AFRICAN INSTITUTE FOR MATHEMATICAL SCIENCES, AIMS RWANDA

SYSTEMS OF LINEAR EQUATIONS

GAUSSIAN ELIMINATION AND ITS VARIANTS

Prof. Franck Kalala Mutombo
University of Lubumbashi

BLOCKS MATRICES AND BLOCK OPERATIONS

$$\mathbf{AX=B},\ \mathbf{A} \in \mathbb{R}^{n\times m},\quad \mathbf{X} \in \mathbb{R}^{m\times p},\quad \mathbf{B} \in \mathbb{R}^{n\times p}$$

$$A = \begin{array}{c} \\ n_1 \\ n_2 \end{array} \begin{array}{c} \begin{array}{cc} m_1 & m_2 \end{array} \\ \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \end{array} \qquad \begin{cases} n_1 + n_2 = n \\ m_1 + m_2 = m. \end{cases}$$

The number m_1 and m_2 are the same for $\mathbf{A}$ and $\mathbf{X}$. The number of rows in $\mathbf{X}_{12}$ is the same as the number of columns in $\mathbf{A}_{12}$

$$X = \begin{array}{c} \\ m_1 \\ m_2 \end{array} \begin{array}{c} \begin{array}{cc} p_1 & p_2 \end{array} \\ \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \end{array} \qquad \begin{cases} m_1 + m_2 = m \\ p_1 + p_2 = p. \end{cases}$$

BLOCKS MATRICES AND BLOCK OPERATIONS

$$\mathbf{AX=B},\ \mathbf{A} \in \mathbb{R}^{n\times m},\quad \mathbf{X} \in \mathbb{R}^{m\times p},\quad \mathbf{B} \in \mathbb{R}^{n\times p}$$

$$B = \begin{array}{c} \\ n_1 \\ n_2 \end{array} \begin{array}{c} \begin{array}{cc} p_1 & p_2 \end{array} \\ \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \end{array} \qquad \begin{cases} n_1 + n_2 = n \\ p_1 + p_2 = p. \end{cases}$$

The row partition of $\mathbf{B}$ is the same as the row partition of $\mathbf{A}$

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}.$$

BLOCKS MATRICES AND BLOCK OPERATIONS

$$\mathbf{AX=B},\ \mathbf{A} \in \mathbb{R}^{n\times m},\quad \mathbf{X} \in \mathbb{R}^{m\times p},\quad \mathbf{B} \in \mathbb{R}^{n\times p}$$

THEOREM: Let $\mathbf{A}$, $\mathbf{X}$ and $\mathbf{B}$ be partitioned matrices, then AX=B if and only if

$$\mathbf{A}_{i1}\mathbf{X}_{1j} + \mathbf{A}_{i2}\mathbf{X}_{2j} = \mathbf{B}_{ij},\ i,j = 1,2$$

Exercise 1.1.20 Consider matrices A, X, and B, partitioned as indicated.

$$A = \left[\begin{array}{c|cc} 1 & 3 & 2 \\ 2 & 1 & 1 \\ \hline -1 & 0 & 1 \end{array}\right] \qquad X = \left[\begin{array}{c|cc} 1 & 0 & 1 \\ \hline 2 & 1 & 1 \\ -1 & 2 & 0 \end{array}\right] \qquad B = \left[\begin{array}{c|cc} 5 & 7 & 4 \\ 3 & 3 & 3 \\ \hline -2 & 2 & -1 \end{array}\right]$$

Thus, for example, $A_{12} = \begin{bmatrix} 3 & 2 \\ 1 & 1 \end{bmatrix}$ and $A_{21} = \begin{bmatrix} -1 \end{bmatrix}$. Show that $AX = B$ and $A_{i1}X_{1j} + A_{i2}X_{2j} = B_{ij}$ for $i, j = 1, 2$. □

BLOCKS MATRICES AND BLOCK OPERATIONS

$$\mathbf{AX=B},\ \mathbf{A} \in \mathbb{R}^{n\times m},\quad \mathbf{X} \in \mathbb{R}^{m\times p},\quad \mathbf{B} \in \mathbb{R}^{n\times p}$$

Generalisations: Make a finer partition of A r block rows and s block columns

$$A = \begin{array}{c} \\ n_1 \\ \vdots \\ n_r \end{array} \begin{array}{c} \begin{array}{ccc} m_1 & \cdots & m_s \end{array} \\ \begin{bmatrix} A_{11} & \cdots & A_{1s} \\ \vdots & & \vdots \\ A_{r1} & \cdots & A_{rs} \end{bmatrix} \end{array} \qquad \begin{cases} n_1 + \cdots + n_r = n \\ m_1 + \cdots + m_s = m. \end{cases}$$

Then partition of $\mathbf{X}$ comformably with $\mathbf{A}$, i.e make th block structure of $\mathbf{X}$ identical to the block column of $\mathbf{A}$

$$X = \begin{array}{c} \\ m_1 \\ \vdots \\ m_s \end{array} \begin{array}{c} \begin{array}{ccc} p_1 & \cdots & p_t \end{array} \\ \begin{bmatrix} X_{11} & \cdots & X_{1t} \\ \vdots & & \vdots \\ X_{s1} & \cdots & X_{st} \end{bmatrix} \end{array} \qquad \begin{cases} m_1 + \cdots + m_s = m \\ p_1 + \cdots + p_t = p. \end{cases}$$

BLOCKS MATRICES AND BLOCK OPERATIONS

$$\mathbf{AX=B},\ \mathbf{A} \in \mathbb{R}^{n\times m},\quad \mathbf{X} \in \mathbb{R}^{m\times p},\quad \mathbf{B} \in \mathbb{R}^{n\times p}$$

Finally, partition the product B conformably with both A and X.

$$B = \begin{array}{c} \\ n_1 \\ \vdots \\ n_r \end{array} \begin{array}{c} \begin{array}{ccc} p_1 & \cdots & p_t \end{array} \\ \begin{bmatrix} B_{11} & \cdots & B_{1t} \\ \vdots & & \vdots \\ B_{r1} & \cdots & B_{rt} \end{bmatrix} \end{array} \qquad \begin{cases} n_1 + \cdots + n_r = n \\ p_1 + \cdots + p_t = p. \end{cases}$$

THEOREM: Let $\mathbf{A}$, $\mathbf{X}$ and $\mathbf{B}$ be partitioned matrices, then $\mathbf{AX} = \mathbf{B}$ if and only if

$$B_{ij} = \sum_{k=1}^{s} A_{ik} X_{kj} \qquad i = 1, \ldots, r,\ j = 1, \ldots, t.$$

SYSTEM OF LINEAR EQUATION

Given $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^{n}$, solve for $\mathbf{x}$, $\mathbf{A}\mathbf{x} = \mathbf{b}$

Nonsingularity and uniqueness of solution

Consider a system of n linear equations in n unknowns

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

SYSTEM OF LINEAR EQUATION

Given $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^{n}$, solve for $\mathbf{x}$, $\mathbf{Ax} = \mathbf{b}$

Nonsingularity and uniqueness of solution

Previous system can be written in matrix form as

$$Ax = b,$$

where

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

A and b are given, and we must solve for x. A is a square matrix; it has n rows and n columns.

SYSTEM OF LINEAR EQUATION

Given $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^{n}$, solve for $\mathbf{x}$, $\mathbf{Ax} = \mathbf{b}$

$$Ax = b,$$

where

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

The system has a unique solution if and only if the matrix is nonsingular, $det(\mathbf{A}) \neq 0$

SYSTEM OF LINEAR EQUATION

Given $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^{n}$, solve for $\mathbf{x}$, $\mathbf{Ax} = \mathbf{b}$

THEOREM Let $\mathbf{A}$ be a square matrix, the following six condition are equivalents.

- $\mathbf{A}^{-1}$ exists,
- There is no nonzero $\mathbf{y}$ such that $\mathbf{Ay} = 0$,
- The columns of $\mathbf{A}$ are linearly independent,
- The rows of $\mathbf{A}$ are linearly independent ,
- $\det(\mathbf{A}) \neq 0$
- Given any vector $\mathbf{b}$, there exist exactly one vector $\mathbf{x}$, such that $\mathbf{Ax} = \mathbf{b}$.

SYSTEM OF LINEAR EQUATION

Given $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^{n}$, solve for $\mathbf{x}$, $\mathbf{Ax} = \mathbf{b}$

If $\mathbf{A}^{-1}$ exists then the solution of $\mathbf{Ax} = \mathbf{b}$ is given by:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

We will see that it a BAD idea to calculate $\mathbf{A}^{-1}$.

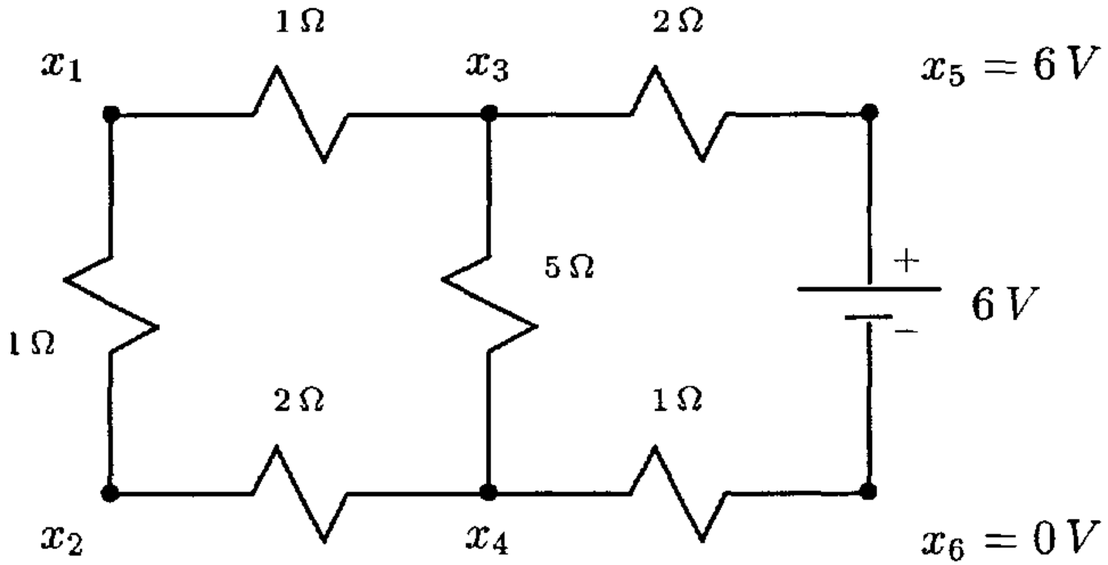
It is generally more efficient to solve $\mathbf{Ax} = \mathbf{b}$ directly without calculating $\mathbf{A}^{-1}$

On most large problems the saving in computation and storage achieved by avoiding the use of$\mathbf{A}^{-1}$ are truly spectacular

SYSTEM OF LINEAR EQUATIONS

Given $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^n$, solve for $\mathbf{x}$, $\mathbf{Ax} = \mathbf{b}$

Example 1: Electrical Circuit



We suppose the circuit is at equilibrium state, I.e all the voltage and current are constant. The four unknown are such that:

1) at each of the four nodes the sum of the currents away the node is equal to zero (Kirkoff law)

Unknowns are nodal voltages x_1, x_2, x_3, x_4

For example the sum of currents of nodes x_3 is given by:

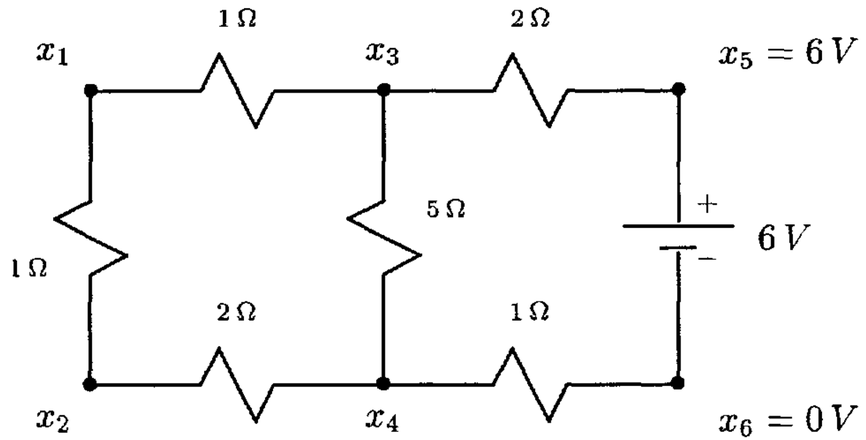
$$.2(x_3 - x_4) + 1(x_3 - x_1) + .5(x_3 - 6) = 0$$

or

$$-x_1 + 1.7x_3 - .2x_4 = 3.$$

SYSTEM OF LINEAR EQUATIONS

Given $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^{n}$, solve for $\mathbf{x}$, $\mathbf{Ax} = \mathbf{b}$

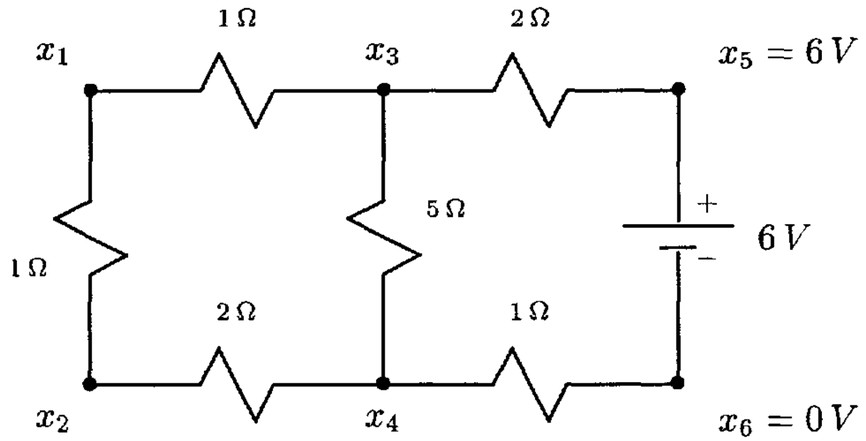


Applying the same procedure to nodes 1, 2, and 4, we obtain a system of four linear equations in four unknowns:

$$
\begin{array}{rcrcrcrcl}
 & 2x_1 & - & x_2 & - & x_3 & & = & 0 \\
- & x_1 & + & 1.5x_2 & & & - \; .5x_4 & = & 0 \\
- & x_1 & & & + & 1.7x_3 & - \; .2x_4 & = & 3 \\
 & & - & .5x_2 & - & .2x_3 & + \; 1.7x_4 & = & 0
\end{array}
\quad ,
$$

SYSTEM OF LINEAR EQUATIONS

Given $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^n$, solve for $\mathbf{x}$, $\mathbf{Ax} = \mathbf{b}$



You can use Julia to show

$$x = \begin{bmatrix} 3.0638 \\ 2.4255 \\ 3.7021 \\ 1.1489 \end{bmatrix}.$$

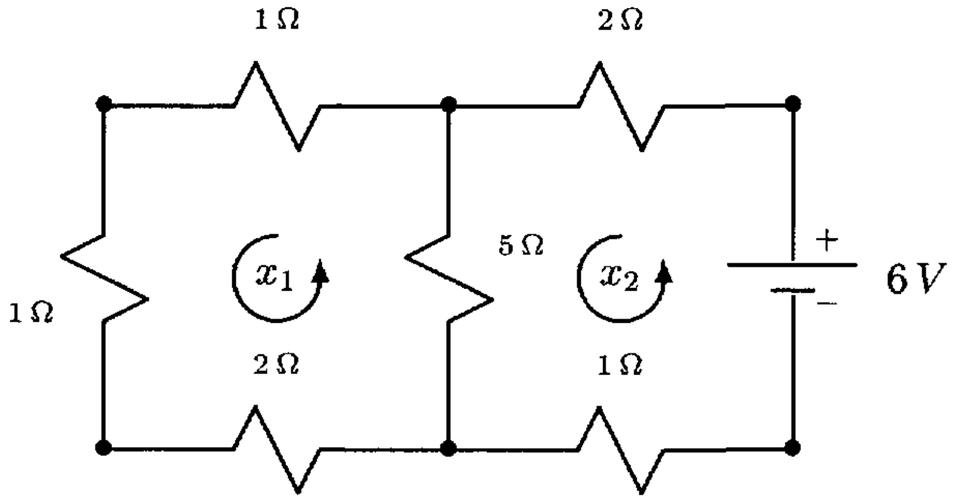
which can be written as a single matrix equation

$$\begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 1.5 & 0 & -.5 \\ -1 & 0 & 1.7 & -.2 \\ 0 & -.5 & -.2 & 1.7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 3 \\ 0 \end{bmatrix}.$$

SYSTEM OF LINEAR EQUATIONS

Given $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{b} \in \mathbb{R}^{n}$, solve for $\mathbf{x}$, $\mathbf{Ax} = \mathbf{b}$

Electrical circuits: Loops currents



Unknowns are loops currents x_1, x_2

Kirchhoff's voltage law

An equation for each loop can be obtained by applying the principle that the sum of voltage Drops around the loop must be zero

For example, the voltage drop across de 5Ω resistor, form the top bottom, $5(x_1 - x_2)$

Summing the voltage drops across the four resistors in loop 1, we have

$$5(x_1 - x_2) + 1x_1 + 1x_1 + 2x_1 = 0$$

Similarly, in loop 2, $$5(x_2 - x_1) + 1x_2 - 6 + 2x_2 = 0$$

Rearranging these equations, we obtain the 2×2 system

$$\begin{bmatrix} 9 & -5 \\ -5 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 6 \end{bmatrix}.$$

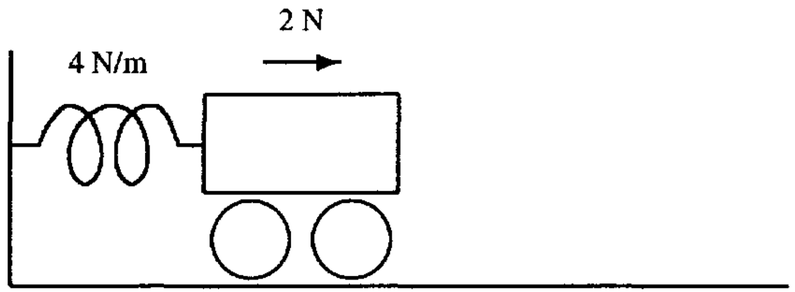
$$x_1 = \frac{30}{47} = 0.6383\ A \quad \text{and} \quad x_2 = \frac{54}{47} = 1.1489\ A$$

Exercise: Use Julia to check that the matrix is nonsingular and solve the system by using Julia.

SYSTEM OF LINEAR EQUATIONS

Given $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^{n}$, solve for $\mathbf{x}$, $\mathbf{Ax} = \mathbf{b}$

Example 2: Mass -Spring Systems



A steady force of 2 newtons is applied to a cart, pushing it to the right and stretching the spring.

How far will the car move before stopping at a new equilibrium point?

It is sufficient to the stiffness of the spring

The new equilibrium will be at the point at which the rightward force of 2 newtons Is exactly balanced by the leftward force applied by the spring

In another words, the equilibrium position is the one at which the sum of the forces on the cart is zero

The restoring force of the spring is -4 newtons/meters $\times x$ meter = -4x newtons

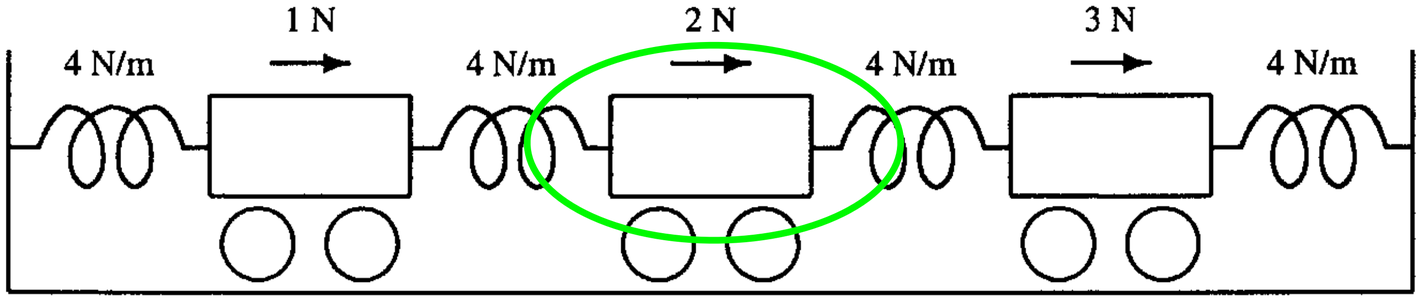
The equilibrium occurs when $-4x + 2 = 0$

Solving this system of one equation in one unknown , we find that $x = 0.5$ meter

SYSTEM OF LINEAR EQUATIONS

Given $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^{n}$, solve for $\mathbf{x}$, $\mathbf{Ax} = \mathbf{b}$

Example 3: Mass -Spring Systems



Now suppose we have three masses attached by springs.

Let x_1, x_2 and x_3 be the amount by which cart 1, 2, and 3, respectively move when the forces are applied.

For each car the new equilibrium position is that at which the sum of the forces on the car is zero.

Let focus on the second car for example

An external force of 2 newtons is applied, and there is the leftward force of the spring to the left, and the rightward force of the spring to the right

The amount by which the spring on the is stretched is $x_2 - x_1$ matters.

It therefore exerts a for of -4newtons/meter $\times (x_2 - x_1)$
meters = -4 $(x_2 - x_1)$ newtons on the second cart.

Similarly the spring on the right applies a force of +4$((x_3 - x_2)$

Thus the equilibrium equations for the second car is

$$-4(x_2 - x_1) + 4(x_3 - x_2) + 2 = 0$$

Similar equations apply to carts 1 and 3, lead the system

$$\begin{bmatrix} 8 & -4 & 0 \\ -4 & 8 & -4 \\ 0 & -4 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}.$$

Check the answer with Julia

$$x = \begin{bmatrix} 0.625 \\ 1.000 \\ 0.875 \end{bmatrix}.$$

SYSTEM OF LINEAR EQUATIONS

Given $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{b} \in \mathbb{R}^{n}$, solve for $\mathbf{x}$, $\mathbf{Ax} = \mathbf{b}$

Example 4: Numerical (Approximate) Solution of Differential Equations

Many physical phenomena can be modeled by differential equations. Let consider briefly one example

$$-u''(x) + c\,u'(x) + d\,u(x) = f(x) \qquad 0 < x < 1$$

with boundary conditions

$$u(0) = 0, \quad u(1) = 0.$$

The problem is to solve for the unknown function u, given the constants c and d and the function f.

For example $u(x)$ could represent the unknown concentration of some chemical pollutant at distance x from the end of the pipe

Pick a possibly large integer m, and subdivide the interval [0, 1] into m equal subintervals of length $h = 1/m$.

The subdivision points of the intervals are $x_j = jh, j = 0,\cdots, m$. These points constitute our computational grid.

The finite difference method will produce approximations to $u_1(x), \cdots, u_n(x)$.

We have at each grid $$-u''(x_i) + c\,u'(x_i) + d\,u(x_i) = f(x_i) \qquad i = 1, \ldots, m-1.$$

SYSTEM OF LINEAR EQUATIONS

Given $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^n$, solve for $\mathbf{x}$, $\mathbf{Ax} = \mathbf{b}$

$$-u''(x) + c\,u'(x) + d\,u(x) = f(x) \qquad 0 < x < 1$$

with boundary conditions

$$u(0) = 0, \quad u(1) = 0.$$

The finite difference method will produce approximations to $u_1(x), \cdots, u_n(x)$. We have at each grid

$$-u''(x_i) + c\,u'(x_i) + d\,u(x_i) = f(x_i) \qquad i = 1, \ldots, m-1.$$

If h is small, good approximation for the first and second derivatives are

$$u'(x_i) \approx \frac{u(x_i + h) - u(x_i - h)}{2h} = \frac{u(x_{i+1}) - u(x_{i-1})}{2h}$$

and

$$u''(x_i) \approx \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2}.$$

Substituting these approximations we obtain for $i = 1, \cdots, m-1$

$$\frac{-u(x_{i+1}) + 2u(x_i) - u(x_{i-1})}{h^2} + c\left(\frac{u(x_{i+1}) - u(x_{i-1})}{2h}\right) + du(x_i) \approx f(x_i).$$

SYSTEM OF LINEAR EQUATIONS

Given $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^{n}$, solve for $\mathbf{x}$, $\mathbf{Ax} = \mathbf{b}$

We now approximate this by a system of difference equations

$$\frac{-u_{i+1} + 2u_i - u_{i-1}}{h^2} + c\left(\frac{u_{i+1} - u_{i-1}}{2h}\right) + du_i = f_i, \qquad i = 1, \cdots, m-1 \qquad \textbf{x}$$

This is a system of $m-1$ linear equations in the unknowns $u_0, \cdots, u_m$.

Applying the boundary conditions, can take $u_0 = 0$ and $u_m = 1$, leaving only $m-1$ unknowns and $u_1, \cdots, u_m$.

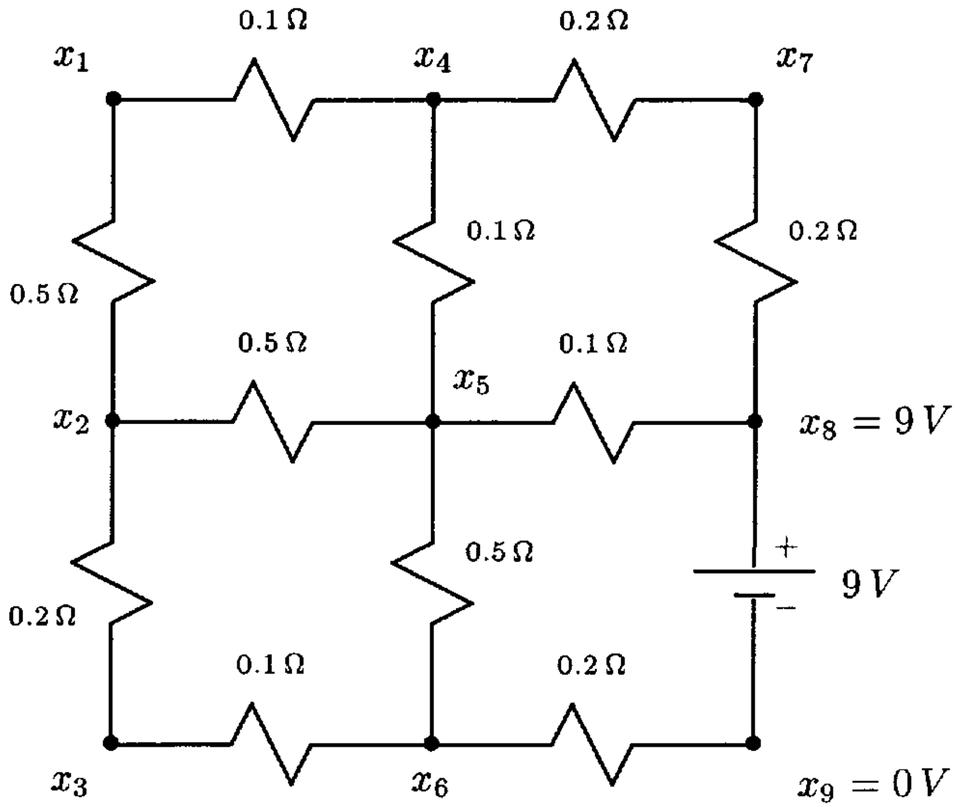
Suppose, for example, m = 6 and $h = 1/6$, then **x** is a system of five equations in five unknowns , which can be written as the single matrix equation

$$\begin{bmatrix} 72+d & -36+3c & 0 & 0 & 0 \\ -36-3c & 72+d & -36+3c & 0 & 0 \\ 0 & -36-3c & 72+d & -36+3c & 0 \\ 0 & 0 & -36-3c & 72+d & -36+3c \\ 0 & 0 & 0 & -36-3c & 72+d \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix}$$

SYSTEM OF LINEAR EQUATIONS

Given $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^{n}$, solve for $\mathbf{x}$, $\mathbf{Ax} = \mathbf{b}$

Exercise Consider the electrical circuit in the figure below



Electric circuit with nodal voltages

- Write down a linear system $\mathbf{Ax} = \mathbf{b}$ with seven equations for the seven unknowns nodal voltages.
- Using Julia, for solving the system to find the nodal voltages, calculate the residual $\mathbf{r} = \mathbf{b} - \mathbf{A}\hat{\mathbf{x}}$, where $\hat{\mathbf{x}}$ denote your computed solution.
- In theory $\mathbf{r}$ should be zero. In practice you will get a thing but nonzero residual because of the roundoff errors in your computations.

SYSTEM OF LINEAR EQUATIONS

TRIANGULAR SYSTEMS

A linear system whose coefficient matrix is triangular is particularly easy to solve. It is a common practice to reduce general systems to a triangular form, which can then be solved inexpensively. For this reason we will study triangular systems in detail.

A matrix $G = (g_{ij})$ is *lower triangular* if $g_{ij} = 0$ whenever $i < j$. Thus a lower-triangular matrix has the form

$$G = \begin{bmatrix} g_{11} & 0 & 0 & \cdots & 0 \\ g_{21} & g_{22} & 0 & \cdots & 0 \\ g_{31} & g_{32} & g_{33} & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ g_{n1} & g_{n2} & g_{n3} & \cdots & g_{nn} \end{bmatrix}.$$

Similarly, an *upper triangular* matrix is one for which $g_{ij} = 0$ whenever $i > j$. A *triangular* matrix is one that is either upper or lower triangular.

SYSTEM OF LINEAR EQUATIONS

TRIANGULAR SYSTEMS

Theorem 1.3.1 *Let G be a triangular matrix. Then G is nonsingular if and only if $g_{ii} \neq 0$ for $i = 1, \ldots, n$.*

Lower Triangular Systems

Consider the system

$$\mathbf{Gy} = \mathbf{b},$$

Where G is nonsingular, lower-triangular matrix. Let write the full system to see how we can solve it easly:

$$\begin{array}{lllllllll}
g_{11}y_1 & & & & & & & = & b_1 \\
g_{21}y_1 & + & g_{22}y_2 & & & & & = & b_2 \\
g_{31}y_1 & + & g_{32}y_2 & + & g_{33}y_3 & & & = & b_3 \\
 & & \vdots & & & & & \vdots & \\
g_{n1}y_1 & + & g_{n2}y_2 & + & g_{n3}y_3 & + \cdots + & g_{nn}y_n & = & b_n.
\end{array}$$

SYSTEM OF LINEAR EQUATIONS

TRIANGULAR SYSTEMS

Lower Triangular Systems

$$\mathbf{Gy} = \mathbf{b},$$

$$\begin{array}{lllllll}
g_{11}y_1 & & & & & = & b_1 \\
g_{21}y_1 & + & g_{22}y_2 & & & = & b_2 \\
g_{31}y_1 & + & g_{32}y_2 & + & g_{33}y_3 & = & b_3 \\
 & & \vdots & & & \vdots & \\
g_{n1}y_1 & + & g_{n2}y_2 & + & g_{n3}y_3 + \cdots + g_{nn}y_n & = & b_n.
\end{array}$$

We can solve the first equation for y_1

$$y_1 = b_1/g_{11} \quad (g_{11} \neq 0)$$

Or

We can substitute y_1 into the second equation to solve for y_2

$$y_2 = (b_2 - g_{21}y_1)/g_{22} \quad (g_{22} \neq 0)$$

Forward substitution

In general, once we have $y_1, y_2, \cdots, y_{i-1}$, we can solve for y_i, using the formula

$$y_i = \frac{b_i - g_{i1}y_1 - g_{i2}y_2 - \cdots - g_{i,i-1}y_{i-1}}{g_{ii}}$$

$$y_i = g_{ii}^{-1}\left(b_i - \sum_{j=1}^{i-1} g_{ij}y_j\right)$$

Forward elimination

SYSTEM OF LINEAR EQUATIONS

TRIANGULAR SYSTEMS

Lower Triangular Systems

$$\mathbf{Gy} = \mathbf{b} \quad \Longrightarrow \quad y_i = g_{ii}^{-1}\left(b_i - \sum_{j=1}^{i-1} g_{ij}y_j\right)$$

Forward substitution/ Forward elimination

It is also called row-oriented forward substitution because it accesses G by rows; the with row is used at the 4th step.

It is also called the inner-product form of forward substitution because the sum $\sum_{j=1}^{i-1} g_{ij}y_j$ can be regarded as an inner or dot product

Exercise : write a Julia program for forward substitution. Use pencil and paper to solve the system

Estimate the execution of the forward substitution by counting the floating-point operations (flops)

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 \\ 3 & 1 & -1 & 0 \\ 4 & 1 & -3 & 3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 2 \\ 9 \end{bmatrix}$$

Check your answer with your Julia program

SYSTEM OF LINEAR EQUATIONS

TRIANGULAR SYSTEMS

Lower Triangular Systems

$$\mathbf{Gy} = \mathbf{b} \quad \Longrightarrow \quad y_i = g_{ii}^{-1}\left(b_i - \sum_{j=1}^{i-1} g_{ij}y_j\right)$$

Forward substitution/ Forward elimination

Algorithm

for $i = 1, \ldots, n$

[for $j = 1, \ldots, i-1$ (not executed when $i = 1$)

[$b_i \leftarrow b_i - g_{ij}b_j$

if $g_{ii} = 0$, set error flag, exit

$b_i \leftarrow b_i / g_{ii}$

FS

SYSTEM OF LINEAR EQUATIONS

TRIANGULAR SYSTEMS

Exploiting Leading Zeros in Forward Substitution

$$\mathbf{Gy = b}$$

If $b_1 = b_2 = \cdots b_k = 0$, we can skip all of the computation involving $y_1, \cdots, y_k$

$$G = \begin{bmatrix} g_{11} & 0 & 0 & \cdots & 0 \\ g_{21} & g_{22} & 0 & \cdots & 0 \\ g_{31} & g_{32} & g_{33} & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ g_{n1} & g_{n2} & g_{n3} & \cdots & g_{nn} \end{bmatrix}.$$

$$y_i = g_{ii}^{-1}\left(b_i - \sum_{j=1}^{i-1} g_{ij}y_j\right) \quad \Longrightarrow \quad y_i = g_{ii}^{-1}\left(b_i - \sum_{j=k+1}^{n} g_{ij}y_j\right) \quad i = k+1, \cdots, n$$

View of partitioned matrices

If $b_1 = b_2 = \cdots b_k = 0$, we can write

$$b = \begin{bmatrix} 0 \\ \hat{b}_2 \end{bmatrix} \begin{matrix} k \\ j \end{matrix},$$

where $j = n - k$. Partitioning G and y also, we have

$$G = \begin{matrix} \\ k \\ j \end{matrix} \begin{matrix} \begin{matrix} k & j \end{matrix} \\ \begin{bmatrix} G_{11} & 0 \\ G_{21} & G_{22} \end{bmatrix} \end{matrix} \qquad y = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix} \begin{matrix} k \\ j \end{matrix},$$

where G_{11} and G_{22} are lower triangular. The equation $Gy = b$ becomes

$$\begin{bmatrix} G_{11} & 0 \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \hat{b}_2 \end{bmatrix}$$

or

$$\begin{aligned} G_{11}\hat{y}_1 &= 0 \\ G_{21}\hat{y}_1 + G_{22}\hat{y}_2 &= \hat{b}_2. \end{aligned}$$

Since G_{11} is nonsingular (why?), the first equation implies $\hat{y}_1 = 0$. The second equation then reduces to

$$G_{22}\hat{y}_2 = \hat{b}_2.$$

SYSTEM OF LINEAR EQUATIONS
TRIANGULAR SYSTEMS

where G_{11} and G_{22} are lower triangular. The equation $Gy = b$ becomes

$$\begin{bmatrix} G_{11} & 0 \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \hat{b}_2 \end{bmatrix}$$

or

$$\begin{aligned} G_{11}\hat{y}_1 &= 0 \\ G_{21}\hat{y}_1 + G_{22}\hat{y}_2 &= \hat{b}_2. \end{aligned}$$

- Thus we only have to solve this $(n-k)\times(n-k)$ lower triangular system.
- G_{11} and G_{21} are not used because they interact only with the unknowns $y_1, \cdots, y_k$ (i.e, $\hat{y}_k$).
- Since the system now being solved is of order $n-k$, the cost is now $(n-k)^2$ flops.

for $i = 1, \ldots, n$
  for $j = 1, \ldots, i-1$ (not executed when $i = 1$)
    $b_i \leftarrow b_i - g_{ij} b_j$
  if $g_{ii} = 0$, set error flag, exit
  $b_i \leftarrow b_i / g_{ii}$

Exercise

Write a modified version of the algorithm on the right that checks for leading zeros in b and take appropriate action.

SYSTEM OF LINEAR EQUATIONS

TRIANGULAR SYSTEMS

Column -Oriented Forward Substitution

We now derive a column-oriented version of forward substitution. Partition the system $Gy = b$ as follows:

$$\begin{bmatrix} g_{11} & 0 \\ \hat{h} & \hat{G} \end{bmatrix} \begin{bmatrix} y_1 \\ \hat{y} \end{bmatrix} = \begin{bmatrix} b_1 \\ \hat{b} \end{bmatrix}.$$

$\hat{h}$, $\hat{y}$, and $\hat{b}$ are vectors of length $n-1$, and $\hat{G}$ is an $(n-1) \times (n-1)$ lower-triangular matrix. The partitioned system can be written as

$$\begin{array}{rcl} g_{11}y_1 & = & b_1 \\ \hat{h}y_1 + \hat{G}\hat{y} & = & \hat{b}. \end{array}$$

This leads to the following algorithm:

Recursive Algorithm

$y_1 = b_1 / g_{11}$
$\tilde{b} = \hat{b} - \hat{h}y_1$
solve $\hat{G}\hat{y} = \tilde{b}$ for $\hat{y}$.

SYSTEM OF LINEAR EQUATIONS

TRIANGULAR SYSTEMS

Column -Oriented Forward Substitution

This leads to the following algorithm:

$$
\begin{aligned}
&y_1 = b_1 / g_{11} \\
&\tilde{b} = \hat{b} - \hat{h} y_1 \\
&\text{solve } \hat{G}\hat{y} = \tilde{b} \text{ for } \hat{y}.
\end{aligned}
$$

Recursive Algorithm

This algorithm reduces the problem of solving an $n \times n$ triangular system to that of solving the $(n-1) \times (n-1)$ system $\hat{G}\hat{y} = \tilde{b}$. This smaller problem can be reduced (by the same algorithm) to a problem of order $n-2$, which can in turn be reduced to a problem of order $n-3$, and so on. Eventually we get to the 1×1 problem $g_{nn} y_n = \check{b}_n$, which has the solution $y_n = \check{b}_n / g_{nn}$.

The procedure will do step one and two of the algorithm and call it self again

SYSTEM OF LINEAR EQUATIONS

TRIANGULAR SYSTEMS

Example: Use column-oriented version of forward substitution to solve the lower-triangular system

$$\begin{bmatrix} 5 & 0 & 0 \\ 2 & -4 & 0 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 15 \\ -2 \\ 10 \end{bmatrix}.$$

First we calculate $y_1 = b_1/g_{11} = 15/5 = 3$. Then

$$\tilde{b} = \hat{b} - \hat{h}y_1 = \begin{bmatrix} -2 \\ 10 \end{bmatrix} - \begin{bmatrix} 2 \\ 1 \end{bmatrix} 3 = \begin{bmatrix} -8 \\ 7 \end{bmatrix}.$$

Now we have to solve $\hat{G}\hat{y} = \tilde{b}$:

$$\begin{bmatrix} -4 & 0 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} -8 \\ 7 \end{bmatrix}.$$

We achieve this by repeating the algorithm: $y_2 = -8/-4 = 2$,

$$\tilde{\tilde{b}} = \hat{\tilde{b}} - \hat{\tilde{h}}y_2 = \begin{bmatrix} 7 \end{bmatrix} - \begin{bmatrix} 2 \end{bmatrix} 2 = \begin{bmatrix} 3 \end{bmatrix},$$

$\begin{bmatrix} 3 \end{bmatrix} y_3 = \begin{bmatrix} 3 \end{bmatrix}$, and $y_3 = 3/3 = 1$. Thus

$$y = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}.$$

You can check that if you multiply G by y, you get the correct right hand side b.

SYSTEM OF LINEAR EQUATIONS

TRIANGULAR SYSTEMS

Question: Which of the two version of forward substitution (row oriented and column oriented) is superior?

1. The answer depends on how G is stored and accessed.
2. This, in turn, depends on the programmer’s choice of data structure and
3. Programming language and
4. Computer architecture

SYSTEM OF LINEAR EQUATIONS

TRIANGULAR SYSTEMS

Upper-Triangular Systems

$$\mathbf{U}\mathbf{y} = \mathbf{b}$$ where $\mathbf{U}$ is upper triangular

Writing out the system in detail we get

$$\begin{array}{ccccccccccc}
u_{11}x_1 & + & u_{12}x_2 & + & \cdots & + & u_{1,n-1}x_{n-1} & + & u_{1,n}x_n & = & y_1 \\
 & & u_{22}x_2 & + & \cdots & + & u_{2,n-1}x_{n-1} & + & u_{2n}x_n & = & y_2 \\
 & & & & & & \vdots & & & & \vdots \\
 & & & & & & u_{n-1,n-1}x_{n-1} & + & u_{n-1,n}x_n & = & y_{n-1} \\
 & & & & & & & & u_{nn}x_n & = & y_n .
\end{array}$$

- We should solve the system from bottom to top.
- The nth equation can be solved for x_n, then $(n-1)$st equation can solved for x_{n-1}, and son on.
- The process is called back substitution, and it has row- and column-oriented versions.
- The cost of back substitution is the same as for the forward substitution, about n^2

SYSTEM OF LINEAR EQUATIONS

TRIANGULAR SYSTEMS

Exercise

1. Develop the row-oriented version of the back substation process. Write pseudocode of it and a Julia program.
2. Develop the column-oriented version of the back substation process. Write pseudocode of it and a Julia program.
3. Using your Julia codes in 1) and 2) solve the upper-triangular system

$$\begin{bmatrix} 3 & 2 & 1 & 0 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & -2 & 1 \\ 0 & 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -10 \\ 10 \\ 1 \\ 12 \end{bmatrix}$$

SYSTEM OF LINEAR EQUATIONS

TRIANGULAR SYSTEMS

Block Algorithms

It is easy to develop block variants of both forward and back substitution. We will focus on forward substitution. Suppose the lower triangular matrix G has been partitioned into blocks as follows:

$$G = \begin{array}{c} \\ r_1 \\ r_2 \\ \vdots \\ r_s \end{array} \begin{array}{c} \begin{array}{cccc} r_1 & r_2 & \cdots & r_s \end{array} \\ \begin{bmatrix} G_{11} & 0 & \cdots & 0 \\ G_{21} & G_{22} & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ G_{s1} & G_{s2} & \cdots & G_{ss} \end{bmatrix} \end{array} .$$

Each G_{ii} is square and lower triangular. Then the equation $Gy = b$ can be written in the partitioned form

SYSTEM OF LINEAR EQUATIONS
TRIANGULAR SYSTEMS

Block Algorithms

Each G_{ii} is square and lower triangular. Then the equation $Gy = b$ can be written in the partitioned form

$$\begin{bmatrix} G_{11} & 0 & \cdots & 0 \\ G_{21} & G_{22} & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ G_{s1} & G_{s2} & \cdots & G_{ss} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_s \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_s \end{bmatrix}.$$

- The entries b_i and y_i are not scalars; they are vectors with r_i component each.
- We find y_1 by solving the system $G_{11}y_1 = b_1$
- Once we have y_1, we can solve the equation $G_{21}y_1 + G_{22}y_2 = b_2$ for y_2, and so on.

Block version of row-oriented forward substitution

for $i = 1, \ldots, s$
  for $j = 1, \ldots, i-1$ (not executed when $i = 1$)
    $b_i \leftarrow b_i - G_{ij}b_j$
  if G_{ii} is singular, set error flag, exit
  $b_i \leftarrow G_{ii}^{-1}b_i$

The operation does not require explicit computation of $b_i \leftarrow G_{ii}^{-1}b_i$.
It can be done by solving the lower triangular system $G_{ii}^{-1}x = b_i$ by either row- or column-oriented forward substitution

SYSTEM OF LINEAR EQUATIONS

POSITIVE DEFINITE SYSTEMS; CHOLESKY DECOMPOSTION

If A is $n \times n$, real, symmetric, and also satisfies the property

$$x^T Ax > 0$$

Then A is positive definite

Theorem: If $\mathbf{A}$ is positive definite, then $\mathbf{A}$ is nonsingular.

Let prove the contrapositive. If A is singular, then A is not positive definite. Assume A is singular. Then, there is a nonzero $y \in \mathbb{R}^n$ such that $Ay = 0$. But then $\mathbf{y}^T\mathbf{A}\mathbf{y} = \mathbf{0}$, so $\mathbf{A}$ is not positive.

Corollary: if $\mathbf{A}$ is positive definite, the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ has exactly one solution.

Theorem: Let $\mathbf{M}$ be any $n \times n$ nonsingular matrix, and let $\mathbf{A} = \mathbf{M}^T\mathbf{M}$. Then $\mathbf{A}$ is positive definite.

Proof: Exercise

SYSTEM OF LINEAR EQUATIONS

POSITIVE DEFINITE SYSTEMS; CHOLESKY DECOMPOSTION

Example 1

Let $M = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$. M is nonsingular, since $\det(M) = -2$.

Therefore $A = M^T M = \begin{bmatrix} 10 & 14 \\ 14 & 20 \end{bmatrix}$ is positive definite. Show it

Example 2

Let

$$M = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

M is nonsingular, since $\det(M) = 1$. Therefore

$$A = M^T M = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{bmatrix}$$

is positive definite.

SYSTEM OF LINEAR EQUATIONS

POSITIVE DEFINITE SYSTEMS; CHOLESKY DECOMPOSTION

Theorem: (Cholesky Decomposition Theorem)

Let $\mathbf{A}$ be positive definite. Then $\mathbf{A}$ can be decomposed in exactly one way into a product

$$\mathbf{A} = \mathbf{R}^T\mathbf{R},$$

Such that $\mathbf{R}$ is upper triangular and has all main diagonal entries $\mathbf{r}_{ii}$ positive. $\mathbf{R}$ is called the Cholesky factor of $\mathbf{A}$.

Example 1

Let

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{bmatrix} \quad \text{and} \quad R = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

$\mathbf{R}$ is upper triangular and has positive main-diagonal entries. Observe that $\mathbf{A} = \mathbf{R}^T\mathbf{R}$. Therefore $\mathbf{R}$ is the Cholesky factor of A.

SYSTEM OF LINEAR EQUATIONS

POSITIVE DEFINITE SYSTEMS; CHOLESKY DECOMPOSTION

We can use Cholesky factor to solve the system $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is positive definite, we can write the system as

$$\mathbf{R}^T\mathbf{Rx} = \mathbf{b}.$$

Let

$$\mathbf{y} = \mathbf{Rx}$$

$\mathbf{y}$ then satisfies the system

$$\mathbf{R}^T\mathbf{y} = \mathbf{b}$$

Since $\mathbf{R}^T$ is lower triangular, we can solve for $\mathbf{y}$ by forward substitution.

Once we have $\mathbf{y}$, we can solve the upper-triangular. System

$$\mathbf{Rx} = \mathbf{y}$$

by back propagation substitution.

Exercise

Find the Cholesky flops count, if we know the Cholesky factor $\mathbf{R}$.

SYSTEM OF LINEAR EQUATIONS

POSITIVE DEFINITE SYSTEMS; CHOLESKY DECOMPOSTION

PRACTICAL METHOD OF CALCULATING CHOLESKY FACTOR

We write out the the decomposition $\mathbf{A} = \mathbf{R}^T\mathbf{R}$

The element a_{ij} is the (inner) product of the ith row of R^T with the jth column of R. Noting that the first row of R^T has only one nonzero entry, we focus on this row:

$$a_{1j} = r_{11}r_{1j} + 0r_{2j} + 0r_{3j} + \cdots + 0r_{nj} = r_{11}r_{1j}.$$

In particular, when $j = 1$ we have $a_{11} = r_{11}^2$, which tells us that

$$r_{11} = +\sqrt{a_{11}}.$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}$$

$$= \begin{bmatrix} r_{11} & 0 & 0 & \cdots & 0 \\ r_{12} & r_{22} & 0 & \cdots & 0 \\ r_{13} & r_{23} & r_{33} & & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ r_{1n} & r_{2n} & r_{3n} & \cdots & r_{nn} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1n} \\ 0 & r_{22} & r_{23} & \cdots & r_{2n} \\ 0 & 0 & r_{33} & \cdots & r_{3n} \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & & r_{nn} \end{bmatrix}$$

SYSTEM OF LINEAR EQUATIONS

POSITIVE DEFINITE SYSTEMS; CHOLESKY DECOMPOSTION

We know that the positive square root is the right one, because the main-diagonal entries of R are positive. Now that we know r_{11}, we can use the equation $a_{1j} = r_{11}r_{1j}$ to calculate the rest of the first row of R:

$$r_{1j} = a_{1j}/r_{11}, \quad j = 2, \ldots, n.$$

This is also the first column of R^T. We next focus on the second row, because the second row of R^T has only two nonzero entries. We have

$$a_{2j} = r_{12}r_{1j} + r_{22}r_{2j}.$$

Only elements from the first two rows of R appear in this equation. In particular, when $j = 2$ we have $a_{22} = r_{12}^2 + r_{22}^2$. Since r_{12} is already known, we can use this

equation to calculate r_{22}:

$$r_{22} = +\sqrt{a_{22} - r_{12}^2}.$$

SYSTEM OF LINEAR EQUATIONS

POSITIVE DEFINITE SYSTEMS; CHOLESKY DECOMPOSTION

$$a_{2j} = r_{12}r_{1j} + r_{22}r_{2j}. \quad \blacktriangle$$

equation to calculate r_{22}:

$$r_{22} = +\sqrt{a_{22} - r_{12}^2}.$$

Once r_{22} is known, the only unknowns is r_{2j}. We can then use $\blacktriangle$ to compute the rest of the second row of $\mathbf{R}$ as follow:

$$r_{2j} = (a_{2j} - r_{12}r_{1j})/r_{22}, \quad j = 3, \ldots, n. \qquad r_{21} = 0$$

We now know the first two rows of $\mathbf{R}$ and its transpose $\mathbf{R}^T$. The same procedure can be use the find the third row of $\mathbf{R}$.

Now let us see how to calculate the ith row of R, assuming that we already have the first $i - 1$ rows. Since only the first i entries in the ith row of R^T are nonzero,

$$a_{ij} = r_{1i}r_{1j} + r_{2i}r_{2j} + \cdots + r_{i-1,i}r_{i-1,j} + r_{ii}r_{ij}. \quad \bigstar$$

SYSTEM OF LINEAR EQUATIONS

POSITIVE DEFINITE SYSTEMS; CHOLESKY DECOMPOSTION

CHOLESKY METHOD

$$a_{ij} = r_{1i}r_{1j} + r_{2i}r_{2j} + \cdots + r_{i-1,i}r_{i-1,j} + r_{ii}r_{ij}. \qquad (**)$$

All entries of $\mathbf{R}$ appearing in **(**)** lie in the first i rows. Since the first $i-1$ rows are known, the only unknowns in **(**)** are r_{ii}. Taking $i = j$ in **(**)** we have

$$a_{ii} = r_{1i}^2 + r_{2i}^2 + \cdots + r_{i-1,i}^2 + r_{ii}^2,$$

And therefore

$$r_{ii} = +\sqrt{a_{ii} - \sum_{k=1}^{i-1} r_{ki}^2}. \qquad (***)$$

Now that we know r_{ii} we can use **(**)** the solve for r_{ij}

$$r_{ij} = \left(a_{ij} - \sum_{k=1}^{i-1} r_{ki}r_{kj}\right) \Big/ r_{ii} \qquad j = i+1, \ldots, n. \qquad r_{ij} = 0, \quad j < i\,. \qquad (****)$$

(***) and **(****)** fully complete the matrix R.

SYSTEM OF LINEAR EQUATIONS

POSITIVE DEFINITE SYSTEMS; CHOLESKY DECOMPOSTION

CHOLESKY METHOD

$$r_{ii} = +\sqrt{a_{ii} - \sum_{k=1}^{i-1} r_{ki}^2}. \qquad (***)$$

$$r_{ij} = \left(a_{ij} - \sum_{k=1}^{i-1} r_{ki} r_{kj}\right) \Big/ r_{ii} \qquad j = i+1, \ldots, n. \qquad r_{ij} = 0, \quad j < i. \qquad (****)$$

Cholesky method is also inner-product formulation because (***) and (****) can regarded as inner product.

- Cholesky decomposition theorem makes 2 assertions:
- (i) $\mathbf{R}$ exits, and (ii) $\mathbf{R}$ is unique.

The equation $\mathbf{A} = \mathbf{R}^T\mathbf{R}$ and the fact that $\mathbf{R}$ is upper triangular with $r_{11} > 0$ imply $r_{11} = +\sqrt{a_{11}}$.

Is the only one value that will work. r_{11} is determined uniquely

SYSTEM OF LINEAR EQUATIONS

POSITIVE DEFINITE SYSTEMS; CHOLESKY DECOMPOSTION

CHOLESKY METHOD

TEST OF POSITIVE DEFINITENESS

Theorem: Let $\mathbf{M}$ be any $n \times n$ nonsingular matrix, and let $\mathbf{A} = \mathbf{M}^T\mathbf{M}$. Then $\mathbf{A}$ is positive definite.

Theorem: (Cholesky Decomposition Theorem)

Let $\mathbf{A}$ be positive definite. Then $\mathbf{A}$ can be decomposed in exactly one way into a product

$$\mathbf{A} = \mathbf{R}^T\mathbf{R},$$

Such that $\mathbf{R}$ is upper triangular and has all main diagonal entries $\mathbf{r}_{ii}$ positive. $\mathbf{R}$ is called the Cholesky factor of $\mathbf{A}$.

Given any symmetric matrix $\mathbf{A}$, can try to compute $\mathbf{R}$ by Cholesky's method. If $\mathbf{A}$ is not positive definite, the algorithm will fail $\mathbf{R}$ must satisfy $\mathbf{A} = \mathbf{R}^T\mathbf{R}$

The algorithm fails if and only if at some step the number under the square root sign is negative or zero.

Cholesky method succeeds if and only if $\mathbf{A}$ is positive definite.

This is the best general test of positive definiteness known.

SYSTEM OF LINEAR EQUATIONS

POSITIVE DEFINITE SYSTMS; CHOLESKY DECOMPOSTION

CHOLESKY METHOD

Cholesky's Algorithm (inner-product form)

for $i = 1, \ldots, n$

  for $k = 1, \ldots, i-1$ (not executed when $i = 1$)

    $a_{ii} \leftarrow a_{ii} - a_{ki}^2$

  if $a_{ii} \leq 0$, set error flag (A is not positive definite), exit

  $a_{ii} \leftarrow \sqrt{a_{ii}}$ (this is r_{ii})

  for $j = i+1, \ldots, n$ (not executed when $i = n$)

    for $k = 1, \ldots, i-1$ (not executed when $i = 1$)

      $a_{ij} \leftarrow a_{ij} - a_{ki}a_{kj}$

    $a_{ij} \leftarrow a_{ij}/a_{ii}$ (this is r_{ij})

Exercice Implement this algorithm in Julia and us your program to solve the system

$$\mathbf{Ax} = \mathbf{b}.$$

Where

$$A = \begin{bmatrix} 4 & -2 & 4 & 2 \\ -2 & 10 & -2 & -7 \\ 4 & -2 & 8 & 4 \\ 2 & -7 & 4 & 7 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 8 \\ 2 \\ 16 \\ 6 \end{bmatrix}$$

SYSTEM OF LINEAR EQUATIONS

POSITIVE DEFINITE SYSTMS; CHOLESKY DECOMPOSTION

CHOLESKY METHOD

EXERCICE

Let

$$A = \begin{bmatrix} 16 & 4 & 8 & 4 \\ 4 & 10 & 8 & 4 \\ 8 & 8 & 12 & 10 \\ 4 & 4 & 10 & 12 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 32 \\ 26 \\ 38 \\ 30 \end{bmatrix}.$$

Notice that A is symmetric. (a) Use the inner-product formulation of Cholesky's method to show that A is positive definite and compute its Cholesky factor. (b) Use forward and back substitution to solve the linear system $Ax = b$. □

SYSTEM OF LINEAR EQUATIONS

POSITIVE DEFINITE SYSTMS; CHOLESKY DECOMPOSTION

Flops Count

To count the flops in Cholesky's algorithm, we need to know that

$$\sum_{i=1}^{n} i^2 = \frac{n^3}{3} + O(n^2)\,.$$

The easiest way to obtain this is to approximate the sum by an integral:

$$\sum_{i=1}^{n} i^2 \approx \int_0^n i^2\,di = \frac{n^3}{3}\,.$$

Cholesky's Algorithm (inner-product form)

for $i = 1, \ldots, n$
  for $k = 1, \ldots, i-1$ (not executed when $i = 1$)
    $a_{ii} \leftarrow a_{ii} - a_{ki}^2$
  if $a_{ii} \leq 0$, set error flag (A is not positive definite), exit
  $a_{ii} \leftarrow \sqrt{a_{ii}}$ (this is r_{ii})
  for $j = i+1, \ldots, n$ (not executed when $i = n$)
    for $k = 1, \ldots, i-1$ (not executed when $i = 1$)
      $a_{ij} \leftarrow a_{ij} - a_{ki}a_{kj}$
    $a_{ij} \leftarrow a_{ij}/a_{ii}$ (this is r_{ij})

Proposition Cholesky's algorithm applied to an $n \times n$matrix performs about $\frac{n^3}{3}$ flops.

SYSTEM OF LINEAR EQUATIONS

POSITIVE DEFINITE SYSTMS; CHOLESKY DECOMPOSTION

Proposition Cholesky's algorithm applied to an $n \times n$matrix performs about $\frac{n^3}{3}$ flops.

Examining the algorithm, we see that in each of the two k loops, two flops are performed.

To see how many times each loop is executed, we look at the limits on the loop indices.

We conclude that the number of flops attributable to the first of the k loops is

$$\sum_{i=1}^{n} 2(i-1) = 2\sum_{i=1}^{n} (i-1) = 2\left(\frac{n(n-1)}{2}\right) = n(n-1) \approx n^2 .$$

Cholesky's Algorithm (inner-product form)

for $i = 1, \ldots, n$
  for $k = 1, \ldots, i-1$ (not executed when $i = 1$)
    $a_{ii} \leftarrow a_{ii} - a_{ki}^2$
  if $a_{ii} \leq 0$, set error flag (A is not positive definite), exit
  $a_{ii} \leftarrow \sqrt{a_{ii}}$ (this is r_{ii})
  for $j = i+1, \ldots, n$ (not executed when $i = n$)
    for $k = 1, \ldots, i-1$ (not executed when $i = 1$)
      $a_{ij} \leftarrow a_{ij} - a_{ki}a_{kj}$
    $a_{ij} \leftarrow a_{ij}/a_{ii}$ (this is r_{ij})

Applying the same procedure to the second of the k loops, we get a flop count of

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}\sum_{k=1}^{i-1} 2 = 2\sum_{i=1}^{n}\sum_{j+1}^{n} (i-1) = 2\sum_{i=1}^{n} (n-i)(i-1)$$

$$= 2n\sum_{i=1}^{n} (i-1) - 2\sum_{i=1}^{n} i^2 + 2\sum_{i=1}^{n} = 2n\left(\frac{n(n-1)}{2}\right) - 2\left(\frac{n(n-1)(2n-1)}{6}\right) = n^3 - \frac{2n^3}{3} + O(n^2) \approx \frac{n^3}{3} .$$

SYSTEM OF LINEAR EQUATIONS

POSITIVE DEFINITE SYSTMS; CHOLESKY DECOMPOSTION

We start with:

$$2n\sum_{i=1}^{n}(i-1)-2\sum_{i=1}^{n}i^2+2\sum_{i=1}^{n}i$$

1. First summation

2. $$(\sum_{i=1}^{n}(i-1)):\sum_{i=1}^{n}(i-1)=\sum_{i=1}^{n}i-n=\frac{n(n+1)}{2}-n=\frac{n(n-1)}{2}$$

Thus, the first term becomes:

$$2n\cdot\frac{n(n-1)}{2}=n^2(n-1)$$

2. Second summation

$$\sum_{i=1}^{n}i^2:\sum_{i=1}^{n}i^2=\frac{n(n+1)(2n+1)}{6}$$

So the second term becomes:

$$-2\cdot\frac{n(n+1)(2n+1)}{6}=-\frac{n(n+1)(2n+1)}{3}$$

3. Third summation

$$\sum_{i=1}^{n}i:\sum_{i=1}^{n}i=\frac{n(n+1)}{2}$$

Thus, the third term becomes:

$$2\cdot\frac{n(n+1)}{2}=n(n+1)$$

Final expression:

Putting everything together:

$$n^2(n-1)-\frac{n(n+1)(2n+1)}{3}+n(n+1)$$

SYSTEM OF LINEAR EQUATIONS

Gaussian Elimination

We want to solve the linear system **Ax = b** by Gaussian Elimination

A is not supposed to Special Properties such Symmetry and Positive Definiteness

Our strategy will be to transform the system $Ax = b$ to an equivalent system $Ux = y$ whose coefficient matrix is upper triangular. The system $Ux = y$ can then be solved easily by back substitution if U is nonsingular. To say that two systems are *equivalent* is to say that they have the same solutions.

We will transform the system by means of *elementary operations* of three types:

1. Add a multiple of one equation to another equation.
2. Interchange two equations.
3. Multiply an equation by a nonzero constant.

SYSTEM OF LINEAR EQUATIONS

Gaussian Elimination

It is convenient to represent the system $Ax = b$ by an augmented matrix $[A \mid b]$. Each equation in the system $Ax = b$ corresponds to a row of the matrix $[A \mid b]$. The elementary operations on the equations amount to the following *elementary row operations* on $[A \mid b]$:

1. Add a multiple of one row to another row.
2. Interchange two rows.
3. Multiply a row by a nonzero constant.

One can also apply elementary row operations to A alone, leaving off the vector b.

SYSTEM OF LINEAR EQUATIONS

Gaussian Elimination

Gaussian Elimination without Row Interchanges

We will approach the problem of solving $Ax = b$ in two stages. In the first stage, which will occupy us for the rest of this section, we will assume that A satisfies a special property that makes it possible to transform A to upper triangular form using elementary row operations of type 1 only. The algorithm that we will derive for carrying out this transformation is called *Gaussian elimination without row interchanges* or *Gaussian elimination without pivoting*.

ASSUMPTION

$$A_k \text{ is nonsingular} \qquad k = 1, \ldots, n.$$

A_k obtained by intersecting the first k rows and columns of A.

In particular, $A\ (= A_n)$ is itself nonsingular, so the system $Ax = b$ has a unique solution.

SYSTEM OF LINEAR EQUATIONS

Gaussian Elimination

Gaussian Elimination without Row Interchanges

The reduction to triangular form is carried out in $n-1$ steps. In the first step appropriate multiples of the first row are subtracted from each of the other rows to create zeros in positions $(2,1), (3,1), \ldots, (n,1)$. It is clear that in order to do this we must have $a_{11} \neq 0$. This is guaranteed by the assumption that $A_1 = [a_{11}]$ is nonsingular. The appropriate multiplier for the ith row is

$$m_{i1} = a_{i1}/a_{11} \qquad i = 2, \ldots, n.$$

Thus we carry out the operations

$$\begin{aligned} a_{ij}^{(1)} &= a_{ij} - m_{i1}a_{1j} \qquad j = 2, \ldots, n, \; i = 2, \ldots, n \\ b_i^{(1)} &= b_i - m_{i1}b_1 \qquad\qquad\qquad\quad i = 2, \ldots, n \end{aligned}$$

SYSTEM OF LINEAR EQUATIONS

Gaussian Elimination

Gaussian Elimination without Row Interchanges

to reduce $[A \mid b]$ to the form

$$\left[\begin{array}{c|ccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ \hline 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{array}\right].$$

SYSTEM OF LINEAR EQUATIONS

Gaussian Elimination

SYSTEM OF LINEAR EQUATIONS

Gaussian Elimination