Numerical Methods in Physics

Numerische Methoden in der Physik, 515.421.

Instructor: Ass. Prof. Dr. Lilia Boeri
Room: PH 03 090
Tel: +43-316-873 8191
Email Address: l.boeri@tugraz.at

Room: TDK Seminarraum

Time: 8:30-10 a.m.

Exercises: Computer Room, PH EG 004 F

http://itp.tugraz.at/LV/boeri/NUM_METH/index.html
(Lecture slides, Script, Exercises, etc).

TOPICS (this year):

- **Chapter 1: Introduction.**
- **Chapter 2: Numerical methods for the solution of linear inhomogeneous systems.**
- Chapter 3: Interpolation of point sets.
- **Chapter 4: Least-Squares Approximation.**
- **Chapter 5: Numerical solution of transcendental equations.**
- Chapter 6: Numerical Integration.
- Chapter 7: Eigenvalues and Eigenvectors of real matrices.
- **Chapter 8: Numerical Methods for the solution of ordinary differential equations: initial value problems.**
- Chapter 9: Numerical Methods for the solution of ordinary differential equations: marginal value problems.

TOPICS (this year):

- **Chapter 1: Introduction (8/10/2013).**
- **Chapter 2: Numerical methods for the solution of linear inhomogeneous systems.**
- Chapter 3: Interpolation of point sets.
- **Chapter 4: Least-Squares Approximation.**
- **Chapter 5: Numerical solution of transcendental equations.**
- Chapter 6: Numerical Integration.
- Chapter 7: Eigenvalues and Eigenvectors of real matrices.
- **Chapter 8: Numerical Methods for the solution of ordinary differential equations: initial value problems.**
- Chapter 9: Numerical Methods for the solution of ordinary differential equations: marginal value problems.

Introduction (8/10/2013)

- Definitions and **basic concepts**: algorithm, machine precision, roundoff.
- **Errors**: absolute and relative error; sources of error.
- **Input Errors**: ill-conditioned problems.
- Sources of errors in the algorithms: **methodological errors**.
- Practical Examples of the interplay of methodological and **roundoff** errors:

  - Function Evaluation: how to avoid **subtractive cancellation**.
  - Numerical Differentiation: **optimal stepsize**.
  - Taylor series and continued fraction.
  - Recursive Algorithms (**stability**).

TOPICS (this year):

- **Chapter 1: Introduction.**
- **Chapter 2: Numerical methods for the solution of linear inhomogeneous systems.**
- Chapter 3: Interpolation of point sets.
- **Chapter 4: Least-Squares Approximation.**
- **Chapter 5: Numerical solution of transcendental equations.**
- Chapter 6: Numerical Integration.
- Chapter 7: Eigenvalues and Eigenvectors of real matrices.
- **Chapter 8: Numerical Methods for the solution of ordinary differential equations: initial value problems.**
- Chapter 9: Numerical Methods for the solution of ordinary differential equations: marginal value problems.

SCRIPT

Chapter 2: Numerical methods for the solution of linear inhomogeneous systems.

Today (15/10/2013)

- **Linear Systems:** Definition and applications.
- Solution: **Direct** vs iterative methods.
- Gaussian methods: **LU decomposition**.
- A **practical** example: 3x3 square matrix (step-by-step).
- Strategies to reduce the error: **partial pivoting**.
- Stability of the solution: Condition numbers.
- **Programs** for the LU decomposition: **LUDCMP** and **LUBKSB**.
- How to **use** the programs for LU decompositions (inhomogeneous systems, matrix inversion, determinant of a matrix).

Linear Systems

Definitions:

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$$

$$\cdots$$

$$a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n$$

with:

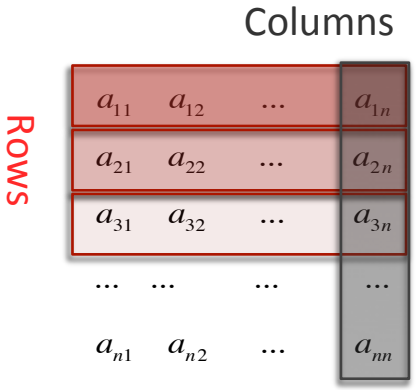
$$|b_1| + |b_2| + \ldots |b_n| \neq 0$$

Inhomogeneous **linear** system of equations.

Linear Systems

Matrix notation:

$$\hat{A}\mathbf{x} = \mathbf{b}$$



$\hat{A} = (a)_{ij}$ **Matrix of coefficients**

$\mathbf{x} = (x_1, \ldots, x_n)$ **Solution** vector

$\mathbf{b} = (b_1, \ldots, b_n)$ **Inhomogeneous** vector

$\det(\hat{A}) \neq 0$ The problem is non-singular and admits a solution

Methods of solution (analytical): Cramer's rule.

Given the inhomogeneous non-linear system:

$$\hat{A}\mathbf{x} = \mathbf{b}$$

The components of the solution vector are given by:

$$x_i = \frac{\det(A_i)}{\det(A)}$$

The A_i'*s* are obtained by replacing the i^{th} column of **A** by the column vector **b**.

The Cramer's rule is hard to implement on a computer,

and practically unusable for n≥4.

Methods of Solution (numerical):

- **Direct Methods:**

No methodological error, BUT computationally expensive; **roundoff** errors can be **large**.

LU decomposition (Doolittle and Crout) (*Gaussian Elimination*).

- **Iterative Methods:**

Simple algorithms; roundoff is easily controlled. The solution is approximate (truncation).

Gauss-Seidel method.

Gaussian elimination: The gaussian elimination method solves a linear set of equations reducing the matrix of coefficients to a is used to an **upper triangular matrix**, through **a sequence** of **elementary row operations**:

1) Swapping two rows.

2) Multiplying a row by a non-zero number

3) Adding a multiple of one row to another row**.**

$$\hat{A}\mathbf{x} = \mathbf{b} \quad \Rightarrow \quad \hat{U}\mathbf{x} = \mathbf{y}$$

U is a **triangular matrix** with the property:

$$U = (u_{ij}) \quad \text{, with:} \quad u_{ij} = 0, \qquad i > j$$

Systems with **triangular matrices** are easy to solve with **back-substitution**.

LU decomposition (*Doolittle and Crout*): Reformulation of the Gaussian elimination.

A real matrix can always be represented as the product of two real matrices **L** and **U**, *i.e.*

$$\hat{A} = \hat{L} \cdot \hat{U}$$

U: Upper triangular matrix

$$\hat{U} = \begin{pmatrix} u_{11} & u_{12} & \dots & \dots & u_{1n} \\ 0 & u_{22} & \dots & \dots & u_{2n} \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ 0 & 0 & \dots & \dots & u_{nn} \end{pmatrix}$$

L: Lower triangular matrix

$$\hat{L} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ m_{21} & 1 & \dots & 0 & 0 \\ m_{31} & m_{32} & 1 & \dots & 0 \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ m_{n1} & m_{n2} & m_{n3} & \dots & 1 \end{pmatrix}$$

Step 1: **LU decomposition of A.**

$$\hat{U} = \begin{pmatrix} u_{11} & u_{12} & \dots & \dots & u_{1n} \\ 0 & u_{22} & \dots & \dots & u_{2n} \\ . & & & & \\ . & & & & \\ . & & & & \\ 0 & 0 & \dots & \dots & u_{nn} \end{pmatrix}$$

$$\hat{L} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ m_{21} & 1 & \dots & 0 & 0 \\ m_{31} & m_{32} & 1 & \dots & 0 \\ . & & & & . \\ . & & & & . \\ m_{n1} & m_{n2} & m_{n3} & \dots & 1 \end{pmatrix}$$

i

j

Expression of the coefficients of the LU matrices:

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} m_{ik} u_{kj} \qquad i = 1, \dots, j-1$$

$$\gamma_{ij} = a_{ij} - \sum_{k=1}^{j-1} m_{ik} u_{kj} \qquad i = j, \dots, N$$

$$u_{jj} = \gamma_{jj}$$

$$m_{ij} = \frac{\gamma_{ij}}{\gamma_{jj}}$$

i=row

j=column

Memory-efficient storage of the LU decomposition: We do not need to store the diagonal elements of L, since mii=1.

All relevant coefficients can then be fitted into an *nxn* matrix: **LU-matrix.**

$$LU = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ m_{21} & u_{22} & u_{23} & \dots & u_{2n} \\ m_{31} & m_{32} & u_{33} & \dots & u_{3n} \\ \dots & & & & \dots \\ \dots & & & & \dots \\ 0 & 0 & \dots & \dots & u_{nn} \end{pmatrix}$$

All relevant coefficients can then be fitted into an *nxn* matrix: the **LU-matrix.**

Step 2: **Finding the solution of the auxiliary system.**

$$\hat{A}\mathbf{x} = \mathbf{b} \quad \Rightarrow \quad \hat{L} \cdot \mathbf{y} = \mathbf{b}$$

Using:

$$\hat{A} \cdot \mathbf{x} \equiv \hat{L} \cdot \hat{U} \cdot \mathbf{x} = \hat{L} \cdot (\hat{U} \cdot \mathbf{x}) = \mathbf{b}$$

$$\hat{U} \cdot \mathbf{x} \equiv \mathbf{y}$$

The **auxiliary system** L**y** =**b** is easy to solve through ***forward substitution***:

$$y_1 = b_1$$

$$y_i = b_i - \sum_{j=1}^{i-1} m_{ij} y_j, \qquad i = 2,3,\ldots,n$$

To obtain the auxiliary vector **y**: U**x**=**y** we can employ ***backward substitution***:

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

$$x_i = \frac{1}{u_{ii}}\left[y_i - \sum_{j=i+1}^{n} u_{ij} x_j \right], \qquad i = n-1, n-2, \ldots, 1$$

LU Decomposition:

In synthesis solving a **linear system** with **LU decomposition** is a **2-step** procedure:

1) Decompose the original matrix of coefficients A into an upper and a lower triangular matrix (U and L).

2) Solve the two auxiliary systems U**x**=**y** and L**y**=**b** with backward and forward substitution.

We need two programs!

LU decomposition in practice: 3 x 3 matrix

$$\begin{array}{c} \\ \text{i=1} \\ \text{i=2} \\ \text{i=3} \end{array} \begin{array}{c} \begin{array}{ccc} \text{j=1} & \text{j=2} & \text{j=3} \end{array} \\ \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \end{array}$$

I want to apply the formulas for the **LU decomposition** (column-by-column).

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} m_{ik} u_{kj} \qquad i = 1, \ldots, j-1$$

$$\gamma_{ij} = a_{ij} - \sum_{k=1}^{j-1} m_{ik} u_{kj} \qquad i = j, \ldots, N$$

$$u_{jj} = \gamma_{jj}$$

$$m_{ij} = \frac{\gamma_{ij}}{\gamma_{jj}}$$

First column: j=1

First row: i=1

$$u_{11} = a_{11}, \qquad \gamma_{11} = a_{11}, \qquad m_{11} = \frac{a_{11}}{a_{11}} = 1$$

Second row: i=2

$$\gamma_{21} = a_{21}$$

$$m_{21} = \frac{\gamma_{21}}{\gamma_{11}}$$

$$u_{21} = a_{21} - m_{21}u_{11} = a_{21} - \frac{\gamma_{21}}{\gamma_{11}}u_{11} = a_{21} - a_{21} = 0$$

Third row: i=3

$$\gamma_{31} = a_{31}$$

$$m_{31} = \frac{\gamma_{31}}{\gamma_{11}}$$

$$u_{31} = a_{31} - m_{32}u_{21} - m_{31}u_{11} = a_{31} - \frac{\gamma_{31}}{\gamma_{11}}u_{11} = a_{31} - a_{31} = 0$$

	j=1	j=2	j=3
i=1	a_{11}	a_{12}	a_{13}
i=2	m_{21}	a_{22}	a_{23}
i=3	m_{31}	a_{32}	a_{33}

Second column: j=2

First row: i=1

$$u_{12} = a_{12},$$

$$\gamma_{12} = a_{12} - m_{11}u_{12} = 0 = m_{12}$$

Second row: i=2

$$u_{22} = \gamma_{22} = a_{22} - m_{21}u_{12}$$

$$m_{22} = \frac{\gamma_{22}}{\gamma_{22}} = 1$$

	j=1	j=2	j=3
i=1	u_{11}	u_{12}	a_{13}
i=2	m_{21}	u_{22}	a_{23}
i=3	m_{31}	m_{32}	a_{33}

Third row: i=3

$$\gamma_{32} = a_{32} - m_{31}u_{12}$$

$$m_{32} = \frac{\gamma_{32}}{\gamma_{22}}$$

$$u_{32} = a_{32} - m_{31}u_{12} - m_{32}u_{22} =$$

$$= a_{32} - m_{31}u_{12} - \frac{\gamma_{32}}{\gamma_{22}}u_{22} = a_{32} - m_{31}u_{12} - a_{32} + m_{31}u_{12} = 0$$

Third column: j=3

First row: i=1

$$u_{13} = a_{13}$$

Second row: i=2

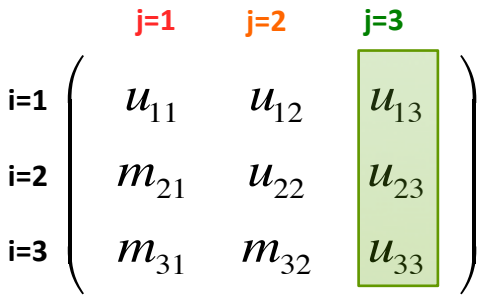
$$u_{23} = a_{23} - m_{21}u_{13}$$

$$\gamma_{23} = 0$$

Third row: i=3

$$\gamma_{33} = a_{33} - m_{31}u_{13} - m_{32}u_{23}$$

$$u_{33} = \gamma_{33}$$



Roundoff error in LU decomposition:

$$x_1 + 5923181x_2 + 1608x_3 = 5924790$$
$$5923181x_1 + 337116x_2 - 7x_3 = 6260290$$
$$6114x_1 + 2x_2 + 9101372x_3 = 9107488$$

a_{11}=1; a_{12}=5923181; a_{13}=1608; b_1=5924790

a_{21}=5923181; a_{22}=337116; a_{23}=-7; b_2=6260290

a_{31}=6114; a_{32}=2; a_{33}=9101372; b_3=9107488

Exact solution: x_1=x_2=x_3=1.

Solving with LU decomposition can lead to severe errors due to roundoff!

Roundoff error in LU decomposition:

$a_{11}=1$; $a_{12}=5923181$; $a_{13}=1608$; $b_1=5924790$

$a_{21}=5923181$; $a_{22}=337116$; $a_{23}=-7$; $b_2=6260290$

$a_{31}=6114$; $a_{32}=2$; $a_{33}=9101372$; $b_3=9107488$

LU matrix (all digits):

$u_{11}=a_{11}=1$; $u_{12}=5923181$; $u_{13}=1608$

$m_{21}=5923181$; $u_{22}=-350840728216 45$; $u_{23}=-9524475048$

$m_{31}=6114$; $m_{32}=0.00103221563859$; $u_{33}=9101372$

LU matrix (7 significant digits, single precision):

0.1000000E+01	0.5923181E+07	0.1608000E+04
0.5923181E+07	-.3508407E+14	-.9524475E+10
0.6114000E+04	0.1032216E-02	0.9101371E+07

x:

0.9398398E+00
0.1000000E+01
0.9995983E+00

u_{22}=-35084072821645

u_{23}=-9524475048

m_{32}=0.00103221563859

Big error on the solution due to roundoff error!

Partial Pivoting

How to reduce the roundoff error in the LU decomposition?

Partial Pivoting: Reordering the order of the equations of the linear system (exchanging rows) leads to a substantial improvement of the solution!

2/3/1:

	0.5923181E+07	0.3371160E+06	-.7000000E+01
LU-Matrix:	0.1032216E-02	-.3459764E+03	0.9101372E+07
	0.1688282E-06	-.1712019E+05	0.1558172E+12

	0.9999961E+00
x:	0.1000068E+01
	0.1000000E+01

2/1/3:

	0.5923181E+07	0.3371160E+06	-.7000000E+01
LU-Matrix:	0.1688282E-06	0.5923181E+07	0.1608000E+04
	0.1032216E-02	-.5841058E-04	0.9101372E+07

	0.1000000E+01
x:	0.1000000E+01
	0.1000000E+01

EXACT!!!!

What is the optimal choice of the order of the equations in the linear system?

0.1000000E+01	0.5923181E+07	0.1608000E+04	
0.5923181E+07	-.3508407E+14	-.9524475E+10	**BAD!**
0.6114000E+04	0.1032216E-02	0.9101371E+07	
0.5923181E+07	0.3371160E+06	-.7000000E+01	
0.1032216E-02	-.3459764E+03	0.9101372E+07	**BETTER!**
0.1688282E-06	-.1712019E+05	0.1558172E+12	
0.5923181E+07	0.3371160E+06	-.7000000E+01	
0.1688282E-06	0.5923181E+07	0.1608000E+04	**GOOD!**
0.1032216E-02	-.5841058E-04	0.9101372E+07	**(EXACT)**

The order is optimal when the m's are as small as possible…

How to achieve this in practice?

To minimize m_{ij} I have to maximize the γ_{ij}'s:

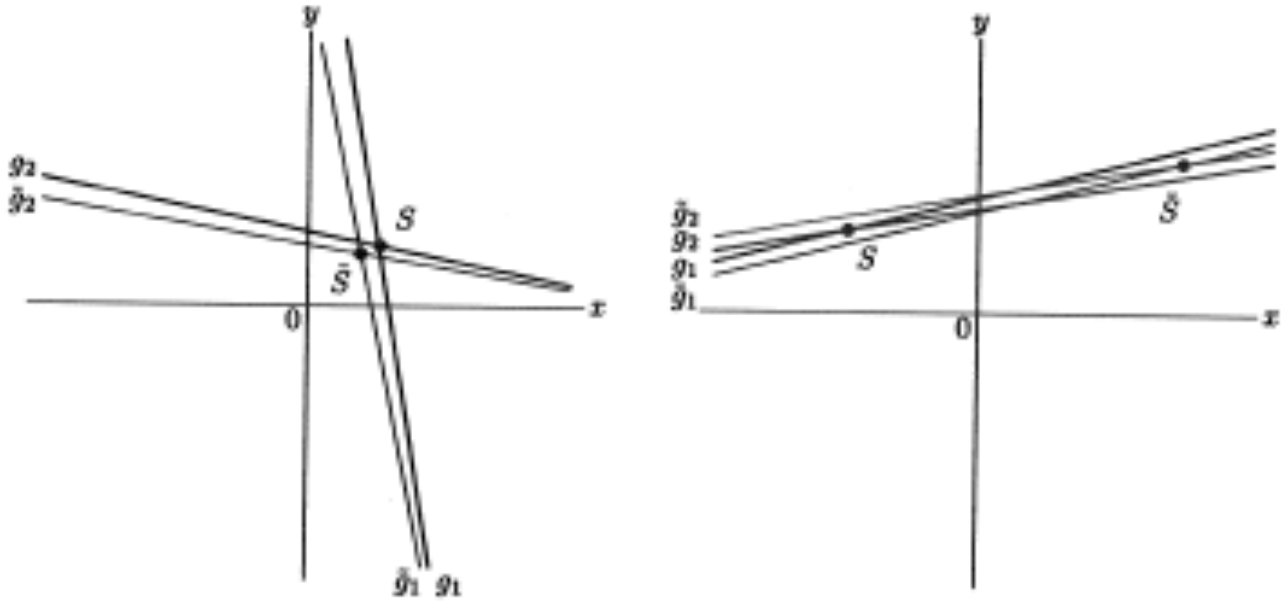
$$m_{ij} = \frac{\gamma_{ij}}{\gamma_{jj}}, \qquad \gamma_{ij} = a_{ij} - \sum_{k=1}^{j-1} m_{ik} u_{kj} \qquad i = j, \ldots, N$$

1) Determine the i-th row with the largest γ_{ij} and exchange this with the *j*-th row.

2) Evaluate the remaining coefficients of the LU decomposition.

Partial pivoting for the LU decomposition: **not** needed **for symmetrical, tridiagonal, cyclically tridiagonal, diagonal-dominant** or **generally positive** matrices of coefficients.

Ill-conditioned systems: In principle, the solution of a linear system with direct methods should coincide with the real solution. Except for the case of roundoff, this can happen also when the system is **ill-conditioned**.



Definition: A system is called **ill-conditioned** if small variations in the input parameters can lead to **large uncertainities** in the output (the system is intrinsically unstable numerically).

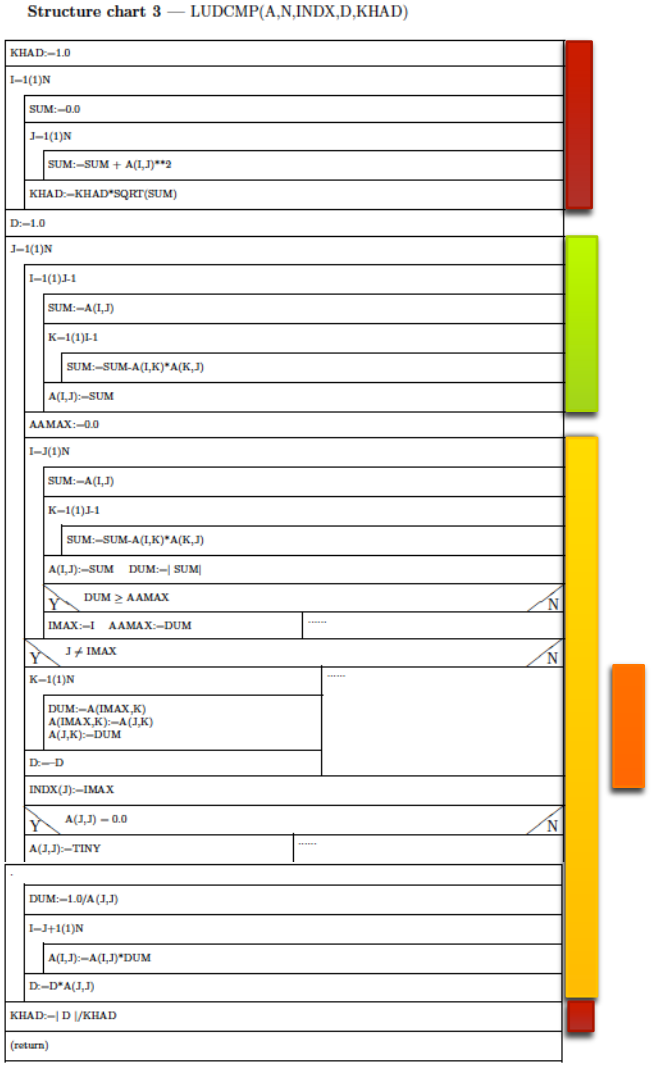
Condition Numbers are used to define whether a system is ill-conditioned.

Hadamard's condition number:

$$K_H(A) = \frac{|\det(A)|}{\alpha_1 \alpha_2 \ldots \alpha_n}, \qquad \alpha_i = \sqrt{a_{i1}^2 + a_{i2}^2 + \ldots + a_{in}^2}$$

$$\begin{cases} K_H(A) < 0.01 & \text{Ill-conditioned} \\ K_H(A) > 0.1 & \text{Well-conditioned} \\ 0.01 \le K_H(A) \le 0.1 & \text{Undefined} \end{cases}$$

LUDCMP: Program for LU decomposition



Hadamard's condition number

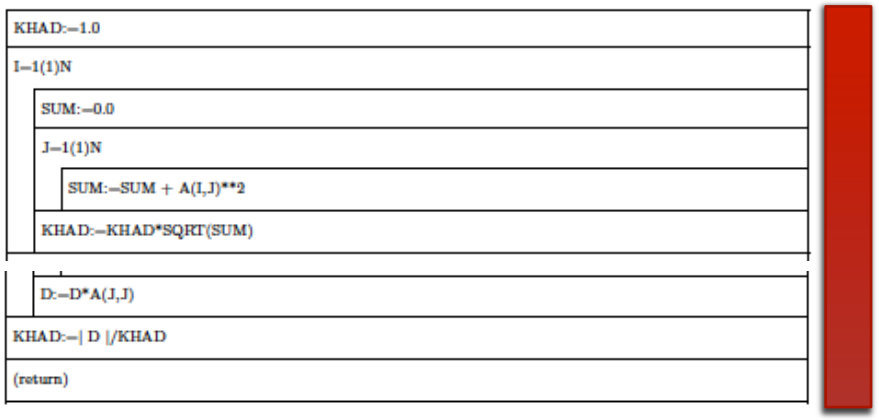
Upper triangular matrix (U)

Lower triangular matrix (L)

Partial pivoting

LUDCMP: Program for LU decomposition

Structure chart 3 — LUDCMP(A,N,INDX,D,KHAD)



Hadamard's condition number

$$K_H(A) = \frac{|\det(A)|}{\alpha_1 \alpha_2 \ldots \alpha_n},$$

$$\alpha_i = \sqrt{a_{i1}^2 + a_{i2}^2 + \ldots + a_{in}^2}$$

LUDCMP: Program for LU decomposition

```
J=1(1)N
  I=1(1)J-1
    SUM:=A(I,J)
    K=1(1)I-1
      SUM:=SUM-A(I,K)*A(K,J)
    A(I,J):=SUM
```

Upper triangular matrix (U)

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} m_{ik} u_{kj} \qquad i = 1, \ldots, j-1$$

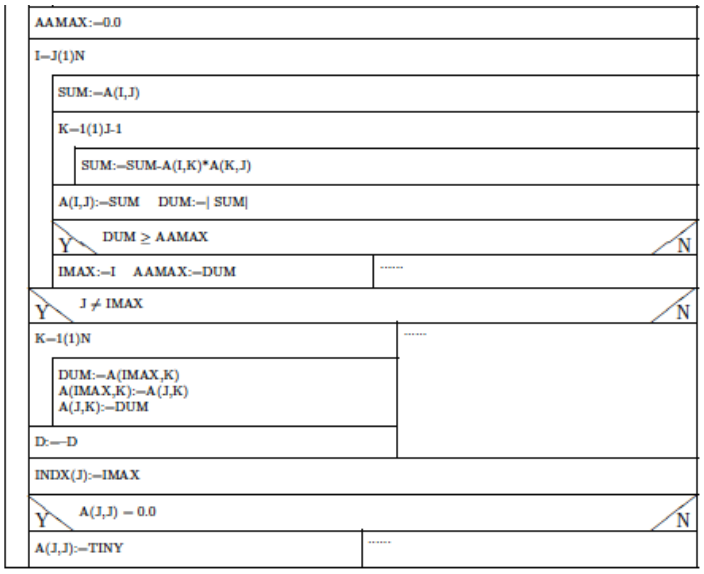
$$\gamma_{ij} = a_{ij} - \sum_{k=1}^{j-1} m_{ik} u_{kj} \qquad i = j, \ldots, N$$

$$u_{jj} = \gamma_{jj}$$

$$m_{ij} = \frac{\gamma_{ij}}{\gamma_{jj}}$$

j=1,…,n

i=1,…,j-1



LUDCMP: Program for LU decomposition

Partial pivoting

j=1,…,n

i=j,…,n

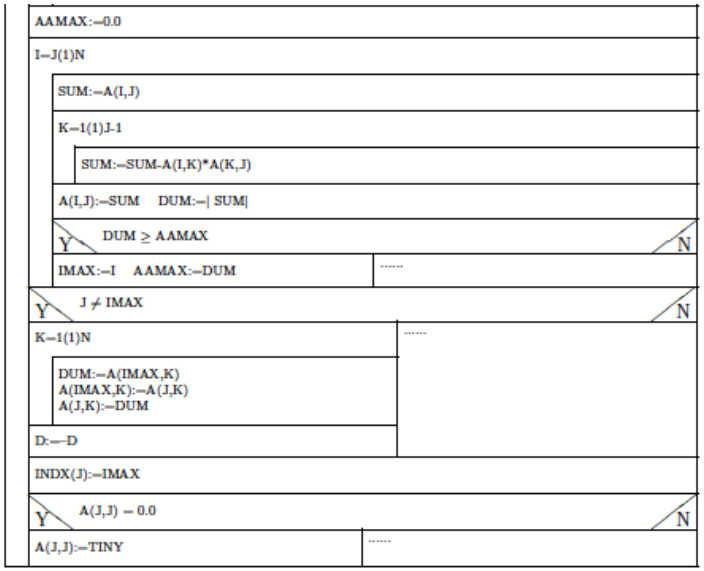
Lower triangular matrix (L)

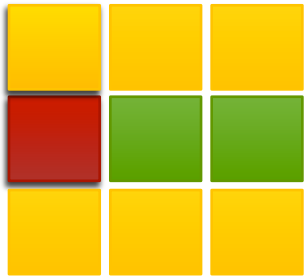
$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} m_{ik} u_{kj} \qquad i = 1, \ldots, j-1$$

$$\gamma_{ij} = a_{ij} - \sum_{k=1}^{j-1} m_{ik} u_{kj} \qquad i = j, \ldots, N$$

$$u_{jj} = \gamma_{jj}$$

$$m_{ij} = \frac{\gamma_{ij}}{\gamma_{jj}}$$





Partial Pivoting (example); 3x3 matrix

Partial Pivoting (example); 3x3 matrix

j=1; i=1,…3

i=1

sum=a(1,1)

Dum=|a(1,1)|

Imax=i=j AAMAX=a(1,1)

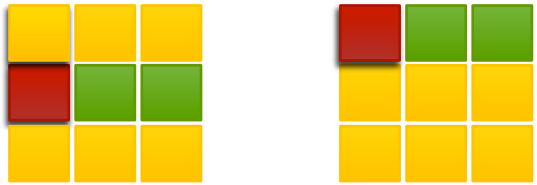
i=2

sum=a(2,1)

a(2,1)=sum

dum=|sum|

imax=2 Aamax=a(2,1)



j≠imax -> (1≠2)

k=1,3

k=1

dum=a(2,1)

a(2,1)=a(1,1)

a(1,1)=dum=a(2,1)

k=2

dum=a(1,2)

a(2,2)=a(1,2)

a(1,2)=dum=a(2,2)

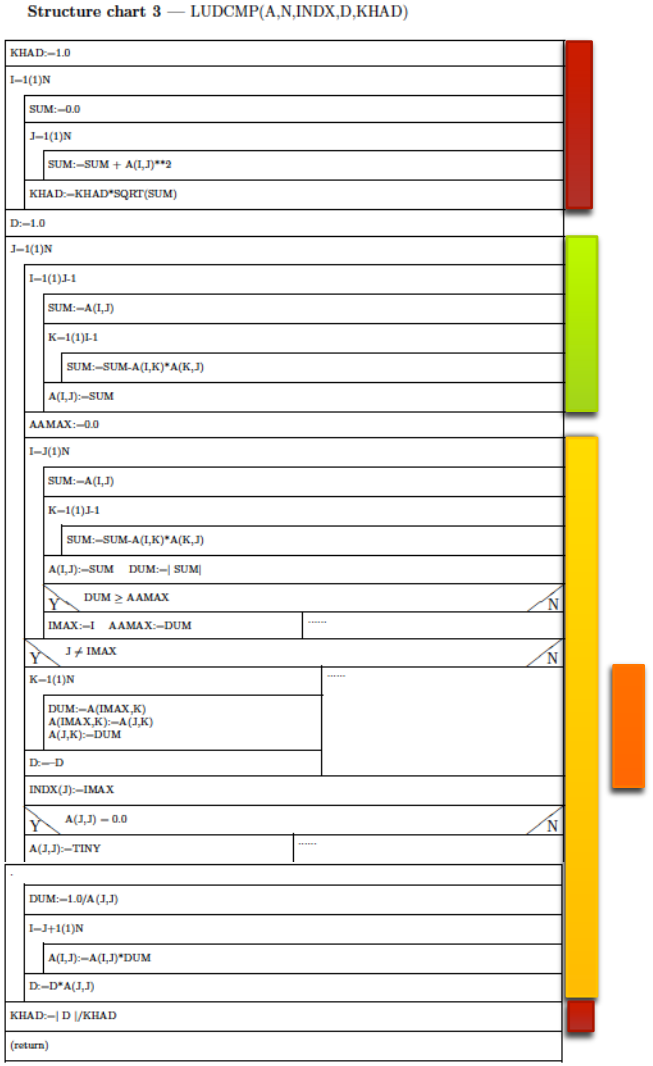
k=3

dum=a(2,3)

a(2,3)=a(1,3)

a(1,3)=dum=a(2,3)

LUDCMP: Program for LU decomposition



Hadamard's condition number

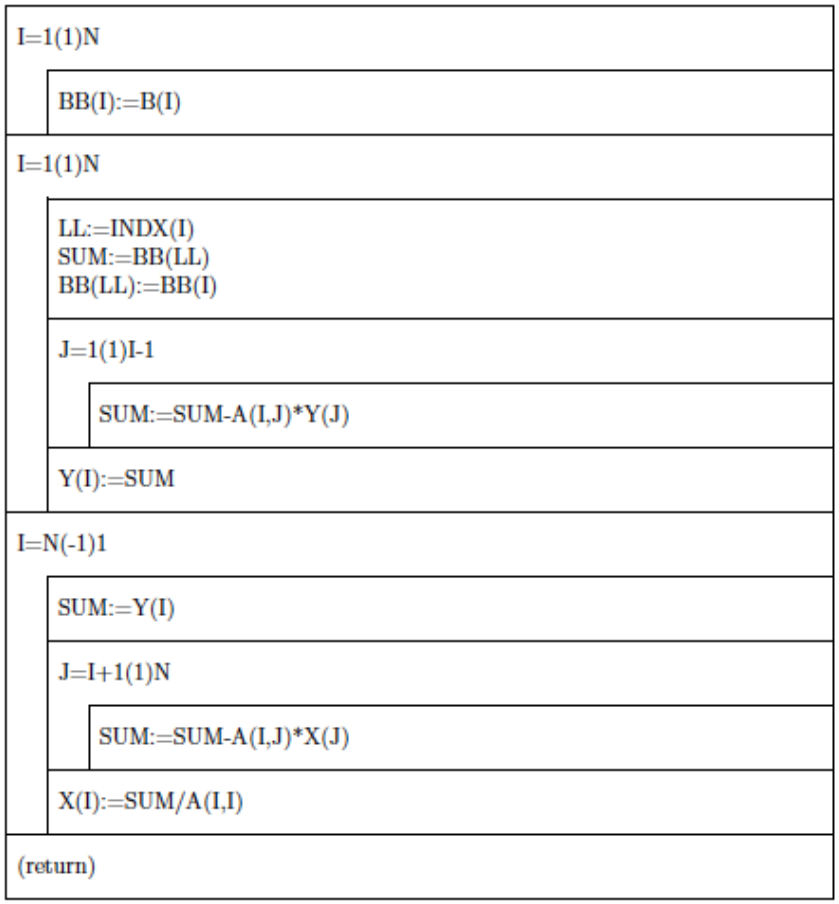
Upper triangular matrix (U)

Lower triangular matrix (L)

Partial pivoting

LUBKSB: Program for LU back substitution

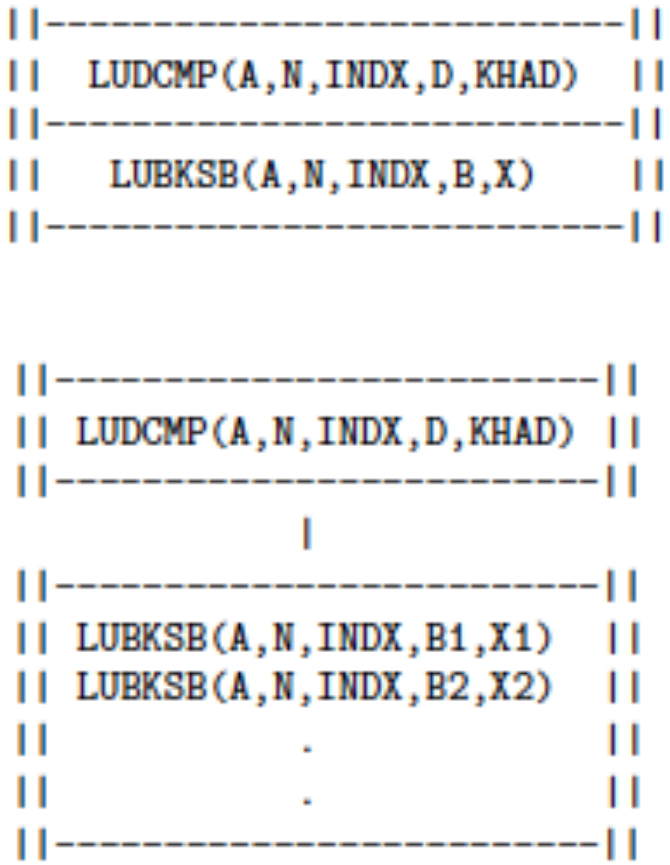
Structure chart 4 — LUBKSB(A,N,INDX,B,X)



$$y_1 = b_1$$

$$y_i = b_i - \sum_{j=1}^{i-1} m_{ij} y_j, \qquad i = 2,3,\ldots,n$$

Possible uses: 1 Inhomogeneous set of equations

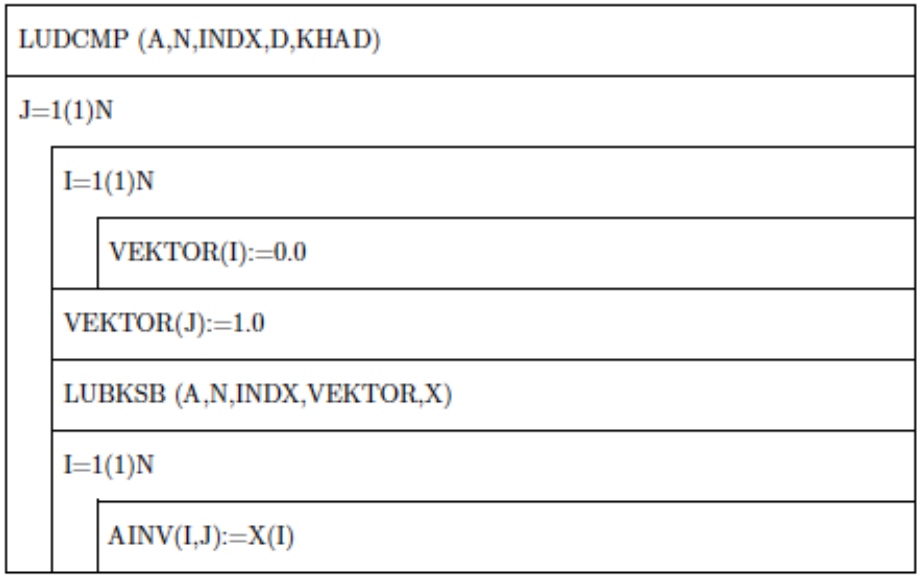


Possible uses: 2 Inversion of a matrix:

$$A \cdot X = I, X \equiv A^{-1}$$

$$A\begin{pmatrix} x_{11} \\ \dots \\ x_{n1} \end{pmatrix} = \begin{pmatrix} 1 \\ \dots \\ 0 \end{pmatrix} \dots \qquad A\begin{pmatrix} x_{1n} \\ \dots \\ x_{nn} \end{pmatrix} = \begin{pmatrix} 0 \\ \dots \\ 1 \end{pmatrix}$$

Structure chart — Inversion of a matrix



Possible uses: (3) Determinant of a matrix.

$$\det(A) = \det(L \cdot U) = \det(L) \cdot \det(U) = 1 \cdot \det(U)$$

$$\det(U) = \sum_{i=1}^{n} u_{ii}$$

The determinant of U is calculated during the LU decomposition.

Today (15/10/2013)

- **Linear Systems:** Definition and applications.
- Solution: **Direct** vs iterative methods.
- Gaussian methods: **LU decomposition**.
- A **practical** example: 3x3 square matrix (step-by-step).
- Strategies to reduce the error: **partial pivoting**.
- Stability of the solution: Condition numbers.
- **Programs** for the LU decomposition: **LUDCMP** and **LUBKSB**.
- How to **use** the programs for LU decompositions (inhomogeneous systems, matrix inversion, determinant of a matrix).