HW4 from Numerical Analysis and Scientific Computation (Due Dec 2)

- Chapter 2.6 Problem 4abcde, 14
- Chapter 2.7 Problem 2, 10ab

§2.6

4. a. What matrix A has $Q_1 = [1/\sqrt{2}\ \ 0; 0\ \ 1; -1/\sqrt{2}\ \ 0]$ and $R_1 = [1\ \ 2; 0\ \ 3]$ as its reduced QR factorization?

b. What is the full QR decomposition of A? (*Hint:* The only orthogonal matrices that have the columns of Q_1 as their first two columns are $[1/\sqrt{2}\ \ 0\ \ 1/\sqrt{2}; 0\ \ 1\ \ 0; -1/\sqrt{2}\ \ 0\ \ 1/\sqrt{2}]$ and $[1/\sqrt{2}\ \ 0\ \ -1/\sqrt{2}; 0\ \ 1\ \ 0; -1/\sqrt{2}\ \ 0\ \ -1/\sqrt{2}]$.)

4a

$$A = Q_1R_1 = \begin{bmatrix} 1/\sqrt{2} & 0 \\ 0 & 1 \\ -1/\sqrt{2} & 0 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & 2/\sqrt{2} \\ 0 & 3 \\ -1/\sqrt{2} & -2/\sqrt{2} \end{bmatrix}$$

4b. To find a full QR factorization it is enough to add a zero row to R and an orthonormal column to Q. By the hint, there are two possible choices for Q. Either

$$Q = \begin{bmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 0 & 1 & 0 \\ -1/\sqrt{2} & 0 & 1/\sqrt{2} \end{bmatrix} \quad \text{or} \quad Q = \begin{bmatrix} 1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 0 & 1 & 0 \\ -1/\sqrt{2} & 0 & -1/\sqrt{2} \end{bmatrix}.$$

It doesn't matter which of these are chosen, since the last column will be multiplied by zero in R anyway. In particular

$$R = \begin{bmatrix} 1 & 2 \\ 0 & 3 \\ 0 & 0 \end{bmatrix}$$

so that

$$A = QR = \begin{bmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 0 & 1 & 0 \\ -1/\sqrt{2} & 0 & 1/\sqrt{2} \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 0 & 3 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & 2/\sqrt{2} \\ 0 & 3 \\ -1/\sqrt{2} & -2/\sqrt{2} \end{bmatrix},$$

which is the same as the matrix in part a.

§2.6 #4

c. Use the reduced QR factorization of A to find the least squares solution of $Ax = (1, 1, 1)^T$. What is the norm of the residual (Eq. (2.22))?

d. Verify that $R^T R = R_1^T R_1 = A^T A$.

e. Find the least squares solution of $Ax = (1, 1, 1)^T$ by forming the normal equation and using the Cholesky decomposition.

4c. To minimize $\|Ax-b\|$ we solve $R_1 x = Q_1^T b$. Thus,

$$Q_1^T b = \begin{bmatrix} 1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

and

$$\begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \text{means} \quad \begin{aligned} x_1 + 2x_2 &= 0 \\ 3x_2 &= 1 \end{aligned}$$

Therefore, using backsubstitution one obtains

$$x_2 = 1/3$$
$$x_1 = -2x_2 = -2/3$$

so that the minimizer of $\|Ax-b\|$ is $x = \begin{bmatrix} -2/3 \\ 1/3 \end{bmatrix}$.

4d.

$$R^T R = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 3 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & 3 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 13 \end{bmatrix}$$

$$R_1^T R_1 = \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 13 \end{bmatrix}$$

$$A^T A = \begin{bmatrix} 1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 2/\sqrt{2} & 3 & -2/\sqrt{2} \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 2/\sqrt{2} \\ 0 & 3 \\ -1/\sqrt{2} & -2/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 13 \end{bmatrix}$$

these are all the same.

4e. The cholesky decomposition of $\begin{bmatrix} 1 & 2 \\ 2 & 13 \end{bmatrix}$ is given by

Step 1: $R_1 = \sqrt{1} = 1$

Step 2: $p = 2$

Step 3: $1\gamma_1 = 2$, $\gamma_1 = 2$

Step 4: $r_{22} = \sqrt{13 - 2 \cdot 2} = \sqrt{9} = 3$

Step 5: $R_2 = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}$

Cholesky Decomposition Algorithm:

1. Set $R_1 = \sqrt{a_{11}}$.
2. Begin loop ($p = 2$ to n):
3. Solve $R_{p-1}^T \gamma_{p-1} = c_{p-1}$ for γ_{p-1} by forward substitution.
4. Set $r_{pp} = \sqrt{a_{pp} - \gamma_{p-1}^T \gamma_{p-1}}$.
5. Set $R_p = \begin{pmatrix} R_{p-1} & \gamma_{p-1} \\ 0 & r_{pp} \end{pmatrix}$.
6. End loop.
7. Set $L = R_n^T$.

Note: This is the same as R in the reduced QR factorization of the first part.

Check that $A^T A = R_2^T R_2$

$$R_2^T R_2 = \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 13 \end{bmatrix} = A^T A.$$

Now solve the normal equations $A^T A x = A^T b$ using the cholesky factorization as $\begin{cases} R^T y = A^T b \\ R x = y \end{cases}$

First

$$A^T b = \begin{bmatrix} 1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 2/\sqrt{2} & 3 & -2/\sqrt{2} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

Now

$$\begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

$y_1 = 0$

$2y_1 + 3y_2 = 3$

so $y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Next

$$\begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$x_1 + 2x_2 = 0$

$3x_2 = 1$

so $x = \begin{bmatrix} -2/3 \\ 1/3 \end{bmatrix}$

This is the same answer found in part 4c.

§2.6

14. Show that if A is positive definite, then there is a positive definite matrix S such that $S^2 = A$. (This is called the matrix square root of A; see Section 2.4.) Use the ~~MATLAB~~ command ~~sqrtm~~ sqrt to compute the matrix square root of $A = [4\ \ 2\ \ 1; 2\ \ 3\ \ 1; 1\ \ 1\ \ 4]$.

```
julia> using LinearAlgebra

julia> A=[4 2 1;2 3 1;1 1 4]
3×3 Array{Int64,2}:
 4  2  1
 2  3  1
 1  1  4

julia> S=sqrt(A)
3×3 Array{Float64,2}:
 1.90975   0.55058   0.222991
 0.55058   1.624     0.243911
 0.222991  0.243911  1.97251

julia> S*S
3×3 Array{Float64,2}:
 4.0  2.0  1.0
 2.0  3.0  1.0
 1.0  1.0  4.0
```

If A is positive definite then by the Spectral Theorem $A = QDQ^T$ where D is a diagonal matrix with positive entries and $Q^T = Q^{-1}$ is orthogonal. Let

$$S = \begin{bmatrix} \sqrt{\lambda_1} & & & 0 \\ & \sqrt{\lambda_2} & & \\ & & \ddots & \\ 0 & & & \sqrt{\lambda_n} \end{bmatrix}$$

where λ_i are the diagonal entries of D.

That is λ_i are the positive eigenvalues of A and $\sqrt{\lambda_i}$ are their positive square roots. It follows that S and consequently QSQ^T are both positive definite matrices. Moreover,

$$(QSQ^T)(QSQ^T) = QS\cancel{Q^T}\cancel{Q}SQ^T = QS^2Q^T = QDQ^T = A$$

shows that $A^{1/2} = QSQ^T$ is a positive definite square root of A.

§2.7

2. Repeat Example 2.7.2 for the matrix $A = [2\ \ 2\ \ 3;\ 4\ \ 5\ \ 6; 7\ \ 8\ \ 9]$. (Supply all intermediate details as in the example; use MATLAB for the multiplications.) Include a check of your final result ($A = QR$).

```
julia> A=[2 2 3;4 5 6;7 8 9]
3×3 Array{Int64,2}:
 2  2  3
 4  5  6
 7  8  9

julia> c=norm(A[:,1])
8.306623862918075

julia> e1=[1,0,0]
3-element Array{Int64,1}:
 1
 0
 0

julia> norm(A[:,1]+c*e1)
13.085354234856322

julia> norm(A[:,1]-c*e1)
10.235892953149115
```

bigger, so choose the +c version of H_1.

```
julia> v=A[:,1]+c*e1
3-element Array{Float64,1}:
 10.306623862918075
  4.0
  7.0

julia> v=v/norm(v)
3-element Array{Float64,1}:
 0.7876457662463306
 0.3056852667652616
 0.5349492168392077
```

```
julia> H1=I-2*v*v'
3×3 Array{Float64,2}:
 -0.240772  -0.481543  -0.842701
 -0.481543   0.813113  -0.327052
 -0.842701  -0.327052   0.427659

julia> A2=H1*A
3×3 Array{Float64,2}:
 -8.30662       -9.63087    -11.1959
 -4.44089e-16    0.486061     0.490578
  0.0            0.100606    -0.641488
```

The first reflector

These are essentially zero

now work on the 2x2 submatrix

```
julia> c=norm(A2[2:3,2])
0.49636358810271547

julia> e1=[1,0]
2-element Array{Int64,1}:
 1
 0

julia> norm(A2[2:3,2]+c*e1)
0.9875623704594368

julia> norm(A2[2:3,2]-c*e1)
0.10113263978235092
```

larger value so choose +c version of H_2

```
julia> v2=A2[2:3,2]+c*e1
2-element Array{Float64,1}:
 0.9824244353572946
 0.10060648269551464

julia> v2=v2/norm(v2)
2-element Array{Float64,1}:
 0.9947973563434257
 0.10187354814735416

julia> v2=[0; v2]
3-element Array{Float64,1}:
 0.0
 0.9947973563434257
 0.10187354814735416

julia> H2=I-2*v2*v2'
3×3 Array{Float64,2}:
  1.0   -0.0        -0.0
 -0.0   -0.979244   -0.202687
 -0.0   -0.202687    0.979244

julia> H2*A2
3×3 Array{Float64,2}:
 -8.30662       -9.63087    -11.1959
  4.34871e-16   -0.496364    -0.350374
  9.00111e-17    0.0         -0.727607
```

The result $R = H_2 H_1 A$ is this upper triangular matrix.

numerically zero

Now check the result.

```
julia> R=UpperTriangular(H2*A2)
3×3 UpperTriangular{Float64,Array{Float64,2}}:
 -8.30662  -9.63087   -11.1959
   ⋅        -0.496364   -0.350374
   ⋅          ⋅         -0.727607

julia> Q=H1*H2
3×3 Array{Float64,2}:
 -0.240772   0.642353  -0.727607
 -0.481543  -0.729946  -0.485071
 -0.842701   0.233583   0.485071

julia> Q*R
3×3 Array{Float64,2}:
 2.0  2.0  3.0
 4.0  5.0  6.0
 7.0  8.0  9.0
```

← This is the same as A.

remove the terms that were numerically zero by converting to an upper triangular matrix.

Since the H_m are orthogonal and symmetric, then $H_m^{-1} = H_m$ and $R = H_2 H_1 A$ implies $Q = H_1 H_2$.

§2.8

10. a. Find the QR decomposition of the Hilbert matrix of order $N = 4$. Use it to solve $QRx = b$, where b is a vector of ones. Find the relative error in your answer (use ~~invhilb(A)*b~~ inv(Hilbert(N))*b to determine the true solution). Give the condition numbers of the Hilbert matrix, Q, and R.

b. Repeat for $N = 5, 6, 7, 8, 9, 10, 11, 12$. Comment.

The 4×4 Hilbert matrix is

```
julia> using SpecialMatrices, LinearAlgebra

julia> N=4
4

julia> H=Hilbert(N)
4×4 Hilbert{Rational{Int32}}:
 1//1  1//2  1//3  1//4
 1//2  1//3  1//4  1//5
 1//3  1//4  1//5  1//6
 1//4  1//5  1//6  1//7
```

The QR factorization is

```
julia> Q,R=qr(H)
LinearAlgebra.QRCompactWY{Float64, Matrix{Float64}}
Q factor:
4×4 LinearAlgebra.QRCompactWYQ{Float64, Matrix{Float64}}:
 -0.838116   0.522648  -0.153973  -0.0263067
 -0.419058  -0.441713   0.727754   0.31568
 -0.279372  -0.528821  -0.139506  -0.7892
 -0.209529  -0.502072  -0.653609   0.526134
R factor:
4×4 Matrix{Float64}:
 -1.19315  -0.670493  -0.474933    -0.369835
  0.0      -0.118533  -0.125655    -0.117542
  0.0       0.0       -0.00622177  -0.00956609
  0.0       0.0        0.0          0.000187905
```

Solving $QRx=b$ or $Rx=Q^Tb$ yields

```
julia> b=ones(N)
4-element Vector{Float64}:
 1.0
 1.0
 1.0
 1.0

julia> x=R\Q'*b
4-element Vector{Float64}:
   -3.999999999999204
   59.99999999998727
 -179.99999999996953
  139.9999999999809
```

The exact solution and relative error is

```
julia> xexact=inv(H)*b
4-element Vector{Float64}:
   -4.0
   60.0
 -180.0
  140.0

julia> norm(x-xexact)/norm(xexact)
1.617925648933708e-13
```

The condition numbers of H, Q and R are

```
julia> cond(H)
15513.738738928138

julia> cond(Q)
1.0000000000000004

julia> cond(R)
15513.738738929966
```

Problem 10b was solved by writing a program and running it. The program and resulting output is included on the next page.

Note that as N increases the condition numbers of H.
In summary,

```
using SpecialMatrices, LinearAlgebra, Printf

@printf("%4s %15s %15s %15s\n",
    "N","κ(H)","relerr","18-log(κ(H))")
for N=5:12
    H=Hilbert(N)
    Q,R=qr(H)
    b=ones(N)
    x=R\Q'*b
    bexact=ones(BigInt,N)
    xexact=inv(H)*bexact
    relerr=norm(x-xexact)/norm(xexact)
    @printf("%4d %15.7e %15.7e %15.6f\n",
        N,cond(H),relerr,18-log10(cond(H)))
end
```

Since the number d of significant digits lost solving for x is given by cond(H) ~ 10^d, then taking logarithms yields a relationship between relative error and the condition number

Produces the output

cond(H)   log10

```
$ julia table.jl
   N            κ(H)          relerr    18-log(κ(H))
   5   4.7660725e+05   4.0720278e-12       12.321839
   6   1.4951059e+07   1.2485577e-10       10.825328
   7   4.7536736e+08   5.2676478e-09        9.322971
   8   1.5257576e+10   3.3012230e-08        7.816514
   9   4.9315395e+11   2.9718703e-06        6.307017
  10   1.6024868e+13   2.3711508e-04        4.795206
  11   5.2237331e+14   6.2671922e-04        3.282019
  12   1.7341931e+16   6.3839821e-02        1.760903
```

about the same

The constant 18 was determined by inspection and is related to how many significant digits used by the floating point arithmetic.

which shows a relation between the relative error and the condition number.

```julia
using SpecialMatrices, LinearAlgebra

for N=5:12
    println("---------------- N = $(N) ----------------")
    H=Hilbert(N)
    println("\nThe NxN Hilbert matrix H is")
    display(H)
    Q,R=qr(H)
    println("\n\nH=QR factorization where Q=R")
    display(Q)
    println("\nand R=")
    display(R)
    b=ones(N)
    x=R\Q'*b
    println("\n\nThe solution to Rx=Q'b is")
    display(x)
    bexact=ones(BigInt,N)
    xexact=inv(H)*bexact
    println("\n\nThe exact solution to Hx=b is")
    display(xexact)
    relerr=norm(x-xexact)/norm(xexact)
    println("\n\nThe relative error is ",relerr)
    println("\ncond(H) = ",cond(H))
    println("cond(Q) = ",cond(Q))
    println("cond(R) = ",cond(R),"\n")
end

```

```
---------------- N = 5 ----------------

The NxN Hilbert matrix H is
5×5 Hilbert{Rational{Int64}}:
 1//1  1//2  1//3  1//4  1//5
 1//2  1//3  1//4  1//5  1//6
 1//3  1//4  1//5  1//6  1//7
 1//4  1//5  1//6  1//7  1//8
 1//5  1//6  1//7  1//8  1//9

H=QR factorization where Q=R
5×5 LinearAlgebra.QRCompactWYQ{Float64, Matrix{Float64}}:
 -0.826584   0.533355  -0.175305    0.0391021   0.00550474
 -0.413292  -0.374054   0.717262   -0.403345   -0.110095
 -0.275528  -0.462946   0.0576647   0.678965    0.495426
 -0.206646  -0.443319  -0.352626    0.206154   -0.770663
 -0.165317  -0.405914  -0.571955   -0.576447    0.385331
and R=
5×5 Matrix{Float64}:
 -1.2098  -0.68882  -0.492014    -0.385411     -0.317759
  0.0     -0.13006  -0.140192    -0.1327       -0.122323
  0.0      0.0      -0.00806538  -0.0126325    -0.0149083
  0.0      0.0       0.0         -0.000338122  -0.000689315
  0.0      0.0       0.0          0.0           8.73768e-6

The solution to Rx=Q'b is
5-element Vector{Float64}:
     4.9999999999558895
  -119.99999999926149
   629.9999999969586
 -1119.9999999955471
   629.9999999978609

The exact solution to Hx=b is
5-element Vector{Rational{BigInt}}:
     5//1
  -120//1
   630//1
 -1120//1
   630//1

The relative error is 4.072027813021613767856359248654849223375418887816832162090715363069068443310927e-12
```

```
cond(H) = 476607.25024331047
cond(Q) = 1.0000000000000004
cond(R) = 476607.25024003594

--------------- N = 6 ---------------

The NxN Hilbert matrix H is
6×6 Hilbert{Rational{Int64}}:
 1//1  1//2  1//3  1//4  1//5   1//6
 1//2  1//3  1//4  1//5  1//6   1//7
 1//3  1//4  1//5  1//6  1//7   1//8
 1//4  1//5  1//6  1//7  1//8   1//9
 1//5  1//6  1//7  1//8  1//9   1//10
 1//6  1//7  1//8  1//9  1//10  1//11

H=QR factorization where Q=R
6×6 LinearAlgebra.QRCompactWYQ{Float64, Matrix{Float64}}:
 -0.81885    0.539681  -0.189261  -0.0482361   0.00902961  -0.00110499
 -0.409425  -0.331988   0.702417   0.448926   -0.161653     0.0331496
 -0.27295   -0.421935   0.152933  -0.572323    0.58539     -0.232047
 -0.204713  -0.406725  -0.201513  -0.38664    -0.468685     0.618792
 -0.16377   -0.373526  -0.396261   0.0915169  -0.428541    -0.696141
 -0.136475  -0.339931  -0.499772   0.557422    0.477276     0.278456
and R=
6×6 Matrix{Float64}:
 -1.22122  -0.701872  -0.50447     -0.396969     -0.328434     -0.280613
  0.0      -0.138467  -0.15113     -0.144364     -0.134008     -0.123669
  0.0       0.0       -0.00956161  -0.0151932    -0.0181303    -0.0195317
  0.0       0.0        0.0          0.000480282   0.000994238   0.0014191
  0.0       0.0        0.0          0.0           1.7339e-5     4.40307e-5
  0.0       0.0        0.0          0.0           0.0           3.98624e-7

The solution to Rx=Q'b is
6-element Vector{Float64}:
    -6.000000001705303
   210.00000004247704
 -1680.0000002736924
  5040.000000687316
 -6300.000000740169
  2772.000000286731

The exact solution to Hx=b is
6-element Vector{Rational{BigInt}}:
    -6//1
```

```
   210//1
 -1680//1
  5040//1
 -6300//1
  2772//1

The relative error is 1.2485577493947676524795599529224212027954003515098907178
7503309345534642895066e-10

cond(H) = 1.4951058641250091e7
cond(Q) = 1.0000000000000007
cond(R) = 1.4951058641925327e7

--------------- N = 7 ---------------

The NxN Hilbert matrix H is
7×7 Hilbert{Rational{Int64}}:
 1//1  1//2  1//3  1//4   1//5   1//6   1//7
 1//2  1//3  1//4  1//5   1//6   1//7   1//8
 1//3  1//4  1//5  1//6   1//7   1//8   1//9
 1//4  1//5  1//6  1//7   1//8   1//9   1//10
 1//5  1//6  1//7  1//8   1//9   1//10  1//11
 1//6  1//7  1//8  1//9   1//10  1//11  1//12
 1//7  1//8  1//9  1//10  1//11  1//12  1//13

H=QR factorization where Q=R
7×7 LinearAlgebra.QRCompactWYQ{Float64, Matrix{Float64}}:
 -0.813305   0.543794  -0.199095  -0.0551133   0.0119578  -0.00196841  -0.000215
862
 -0.406652  -0.303317   0.688561   0.475965   -0.197393    0.0541101    0.009066
22
 -0.271102  -0.39395    0.207135  -0.490111    0.61084    -0.326878    -0.090662
2
 -0.203326  -0.381744  -0.112389  -0.439598   -0.254179    0.641039     0.362649
 -0.162661  -0.351413  -0.291518  -0.112277   -0.499207   -0.202238    -0.679966
 -0.135551  -0.320235  -0.389192   0.230887   -0.150595   -0.541822     0.59837
 -0.116186  -0.292096  -0.440745   0.520624    0.501273    0.380549    -0.199457
and R=
7×7 Matrix{Float64}:
 -1.22955  -0.711642  -0.513963   -0.405899     -0.336772    -0.288395    -0.252
494
  0.0      -0.144874  -0.159665   -0.153626     -0.143414    -0.132953    -0.123
278
  0.0       0.0       -0.0107957  -0.0173527    -0.0208948   -0.0226743   -0.023
```

```
4339
  0.0       0.0        0.0         0.000611647   0.00128213   0.00184868   0.002
28727
  0.0       0.0        0.0         0.0           2.66433e-5   6.85455e-5   0.000
114852
  0.0       0.0        0.0         0.0           0.0          8.5868e-7    2.610
33e-6
  0.0       0.0        0.0         0.0           0.0          0.0         -1.797
06e-8

The solution to Rx=Q'b is
7-element Vector{Float64}:
      7.000000073077899
   -336.00000296445796
   3780.0000275513157
 -16800.00010466203
  34650.000188946724
 -33264.00016145781
  12012.000052550808

The exact solution to Hx=b is
7-element Vector{Rational{BigInt}}:
      7//1
   -336//1
   3780//1
 -16800//1
  34650//1
 -33264//1
  12012//1

The relative error is 5.26764784060677669817257335587866451441756004492777149708
7119928821312316910053e-09

cond(H) = 4.7536735647344047e8
cond(Q) = 1.0000000000000009
cond(R) = 4.7536735763412255e8

---------------- N = 8 ----------------

The NxN Hilbert matrix H is
8×8 Hilbert{Rational{Int64}}:
 1//1  1//2  1//3   1//4   1//5   1//6   1//7   1//8
 1//2  1//3  1//4   1//5   1//6   1//7   1//8   1//9
 1//3  1//4  1//5   1//6   1//7   1//8   1//9   1//10
```

```
 1//4  1//5  1//6   1//7   1//8   1//9   1//10  1//11
 1//5  1//6  1//7   1//8   1//9   1//10  1//11  1//12
 1//6  1//7  1//8   1//9   1//10  1//11  1//12  1//13
 1//7  1//8  1//9   1//10  1//11  1//12  1//13  1//14
 1//8  1//9  1//10  1//11  1//12  1//13  1//14  1//15

H=QR factorization where Q=R
8×8 LinearAlgebra.QRCompactWYQ{Float64, Matrix{Float64}}:
 -0.809134   0.546655   0.206392   -0.0604753   0.0144143  -0.0027762   0.000412
786   4.13694e-5
 -0.404567  -0.28253   -0.67657     0.493355   -0.223607    0.0713471  -0.016129
9    -0.00231669
 -0.269711  -0.373639  -0.24129    -0.427368    0.613506   -0.385444    0.144399
      0.0312753
 -0.202284  -0.363608   0.0543266  -0.451867   -0.110896    0.592016   -0.476199
     -0.173752
 -0.161827  -0.335355   0.222629   -0.212438   -0.448134    0.0757705   0.576005
      0.477817
 -0.134856  -0.305932   0.316135    0.0579406  -0.335968   -0.442769    0.050805
     -0.688056
 -0.115591  -0.279243   0.366918    0.293215    0.0291512  -0.331777   -0.574624
      0.496929
 -0.101142  -0.255918   0.392873    0.481305    0.496545    0.428941    0.295917
     -0.14198
and R=
8×8 Matrix{Float64}:
 -1.23589  -0.71923   -0.521442   -0.413013     -0.343473    -0.294696    -0.258
423     -0.230313
  0.0      -0.149919  -0.166512   -0.161161     -0.151153    -0.140662    -0.130
84      -0.121966
  0.0       0.0        0.0118284   0.0191923     0.0232833    0.0254209    0.026
4065     0.0267049
  0.0       0.0        0.0         0.000731517   0.00154947   0.00225333   0.002
80789    0.00322702
  0.0       0.0        0.0         0.0           3.61918e-5   9.41426e-5   0.000
159206   0.000222793
  0.0       0.0        0.0         0.0           0.0          1.41303e-6   4.344
47e-6    8.36031e-6
  0.0       0.0        0.0         0.0           0.0          0.0          4.154
25e-8    1.47074e-7
  0.0       0.0        0.0         0.0           0.0          0.0          0.0
        -8.03602e-10

The solution to Rx=Q'b is
```

```
8-element Vector{Float64}:
      -7.999999611951353
     503.99998378288
   -7559.999741487205
   46199.99839654565
 -138599.99526166916
  216215.9928253889
 -168167.99462258816
   51479.99841931462

The exact solution to Hx=b is
8-element Vector{Rational{BigInt}}:
      -8//1
     504//1
   -7560//1
   46200//1
 -138600//1
  216216//1
 -168168//1
   51480//1

The relative error is 3.30122296850181669020746089429976036948746563957310325256142119358396529833279e-08

cond(H) = 1.525757551642611e10
cond(Q) = 1.0000000000000004
cond(R) = 1.5257575354081062e10

--------------- N = 9 ---------------

The NxN Hilbert matrix H is
9×9 Hilbert{Rational{Int64}}:
 1//1  1//2   1//3   1//4   1//5   1//6   1//7   1//8   1//9
 1//2  1//3   1//4   1//5   1//6   1//7   1//8   1//9   1//10
 1//3  1//4   1//5   1//6   1//7   1//8   1//9   1//10  1//11
 1//4  1//5   1//6   1//7   1//8   1//9   1//10  1//11  1//12
 1//5  1//6   1//7   1//8   1//9   1//10  1//11  1//12  1//13
 1//6  1//7   1//8   1//9   1//10  1//11  1//12  1//13  1//14
 1//7  1//8   1//9   1//10  1//11  1//12  1//13  1//14  1//15
 1//8  1//9   1//10  1//11  1//12  1//13  1//14  1//15  1//16
 1//9  1//10  1//11  1//12  1//13  1//14  1//15  1//16  1//17

H=QR factorization where Q=R
9×9 LinearAlgebra.QRCompactWYQ{Float64, Matrix{Float64}}:
```

```
 -0.805884    0.548745   0.212017   -0.0647694   0.0164927  -0.00351697   0.0006
15949  -8.41809e-5   7.81634e-6
 -0.402942   -0.266772  -0.666365    0.505198   -0.24357     0.0856727   -0.0227
202     0.00443313  -0.000562776
 -0.268628   -0.358229  -0.264392   -0.378624    0.607285   -0.423801     0.1864
15     -0.0547892    0.00984859
 -0.201471   -0.349844   0.0138322  -0.449277   -0.0125333   0.530283    -0.5274
05      0.265044    -0.072223
 -0.161177   -0.323166   0.174178   -0.266485   -0.379287    0.226684     0.4226
22     -0.566593     0.270836
 -0.134314   -0.295074   0.264551   -0.0437379  -0.385232   -0.271316     0.3190
37      0.426426    -0.563339
 -0.115126   -0.269486   0.314672    0.155525   -0.172589   -0.42605     -0.2878
28      0.25459      0.657229
 -0.100735   -0.247074   0.341243    0.317624    0.139477   -0.157156    -0.4443
56     -0.554939    -0.402385
 -0.0895426  -0.227639   0.353726    0.444267    0.479528    0.447913     0.3548
04      0.226031     0.100596
and R=
9×9 Matrix{Float64}:
 -1.24087  -0.725295  -0.527488   -0.418815     -0.348981    -0.299908     -0.26
3355     -0.234984     -0.212277
  0.0      -0.153995  -0.172127   -0.167412     -0.157633    -0.147167     -0.13
7264     -0.128252     -0.12016
  0.0       0.0        0.0127039   0.020775      0.0253625    0.0278351     0.02
90407     0.0294786     0.0294459
  0.0       0.0        0.0         0.000840305   0.00179566   0.00263046    0.00
329805    0.00381032    0.00419341
  0.0       0.0        0.0         0.0           4.56801e-5   0.000119951   0.00
0204476   0.000288107   0.000365262
  0.0       0.0        0.0         0.0           0.0          2.03414e-6    6.31
581e-6    1.22568e-5    1.91553e-5
  0.0       0.0        0.0         0.0           0.0          0.0           7.26
299e-8    2.59722e-7    5.645e-7
  0.0       0.0        0.0         0.0           0.0          0.0           0.0
          1.97637e-9    7.98575e-9
  0.0       0.0        0.0         0.0           0.0          0.0           0.0
          0.0           3.57254e-11

The solution to Rx=Q'b is
9-element Vector{Float64}:
       8.999956727115205
    -719.9966789949685
   13859.942553758621
```

```
 -110879.584389925
  450448.45940589905
      -1.0090048272972107e6
       1.2612563304672241e6
 -823677.770368576
  218789.44632053375

The exact solution to Hx=b is
9-element Vector{Rational{BigInt}}:
        9//1
     -720//1
    13860//1
  -110880//1
   450450//1
 -1009008//1
  1261260//1
  -823680//1
   218790//1

The relative error is 2.97187031427254243950528209929537480780746282287315853572
3470452449254891769 22e-06

cond(H) = 4.931539500738534e11
cond(Q) = 1.0000000000000007
cond(R) = 4.931536560992592e11

---------------- N = 10 ----------------

The NxN Hilbert matrix H is
10×10 Hilbert{Rational{Int64}}:
 1//1   1//2   1//3   1//4   1//5   1//6   1//7   1//8   1//9   1//10
 1//2   1//3   1//4   1//5   1//6   1//7   1//8   1//9   1//10  1//11
 1//3   1//4   1//5   1//6   1//7   1//8   1//9   1//10  1//11  1//12
 1//4   1//5   1//6   1//7   1//8   1//9   1//10  1//11  1//12  1//13
 1//5   1//6   1//7   1//8   1//9   1//10  1//11  1//12  1//13  1//14
 1//6   1//7   1//8   1//9   1//10  1//11  1//12  1//13  1//14  1//15
 1//7   1//8   1//9   1//10  1//11  1//12  1//13  1//14  1//15  1//16
 1//8   1//9   1//10  1//11  1//12  1//13  1//14  1//15  1//16  1//17
 1//9   1//10  1//11  1//12  1//13  1//14  1//15  1//16  1//17  1//18
 1//10  1//11  1//12  1//13  1//14  1//15  1//16  1//17  1//18  1//19

H=QR factorization where Q=R
10×10 LinearAlgebra.QRCompactWYQ{Float64, Matrix{Float64}}:
 -0.80328     0.55033    0.216481   -0.0682831   0.0182675  -0.0041892   0.00081
```

```
6976  -0.000132174   1.68094e-5   1.46064e-6
 -0.40164    -0.254417  -0.657679    0.51362    -0.259222    0.0976902  -0.02874
67     0.00661795   -0.00114721  -0.000131459
 -0.26776    -0.346139  -0.280854   -0.339936    0.59766    -0.4499      0.21976
5     -0.0761352     0.0186571    0.00289211
 -0.20082    -0.339042  -0.0158501  -0.441101    0.057403    0.471067   -0.54812
9      0.329261     -0.122944    -0.0269932
 -0.160656   -0.3136     0.138397   -0.297418   -0.315083    0.309618    0.28624
3     -0.568458      0.388197     0.131592
 -0.13388    -0.286551   0.226325   -0.108015   -0.386287   -0.128829    0.40966
       0.176258     -0.58018     -0.368459
 -0.114754   -0.261827   0.275879    0.0658122  -0.263812   -0.376573   -0.02699
07     0.439747      0.234963     0.614101
 -0.10041    -0.240132   0.302857    0.209404   -0.0460717  -0.328877   -0.39675
8     -0.0963189     0.397264    -0.60157
 -0.0892533  -0.221296   0.31625     0.322969    0.205626   -0.0274633  -0.30245
7     -0.496897     -0.505156     0.319585
 -0.080328   -0.204944   0.321313    0.41084     0.457693    0.450493    0.38865
       0.286312      0.170352    -0.0710189
and R=
10×10 Matrix{Float64}:
 -1.2449  -0.730254  -0.532477   -0.423641     -0.353591    -0.304294     -0.267
524     -0.23895      -0.216054     -0.197267
  0.0     -0.157357  -0.176816   -0.172683     -0.16314     -0.152733     -0.142
791     -0.133686     -0.125472     -0.118095
  0.0      0.0        0.0134548   0.022149      0.0271856    0.0299697     0.031
3865     0.0319636     0.0320187     0.0317474
  0.0      0.0        0.0         0.000938887   0.00202152   0.00298003    0.003
75637    0.00435984    0.00481744    0.00515741
  0.0      0.0        0.0         0.0           5.49241e-5   0.000145408   0.000
249603   0.000353809   0.000450912   0.000537824
  0.0      0.0        0.0         0.0           0.0          2.69847e-6    8.450
72e-6    1.65221e-5    2.59893e-5    3.60732e-5
  0.0      0.0        0.0         0.0           0.0          0.0           1.101
9e-7     3.97533e-7    8.70741e-7    1.50382e-6
  0.0      0.0        0.0         0.0           0.0          0.0           0.0
         3.64773e-9    1.4872e-8     3.6032e-8
  0.0      0.0        0.0         0.0           0.0          0.0           0.0
         0.0           9.28432e-11   4.21587e-10
  0.0      0.0        0.0         0.0           0.0          0.0           0.0
         0.0           0.0          -1.58157e-12

The solution to Rx=Q'b is
10-element Vector{Float64}:
```

```
     -9.995088042342104
    989.6014024019241
 -23751.539286375046
 240163.20708084106
     -1.260894174835205e6
      3.782775422241211e6
     -6.7250733478393555e6
      6.999690047912598e6
     -3.9373859271850586e6
 923596.7021560669

The exact solution to Hx=b is
10-element Vector{Rational{BigInt}}:
      -10//1
      990//1
   -23760//1
   240240//1
 -1261260//1
  3783780//1
 -6726720//1
  7001280//1
 -3938220//1
   923780//1

The relative error is 0.0002371150814128215669269796422649848396163032457112750865452727412232094875362303

cond(H) = 1.6024868379056498e13
cond(Q) = 1.0000000000000013
cond(R) = 1.602254021475034e13

--------------- N = 11 ---------------

The NxN Hilbert matrix H is
11×11 Hilbert{Rational{Int64}}:
 1//1   1//2   1//3   1//4   1//5   1//6   1//7   1//8   1//9   1//10  1//11
 1//2   1//3   1//4   1//5   1//6   1//7   1//8   1//9   1//10  1//11  1//12
 1//3   1//4   1//5   1//6   1//7   1//8   1//9   1//10  1//11  1//12  1//13
 1//4   1//5   1//6   1//7   1//8   1//9   1//10  1//11  1//12  1//13  1//14
 1//5   1//6   1//7   1//8   1//9   1//10  1//11  1//12  1//13  1//14  1//15
 1//6   1//7   1//8   1//9   1//10  1//11  1//12  1//13  1//14  1//15  1//16
 1//7   1//8   1//9   1//10  1//11  1//12  1//13  1//14  1//15  1//16  1//17
 1//8   1//9   1//10  1//11  1//12  1//13  1//14  1//15  1//16  1//17  1//18
 1//9   1//10  1//11  1//12  1//13  1//14  1//15  1//16  1//17  1//18  1//19
```

```
 1//10  1//11  1//12  1//13  1//14  1//15  1//16  1//17  1//18  1//19  1//20
 1//11  1//12  1//13  1//14  1//15  1//16  1//17  1//18  1//19  1//20  1//21

H=QR factorization where Q=R
11×11 LinearAlgebra.QRCompactWYQ{Float64, Matrix{Float64}}:
 -0.801146    0.55157    0.220109   -0.0712098    0.0197971  -0.00479673  -0.001
01098  0.000182924  -2.76453e-5  -3.3017e-6    -2.70544e-7
 -0.400573   -0.244472  -0.65024     0.519816    -0.271785    0.107872     0.034
2004   -0.00878032    0.00180389   0.000283399   2.97613e-5
 -0.267049   -0.3364    -0.293065   -0.30861      0.58698    -0.468176    -0.246
579     0.0951817    -0.0276033   -0.00585652   -0.000803584
 -0.200287   -0.330339  -0.0384485  -0.431045     0.108753    0.418203     0.552
989    -0.375024      0.166507     0.0501219     0.00928611
 -0.160229   -0.305892   0.110974   -0.315769    -0.259805    0.355243    -0.175
503     0.536421     -0.455077    -0.214841     -0.0568787
 -0.133524   -0.279684   0.196938   -0.150885    -0.369601   -0.0216586   -0.423
031     0.0070383     0.498284     0.485903      0.204767
 -0.114449   -0.255655   0.246004    0.00409302  -0.304556   -0.297945    -0.140
409    -0.419894      0.0724829   -0.520708     -0.455044
 -0.100143   -0.234537   0.273263    0.133875    -0.148497   -0.358998     0.236
397    -0.193428     -0.444865     0.0380331     0.631497
 -0.0890162  -0.216185   0.287334    0.237607     0.0445817  -0.220439     0.387
81      0.286184     -0.0885458    0.481346     -0.532831
 -0.0801146  -0.200242   0.293273    0.318656     0.2442      0.0649919    0.177
899     0.399303      0.503711    -0.441333      0.249972
 -0.0728315  -0.186338   0.294131    0.381138     0.434554    0.443986    -0.405
994    -0.32765      -0.226724     0.127059     -0.0499949
and R=
11×11 Matrix{Float64}:
 -1.24821  -0.734384  -0.536665   -0.427718    -0.357508    -0.308038     -0.271
098     -0.242361     -0.219313     -0.200385     -0.184543
  0.0      -0.160177  -0.180791   -0.177188    -0.167878    -0.157549     -0.147
597     -0.138433     -0.13013      -0.122648     -0.115907
  0.0       0.0        0.0141054   0.0233519    0.0287955    0.0318682     0.033
4857     0.0341994     0.0343444     0.0341289     0.0336852
  0.0       0.0        0.0         0.00102828   0.00222851   0.00330325    0.004
18338    0.00487521    0.00540608    0.00580578    0.0061008
  0.0       0.0        0.0         0.0          6.38188e-5   0.000170168   0.000
293898   0.000418815   0.000536249   0.000642245   0.00073552
  0.0       0.0        0.0         0.0          0.0          3.38762e-6    1.068
95e-5    2.10372e-5    3.32832e-5    4.64341e-5    5.97525e-5
  0.0       0.0        0.0         0.0          0.0          0.0          -1.530
46e-7   -5.56501e-7   -1.22738e-6   -2.13279e-6   -3.21902e-6
  0.0       0.0        0.0         0.0          0.0          0.0           0.0
```

```
        -5.80505e-9   -2.38584e-8   -5.82181e-8   -1.10138e-7
  0.0       0.0        0.0         0.0          0.0          0.0           0.0
         0.0          -1.79962e-10  -8.23853e-10  -2.20157e-9
  0.0       0.0        0.0         0.0          0.0          0.0           0.0
         0.0           0.0           4.31852e-12   2.17699e-11
  0.0       0.0        0.0         0.0          0.0          0.0           0.0
         0.0           0.0           0.0          -6.97894e-14

The solution to Rx=Q'b is
11-element Vector{Float64}:
      10.974787086248398
   -1318.6764515042305
   38574.49068450928
 -480073.0846710205
       3.1506846412353516e6
      -1.209931606640625e7
       2.8569246848632812e7
      -4.1981129884765625e7
       3.739088190625e7
      -1.8465263993164062e7
       3.8778238482666016e6

The exact solution to Hx=b is
11-element Vector{Rational{BigInt}}:
        11//1
     -1320//1
     38610//1
   -480480//1
   3153150//1
 -12108096//1
  28588560//1
 -42007680//1
  37413090//1
 -18475600//1
   3879876//1

The relative error is 0.0006267192156404839336430541467883360386669968902311560855247682859655890370617241

cond(H) = 5.223733117324263e14
cond(Q) = 1.0000000000000002
cond(R) = 5.2271770214303794e14

---------------- N = 12 ----------------
```

```
The NxN Hilbert matrix H is
12×12 Hilbert{Rational{Int64}}:
 1//1   1//2   1//3   1//4   1//5   1//6   1//7   1//8   1//9   1//10  1//11  1/
/12
 1//2   1//3   1//4   1//5   1//6   1//7   1//8   1//9   1//10  1//11  1//12  1/
/13
 1//3   1//4   1//5   1//6   1//7   1//8   1//9   1//10  1//11  1//12  1//13  1/
/14
 1//4   1//5   1//6   1//7   1//8   1//9   1//10  1//11  1//12  1//13  1//14  1/
/15
 1//5   1//6   1//7   1//8   1//9   1//10  1//11  1//12  1//13  1//14  1//15  1/
/16
 1//6   1//7   1//8   1//9   1//10  1//11  1//12  1//13  1//14  1//15  1//16  1/
/17
 1//7   1//8   1//9   1//10  1//11  1//12  1//13  1//14  1//15  1//16  1//17  1/
/18
 1//8   1//9   1//10  1//11  1//12  1//13  1//14  1//15  1//16  1//17  1//18  1/
/19
 1//9   1//10  1//11  1//12  1//13  1//14  1//15  1//16  1//17  1//18  1//19  1/
/20
 1//10  1//11  1//12  1//13  1//14  1//15  1//16  1//17  1//18  1//19  1//20  1/
/21
 1//11  1//12  1//13  1//14  1//15  1//16  1//17  1//18  1//19  1//20  1//21  1/
/22
 1//12  1//13  1//14  1//15  1//16  1//17  1//18  1//19  1//20  1//21  1//22  1/
/23

H=QR factorization where Q=R
12×12 LinearAlgebra.QRCompactWYQ{Float64, Matrix{Float64}}:
 -0.799367    0.552564   0.223115   -0.0736842   0.0211268  -0.00534544  -0.0011
9547   0.000234667  -3.97999e-5  -5.66604e-6    6.39982e-7  -4.99799e-8
 -0.399683   -0.236295  -0.643823    0.524502   -0.282068    0.116583     0.0391
158   -0.0108671     0.00249954   0.000467054  -6.7466e-5    6.59254e-6
 -0.266456   -0.328389  -0.302417   -0.282791    0.576293   -0.481264    -0.2684
19     0.112036     -0.0363289   -0.00915365    0.00172483  -0.000214113
 -0.199842   -0.323178  -0.0561755  -0.420673    0.147518    0.372102     0.5494
1     -0.407921      0.203397     0.0728254    -0.018534     0.00299575
 -0.159873   -0.299549   0.0893336  -0.326847   -0.213214    0.379659    -0.0875
169    0.492141     -0.490959    -0.279971      0.102517    -0.0224558
 -0.133228   -0.274032   0.173686   -0.180681   -0.346946    0.0575096   -0.4033
75     0.133487      0.393596     0.524498     -0.316241     0.100553
 -0.114195   -0.250576   0.222329   -0.0401903  -0.320553   -0.221153    -0.2393
77    -0.33773       0.24984     -0.341577      0.538432    -0.284774
```

```
 -0.0999208  -0.229933   0.249787    0.078914   -0.207102   -0.338324     0.0903
496   -0.321451     -0.298896    -0.269865     -0.405148     0.522846
 -0.0888185  -0.211978   0.26438     0.174994   -0.0550815  -0.293613     0.3229
97     0.0393366    -0.339192     0.363183     -0.139182    -0.620657
 -0.0799367  -0.196372   0.271003    0.250692    0.107459   -0.123918     0.3290
75     0.358495      0.135252     0.242117      0.516628     0.459596
 -0.0726697  -0.182756   0.272615    0.309543    0.265523    0.129639     0.0767
728    0.29611       0.450858    -0.479394     -0.374125    -0.192973
 -0.0666139  -0.170812   0.271057    0.354864    0.411771    0.432567    -0.4125
8     -0.35464      -0.270128     0.176886      0.0939965    0.0350766
and R=
12×12 Matrix{Float64}:
 -1.25099  -0.737877  -0.540231   -0.431209    -0.360877    -0.311272     -0.274
196     -0.245329     -0.222157     -0.203112     -0.187161     -0.173593
  0.0      -0.162577  -0.184203   -0.181082    -0.171999    -0.161758     -0.151
816     -0.142615     -0.134249     -0.126686     -0.119856     -0.113679
  0.0       0.0        0.0146743   0.0244132    0.0302263    0.0335662     0.035
3734     0.0362195     0.0364546     0.0362978     0.0358894     0.0353209
  0.0       0.0        0.0         0.00110949   0.00241828   0.00360191    0.004
58059    0.00535742    0.00595969    0.00641837    0.00676149    0.00701253
  0.0       0.0        0.0         0.0          7.23101e-5   0.000194028   0.000
336931   0.000482414   0.000620258   0.000745609   0.000856722   0.000953535
  0.0       0.0        0.0         0.0          0.0          4.08771e-6    1.298
56e-5    2.57065e-5    4.08818e-5    5.72982e-5    7.40371e-5    9.04618e-5
  0.0       0.0        0.0         0.0          0.0          0.0          -2.000
75e-7   -7.3263e-7    -1.6259e-6    -2.84092e-6   -4.30908e-6   -5.95588e-6
  0.0       0.0        0.0         0.0          0.0          0.0           0.0
        -8.41254e-9   -3.48257e-8   -8.55294e-8   -1.62744e-7   -2.65494e-7
  0.0       0.0        0.0         0.0          0.0          0.0           0.0
         0.0          -2.99218e-10  -1.37991e-9   -3.71198e-9   -7.63535e-9
  0.0       0.0        0.0         0.0          0.0          0.0           0.0
         0.0           0.0           8.75304e-12   4.44541e-11   1.29889e-10
  0.0       0.0        0.0         0.0          0.0          0.0           0.0
         0.0           0.0           0.0           1.99298e-13   1.10439e-12
  0.0       0.0        0.0         0.0          0.0          0.0           0.0
         0.0           0.0           0.0           0.0           2.89253e-15

The solution to Rx=Q'b is
12-element Vector{Float64}:
    -12.86316691711545
   1901.8343653678894
 -66099.44944763184
 983258.5498046875
     -7.8112615341796875e6
```

```
      3.69627893984375e7
     -1.10328209546875e8
      2.12962507875e8
     -2.6517698615625e8
      2.0554468946875e8
     -9.0164200625e7
      1.7091768482421875e7

The exact solution to Hx=b is
12-element Vector{Rational{BigInt}}:
        -12//1
       1716//1
     -60060//1
     900900//1
   -7207200//1
   34306272//1
 -102918816//1
  199536480//1
 -249420600//1
  193993800//1
  -85357272//1
   16224936//1

The relative error is 0.06383982145258038366103931176825124931641043209953978497433393094928931051754233

cond(H) = 1.7341930983785612e16
cond(Q) = 1.0000000000000009
cond(R) = 1.8172309519510468e16
```