**modul.m**

function modul = modul(liczba)

%MODUL Wartosc bezwzgledna skalara

% Detailed explanation goes here

if liczba < 0

modul = liczba \* -1;

else

modul = liczba;

end

end

**znajdzMax.m**

function [wartosc, pozycja] = znajdzMax(wektor)

%ZNAJDZMAX Wyszukuje maksymalna wartość w wektorze co do modulu

% Detailed explanation goes here

n = length(wektor);

if n == 0

wartosc = -1;

pozycja = -1;

else

wartosc = modul(wektor(1));

pozycja = 1;

for i = 2:n

if modul(wektor(i)) > wartosc

wartosc = modul(wektor(i));

pozycja = i;

end

end

end

end

**zamianaWierszy.m**

function [A, b] = zamianaWierszy(A , b, numerWierszaA, numerWierszaB)

%ZAMIANAWIERSZY Zamienia wiersze A i B w macierzach stanowiących równanie

% Detailed explanation goes here

x = numerWierszaA;

y = numerWierszaB;

n = length(A);

tmp = A(x, :);

A(x, :) = A(y, :);

A(y, :) = tmp;

tmp = b(x);

b(x) = b(y);

b(y) = tmp;

end

**eliminacjaGaussa.m**

function X = eliminacjaGaussa(A ,b)

%ELIMINACJAGAUSSA Summary of this function goes here

% Detailed explanation goes here\

n = length(b);

% Doprowadzenie do macierzy trójkątnej

for k = 1:n % Dla każdej kolumny

doDolu = A(k:n, k);

[~, indexMaksa] = znajdzMax(doDolu);

indexMaksa = indexMaksa + k - 1;

[A, b] = zamianaWierszy(A, b, k, indexMaksa);

diag = A(k, k); % Element na diagonali

for i = (k+1):n % Dla każdego elementu pod diagonalem

mul = A(i, k) / diag;

for j = k:n % Dla każdego elemetnu z aktualnego wiersza od elementu pod diag do końca

A(i, j) = A(i, j) - A(k, j)\*mul;

end

% Zmiana b

b(i) = b(i) - b(k)\*mul;

end

end

%Policzenie x-ów

X = zeros(n, 1);

for i = n:-1:1

suma = 0;

for j = (i+1):n

suma = suma + A(i, j) \* X(j);

end

X(i) = (b(i) - suma) / A(i, i);

end

end

**rozkladLU.m**

function [L, R] = rozkladLU(A)

%ROZKLADLU Metodą Doolittle'a

% Tworzy macierze L i R dolno i górno trójkątne, takie że A = L\*R

% Wersja z dodatkowymi macierzami (można też nadpisywać A)

n = length(A);

L = diag(ones(1, n));

R = zeros(n);

R(1, :) = A(1, :);

L(2:n, 1) = A(2:n, 1) / R(1,1);

for k = 2:n

for i = k:n

suma = 0;

for j = 1:(k-1)

suma = suma + L(k, j)\*R(j, i);

end

R(k, i) = A(k, i) - suma;

end

for i = k:n

suma = 0;

for j = 1:(k-1)

suma = suma + L(i, j)\*R(j, k);

end

L(i, k) = (A(i, k) - suma) / R(k, k);

end

end

end

**rozwiazUkladL.m**

function y = rozwiazUkladL(L, b)

%ROZWIAZUKLADL Rozwiązuje układ równań macierzowych postaci Ly = b

% L jest macierzą dolno trójkątną z 1 na diagonali

n = length(L);

y = zeros(n, 1);

for i = 1:n

suma = 0;

for j = 1:(i-1)

suma = suma + L(i, j)\*y(j);

end

y(i) = (b(i) - suma) / L(i, i);

end

end

**rozwiazUkladR.m**

function x = rozwiazUkladR(R, y)

%ROZWIAZUKLADL Rozwiązuje układ równań macierzowych postaci Rx = y

% R jest macierzą górno trójkątną ze zwykłymi wartościami na diagonali

n = length(R);

x = zeros(n, 1);

for i = n:-1:1

suma = 0;

for j = (i+1):n

suma = suma + R(i, j)\*x(j);

end

x(i) = (y(i) - suma) / R(i, i);

end

end

**metodaLU.m**

function X = metodaLU(A, b)

%METODALU Rozwiązuje układ równań metodą LU

[L, R] = rozkladLU(A);

y = rozwiazUkladL(L, b);

X = rozwiazUkladR(R, y);

end

**metodaGaussaJordana.m**

function X = metodaGaussaJordana(A, b)

[n, m] = size(b);

% Doprowadzenie do macierzy jednostkowej

for k = 1:n % Dla każdej kolumny

diag = A(k, k); % Element na diagonali

if diag ~= 1

for i = k:n

A(k, i) = A(k, i) / diag;

end

for i = 1:m

b(k,i) = b(k,i) / diag;

end

end

for i = 1:n

if i ~= k % Dla każdego elementu nad i pod diagonalem

mul = A(i, k);

for j = k:n % Dla każdego elemetnu z aktualnego wiersza od elementu pod diag do końca

A(i, j) = A(i, j) - A(k, j)\*mul;

end

% Zmiana b

for j = 1:m

b(i, j) = b(i, j) - b(k, j)\*mul;

end

end

end

end

X = b;

end

**odwracanieMetodaGaussaJordana.m**

function macierzOdwrocona = odwracanieMetodaGaussaJordana(A)

B = eye(size(A));

macierzOdwrocona = metodaGaussaJordana(A, B);

end

**main.m**

%% Testy Gauss

clc

A = [2 1 5 -1.5; 5 3 2 4 ; -2 3 2 -4; 4.5 3 5 2];

b = [1.7 47.3 -46.4 34.95]';

X = eliminacjaGaussa(A, b);

%% Testy LU

clc

A = [2 1 5 -1.5; 5 3 2 4 ; -2 3 2 -4; 4.5 3 5 2];

b = [1.7 47.3 -46.4 34.95]';

x = metodaLU(A, b);

%% Testy Gaussa-Jordana

clc

A = [1 2 3; 4 3 -1; 1 -1 1];

b = [14 7 2]';

y = metodaGaussaJordana(A, b);

%% Testy odwracania GJ

A = [8 5; 13 8];

y2 = odwracanieMetodaGaussaJordana(A);