*Дубкова Валерия КС-26*

*Вариант 5*

**5-ая лабораторная работа**

**Код**

clc

% 2 \* exp(-3 \* x) - 3 \* x + 2

% x = -1: 0.0001 : 1;

f = @(x) 2\*exp(-3\*x) - 3\*x + 2;

df1 = @(x) -6 \* exp(-3 \* x) - 3; % -6 \* exp(-3 \* x) - 3

df2 = @(x) 18 \* exp(-3 \* x); % 18 \* exp(-3 \* x)

figure;

ezplot('2 \* exp(-3 \* x) - 3 \* x + 2', [-1 2 -2 5]);

grid on;

% интервал локализации, определенный графическим методом

a = -2;

b = 2;

x0 = fzero('2 \* exp(-3 \* x) - 3 \* x + 2', [a b]);

fprintf('Корень уравнения, полученный функцией fzero x = %f', x0);

hold on;

plot(x0, 0, '\*r');

xlabel('x'); ylabel('y'); legend('2 \* exp(-3x) - 3 \* x + 2', 'Корень уравнения');

% метод половинного деления

eps = 0.001;

k = 0;

c = 0;

while abs(a-b) > eps

c = (a+b) / 2;

if ((f(c)\*f(a))<0)

b = c;

else

a = c;

end

k = k + 1;

end

fprintf('\n\nМЕТОД ПОЛОВИННОГО ДЕЛЕНИЯ с точностью e = %.3f', eps);

fprintf('\nКоличество итераций: %d', k);

fprintf('\nИнтервал локализации корня [%f; %f]', a, b);

% метод перебора

h = 0.01;

x = 0;

k = 0;

while(f(x) \* f(x+h)) > 0 && x < 1

x = x + h;

t = x + h;

k = k + 1;

end

fprintf('\n\nМЕТОД ПЕРЕБОРА c шагом h = %.2f', h);

fprintf('\nКоличество итераци: %d', k);

fprintf('\nИнтервал локализации корня [%f; %f]', x, t);

% метод простых итераций

g = @(x) (2 \* exp(-3\*x) / 3) + (x / 3);

x0 = a;

x1 = g(x0);

eps = 0.1;

k = 0;

while abs(x1-x0) > eps

x0 = x1;

x1 = g(x0);

k = k+1;

end

fprintf('\n\nМЕТОД ПРОСТЫХ ИТЕРАЦИЙ с точностью e = %.1f', eps);

fprintf('\nКоличество итераций k=%d', k);

fprintf('\nКорень уравнения x=%.4f', x1);

% существует два корня: один в районе 0.7392 и другой в районе 0.3624

fprintf('\nПроверка (погрешность): %f', f(x));

% метод хорд

eps = 0.00001;

k\_max = 1000;

k=0;

x = b;

z = a;

h = ((x-z)\*f(x))/(f(x)-f(z));

while abs(h) > eps

h = ((x-z)\*f(x))/(f(x)-f(z));

x = x-h;

k = k + 1;

end

fprintf('\n\nМЕТОД ХОРД с точностью e = %.5f', eps);

fprintf('\nКоличество итераций k=%d', k);

fprintf('\nКорень уравнения x=%.3f', x);

fprintf('\nПроверка (погрешность): %f', f(x));

% метод касательных

eps = 0.00001;

k=0;

if (f(b)\*subs(df2,b)) > 0

x = b;

else

x = a;

end

h = f(x) / df1(x);

while abs(h) > eps

h = f(x)/df1(x);

x = x-h;

k = k + 1;

end

fprintf('\n\nМЕТОД КАСАТЕЛЬНЫХ с точностью e = %.5f', eps);

fprintf('\nКоличество итераций k=%d', k);

fprintf('\nКорень уравнения x=%.3f', x);

fprintf('\nПроверка (погрешность): %f', f(x));

% метод секущих

eps = 0.00001;

k=0;

if (f(b)\*subs(df2,b)) > 0

x = b;

else

x = a;

end

while abs(x - x0) > eps

x0 = x;

x = x - (f(x)\*eps)/(f(x+eps)-f(x));

k = k + 1;

end

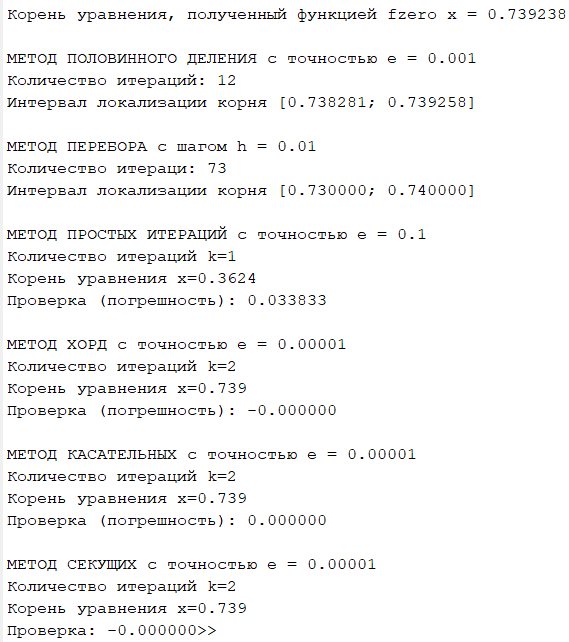
fprintf('\n\nМЕТОД СЕКУЩИХ с точностью e = %.5f', eps);

fprintf('\nКоличество итераций k=%d', k);

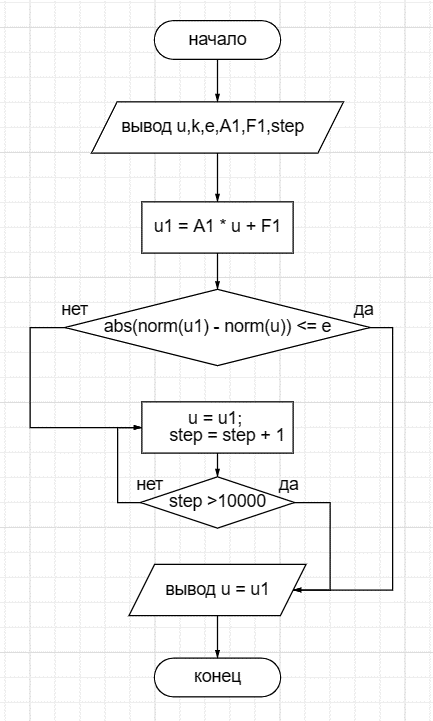
fprintf('\nКорень уравнения x=%.3f', x);

fprintf('\nПроверка: %f', f(x));

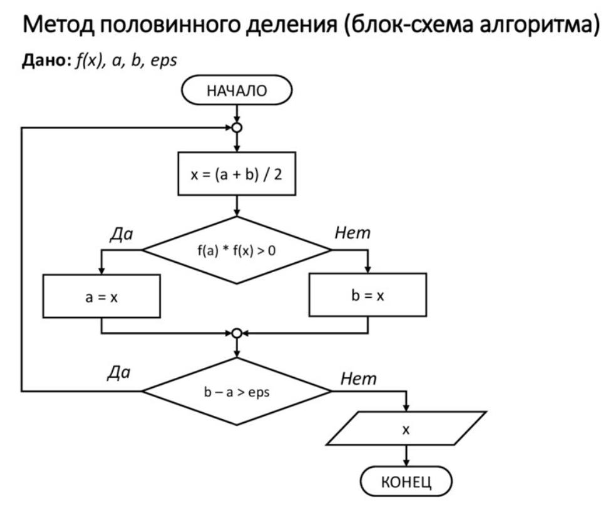
**Результат выполнения программы**

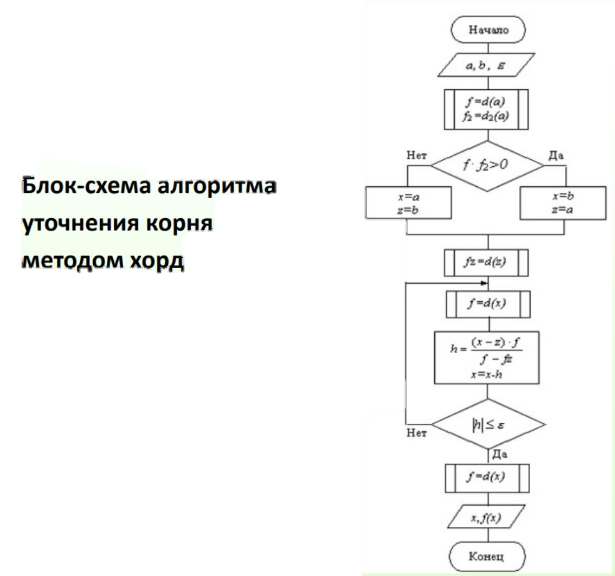


**Блок-схемы**



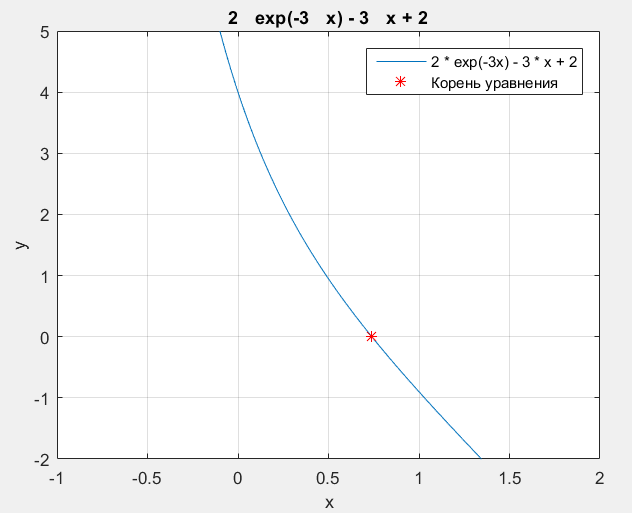
*Рис.1 Метод простых итераций*





*Рис.3 Алгоритм метода хорд*

**График**



**Выводы**

Как можно увидеть из результатов выполнения программы, все методы сошлись на одном и том же ответе, кроме метода простых итераций. Так как существует два корня: один в районе 0.7392 и другой в районе 0.3624. В методе простых итераций сходится тот корень, который ближе к начальному приближению x0.

Но по скорости выполнения методы практически сошлись, у всех их меньше 10.

Методы, которые локализовали корень, тратили для уточнения много времени. Метод простых итераций же работал не так точно, если задавать точность выше, поэтому полноценно проверить его не удалось.