

Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной
математики

Кафедра вычислительной математики и программирования

Лабораторная работа №2 по курсу «Численные методы»

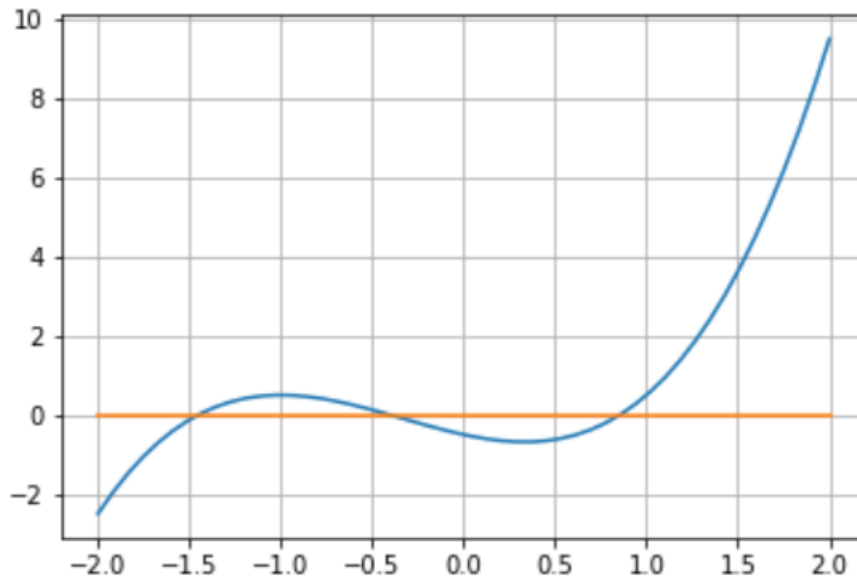
Студент: П. А. Гамов
Преподаватель: Д. Л. Ревизников
Группа: М8О-407Б
Дата:
Оценка:
Подпись:

Москва, 2021

1 Решение нелинейных уравнений

Данная функция.

```
1 def f(x):  
2     return x**3 + x**2 - x - 0.5
```



Понимаем что корень больше нуля в пределах 0.5 и 1.

1 Метод простых итераций

```
1 def solve_simple_iter(f, a, b, err):  
2     step, q, alpha = 0.1, [], diff(f, (a + b) / 2)  
3     fi = lambda x: x - (1/alpha)*f(x)  
4     diffi = lambda x: (x-(1/alpha)*fi(x + err) - (x-(1/alpha)*fi(x))) / err  
5     x = a + err  
6     while x < b:  
7         q.append(abs(diffi(x)))  
8         if fi(x) <= a or fi(x) >= b:  
9             print("fi(x) not in [a,b] : Error")  
10            exit()  
11        elif abs(diffi(x)) > 1:  
12            print("det fi(x) > 1 : Error")  
13            exit()  
14        x += step # check step  
15    q, x, num_of_it = max(q), (a + b) / 2, 0  
16    while True:  
17        x_ = fi(x)  
18        num_of_it += 1
```

```

19 |         if abs(x - x_) * q / (1 - q) < err:
20 |             return x_, f(x_), num_of_it
21 |         x = x_

1 | (0.8546377050834388, 7.357091758031231e-08, 13)

```

2 Метод Ньютона

```

1 | def solve_Newton(f, a, b, err):
2 |     x, num_of_it = (a + b) / 2, 0
3 |     dif = lambda x: (f(x + err) - f(x)) / err
4 |     diff = lambda x: (dif(x + err) - dif(x)) / err
5 |     if f(a) * f(b) >= 0:
6 |         print('f(a)f(b) >= 0 : Error')
7 |         exit()
8 |     if f(x) * diff(x) <= 0:
9 |         print(f(x) * diff(x))
10 |        print("f(x)f'(x) <= 0")
11 |        exit()
12 |
13 |     while True:
14 |         x_ = x - f(x) / dif(x)
15 |         num_of_it += 1
16 |         if abs(x - x_) < err:
17 |             return x_, f(x_), num_of_it
18 |         x = x_

1 | (0.8546376797184615, 3.3306690738754696e-16, 4)

```

2 Решение системы нелинейных уравнений

```

1 | def f1(x,y):
2 |     return x - cos(y) - 1
3 | def f2(x,y):
4 |     return y - log10(x+1) - 1
5 | def fi1(x,y):
6 |     return cos(y) + 1
7 | def fi2(x,y):
8 |     return log10(x+1) + 1

```

Начальная точка: 0.5, 0.5

1 Метод простых итераций

```

1 | def solve_simple_iter(fi1, fi2, x, y, err):
2 |     num_of_it = 0
3 |     while True:
4 |         x_, y_ = fi1(x,y), fi2(x,y)
5 |         num_of_it += 1
6 |         if max(abs(x - x_), abs(y - y_)) < err:
7 |             break
8 |         x, y = x_, y_

1 | x = 1.22211351; y = 1.34674495; f1 = -0.000068; f2 = -0.000021; 13

```

2 Метод Ньютона

```

1 | def solve_Newton(f1, f2, x, y, err):
2 |     num_of_it = 0
3 |     dif1 = lambda f,x,y: (f(x + err,y) - f(x,y)) / err
4 |     dif2 = lambda f,x,y: (f(x,y + err) - f(x,y)) / err
5 |     det = lambda A: A[0][0]*A[1][1] - A[0][1]*A[1][0]
6 |     while True:
7 |         A1 = det([[f1(x,y),dif2(f1,x,y)],
8 |                  [f2(x,y),dif2(f2,x,y)]]))
9 |         A2 = det([[dif1(f1,x,y),f2(x,y)],
10 |                  [dif1(f2,x,y),f2(x,y)]]))
11 |         J = det([[dif1(f1,x,y),dif2(f1,x,y)],
12 |                  [dif1(f2,x,y),dif2(f2,x,y)]]))
13 |         x_, y_ = x - A1 / J, y - A2 / J
14 |         num_of_it += 1
15 |         if max(abs(x - x_), abs(y - y_)) < err:
16 |             break
17 |         x, y = x_, y_

1 | x = 1.22215323; y = 1.34679931; f1 = 0.000025; f2 = 0.000025; 5

```

3 Выводы

В данной лабораторной работе я научился решать системы и нелинейных уравнений итерационными методами.