Table of Contents


```
% Multi-Step Problem

format long;
```

Setup

```
% Change to format long to show more significant digits:
format long;

% Initialize the tridiagonal matrix:
A = [-2, 1, 0, 0, 0;
     1, -2, 1, 0, 0;
     0, 1. -2, 1, 0;
     0, 0, 1, -2, 1;
     0, 0, 0, 1, -2];

% Initialize the vector of knowns:
b = [1; 2; 3; 4; 5].*6^(-3);
```

Part 1.

```
% This section of code, copied from the assignment specification, performs
% Gaussian elimination on the augmented matrix that represents the given
% linear system. At the end, the code will output the upper triangular
% augmented matrix.

M = [A,b];
[m,n] = size(M);
for j = 1:m
    if M(j,j)==0
        error('System cannot be solved by regular Gaussian elimination.');
    end
    for i = j+1:m
        l_ij = M(i,j)/M(j,j);
        M(i,j:n) = M(i,j:n)-l_ij*M(j,j:n);
    end
end
M
```

```
M =
```

```
  Columns 1 through 3

  -2.000000000000000   1.000000000000000                   0
                   0  -1.500000000000000   1.000000000000000
                   0                   0  -1.333333333333333
                   0                   0                   0
                   0                   0                   0

  Columns 4 through 6

                   0                   0   0.004629629629630
                   0                   0   0.011574074074074
   1.000000000000000                   0   0.021604938271605
  -1.250000000000000   1.000000000000000   0.034722222222222
                   0  -1.200000000000000   0.050925925925926
```

Part 2.

```
% This section of code, copied from Exercise 1, performs back substitution
% on the upper triangular augmented matrix obtained in from Part 1. At the
% end, the code will output the resulting solution vector.

U = M;
[m,n] = size(U);
x = U(:,m+1);
x(m) = U(m,m+1)/U(m,m);
for i = m-1:-1:1
    SUM = 0;
    for j = i+1:m
        SUM = SUM + U(i,j)*x(j);
    end
    x(i) = (U(i,n) - SUM)/U(i,i);
end
x
```

```
x =

  -0.027006172839506
  -0.049382716049383
  -0.062500000000000
  -0.061728395061728
  -0.042438271604938
```

Part 3

```
% This section of code, copied from Part 1, modifies this copied code to
% compute the LU factorization of the coefficient matrix A of the above
% linear system. In the end this code will display the L and U matrices.
```

```
U = A;
[m,n] = size(U);
L = eye(m);
for j = 1:m
    if U(j,j)==0
        error('System cannot be solved by regular Gaussian elimination.');
    end
    for i = j+1:m
        l_ij = U(i,j)/U(j,j);
        U(i,j:n) = U(i,j:n)-l_ij*U(j,j:n);
        L(i,j) = l_ij;
    end
end
L
U

% Lastly, to check if our L and U are correct, we will perform A-LU and see
% if it equals the zero matrix
A-L*U
```

```
L =

  Columns 1 through 3

   1.000000000000000                   0                   0
  -0.500000000000000   1.000000000000000                   0
                   0  -0.666666666666667   1.000000000000000
                   0                   0  -0.750000000000000
                   0                   0                   0

  Columns 4 through 5

                   0                   0
                   0                   0
                   0                   0
   1.000000000000000                   0
  -0.800000000000000   1.000000000000000


U =

  Columns 1 through 3

  -2.000000000000000   1.000000000000000                   0
                   0  -1.500000000000000   1.000000000000000
                   0                   0  -1.333333333333333
                   0                   0                   0
                   0                   0                   0

  Columns 4 through 5

                   0                   0
                   0                   0
```

```
   1.000000000000000                   0
  -1.250000000000000   1.000000000000000
                   0  -1.200000000000000


ans =

     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
```

Published with MATLAB® R2023a