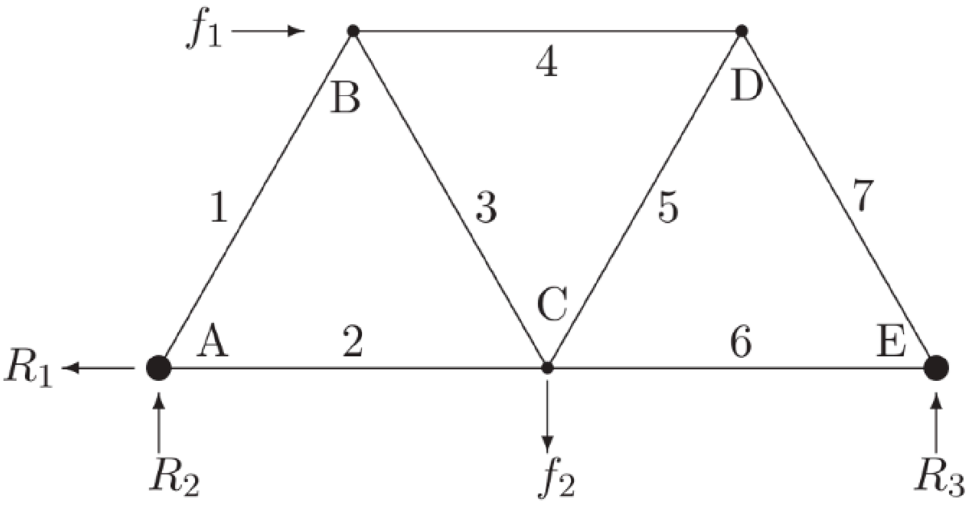
System of Linear Equations

Systems of linear equations are very important in applied and numerical mathematics. Since engineers are dealing mostly with linear models, this requires solving systems of linear equations in engineering problems.

Consider a problem in statics. Let us analyze the forces on a truss with equilateral triangles as shown on the figure below.



Compute member forces by the method of joints. Make the following approximations: $\cos(60°) = 0.5$ and $\sin(60°) = \frac{\sqrt{3}}{2} \approx 0.8660$ ($f_1 = 1000\,N$ and $f_2 = 5000\,N$).

Equations of equilibrium of rectangular components of forces and moments are used to determine reactions at supports (R_1, R_2, R_3). The following system of linear equations can be formulated by considering forces at nodes, where T_i represent tension in members:

$$0.5\,T_1 + T_2 = R_1 = f_1$$

$$0.866\,T_1 = -R_2 = 0.433\,f_1 - 0.5\,f_2$$

$$-0.5\,T_1 + 0.5\,T_3 + T_4 = -f_1$$

$$0.866\,T_1 + 0.866\,T_3 = 0$$

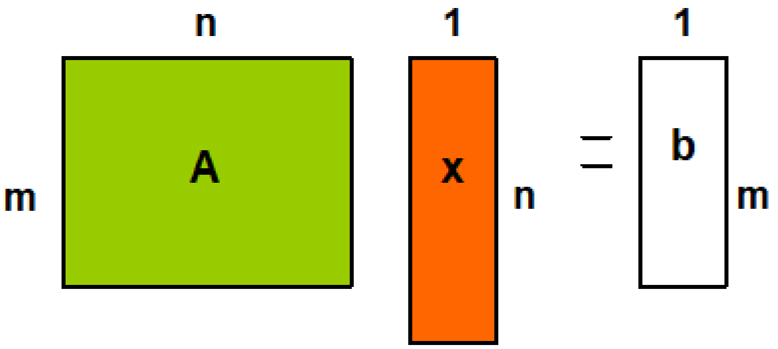
$$-T_2 - 0.5\,T_3 + 0.5\,T_5 + T_6 = 0$$

$$0.866\,T_3 + 0.866\,T_5 = f_2$$

$$-T_4 - 0.5\,T_5 + 0.5\,T_7 = 0$$

The above linear system is equivalent to the matrix equation $\mathrm{Ax} = b$.

In general form:

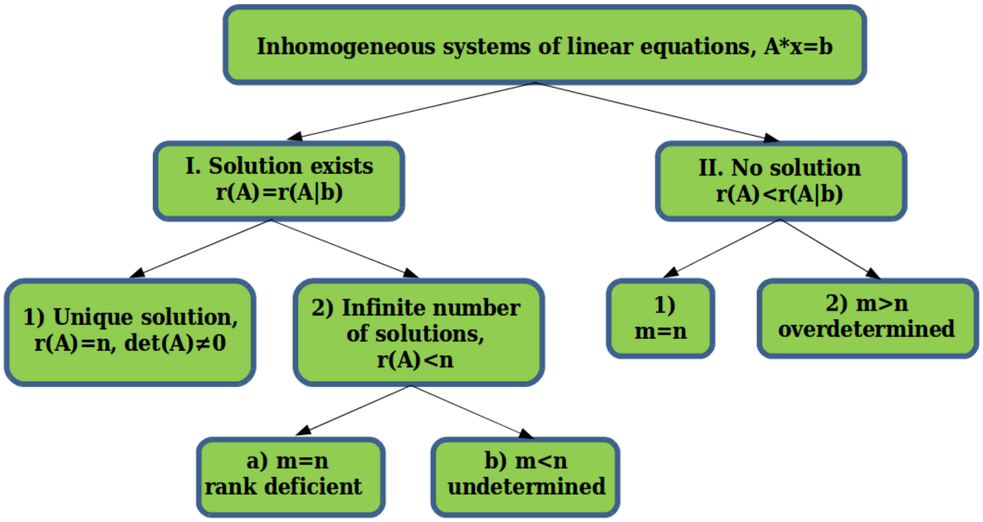


$$A = \begin{pmatrix} 0.5 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0.866 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.5 & 0 & 0.5 & 1 & 0 & 0 & 0 \\ 0.866 & 0 & 0.866 & 0 & 0 & 0 & 0 \\ 0 & -1 & -0.5 & 0 & 0.5 & 1 & 0 \\ 0 & 0 & 0.866 & 0 & 0.866 & 0 & 0 \\ 0 & 0 & 0 & -1 & -0.5 & 0 & 0.5 \end{pmatrix}, b = \begin{pmatrix} f_1 \\ 0.433\, f_1 - 0.5\, f_2 \\ -f_1 \\ 0 \\ 0 \\ f_2 \\ 0 \end{pmatrix}$$

Existence and uniqueness of solution

The system of equations can be inhomogeneous (*b ≠ 0*) or homogeneous (*b* = 0). When the system is homogeneous, we have a nontrivial solution (*x* ≠ 0) only if the determinant of A is zero: **det**(A)=0.

Discussion of solutions when the system is inhomogeneous:


Inhomogeneous systems of linear equations, A*x=b
I. Solution exists r(A)=r(A|b)
II. No solution r(A)<r(A|b)
1) Unique solution, r(A)=n, det(A)≠0
2) Infinite number of solutions, r(A)<n
1) m=n
2) m>n overdetermined
a) m=n rank deficient
b) m<n undetermined

Solution exists and is unique

Solution exists, if rank of matrix *A* (number of its linearly independent columns, *r*(*A*)) is the same as the rank of the augmented matrix *A*|*b*, i.e. r(A)=r(A|b). The augmented matrix is obtained by appending column vector *b* to matrix *A*.

The solution is unique, if r(A)=r(A|b)=n or det(A) ≠ 0, that is rank of matrix *A* is equal to the rank of the augmented matrix and this rank is full (equals to the number of columns).

Load A and b from file **`truss.txt`**.

```
Ab=[0.500000000000000,1,0,0,0,0,0,1000;...
    0.866000000000000,0,0,0,0,0,0,-2067;...
    -0.500000000000000,0,0.500000000000000,1,0,0,0,-1000;...
    0.866000000000000,0,0.866000000000000,0,0,0,0,0;...
    0,-1,-0.500000000000000,0,0.500000000000000,1,0,0;...
    0,0,0.866000000000000,0,0.866000000000000,0,0,5000;...
    0,0,0,-1,-0.500000000000000,0,0.500000000000000,0];
% Ab = load('truss.txt')
A = Ab(:,1:end-1)
b = Ab(:,end)
```

Check whether we have a unique solution

```
rank(A), rank([A b]) % 7=7,  solution exists and unique: n=7
```

Check whether the determinant is non-zero:

```
det(A) % =-0.3247 – we have a unique solution
```

Solution methods in matlab if there is a unique solution

Both direct and iterative methods may be used for solving systems of linear equations. Let us consider now direct methods. Consider four possibilities:

```
x1 = inv(A)*b; % computation of inverse (time consuming)
x2 = A\b; % nxn: Cholesky or LU factorization; mxn: QR factorization
x2 = mldivide(A,b); % the same as above
x3 = pinv(A)*b; % pseudoinverse by SVD factorization
x4 = linsolve(A,b); % LU or QR factorization can be specified
[x1 x2 x3 x4] % results are shown side by side
```

Check our solutions by computing the length (norm - function **norm**) of the residual vector if it is zero or is within the required tolerance.

```
norm(A*x1-b) % 1.0904e-12
norm(A*x2-b) % 5.5695e-13
norm(A*x3-b) % 5.1348e-12
norm(A*x4-b) % 6.0157e-13
```

The most precise method is **`x2 = A\b`**.

Matrix decompositions

Solution of linear equations by LU decomposition

LU decomposition for **`A`** consists of three matrices (**`L`**, **`U`**, **`P`**– permutation matrix) such that

$P \cdot A = L \cdot U$.

Matlab will produce an LU decomposition with pivoting for a matrix `A`a with the following command:

```
[L U P] = lu(A)
```

To solve $\mathrm{A}x = b$ we first multiply both sides with the permutation (pivot) matrix:

$P \cdot A \cdot x = P \cdot b = d$.

Substituting $L \cdot U$ for $P \cdot A$ we get

$$L \cdot U \cdot x = d$$

Then we define $y = U \cdot x$, which is unknown since *x* is unknown. Using forward substitution, we can (easily) solve

- $L \cdot y = d$ (for *y*, where L is lower triangular and $d = P \cdot b$)

and then using back substitution we can (easily) solve

- $U \cdot x = y$(for *x*, where U is upper triangular)

Solve the equation system of the truss problem by LU decomposition

```
[L U P] = lu(A)
d = P*b;
opt1.LT=true
y = linsolve(L,d,opt1);
opt2.UT=true
x = linsolve(U,y,opt2)
residual = norm(A*x - b) % check
```

Cholesky decomposition

Cholesky decomposition is very similar to LU decomposition: $A = L^T \cdot L$.

Steps of the solution using $A \cdot x = L^T \cdot L \cdot x = L^T \cdot y = b$:

- $L^T \cdot y = b$ (where L^T is lower triangular)
- $L \cdot x = y$ (where *L* is upper triangular)

Conditions:

- Matrix *A* is symmetric, i.e. $A^T = A$ and
- positive definite, i.e. it has got only positive eigenvalues

Matrix A of the truss problem is symmetric and positive definite:

```
norm(A-A') %  1.5946
```

```
real(eig(A)) % [0.5000; -0.7136; -0.7136; -0.7136; 1.2136; 1.2136; 1.2136]
%chol(A)  % issues an error message: Matrix must be positive definite
```

Instead consider the following symmetric and positive definite matrix:

```
A = pascal(4) % Pascal matrix composed of binomial coefficients
norm(A-A') % symmetric because A = A’
[V D] = eig(A) % compute eigenvectors and eigenvalues
min(diag(D)) % positive definite since even the smallest eigenvalue >0
L = chol(A) % Cholesky decomposition is allowed
norm(A-L'*L) % check
b = rand(4,1) % arbitrary vector b
```

Solution of a linear system is similar to LU decomposition (there is no permutation, hence instead of d we put vector b, put L' in place of L and put L in place of U):

```
opt1.LT=true
y = linsolve(L',b,opt1);
opt2.UT=true
x = linsolve(L,y,opt2)
residual = norm(A*x - b) % check
```