Московский авиационный институт
(национальный исследовательский университет)

Институт №8 «Компьютерные науки и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

Лабораторные работы по курсу «Численные методы»

Студент: О. В. Гребнева
Преподаватель: Д. Е. Пивоваров
Группа: М8О-303Б-21
Дата:
Оценка:
Подпись:

Москва, 2024

2.1. Нахождение положительных корней нелинейного уравнения методами простой итерации и Ньютона

1 Постановка задачи

Реализовать методы простой итерации и Ньютона решения нелинейных уравнений в виде программ, задавая в качестве входных данных точность вычислений. С использованием разработанного программного обеспечения найти положительный корень нелинейного уравнения (начальное приближение определить графически). Проанализировать зависимость погрешности вычислений от количества итераций.

Вариант: 5

$$cos(x) + 0,25x - 0,5 = 0$$

2 Результаты работы

```
Метод простой итерации:
x = 1.42662
Число итераций: 5
Метод Ньютона:
x = 1.42707
Число итераций: 4
```

3 Исходный код

```
#include <iostream>
#include <vector>
#include <fstream>

using namespace std;

double g(double x) {
    return acos(-0.25*x + 0.5);
}

double f(double x) {
    return cos(x) + 0.25*x - 0.5;
}

double g6(double x) {
    return -sin(x) + 0.25;
}

double h(double x) {
    return -cos(x);
}

pair<double, int> simple_iter(const double q, const double a, const double b, const
    double eps){

    double x, y, r;
    int k;

    x = (a + b)/2;
    r = 2*eps;
    k = 0;

    while(r > eps){
        y = x;
        x = g(y);
        if ((x < a) || (x > b)) break;
        if (x > y)
            r = x - y;
        else
            r = y - x;
        r = r*q/(1 - q);
        k = k + 1;
    }
    return make_pair(x, k);
}

pair<double, int> newton(const double a, const double b, const double eps){
```

```
47      double x, y, r;
48      int k = 0;
49
50      if(f(a)*h(a) > 0)
51          x = a;
52      else
53          x = b;
54
55      r = 2 * eps;
56      while(r > eps){
57          y = x;
58          x = y - f(y)/g6(y);
59          if (x > y)
60              r = x - y;
61          else
62              r = y - x;
63          k = k + 1;
64      }
65      return make_pair(x, k);
66  }
67
68  int main() {
69      double q, a, b, eps, x;
70      int k;
71
72      ifstream fin("input.txt");
73      fin >> q;
74      fin >> a;
75      fin >> b;
76      fin >> eps;
77
78      ofstream fout("answer.txt");
79
80      x = simple_iter(q, a, b, eps).first;
81      k = simple_iter(q, a, b, eps).second;
82
83      fout << "  :" << endl;
84      fout << "x = " << x << endl;
85      fout << " : " << k << endl;
86
87      x = newton(a, b, eps).first;
88      k = newton(a, b, eps).second;
89
90      fout << " :" << endl;
91      fout << "x = " << x << endl;
92      fout << " : " << k << endl;
93
94      return 0;
95  }
```

2.2. Решение СНУ методами простой итерации и Ньютона

1 Постановка задачи

Реализовать методы простой итерации и Ньютона решения систем нелинейных уравнений в виде программного кода, задавая в качестве входных данных точность вычислений. С использованием разработанного программного обеспечения решить систему нелинейных уравнений (при наличии нескольких решений найти то из них, в котором значения неизвестных являются положительными); начальное приближение определить графически. Проанализировать зависимость погрешности вычислений от количества итераций.

Вариант: 5

$$\begin{cases} x_1 - cos(x_2) = 1, \\ x_2 - lg(x_1 + 1) = 2 \end{cases}$$

2 Результаты работы

```
Метод простой итерации:
x1 = 0.443593
x2 = 2.15971
Число итераций: 11
Метод Ньютона:
x1 = 0.444529
x2 = 2.15973
Число итераций: 4
```

3 Исходный код

```
#include <iostream>
#include <vector>
#include <fstream>

using namespace std;

double f(double x1, double x2) {
    return cos(x2) + 1;
}

double g(double x1, double x2) {
    return log10(x1+1) + 2;
}

double f1(double x1, double x2) {
    return x1 - cos(x2) - 1;
}

double g1(double x1, double x2) {
    return x2 - log10(x1+1) - 2;
}

double f1_x1(double x1, double x2) {
    return 1;
}

double g1_x1(double x1, double x2) {
    return -1/(log(10)*(x1+1));
}

double f1_x2(double x1, double x2) {
    return sin(x2);
}

double g1_x2(double x1, double x2) {
    return 1;
}


pair<double, double> simple_iter(const double q, const double a, const double b, const
     double c, const double d, const double eps, int &k){

    double x1, x2, u, v, r, s;

    x1 = (a + b)/2;
    x2 = (c + d)/2;
    r = 2*eps;
```

```
47      k = 0;
48
49      while(r > eps){
50          u = x1;
51          v = x2;
52          x1 = f(u, v);
53          x2 = g(u, v);
54          //if ((x1 < a) || (x1 > b) || (x2 < c) || (x2 > d)) exit(0);
55          if (x1 > u)
56              r = x1 - u;
57          else
58              r = u - x1;
59
60          if (x2 > v)
61              s = x2 - v;
62          else
63              s = v - x2;
64
65          if (s > r)
66              r = s;
67
68          r = r*q/(1 - q);
69          k = k + 1;
70      }
71      return make_pair(x1, x2);
72  }
73
74  pair<double, double> newton(const double a, const double b, const double c, const
        double d, const double eps, int &k){
75      double x1, x2, u, v, r, s;
76
77      x1 = (a + b)/2;
78      x2 = (c + d)/2;
79      r = 2*eps;
80      k = 0;
81
82      while(r > eps){
83          u = x1;
84          v = x2;
85          s =f1_x1(u, v)*g1_x2(u, v) - f1_x2(u, v)*g1_x1(u,v);
86          //if (s == 0) exit(0);
87          x1 = u - (f1(u, v)*g1_x2(u, v) - g1(u, v)*f1_x2(u, v))/s;
88          x2 = v - (f1_x1(u, v)*g1(u, v) - g1_x1(u, v)*f1(u, v))/s;
89          if (x1 > u)
90              r = x1 - u;
91          else
92              r = u - x1;
93
94          if (x2 > v)
```

```
95              s = x2 - v;
96          else
97              s = v - x2;
98
99          if (s > r)
100             r = s;
101
102         k = k + 1;
103     }
104     return make_pair(x1, x2);
105 }
106
107 int main() {
108     double q, a, b, c, d, eps, x1, x2;
109     int k;
110
111     ifstream fin("input.txt");
112     fin >> q;
113     fin >> a;
114     fin >> b;
115     fin >> c;
116     fin >> d;
117     fin >> eps;
118
119     ofstream fout("answer.txt");
120
121     x1 = simple_iter(q, a, b, c, d, eps, k).first;
122     x2 = simple_iter(q, a, b, c, d, eps, k).second;
123
124     fout << "  :" << endl;
125     fout << "x1 = " << x1 << endl;
126     fout << "x2 = " << x2 << endl;
127     fout << " : " << k << endl;
128
129     x1 = newton(a, b, c, d, eps, k).first;
130     x2 = newton(a, b, c, d, eps, k).second;
131
132     fout << " :" << endl;
133     fout << "x1 = " << x1 << endl;
134     fout << "x2 = " << x2 << endl;
135     fout << " : " << k << endl;
136
137     return 0;
138 }
```