Tut III of MA2213

Qian Lilong

Department of Mathematics
National University of Singapore

qian.lilong@u.nus.edu

October 8, 2018

1 Solving Linear system

1.1 Question 1

A matrix $A = (a_{ij})_{n\times n}$ is called a lower triangular matrix if $a_{ij} = 0$ for all $i < j$. Prove the following statements.

(a) The product of two n × n lower triangular matrices is lower triangular.

(b) The product of two n × n upper triangular matrices is upper triangular.

(c) The inverse of a nonsingular n × n lower (respectively upper) triangular matrix is lower (respectively upper) triangular. (Hint: Consider solving for columns of the matrix B in AB = I by applying the "forward substitution" procedure (cf. Question 1).)

Proof. (a) We can solve this problem by mathematical induction.

First, when $n = 1$, this is true.

Next, assume $n = k$ holds. We consider $n = k + 1$. Suppose

$$A = \begin{pmatrix} a_{11} & 0 \\ h & A_{22} \end{pmatrix}, B = \begin{pmatrix} b_{11} & 0 \\ f & B_{22} \end{pmatrix}, \tag{1}$$

where A_{22}, B_{22} are two lower triangular matrices of dimension $(k-1)\otimes(k-1)$ and h, f are two $(k-1)$-dimension column vectors. Then

$$AB = \begin{pmatrix} a_{11}b_{11} & 0 \\ b_{11}h + A_{22}f & A_{22}B_{22} \end{pmatrix}. \tag{2}$$

Since A_{22} and B_{22} are both (k-1)-dimension lower triangular matrix, by the assumption, their product remains a lower triangular matrix.

Since both the base case and the inductive step have been performed, the conclusion holds for any natural n.

(b) Suppose A, B are two upper triangular matrices. Then $A^\top, B^\top$ are two lower triangular matrices. By the conclusion of (a), we have

$$AB = (B^\top A^\top)^\top \tag{3}$$

is an upper triangular matrix.

(c) We can also solve this by mathematical induction.

First, consider the case when $n = 1$. The inverse of a scalar is also a scalar. The conclusion holds.

Next, assume conclusion holds for k. We consider the case k+1.

Suppose A is a upper triangular with the block representation

$$A = \begin{pmatrix} a_{11} & f \\ 0 & A_{22} \end{pmatrix}, \tag{4}$$

where A_{22} is an upper triangular matrix of dimension $(k-1)\times(k-1)$, f is a (k-1)-dimension row vector.

Suppose $B = A^{-1}$. Write B in the block representation accordingly:

$$B = \begin{pmatrix} b_{11} & h \\ g & B_{22} \end{pmatrix}, \tag{5}$$

where $g, f^\top$ are two (k-1)-dimension column vector and B_{22} is a $(k-1) \times (k-1)$ matrix.

Since $AB = I$, we have

$$\begin{aligned} a_{11}b_{11} &= 1, \\ a_{11}h + fB_{22} &= 0, \\ A_{22}g &= 0, \\ A_{22}B_{22} &= I_{k-1}. \end{aligned} \tag{6}$$

Since A is invertible, we have $g = 0$. And by the last in equality in eq. (6) and the assumption, B_{22} is an upper triangular matrix. Therefore, B is upper triangular.

By the mathematical induction, the proof completes.

■

1.2 Question 2

Prove that

(a) The LU decomposition of an invertible matrix A, i.e., A = LU where L has 1's on its diagonal, is unique.

(b) If A is nonsingular and symmetric ($A = A^\top$) and A = LU where L has 1's on its diagonal, then U = DLT, with D the diagonal matrix having the same diagonal entires as U.

Proof. (a) Suppose there are two LU decomposition

$$A = L_1U_1 = L_2U_2, \tag{7}$$

where L_1, L_2 are two lower triangular matrices with diagonal entries being 1. U_1, U_2 are two upper triangular matrices.

Since A is invertible, L_i, U_i are all invertible. We have

$$L_2^{-1}L_1 = U_2U_1^{-1}. \tag{8}$$

Note that L_2^{-1} is lower triangular and U_1^{-1} is upper triangular by the results in question 1. We have $L_2^{-1}L_1$ is lower triangular and $U_2U_1^{-1}$ is upper triangular. Hence $L_2^{-1}L_1$ is both lower and upper triangular, which must be diagonal.

Compute the diagonal entries of $L_2^{-1}L_1$, which is an identity matrix. Therefore, $L_2^{-1}L_1 = I$, i.e., $L_1 = L_2$.

(b) Suppose $A = LU$, where L is lower triangular with diagonal being 1 and U is upper triangular. By the symmetry of A, we have

$$LU = U^\top L^\top = U_1^\top D^{-\top} L^\top, \tag{9}$$

where

$$U = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{pmatrix}, D = \begin{pmatrix} u_{11} & 0 & \cdots & 0 \\ 0 & u_{22} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{pmatrix}, U_1 = D^{-1}U. \tag{10}$$

Note that A is invertible, hence $U_{ii} \neq 0$. Forward, D is invertible and diagonal.

Now $U_1^\top$ is a lower diagonal matrix with diagonal entries being 1, by the uniqueness of LU decomposition in (a), we have

$$U = D^\top L. \tag{11}$$

■

1.3 Question 3

Solve the linear equation $Ax = b$ in four-digit rounding floating point arithmetic, where

$$A = \begin{pmatrix} 0.00211 & 0.08204 \\ 0.337 & 12.84 \end{pmatrix}, b = \begin{pmatrix} 0.04313 \\ 6.757 \end{pmatrix} \tag{12}$$

Proof. (a) Gaussian elimination; m_{21} is 159.7. The new A is:

$$\begin{pmatrix} 0.00211 & 0.08204 \\ 0 & -0.26 \end{pmatrix} \tag{13}$$

The new b is:

$$\begin{pmatrix} 0.04313 \\ -0.131 \end{pmatrix} \tag{14}$$

By back forward substitution :

$$x = \begin{pmatrix} 0.8531 \\ 0.5038 \end{pmatrix} \tag{15}$$

(b) Gaussian elimination with partial pivoting

Change 1-row and 2-th row. New A after row exchange

$$\begin{pmatrix} 0.337 & 12.84 \\ 0.00211 & 0.08204 \end{pmatrix}$$

New b after row exchange

$$\begin{pmatrix} 6.757 \\ 0.04313 \end{pmatrix}$$

m(21) is 0.006261 The new A is:

$$\begin{pmatrix} 0.337 & 12.84 \\ 0 & 0.00165 \end{pmatrix} \tag{16}$$

The new b is:

$$\begin{pmatrix} 6.757 \\ 0.00082 \end{pmatrix}$$

By the backward substitution,

$$x = \begin{pmatrix} 1.116 0.497 \end{pmatrix}$$

(c) Gaussian elimination with scaled partial pivoting

Gaussian Elimination with scaled partial pivoting Since

$$0.002110/0.08204 < 0.3370/12.84; \tag{17}$$

so, row exchange is needed. The result is same as that in part (b)

Multiply the first equation by 1000 and do (b) and (c) for the resulting system

(b') The result is same as (b)

(c') The result is same as (c);

■

1.4 Question 4

Use Gaussian elimination with partial pivoting and three-digit chopping arithmetic to solve the following linear system and compare the approximation to the actual solution:

$$Ax = b \tag{18}$$

, where

$$A = \begin{pmatrix} 3.03 & -12.1 & 14 \\ -3.03 & 12.1 & -7 \\ 6.11 & -14.2 & 21 \end{pmatrix}, b = \begin{pmatrix} -119 \\ 120 \\ -139 \end{pmatrix}. \tag{19}$$

Proof. The exact solution can be solved:

$$x = \begin{pmatrix} 0 \\ 10 \\ \frac{1}{7} \end{pmatrix}. \tag{20}$$

The results for computing by GS partial pivoting are:

```
    change 1-row and 3-th row
New A after row exchange
         6.11        -14.2           21
        -3.03         12.1           -7
         3.03        -12.1           14

```

```
New b after row exchange
  -139
   120
  -119

m(21) is -0.495
m(31) is 0.495
The new A is:
         6.11        -14.2           21
            0       -0.581       -0.597
            0        0.581         0.37

The new b is:          -139
       -0.568
        0.569

  change 2-row and 3-th row
New A after row exchange
         6.11        -14.2           21
            0        0.581         0.37
            0       -0.581       -0.597

New b after row exchange
         -139
        0.569
       -0.568

m(32) is -1
The new A is:
         6.11        -14.2           21
            0        0.581         0.37
            0            0       -0.226

The new b is:          -139
        0.569
          0.1

x(3) is -0.442 new b is -0.138
new b is 0.732
new b is 0.1
x(2) is 0.125 new b is 0.389
new b is 0.732
new b is 0.1
x(1) is 0.636 new b is 0.389
new b is 0.732
new b is 0.1

x =

        0.636
        0.125
       -0.442

```

■

1.5 Question 5

Solve the following linear system of equations by Gaussian elimination with partial pivoting with four-digit arithmetic with rounding

$$Ax = b \tag{21}$$

, where

$$A = \begin{pmatrix} 0.003 & 59.14 & 0 & 0 & 0 \\ 5.291 & -6.13 & 0 & 0 & 0 \\ 0 & 0 & 0.6 & 0 & 0 \\ 0 & 0 & 0 & 0.00211 & 0.08204 \\ 0 & 0 & 0 & 0.337 & 12.84 \end{pmatrix}, b = \begin{pmatrix} 59.17 \\ 46.78 \\ 1 \\ 0.04313 \\ 6.757 \end{pmatrix} \tag{22}$$

The results for solving this linear equation are:

```
   change 1-row and 2-th row
New A after row exchange
        5.291        -6.13            0            0            0
        0.003        59.14            0            0            0
            0            0          0.6            0            0
            0            0            0      0.00211      0.08204
            0            0            0        0.337        12.84

New b after row exchange
        46.78
        59.17
            1
      0.04313
        6.757

m(21) is 0.000567
m(31) is 0
m(41) is 0
m(51) is 0
The new A is:
        5.291        -6.13            0            0            0
            0        59.14            0            0            0
            0            0          0.6            0            0
            0            0            0      0.00211      0.08204
            0            0            0        0.337        12.84

The new b is:        46.78
        59.14
            1
      0.04313
        6.757

  change 2-row and 2-th row
New A after row exchange
        5.291        -6.13            0            0            0
            0        59.14            0            0            0
            0            0          0.6            0            0
            0            0            0      0.00211      0.08204
            0            0            0        0.337        12.84

New b after row exchange
        46.78
        59.14
            1
      0.04313
        6.757

m(32) is 0
m(42) is 0
m(52) is 0
The new A is:
        5.291        -6.13            0            0            0
            0        59.14            0            0            0
            0            0          0.6            0            0
            0            0            0      0.00211      0.08204
```

```
             0             0             0         0.337         12.84

The new b is:        46.78
        59.14
            1
      0.04313
        6.757

  change 3-row and 3-th row
New A after row exchange
        5.291         -6.13             0             0             0
            0         59.14             0             0             0
            0             0           0.6             0             0
            0             0             0       0.00211       0.08204
            0             0             0         0.337         12.84

New b after row exchange
        46.78
        59.14
            1
      0.04313
        6.757

m(43) is 0
m(53) is 0
The new A is:
        5.291         -6.13             0             0             0
            0         59.14             0             0             0
            0             0           0.6             0             0
            0             0             0       0.00211       0.08204
            0             0             0         0.337         12.84

The new b is:        46.78
        59.14
            1
      0.04313
        6.757

  change 4-row and 5-th row
New A after row exchange
        5.291         -6.13             0             0             0
            0         59.14             0             0             0
            0             0           0.6             0             0
            0             0             0         0.337         12.84
            0             0             0       0.00211       0.08204

New b after row exchange
        46.78
        59.14
            1
        6.757
      0.04313

m(54) is 0.006261
The new A is:
        5.291         -6.13             0             0             0
            0         59.14             0             0             0
            0             0           0.6             0             0
            0             0             0         0.337         12.84
            0             0             0             0       0.00165

The new b is:        46.78
        59.14
            1
        6.757
      0.00082

x(5) is 0.497
x(4) is 1.116
```

```
x(3) is 1.667
x(2) is 1
x(1) is 10

x =

          10
           1
       1.667
       1.116
       0.497


```

1.6 Question 6

Proof. After one step Gauss elimination with partial pivoting we have

$$\left[\begin{array}{ccc|c} 6 & \alpha & 10 & 5 \\ 2 & 1 & 3 & 1 \\ 4 & 6 & 8 & 5 \end{array}\right] \Rightarrow \left[\begin{array}{ccc|c} 6 & \alpha & 10 & 5 \\ 0 & 1-\frac{\alpha}{3} & -\frac{1}{3} & -\frac{2}{3} \\ 0 & 6-\frac{2}{3}\alpha & \frac{4}{3} & \frac{5}{3} \end{array}\right].$$

- If $\alpha = 6$ The augmented matrix is

  $$\left[\begin{array}{ccc|c} 6 & \alpha & 10 & 5 \\ 0 & -1 & -\frac{1}{3} & -\frac{2}{3} \\ 0 & 2 & \frac{4}{3} & \frac{5}{3} \end{array}\right].$$

  Since $|-1| < 2$, we need to exchange the second and last row in the next step.

- If $\alpha = 9$

  The augmented matrix is

  $$\left[\begin{array}{ccc|c} 6 & \alpha & 10 & 5 \\ 0 & -2 & -\frac{1}{3} & -\frac{2}{3} \\ 0 & 0 & \frac{4}{3} & \frac{5}{3} \end{array}\right].$$

  Since $|-2| > 0$, we need not to exchange the second and last row in the next step.

- If $\alpha = -3$

  The augmented matrix is

  $$\left[\begin{array}{ccc|c} 6 & \alpha & 10 & 5 \\ 0 & 2 & -\frac{1}{3} & -\frac{2}{3} \\ 0 & 8 & \frac{4}{3} & \frac{5}{3} \end{array}\right].$$

  Since $2 < 8$, we need to exchange the second and last row in the next step.

it is clear that only for $\alpha = 9$ there will be no row interchange required when solving this system using Gaussian Elimination with partial pivoting. ■

2 The MATLAB code for performing the Gauss elimination

2.1 Gauss elimination with rounding or chopping arithmetic

The source file is

```
        function x      =  GS_round(A,b,digit,type)
%%  function to perform the Gauss elimination with rounding or chopping
    arithmatics
%   input: Ax = b
%          digit, how many digits to keep when using rounding or chopping
%          type, be 'round' or 'chop', which is a string
if ~exist('digit','var')
    digit       = 4;
end
if ~exist('type','var')
    type        = 'round';    % using rounding arithmatic
end
n               = length(A);  % dimension of A
global d t
d               = digit;
t               = type;
%% perform GS elimination
for i           = 1:n-1
   for j        = i+1:n
      mji       = tdiv(A(j,i),A(i,i));
      fprintf('m(%d%d) is %g\n',j,i,mji);
                              % -mji row i + row j

      for k     = i+1:n
         A(j,k) = tsub(A(j,k),   tmulti(mji,A(i,k)));

      end
      b(j)      = tsub(b(j),tmulti(mji,b(i)));
   end
   A(i+1:n,i)    =0;
   fprintf('The new A is:\n');
   disp(A);
   fprintf('The new b is:');
   disp(b);
end
%% call backward substitution
x               = backsub(A,b);
end


function y      = tadd(x,y)   %
x               = tround(x);
y               = tround(y);
y               = tround(x+y);
end

function y      = tmulti(x,y)
x               = tround(x);
y               = tround(y);
y               = tround(x*y);
end
function y      = tdiv(x,y)
x               = tround(x);
y               = tround(y);
y               = tround(x/y);
end

function y      = tsub(x,y)
x               = tround(x);
```

```
y                = tround(y);
y                = tround(x-y);
end

function y       = tround(x)
global d t
switch t
    case 'round'
        y        = round(x,d,'significant');
    otherwise
        ex       = ceil(log10(abs(x)));
        x        = x/(10^ex);
        y        = fix(x*(10^d))/(10^d);
        x        = x*(10^ex);
end

end

function x       = backsub(A,b)
%% A is a upper triangular matrix
n                = length(A);
x                = zeros(n,1);
for i            =n:-1:1
    x(i)         = tround(b(i)/A(i,i));
    fprintf('x(%d) is %g ',i,x(i));
    for j        = 1:i-1
       b(j)      = tround(b(j) -  tmulti(A(j,i),x(i)));
    end
    fprintf('new b is %g\n',b);

end
end
```

2.2 Gauss elimination with partial pivoting

The source file is

```
  function x =  GS_partial(A,b,digit,type)
%%  function to perform the Gauss elimination with partial
%        pivoting  and rounding or chopping arithmatics
%   input: Ax = b
%          digit, how many digits to keep when using rounding or chopping
%          type, be 'round' or 'chop', which is a string
if ~exist('digit','var') % the default value for digit
    digit = 4;
end
if ~exist('type','var') % by default, using rouding arithmatic
    type = 'round';          % using rounding arithmatic
end
n = length(A);               % dimension of A
global d t
d = digit;
t = type;
%% perform GS elimination
for i = 1:n-1
    %% partial pivoting
                             % largest value of A(i,j);
    v            = A(i,i:end);
    v            = abs(v);
    [~,idx]      = max(v);
    idx          = idx + i-1;
    fprintf('  change %d-row and %d-th row\n',i,idx);
    v            = A(i,:);
    A(i,:)       = A(idx,:);
    A(idx,:)     = v;
```

```
    fprintf('New A after row exchange\n');
    disp(A);
                                % exchange b also
    v            = b(i);
    b(i)         = b(idx);
    b(idx)       = v;
    fprintf('New b after row exchange\n');
    disp(b);
   for j         = i+1:n
      mji        = tdiv(A(j,i),A(i,i));
      fprintf('m(%d%d) is %g\n',j,i,mji);
                                % -mji row i + row j

      for k      = i+1:n
         A(j,k) = tsub(A(j,k),   tmulti(mji,A(i,k)));

      end
      b(j)       = tsub(b(j),tmulti(mji,b(i)));
   end
   A(i+1:n,i)   =0;
   fprintf('The new A is:\n');
   disp(A);
   fprintf('The new b is:');
   disp(b);
end
%% call backward substitution
x                = backsub(A,b);
end


function y       = tadd(x,y)  %
x                = tround(x);
y                = tround(y);
y                = tround(x+y);
end

function y       = tmulti(x,y)
x                = tround(x);
y                = tround(y);
y                = tround(x*y);
end
function y       = tdiv(x,y)
x                = tround(x);
y                = tround(y);
y                = tround(x/y);
end

function y       = tsub(x,y)
x                = tround(x);
y                = tround(y);
y                = tround(x-y);
end

function y       = tround(x)
global d t
switch t
    case 'round'
        y        = round(x,d,'significant');
    otherwise
        ex       = ceil(log10(abs(x)));
        x        = x/(10^ex);
        y        = fix(x*(10^d))/(10^d);
        x        = x*(10^ex);
end

end

function x       = backsub(A,b)
%% A is a upper triangular matrix
```

```
n                = length(A);
x                = zeros(n,1);
for i            =n:-1:1
    x(i)         = tround(b(i)/A(i,i));
    fprintf('x(%d) is %g ',i,x(i));
    for j        = 1:i-1
       b(j)      = tround(b(j) -  tmulti(A(j,i),x(i)));
    end
    fprintf('new b is %g\n',b);

end
end
```

2.3 Gauss elimination with scaled partial pivoting

```
  function x       =  GS_scaled(A,b,digit,type)
    %%  function to perform the Gauss elimination with scaled paritial
%         pivoting and rounding or chopping arithmatics
%   input: Ax     = b
%           digit, how many digits to keep when using rounding or chopping
%           type, be 'round' or 'chop', which is a string
if ~exist('digit','var')
    digit         = 4;
end
if ~exist('type','var')
    type          = 'round';    % using rounding arithmatic
end
n                 = length(A);  % dimension of A
global d t
d                 = digit;
t                 = type;
%% perform GS elimination
for i             = 1:n-1
    %% partial pivoting
                               % largest value of A(i,j);
                               % largest value of each row
    v             = zeros(n-i+1,1);
    for j         =i:n
       v(j-i+1)   = max(abs(A(j,i:n)));
    end
    for j         =i:n
        v(j-i+1)  = abs(A(j,i))/v(j-i+1);
    end
    fprintf('The relative pivoting is:\n');
    disp(v');
    [~,idx]       = max(v);
    idx           = idx + i-1;

    fprintf('  change %d-row and %d-th row\n',i,idx);
    v             = A(i,:);
    A(i,:)        = A(idx,:);
    A(idx,:)      = v;
    fprintf('New A after row exchange\n');
    disp(A);
                               % exchange b also
    v             = b(i);
    b(i)          = b(idx);
    b(idx)        = v;
    fprintf('New b after row exchange\n');
    disp(b);
   for j          = i+1:n
      mji         = tdiv(A(j,i),A(i,i));
      fprintf('m(%d%d) is %g\n',j,i,mji);
                               % -mji row i + row j
```

```

        for k       = i+1:n
           A(j,k)  = tsub(A(j,k),   tmulti(mji,A(i,k)));

        end
        b(j)        = tsub(b(j),tmulti(mji,b(i)));
     end
     A(i+1:n,i)     =0;
     fprintf('The new A is:\n');
     disp(A);
     fprintf('The new b is:\n');
     disp(b);
end
%% call backward substitution
x                   = backsub(A,b);
end


function y          = tadd(x,y)   %
x                   = tround(x);
y                   = tround(y);
y                   = tround(x+y);
end

function y          = tmulti(x,y)
x                   = tround(x);
y                   = tround(y);
y                   = tround(x*y);
end
function y          = tdiv(x,y)
x                   = tround(x);
y                   = tround(y);
y                   = tround(x/y);
end

function y          = tsub(x,y)
x                   = tround(x);
y                   = tround(y);
y                   = tround(x-y);
end

function y          = tround(x)
global d t
switch t
    case 'round'
        y           = round(x,d,'significant');
    otherwise
        ex          = ceil(log10(abs(x)));
        x           = x/(10^ex);
        y           = fix(x*(10^d))/(10^d);
        x           = x*(10^ex);
end

end

function x          = backsub(A,b)
%% A is a upper triangular matrix
n                   = length(A);
x                   = zeros(n,1);
for i               =n:-1:1
    x(i)            = tround(b(i)/A(i,i));
    fprintf('x(%d) is %g ',i,x(i));
    for j           = 1:i-1
       b(j)         = tround(b(j) -  tmulti(A(j,i),x(i)));
    end
    fprintf('new b is %g\n',b);

end
end
```

2.4 Usage of there functions

First, given value of digits, and choosing rounding or chopping. Then input your matrix A, and vector b.

```
digit = 4;
type = 'round';
A = [3.03,-12.1,14;
     -3.03,12.1,-7;
     6.11 -14.2 21];
b = [-119;120;-139];
x =  GS_partial(A,b,digit,type)
```