UPPSALA
UNIVERSITET

Department of Information Technology

Scientific Computing for Data Analysis

Davoud Mirzaei

Lecture 7: QR factorization (continued)

Agenda

- Solving the least squares problem using QR
- QR algorithm using Householder reflections
- Complexity of the QR algorithm
- Review some old exam questions

Solving the least squares problem using QR factorization

Coming back to the least squares problem:

$$\text{find } \boldsymbol{x} \in \mathbb{R}^n \text{ such that } \|A\boldsymbol{x} - \boldsymbol{b}\|_2 \text{ is mimimized}$$

- Remember that orthogonal matrices preserve norm 2:

$$\begin{aligned}\|A\boldsymbol{x} - \boldsymbol{b}\|_2^2 &= \|QR\boldsymbol{x} - \boldsymbol{b}\|_2^2 = \|Q^T(QR\boldsymbol{x} - \boldsymbol{b})\|_2^2 = \|R\boldsymbol{x} - Q^T\boldsymbol{b}\|_2^2 \\ &= \left\| \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} Q_1^T\boldsymbol{b} \\ Q_2^T\boldsymbol{b} \end{bmatrix} \right\|_2^2 = \|R_1\boldsymbol{x} - Q_1^T\boldsymbol{b}\|_2^2 + \|Q_2^T\boldsymbol{b}\|_2^2\end{aligned}$$

- The minimum is obtained if

$$R_1\boldsymbol{x} = Q_1^T\boldsymbol{b}$$

- So, a backward substitution gives a least square solution $\boldsymbol{x}$
- The residual then is

$$residual = \|A\boldsymbol{x} - \boldsymbol{b}\|_2 = \|Q_2^T\boldsymbol{b}\|_2$$

- If the residual is not important a reduced QR factorization is enough for obtaining the least squares solution

Steps for obtaining least squares solution of $A\boldsymbol{x} = \boldsymbol{b}$ using QR factorization:

- Compute the reduced QR factorization of A such that $A = Q_1 R_1$
- Solve the triangular system $R_1 \boldsymbol{x} = Q_1^T \boldsymbol{b}$ for $\boldsymbol{x}$ with backward substitution
- If you need to compute the residual then Q_2 is also needed, so a complete QR must be obtained to compute $residual = \|Q_2^T \boldsymbol{b}\|_2$.

Example from polynomial curve fitting

Least squares fitting, quadratic polynomial:

x	1	2	3	4	5
y	1	2	1	2	3

$$\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \\ 1 & x_4 & x_4^2 \\ 1 & x_5 & x_5^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \approx \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} \Longrightarrow \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \\ 1 & 5 & 25 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \approx \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \\ 3 \end{bmatrix}$$

```python
import numpy as np
from scipy.linalg import solve_triangular

A = np.array([[1,1,1],[1,2,4],[1,3,9],[1,4,16],[1,5,25]])
b = np.array([1,2,1,2,3])

Q,R = np.linalg.qr(A, mode = 'reduced')
a = solve_triangular(R, Q.T@b, lower = False)
print("a = ", a)
```

numpy/scipy and least squares

- `numpy.Polynomial.fit` one option
- Another option is `numpy.linalg.lstsq`: uses Lapack-drivers (written in C and Fortran)
- Another option is `scipy.linalg.lstsq`: the same as numpy

```
import scipy as sp
sp.linalg.lstsq(A,b, lapack_driver = 'gelsy')
```

LAPACK driver:

- `'gelsy'`, faster, uses QR as discussed above
- `'gelss'`, uses SVD (later in course)
- `'gelsd'`, (default), uses SVD (uses a divide and conquer technique. For large matrices it is often much faster than `'gelss'` but uses more workspace)

1TD352_Algorithm_04

For a quadratic least squares problem (with ansatz $y = a_0 + a_1 x + a_2 x^2$) on four points with function values $y = [-2, 1, 3, 2]$, the QR factorization of the matrix results in factors (rounded to 2 decimal places)

$$Q = \begin{bmatrix} -0.5 & -0.59 & 0.56 & -0.29 \\ -0.5 & -0.25 & -0.32 & 0.76 \\ -0.5 & 0.08 & -0.64 & -0.57 \\ -0.5 & 0.76 & 0.40 & 0.10 \end{bmatrix} \qquad R = \begin{bmatrix} -2. & 0.25 & -1.12 \\ 0. & 1.48 & 0.11 \\ 0. & 0. & 0.89 \\ 0. & 0. & 0. \end{bmatrix}$$

What is the computed coefficient a_2 and what is the residual of the solution (round to 2 decimal places)?
(You can use python as a calculator)

Select one alternative:

- $a_2 = 3.12,\ residual = 0.17$
- $a_2 = 3.12,\ residual = 0.23$
- $a_2 = -0.50,\ residual = 0.00$
- $a_2 = -2.88,\ residual = 0.17$
- $a_2 = -2.88,\ residual = 0.23$

Computing QR factorization via Householder reflections

The idea is to construct some appropriate orthogonal matrices $H_1, H_2, \ldots, H_n$ and multiplying them from the left by the original matrix A to construct the upper triangular matrix R.

Step 1: Find an orthogonal matrix H_1 such that

$$H_1 A = H_1 \begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{pmatrix} = \begin{pmatrix} + & + & + \\ 0 & + & + \\ 0 & + & + \\ 0 & + & + \\ 0 & + & + \end{pmatrix} =: A^{(1)}$$

Step 2: Find an orthogonal matrix H_2 such that

$$H_2 A^{(1)} = H_2 \begin{pmatrix} + & + & + \\ 0 & + & + \\ 0 & + & + \\ 0 & + & + \\ 0 & + & + \end{pmatrix} = \begin{pmatrix} + & + & + \\ 0 & * & * \\ 0 & 0 & * \\ 0 & 0 & * \\ 0 & 0 & * \end{pmatrix} =: A^{(2)}$$

Step 3: Find an orthogonal matrix H_3 such that

$$H_3A^{(2)} = H_3 \begin{pmatrix} + & + & + \\ 0 & * & * \\ 0 & 0 & * \\ 0 & 0 & * \\ 0 & 0 & * \end{pmatrix} = \begin{pmatrix} + & + & + \\ 0 & * & * \\ 0 & 0 & \star \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} =: A^{(3)} =: R$$

Then we have

$$R = A^{(3)} = H_3A^{(2)} = H_3H_2A^{(1)} = H_3H_2H_1A$$

If we define $Q^T = H_3H_2H_1$ then $R = Q^TA$ or

$$A = QR$$

Matrices H_k are called Householder matrices or Householder reflections.

How to construct Householder matrices? (next slides)

A matrix of the form

$$H = I - \frac{2}{\boldsymbol{u}^T\boldsymbol{u}}\boldsymbol{u}\boldsymbol{u}^T$$

where $\boldsymbol{u}$ is a non-zero vector in $\mathbb{R}^n$ is called a Householder matrix or a **Householder reflection**.

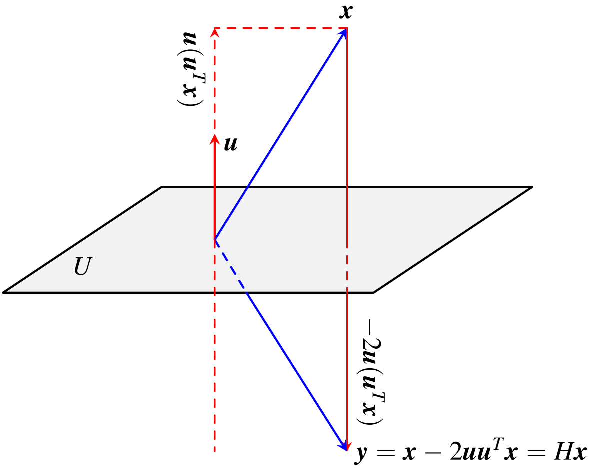
Example:

$$\boldsymbol{u} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix} \Longrightarrow H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{2}{\begin{bmatrix} 2 & 1 & 3 \end{bmatrix}\begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}} \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}\begin{bmatrix} 2 & 1 & 3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{2}{14}\begin{bmatrix} 4 & 2 & 6 \\ 2 & 1 & 3 \\ 6 & 3 & 9 \end{bmatrix} = \frac{1}{7}\begin{bmatrix} 3 & -2 & -6 \\ -2 & 6 & -3 \\ -6 & -3 & -2 \end{bmatrix}$$

Householder reflections

If $\boldsymbol{u}$ is the normal vector of a plane U then $\boldsymbol{y} = H\boldsymbol{x}$ is the reflection of $\boldsymbol{x}$ with respect to plane U. (U acts as a mirror).



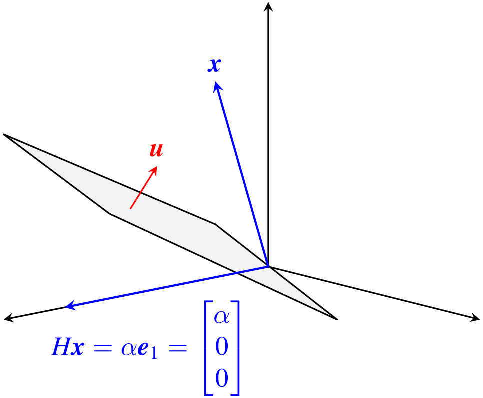
Some properties of Householder matrices

Let $H = I - 2\boldsymbol{u}\boldsymbol{u}^T/\boldsymbol{u}^T\boldsymbol{u}$ be a Householder matrix with vector $\boldsymbol{u} \in \mathbb{R}^n$. Then

1. H is symmetric (why?)
2. $H^2 = I$ (why?)
3. H is orthogonal (why?)
4. $H\boldsymbol{u} = -\boldsymbol{u}$ (why?)
5. $H\boldsymbol{v} = \boldsymbol{v}$ if $\boldsymbol{u}^T\boldsymbol{v} = 0$ i.e. (if $\boldsymbol{v}$ is in plane U) (why?)
6. If $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ are such that $\boldsymbol{x} \neq \boldsymbol{y}$ and $\|\boldsymbol{x}\|_2 = \|\boldsymbol{y}\|_2$, and $\boldsymbol{u}$ is chosen parallel to $\boldsymbol{x} - \boldsymbol{y}$ then $H\boldsymbol{x} = \boldsymbol{y}$ (why?)

How to make zeros?

Given a vector $\boldsymbol{x}$, we want to find a reflection matrix H (or the mirror) such that $H\boldsymbol{x} = \alpha \boldsymbol{e}_1$ (reflect $\boldsymbol{x}$ on x-axis)



- $\boldsymbol{u}$ must be parallel to $\boldsymbol{x} - \alpha \boldsymbol{e}_1$
- α is indeed $\mp\|\boldsymbol{x}\|_2$ because H preserves the length

An important property for our purpose: Given a nonzero vector $\boldsymbol{x} \neq \boldsymbol{e}_1$, the Householder matrix H define by

$$\boldsymbol{u} = \boldsymbol{x} \pm \|\boldsymbol{x}\|_2 \boldsymbol{e}_1 \;\; gives \;\; H\boldsymbol{x} = \mp\|\boldsymbol{x}\|_2 \boldsymbol{e}_1$$

This is a simple consequence of item 5 above by letting $\boldsymbol{y} = \mp\|\boldsymbol{x}\|_2 \boldsymbol{e}_1$.

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \Longrightarrow \boldsymbol{u} = \begin{bmatrix} x_1 \pm \|\boldsymbol{x}\|_2 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$\Longrightarrow H = I - \frac{2}{\boldsymbol{u}^T\boldsymbol{u}}\boldsymbol{u}\boldsymbol{u}^T \Longrightarrow H\boldsymbol{x} = \begin{bmatrix} \times \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Both sings $\pm$ work but to avoid cancellations error in computing u_1, we always use $\mathrm{sign}(x_1)$ instead of $\pm$ (to always have addition)

Example: making zeros using Householder matrices

Example 1: Transferring a vector to another vector with zeros under its first element:

$$\boldsymbol{x} = \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix} \Longrightarrow \boldsymbol{u} = \begin{bmatrix} x_1 + \operatorname{sign}(x_1)\|\boldsymbol{x}\|_2 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2+3 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \\ 1 \end{bmatrix}$$

$$\Longrightarrow H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{2}{30}\begin{bmatrix} 25 & 10 & 5 \\ 10 & 4 & 2 \\ 5 & 2 & 1 \end{bmatrix} = \frac{1}{15}\begin{bmatrix} -10 & -10 & -5 \\ -10 & 11 & -2 \\ -5 & -2 & 14 \end{bmatrix}$$

Now check:

$$H\boldsymbol{x} = \frac{1}{15}\begin{bmatrix} -10 & -10 & -5 \\ -10 & 11 & -2 \\ -5 & -2 & 14 \end{bmatrix}\begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} -3 \\ 0 \\ 0 \end{bmatrix}$$

Step 1: Use a Householder transformation to transfer matrix A into a new matrix that all the components of its first column except the first are annihilated:

$$A = \begin{bmatrix} 2 & 3 & 5 \\ 1 & 2 & -1 \\ 2 & 5 & 3 \\ 1 & -1 & 0 \end{bmatrix}, \text{ set } \boldsymbol{x} := \boldsymbol{a}_1 = \begin{bmatrix} 2 \\ 1 \\ 2 \\ 1 \end{bmatrix} \implies \boldsymbol{u} = \begin{bmatrix} 2+\sqrt{10} \\ 1 \\ 2 \\ 1 \end{bmatrix}$$

$$H_1 = I - \frac{2}{\boldsymbol{u}^T\boldsymbol{u}}\boldsymbol{u}\boldsymbol{u}^T \doteq \begin{bmatrix} -0.636 & -0.316 & -0.633 & -0.316 \\ -0.316 & +0.939 & -0.123 & -0.061 \\ -0.633 & -0.123 & +0.755 & -0.123 \\ -0.316 & -0.061 & -0.123 & +0.939 \end{bmatrix}$$

$$H_1 A \doteq \begin{bmatrix} -3.162 & -5.376 & -4.743 \\ 0.000 & +0.378 & -2.888 \\ 0.000 & +1.755 & -0.775 \\ 0.000 & -2.623 & -1.887 \end{bmatrix}$$

Note: All numbers are edited into 3 decimals to keep some spaces.

Step 2: From step 1 assume $A^{(1)} = H_1 A$. Use another Householder matrix to annihilate the entries under the diagonal of the second column of $A^{(1)}$:

$$A^{(1)} \doteq \begin{bmatrix} -3.162 & -5.376 & -4.743 \\ 0.000 & +0.378 & -2.888 \\ 0.000 & +1.755 & -0.775 \\ 0.000 & -2.623 & -1.887 \end{bmatrix}, \text{ set } \boldsymbol{x} := \begin{bmatrix} +0.378 \\ +1.755 \\ -2.623 \end{bmatrix}$$

$$\boldsymbol{u} = \boldsymbol{x} + \operatorname{sign}(x_1)\|\boldsymbol{x}\|_2 \doteq \begin{bmatrix} +3.556 \\ +1.755 \\ -2.623 \end{bmatrix} \Longrightarrow \widetilde{H} \doteq \begin{bmatrix} -0.119 & -0.552 & 0.825 \\ -0.552 & 0.727 & 0.407 \\ 0.825 & 0.407 & 0.391 \end{bmatrix}$$

$$H_2 \doteq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -0.119 & -0.552 & 0.825 \\ 0 & -0.552 & 0.727 & 0.407 \\ 0 & 0.825 & 0.407 & 0.391 \end{bmatrix} \Rightarrow H_2 A^{(1)} \doteq \begin{bmatrix} -3.162 & -5.376 & -4.743 \\ 0.000 & -3.178 & -0.787 \\ 0.000 & 0.000 & 0.262 \\ 0.000 & 0.000 & -3.437 \end{bmatrix}$$

Note: The first row and the first column of $A^{(1)}$ remain unchanged! So, previous zeros are not destroyed.

Step 3: From step 2 assume $A^{(2)} = H_2 A^{(1)}$. Use another Householder matrix to annihilate the entries under the diagonal of the third column of $A^{(2)}$:

$$A^{(2)} \doteq \begin{bmatrix} -3.162 & -5.376 & -4.743 \\ 0.000 & -3.178 & -0.787 \\ 0.000 & 0.000 & +0.262 \\ 0.000 & 0.000 & -3.437 \end{bmatrix} \Longrightarrow \boldsymbol{x} := \begin{bmatrix} +0.262 \\ -3.437 \end{bmatrix}$$

$$\boldsymbol{u} = \boldsymbol{x} + \operatorname{sign}(x_1)\|\boldsymbol{x}\|_2 \doteq \begin{bmatrix} +3.709 \\ -3.437 \end{bmatrix} \Longrightarrow \widetilde{H} \doteq \begin{bmatrix} -0.076 & 0.997 \\ +0.997 & 0.076 \end{bmatrix}$$

$$H_3 \doteq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -0.076 & 0.997 \\ 0 & 0 & +0.997 & 0.076 \end{bmatrix} \Rightarrow H_3 A^{(2)} \doteq \begin{bmatrix} -3.162 & -5.376 & -4.743 \\ 0.000 & -3.178 & -0.787 \\ 0.000 & 0.000 & -3.447 \\ 0.000 & 0.000 & 0.000 \end{bmatrix} =: R$$

Note: Previous zeros are not destroyed.

QR factorization using Householder matrices

Substituting back we have:

$$R = H_3 A^{(2)} = H_3 H_2 A^{(1)} = H_3 H_2 H_1 A =: Q^T A$$

Since H_1, H_2, H_3 are symmetric and orthogonal we have

$$A = QR, \quad Q = H_1 H_2 H_3$$

In the example above:

$$Q = H_1 H_2 H_3 \doteq \begin{bmatrix} -0.633 & +0.126 & -0.609 & -0.462 \\ -0.316 & -0.094 & +0.747 & -0.577 \\ -0.633 & -0.504 & +0.115 & +0.577 \\ -0.316 & +0.850 & +0.241 & +0.346 \end{bmatrix}$$

$$R \doteq \begin{bmatrix} -3.162 & -5.376 & -4.743 \\ 0.000 & -3.178 & -0.787 \\ 0.000 & 0.000 & -3.447 \\ 0.000 & 0.000 & 0.000 \end{bmatrix}$$

The procedure for a $m \times n$ matrix A is similar. It requires n steps and n Householder matrices

Complexity (computational cost) is total number of operations $(+, -, \times, /)$ in the algorithm. For example the cost of inner product $\boldsymbol{x}^T\boldsymbol{y}$ is about $2n$

- If A is $m \times n$ then H_k matrices are $m \times m$ and $A^{(k)}$ matrices are $m \times n$.
- The majority of QR cost comes from products $H_k A^{(k-1)}$ in each step to produce R and n products $H_1 H_2 \cdots H_n$ to produce Q.
- Direct computation of HA needs $2m^2n$ flops (**fl**oating-point **op**erator**s**). (because of mn inner products)
- Direct computation of products of two Householder matrices costs $2m^3$ flops. Why?
- The total cost then is $2m^2n^2 + 2m^3n$ because n products of each type are needed. If $m = n$ (square case) the cost is $4n^4$ which is not efficient!
- However, this is not the right way the QR factorization is implemented. See the next page!

Efficient implementation:

- The idea is to avoid forming H explicitly, but working with $\boldsymbol{u}$ directly.
- Exercise: If $H = I - (2/\boldsymbol{u}^T\boldsymbol{u})\boldsymbol{u}\boldsymbol{u}^T$ show that the cost of $H\boldsymbol{x}$ is about $6m$
- Exercise: If $H = I - (2/\boldsymbol{u}^T\boldsymbol{u})\boldsymbol{u}\boldsymbol{u}^T$ show that the cost of HA is about $4mn$
- Another point: Since the first k rows and columns of $A^{(k)}$ and $A^{(k+1)}$ are the same, at step k we can only multiply $\widetilde{H}_k$ and the corresponding submatrix of size $(m-k)\times(n-k)$.
- These all result in final cost $2mn^2 - \frac{2}{3}n^3$ for computing R and $\frac{4}{3}m^3$ for Q. If $m = n$ the complexity is $\frac{4}{3}n^3$ for both R and Q.
- Exercise: How can we optimize the *space complexity* in QR algorithm?

QR factorization: conclusion and some points

- QR-factorization $A = QR$ (where Q is orthogonal and R is triangular) is one example of matrix decomposition
- The reduced QR-factorization $A = Q_1 R_1$ can be used to solve the least square problem $\min \|A\boldsymbol{x} - \boldsymbol{b}\|_2$ by reducing it to triangular system
$$R_1 \boldsymbol{x} = Q_1^T \boldsymbol{b}$$
- If the residual is requested, use a full QR and $residual = \|Q_2^T \boldsymbol{b}\|_2$
- The computational cost using Householder matrices for the case of $m = n$ (when A is square) is approximately $\frac{4}{3}n^3$. This cost is twice of that of Gaussian elimination (LU decomposition).
- Using QR factorization for solving square systems offers better stability, but at a higher computational cost, when compared to Gaussian elimination.
- QR-factorization has applications in many other areas, for example in computations of eigenvalues (later in the course)

Exam questions

1TD352_concept_01

Select all correct statements regarding the QR factorization:

- [ ] *The Householder method is the only algorithm to compute it*
- [x] *Its time complexity using Householder method is* $O(n^3)$ *when the matrix is* $(n \times n)$
- [x] *If* A *is a square matrix then both its* Q *and* R *factors are square*
- [ ] *Both matrices* Q *and* R *are orthogonal*
- [x] *It can be used for solving the least squares problem*
- [x] *It can be used for solving the square linear system* $Ax = b$ *for nonsingular matrix* A
- [ ] *Its time complexity using Householder method is* $O(n^4)$ *when the matrix is* $(n \times n)$

Exam questions

3 (Algorithm)

A matrix A has the QR-factorization

$$Q = \begin{pmatrix} -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \quad R = \begin{pmatrix} -2 & -3 & -2 \\ 0 & -5 & 2 \\ 0 & 0 & -4 \end{pmatrix}.$$

Given a right-hand-side

$$y = \begin{pmatrix} 2 \\ -2 \\ 8 \\ -4 \end{pmatrix}$$

solve the least squares problem $Ax = y$ using the QR-factorization and without forming the matrix A explicitly.