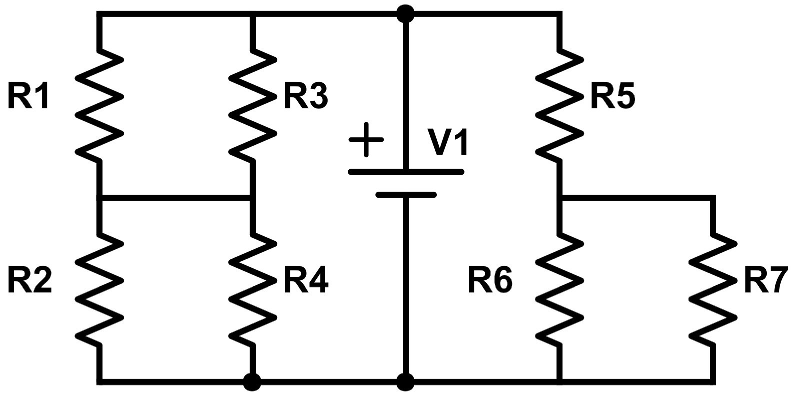
Computational Physics

Systems of Linear equations

direct and iterative methods

Fernando Barao, Phys Department IST (Lisbon)

Electrical circuit



$$\begin{pmatrix} R_1 & 0 & R_3 & 0 & 0 & 0 & 0 \\ 0 & R_2 & 0 & R_4 & 0 & 0 & 0 \\ 0 & 0 & R_3 & R_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & R_5 & R_6 & 0 \\ 0 & 0 & 0 & 0 & 0 & R_6 & R_7 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 1 \end{pmatrix} \begin{pmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ i_6 \\ i_7 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -V_1 \\ -V_1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Defining a set of equations involving loop voltages and node currents will allow to determine the different loop currents:

$$\begin{cases} i_1 R_1 + i_3 R_3 = 0 \\ i_2 R_2 + i_4 R_4 = 0 \\ i_3 R_3 + i_4 R_4 = -V_1 \\ i_5 R_5 + i_6 R_6 = -V_1 \\ i_6 R_6 + i_7 R_7 = 0 \\ i_1 - i_2 - i_3 + i_4 = 0 \\ -i_5 + i_6 - i_7 = 0 \end{cases}$$

Systems of linear equations

A system of algebraic equations has the form:

$$A_{11}x_1 + A_{12}x_2 + \cdots + A_{1n}x_n = b_1$$
$$A_{21}x_1 + A_{22}x_2 + \cdots + A_{2n}x_n = b_2$$
$$\vdots$$
$$A_{n1}x_1 + A_{n2}x_2 + \cdots + A_{nn}x_n = b_n$$

where both the coefficients $\mathbf{A_{ij}}$ and the constants $\mathbf{b_j}$ are known and $\mathbf{x_i}$ represent the unknowns

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

Conditioning

- A system of $\mathbf{n}$ linear equations with $\mathbf{n}$ unknowns has a unique solution if the determinant is *nonsingular*: $|\mathbf{A}| \neq \mathbf{0}$
  a nonsingular matrix has all rows and columns independent (not linear combination)
- what happens when the determinant of the coeff matrix is small?
  coeff matrix "almost"singular!

we can compare the **determinant** ($|\mathbf{A}|$) with **norm** ($||\mathbf{A}||$) of the matrix,

$|\mathbf{A}| << ||\mathbf{A}||$

The norm of the matrix (several definitions):

$||\mathbf{A}|| = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij}^2}$

$||\mathbf{A}|| = \max_{1 \leq i \leq n} \sum_{i=1}^{n} \left|A_{ij}\right| = max\,(|A_{11}| + |A_{21}| + \cdots, |A_{12}| + |A_{22}| + \cdots, |A_{13}| + |A_{23}| + \cdots)$

Conditioning (cont.)

- ✔ In most cases it is sufficient compare the determinant with the magnitudes of the matrix elements
- ✔ An "almost"singular coefficient matrix (small determinant) will define a ill-conditioned system

  their numerical solutions are not precise

  small changes in the coefficient matrix will imply large solutions variations

ill-conditioned system

Suppose the system:

$$\begin{cases} 2x + y & = 3 \\ 2x + 1.001y & = 0 \end{cases}$$

Solution:

$$\begin{cases} x & = 1501.5 \\ y & = -3000 \end{cases}$$

Determinant:

$$|\mathbf{A}| = \mathbf{0.002}$$

The system is ill-conditioned since $|\mathbf{A}|$ is much smaller than the norm of the coefficients matrix $\mathbf{A}$ or more simple, than the coefficients of the matrix.

To verify the ill-conditioning of the system just change by 0.1% the value 1.001 and check the new result:

$$\begin{cases} 2x + y & = 3 \\ 2x + \underbrace{\mathbf{1.002}}_{1+\delta}\, y & = 0 \end{cases} \Rightarrow \begin{cases} x & = 3\left(\frac{1}{2} + \frac{1}{\delta}\right) \\ y & = -\frac{3}{\delta} \end{cases}$$

The solutions of ill-conditioned cannot be trusted because round-off errors during computation can change completely the solution!

System of linear eqs: solving

- ✔ Systems of linear algebraic equations can be solved with **direct** and **iterative** methods
- ✔ Direct methods transform original eqs into equivalent eqs
  - ☞ equivalent eqs have the same solution
  - ☞ matrix determinant may change
- ✔ Elementary operations that leave the solution unchanged are:
  - ☞ exchanging equations
    changes sign of determinant $\left|\mathbf{A}'\right| = -|\mathbf{A}|$
  - ☞ multiply equation by nonzero constant λ
    determinant changes $\left|\mathbf{A}'\right| = \lambda|\mathbf{A}|$
  - ☞ multiply equation by nonzero constant and subtract it from another equation
    determinant remains unchanged $\left|\mathbf{A}'\right| = |\mathbf{A}|$

Solving (cont.)

Method	Initial form	Final form
Gauss elimination	$\mathbf{A}\ \mathbf{x} = \mathbf{b}$	$\mathbf{U}\ \mathbf{x} = \mathbf{c}$
LU decomposition	$\mathbf{A}\ \mathbf{x} = \mathbf{b}$	$\mathbf{LU}\ \mathbf{x} = \mathbf{b}$
Gauss-Jordan elimination	$\mathbf{A}\ \mathbf{x} = \mathbf{b}$	$\mathbf{I}\ \mathbf{x} = \mathbf{c}$

$\mathbf{A}$ ≡ matrix of coefficients

$\mathbf{U}$ ≡ upper triangular matrix

$\mathbf{L}$ ≡ lower triangular matrix

$\mathbf{I}$ ≡ identity matrix

$$\mathbf{U} = \begin{pmatrix} U_{11} & U_{12} & U_{13} & \cdots & U_{1n} \\ 0 & U_{22} & U_{23} & \cdots & U_{2n} \\ 0 & 0 & U_{33} & \cdots & U_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \vdots & U_{nn} \end{pmatrix}$$

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \vdots & 1 \end{pmatrix}$$

$$\mathbf{L} = \begin{pmatrix} L_{11} & 0 & 0 & \cdots & 0 \\ L_{21} & L_{22} & 0 & \cdots & 0 \\ L_{31} & L_{32} & L_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ L_{n1} & L_{32} & L_{n3} & \vdots & L_{nn} \end{pmatrix}$$

Gauss elimination

- Solves the system in two steps: elimination phase and solution phase
- The elimination phase transforms the equation $\mathbf{A}\mathbf{x} = \mathbf{b}$ into $\mathbf{U}\mathbf{x} = \mathbf{c}$
  - a *pivot* equation (i) is multiplied by a constant λ and subtracted to another one (j)

$$\text{Row}_j - \lambda_{ij} \times \text{Row}_i \rightarrow \text{Row}_j$$

- The equations are then solved by back substitution
- *Note: the determinant of a triangular matrix (U or L) is easy to compute:* $|\mathbf{A}| = |\mathbf{U}| = U_{11} \times U_{22} \times \cdots \times U_{nn}$

The augmented coefficient matrix is very convenient for making the computations

$$\left(\begin{array}{ccccc|c} A_{11} & A_{12} & A_{13} & \cdots & A_{1n} & b_1 \\ A_{21} & A_{22} & A_{23} & \cdots & A_{2n} & b_2 \\ A_{31} & A_{32} & A_{33} & \cdots & A_{3n} & b_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{n1} & A_{n2} & A_{n3} & \vdots & A_{nn} & b_n \end{array}\right)$$

As example, to transform $\mathbf{Row}_2$ we have to multiply the *pivot* line (here $\mathbf{Row}_1$ by hypothesis...you can interchange rows!) by:

$$\lambda_{12} = \frac{A_{21}}{A_{11}}$$

Gauss elimination: example

$$\begin{cases} 4x_1 - 2x_2 + x_3 = 11 & (1) \\ -2x_1 + 4x_2 - 2x_3 = -16 & (2) \\ x_1 - 2x_2 + 4x_3 = 17 & (3) \end{cases}$$

$$\left(\begin{array}{ccc|c} 4 & -2 & 1 & 11 \\ -2 & 4 & -2 & -16 \\ 1 & -2 & 4 & 17 \end{array}\right)$$

$$\left(\begin{array}{ccc|c} 4 & -2 & 1 & 11 \\ -2 & 4 & -2 & -16 \\ 1 & -2 & 4 & 17 \end{array}\right)$$

(1) pivot line
$(2) - (1) \times \frac{-2}{4}$
$(3) - (1) \times \frac{1}{4}$

$$\left(\begin{array}{ccc|c} 4 & -2 & 1 & 11 \\ 0 & 3 & -3/2 & -21/2 \\ 0 & -3/2 & 15/4 & 17 - 11/4 \end{array}\right)$$

$$\left(\begin{array}{ccc|c} 4 & -2 & 1 & 11 \\ 0 & 3 & -3/2 & -10.5 \\ 0 & 0 & 3 & 9 \end{array}\right)$$

The determinant of the coefficient matrix remains unchanged after the **elimination phase**

$$|A| = 4 \times 3 \times 3 = 36$$

Gauss elimination algorithm

Let's suppose we already transformed our matrix up to row $k = 3$ $(k = 1, \cdots, n)$

It means, $\mathbf{Row}_{k=3}$ is now the ***pivot* line** and all equations below (Row $>$ 3) are still to be transformed

$$\left(\begin{array}{ccccc|c} A_{11} & A_{12} & A_{13} & \cdots & A_{1n} & b_1 \\ 0 & A_{22} & A_{23} & \cdots & A_{2n} & b_2 \\ 0 & 0 & A_{33} & \cdots & A_{3n} & b_3 \\ \hline 0 & 0 & A_{i3} & \vdots & A_{in} & b_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & A_{n3} & \vdots & A_{nn} & b_n \end{array}\right)$$

To eliminate the element A_{i3} of the row below the *pivot (row* k*)* we do:

$$\text{Row}_i - \lambda_{ki} \times \text{Row}_{pivot} \longrightarrow \text{Row}_i$$

$$\lambda_{ki} = A_{i3}/A_{k3}$$

```
// matrix n x n
loop on pivot row (k): k = 0, n-2
  loop on rows below pivot: i = k+1, n-1
     - for every row:
       compute lambda A(i,k)/A(k,k)
     - transform row i:
       only elements (i, k+1:n)
       need to be stored
     - transform also constant value
```


Back substitution phase

✔ After Gauss elimination we got an equation involving an upper triangular matrix $\mathbf{U}$: $\mathbf{Ux} = \mathbf{c}$

$$\left(\begin{array}{ccccc|c} A_{11} & A_{12} & A_{13} & \cdots & A_{1n} & c_1 \\ 0 & A_{22} & A_{23} & \cdots & A_{2n} & c_2 \\ 0 & 0 & A_{33} & \cdots & A_{3n} & c_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \vdots & A_{nn} & c_n \end{array}\right)$$

✔ Now, we need to solve equations by starting on the simplest one (the last row) and going back

The solutions:

1) $A_{nn} x_n = c_n \quad \Rightarrow x_n = c_n / A_{nn}$

k) $A_{kk} x_k + A_{k,k+1} x_{k+1} + \cdots + A_{kn} x_n = c_k \quad \Rightarrow x_k = \frac{1}{A_{kk}} \left(c_k - \sum_{j=k+1}^{n} A_{kj} x_j \right)$

Back substitution algorithm

```
// Let's suppose we are solving a system involving a coeff matrix A(3x3)
const int n=3;

// solution array
double x[n];
                                              x2 = b2/A22
                                              x1 = (b1 - A12 x2)/A11
                                              x0 = (b0 - A01 x1 - A02 x2)/A00

// loop on rows from end to begin
for (int k=n-1; k>=0; k--) {
  double sumX=0;

  // scan values of a row from diagonal to the right
  for (int j=k+1; j<n; j++) {
     sumX += x[j]*A[k][j]
  }

  // compute solution values
  x[k] = (b[k] - sumX)/A[k][k];
}
```


Pivoting

- ✔ If the element of the *pivot* row and column being used to transform subsequent rows is zero, just reorder the equations by moving the pivot row to the end of the matrix
- ✔ Reordering of the equations may also be needed if the pivot element, although different from zero, is very small

$$[\mathbf{A}|\mathbf{b}] = \left(\begin{array}{ccc|c} \delta & -1 & 1 & 0 \\ -1 & 2 & -1 & 0 \\ 2 & -1 & 0 & 1 \end{array}\right)$$

$$[\mathbf{A}^{'}|\mathbf{b}^{'}] = \left(\begin{array}{ccc|c} \delta & -1 & 1 & 0 \\ 0 & 2-1/\delta & -1+1/\delta & 0 \\ 0 & -1+2/\delta & 2/\delta & 1 \end{array}\right)$$

$$[\mathbf{A}^{'}|\mathbf{b}^{'}] \simeq \left(\begin{array}{ccc|c} \delta & -1 & 1 & 0 \\ 0 & -1/\delta & +1/\delta & 0 \\ 0 & +2/\delta & -2/\delta & 1 \end{array}\right)$$

Notice that after approximation the two last equations contradict each other!

Pivoting with reordering

The augmented coeff matrix

$$[\mathbf{A}|\mathbf{b}] = \left(\begin{array}{ccc|c} \delta & \mathbf{-1} & \mathbf{1} & \mathbf{0} \\ -1 & 2 & -1 & 0 \\ 2 & -1 & 0 & 1 \end{array}\right)$$

$\text{Row}_1 \leftrightarrow \text{Row}_2$

$$[\mathbf{A}|\mathbf{b}] = \left(\begin{array}{ccc|c} -1 & 2 & -1 & 0 \\ \delta & \mathbf{-1} & \mathbf{1} & \mathbf{0} \\ 2 & -1 & 0 & 1 \end{array}\right)$$

$\text{Row}_2 - (-\delta) \times \text{Row}_1 \rightarrow \text{Row}_2$

$\text{Row}_3 - (-2) \times \text{Row}_1 \rightarrow \text{Row}_3$

$$[\mathbf{A}'|\mathbf{b}'] = \left(\begin{array}{ccc|c} -1 & 2 & -1 & 0 \\ 0 & -1+2\delta & +1-\delta & 0 \\ 0 & 3 & -2 & 1 \end{array}\right)$$

$\text{Row}_3 - 3/(-1+2\delta) \times \text{Row}_2 \rightarrow \text{Row}_3$

$$[\mathbf{A}'|\mathbf{b}'] = \left(\begin{array}{ccc|c} -1 & 2 & -1 & 0 \\ 0 & -1+2\delta & +1-\delta & 0 \\ 0 & 0 & -\frac{1+\delta}{2\delta-1} & 1 \end{array}\right)$$

$$[\mathbf{A}'|\mathbf{b}'] \simeq \left(\begin{array}{ccc|c} -1 & 2 & -1 & 0 \\ 0 & -1 & +1 & 0 \\ 0 & 0 & 1 & 1 \end{array}\right)$$

Back substitution gives:

$x_3 = 1$

$x_2 = x_3 = 1$

$x_1 = 2x_2 - x_3 = 1$

Diagonal dominance

- ✔ A matrix $\mathbf{A}$ $n \times n$ is said to be **diagonally dominant** if each diagonal element is larger in absolute than the sum of the other elements on the same row

$$|\mathbf{A_{ii}}| > \sum_{\substack{j=1 \\ i \neq j}}^{n} |A_{ij}| \qquad i = 1, 2, ..., n$$

- ✔ If the coefficient matrix of the equation system $\mathbf{Ax} = \mathbf{b}$ is diagonally dominant, it means the equations are already arranged in a optimal order
  - ☞ the strategy shall be to reorder the coefficient matrix in order to get diagonal dominance approach
  - ☞ the pivot element shall be as large as possible when compared to other elements in the pivot row

scaled row pivoting

Let's make the pivot matrix element as large as possible.
Let's define s_i as the **scale factor** of row **i** corresponding to the absolute value of the largest element in *ith row*:

$$s_i = \max_j |A_{ij}|$$

Next, we can define the **relative size** of an element A_{ij} in the row **i** of the matrix $\mathbf{A}$:

$$r_{ij} = \frac{|A_{ij}|}{s_i}$$

Gauss elimination with pivoting

algorithm

```
1) store the maximum absolute value of every row on array s(i)

2) loop on rows i = 0, n-1

   - we are dealing with "pivot candidate" A(i,i);
     we are NOT going to accept it automatically...

   - check if pivot element A(i,i) is the best one by looking to all
     elements from the same column and below the pivot candidate;
     choose the one with the largest relative size

   - identify the row with largest relative size element

   - if different from the pivot row candidate, swap those rows and
     swap [s] elements
     void SwapRows(int i, int j, double *s);

   - if largest relative size element is very small (<tolerance)
     the matrix is singular => PROBLEM WITH THE LINEAR SYSTEM!!!

   - proceed with elimination phase
```


Gauss elimination with pivoting

Example

The coeff matrix

$$[\mathbf{A}] = \begin{pmatrix} 2 & -2 & 6 \\ -2 & 4 & 3 \\ -1 & 8 & 4 \end{pmatrix}$$

The vector of constants

$$[\mathbf{b}] = \begin{pmatrix} 16 \\ 0 \\ -1 \end{pmatrix}$$

The augmented coeff matrix and the vector of max row values

$$[\mathbf{A}] = \left(\begin{array}{ccc|c} 2 & -2 & 6 & 16 \\ -2 & 4 & 3 & 0 \\ -1 & 8 & 4 & -1 \end{array}\right) \qquad [\mathbf{s}] = \begin{pmatrix} 6 \\ 4 \\ 8 \end{pmatrix}$$

LU decomposition

- Any square matrix $\mathbf{A}$ can be expressed as the product of a lower triangular matrix $\mathbf{L}$ and an upper trinagular matrix $\mathbf{U}$

$$\boxed{\mathbf{A} = \mathbf{L}\,\mathbf{U}}$$

  - *the computation of $\mathbf{L}$ and $\mathbf{U}$ is known as LU decomposition or $\mathbf{LU}$ factorization*
  - *the factorization is not unique unless constraints on L and U are applied*

- common decompositions:

Decomposition	Constraints
Doolittle	$L_{ii} = 1$ with $i = 1, 2, ..., n$
Crout	$U_{ii} = 1$ with $i = 1, 2, ..., n$
Choleski	$\mathbf{L} = \mathbf{U}^{\mathrm{T}}$

After decomposing $\mathbf{A}$:

$\mathbf{Ax} = \mathbf{b} \Rightarrow \mathbf{LUx} = \mathbf{b}$

We have:

$\mathbf{Ly} = \mathbf{b}$ with $(\mathbf{Ux} = \mathbf{y})$

Therefore: we start getting $\mathbf{y}$ and then $\mathbf{x}$

Doolittle decomposition

- Considere a 3×3 $\mathbf{A}$ matrix and the respective triangular lower and upper matrices $\mathbf{L}$ and $\mathbf{U}$

$$[\mathbf{A}] = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \quad [\mathbf{L}] = \begin{pmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{pmatrix} \quad [\mathbf{U}] = \begin{pmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{pmatrix}$$

- Making the operation: $\mathbf{A} = \mathbf{LU}$

$$[\mathbf{A}] = \begin{pmatrix} U_{11} & U_{12} & U_{13} \\ U_{11}L_{21} & U_{12}L_{21} + U_{22} & U_{13}L_{21} + U_{23} \\ U_{11}L_{31} & U_{12}L_{31} + U_{22}L_{32} & U_{13}L_{31} + U_{23}L_{32} + U_{33} \end{pmatrix}$$