Systems of Linear Equations

Last time, we found that solving equations such as Poisson's equation or Laplace's equation on a grid is equivalent to solving a system of linear equations.

There are many other examples where systems of linear equations appear, such as eigenvalue problems. In this lecture, we look into different approaches to solving systems of linear equations (SLE's).

SLE:

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= b_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots + a_{3n}x_n &= b_3 \\
\ldots & \\
a_{m}x_1 + a_{m2}x_2 + a_{m3}x_3 + \cdots + a_{mn}x_n &= b_m
\end{aligned}
$$

n unknowns x_j
m equations
a_{ij} and b_i are known

SLE's

If *m<n,* there is not a unique solution for the *x's* (underconstrained). If *m>n,* the system can be overconstrained and usually the job is to find the best set *{x}* to represent the set if equations. We will consider here square matrices *m=n,* with *detA≠0,* in which case the equations are linearly independent and there is a unique solution.

Matrix representation:

$$A\vec{x} = \vec{b}$$

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ & \cdots & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \qquad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

SLE's

Note that if the matrix is diagonal, then the solutions is easy:

$$\begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ & \cdots & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

$$a_{11}x_1 = b_1 \qquad\qquad x_1 = b_1 / a_{11}$$

$$a_{21}x_1 + a_{22}x_2 = b_2 \qquad\qquad x_2 = \frac{\left(b_2 - {a_{21}b_1}/{a_{11}}\right)}{a_{22}}$$

$$\vdots$$

Goal of 'Gaussian elimination' method is to bring *A* into this form.

SLE's

$$A\vec{x} = \vec{b}$$

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ & \cdots & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

Note:

1. Interchanging two rows of A and the same rows of b gives the same set of equations.
2. Solution not changed if replace a row with a linear combination including another row, as long as the *b*'s are treated in a similar way. E.g., $a_{ij} \rightarrow k_i a_{ij} + k_{i'} a_{i'j} \qquad j = 1, \cdots, n$

   $$b_i \rightarrow k_i b_i + k_{i'} b_{i'}$$

3. Interchanging two columns of A gives the same result if simultaneously two rows of vector x interchanged.

Gauss Elimination Method

define $l_{i1} = \dfrac{a_{i1}}{a_{11}}$

Transform A by subtracting $l_{i1}\vec{a}_1^t$ from row i where $\vec{a}_1^t$ is the transpose of row 1 .

$$\begin{pmatrix} \vec{a}_1^t \\ \vec{a}_2^t \\ \vdots \\ \vec{a}_n^t \end{pmatrix} \rightarrow \begin{pmatrix} \vec{a}_1^t \\ \vec{a}_2^t - l_{21}\vec{a}_1^t \\ \vdots \\ \vec{a}_n^t - l_{n1}\vec{a}_1^t \end{pmatrix}$$

or $A^{(1)} = L_1 A$ where

$$L_1 = \begin{pmatrix} 1 & & & & \\ -l_{21} & 1 & & 0 & \\ -l_{31} & & 1 & & \\ \vdots & & 0 & \ddots & \\ -l_{n1} & & & & 1 \end{pmatrix}$$

is the Frobenius matrix

Gauss Elimination Method

The result is

$$A^{(1)} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \cdots & & \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{pmatrix}$$

now subtract $a_{i2}^{(1)}/a_{22}^{(1)}$ times the second row from rows $3 \ldots n$

$$A^{(2)} = L_2 A^{(1)} = L_2 L_1 A$$

$$L_2 = \begin{pmatrix} 1 & & & & 0 \\ 0 & 1 & & & \\ 0 & -l_{32} & 1 & & \\ 0 & \vdots & 0 & \ddots & \\ 0 & -l_{n2} & 0 & 0 & 1 \end{pmatrix} \quad \text{where } l_{i2} = a_{i2}^{(1)}/a_{22}^{(1)}$$

Gauss Elimination Method

Now have:

$$A^{(2)} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} \end{pmatrix}$$

Keep going until have upper triangular matrix $(n-1)$ steps

$$A^{(n-1)} = L_{n-1}L_{n-2}\cdots L_1 A = U$$

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & u_{22} & u_{23} & \cdots & u_{2n} \\ 0 & 0 & u_{33} & \cdots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & u_{nn} \end{pmatrix} \quad \text{with } U\vec{x} = \vec{c} \text{ and } \vec{c} = L_{n-1}L_{n-2}\cdots L_1\vec{b}$$

Gauss Elimination Method

Note that:

$$L_1 = \begin{pmatrix} 1 & & & & \\ -l_{21} & 1 & & 0 & \\ -l_{31} & & 1 & & \\ \vdots & & 0 & \ddots & \\ -l_{n1} & & & & 1 \end{pmatrix} \quad L_1^{-1} = \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & 0 & \\ l_{31} & & 1 & & \\ \vdots & & 0 & \ddots & \\ l_{n1} & & & & 1 \end{pmatrix}$$

and similarly for the other L_i, so

$$L_{n-1}L_{n-2}\cdots L_1 A = U \quad \text{implies} \quad A = L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1}U = LU, \text{with}$$

$$L = L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1} = \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & 0 & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & l_{43} & \ddots & \\ l_{n1} & l_{n2} & & l_{nn-1} & 1 \end{pmatrix}$$

Gauss Elimination Method

i.e., the matrix A has been decomposed (LU decomposition) into an upper triangular and a lower triangular matrix. The solution is now easy.

$A\vec{x} = LU\vec{x} = \vec{b}$ First solve $L\vec{y} = \vec{b}$, then $U\vec{x} = \vec{y}$ so

$$y_1 = b_1$$

$$y_2 = b_2 - l_{21} y_1$$

…

then going from the bottom

$$x_n = \frac{y_n}{u_{nn}}$$

$$x_{n-1} = \frac{y_{n-1} - x_n u_{n-1,n}}{u_{n-1,n-1}}$$

…

Gauss Elimination Method

Take a concrete example:

$$\begin{pmatrix} 4 & 1 & 1 & 1 \\ 1 & 4 & 1 & 1 \\ 1 & 1 & 4 & 1 \\ 1 & 1 & 1 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Here is some code:

```
* Loop over the rows.  The index i is the row which is not touched in this step
    Do i=1,n-1
*
* For step i, we modify all rows j>i
      Do j=i+1,n
*
* Loop over the column elements in this row.  Perform linear transformation
* on matrix A. Keep the upper diagonal elements in A as we go. Also build
* up the lower diagonal matrix at the same time.
        Lji=A(j,i)/A(i,i)
        Do k=1,n
          A(j,k)=A(j,k)-Lji*A(i,k)
        Enddo
        L(j,i)=Lji
      Enddo
    Enddo
```

Gauss Elimination Method

$$A^{(1)}=\begin{matrix} 4.00000 & 1.00000 & 1.00000 & 1.00000 \\ 0.00000 & 3.75000 & 0.75000 & 0.75000 \\ 0.00000 & 0.75000 & 3.75000 & 0.75000 \\ 0.00000 & 0.75000 & 0.75000 & 3.75000 \end{matrix} \qquad L^{(1)}=\begin{matrix} 1.00000 & 0.00000 & 0.00000 & 0.00000 \\ 0.25000 & 1.00000 & 0.00000 & 0.00000 \\ 0.25000 & 0.00000 & 1.00000 & 0.00000 \\ 0.25000 & 0.00000 & 0.00000 & 1.00000 \end{matrix}$$

$$A^{(2)}=\begin{matrix} 4.00000 & 1.00000 & 1.00000 & 1.00000 \\ 0.00000 & 3.75000 & 0.75000 & 0.75000 \\ 0.00000 & 0.00000 & 3.60000 & 0.60000 \\ 0.00000 & 0.00000 & 0.60000 & 3.60000 \end{matrix} \qquad L^{(2)}=\begin{matrix} 1.00000 & 0.00000 & 0.00000 & 0.00000 \\ 0.25000 & 1.00000 & 0.00000 & 0.00000 \\ 0.25000 & 0.20000 & 1.00000 & 0.00000 \\ 0.25000 & 0.20000 & 0.00000 & 1.00000 \end{matrix}$$

$$A^{(3)}=\begin{matrix} 4.00000 & 1.00000 & 1.00000 & 1.00000 \\ 0.00000 & 3.75000 & 0.75000 & 0.75000 \\ 0.00000 & 0.00000 & 3.60000 & 0.60000 \\ 0.00000 & 0.00000 & 0.00000 & 3.50000 \end{matrix} \qquad L^{(3)}=\begin{matrix} 1.00000 & 0.00000 & 0.00000 & 0.00000 \\ 0.25000 & 1.00000 & 0.00000 & 0.00000 \\ 0.25000 & 0.20000 & 1.00000 & 0.00000 \\ 0.25000 & 0.20000 & 0.16667 & 1.00000 \end{matrix}$$

Gauss Elimination Method

```
* Now build up the solution matrix
* First the vector y
      Do i=1,n
        y(i)=b(i)
        Do j=1,i-1
          y(i)=y(i)-L(i,j)*y(j)
        Enddo
      Enddo
*
* and now x
*
      Do i=n,1,-1
        x(i)=y(i)
        Do j=i+1,n
          x(i)=x(i)-A(i,j)*x(j)
        Enddo
        x(i)=x(i)/A(i,i)
      Enddo
```

$$y = \begin{matrix} 1.00 \\ 0.75 \\ 0.60 \\ 0.50 \end{matrix}$$

$$x = \begin{matrix} 0.14 \\ 0.14 \\ 0.14 \\ 0.14 \end{matrix}$$

It’s good practice to check result. Is Ax=b ? Is A=LU ?

Gauss-Jordan Elimination

The Gauss-Jordan method changes the matrix A into the diagonal matrix in one pass. First try, use 2. from P.4 to recast equations:

$$a_{1j} \rightarrow a'_{1j} = \frac{a_{1j}}{a_{11}} \qquad j = 1,2,3,4$$

Then take the following linear combination:

$$a'_{ij} = a_{ij} + k_i a'_{1j} = a_{ij} + k_i \frac{a_{1j}}{a_{11}} \qquad i = 2,3,4$$

and choose k_i such that

$$a'_{i1} = 0 = a_{i1} + k_i a'_{11} = a_{i1} + k_i \qquad i = 2,3,4$$
$$\text{or } \; k_i = -a_{i1}$$

Gauss-Jordan Elimination

Work with a 4x4 matrix to be concrete:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 {\scriptstyle a_{11}} \end{pmatrix}$$

After the first step, our matrix looks like this

$$\begin{pmatrix} 1 & a_{12}/a_{11} & a_{13}/a_{11} & a_{14}/a_{11} \\ 0 & a_{22} - a_{21}\, a_{12}/a_{11} & a_{23} - a_{21}\, a_{13}/a_{11} & a_{24} - a_{21}\, a_{14}/a_{11} \\ 0 & a_{32} - a_{31}\, a_{12}/a_{11} & a_{33} - a_{31}\, a_{13}/a_{11} & a_{34} - a_{31}\, a_{14}/a_{11} \\ 0 & a_{42} - a_{41}\, a_{12}/a_{11} & a_{43} - a_{41}\, a_{13}/a_{11} & a_{44} - a_{41}\, a_{14}/a_{11} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1/a_{11} \\ b_2 - a_{21}\, b_1/a_{11} \\ b_3 - a_{31}\, b_1/a_{11} \\ b_4 - a_{41}\, b_1/a_{11} \end{pmatrix}$$

Now we move on and make the second column look like the unit matrix, and so on.

Gauss-Jordan Elimination

1. $$a''_{2j} = {a'_{2j}}/{a'_{22}}$$

2. $$a''_{ij} = a'_{ij} - a'_{i2}\, {a'_{2j}}/{a'_{22}} \qquad i = 1,3,4$$

3. $$b''_{i} = b'_{i} - a'_{i2}\, {b'_{2}}/{a'_{22}} \qquad i = 1,3,4$$

Note that first column is not affected:

$$a''_{21} = {0}/{a'_{22}} = 0$$

$$a''_{i1} = a'_{i1} - a'_{i2}\, {a'_{21}}/{a'_{22}} = a'_{i1} - a'_{i2}\, {0}/{a'_{22}} = a'_{i1} \qquad i = 1,3,4$$

Gauss-Jordan Elimination

After these two sets of transformations, we have

$$\begin{pmatrix} 1 & 0 & a''_{13} & a''_{14} \\ 0 & 1 & a''_{23} & a''_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & a''_{43} & a''_{44} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b''_1 \\ b''_2 \\ b''_3 \\ b''_4 \end{pmatrix}$$

Keep going until we have

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1^{(4)} \\ b_2^{(4)} \\ b_3^{(4)} \\ b_4^{(4)} \end{pmatrix} \Rightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1^{(4)} \\ b_2^{(4)} \\ b_3^{(4)} \\ b_4^{(4)} \end{pmatrix}$$

Gauss-Jordan Elimination

Algorithm looks like this:

1. Loop over the n columns. We want to turn the columns one at a time into the unit matrix
2. For column k, we find the linear transformation on the rows which gives the desired result (1 in row k, 0 in other rows)
   a) Loop over the n rows and make the diagonal element a 1. For i=k, do this by dividing every element in A(i,j) by A(k,k), where j is the column index and i is the row index. Also need to divide b(i) by A(k,k)
   b) For i≠k, make the element in column k a 0. We do this by subtracting A(i,k)*A(k,j)/A(k,k) from A(i,j). For b(i), we subtract A(i,k)*b(k)/A(k,k)

Test matrices:

$$A_1 = \begin{pmatrix} 4 & 1 & 1 & 1 \\ 1 & 4 & 1 & 1 \\ 1 & 1 & 4 & 1 \\ 1 & 1 & 1 & 4 \end{pmatrix} \qquad b_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \qquad A_2 = \begin{pmatrix} 1 & 1/2 & 1/3 & 1/4 \\ 1/2 & 1/3 & 1/4 & 1/5 \\ 1/3 & 1/4 & 1/5 & 1/6 \\ 1/4 & 1/5 & 1/6 & 1/7 \end{pmatrix} \qquad b_2 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Gauss-Jordan Elimination

Start with A_1:

4.00	1.00	1.00	1.00		1.00
1.00	4.00	1.00	1.00	b_1	1.00
1.00	1.00	4.00	1.00		1.00
1.00	1.00	1.00	4.00		1.00

First iteration

1.00	0.25	0.25	0.25	0.25
0.00	3.75	0.75	0.75	0.75
0.00	0.75	3.75	0.75	0.75
0.00	0.75	0.75	3.75	0.75

Second iteration

1.00	0.00	0.20	0.20	0.20
0.00	1.00	0.20	0.20	0.20
0.00	0.00	3.60	0.60	0.60
0.00	0.00	0.60	3.60	0.60

Third iteration

1.00	0.00	0.00	0.17	0.17
0.00	1.00	0.00	0.17	0.17
0.00	0.00	1.00	0.17	0.17
0.00	0.00	0.00	3.50	0.50

Fourth iteration

1.00	0.00	0.00	0.00	0.14
0.00	1.00	0.00	0.00	0.14
0.00	0.00	1.00	0.00	0.14
0.00	0.00	0.00	1.00	0.14

Gauss-Jordan Elimination

We can use the same transformations to get the inverse of A:

$$AA^{-1} = E$$

now suppose the transformation of A to E consists

of the following steps:

$$T_4T_3T_2T_1A = E$$

Apply to the equation above

$$T_4T_3T_2T_1AA^{-1} = T_4T_3T_2T_1E$$

or

$$EA^{-1} = A^{-1} = T_4T_3T_2T_1E$$

Gauss-Jordan Elimination

Let's try it on the previous example:

1.00000	0.00000	0.00000	0.00000
0.00000	1.00000	0.00000	0.00000
0.00000	0.00000	1.00000	0.00000
0.00000	0.00000	0.00000	1.00000

After T_1

0.25000	0.00000	0.00000	0.00000
-.25000	1.00000	0.00000	0.00000
-.25000	0.00000	1.00000	0.00000
-.25000	0.00000	0.00000	1.00000

After T_2

0.26667	-.06667	0.00000	0.00000
-.06667	0.26667	0.00000	0.00000
-.20000	-.20000	1.00000	0.00000
-.20000	-.20000	0.00000	1.00000

After T_3

0.27778	-.05556	-.05556	0.00000
-.05556	0.27778	-.05556	0.00000
-.05556	-.05556	0.27778	0.00000
-.16667	-.16667	-.16667	1.00000

After T_4

0.28571	-.04762	-.04762	-.04762
-.04762	0.28571	-.04762	-.04762
-.04762	-.04762	0.28571	-.04762
-.04762	-.04762	-.04762	0.28571

Gauss-Jordan Elimination

With A_2:

1.00000	0.50000	0.33333	0.25000	1.00
0.50000	0.33333	0.25000	0.20000	1.00
0.33333	0.25000	0.20000	0.16667	1.00
0.25000	0.20000	0.16667	0.14286	1.00

1.00000	0.50000	0.33333	0.25000	1.00
0.00000	0.08333	0.08333	0.07500	0.50
0.00000	0.08333	0.08889	0.08333	0.67
0.00000	0.07500	0.08333	0.08036	0.75

1.00000	0.00000	-.16667	-.20000	-2.00
0.00000	1.00000	1.00000	0.90000	6.00
0.00000	0.00000	0.00556	0.00833	0.17
0.00000	0.00000	0.00833	0.01286	0.30

1.00000	0.00000	0.00000	0.05000	3.00
0.00000	1.00000	0.00000	-.60000	-24.00
0.00000	0.00000	1.00000	1.50000	30.00
0.00000	0.00000	0.00000	0.00036	0.05

Start to see rounding issues:

1.00000	0.00000	0.00000	0.00000	-4.00
0.00000	1.00000	0.00000	0.00000	59.99
0.00000	0.00000	1.00000	0.00000	-179.99
0.00000	0.00000	0.00000	1.00000	139.99

Gauss-Jordan Elimination

As a check, we calculate the result for x for different initial matrices. Use matrices of the form A_2:

$$a_{ij} = \frac{1}{i+j-1}$$

Try different size square matrices. Should give us our initial vector b. Find that start into mumerical problems with n=m=10.

Single precision calculation. All values should be 1.

0.971252441 0.977539062 0.981124878 0.983856201 0.985870361
0.987503052 0.988098145 0.989746094 0.99067688 0.991607666

Double precision calculation. All values are =1.

Gauss-Jordan Elimination

Check 100x100 matrix double precision

0.999999987 0.999999987 0.999999988 0.999999988 0.999999988 0.999999989
0.999999989 0.999999989 0.999999989 0.99999999 0.99999999 0.99999999
0.99999999 0.999999991 0.999999991 0.999999991 0.999999991 0.999999991
0.999999992 0.999999992 0.999999992 0.999999992 0.999999992 0.999999992
0.999999993 0.999999993 0.999999993 0.999999993 0.999999993 0.999999993
0.999999993 0.999999993 0.999999993 0.999999994 0.999999994 0.999999994
0.999999994 0.999999994 0.999999994 0.999999994 0.999999994 0.999999994
0.999999994 0.999999994 0.999999995 0.999999995 0.999999995 0.999999995
0.999999995 0.999999995 0.999999995 0.999999995 0.999999995 0.999999995
0.999999995 0.999999995 0.999999995 0.999999995 0.999999995 0.999999995
0.999999996 0.999999996 0.999999996 0.999999996 0.999999996 0.999999996
0.999999996 0.999999996 0.999999996 0.999999996 0.999999996 0.999999996
0.999999996 0.999999996 0.999999996 0.999999996 0.999999996 0.999999996
0.999999996 0.999999996 0.999999996 0.999999996 0.999999996 0.999999996
0.999999996 0.999999996 0.999999997 0.999999997 0.999999997 0.999999997
0.999999997 0.999999997 0.999999997 0.999999997 0.999999997 0.999999997
0.999999997 0.999999997 0.999999997 0.999999997

Looks pretty good. However, there are cases where this simple Gauss-Jordan technique does not work so well.

Pivoting

What if the diagonal element which we divide by is very small or zero ? (The element we divide by is called the pivot). This obviously causes problems. Resolved by a technique called pivoting (exchanging rows and columns).

Partial pivoting - exchange rows in the region which have not yet been transformed to the unit matrix.

Full pivoting - exchange rows and columns.

These tricks are allowed according to the rules we stated on pages 4,5.

Partial Pivoting

Algorithm looks like this:

1. Loop over the n columns. We want to turn the columns one at a time into the unit matrix. Use the index k to specify which column element we want to turn into a 1.
2. For each value of k, we loop over the rows $i \geq k$ and look for the maximum value A(i,k). The row i which gives this maximum is swapped with row k. Use an index vector to keep track of the permutations Ind(i)=row which is to be considered as i^{th} row.
3. Find the linear transformation:
   a) Loop over the n rows and make the diagonal element a 1. For Ind(i)=k, do this by dividing every element in A(Ind(i),j) by A(Ind(i),k), where j is the column index and Ind(i) is the row index. Also need to divide b(Ind(i)) by A(Ind(i),k)
   b) For $Ind(i) \neq k$, make the element in column k a 0. We do this by subtracting A(Ind(i),k)*A(Ind(k),j)/A(Ind(k),k) from A(Ind(i),j). For b(i), we subtract A(Ind(i),k)*b(Ind(k))/A(Ind(k),k)

Partial Pivoting

Example:

1.00000	0.50000	0.33333	0.25000	1.00
0.50000	0.33333	0.25000	0.20000	1.00
0.33333	0.25000	0.20000	0.16667	1.00
0.25000	0.20000	0.16667	0.14286	1.00

1.00000	0.50000	0.33333	0.25000	1.00
0.00000	0.08333	0.08333	0.07500	0.50
0.00000	0.08333	0.08889	0.08333	0.67
0.00000	0.07500	0.08333	0.08036	0.75

1.00000	0.00000	-.16667	-.20000	-2.00
0.00000	1.00000	1.00000	0.90000	6.00
0.00000	0.00000	0.00556	0.00833	0.17
0.00000	0.00000	0.00833	0.01286	0.30

1.00000	0.00000	0.00000	0.05714	4.00
0.00000	1.00000	0.00000	-.64286	-30.00
0.00000	0.00000	0.00000	-.00024	-0.03
0.00000	0.00000	1.00000	1.54286	36.00

1.00000	0.00000	0.00000	0.00000	-4.00
0.00000	1.00000	0.00000	0.00000	60.00
0.00000	0.00000	0.00000	1.00000	139.99
0.00000	0.00000	1.00000	0.00000	-179.99

Exercizes

1. Solve for x in Ax=b using the LU decomposition method

$$a_{ij} = \frac{1}{i+j+1} \qquad b_j = j$$

2. Do it again with the Gauss-Jordan method

3. (more difficult). Now try

$$a_{ij} = i - j\delta_{i,j} \qquad b_j = j$$

You will need to use pivoting.