**Звіт до лабораторної роботи 5**

**5 варіант**



function a = getA

a = [-0.68, -0.18, 0.02, 0.21;

0.16, -0.88, -0.14, 0.27;

0.37, 0.27, -1.02, -0.24;

0.12, 0.21, -0.18, -0.75];

end

function b = getB

b = [-1.83;

0.65;

-2.23;

1.13];

end

function [x, k] = yakobi(a, b, eps)

for i = 1:4

b(i) = b(i) / a(i, i);

for j = 1:4

if i ~= j

a(i, j) = - a(i, j) / a(i, i);

end

end

a(i, i) = 0;

end

x = [0, 0, 0, 0];

k = 0;

x0 = b';

if norm(a, 1) < 1

if norm(a, 1) > 0.5

eps = (1 - norm(a, 1))/norm(a, 1) \* eps;

end

while(norm(x0 - x) > eps && k < 100)

x0 = x;

k = k+1;

for i = 1:4

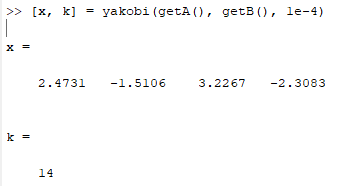
x(i) = dot(x0, a(i, :)) + b(i);

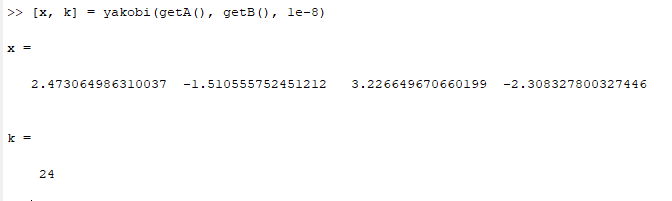
end

end

end

end





Якщо замінити нульове наближення на одиниці то отримаємо такий самий результат

function [x, k] = zedel(a, b, eps)

for i = 1:4

b(i) = b(i) / a(i, i);

for j = 1:4

if i ~= j

a(i, j) = - a(i, j) / a(i, i);

end

end

a(i, i) = 0;

end

k = 0;

x = [0, 0, 0, 0];

x0 = b';

x1 = x0;

if norm(a, 1) < 1

if norm(a, 1) > 0.5

eps = (1 - norm(a, 1))/norm(a, 1) \* eps;

end

while(norm(x0 - x) > eps && k < 100)

x0 = x;

k = k+1;

for i = 1:4

x(i) = dot(x1, a(i, :)) + b(i);

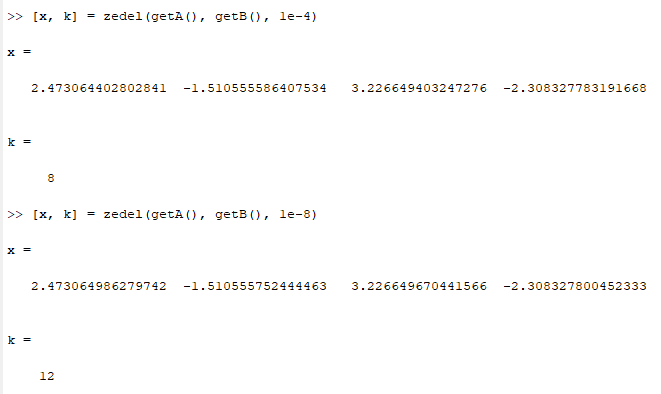
x1(i) = x(i);

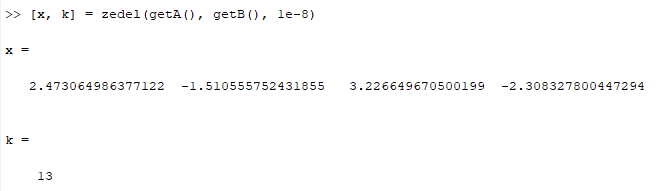
end

end

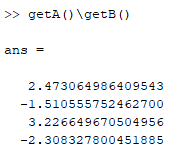
end

end



Якщо замінити нульове наближення в методі Зеделя, то зміниться результат наступним чином:  


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Eps = 0.1 | 0.01 | 0.001 | 0.0001 |
| Метод Якобі | K = 6  X =  2.4719 -1.5083 3.2295 -2.3048 | 8  2.4728 -1.5104 3.2270 -2.3079 | 11  2.4730 -1.5105 3.2266 -2.3083 | 14  2.4730 -1.5105 3.2266 -2.3083 |
| Метод Зеделя | 4  2.4 -1.5 3.2 -2.3 | 6  2.47 -1.51 3.22 -2.30 | 7  2.473 -1.510 3.226 -2.308 | 8  2.4730 -1.5105 3.2266 -2.3083 |



Висновок: метод Зеделя збігається значно краще ніж метод Якобі, особливо якщо нульове наближення брати не універсальним, а для заданої системи.