Homework no. 3

Let $\boldsymbol{n}$ be the size of the linear system, $\boldsymbol{\varepsilon}$ - the computations error, the square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ and a vector $\boldsymbol{s} \in \mathbb{R}^{n}$.

1. Compute vector $\boldsymbol{b} \in \mathbb{R}^{n}$ in the following way:

$$b_i = \sum_{j=1}^{n} s_j a_{ij} \quad , \; i = 1,\dots,n$$

2. Implement the $\boldsymbol{QR}$ decomposition for matrix $\boldsymbol{A}$ using Householder's algorithm.
3. Solve the linear system:

$$Ax = b \quad ,$$

using the $\boldsymbol{QR}$ decomposition implemented in one of the libraries mention on the lab webpage (one obtains the solution $\boldsymbol{x_{QR}}$) and the $\boldsymbol{QR}$ decomposition computed at item 2. (one obtains the solution $\boldsymbol{x_{Householder}}$). Compute and display:

$$\left\| x_{QR} - x_{Householder} \right\|_2 .$$

4. Compute and display the following errors ($\|\cdot\|_2$ is the Euclidean norm):

$$\| A^{init} x_{Householder} - b^{init} \|_2 \quad ,$$

$$\| A^{init} x_{QR} - b^{init} \|_2 \quad ,$$

$$\frac{\| x_{Householder} - s \|_2}{\| s \|_2} \quad ,$$

$$\frac{\| x_{QR} - s \|_2}{\| s \|_2} .$$

(these values should be smaller than $\mathbf{10^{-6}}$)

5. Compute the inverse matrix of matrix $\boldsymbol{A}$ using the $\boldsymbol{QR}$ decomposition computed at item 2. Compare it with inverse matrix computed using the corresponding function from the library. Display the norm:

$$\left\| A^{-1}_{Householder} - A^{-1}_{bibl} \right\| .$$

6. Write your program with the possibility of random initialization of the input data (thus your program can run for any value of $\boldsymbol{n}$).

Bonus (15pt): Let matrix $A \in \mathbb{R}^{n\times n}$ be symmetric. Compute the approximate limit of the sequence of matrices:

1) $k=0,\ A^{(0)}=A$;
2) For $A^{(k)}$ compute the QR decomposition using Householder's algorithm, $A^{(k)} = Q * R$;
3) $A^{(k+1)} = R * Q$;
4) The approximate limit of the sequence of these matrices is the matrix $A^{(k+1)}$ that satisfies the property:

$$\left\| A^{(k+1)} - A^{(k)} \right\| \le \varepsilon \,.$$

What particular form has this matrix $A^{(k+1)}$ and what represent the elements of this matrix for the initial matrix A?

Computations restrictions:

a) In computing the sequence of matrices $A^{(k)}$ use as few matrices as possible;
b) When computing the product of matrices $R*Q$ use only the elements from the upper triangular part of matrix R (r_{ij} cu $i\le j$), and the elements of the entire matrix Q, without using instructions „*if-then-else*".

Solving linear systems using a *QR* decomposition

Let A be a real, square matrix of size n. Assume that for matrix A one has a decomposition of the following form:

$$A=Q*R$$

where Q is an orthogonal matrix ($Q^TQ = QQ^T = I_n$) and R is a upper triangular matrix. If we have such a decomposition for matrix A, solving the linear system $Ax=b$ is equivalent with solving the upper triangular system $Rx=Q^Tb$.

$$Ax=b \;\leftrightarrow\; Q*Rx = b \;\leftrightarrow\; Q^T*Q*Rx = Q^T b \;\leftrightarrow\; Rx=Q^T b$$

$$Ax=b \quad \Leftrightarrow \quad Rx=Q^T b$$

Householder's Algorithm

For transforming the linear system $\boldsymbol{Ax=b}$ into $\boldsymbol{Rx = Q^T b}$ one uses the reflection matrices. A reflection matrix $\boldsymbol{P = (p_{ij})_{i,j=1,n}}$ has the following form:

$$\boldsymbol{P = I_n - 2vv^T} \quad , \quad \boldsymbol{v \in \mathbb{R}^n}, \ \|\boldsymbol{v}\|_2 = |\boldsymbol{v}| = \sqrt{\sum_{i=1}^{n} \boldsymbol{v}_i^2} = \mathbf{1}.$$

(we denoted by $\boldsymbol{I_n}$ the unity matrix of size $\boldsymbol{n}$)

One can show that the reflection matrices are symmetric and orthogonal:

$$\boldsymbol{P = P^T} \quad , \quad \boldsymbol{P^2 = I_n}.$$

Householder's algorithm for computing the $\boldsymbol{QR}$ decomposition has $\boldsymbol{(n\text{-}1)}$ steps. In step $\boldsymbol{r}$ the column $\boldsymbol{r}$ of matrix $\boldsymbol{A}$ is transformed in upper triangular form without modifying the first $\boldsymbol{(r-1)}$ columns. In this step one also obtains column $\boldsymbol{r}$ of matrix $\boldsymbol{R}$. One can compute and store matrix $\boldsymbol{R}$ directly in matrix $\boldsymbol{A}$ (one can say that matrix $\boldsymbol{A}$ is transformed in an upper triangular matrix). In the same time, one can perform the necessary transformations on vector $\boldsymbol{b}$ to obtain $\boldsymbol{Q^T b}$ one can also compute matrix $\boldsymbol{Q^T}$. For computing the matrix $\boldsymbol{Q^T}$, one start with $\boldsymbol{\bar{Q} = I_n}$ and then we perform the same transformations on matrix $\boldsymbol{\bar{Q}}$ as those performed on matrix $\boldsymbol{A}$.

Step *r (r=1,2,…,n-1)*

When entering this step, matrix ***A*** has the first ***(r-1)*** columns in upper triangular form. In this step we want to transform column ***r*** of matrix ***A*** in upper triangular form (without modifying the upper triangular form of the first ***(r-1)*** columns). To do this, one uses a reflection matrix P_r:

$$\begin{aligned} A &= P_r * A \\ b &= P_r * b \\ \bar{Q} &= P_r * \bar{Q} \end{aligned}$$

where matrix P_r is computed in the following way:

$$P_r = I_n - \frac{1}{\beta} u u^T$$

$$\beta = \sigma - k a_{rr}, \quad \sigma = \sum_{j=r}^{n} a_{jr}^2, \quad k = -\mathrm{sign}(a_{rr})\sqrt{\sigma}, \quad \mathrm{sign}(x) = \begin{cases} +1 & \text{if } x \geq 0, \\ -1 & \text{if } x < 0. \end{cases}$$

$$u = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ a_{rr} - k \\ a_{r+1r} \\ \vdots \\ a_{nr} \end{pmatrix}, \qquad \begin{gathered} u_i = 0,\ i = 1,\dots,r-1, \\ u_r = a_{rr} - k, \\ u_i = a_{ir},\ i = r+1,\dots,n \end{gathered}$$

Matrix $V = uu^T \in \mathbb{R}^{n \times n}$, $V = \left(v_{ij}\right)_{i,j=1,\dots,n}$ has the following elements:

$$v_{ij} = \begin{cases} 0 & \text{for } i = 1,\dots,r-1,\ j = 1,\dots,n \\ 0 & \text{for } i = r,\dots,n,\ j = 1,\dots,r-1 \\ u_i u_j & \text{for } i = r,\dots,n,\ j = r,\dots,n \end{cases}$$

The reflection matrix $\boldsymbol{P_r} = (\boldsymbol{p_{ij}})_{\boldsymbol{i,j=1,n}}$ has the elements:

$$\boldsymbol{p_{ij}} = \begin{cases} \boldsymbol{0} & \text{for } \boldsymbol{i = 1,...,r-1\ ,\ \ j = 1,...,n,\ i \neq j} \\ \boldsymbol{1} & \text{for } \boldsymbol{i = 1,...,r-1,\ j = i} \\ \boldsymbol{0} & \text{for } \boldsymbol{i = r,...,n\ ,\ \ j = 1,...,r-1} \\ \boldsymbol{-f\, u_i u_j} & \text{for } \boldsymbol{i = r,...,n\ ,\ \ j = r,...,n,\ i \neq j} \\ \boldsymbol{1 - f\, u_i^2} & \text{for } \boldsymbol{i = r,...,n\ ,\ j = i} \end{cases} \qquad , \quad \boldsymbol{f = \frac{1.0}{\beta}}$$

After applying this algorithm, the upper triangular matrix $\boldsymbol{R}$ is stored in matrix $\boldsymbol{A}$, and the transpose of matrix $\boldsymbol{Q}$ can be found in matrix $\bar{\boldsymbol{Q}}$. In vector $\boldsymbol{b}$ we'll have $\boldsymbol{Q^T b^{init}}$. The verification that the initial matrix is non-singular ($\mathbf{det}\,\boldsymbol{A} \neq \mathbf{0}$) is done by testing if all the diagonal elements of matrix $\boldsymbol{R}$ (or $\boldsymbol{A}$) are non-zero ($|\boldsymbol{r_{ii}}| \leq \boldsymbol{\varepsilon}, \forall \boldsymbol{i}$)
.

Householder's algorithm for computing the ***A=QR*** *factorization*

$\widetilde{Q} = I_n;$

for $r = 1,\ldots,n-1$

$\quad$ // **building of matrix** P_r – **constant** β **and vector** u

- $\sigma = \sum_{i=r}^{n} a_{ir}^2;$
- **if** $(\sigma \le \varepsilon)$ **break ;** // $r = r+1 \leftrightarrow P_r = I_n$ (A **singular)**
- $k = \sqrt{\sigma};$
- **if** $(a_{rr} > 0)\ k = -k;$
- $\beta = \sigma - k\, a_{rr};$
- $u_r = a_{rr} - k;\ \ u_i = a_{ir}\ ,\ i = r+1,\ldots,n;$

// $A = P_r * A$

// **columns** $j = r+1,\ldots,n$ **transformation**

- **for** $j = r+1,\ldots,n$

$$* \ \gamma = (\gamma_j / \beta) = (Ae_j, u)/\beta = \Big(\sum_{i=r}^{n} u_i a_{ij}\Big)/\beta;$$

$\quad * $ **for** $i = r,\ldots,n$

$$a_{ij} = a_{ij} - \gamma * u_i;$$

// **column** r **of matrix** A **transformation**

- $a_{rr} = k;\ \ a_{ir} = 0,\ \ i = r+1,\ldots,n;$

// $b = P_r * b$

- $\gamma = (\gamma / \beta) = (b,u)/\beta = \Big(\sum_{i=r}^{n} u_i b_i\Big)/\beta;$
- **for** $i = r,\ldots,n \quad b_i = b_i - \gamma * u_i;$

// $\widetilde{Q} = P_r * \widetilde{Q}$

- **for** $j = 1,\ldots,n$

$$* \ \gamma = (\widetilde{Q}e_j, u)/\beta = \Big(\sum_{i=r}^{n} u_i \tilde{q}_{ij}\Big)/\beta;$$

$\quad * $ **for** $i = r,\ldots,n$

$$\tilde{q}_{ij} = \tilde{q}_{ij} - \gamma * u_i;$$

Computing an approximation for the inverse of a matrix

Assume we know a numerical method for solving linear systems, in our situation the QR decomposition provided by the Householder algorithm. Then the columns of the inverse matrix can be approximated by solving ***n*** linear systems.

Column ***j*** of matrix A^{-1} is approximated by solving the linear system:

$$Ax = e_j \ , \ e_j = (0,0,\ldots,\underbrace{1}_{\text{position } j},\ldots,0)^T \ , \ j = 1,\ldots,n \ .$$

The procedure of computing matrix $A^{-1}_{Householder}$ is the following:

- Compute the ***QR*** decomposition of matrix ***A*** using Householder's algorithm.
- If matrix ***A*** is singular ($\det A = 0 \leftrightarrow \exists\ |r_{ii}| < \varepsilon$), STOP, the inverse matrix cannot be computed.
- Else, compute the columns of matrix $A^{-1}_{Householder}$ in the following way:

  for ***j=1,..,n***

  1. Set vector $b = Q^T e_j$ = column ***j*** of matrix Q^T or row ***j*** of matrix ***Q*** ;
     (use the initialization with the column ***j*** of matrix Q^T or row ***j*** of matrix ***Q*** depending on the matrix returned by the Householder algorithm)
  2. Solve the upper triangular system ***Rx=b***, using the back substitution. One obtains the solution x^* (x^* is also the solution of the linear system $Ax=e_j$);
  3. Store x^* in column ***j*** of matrix $A^{-1}_{Householder}$.

The procedure described above is a numerical method that uses the ***QR*** decomposition, for solving the matrix equation:

$$AX=I_n \ .$$

Example

$$A=\begin{pmatrix}0&0&4\\1&2&3\\0&1&2\end{pmatrix},\quad s=\begin{pmatrix}3\\2\\1\end{pmatrix}$$

$$\begin{array}{rcrcrcr} & & & & 4x_3 & = & 4\\ x_1 & + & 2x_2 & + & 3x_3 & = & 10\\ & & x_2 & + & 2x_3 & = & 4\end{array}$$

$$A=\begin{pmatrix}0&0&4\\1&2&3\\0&1&2\end{pmatrix}=QR=\begin{pmatrix}0&0&1\\1&0&0\\0&1&0\end{pmatrix}*\begin{pmatrix}1&2&3\\0&1&2\\0&0&4\end{pmatrix},\quad b=A*s=\begin{pmatrix}4\\10\\4\end{pmatrix}$$

$$Ax=b\Leftrightarrow Rx=Q^Tb\Leftrightarrow \begin{array}{rcrcrcr} x_1 & + & 2x_2 & + & 3x_3 & = & 10\\ & & x_2 & + & 2x_3 & = & 4\\ & & & & 4x_3 & = & 4\end{array}\Rightarrow x^*=\begin{pmatrix}3\\2\\1\end{pmatrix}$$

The first column of matrix $A^{-1}_{Householder}$

$$\begin{array}{rcrcrcr} x_1 & + & 2x_2 & + & 3x_3 & = & 0\\ & & x_2 & + & 2x_3 & = & 0\\ & & & & 4x_3 & = & 1\end{array}\Rightarrow x^*=\begin{pmatrix}0.25\\-0.5\\0.25\end{pmatrix}$$

Second column of matrix $A^{-1}_{Householder}$

$$\begin{array}{rcrcrcr} x_1 & + & 2x_2 & + & 3x_3 & = & 1\\ & & x_2 & + & 2x_3 & = & 0\\ & & & & 4x_3 & = & 0\end{array}\Rightarrow x^*=\begin{pmatrix}1.0\\0.0\\0.0\end{pmatrix}$$

Third column of matrix $A^{-1}_{Householder}$

$$\begin{array}{rcrcrcl} x_1 & + & 2x_2 & + & 3x_3 & = & 0 \\ & & x_2 & + & 2x_3 & = & 1 \\ & & & & 4x_3 & = & 0 \end{array} \Rightarrow x^* = \begin{pmatrix} -2.0 \\ 1.0 \\ 0.0 \end{pmatrix}$$

$$A^{-1}_{Householder} = \begin{pmatrix} 0.25 & 1.0 & -2.0 \\ -0.5 & 0.0 & 1.0 \\ 0.25 & 0.0 & 0.0 \end{pmatrix}$$