

Задача 1.

Величина критической силы для стержня, один конец которого закреплен, а другой может перемещаться в вертикальном направлении, определяется уравнением

$$\operatorname{tg} \sqrt{\frac{P}{E}} \cdot L = \sqrt{\frac{P}{E}} \cdot L,$$

где P – критическая сила, при которой стержень теряет устойчивость; E – изгибная жесткость стержня; L – длина стержня.

Определить величину минимальной критической силы, при которой стержень теряет устойчивость.

Решение.

Введем обозначение $x = \sqrt{\frac{P}{E}} \cdot L$. Тогда задача определения критической силы, при которой стержень теряет устойчивость, сводится к поиску первого положительного корня трансцендентного уравнения

$$x - \operatorname{tg} x = 0$$

Для решения воспользуемся методом половинного деления. Из графика функции $f(x) = x - \operatorname{tg} x$, следует, что значение первого ненулевого корня уравнения, заключено между π и $3\pi/2$. Выберем интервал $[3.1; 4.7]$. Ограничим вычислительный процесс достижением интервала, содержащего корень, равного $\varepsilon = 10^{-5}$. Если корень уравнения $x^* \in [a; b]$, то условие прекращения вычислительного процесса

$$|b - a| < \varepsilon.$$

Для проверки точности нахождения корня полезно контролировать невязку δ уравнения $\delta = |x^* - \operatorname{tg} x^*|$, где x^* - вычисленное значение корня уравнения.

Программа решения (Python)

```
# Решить уравнение f(x) = x - tg(x) = 0
import math
import numpy as np
import matplotlib.pyplot as plt
def funct(x):
    y = x - math.tan(x)
    return y
x1 = math.pi
x2 = 3*math.pi/2
eps = 10**(-6)
f1 = funct(x1)
f2 = funct(x2)
#print ("f1",f1,"f2",f2)
if (f1*f2 < 0):
    xs = (x1+x2)/2
    while (abs(x2-x1) > eps):
        if ((funct(x1)*funct(xs)) < 0):
            x2 = xs
        else:
            x1 = xs
    xs = (x1+x2)/2
    print ("x1",x1,"x2",x2, "xs",xs)
```

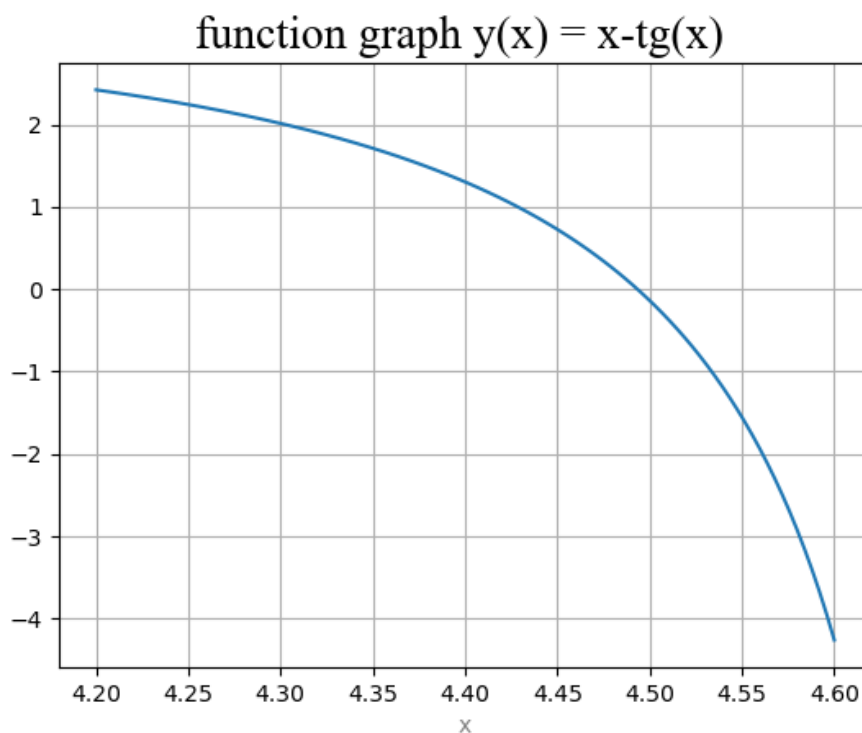
```

print("result",xs, "nev", abs(func(x)))
# Построение графика функции f(x)=x - tg(x)
n = 100
x = np.linspace(4.2, 4.6, n)
y = np.zeros(n)
for i in range(n):
    y[i] = func(x[i])
plt.title('function graph y(x) = x-tg(x) ', fontsize=20, fontname='Times New Roman')
plt.xlabel('x', color='gray')
plt.plot(x,y)
plt.grid(True)
plt.show()
else:
    print("Mistake")

```

Результат работы программы

result 4.493409103336704 nev 7.15906230119856e-06



Для решения воспользуемся GNU Octave.

Построим график функции $y = x - \tan(x)$ на интервале $[\pi; 3 \cdot \pi/2]$. Для этого создадим программу (аналог <http://gnu-octave.narod.ru/olderfiles/1/glava7.pdf>)

Создадим файл f1.m расчета функции $y = x - \tan(x)$

```
function y = f1(x)
```

```
y = x - tan(x);
```

```
endfunction
```

```

Создадим файл graphic.m с программой отображения графика
cla; % Очистка графических объектов
okno1=figure(); % Создание окна с идентификатором okno1
x= pi():0.01:1.5*pi(); % Создание массива переменных x
y=f1(x); % Создание массива переменных y
pol=plot(x,y); % Создание графика
set(pol,'LineWidth',3,'Color','k') % Установка толщины и цвета линии
set(gca,'xlim',[pi,1.5*pi]); % Установка осей
set(gca,'ylim',[-10,10]);
set(gca,'xtick',[3:0.5:1.5*3]); % Вывод сетки с заданным диапазоном
set(gca,'ytick',[-10:1:10]);
grid on; % Установка сетки
xlabel('x');ylabel('y'); % Подписи осей
title('Plot y=x-tg(x)'); % подпись графика

```

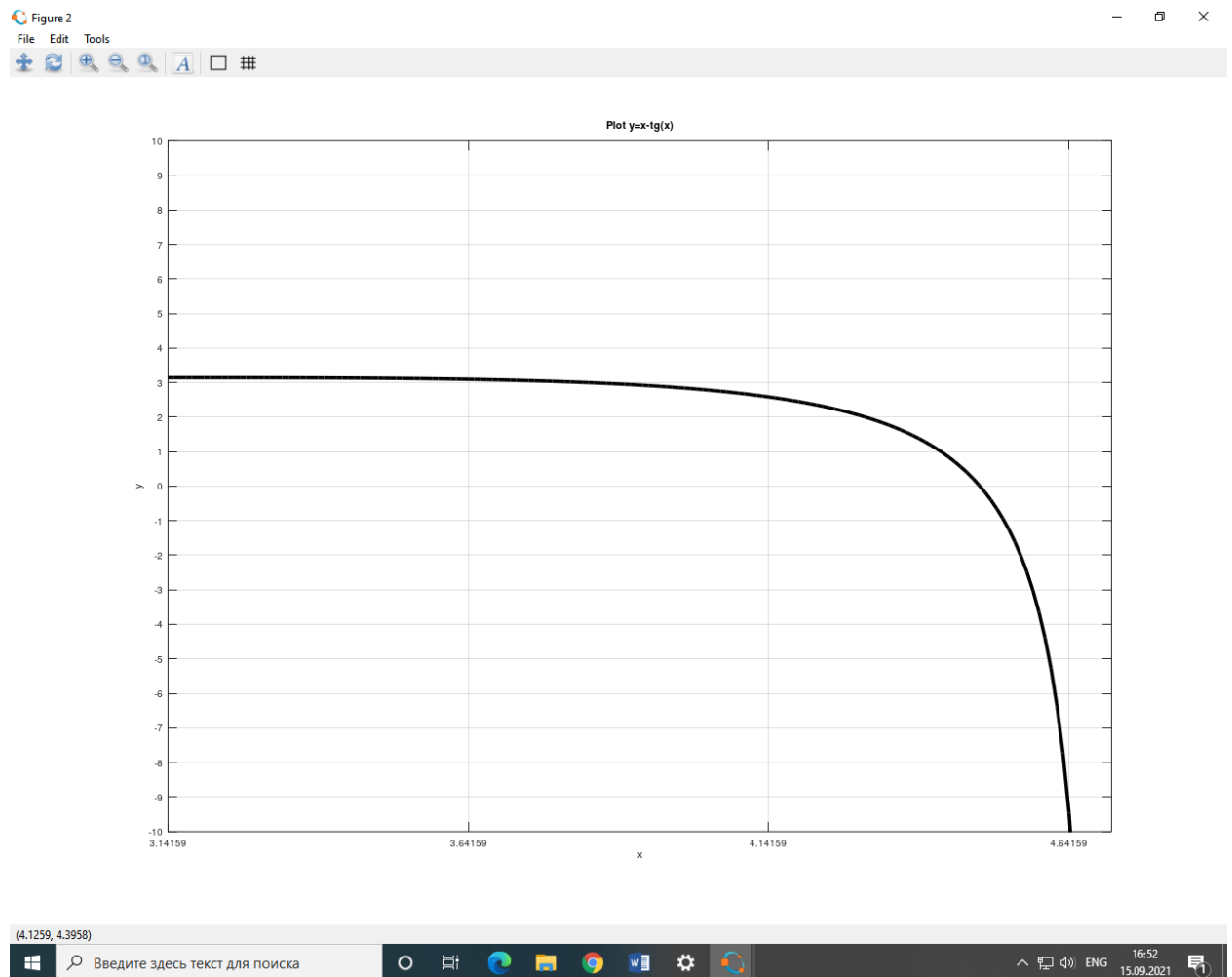


Рис. 1.1. График функции $y = x - \operatorname{tg}(x)$

Программа построит график (рис. 1.1), из которого следует, что приближенное значение корня $x^* \approx 4.5$.

Перейдем в режим команд и выполним поиск корня трансцендентного уравнения, используя функцию `fzero(name,x0)` –

```
>> z=fzero('f1',4.5)
z = 4.4934
```

Вычислим невязку (проверка решения, то есть подставим решение в исходное уравнение)

```
>> nev =f1(z)
nev = 5.3291e-14
```

Таким образом, найден корень уравнения $x^* = z = 4.4934$ с точностью 10^{-14}

```
% Решение уравнения  $x - \operatorname{tg} x = 0$  методом бисекции
a = 3.1; b = 4.7; % начальные значения
eps = 10^(-7);
```

```
if(f1(a)*f1(b)) < 0 % при выполнении - вычисления
while abs (b-a)> eps
    x = (a+b)/2;
    if f1(a)*f1(x) < 0
        b=x;
    else
        a=x;
    endif
endwhile
xx = (a+b)/2;
disp( 'Root '); disp(xx);
nev=f1(xx);
disp( 'Discrepancy '); disp (nev);
else
    disp( 'change interval');
endif
```

Для решения воспользуемся онлайн компилятором (например, https://www.onlinegdb.com/online_c++_compiler).

Листинг программы решения уравнения методом половинного деления, представлен ниже.

```
#include <iostream> // Листинг на C++
#include <math.h>
using namespace std;

double fx(double x)
{
    double ff;
    ff = x-sin(x)/cos(x);
    return(ff);
}

int main()
{
```

```

std::cout << "      Root- Bisection \n";
std::cout << "      The equation x- tg(x)=0 \n";
double a = 3.1, b = 4.7;
double eps = 1.0E-7;
int i, niter = 100;
if (fx(a) * fx(b) > 0)      printf("\t Mistake a= %f  b= %f \n", a, b);
else {
    double xx, dx;
    for (i = 1; i < niter; i++) {
        xx = (a + b) / 2;
        if (fx(a) * fx(xx) < 0) b = xx; else a = xx;
// После отладки программы следующий оператор печати можно будет закомментировать
        printf("\t i=%i  a= %f  b= %f  x = %f \n", i, a, b, xx);
        dx = b - a;
        if (fabs(dx) < eps) {
            double nev = fx(xx);
            printf("\tРешение x = %f \n\tНевязка = %e \n\tЧисло итераций = %i \n", xx, nev, i);
            break;}
        }
    }
    return 0;
}

```

Листинг с ответом –

```

Root- Bisection

      The equation x- tg(x)=0

Решение x = 4.493409
Невязка = 3.030114e-07
Число итераций = 24

```

```

//                      Листинги на Java
package lab2;
public class prim1 {
    public static void main(String[] args) {
        double x, a=3.14,b=3.0/2*3.14, fa, fb, fx, eps=1e-7;
        fa = Math.tan(a)-a;    fb = Math.tan(b)-b;
        if (fa*fb>0)System.out.println("Необходимо изменить интервал, т.к. fa*fb>0 ");
        for(int i=0; i<100; i++){
            x=(a+b)/2;
            if (Math.abs(b-a)<eps) {
                double sigma = Math.tan(x)-x;
                System.out.println(" Решение x="+x+"  Невязка = "+sigma); break; }
            fx = Math.tan(x)-x;
            fa = Math.tan(a)-a;
            if (fx*fa>0)a=x; else b=x;
        }
    }
}

```

```
}  
} }
```

Листинг с результатом решения –

Решение $x=4.493409464657308$ Невязка = $1.3625196260136363E-7$

Листинг программы, с использованием методов созданного класса MetSol -

```
package lab2;  
public class prim1 {  
    public static void main(String[] args) {  
        double a=3.14,b=3.0/2*3.14, fa, fb, fx, eps=1e-7;  
        MetSol md = new MetSol();  
        double xx=md.fx(a,b,eps); // Метод поиска корня уравнения  
        System.out.print ("Корень x="+xx+ " Невязка = "+Math.abs(md.ff(xx)));  
    }  
}  
  
package lab2;  
public class MetSol {  
    MetSol() {};  
    public double fx (double a, double b , double eps) { // Метод половинного деления  
        double fa=ff(a), fb=ff(b);  
        if (fa*fb>0)System.out.println("Необходимо изменить интервал, т.к. fa*fb > 0");  
        double z=0;  
        for (int i=0;i<100;i++){ // Предельное число итераций  
            z=(a+b)/2;  
            if (ff(a) * ff(z)<0) b=z; else a=z;  
            if (Math.abs(b-a)<eps) return z; }  
        return -1000;}  
  
    // Метод определения значения функции  
    public double ff (double x){ return Math.tan(x)-x; }  
}
```

Листинг с результатом решения –

Корень $x=4.493409511446954$ Невязка = $1.0809692714275343E-6$