Algorithm and software implementation of QR decomposition of rectangular matrices

Todor Stoilov, Krasimira Stoilova

***Abstract**: The QR decomposition in linear algebra calculation is widely applied. Despite that the algorithms for the decomposition are available, it lacks explanations how to implement them for non-square matrices. The research gives descriptions about the modifications, needed to apply for rectangular form of matrices. A software implementation of the algorithm is presented. It can be used for client or server side programming tools, performing linear algebra calculations and optimization.*

***Key words:** linear algebra, decomposition, QR evaluations, optimization.*

INTRODUCTION

The matrix factorization is very useful linear algebra transformations, which targets the presentation of a square matrix **A** in appropriate product of matrices with well supported peculiarities. An important factorization is the QR decomposition of **A** [1]:

(1) $\mathbf{A} = \mathbf{Q.R}$.

Here **R** is an upper triangular, while **Q** is an orthogonal one which satisfies $\mathbf{Q}^T.\ \mathbf{Q} = \mathbf{I}$, where $\mathbf{Q}^\mathbf{T}$ is the transposed of **Q** and **I** is an identity matrix.

Such decomposition can be performed both for square matrix **A**, as for the more general form of rectangular **A** with dimensions MxN. The factorization of a matrix **A** is widely applied technique for solving a linear equation system $\mathbf{A.x} = \mathbf{C}$.

Others kind of factorization techniques are LU, SVD, Cholesky decompositions [1]. To solve the linear system of equation, firstly the component $\mathbf{Q}^{\mathbf{T}}\cdot\mathbf{C} = \mathbf{c_q}$ is evaluated and then the triangular linear system is solved $\mathbf{R.x} = \mathbf{c_q}$ .Due to the upper triangular form of **R**, this system of linear equations is easily solved by back substitution.

The standard algorithm for the QR decomposition involves sequential evaluation of Householder transformations. The Householder matrix is written in the form

(2) $\mathbf{I}\text{-}\mathbf{u}\otimes\mathbf{u}/c$, where $c=0.5\mathbf{u}^\mathbf{T}\mathbf{u}$.

An appropriate Householder matrix, applied to a given matrix can zeros all elements in a column of the matrix, situated below of a given element. For the first column of matrix **A** appropriate matrix $\mathbf{Q_1}$ is evaluated, which put on zero all elements bellow the first in the first column of **A**. Similarly $\mathbf{Q_2}$ zeroes all elements in the second column below the second element and so on up to $\mathbf{Q_{n-1}}$. Hence

(3) $\mathbf{R} = \mathbf{Q_{n-1}.....Q_1.\ A}$.

Since the Householder matrices are orthogonal it follows:

(4) $\mathbf{Q} = (\mathbf{Q_{n-1}......Q_1})^{-1} = \mathbf{Q_1....Q_{n-1}}$.

The general QR decomposition for rectangular matrix **A** performs appropriate pivoting in the choice of leading column of **A** for the factorization. Such a pivoting leads to the more general form of decomposition

(5) $\mathbf{A} = \mathbf{Q.R.E^T}$,

where **E** is a matrix, defining the sequence of utilization of the columns of **A**.

QR FACTORISATION

The of QR factorisation keeps the equality in (1), where **R** is upper (also known as right) triangular, **Q** is orthogonal (so that $\mathbf{Q}^{-1} = \mathbf{Q}^\mathbf{T}$) and both are real. The QR factorization of a matrix requires that it must be reduced to zero certain elements of a vector (a column or portion of a column of the matrix, whose factorization is sought). The transformations are often described in terms of multiplication by a matrix with appropriate special properties, although in practice the equivalent operations are usually carried out more efficiently without explicitly forming the matrix or performing the matrix multiplication.

HOUSEHOLDER MATRIX

The Householder transformation is often described in terms of multiplication by a matrix known as Householder matrix. A Householder matrix has the form $\mathbf{H} = \mathbf{I} - 2\mathbf{ww}^T$, where **w** is a column vector, $\|w\|_2 = 1$.The formation of the Householder matrix to reduce to zero a vector **x** from position k to position n is summarized in the following algorithm:

Given an n-dimensional vector **x** and an index k such that 1≤k≤n, find a vector **w** so that the matrix $\mathbf{H} = \mathbf{I} - 2\mathbf{ww}^T$ reduces positions k+1, …, n of vector **x** to zero, so the vector **Hx** has the form $[z_1,\ldots,z_k, 0,0,\ldots0]^T$.

Step 1: Set $w_i=0$ for i=1,…,k-1.

Step2: Find $g = \sqrt{x_k^2 + \ldots + x_n^2}$

Find $s = \sqrt{2g(g + |x_k|)}$.

Step3: Set $w_k = (x_k + \mathrm{sign}(x_k)g)/s$

Set $w_i = x_i/s$ for i=k+1,…,n.

In applications, Householder matrices are not explicitly constructed. To avoid this construction, the relation is workout as

$$\mathbf{HA} = (\mathbf{I} - 2\mathbf{ww}^T)\mathbf{A} = \mathbf{A} - 2\mathbf{ww}^T\mathbf{A} = \mathbf{A} - \mathbf{wu}^T,$$

where $\mathbf{u} = 2\mathbf{A}^T\mathbf{w}$. Computation of **u** requires the multiplication of a matrix and a vector and $\mathbf{A} - \mathbf{wu}^T$ requires the subtraction of one matrix from another. Altogether require less computational efforts in order explicitly to form the matrix $\mathbf{H} = \mathbf{I} - 2\mathbf{ww}^T$ and multiplication of the matrices **H** and **A.** For that reason, Householder matrices are virtually never explicitly formed in practice. For the case, when the matrix **A** is not in full rank or it is a rectangular one, the more general form of the QR factorization is applied as

(6) **QR** = **AE** , where **E** is the permutation matrix.

For these cases the algorithm must be modified, applying appropriate rearrangement of the columns of **A**.

QRE DECOMPOSITION OF RECTANGULAR MATRIX

To be evident, the algorithm is explained by an example. The initial matrix **A** is defined as

A = [
8.4288 -8.4288 8.4288 4.5598 0 -1.9400 0.1272 -2.0561 -2.8924 -1.6674 1.9400 -0.1272 2.0561 2.8924 1.6674
-8.4288 8.4288 -8.4288 -4.5598 0 1.9400 -0.1272 2.0561 2.8924 1.6674 -1.9400 0.1272 -2.056 -2.8924 -1.6674
8.4288 -8.4288 8.4288 4.5598 0 -1.9400 0.1272 -2.0561 -2.8924 -1.6674 1.9400 -0.1272 2.0561 2.8924 1.6674
4.5598 -4.5598 4.5598 4.2303 0 -0.4521 -0.0501 0.1726 -2.6390 -1.5913 0.4521 0.0501 -0.1726 2.6390 1.5913
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
-1.9400 1.9400 -1.9400 -0.4521 0 1.3567 0.0507 0.0805 0.3484 0.1037 -1.3567 -0.0507-0.0805 -0.3484 -0.1037
0.1272 -0.1272 0.1272 -0.050 0 0.0507 0.5418 -0.7698 0.0449 0.0051 -0.0507 -0.5418 0.7698 -0.0449 -0.0051
-2.0561 2.0561 -2.0561 0.1726 0 0.0805 -0.7698 2.9180 -0.1399 -0.0327 -0.0805 0.7698 -2.9180 0.13990 .0.0327
-2.8924 2.8924 -2.8924 -2.6390 0 0.3484 0.0449 -0.1399 1.8926 0.7463 -0.3484 -0.0449 0.1399 -1.8926 -0.7463
-1.6674 1.6674 -1.6674 -1.5913 0 0.1037 0.0051 -0.0327 0.7463 0.8450 -0.1037 -0.0051 0.0327 -0.7463 -0.8450
1.9400 -1.9400 1.9400 0.4521 0 -1.3567 -0.0507 -0.0805 -0.3484 -0.1037 1.3567 0.0507 0.0805 0.3484 0.1037
-0.1272 0.1272-0.1272 0.0501 0 -0.0507 -0.5418 0.7698 -0.0449 -0.0051 0.0507 0.5418 -0.7698 0.0449 0.0051
2.0561 -2.0561 2.0561 -0.1726 0 -0.0805 0.7698 -2.9180 0.1399 0.0327 0.0805 -0.7698 2.9180 -0.1399 -0.0327
2.8924 -2.8924 2.8924 2.6390 0 -0.3484 -0.0449 0.1399 -1.8926 -0.7463 0.3484 0.0449 -0.1399 1.8926 0.7463
1.6674 -1.6674 1.6674 1.5913 0 -0.1037 -0.0051 0.0327 -0.7463 -0.8450 0.1037 0.0051 -0.0327 0.7463 0.8450]

The dimensions of this matrix are NxN, where N=15. The computational sequence for the evaluation of matrices **Q** and **R** is:

Initial replacement of $\mathbf{R}^{(0)} = \mathbf{A}$.

- For k = 1, … , N – 1
  - the Householder matrix $\mathbf{H}^{(\kappa)}$ is evaluated, which turn to zero components k+1, … , N in k-th column of $\mathbf{R}^{(k-1)}$;
  - the evaluation of the matrix $\mathbf{R}^{(k)}$, according to the product $\mathbf{R}^{(k)} = \mathbf{H}^{(k)}.\mathbf{R}^{(k-1)}$ is performed.

  End.
- The matrix **Q** starts with initial value **Q** = **I**, where **I** is a NxN identity matrix.
- For k = N – 1, …. ,1 $\mathbf{Q} = \mathbf{H}^{(k)}.\mathbf{Q}$; End.
- The final matrix is $\mathbf{R} = \mathbf{R}^{(N-1)}$.

The evaluation of matrix **Q** is performed by sequential multiplication of the Househoder matrices, obtained on the sequential evaluations $\mathbf{Q} = \mathbf{H}^{(N-1)}.\mathbf{H}^{(N+2)} \dots \mathbf{H}^{(1)}$. But for speeding the calculations, multiplication of **Q** and the current Householder matrix is performed and the result is stored in **Q**. Thus the matrices $\mathbf{H}^{(i)}$, **i**=1,N-1 are not kept in explicit numerical way, which speed up the calculations.

This algorithm of QR factorization is complicated for the case of the general rectangular form of matrix $\mathbf{A}_{NxM}$, N≠M. To have a numerical stability in the evaluations, an option is applied, concerning a QR factorization, which rearranges the columns of **R** under the decreasing value of the leading element W_k. This kind of decomposition is noted as QRE one and it satisfies the equality (6) where **E** is an identity matrix with interchanged columns, according to the rearrangement of the columns of **A** (respectively **R**), $\mathbf{E}.\mathbf{E}^T = \mathbf{I}$. The choice of the leading column, used for the evaluation of the current Householder matrix (respectively the arrangement of the columns of **E**), is performed according to the value of the scalar s_i i=1,…N, which is calculated for each column of the matrix **A.** The leading column is defined according to the maximal value of **s**. Here the rearrangement of the components of **s** is performed with the function InsertionSort().

The QRE decomposition is performed in following sequence:

- The initial matrix **A** is stored in the working matrix **a**.
- Calculation of the square sum of the columns of matrix **a.** The results are stored in the vector $s_{i|15x1}$

```
for j=1:N,
  lnum(j)=j;
  for i=1:M,
    if abs(a(i,j))>eps    s(j) = s(j) + a(i,j)*a(i,j) ;
    end, end,end
```

For the current matrix **A**, the numerical value of the vector **s** is:

```
s =[
 column 1 till 7
 272.2325  272.2325  272.2325   99.7384        0   15.4593    1.8326
 Column  8 till 15
  30.9811   40.6271   13.4387   15.4593    1.8326   30.9811   40.6271  13.4387 ]
```

- The values of the components s_i are put in decreasing order. Such an arrangement is performed by the functions insertion_Sort() and insertz(). Their peculiarities concern the recursive invocation of each other.

```
function [x,lnum]=insertionSort(n,x,lnum)
%insertionSort( int n, float x[])
% sort x[1],<...<,x[n]
if (n>1)
  [x,lnum]=insertionSort(n-1,x,lnum);
  [x,lnum]=insertz(x,n,x(n),lnum,lnum(n));
end
return

function [x,lnum]=insertz(x,i,a,lnum,c)
% inser a into x[1],...,x[i]
% x[1],...x[i-1] are sorted
%x[i] is unoccupied
if (i==1 | x(i-1)>=a)
```

```
  x(i)=a;
  lnum(i)=c;
else
  x(i)=x(i-1);
  lnum(i)=lnum(i-1);
  [x,lnum]=insertz(x,i-1,a,lnum,c);
end
return
```

Finaly the components of the vector **s** are put in decreasing order as follows:

```
s =[
  column 1 till 7
  272.2325  272.2325  272.2325   99.7384   40.6271   40.6271   30.9811
  column 8 till 15
  30.9811   15.4593   15.4593   13.4387   13.4387    1.8326    1.8326   0 ]
lnum = [1     2     3     4     9    14     8    13     6    11    10    15 7    12     5]
```

The results of the rearrangements are stored again in the vector **s**. The correspondence between the initial column sequence of **A** and the current rearrangement is given in the vector lnum. According to the example, lnum (5) = 9 means that the 5-th column of the initial matrix **a**, will be QR evaluated and stored in column 9 of the final matrix **a:**

lnum (5) = 9 $\Rightarrow a_{i,5} = a_{i,9}$, **i** = 1,15.

- Evaluation of the components of the matrices Q and R

The complete logical rules of the general QR decomposition, applied for general rectangular matrices are the following:

```
R = a; Q = I
For   k = 1, N-1
        x = 0,
        x (k:N,1) = R(k:N,k) ;
% vector x consists the components from position k till position  n of  κ-th column of matrix R.
        s = norm(x); v = x; v(k) = x(k) + s
        g = norm(v); w = v/g; u = 2.R^T.W
        R = R – W.u^T; Q = Q – 2.Q.W.W^T
End.
```

This logical sequence is complicated, rendering that matrix **a** is a rectangular one, ($a_{|MxN}$), and M could differ from N. Additionally logical checks perform the tracing for the choice of leading columns from **R** and the corresponding rearrangement, done in matrix **E**.

The code of this QRE decomposition follows:

```
%the matrix a is prepared accordingly and the first column must be the leading one
for k=1:MNmin,
  %evaluation of the leading column
  if (k ~= 1)
    % recalculation of  s
    for m=k:N,
      if abs(a(k-1,m))>eps   s(m)=s(m) - a(k-1,m)*a(k-1,m);
     end,  end
    s1=s(k);
    im=k;
    for j=k+1:N
      if s(j)>s1
        s1=s(j);
        im=j;
      end,  end
    if (s1>= TOL(1))
      if im~=k
        b(1)=s(k);
        s(k)=s(im);
        s(im)=b(1);
        %interchange of the k-th column with  imax+k one in aij
```

```
          for i=1:M
             b(1)=a(i,k);
             a(i,k)=a(i,im);
             a(i,im)=b(1);
          end
          for i= 1:N
             b(1)=e(i,k);
             e(i,k)=e(i,im);
             e(i,im)=b(1);
          end, end, end, end
   %the matrix a(i,j) is reordered and it is ready to perform first Householder transformation
   scale=max(abs(a(k:M,k)));
   if (scale <=  eps)
      asign=1;
      c(k)=0.0;
      d(k)=c(k);
   else
      for i=k:M,
         if abs(a(i,k))>eps   a(i,k)=a(i,k)/scale;
         else   a(i,k)=0.;
         end, end
      asum=0.0;
      for  i=k:M,
         if abs(a(i,k))>eps   asum = asum + a(i,k)*a(i,k);
         end,  end %сбор по колона
      if a(k,k) ~=0   sigma=sign(a(k,k))*sqrt(asum);
      else  sigma=sqrt(asum);
      end
      a(k,k)=a(k,k)+sigma;
      if abs(a(k,k))>eps   c(k)=sigma*a(k,k);
      else   c(k)=0;
      end
      d(k)=-scale*sigma;
      for j=k+1:N,
         asum=0.0;
         for i=k:M,
            if abs(a(i,j))>eps
               if abs(a(i,k))>eps   asum=asum+a(i,k)*a(i,j);
               end, end, end
         if abs(asum)>eps   tau=asum/c(k);
            for i=k:M,
               if abs(a(i,k))>eps   a(i,j)=a(i,j)-tau*a(i,k);
               end, end, end, end, end,end
% evaluation of matrix R
%r=zeros(M,N);
rankR=0;
for i=1:MNmin,
   %evaluation of the rank of  R, which correspond to the rank of the initial matrix A
   if (abs(d(i)) > TOL(1)),
       r(i,i)=d(i);
      rankR=rankR+1;
   end,end
for i=1:rankR
   for j=i+1:N,  r(i,j)=a(i,j); end
 end
d(N)=a(N,N);
if d(N)==0., asign=1;end
%end of the QRE decomposition
```

Following the example, matrix **R** has an upper triangular form and rank(**R**)=5. The matrix **e** consists the final allocation of the columns of the initial matrix A:

R=[
  column 1 till 7

```
-16.4995   3.8057  -9.4667   6.0229   3.5796  -6.0229  16.4995
      0   -4.0617  -2.5818   1.7097   0.3024  -1.7097   0
      0    0        1.8586  -1.0933   1.5299   1.0933   0
      0    0        0       -0.4827  -0.0507   0.4827   0
      0    0        0        0       -0.4594   0        0

Column  8 till 15
-3.8057 -16.4995  -3.5796   3.4438  -3.4438  -0.3526   0.3526  0
 4.0617   0       -0.3024   0.8721  -0.8721   1.1775  -1.1775  0
-0.0000   0       -1.5299  -0.7652   0.7652   0.3287  -0.3287  0
 0.0000   0        0.0507   0.4827  -0.4827   0.0507  -0.0507  0
    0     0        0.4594   0        0        0.4594  -0.4594  0  ]

e =[
 column 1 till 12
   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0
   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1
   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0
   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0
   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0
   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0
   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   ]
```

CONCLUSIONS

The QR factorisation of rectangular and/or not full rank matrices is difficult to perform without appropriate rearrangement of the columns of the initial matrix A. Additionally logical rules must be applied for the identification of the appropriate leading columns, which define the sequence of the Householder transformation. The code, presented in the paper can be implemented under different programming languages. It is a benefit for users, which can include it in complex evaluation algorithms. This overcome the lack of code for general rectangular QRE decomposition in the Matlab software suite.

REFERENCES

[1] W.Press, S. Teukolsky. W.Vetterling, B.Flannery. Numerical receipts in C. Second edition. Cambridge University press, 1997. p.965.

[2] Fausett L. Applied numerical analysis. Prentice Hall, Upper Saddle River, NY, 07458, 1999, p.596.

ABOUT THE AUTHOR

Prof. Todor Stoilov, D.Sc., PhD, Institute of Computer and Communication Systems, Bulgarian Academy of Sciences, Phone: +359 2 73 78 20, E-mail: todor@hsi.iccs.bas.bg

Assoc. Prof. Krasimira Stoilova, PhD, Institute of Computer and Communication Systems, Bulgarian Academy of Sciences, Phone: +359 2 979 27 74, E-mail: k.stoilova@hsi.iccs.bas.bg