldsymbol{A}\boldsymbol{x}\|} / \frac{\|\delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \right) = \sup_{\delta x} \left(\frac{\|\boldsymbol{A}\,\delta\boldsymbol{x}\|}{\|\delta\boldsymbol{x}\|} \right) \frac{\|\boldsymbol{x}\|}{\|\boldsymbol{A}\boldsymbol{x}\|} = \|\boldsymbol{A}\| \frac{\|\boldsymbol{x}\|}{\|\boldsymbol{A}\boldsymbol{x}\|}.$$

The matrix norm property $\|\boldsymbol{A}\boldsymbol{y}\| \leqslant \|\boldsymbol{A}\|\,\|\boldsymbol{y}\|$ can be used to obtain

$$\|\boldsymbol{x}\| = \|\boldsymbol{I}_n \boldsymbol{x}\| = \|\boldsymbol{A}^+ \boldsymbol{A}\boldsymbol{x}\| \leqslant \|\boldsymbol{A}^+\|\,\|\boldsymbol{A}\boldsymbol{x}\| \Rightarrow \frac{\|\boldsymbol{x}\|}{\|\boldsymbol{A}\boldsymbol{x}\|} \leqslant \|\boldsymbol{A}^+\|$$

leading to

$$\kappa \leqslant \|\boldsymbol{A}\|\,\|\boldsymbol{A}^+\| = \kappa(\boldsymbol{A}),$$

where $\kappa(\boldsymbol{A})$ is the condition number of the matrix $\boldsymbol{A}$. If $\boldsymbol{A}$ is of full rank with $m > n$, the 2-norm condition number is given by the ratio of largest to smallest singular values.

$$\|\boldsymbol{A}\| = \sigma_1, \|\boldsymbol{A}^+\| = 1/\sigma_n \Rightarrow \kappa(\boldsymbol{A}) = \sigma_1/\sigma_n \geqslant 1.$$

By convention, if $\boldsymbol{A}$ is singular, the condition number $\kappa(\boldsymbol{A}) = \infty$.

Coordinate transformation. For $\boldsymbol{A} \in \mathbb{C}^{m \times m}$ of full rank, the coordinates of vector $\boldsymbol{b} \in \mathbb{C}^m$ expressed in the $\boldsymbol{I}$ basis can be transformed its coordinates $\boldsymbol{x} \in \mathbb{C}^m$ in the $\boldsymbol{A}$ basis by solving the linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{I}\boldsymbol{b}$, with the solution $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$ (so written formally, even though the inverse is almost never explicitly computed). This is simply another linear combination of the columns of $\boldsymbol{A}^{-1}$, hence the problem $\boldsymbol{f}: \mathbb{C}^m \to C^m, \boldsymbol{f}(\boldsymbol{b}) = \boldsymbol{A}^{-1}\boldsymbol{b}$ again has a condition number bounded by the condition number of the matrix $\boldsymbol{A}$.

$$\kappa \leqslant \|\boldsymbol{A}^{-1}\|\,\|\boldsymbol{A}\| = \kappa(\boldsymbol{A}) = \kappa(\boldsymbol{A}^{-1}).$$

Operator perturbation. Instead of changing the input data as above, the linear mapping represented by the matrix $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ might itself be perturbed. Two mathematical problems may now be formulated:

1. For fixed $\boldsymbol{b} \in \mathbb{C}^m, \boldsymbol{f}: \mathbb{C}^{m \times n} \to \mathbb{C}^n, \boldsymbol{f}(A, \boldsymbol{b}) = \boldsymbol{A}^+\boldsymbol{b} = \boldsymbol{x}$. Perturbation of the input $\boldsymbol{A}$ induces perturbation of $\boldsymbol{x}$ in order for $\boldsymbol{b}$ to be kept fixed

   $$(\boldsymbol{A} + \delta\boldsymbol{A})(\boldsymbol{x} + \delta\boldsymbol{x}) = \boldsymbol{b}.$$

   Using $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, and assuming that $\delta A\,\delta\boldsymbol{x}$ is negligible gives

   $$\boldsymbol{A}\,\delta\boldsymbol{x} + \delta\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0} \Rightarrow \delta\boldsymbol{x} = -\boldsymbol{A}^+\,\delta\boldsymbol{A}\boldsymbol{x},$$

   hence the relative condition number is

   $$\kappa = \frac{\|\boldsymbol{A}^+\,\delta\boldsymbol{A}\,\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \cdot \frac{\|\boldsymbol{A}\|}{\|\delta\boldsymbol{A}\|} \leqslant \frac{\|\boldsymbol{A}^+\|\,\|\delta\boldsymbol{A}\,\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \cdot \frac{\|\boldsymbol{A}\|}{\|\delta\boldsymbol{A}\|} \leqslant \frac{\|\boldsymbol{A}^+\|\,\|\delta\boldsymbol{A}\|\,\|\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \cdot \frac{\|\boldsymbol{A}\|}{\|\delta\boldsymbol{A}\|} = \kappa(\boldsymbol{A}).$$

For all above linear algebra problems the condition number is bounded by the associated matrix condition number. Unitary matrices $\boldsymbol{Q} \in \mathbb{C}^{m \times m}$ have $\kappa(\boldsymbol{Q}) = 1$, and define an orthonormal basis for $\mathbb{C}^m$. A rank-deficient matrix $\boldsymbol{Z} \in \mathbb{C}^{m \times m}$ has $\kappa(\boldsymbol{Z}) = \infty$, and corresponds to a linearly dependent vector set $\{z_1, \ldots, z_m\}$. The behavior of many numerical approximation procedures based upon linear combinations is determined by condition number of the basis set.

- *Monomial basis with uniform sampling*. Sampling the monomial basis on interval $[a, b]$ at $t_i = ih + a, i = 0, m$, $h = (b - a)/(m - 1)$ leads to the Vandermonde matrix

  $$\boldsymbol{V} = [\ \boldsymbol{1}\ \ \boldsymbol{t}\ \ \ldots\ \ \boldsymbol{t}^m\ ],$$

an extremely ill-conditioned matrix (Fig.). This can readily be surmised from the example $a=0$, $b=1$, in which case for large m the last columns of $\boldsymbol{V}$ become ever more colinear to the same $\boldsymbol{e}_m$ vector. Series expansions based on the monomials such as the Taylor series

$$f(t)=f(0)+f'(0)t+\cdots+\frac{f^{(n)}(0)}{n!}t^n+\cdots$$

are highly sensitive to pertubations, small changes in $f(t)$ lead to large changes in the coordinates $\{f(0),f'(0),\ldots\}$.

```
∴ function Vandermonde(a,b,m)
    t=LinRange(a,b,m); v=ones(m,1); V=copy(v)
    for j=2:m
      v = v .* t; V=[V v]
    end
    return V
  end;
```

```
∴
```

- *Monomial basis with Chebyshev sampling*. Changing the sampling so that points are clustered towards the interval endpoints reduces the condition number at fixed number of sampling points m, but the same limiting behavior for large m is obtained.

```
∴ function VandermondeC(m)
    δ=π/(2*m); Θ=LinRange(δ,π-δ,m)
    t=cos.(Θ)
    v=ones(m,1); V=copy(v)
    for j=2:m
      v = v .* t; V=[V v]
    end
    return V
  end;
```

```
∴
```

- *Triangular basis with uniform sampling*. LU-factorization of the monomial basis leads to a different family of polynomials, known as a triangular basis

$$\{1,t-x_1,(t-x_1)\cdot(t-x_2),\ldots,(t-x_1)\cdot\ldots\cdot(t-x_{m-1})\},$$

where $\{x_1,\ldots,x_m\}$ are known as the nodes of the system. These can be chosen to uniformly sample an interval. As to be expected, applying a sequence of non-unitary linear transformations onto an ill-conditioned basis yields even worse conditioning.

```
∴ function Triangular(a,b,m)
    x=LinRange(a,b,m); T=ones(m,1); Tj=copy(T); t=copy(x)
    for j=2:m
      Tj = Tj .* (t .- x[j-1]); T=[T Tj]
    end
    return T
  end;
```

```
∴
```

- *Triangular basis with Chebyshev sampling*. Adopting Chebyshev sampling ameliorates the conditioning, but only marginally.

```
∴ function TriangularC(m)
    δ=π/(2*m); Θ=LinRange(δ,π-δ,m)
    x=cos.(Θ); T=ones(m,1); Tj=copy(T); t=copy(x)
    for j=2:m
      Tj = Tj .* (t .- x[j-1]); T=[T Tj]
    end
    return T
  end;
```

```
∴
```

•

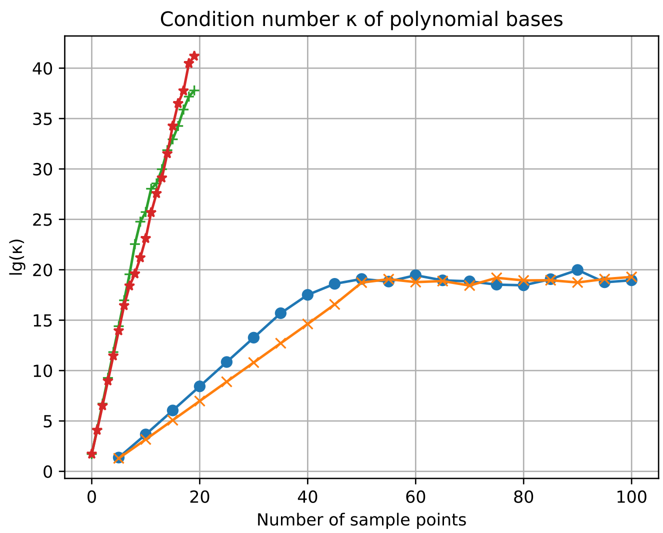

Figure 1. Monomial basis with: (o) uniform sampling, (x) Chebyshev sampling. Triangular basis with: (+) uniform sampling, (*) Chebyshev sampling.

```
∴ mr=5:5:100; κVDMU=log10.(cond.(Vandermonde.(-1,1,mr)));
∴ κVDMC=log10.(cond.(VandermondeC.(mr)));
∴ κTU=log10.(cond.(Triangular.(-1,1,mr)));
∴ κTC=log10.(cond.(TriangularC.(mr)));
∴
```

```
∴ x=collect(mr); clf();
∴ plot(x,κVDMU,"o-",x,κVDMC,"x-",κTU,"+-",κTC,"*-");
∴ grid("on"); title("Condition␣number␣κ␣of␣polynomial␣bases");
∴ xlabel("Number␣of␣sample␣points"); ylabel("lg(κ)");
∴ pre=homedir()*"/courses/MATH661/images/";
∴ savefig(pre*"PolyBasesCondNr.eps");
∴
```

2. Orthogonal factorization through Householder reflectors

The Gram-Schmidt procedure constructs an orthogonal factorization by linear combinations of the column vectors of $\boldsymbol{A} \in \mathbb{C}^{m\times n}$, $m \geqslant n$, $\operatorname{rank}(\boldsymbol{A}) = n$

$$\boldsymbol{A}\boldsymbol{R}_1\boldsymbol{R}_2\ldots\boldsymbol{R}_n = \boldsymbol{Q} \Rightarrow \boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}, \boldsymbol{R} = \boldsymbol{R}_n^{-1}\ldots\boldsymbol{R}_1^{-1}.$$

In exact arithmetic $C(\boldsymbol{Q}) = C(\boldsymbol{A})$ by construction, and $\kappa(\boldsymbol{Q}) = 1$, but the sequence of multiplications with $\boldsymbol{R}_1, \ldots, \boldsymbol{R}_n$ might amplify perturbations in $\boldsymbol{A}$ (due for example to floating point representation errors or inherent uncertainty in knowledge of $\boldsymbol{A}$). The problem $\boldsymbol{f}: \mathbb{C}^{m\times n} \to C^{m\times n} \times \mathbb{C}^{n\times n}$, $\boldsymbol{A} \xrightarrow{f} \boldsymbol{Q}, \boldsymbol{R}$ has condition number

$$\kappa = \frac{\|\delta \boldsymbol{Q}\|}{\|\boldsymbol{Q}\|} \cdot \frac{\|\boldsymbol{A}\|}{\|\delta \boldsymbol{A}\|} + \frac{\|\delta \boldsymbol{R}\|}{\|\boldsymbol{R}\|} \cdot \frac{\|\boldsymbol{A}\|}{\|\delta \boldsymbol{A}\|},$$

and numerical experimentation (Fig. 2) readily exhibits large condition numbers.
An alternative approach is to obtain an orthogonal factorization through unitary transformations

$$\boldsymbol{Q}_n \ldots \boldsymbol{Q}_1 \boldsymbol{A} = \boldsymbol{R} \Rightarrow \boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}, \boldsymbol{Q} = \boldsymbol{Q}_1^* \ldots \boldsymbol{Q}_n^*.$$

Unitary transformations do not modify vector 2-norms or relative orientations

$$\|\boldsymbol{Q}\boldsymbol{x}\|^2 = \boldsymbol{x}^*\boldsymbol{Q}^*\boldsymbol{Q}\boldsymbol{x} = \|\boldsymbol{x}\|^2, (\boldsymbol{Q}\boldsymbol{y})^*(\boldsymbol{Q}\boldsymbol{x}) = \boldsymbol{y}^*\boldsymbol{x},$$

and are hence said to be isometric. In Euclidean space reflections and rotations are isometric.

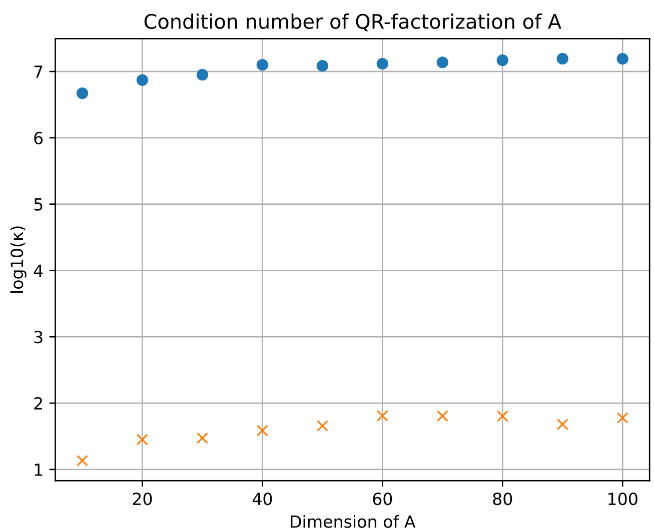

Figure 2. QR-conditioning: (o) modified Gram-Schmidt, (x) Householder.

Construction of an isometric reflection transformation suitable for a QR factorization is represented in Fig. 3. Let vector $\boldsymbol{x} \in \mathbb{C}^{m+1-k}$ represent the portion of the k^{th} column from the diagonal downwards in stage k of reduction of $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ to upper triangular form

$$\boldsymbol{Q}_{k-1}\ldots\boldsymbol{Q}_1\boldsymbol{A} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{C} \\ \boldsymbol{0} & \boldsymbol{B} \end{bmatrix}, \boldsymbol{B} = [\, \boldsymbol{x} \;\; \boldsymbol{b}_2 \;\; \ldots \;\; \boldsymbol{b}_{n-k} \,].$$

The next stage of in reduction to upper triangular form is the isometric transformation of $\boldsymbol{x}$ into $\pm\|\boldsymbol{x}\|\boldsymbol{e}_1$, with $\boldsymbol{e}_1 \in \mathbb{C}^{m+1-k}$ the unit vector along the first direction. With $\boldsymbol{v} = \pm\|\boldsymbol{x}\boldsymbol{e}_1\| - \boldsymbol{x}$, $\boldsymbol{q} = \boldsymbol{v}/\|\boldsymbol{v}\|$, the projection of $\boldsymbol{x}$ onto the span of $\boldsymbol{v}$, $C(\boldsymbol{v})$ is

$$\boldsymbol{y} = \boldsymbol{P}_{\boldsymbol{v}}\boldsymbol{x} = \boldsymbol{q}\boldsymbol{q}^*\boldsymbol{x},$$

and its complementary projector onto $N(\boldsymbol{v}^*)$ is

$$\boldsymbol{z} = \boldsymbol{P}_{\perp\boldsymbol{v}} = (\boldsymbol{I} - \boldsymbol{q}\boldsymbol{q}^*)\boldsymbol{x}.$$

The reflector transforming $\boldsymbol{x}$ into $\pm\|\boldsymbol{x}\|\boldsymbol{e}_1$ is obtained by doubling the above displacements, and is known as a Householder reflector

$$\boldsymbol{H} = \boldsymbol{I} - 2\boldsymbol{q}\boldsymbol{q}^*.$$

Of the two possibilities $\pm\|\boldsymbol{x}\|\boldsymbol{e}_1$, the choice

$$\boldsymbol{v} = -\text{sign}(x_1)\|\boldsymbol{x}\|\boldsymbol{e}_1 - \boldsymbol{x},$$

avoids loss of floating accuracy $\boldsymbol{x} \cong \|\boldsymbol{x}\|\boldsymbol{e}_1$. For $\boldsymbol{x} \in \mathbb{C}^{m+1-k}$, $\text{sign}(x_1) = \exp(\arg(x_1))$.

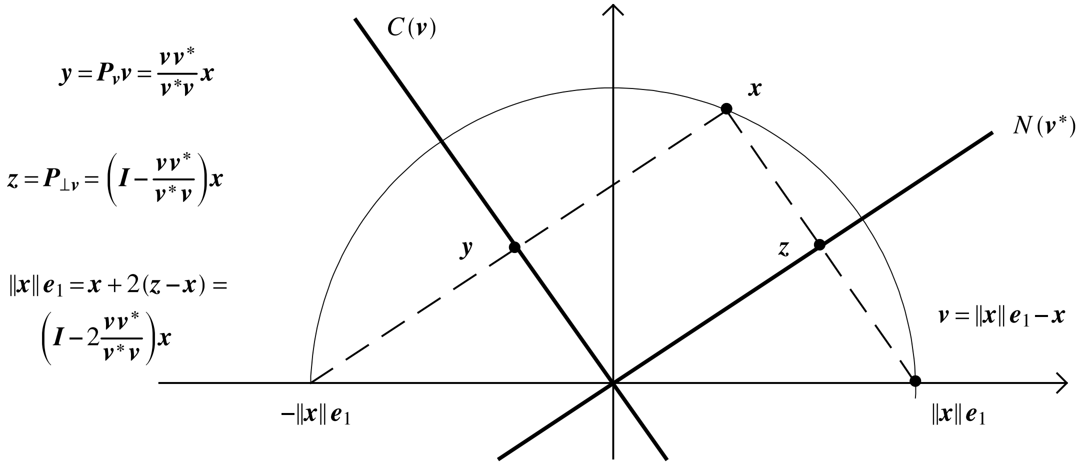

Figure 3. Geometry of Householder reflector

The resulting Householder QR-factorization is given

Input: $\boldsymbol{A} \in \mathbb{C}^{m\times n}$
$\boldsymbol{Q} = \boldsymbol{0}_{m,n}$
for $k = 1:n$
  $\boldsymbol{x} = \boldsymbol{A}[k:m,k]$
  $\boldsymbol{v} = \text{sign}(x_1)\,\|\boldsymbol{x}\| + \boldsymbol{x}$
  $\boldsymbol{q} = \boldsymbol{v}/\|\boldsymbol{v}\|$; $\boldsymbol{Q}[k:m,k] = \boldsymbol{q}$
  for $j = k:n$
    $\boldsymbol{A}[k:m,j] = \boldsymbol{A}[k:m,j] - 2\boldsymbol{q}\,(\boldsymbol{q}^*\boldsymbol{A}[k:m,j])$

```
∴ function HouseholderQR(A)
    m,n=size(A)
    Q=zeros(m,n); R=copy(A)
    for k=1:n
      x=R[k:m,k]
      e1=zeros(size(x)); e1[1]=1
      v=sign(x[1])*norm(x)*e1+x
      q=v/norm(v); Q[k:m,k]=q
      for j=k:n
        aj=R[k:m,j]; c=2*q'*aj
        R[k:m,j]=aj.-c*q
      end
    end
    return Q,R
  end;
∴
```

Note that the above implementation does not return the $\boldsymbol{Q}$ matrix, but rather the $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_n$ reflectors from which $\boldsymbol{Q}$ can be reconstructed if needed. Usually though, the $\boldsymbol{Q}$ matrix itself is not required, but rather the product $\boldsymbol{Q}\boldsymbol{u}$ which can readily be evaluated as $\boldsymbol{Q}_n \ldots \boldsymbol{Q}_1 \boldsymbol{u}$. The Householder reflector algorithm is typically the default procedure in QR-factorizations implemented in software systems, and as seen in (Fig. 2), leads to much better conditioning.

3. Orthogonal factorization through Given rotators

An alternative approach to orthogonal factorization utilizes isometric rotation transformations of the form

$$\boldsymbol{R}(i,k,\theta) = \boldsymbol{I} + (\cos\theta - 1)(\boldsymbol{e}_i\boldsymbol{e}_i^* + \boldsymbol{e}_k\boldsymbol{e}_k^*) - \sin\theta\,(\boldsymbol{e}_i\boldsymbol{e}_k^* - \boldsymbol{e}_k\boldsymbol{e}_i^*),$$

with the rotation angle θ chosen to nullify the subdiagonal element (i,k), $i>k$

$$(\boldsymbol{R}(i,k,\theta)\,\boldsymbol{A})_{ik} = a_{kk}\sin\theta + a_{ik}\cos\theta = 0 \Rightarrow \theta_{ik} = \arctan\left(-\frac{a_{ik}}{a_{kk}}\right).$$

Composition of repeated rotations $\boldsymbol{Q}_{ik} = \boldsymbol{R}(i,k,\theta_{ik})$ can be organized to lead to an upper triangular matrix

$$\boldsymbol{Q}_{mn}\ldots\boldsymbol{Q}_{32}\boldsymbol{Q}_{m1}\ldots\boldsymbol{Q}_{31}\boldsymbol{Q}_{21}\boldsymbol{A} = \boldsymbol{R}.$$

Whereas Householder reflectors work on entire vectors, Givens rotators nullify individual subdiagonal elements. For full matrices Householder reflectors typically require fewer floating point operations, but the special structure of a sparse matrix is better exploited by use of Givens rotators.

Input: $\boldsymbol{A}\in\mathbb{C}^{m\times n}$
$\boldsymbol{Q} = \boldsymbol{0}_{m,n}$
for $k=1:n$
  for $i=k+1:m$
  $\theta = \arctan(-a_{ik}/a_{kk})$
  $c=\cos(\theta)$; $s=\sin(\theta)$
  for $j=k:n$
    $u=a_{kj}$; $v=a_{ij}$
    $a_{kj}=c\,u-s\,v$
    $a_{ij}=s\,u+c\,v$

```
∴ function GivensQR(A)
    m,n=size(A)
    Q=zeros(m,n); R=copy(A)
    for k=1:n
      for i=k+1:m
        θ = atan(-R[i,k],R[k,k]); Q[i,k]=
        c = cos(θ); s = sin(θ)
        for j=k:n
          u = R[k,j]; v = R[i,j]
          R[k,j]=c*u-s*v
          R[i,j]=s*u+c*v
        end
      end
    end
    return Q,R
  end;
∴
```

As in the Householder implementation the above implementation returns data to reconstruct $\boldsymbol{Q}$ if needed.