Chapter 9 – Problem 9.18

Develop, debug, and test a program in either a high-level language or macro language of your choice to solve a system of equations using the following methods:

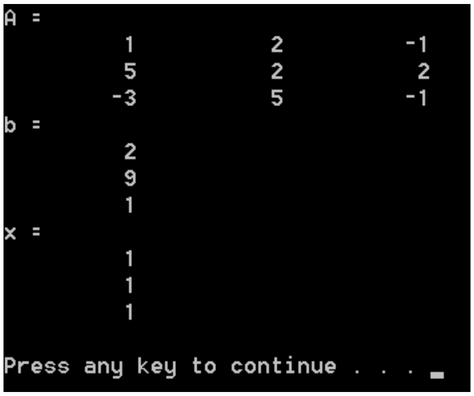
(a) Gauss Elimination with Partial Pivoting (Fig. 9.6 – Pg. 264 | Chapra & Canale)
(b) LU Decomposition (Fig. 10.2 – Pg. 282 | Chapra & Canale)

Test the program using the following system (which has an answer of $x_1 = x_2 = x_3 = 1$),

$$\begin{aligned} x_1 + 2x_2 - x_3 &= 2 \\ 5x_1 + 2x_2 + 2x_3 &= 9 \\ -3x_1 + 5x_2 - x_3 &= 1 \end{aligned}$$

(c) Solve the system using SCILAB.
(d) Modify the program from part (b) to calculate the inverse matrix of the matrix of coefficients from the system (Fig. 10.5 – Pg. 286 | Chapra & Canale).

```
// 02/19/2014 - ENGR 2450 - Meine, Joel
// Chapter 9 - Problem 9.18

// Gauss Elimination with Partial Pivoting

#include <iostream>
#include <iomanip>
#include <math.h>
using namespace std;

const int n = 3;
const double tol = 0.0001;

void Pivot(double a[][n],double b[],double s[],int k)
{
       int p = k;
       double dummy = 0;
       double big = abs(a[k][k] / s[k]);
       for (int ii = k+1; ii < n; ii++)
       {
              dummy = abs(a[ii][k] / s[ii]);
              if (dummy > big)
              {
                     big = dummy;
                     p = ii;
              }
       }
       if (p != k)
       {
              for (int jj = k; jj < n; jj++)
              {
                     dummy = a[p][jj];
                     a[p][jj] = a[k][jj];
```

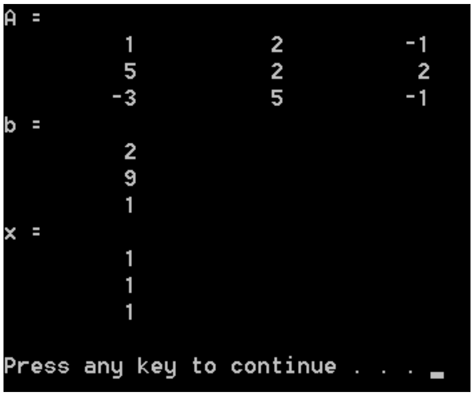


```
A =
         1          2         -1
         5          2          2
        -3          5         -1
b =
         2
         9
         1
x =
         1
         1
         1

Press any key to continue . . .
```

```
                        a[k][jj] = dummy;
                }
                dummy = b[p];
                b[p] = b[k];
                b[k] = dummy;
                dummy = s[p];
                s[p] = s[k];
                s[k] = dummy;
        }
}

void Eliminate(double a[][n],double s[],double b[],int er)
{
        for (int k = 0; k < (n-1); k++)
        {
                Pivot(a,b,s,k);
                if (abs(a[k][k] / s[k]) < tol)
                {
                        er = -1;
                        break;
                }
                for (int i = k+1; i < n; i++)
                {
                        double factor = a[i][k] / a[k][k];
                        for (int j = k+1; j < n; j++) a[i][j] = a[i][j] - factor * a[k][j];
                        b[i] = b[i] - factor * b[k];
                }
        }
        if (abs(a[n-1][n-1] / s[n-1]) < tol) er = -1;
}

void Substitute(double a[][n],double b[],double x[])
{
        x[n-1] = b[n-1] / a[n-1][n-1];
        for (int i = n-2; i >= 0; i--)
        {
                double sum = 0;
                for (int j = i+1; j < n; j++) sum = sum + a[i][j] * x[j];
                x[i] = (b[i] - sum) / a[i][i];
        }
}

void Gauss(double a[][n],double b[],double x[],int er)
{
        double s[n];
        er = 0;
        for (int i = 0; i < n; i++)
        {
                s[i] = abs(a[i][0]);
                for (int j = 1; j < n; j++)
                {
                        if (abs(a[i][j]) > s[i]) s[i] = abs(a[i][j]);
                }
        }
        Eliminate(a,s,b,er);
        if (er != -1) Substitute(a,b,x);
}
```

```
void printmatrix(double A[][n],int n,int m)
{
        int i; int j;
        for (i = 0; i < n; i++){
                for (j = 0; j < m; j++){
                        cout << setw(10) << A[i][j] << " ";
                }; cout << endl;
        };
}

void printvector(double u[],int n)
{
        int i;
        for (i = 0; i < n; i++){
                cout << setw(10) << u[i]; cout << endl;
        };
}

int main()
{
        double A[n][n] = {{1,2,-1},{5,2,2},{-3,5,-1}}; // Matrix A, Initialize
        cout << "A = \n"; printmatrix(A,n,n); // Matrix A, Print
        double b[n] = {2,9,1}; // Vector b, Intialize
        cout << "b = \n"; printvector(b,n); // Vector b, Print
        double x[n]; // Vector x, Intialize
        Gauss(A,b,x,0);
        cout << "x = \n"; printvector(x,n); // Vector x, Print

        cout << "\n";
        system("pause");
        return 0;
}
```

```
// 02/19/2014 - ENGR 2450 - Meine, Joel
// Chapter 10 - Problem 10.18

// LU Decomposition

#include <iostream>
#include <iomanip>
#include <math.h>
using namespace std;

const int n = 3;
const double tol = 0.0001;

void Pivot(double a[][n],int o[],double s[],int k)
{
        int p = k;
        double dummy = 0;
        double big = abs(a[o[k]][k] / s[o[k]]);
        for (int ii = k+1; ii < n; ii++)
        {
                dummy = abs(a[o[ii]][k] / s[o[ii]]);
```



```
A =
         1           2          -1
         5           2           2
        -3           5          -1
b =
         2
         9
         1
x =
         1
         1
         1

Press any key to continue . . .
```

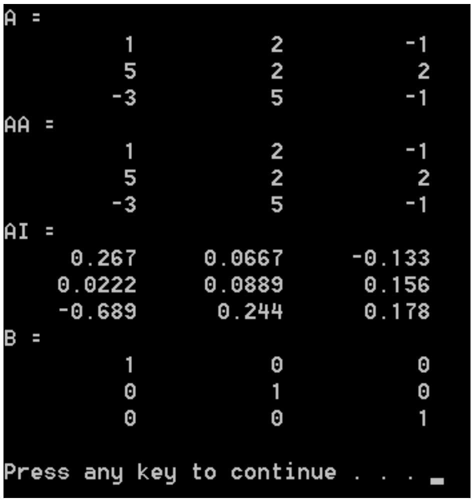
```
                        if (dummy > big)
                        {
                                big = dummy;
                                p = ii;
                        }
                }
                dummy = o[p];
                o[p] = o[k];
                o[k] = dummy;
}

void Decompose(double a[][n],double s[],int o[],int er)
{
        for (int i = 0; i < n; i++)
        {
                o[i] = i;
                s[i] = abs(a[i][0]);
                for (int j = 1; j < n; j++)
                {
                        if (abs(a[i][j]) > s[i]) s[i] = abs(a[i][j]);
                }
        }
        for (int k = 0; k < (n-1); k++)
        {
                Pivot(a,o,s,k);
                if (abs(a[o[k]][k] / s[o[k]]) < tol)
                {
                        er = -1;
                        break;
                }
                for (int i = k+1; i < n; i++)
                {
                        double factor = a[o[i]][k] / a[o[k]][k];
                        a[o[i]][k] = factor;
                        for (int j = k+1; j < n; j++) a[o[i]][j] = a[o[i]][j] - factor * a[o[k]][j];
                }
        }
        if (abs(a[o[n-1]][n-1] / s[o[n-1]]) < tol) er = -1;
}

void Substitute(double a[][n],int o[],double b[],double x[])
{
        for (int i = 1; i < n; i++)
        {
                double sum = b[o[i]];
                for (int j = 0; j <= (i-1); j++) sum = sum - a[o[i]][j] * b[o[j]];
                b[o[i]] = sum;
        }
        x[n-1] = b[o[n-1]] / a[o[n-1]][n-1];
        for (int i = n-2; i >= 0; i--)
        {
                double sum = 0;
                for (int j = i+1; j < n; j++) sum = sum + a[o[i]][j] * x[j];
                x[i] = (b[o[i]] - sum) / a[o[i]][i];
        }
}
```

```
void Ludecomp(double a[][n],double b[],double x[],int er)
{
       int o[n];
       double s[n];
       er = 0;
       Decompose(a,s,o,er);
       if (er != -1) Substitute(a,o,b,x);
}

void printmatrix(double A[][n],int n,int m)
{
       int i; int j;
       for (i = 0; i < n; i++){
              for (j = 0; j < m; j++){
                     cout << setw(10) << A[i][j] << " ";
              }; cout << endl;
       };
}

void printvector(double u[],int n)
{
       int i;
       for (i = 0; i < n; i++){
              cout << setw(10) << u[i]; cout << endl;
       };
}

int main()
{
       double A[n][n] = {{1,2,-1},{5,2,2},{-3,5,-1}}; // Matrix A, Initialize
       cout << "A = \n"; printmatrix(A,n,n); // Matrix A, Print
       double b[n] = {2,9,1}; // Vector b, Intialize
       cout << "b = \n"; printvector(b,n); // Vector b, Print
       double x[n]; // Vector x, Intialize
       Ludecomp(A,b,x,0);
       cout << "x = \n"; printvector(x,n); // Vector x, Print

       cout << "\n";
       system("pause");
       return 0;
}
```

```
-->A = [1,2,-1;5,2,2;-3,5,-1]
 A  =

    1.    2.  - 1.
    5.    2.    2.
  - 3.    5.  - 1.

-->b = [2;9;1]
 b  =

    2.
    9.
    1.
```

```
-->x = inv(A)*b
 x  =

    1.
    1.
    1.
```

```
// 02/19/2014 - ENGR 2450 - Meine, Joel
// Chapter 10 - Problem 10.19

// Inverse Matrix

#include <iostream>
#include <iomanip>
#include <math.h>
using namespace std;

const int n = 3;
const double tol = 0.0001;

void Pivot(double a[][n],int o[],double s[],int k)
{
        int p = k;
        double dummy = 0;
        double big = abs(a[o[k]][k] / s[o[k]]);
        for (int ii = k+1; ii < n; ii++)
        {
                dummy = abs(a[o[ii]][k] / s[o[ii]]);
                if (dummy > big)
                {
                        big = dummy;
                        p = ii;
                }
        }
        dummy = o[p];
        o[p] = o[k];
        o[k] = dummy;
}

void Decompose(double a[][n],double s[],int o[],int er)
{
        for (int i = 0; i < n; i++)
        {
                o[i] = i;
                s[i] = abs(a[i][0]);
                for (int j = 1; j < n; j++)
                {
                        if (abs(a[i][j]) > s[i]) s[i] = abs(a[i][j]);
                }
        }
        for (int k = 0; k < (n-1); k++)
        {
                Pivot(a,o,s,k);
                if (abs(a[o[k]][k] / s[o[k]]) < tol)
```



```
A =
         1          2         -1
         5          2          2
        -3          5         -1
AA =
         1          2         -1
         5          2          2
        -3          5         -1
AI =
     0.267     0.0667     -0.133
    0.0222     0.0889      0.156
    -0.689      0.244      0.178
B =
         1          0          0
         0          1          0
         0          0          1

Press any key to continue . . .
```

```
                {
                        er = -1;
                        break;
                }
                for (int i = k+1; i < n; i++)
                {
                        double factor = a[o[i]][k] / a[o[k]][k];
                        a[o[i]][k] = factor;
                        for (int j = k+1; j < n; j++) a[o[i]][j] = a[o[i]][j] - factor * a[o[k]][j];
                }
        }
        if (abs(a[o[n-1]][n-1] / s[o[n-1]]) < tol) er = -1;
}

void Substitute(double a[][n],int o[],double b[],double x[])
{
        for (int i = 1; i < n; i++)
        {
                double sum = b[o[i]];
                for (int j = 0; j <= (i-1); j++) sum = sum - a[o[i]][j] * b[o[j]];
                b[o[i]] = sum;
        }
        x[n-1] = b[o[n-1]] / a[o[n-1]][n-1];
        for (int i = n-2; i >= 0; i--)
        {
                double sum = 0;
                for (int j = i+1; j < n; j++) sum = sum + a[o[i]][j] * x[j];
                x[i] = (b[o[i]] - sum) / a[o[i]][i];
        }
}

void MatrixInverse(double a[][n],double ai[][n],int er)
{
        int o[n];
        double s[n];
        double b[n];
        double x[n];
        er = 0;
        Decompose(a,s,o,er);
        if (er == 0)
        {
                for (int i = 0; i < n; i++)
                {
                        for (int j = 0; j < n; j++)
                        {
                                if (i == j) b[j] = 1;
                                else b[j] = 0;
                        }
                        Substitute(a,o,b,x);
                        for (int j = 0; j < n; j++) ai[j][i] = x[j];
                }
        }
        else std::cout << "Ill-Conditioned System" << std::endl;
}

void copymatrix(double A[][n],double B[][n],int n,int m)
{
        int i; int j;
        for (i = 0; i < n; i++)
        {
                for (j = 0; j < m; j++)
                {
                        B[i][j] = A[i][j];
                }
        }
}

void multiply_matrices(double A[][n],double B[][n],double C[][n],int n,int m,int p)
{
        int i; int j; int k;
        for (i = 0; i < n; i++){
                for (j = 0; j < m; j++){
                        C[i][j] = 0;
                        for (k = 0; k < p; k++){
                                C[i][j] = C[i][j] + A[i][k] * B[k][j];
                        };
                };
        };
}

void printmatrix(double A[][n],int n,int m,int p)
{
        int i; int j;
        for (i = 0; i < n; i++){
                for (j = 0; j < m; j++){
                        cout << setw(10) << setprecision(p) << A[i][j] << " ";
                }; cout << endl;
        };
}

void printvector(double u[],int n)
{
        int i;
        for (i = 0; i < n; i++){
                cout << setw(10) << u[i]; cout << endl;
        };
}

int main()
{
        double A[n][n] = {{1,2,-1},{5,2,2},{-3,5,-1}}; // Matrix A, Initialize
        cout << "A = \n"; printmatrix(A,n,n,0); // Matrix A, Print
        double AA[n][n]; // Matrix AA, Initialize
        copymatrix(A,AA,n,n); // AA = A
        cout << "AA = \n"; printmatrix(AA,n,n,0); // Matrix AA, Print
        double AI[n][n]; // Matrix AI, Intialize
        MatrixInverse(AA,AI,0); // AI = AA^-1
        cout << "AI = \n"; printmatrix(AI,n,n,3); // Matrix AI, Print
        double b1[n][n]; // Matrix b1, Initialize
        multiply_matrices(A,AI,b1,n,n,n); // b1 = A * AI
        int b2[n][n]; // Matrix b2, Initialize
        double B[n][n]; // Matrix B, Initialize
        for (int i = 0; i < n; i++)
        {
                for (int j = 0; j < n; j++)
                {
                        b2[i][j] = b1[i][j];
                        B[i][j] = b2[i][j];
                }
        }
        cout << "B = \n"; printmatrix(B,n,n,0); // Matrix B, Print

        cout << "\n";
        system("pause");
        return 0;
}
```

Chapter 11 – Problem 11.25

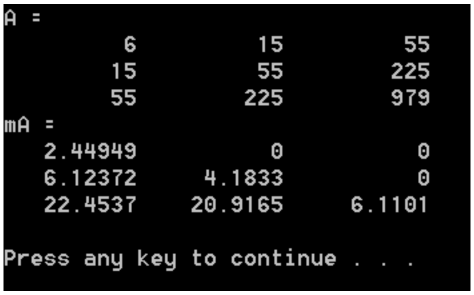
Develop a user-friendly program in either a high-level or macro language of your choice for Cholesky decomposition based on Fig. 11.3 (Pg. 300 | Chapra & Canale). Test your program using the following system based on Example 11.2 (Pg. 299 | Chapra & Canale),

$$[A]=\begin{bmatrix} 6 & 15 & 55 \\ 15 & 55 & 225 \\ 55 & 225 & 979 \end{bmatrix}$$

After matrix $[A]$ completes Cholesky decomposition, the resulting matrix $[L]$ will be,

$$[L]=\begin{bmatrix} 2.4495 & & \\ 6.1237 & 4.1833 & \\ 22.454 & 20.917 & 6.1101 \end{bmatrix}$$

```
// 02/19/2014 - ENGR 2450 - Meine, Joel
// Chapter 11 - Problem 11.25

// Cholesky Decomposition

#include <iostream>
#include <iomanip>
#include <math.h>
using namespace std;

const int n = 3;

void copymatrix(double A[][n],double B[][n],int n,int m)
{
        int i; int j;
        for (i = 0; i < n; i++)
        {
                for (j = 0; j < m; j++)
                {
                        B[i][j] = A[i][j];
```



```
A =
           6          15          55
          15          55         225
          55         225         979
mA =
     2.44949           0           0
     6.12372      4.1833           0
     22.4537     20.9165      6.1101

Press any key to continue . . .
```

```
                }
        }
}

void printmatrix(double A[][n],int n,int m, int s)
{
        int i; int j;
        for (i = 0; i < n; i++){
                for (j = 0; j < m; j++){
                        if (i == 0 && j == 1 && s == 1) A[i][j] = 0;
                        else if (i == 0 && j == 2 && s == 1) A[i][j] = 0;
                        else if (i == 1 && j == 2 && s == 1) A[i][j] = 0;
                        cout << setw(10) << A[i][j] << " ";
                }; cout << endl;
        };
}

void cLudecomp(double a[][n],double b[][n])
{
        double sum = 0;
        for (int k = 0; k < n; k++)
        {
                for (int i = 0; i <= (k-1); i++)
                {
                        sum = 0;
                        for (int j = 0; j <= (i-1); j++) sum = sum + a[i][j] * a[k][j];
                        a[k][i] = (a[k][i] - sum) / a[i][i];
                }
                sum = 0;
                for (int j = 0; j <= (k-1); j++) sum = sum + pow(a[k][j],2);
                a[k][k] = sqrt(a[k][k] - sum);
        }
        copymatrix(a,b,n,n);
}

int main()
{
        double A[n][n] = {{6,15,55},{15,55,225},{55,225,979}}; // Matrix A, Initialize
        cout << "A = \n"; printmatrix(A,n,n,0); // Matrix A, Print
        double mA[n][n]; // Matrix mA, Intialize
        cLudecomp(A,mA); // mA ~ A
        cout << "mA = \n"; printmatrix(mA,n,n,1); // Matrix mA, Print

        cout << "\n";
        system("pause");
        return 0;
}
```

Chapter 11 – Problem 11.26

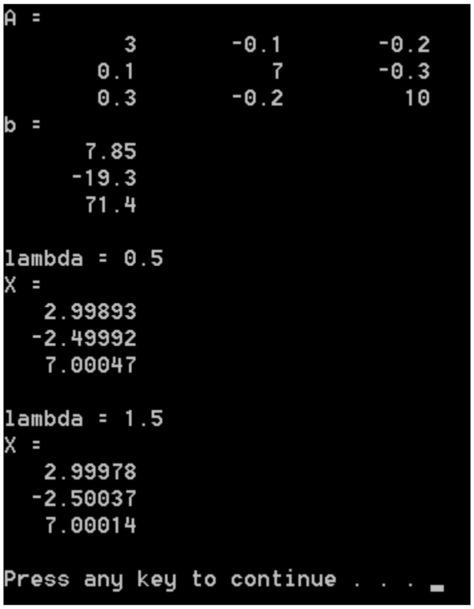
Develop a user-friendly program in either a high-level or macro language of your choice for Gauss-Seidel method based on Fig. 11.6 (Pg. 306 | Chapra & Canale). Test your program using the following system based on Example 11.3 (Pg. 301 | Chapra & Canale),

$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85$$
$$0.1x_1 + 7x_2 - 0.3x_3 = -19.3$$
$$0.3x_1 - 0.2x_2 + 10x_3 = 71.4$$

The true solutions are $x_1 = 3$, $x_2 = -2.5$, and $x_3 = 7$.

```
// 02/19/2014 - ENGR 2450 - Meine, Joel
// Chapter 11 - Problem 11.26

// Gauss-Seidel with Relaxation

#include <iostream>
#include <iomanip>
#include <math.h>
using namespace std;

const double es = 0.05;
const int imax = 50;
const int n = 3;

void Gseid(double a[][n],double b[],double x[],int imax,double es,double lambda)
{
    double sum = 0;
    for (int i = 0; i < n; i++)
    {
        double dummy = a[i][i];
        for (int j = 0; j < n; j++) a[i][j] = a[i][j] / dummy;
        b[i] = b[i] / dummy;
    }
    for (int i = 0; i < n; i++)
    {
        sum = b[i];
        for (int j = 0; j < n; j++)
        {
            if (i != j) sum = sum - a[i][j] * x[j];
        }
        x[i] = sum;
    }
    int iter = 1;
    int sentinel = 0;
    do
    {
        sentinel = 1;
        for (int i = 0; i < n; i++)
        {
            double old = x[i];
            sum = b[i];
            for (int j = 0; j < n; j++)
            {
                if (i != j) sum = sum - a[i][j] * x[j];
            }
            x[i] = lambda*sum + (1 - lambda)*old;
            if (sentinel == 1 && x[i] != 0)
```

```
                {
                        double ea = abs((x[i]-old)/x[i])*100;
                        if (ea > es) sentinel = 0;
                }
            }
            iter++;
        } while (sentinel != 1 || iter < imax);
        if (sentinel == 1) ;
        else if (iter >= imax) std::cout << "No Solution due to Divergence" << std::endl;
}

void printmatrix(double A[][n],int n,int m)
{
        int i; int j;
        for (i = 0; i < n; i++){
                for (j = 0; j < m; j++){
                        cout << setw(10) << A[i][j] << " ";
                }; cout << endl;
        };
}

void printvector(double u[],int n)
{
        int i;
        for (i = 0; i < n; i++){
                cout << setw(10) << u[i]; cout << endl;
        };
}

int main()
{
        const double L1 = 0.5;
        const double L2 = 1.5;

        double A[n][n] = {{3,-0.1,-0.2},{0.1,7,-0.3},{0.3,-0.2,10}}; // Matrix A, Initialize
        cout << "A = \n"; printmatrix(A,n,n); // Matrix A, Print
        double b[n] = {7.85,-19.3,71.4}; // Vector b, Intialize
        cout << "b = \n"; printvector(b,n); // Vector b, Print
        double X1[n]; // Vector X1, Intialize
        Gseid(A,b,X1,imax,es,L1); // lambda = 0.5
        std::cout << "\nlambda = " << L1 << std::endl;
        cout << "X = \n"; printvector(X1,n); // Vector X1, Print
        double X2[n]; // Vector X2, Initialize
        Gseid(A,b,X2,imax,es,L2); // lambda = 1.5
        std::cout << "\nlambda = " << L2 << std::endl;
        cout << "X = \n"; printvector(X2,n); // Vector X2, Print

        cout << "\n";
        system("pause");
        return 0;
}
```



```
A =
         3       -0.1       -0.2
       0.1          7       -0.3
       0.3       -0.2         10
b =
      7.85
     -19.3
      71.4

lambda = 0.5
X =
   2.99893
  -2.49992
   7.00047

lambda = 1.5
X =
   2.99978
  -2.50037
   7.00014

Press any key to continue . . . _
```