**Project 3 Dimitrije Prosevski MATH 414**

**Source Code:**

clear

clc

format longG

N = [5, 10, 20]; %matrix size

fprintf("For N = %d\n", N(1));

a = [1;1;1;1]; %sub

b = [1,2,3,4,5]; %right vector

c = [2;2;2;2]; %sup

d = [3;3;3;3;3]; %diag

triD(a, d, c, b, N(1));

fprintf("\n\n\n\n\nFor N = %d\n", N(2));

a = [1;1;1;1;1;1;1;1;1]; %sub

b = [1,2,3,4,5,6,7,8,9,10]; %right vector

c = [2;2;2;2;2;2;2;2;2]; %sup

d = [3;3;3;3;3;3;3;3;3;3]; %diag

triD(a, d, c, b, N(2));

fprintf("\n\n\n\n\nFor N = %d\n", N(3));

a = [1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1]; %sub

b = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]; %right vector

c = [2;2;2;2;2;2;2;2;2;2;2;2;2;2;2;2;2;2;2]; %sup

d = [3;3;3;3;3;3;3;3;3;3;3;3;3;3;3;3;3;3;3;3]; %diag

triD(a, d, c, b, N(3));

% tridiagonal function

function x = triD(a, d, c, b, n)

bInitial = b;

%output A matrix

A = zeros(n);

for i = 1:n

for j=1:n

if i == j + 1

A(i,j) = a(1);

elseif i == j

A(i,j) = d(j);

elseif i == j - 1

A(i,j) = c(1);

end

end

end

fprintf("Matrix A:\n");

disp(A);

%output b right hand vector

fprintf("Vector b = [ ");

for i = 1:length(b)

fprintf("%d ", b(i));

end

fprintf("]\n");

%tridiagonal algorithm

for i = 2:n

xMult = a(i-1) / d(i-1);

d(i) = d(i) - xMult\*c(i-1);

b(i) = b(i) - xMult\*b(i-1);

end

x(n) = b(n)/d(n);

for i = n-1:-1:1

x(i) = (b(i) - c(i) \* x(i+1))/d(i);

end

%output x vector

fprintf("\nCalculated vector x");

fprintf("\nVector x = [ ");

for i = 1:length(x)

if (i == 11) %new line so the output can fit

fprintf("\n\t\t\t");

end

fprintf("%1.10f ", x(i));

end

fprintf("]\n");

%residual r=Ax-b

fprintf("\nResidual r=Ax-b");

r = mtimes(A, x') - bInitial';

%output r vector

fprintf("\nVector r = [ ");

for i = 1:length(r)

if (i == 11) %new line so the output can fit

fprintf("\n\t\t\t");

end

fprintf("%1.10f ", r(i));

end

fprintf("]\n");

end %end tridiagonal function

**Output:**

**For N = 5**

**Matrix A:**

**3 2 0 0 0**

**1 3 2 0 0**

**0 1 3 2 0**

**0 0 1 3 2**

**0 0 0 1 3**

**Vector b = [ 1 2 3 4 5 ]**

**Calculated vector x**

**Vector x = [ 0.6190476190 -0.4285714286 1.3333333333 -0.2857142857 1.7619047619 ]**

**Residual r=Ax-b**

**Vector r = [ 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 ]**

**For N = 10**

**Matrix A:**

**3 2 0 0 0 0 0 0 0 0**

**1 3 2 0 0 0 0 0 0 0**

**0 1 3 2 0 0 0 0 0 0**

**0 0 1 3 2 0 0 0 0 0**

**0 0 0 1 3 2 0 0 0 0**

**0 0 0 0 1 3 2 0 0 0**

**0 0 0 0 0 1 3 2 0 0**

**0 0 0 0 0 0 1 3 2 0**

**0 0 0 0 0 0 0 1 3 2**

**0 0 0 0 0 0 0 0 1 3**

**Vector b = [ 1 2 3 4 5 6 7 8 9 10 ]**

**Calculated vector x**

**Vector x = [ -0.7782120176 1.6673180264 -1.1118710308 2.3341475330 -0.9452857841 2.7508549096 -0.6536394724 3.1050317538 -0.3307278945 3.4435759648 ]**

**Residual r=Ax-b**

**Vector r = [ 0.0000000000 -0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 ]**

**For N = 20**

**Matrix A:**

**3 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

**1 3 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

**0 1 3 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

**0 0 1 3 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

**0 0 0 1 3 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

**0 0 0 0 1 3 2 0 0 0 0 0 0 0 0 0 0 0 0 0**

**0 0 0 0 0 1 3 2 0 0 0 0 0 0 0 0 0 0 0 0**

**0 0 0 0 0 0 1 3 2 0 0 0 0 0 0 0 0 0 0 0**

**0 0 0 0 0 0 0 1 3 2 0 0 0 0 0 0 0 0 0 0**

**0 0 0 0 0 0 0 0 1 3 2 0 0 0 0 0 0 0 0 0**

**0 0 0 0 0 0 0 0 0 1 3 2 0 0 0 0 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0 1 3 2 0 0 0 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0 0 1 3 2 0 0 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0 0 0 1 3 2 0 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0 0 0 0 1 3 2 0 0 0 0**

**0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 3 2 0 0 0**

**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 3 2 0 0**

**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 3 2 0**

**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 3 2**

**0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 3**

**Vector b = [ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 ]**

**Calculated vector x**

**Vector x = [ -1.6111119323 2.9166678985 -2.5694458816 3.8958348731 -2.5590293689 4.3906266168 -2.3064252407 4.7643245527 -1.9932742087 5.1077490367**

**-1.6649864507 5.4436051577 -1.3329145112 5.7775691879 -0.9998965263 6.1110601955 -0.6666420301 6.4444329474 -0.3333284060 6.7777761353 ]**

**Residual r=Ax-b**

**Vector r = [ 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 -0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000**

**0.0000000000 0.0000000000 -0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000 ]**