Matrix Algebra

Matrices represent linear transformations from $\mathbb{R}^n \to \mathbb{R}^n$. We represent a matrix $A \in \mathbb{R}^{n \times n}$ as

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

For a vector $x \in \mathbb{R}^n$ with $x = (x_1, x_2, \dots, x_n)$ we define $b = Ax$ as

$$b_1 = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n$$
$$b_2 = a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n$$
$$\vdots$$
$$b_n = a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n.$$

Then we define matrix addition and multiplication as follows:

$$(A+B)x = Ax + Bx$$
$$ABx = y \Leftrightarrow Bx = z \wedge Az = y$$

These operations are well-defined and satisfy:

$$C = A + B \Leftrightarrow C_{ab} = A_{ab} + B_{ab} \quad , \forall a, b \in [n]^2$$

$$C = AB \Leftrightarrow C_{ab} = \sum_{c=1}^{n} A_{ac} B_{cb} , \forall a, b \in [n]^2 .$$

The transpose matrix is defined as:

$$B = A^T \Leftrightarrow B_{ab} = A_{ba} \quad , \forall a, b \in [n]^2 .$$

We also define identity matrix $I = [\delta_{ab}]$ with

$$\delta_{ab} = \begin{cases} 1 & , a = b \\ 0 & , a \neq b \end{cases}$$

such that $Ix = x , \forall x \in \mathbb{R}^n$.

▸ Properties

$$A + B = B + A \quad , \forall A, B \in \mathbb{R}^{n \times n}$$
$$(A+B) + C = A + (B+C) \quad , \forall A, B, C \in \mathbb{R}^{n \times n}$$
$$AI = IA = A \quad , \forall A \in \mathbb{R}^{n \times n}$$
$$(AB)C = A(BC) \quad , \forall A, B, C \in \mathbb{R}^{n \times n} .$$

▸ Matrix Inverse

A matrix B is the inverse of a matrix A iff

$$AB = BA = I.$$

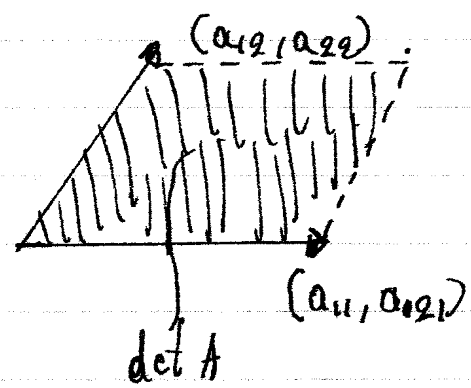
If the inverse B exists we denote it as $B = A^{-1}$.
If a matrix A does not have an inverse, then we say that A is singular.

▸ Determinants

Let $A = [v_1\ v_2 \cdots v_n]$ be a matrix whose columns are given by the vectors $v_1, v_2, \ldots, v_n$. The determinant $\det A$ is defined as the "volume" of the "box" defined by the vectors $v_1, v_2, \ldots, v_n$.

↳ In two dimensions

$$\det A = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$$



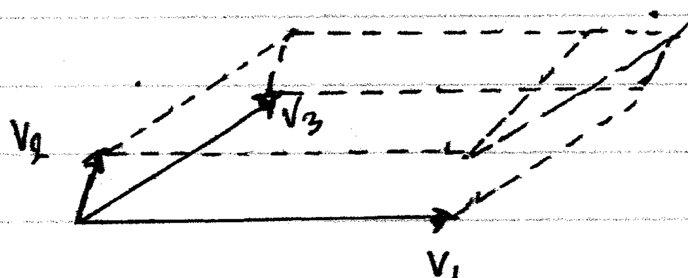
In three dimensions

$$\det A = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \begin{matrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{matrix}$$

$\ominus$

$\oplus$

$$= a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32}$$
$$- a_{31}a_{22}a_{13} - a_{32}a_{23}a_{11} - a_{33}a_{21}a_{12}$$



$v_1 = (a_{11}, a_{21}, a_{31})$
$v_2 = (a_{12}, a_{22}, a_{32})$
$v_3 = (a_{13}, a_{23}, a_{33})$

Properties of determinants

1) $\det I = 1$
2) $\det(AB) = \det A \det B$
3) For $A = [v_1, v_2, \ldots, v_n]$
   $\det A = 0 \Leftrightarrow A$ is singular
   $\Leftrightarrow v_1, v_2, \ldots, v_n$ are linearly dependent.
4) $\det A^T = \det A$.

The two big problems of linear algebra

① <u>Linear system</u> : $Ax = b$.

Given A, b, find x such that $Ax = b$.
<u>Formal solution</u> : If A is not singular, then

$$Ax = b \Leftrightarrow A^{-1}Ax = A^{-1}b \Leftrightarrow Ix = \cancel{b}\ \cancel{\Rightarrow}\ A^{-1}b$$
$$\Leftrightarrow x = A^{-1}b.$$

Methods

1) Cramer Rule – Method of Determinants
   (problem: requires $n!$ operations)
2) Gaussian Elimination
   (problem: messy implementation)
3) QR decomposition
   (the best method)
4) LU decomposition
   (second best – twice as fast)

② Eigenvalue problem : $Ax = \lambda x$

Find $\lambda \in \mathbb{R}$ and $x \in \mathbb{R}^n$ such that $Ax = \lambda x$.
λ = eigenvalue of A
x ~~A~~ = eigenvector of A.

Formal solution:
λ eigenvalue of $A \Leftrightarrow \det(A - \lambda I) = 0$
This leads to an n-order polynomial equation.
1) Calculation of polynomial difficult
2) Solution of polynomial equation difficult.

Numerical algorithm : QR decomposition is used iteratively!

Notation

1) The set of all eigenvalues $\lambda \in \mathbb{C}$ of a matrix A is the spectrum $sp(A)$ of the matrix

$$sp(A) = \{ \lambda \in \mathbb{C} \mid \exists x \in \mathbb{C} : Ax = \lambda x \}$$

2) The largest eigenvalue in "modulus" is the spectral radius $\rho(A)$:

$$\rho(A) = \max_{\lambda \in sp(A)} |\lambda|$$

How close is a matrix to being singular?

The determinant is not a good measure of how close is a matrix to being singular.
Why? One can "inflate" $\det A$ simply by setting $B = \lambda A$ with $\lambda \to +\infty$. B is as ill-conditioned as A but $\det B \gg \det A$.
Best Approach: Use matrix norm to introduce the "condition number".

Matrix Norms

Let $A \in \mathbb{R}^{n \times n}$ be a matrix. Let $\mathbb{B}_p(\rho)$ be the "ball" of the vector p-norm:

$$\mathbb{B}_p(\rho) = \{ x \in \mathbb{R}^n \mid \|x\| = \rho \}.$$

The p-norm of the matrix A is defined as

$$\|A\|_p = \max_{x \in \mathbb{B}_p(1)} \|Ax\|_p$$

▶ Properties of the matrix p-norm

1) $\|A\|_p$ satisfies the norm definition for $p \geq 1$.

2) $\forall A, B \in \mathbb{R}^{n \times n} : \|AB\|_p \leq \|A\|_p \|B\|_p$.

3) Matrix-Vector norm consistency:

$$\forall A \in \mathbb{R}^{n \times n} : \forall x \in \mathbb{R}^n : \|Ax\|_p \leq \|A\|_p \|x\|_p.$$

4) Matrix norm self-consistency,

$$\forall a, b \in [1, +\infty) : \exists c_1, c_2 \in (0, +\infty) : c_1 \|A\|_a \leq \|A\|_b \leq c_2 \|A\|_a , \forall A \in \mathbb{R}^{n \times n}.$$

▶ Evaluation of matrix p-norms

1) $\|A\|_1 = \max_{b \in [n]} \sum_{a=1}^{n} |A_{ab}|$

2) $\|A\|_\infty = \max_{a \in [n]} \sum_{b=1}^{n} |A_{ab}|$

3) $\|A\|_2 = \rho(A^T A) \leq \sqrt{\|A\|_1 \|A\|_\infty}$

Condition Number

Let $Ax = b$ and $A(x+\Delta x) = b + \Delta b$. Then the relative error in x and the relative error in b are related by

$$\boxed{\frac{\|\Delta x\|_p}{\|x\|_p} \leqslant \kappa_p(A)\,\frac{\|\Delta b\|_p}{\|b\|_p}}$$

with $\kappa_p(A)$ defined as the condition number of A, given by

$$\boxed{\kappa_p(A) = \|A\|_p \, \|A^{-1}\|_p}$$

Proof

Note that $A(\Delta x) = \Delta b \Rightarrow \Delta x = A^{-1}(\Delta b)$. Then

$$\frac{\|\Delta x\|_p}{\|x\|_p} = \frac{\|A^{-1}(\Delta b)\|_p}{\|x\|_p} \leqslant \frac{\|A^{-1}\|_p\,\|\Delta b\|_p}{\|x\|_p} =$$

$$= \frac{\|A^{-1}\|_p\,\|b\|_p}{\|x\|_p}\,\frac{\|\Delta b\|_p}{\|b\|_p} =$$

$$= \frac{\|A^{-1}\|_p \|Ax\|_p}{\|x\|_p} \frac{\|\Delta b\|_p}{\|b\|_p} \leq \frac{\|A^{-1}\|_p \|A\|_p \|x\|_p}{\|x\|_p} \frac{\|\Delta b\|_p}{\|b\|_p}$$

$$= \|A^{-1}\|_p \|A\|_p \frac{\|\Delta b\|_p}{\|b\|_p} = \kappa_p(A) \frac{\|\Delta b\|_p}{\|b\|_p} .$$

▸ Kahan Theorem : The condition number $\kappa_p(A)$ gives the relative p-norm distance of A from the space of singular matrices:

$$\boxed{\frac{1}{\kappa_p(A)} = \min \left\{ \frac{\|B-A\|_p}{\|A\|_p} \;\middle|\; \det B = 0 \right\}}$$

▸ Examples.

For

$$B_n = \begin{bmatrix} 1 & -1 & \cdots & -1 \\ 0 & 1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \qquad \text{(very ill-conditioned)}$$

it can be shown that $\det B_n = 1$ but $\kappa_\infty(B_n) = n 2^{n-1}$.

For $D_n = \begin{bmatrix} 10^{-1} & 0 & \cdots & 0 \\ 0 & 10^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 10^{-n} \end{bmatrix}$

we have $K_p(D_n) = 1$ but $\det D_n = 10^{-n}$.

▶ <u>Properties of condition number</u>

1) $K_p(A) \geq 1 \quad , \forall A \in \mathbb{R}^{n \times n}$
2) $K_p(\lambda A) = K(A) \ , \forall \lambda \in \mathbb{R} - \{0\}, \ \forall A \in \mathbb{R}^{n \times n}$.

Numerical Solution of Linear System

LU decomposition method

Def: Let $A = [A_{ab}] \in \mathbb{R}^{n \times n}$ be a matrix

a) A upper triangular $\Leftrightarrow \forall a, b \in [n] : (a > b \Rightarrow A_{ab} = 0)$
b) A lower triangular $\Leftrightarrow \forall a, b \in [n] : (a < b \Rightarrow A_{ab} = 0)$

Def: A matrix $A \in \mathbb{R}^{n \times n}$ is LU decomposable iff $\exists L, U \in \mathbb{R}^{n \times n}$ such that

a) L lower triangular
b) U upper triangular
c) $\forall a \in [n] : L_{aa} = 1$
d) $A = LU$

example: For a 3x3 matrix, the LU decomposition, if it exists, has the form

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

▶ Existence of LU decomposition

<u>Theorem</u> : Let B_k be a submatrix of A such that

$B_k = [B_{ab}] \in \mathbb{R}^{k \times k} \Leftrightarrow \forall a, b \in [k] : B_{ab} = A_{ab}$.

Then,

$$\left.\begin{array}{l}A \text{ has an LU} \\ \text{decomposition}\end{array}\right\} \Leftrightarrow \forall k \in [n] : \det B_k \neq 0 .$$

↳ <u>Special cases</u>: Diagonal Dominance

<u>Def</u>: A matrix $A \in \mathbb{R}^{n \times n}$ is diagonally dominant iff

$$\forall a \in [n] : |A_{aa}| > \sum_{b \in [n] - \{a\}} |A_{ab}|$$

<u>Thm</u> : $\left.\begin{array}{l}A \text{ diagonally} \\ \text{dominant}\end{array}\right\} \Rightarrow \Big\{ A$ has an LU decomposition.

▶ Algorithms

If it exists, the LU decomposition can be evaluated with the following algorithm:

<u>Algorithm</u> : Crout LU decomposition

for $a = 1, \dots, n$ $\{ L_{aa} = 1 \}$
for $b = 1, \dots, n \{$ for $a = 1, \dots, b$
$\{ U_{ab} = A_{ab} - \sum_{\gamma=1}^{a-1} L_{a\gamma} U_{\gamma b} \} \}$
for $a = b+1, \dots, n$ $\{$

$$L_{ab} = \frac{1}{U_{bb}} \left(A_{ab} - \sum_{\gamma=1}^{b-1} L_{a\gamma} U_{\gamma b} \right) \} \}$$

▶ Applications of LU decomposition

1) Solution of linear system

$$Ax = b \Leftrightarrow x = A^{-1} b = (LU)^{-1} x = U^{-1} L^{-1} x$$

The operation of L^{-1} and U^{-1} on a matrix can be done via the following algorithms:

<u>Algorithm</u> : $y = L^{-1} b$ (forward substitution)

$$y_1 = b_1$$
$$\text{for } a = 2, 3, \ldots, n : \left\{ y_a = b_a - \sum_{b=1}^{a-1} L_{ab} y_b \right\}$$

<u>Algorithm</u> : $x = U^{-1} y$ (back substitution)

$$x_n = y_n / U_{nn}$$
$$\text{for } a = n-1, n-2, \ldots, 1 : \left\{ x_a = \frac{1}{U_{aa}} \left(y_a - \sum_{b=a+1}^{n} A_{ab} x_b \right) \right\}$$

2) <u>Determinant of A</u>

$$\left.\begin{array}{l} \det A = \det L \det U \\ \det L = 1 \end{array}\right\} \Rightarrow \det A = \det U = \prod_{a=1}^{n} U_{aa}.$$

▶ Pivoting

LU is not a reliable algorithm because not every non-singular matrix has an LU decomposition. Introduce "pivoting"

a) With "partial pivoting" LU algorithm is "statistically stable"

b) With "full pivoting" LU algorithm is provably stable, but too slow!

QR decomposition

The QR decomposition method is twice as slow as LU decomposition with pivoting but more reliable

Preliminaries

Def: A matrix $Q = [q_1\ q_2 \cdots q_n] \in \mathbb{R}^{n \times n}$ is orthogonal iff

a) The vectors $q_1, q_2, \ldots, q_n$ are orthogonal
b) $\|q_k\|_2 = 1$, $\forall k \in [n]$.

An orthogonal matrix has the following properties: (assume Q orthogonal)

1) $Q^{-1} = Q^T$
2) $\forall p \in (1, +\infty)$: $k_p(Q) = 1$
3) $\forall x \in \mathbb{R}^n$: $\|Qx\|_2 = \|x\|_2$.
4) $\det Q = \pm 1$
   - An orthogonal matrix with $\det Q = +1$ is called a rotation. An orthogonal matrix with $\det Q = -1$ is called a reflection.

QR decomposition theorem

<u>Thm</u>: For every matrix $A \in \mathbb{R}^{n \times n}$, there is a matrix $Q \in \mathbb{R}^{n \times n}$ and a matrix $R \in \mathbb{R}^{n \times n}$ such that

a) $A = QR$
b) Q is orthogonal
c) R is upper triangular.

The QR decomposition can be calculated with
a) Householder transformations
b) Givens rotations.
Both transformations are rank-1 updates.

Rank-1 updates

<u>Def</u>: Let $u, v \in \mathbb{R}^n$ be two vectors. The <u>tensor product</u> is defined as $A = u \otimes v$ such that

$$A_{ab} = u_a v_b \, , \; \forall a, b \in [n]$$

<u>Def</u>: A rank-1 update is a transformation

$$A \to A - u \otimes v$$

The Householder transformation

Def: The Householder transformation associated with a vector $v \in \mathbb{R}^n$ is defined as

$$A \to H(u)A$$

with $H(u)$ given by:

$$\boxed{H(u) = I - \frac{1}{\|u\|_2^2} u \otimes u}$$

- Note that a rank-1 update, in general, satisfies the following properties:

a) $(u \otimes v) A = u \otimes (A^T v)$

b) $A(u \otimes v) = (Au) \otimes v$

c) $(v \otimes u) = (u \otimes v)^T$ d) $(u \otimes v) w = (v, w) u$

It follows that:

$$H(u)A = \left(I - \frac{1}{\|u\|_2^2} u \otimes u\right) A = A - u \otimes (A^T u) \frac{1}{\|u\|_2^2}$$

$$= A - u \otimes \left[\frac{1}{\|u\|_2^2} A^T u\right]$$

The key property of the Householder transformation is:

$$\boxed{u = x \pm \cancel{\text{ske}}\,\|x\|_2 e_1 \Rightarrow H(u)x = \mp \|x\|_2 e_1}$$

with $e_1 = (1, 0, \dots, 0)$ the unit vector.

<u>Proof</u>

$$H(u)x = \left(I - \frac{2}{\|u\|_2^2}(u \otimes u)\right)x =$$

$$= x - \frac{2}{\|u\|_2^2}(u, x)u = x - \frac{2(u,x)}{\|u\|_2^2}u.$$

For $u = x \pm \|x\|_2 e_1$ it follows that

$$(u,x) = (x \pm \|x\|_2 e_1, x) = (x, x) \pm (\|x\|_2 e_1, x)$$
$$= \|x\|_2^2 \pm x_1 \|x\|_2 \quad , \text{ and}$$

$$(u,u) = (x \pm \|x\|_2 e_1, x \pm \|x\|_2 e_1) =$$
$$= (x, x) \pm 2(x, \|x\|_2 e_1) + (\|x\|_2 e_1, \|x\|_2 e_1)$$
$$= \|x\|_2^2 \pm 2\|x\|_2 x_1 + \|x\|_2^2 \cdot 1$$
$$= 2\|x\|_2^2 + 2x_1 \|x\|_2 = 2(u, x).$$

Therefore

$$H(u)x = x - \frac{2(u,x)}{(u,u)}\, u = x - u =$$

$$= x - (x \pm \|x\|_2 e_1) = \mp \|x\|_2 e_1 . \qquad \square$$

The QR algorithm

We apply a sequance of Householder transformations on the matrix A with the intention to make it upper triangular. Each transformation "zeroes" the elements bellow the diagonal for one matrix column:

$$\begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} \xrightarrow{H(u_1)A} \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \end{bmatrix} \xrightarrow{H(u_2)}$$

u_1 $\qquad\qquad\qquad\qquad$ u_2

$$\begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & x & x \end{bmatrix} \xrightarrow{H(u_3)} \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix}$$

The result is a series of transformations that gives:

$$H(u_n) \cdots H(u_2) H(u_1) A = R$$

Then:

$$A = [H(u_n) \cdots H(u_2) H(u_1)]^{-1} R$$
$$= \underbrace{[H(u_1)]^T [H(u_2)]^T \cdots [H(u_n)]^T}_{Q} R$$

$$= QR.$$

To solve $Ax = b \Leftrightarrow x = A^{-1} b = (QR)^{-1} x$
$$= R^{-1} Q^{-1} x$$
$$= R^{-1} Q^T x.$$

Operating with Q^T is easy.
Operating with R^{-1} requires the back substitution algorithm.

Implementation of QR algorithm

We use the notation $A(a:b, c:d)$ to represent the submatrix of A that includes columns $c, \dots, d$ and rows $a, \dots, b$. Likewise for $a=b$ we write $A(a, c:d)$ and for $c=d$ we write $A(a:b, c)$. The QR algorithm can be written as:

<u>Algorithm</u> : Let $A \in \mathbb{R}^{m \times n}$ with $m > n$.

for $a = 1, \dots, n$
    let $x = A(a:m, a)$; $x = x/\|x\|_2$
    let $u = x - \|x\|_2 e_1$; $u = u/u(1)$
    $A(a:m, a:n) = H(u) A(a:m, a:n)$
    if $a < m$ then
        $A(a+1:m, a) = u(2:m-a+1)$
    end if
end for.

This algorithm overwrites A with the upper triangular portion of R. The lower triangular portion of A has the Householder vectors u that can be used to rebuild Q.

Upon completion, the matrix A has:

$$A = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{1n} \\ u_2^{(1)} & r_{22} & r_{23} & r_{2n} \\ u_3^{(1)} & u_3^{(2)} & r_{33} & r_{3n} \\ u_n^{(1)} & u_n^{(2)} & u_n^{(3)} & r_{nn} \end{bmatrix}$$

with r_{ab} the elements of R.
Q is given by: $Q = H(u_1)H(u_2)\cdots H(u_n)$.
with

$$u_k = [\underbrace{0, \ldots, 0}_{k-1}, \underset{\uparrow}{1}, \underbrace{u_{k+1}^{(k)}, \ldots, u_n^{(k)}}_{\text{essential part of Householder vector.}}]$$

The Givens rotation

The Givens matrix $G_{ab}(\vartheta)$ is defined as

$$G_{ab}(\vartheta) = \begin{array}{c c} \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & & c & \cdots & s & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} & \begin{array}{c} \\ \\ a \\ \\ b \\ \\ \\ \end{array} \\ \begin{array}{ccccccc} & & a & & b & & \end{array} & \end{array}$$

with $c = \cos\vartheta$ and $s = \sin\vartheta$. More formally, if $x \in \mathbb{R}^n$ and $y = G_{ab}(\vartheta) = x$ then

$$y_k = \begin{cases} c x_a - s x_b & , k = a \\ s x_a + c x_b & , k = b \\ x_k & , \text{otherwise} \end{cases}$$

- We can force $y_b = 0$ if we choose:

$$c = \frac{x_a}{\sqrt{x_a^2 + x_b^2}} \quad \text{and} \quad s = \frac{-x_b}{\sqrt{x_a^2 + x_b^2}}$$

We designate this calculation as

$$[c, s] = \text{givens}(x_a, x_b).$$

We use Givens rotations to zero the lower triangular part of A one at a time:

$$\begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} \xrightarrow{G_{34}} \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ 0 & x & x & x \end{bmatrix} \xrightarrow{G_{23}}$$

$$\begin{bmatrix} x & x & x & x \\ x & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \end{bmatrix} \xrightarrow{G_{12}} \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \end{bmatrix} \xrightarrow{G_{34}}$$

$$\begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \end{bmatrix} \xrightarrow{G_{23}} \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & x & x \end{bmatrix} \xrightarrow{G_{34}}$$

$$\begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix}$$

It is therefore possible to calculate QR using givens rotations instead of Householder. The advantage of Givens over Householder is that it is more resilient with ill-conditioned matrices.