ЛАБОРАТОРНАЯ РАБОТА 2.
РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

Евтеева Марина, БПМ-143

Вариант 7

2017

Оглавление

Задание 2.1.7 Даны два уравнения $f(x) = 0$ и $g(x) = 0$. Найти с точностью $\epsilon = 10^{-10}$ все корни уравнений, содержащиеся на отрезке $[a, b]$. Для решения задачи использовать метод бисекции. Найти корни с помощью встроенной функции.

Вариант 7

$$f(x) = (lnx)^2 - 5lnx + 6$$

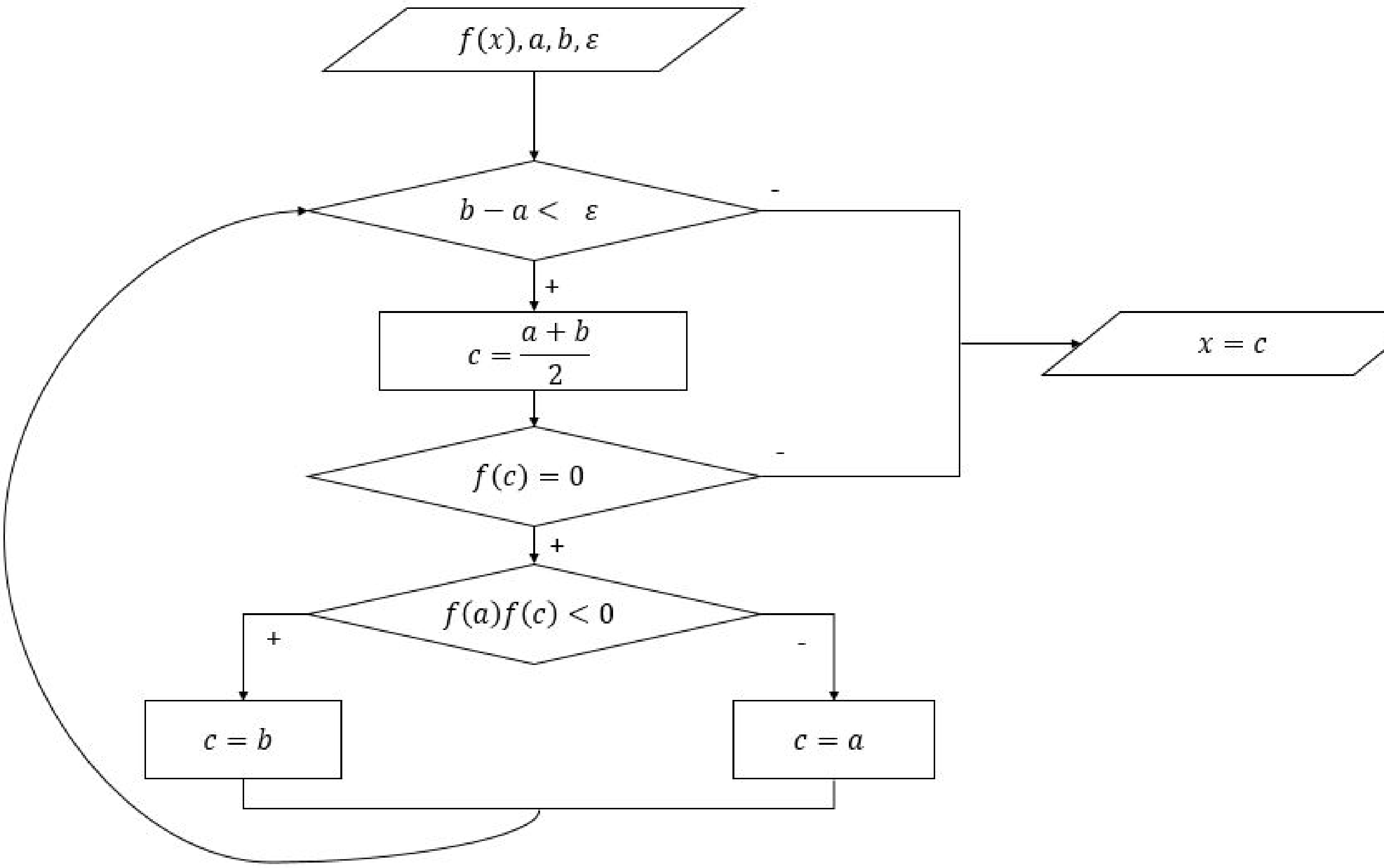
$g(x) = (lnx)^2 - 4lnx + 4$ на отрезке $[5, 25]$

Решение

Метод бисекции основывается на теореме Больцано — Коши, которая утверждает, что если функция f(x) непрерывна на отрезке [a,b] и на концах отрезков функция принимает значения разных знаков (f(a)<0<f(b) или f(b)<0<f(b)), то существует по крайней мере одна точка, в которой функция обращается в ноль.

Примечание: Теорема Больцано — Коши утверждает, что функция, непрерывная на отрезке, принимает все промежуточные значения.

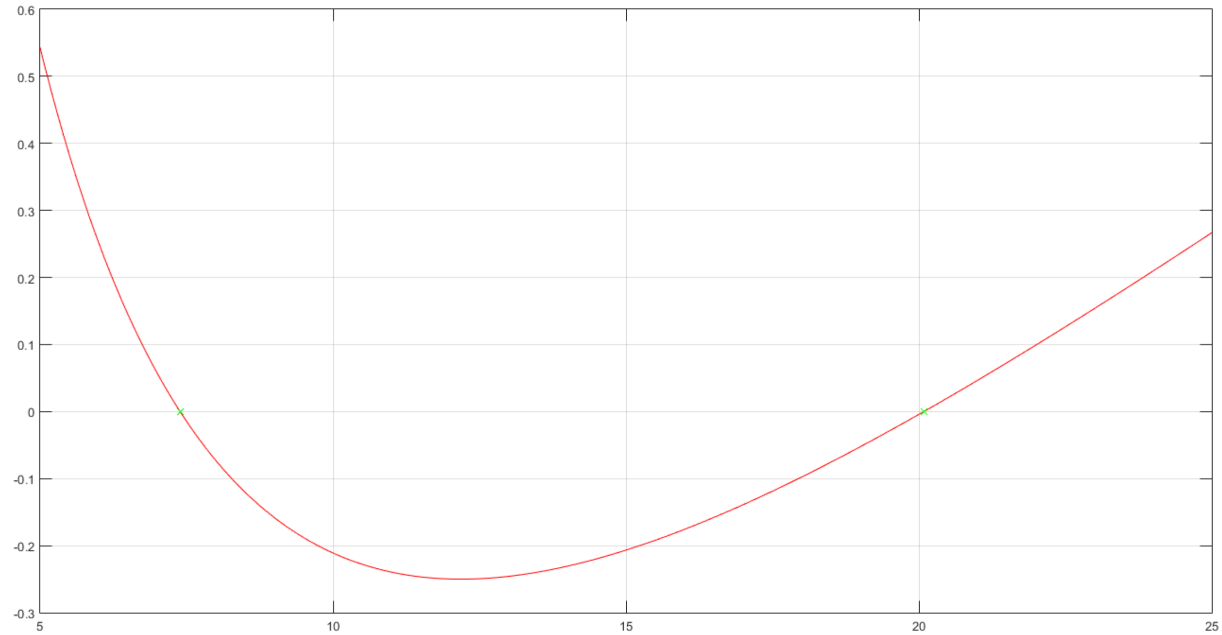
Алгоритм решения методом бисекции

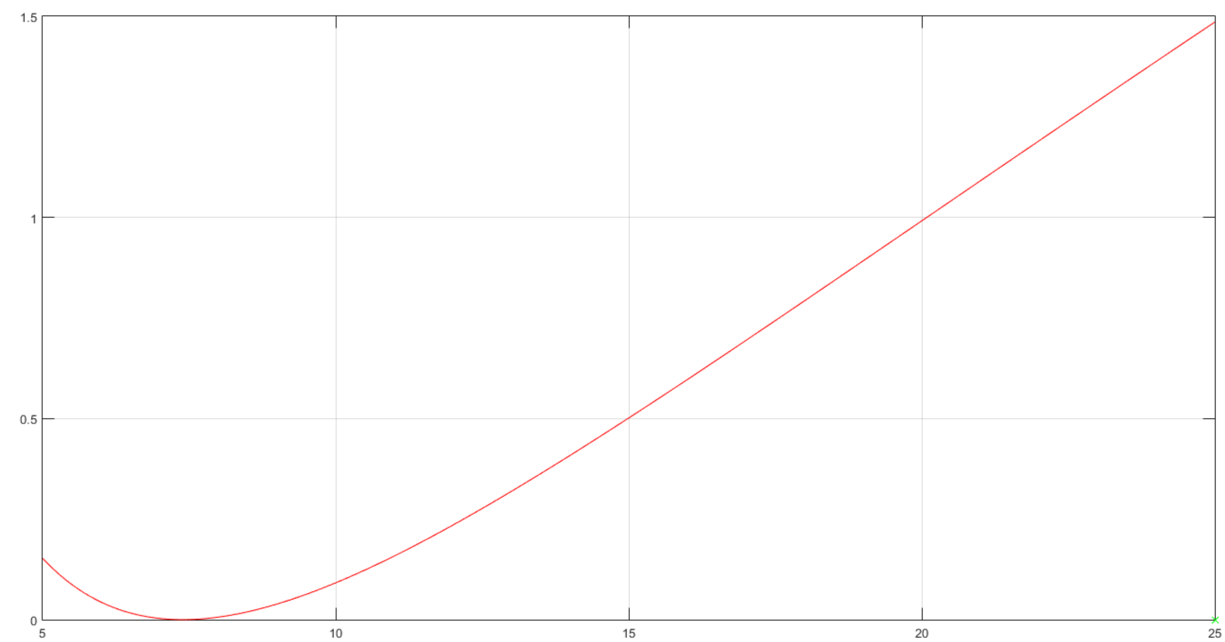


Аналитическое решение f(x):

$$(lnx)^2 - 5lnx + 6 = 0$$
$$y^2 - 5y + 6 = 0$$
$$y_1 = 3 \; y_2 = 2$$
$$x_1 = e^3 \; x_2 = e^2$$
$$x_1 \approx 20.085536 \; x_2 \approx 7.389056$$

Графики уравнений f(x) и g(x) соответственно:





Код программы на MATLAB

```
function [center, n] = bisec(f, left, right, eps)
n = 0;
while right - left > eps
    center = (right - left) / 2 + left;
    if f(center) * f(left) > 0
```

```
        left = center;
    else
        right = center;
    end
    n = n + 1;
end
end


format long
fun = @(x)(log(x).^2 - 5*log(x) + 6);
x = linspace(5, 25, 200);
plot(x, fun(x), 'r')
hold on
grid on
y1 = bisec(fun, 5, 25, 1e-10)
fzero(fun, 5)
y2 = bisec(fun, 15, 25, 1e-10)
fzero(fun, 15)
plot(y1, 0, 'gx')
plot(y2, 0, 'gx')

hold off
figure
fun2 = @(x)(log(x).^2 - 4*log(x) + 4);
plot(x, fun2(x), 'r')
hold on
grid on
y1 = bisec(fun2, 5, 25, 1e-10)
fzero(fun2, 5)
y2 = bisec(fun2, 15, 25, 1e-10)
fzero(fun2, 15)
plot(y1, 0, 'gx')
plot(y2, 0, 'gx')
```

Получившиеся значения корней для f(x) методом бисекции и встроенной функцией поиска корней:

Методом бисекции:

$x_1 = 7.389056098982110$

$x_2 = 20.085536923215841$

Встроенной функцией fzero:

$x_1 = 7.389056098930658$

$x_2 = 20.085536923187664$

Для g(x) данный метод не подходит, т.к. он опирается на то, что функция имеет разные знаки на разных концах интервала. В данном же случае она положительна на всем интервале.

Задание 2.2.2 Найти указанный в варианте корень уравнения $f(x) = 0$ с точностью $\epsilon = 10^{-6}$ двумя способами:

1. Использовать метод бисекции. Предварительно определить отрезок локализации [a, b].

2. Использовать метод Ньютона. В качестве начального приближения для метода Ньютона взять середину отрезка локализации из п.1

Сравнить число итераций в п. 1, 2.

Вариант 7

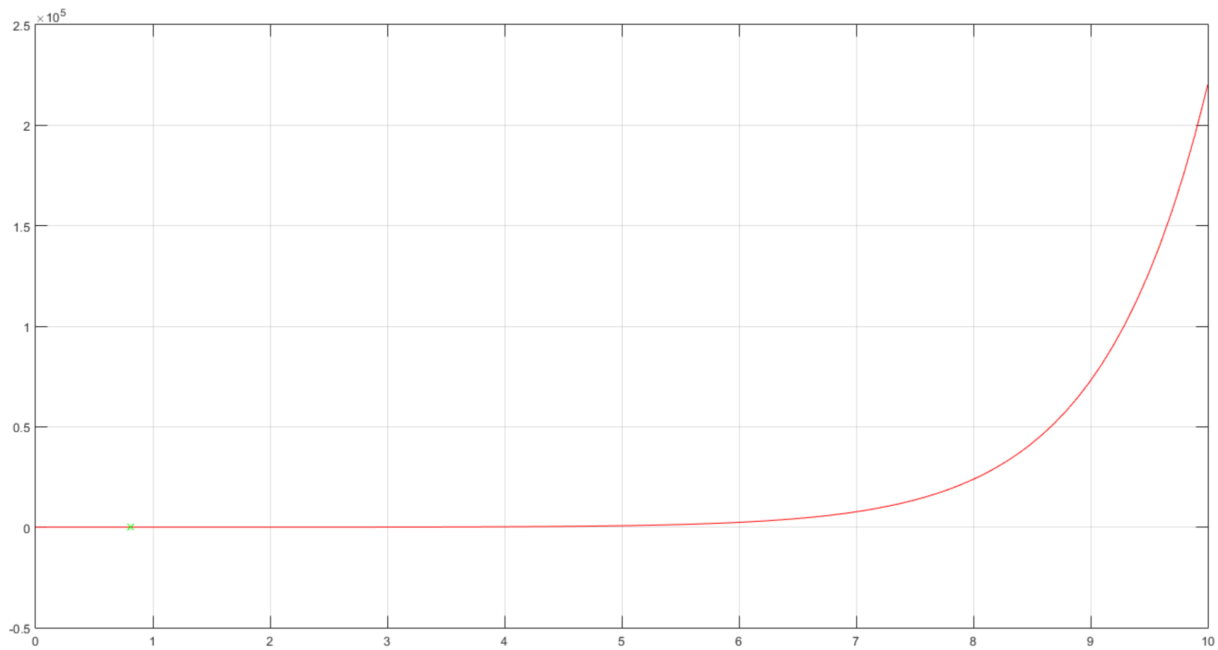
$f(x) = xe^x - x - 1$, ищем положительный корень

Решение

Про метод бисекции см. задачу 2.1.7.

Основная идея метода Ньютона заключается в следующем: задаётся начальное приближение вблизи предположительного корня, после чего строится касательная к графику исследуемой функции в точке приближения, для которой находится пересечение с осью абсцисс. Эта точка берётся в качестве следующего приближения. И так далее, пока не будет достигнута необходимая точность.

График функции $f(x)$. (с уже отмеченной точкой пересечения)



Код решения на MATLAB

```
function [center, n] = bisec(f, left, right, eps)
n = 0;
while right - left > eps
    center = (right - left) / 2 + left;
    if f(center) * f(left) > 0
        left = center;
    else
        right = center;
```

```
    end
    n = n + 1;
end
end

function [root, n] = newton(f, df, x0, eps)
root = x0 - f(x0) / df(x0);
n = 0;
while abs(root - x0) > eps
    x0 = root;
    root = root - f(root) / df(root);
    n = n + 1;
end

format long
fun = @(x)(x.*exp(x) - x - 1);
dfun = @(x)(exp(x)*(x + 1) - 1);
x = linspace(0, 10, 100);
plot(x, fun(x), 'r')
grid on
hold on
[y1, n1] = bisec(fun, 0, 10, 1e-6)
plot(y1, 0, 'gx')
[y2, n2] = newton(fun, dfun, 5, 1e-6)
plot(y2, 0, 'gx')
```

Поскольку мы ищем положительные корни, то для метода бисекции был выбран отрезок $[0, 10]$, в котором точно есть искомый корень. Для метода Ньютона стартовой точкой был $x_0 = 5$, как сказано в задании: середина отрезка из метода бисекции.

Методы дали очень близкие результаты:

Метод бисекции:

$x = 0.806465744972229$

Метод Ньютона:

$x = 0.806465994236327$

Но метод Ньютона нашел ответ за 10 шагов, а метод бисекции за 24. Это связано с тем, что метод бисекции сужает интервал всегда с одинаковой "скоростью поскольку делит отрезок пополам, а в методе Ньютона в данной задаче мы приближаемся к искомой точке с гораздо большей "скоростью".

Задание 2.10.3 Функция $y = f(x)$ задана неявно уравнением $F(x, y) = 0$. На отрезке $[1, 5]$ построить таблицу значений функции $y = f(x)$ с шагом $h = 0.5$, применяя один из методов численного решения нелинейного уравнения (с точностью $\epsilon = 10^{-7}$). Построить график функции $y = f(x)$ на заданном отрезке

Вариант 7

$F(x, y) = e^{xy} - \cos(xy^3)$, интервал $0.5 \leq x \leq 1.5, -1.3 \leq y \leq -0.3$

Решение

Код решения на MATLAB

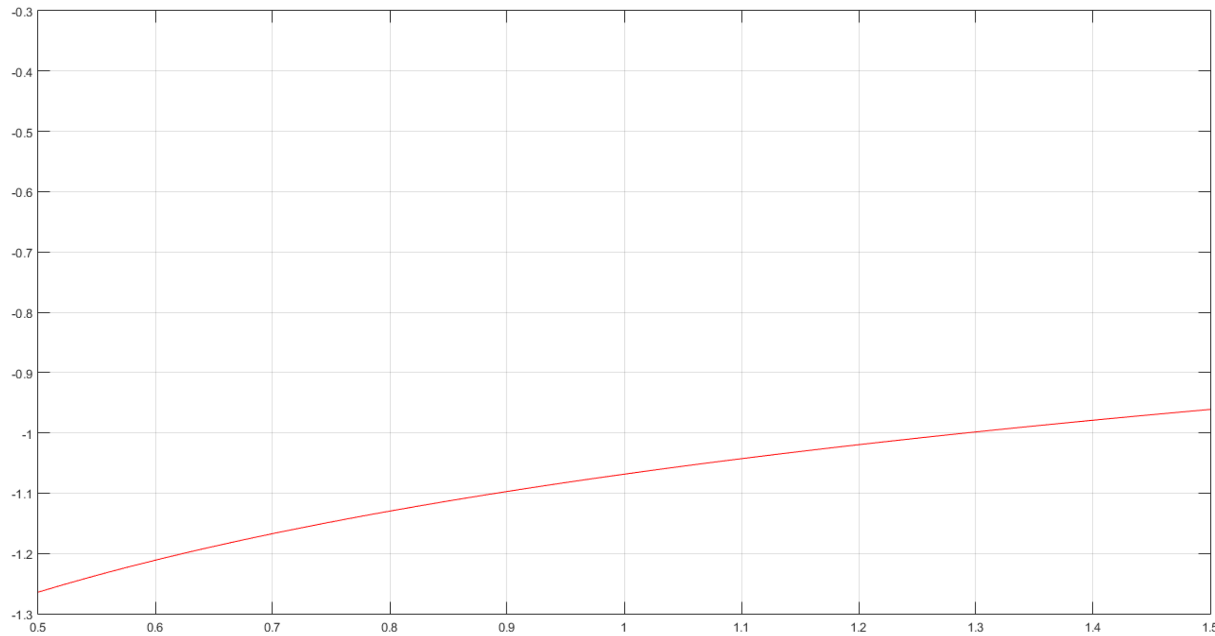
```
format long
x = 1:0.5:5;
y1 = [];

for i = 1:numel(x)
    y1 = [y1; fsolve(@(y)(exp(x(i)*y) - cos(x(i) * y^3)), -1, optimset('TolFun', 1e-7))]
end
f = @(x, y)(exp(x.*y) - cos(x .* y.^3));
figure
x2 = linspace(0.5, 1.5, 50);
y2 = [];
for i = 1:numel(x2)
    y2 = [y2; fsolve(@(y)(exp(x2(i)*y) - cos(x2(i) * y^3)), -1, optimset('TolFun', 1e-7)
end
plot(x2, y2)
axis([0.5 1.5 -1.3 -0.3])
```

Значения $y = f(x)$ для x из интервала $[1, 5]$ с шагом 0.5

X	Y
1	-1.06857585780955
1.5	-0.961185673644079
2	-0.888047533404283
2.5	-0.833169563882069
3	7.16312172301892e-16
3.5	-1.30863392582153
4	-1.05724073109191
4.5	-1.01623253229149
5	-0.980959769294884

График функции $y = f(x)$ на интервале $0.5 \leq x \leq 1.5, -1.3 \leq y \leq -0.3$


-0.3
-0.4
-0.5
-0.6
-0.7
-0.8
-0.9
-1
-1.1
-1.2
-1.3
0.5
0.6
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5