МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ

по практической работе №3

по дисциплине «Вычислительная математика»

Тема: Решение нелинейных уравнений и систем

Студент гр. 0303 _____ Болкунов В.О.

Преподаватель _____ Сучков А.И.

Санкт-Петербург

2021

Цель работы.

Сформировать представление о методах решения нелинейных уравнений и систем нелинейных уравнений, выработать умение составлять и применять алгоритмы для решения уравнений и систем уравнений, привить навык использования программных средств для решения нелинейных уравнений и систем нелинейных уравнений.

Основные теоретические положения.

Решение нелинейных уравнений

Численное решение нелинейных (алгебраических или трансцендентных) уравнений вида $f(x)=0$ заключается в нахождении значений x, удовлетворяющих (с заданной точностью) данному уравнению и состоит из следующих основных этапов:

1. Отделение (изоляция, локализация) корней уравнения.
2. Уточнение с помощью некоторого вычислительного алгоритма конкретного выделенного корня с заданной точностью.

Целью первого этапа является нахождение отрезков из области определения функции f(x), внутри которых содержится только один корень решаемого уравнения. Иногда ограничиваются рассмотрением лишь какой-нибудь части области определения, вызывающей по тем или иным соображениям интерес.

Для реализации данного этапа используются графические или аналитические способы.

При аналитическом способе отделения корней полезна следующая теорема.

Теорема. Непрерывная строго монотонная функция f(x) имеет и притом единственный нуль на отрезке [a,b] тогда и только тогда, когда на его концах она принимает значения разных знаков.

Достаточным признаком монотонности функции f(x) на отрезке [a,b] является сохранение знака производной функции.

Графический способ отделения корней целесообразно использовать в том случае, когда имеется возможность построения графика функции y=f(x). Наличие графика исходной функции дает непосредственное представление о количестве и расположении нулей функции, что позволяет определить промежутки, внутри которых содержится только один корень. Если построение графика функции $y=f(x)$ вызывает затруднение, часто оказывается удобным преобразовать уравнение $f(x)=0$ к эквивалентному виду $f1(x)=f2(x)$ и построить графики функций $y=f1(x) u y=f2(x)$. Абсциссы точек пересечения этих графиков будут соответствовать значениям корней решаемого уравнения. Так или иначе, при завершении первого этапа, должны быть определены промежутки, на каждом из которых содержится только один корень уравнения. Для уточнения корня с требуемой точностью применяются различные итерационные методы, заключающиеся в построении числовой последовательности $x_k, k=(1,2,3...)$, сходящейся к искомому корню $\tilde{x}$ уравнения $f(x)=0$.

Решение систем нелинейных уравнений

Систему нелинейных уравнений с n неизвестными можно записать в виде

$$\begin{cases} f_1(x_1,...,x_n)=0 \\ f_2(x_1,...,x_n)=0 \\ f_3(x_1,...,x_n)=0 \end{cases}$$

или, более коротко, в векторной форме $F(X)=0$, где $X=(x_1,x_2,...,xn)^T$ – вектор неизвестных, $F=(f_1,f_2,...,f_n)^T$ вектор-функция.

В редких случаях для решения такой системы удается применить метод последовательного исключения неизвестных и свести решение исходной задачи к решению одного нелинейного уравнения с одним неизвестным. Значения

других неизвестных величин находятся соответствующей подстановкой в конкретные выражения. Однако в подавляющем большинстве случаев для решения систем нелинейных уравнений используются итерационные методы. В дальнейшем предполагается, что ищется изолированное решение нелинейной системы.

Как и в случае одного нелинейного уравнения, локализация решения может осуществляться на основе специфической информации по конкретной решаемой задаче (например, по физическим соображениям), и – с помощью методов математического анализа. При решении системы двух уравнений, достаточно часто удобным является графический способ, когда месторасположение корней определяется как точки пересечения кривых $f1(x_1,x_2)=0$, $f2(x_1,x_2)=0$ на плоскости (x_1,x_2).

Постановка задачи.

Численно решить уравнение и систему уравнений методами Ньютона и простых итераций с заданной точностью ε. Значение ε варьируется от 0.1 до 0.000001.

Выполнение работы.

Уравнение

$$7.7x^2-2.5x-15.1=9.2\sin(2.7x+3.8)$$

График представлен на рис. 1.

По графику отобраны отрезки [-2, -1] и [1.5, 2.0], которые удовлетворяют условию Больцмана-Коши.

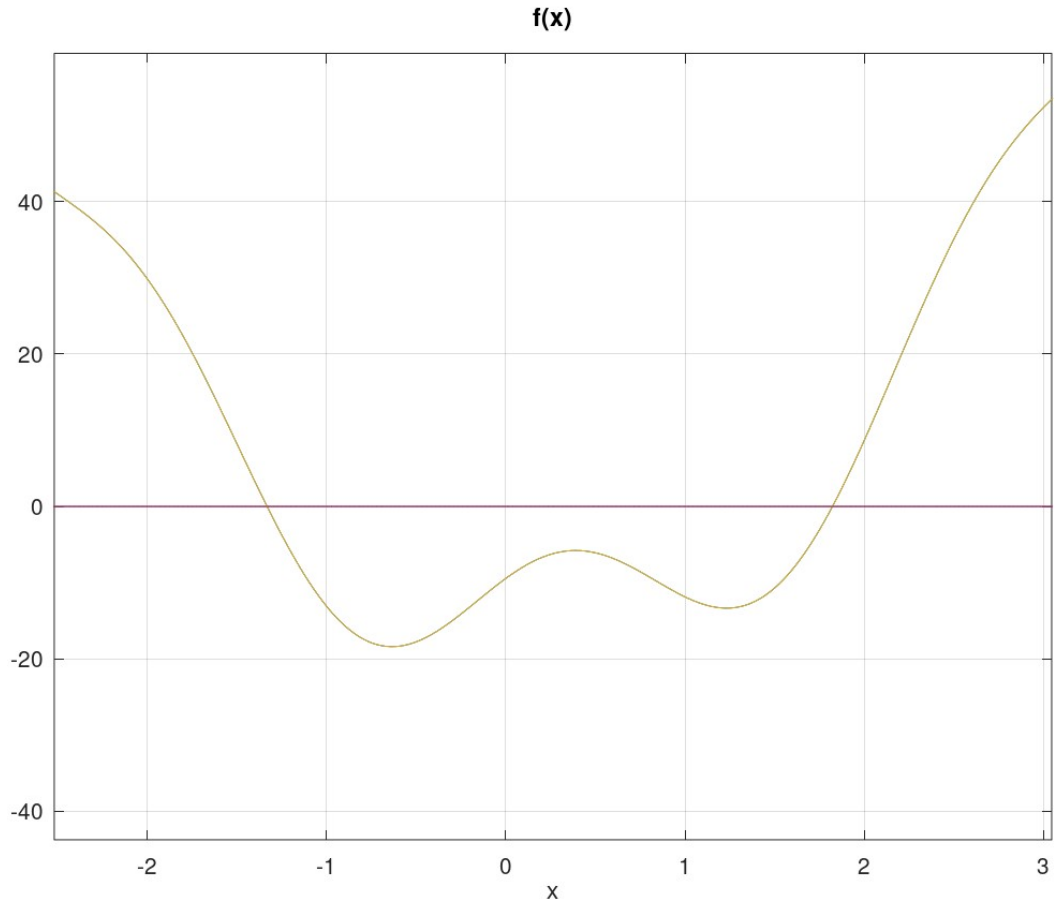

Рисунок 1

Метод Ньютона

Результаты вычислений каждого корня с различной точностью представлены в табл. 1.

Таблица 1

Значение ε	Значение x_1	Значение x_2	Число итераций k
0.1	-1.331920	1.823430	3
0.01	-1.331338	1.823303	4
0.001	-1.331338	1.823303	4
0.0001	-1.331338	1.823303	5
0.00001	-1.331338	1.823303	5
0.000001	-1.331338	1.823303	5

Подробные результаты вычислений наименьшего корня с точность 10^{-6} представлены в табл. 2.

Таблица 2

Значение k	Значение x_k	Значение $f(x_k)$	Значение $f`(x_k)$	Значение $-f(x_k)/`f(x_k)$
0	-2	29.89608	-32.57468	0.9177703
1	-1.082230	-10.45496	-35.03182	-0.2984420
2	-1.380672	2.366298	-48.53765	0.04875180
3	-1.331920	0.02752587	-47.33741	0.0005814824
4	-1.331338	0.000004904368	-47.32053	0.0000001036415
5	-1.331338	1.569855e-13	-47.32053	3.317493e-15

Вычисленные корни совпадают с точками на графике. Рассуждения об обусловленности метода схожи с методом Простых Итераций (см ниже). Скорость сходимости соответствует теоретическим предположениям: $|x_i - \tilde{x}| \le c * |x_{i+1} - \tilde{x}|^2$

Метод простых итераций

$$\varphi(x) = x - \lambda * f(x)$$

Лямбда вычисляется в программе по формуле

$$\lambda = \frac{2}{m+M}, где\ m = \min_{[a,b]} f'(x), и\ M = \max_{[a,b]} f'(x)$$

Результаты вычислений каждого корня с различной точностью представлены в табл. 3.

Таблица 3

Значение ε	Значение x_1	Значение x_2	Число итераций k
0.1	-1.345832	1.829090	2
0.01	-1.328360	1.822002	3
0.001	-1.331219	1.823240	5
0.0001	-1.331362	1.823317	6
0.00001	-1.331339	1.823304	8
0.000001	-1.331338	1.823303	9

Подробные результаты вычислений наименьшего корня с точность 10^{-6} представлены в табл. 4.

Таблица 4

Значение k	Значение x_k	Значение $\varphi(x_k)$
0	-2	-1.241593
1	-1.241593	-1.345822
2	-1.345822	-1.328360
3	-1.328360	-1.331932
4	-1.331932	-1.331219
5	-1.331219	-1.331362
6	-1.331362	-1.331334
7	-1.331334	-1.331339
8	-1.331339	-1.331338
9	-1.331338	-1.331338

Вычисленные корни совпадают с точками на графике.

Обусловленность метода обратно пропорциональна тангенсу угла наклона касательной к точке локализации корня: $\nu_\Delta = \frac{1}{|f'(x)|} = \frac{1}{|1-\varphi'(x)|}$

Скорость сходимости соответствует теоретическим предположениям: $|x_i - \tilde{x}| \le c * |x_{i+1} - \tilde{x}|$

Система уравнений

$$\begin{cases} 9e^{-0.2y} - 4x = 2e^{-0.7x} - 6y + 7 \\ 3x^6 + 4y^6 = 685 - 426x + 704y \end{cases}$$

Кривые описываемые уравнениями представлены на рис. 2.

По графику выбраны следующие начальные точки для поиска корней: (-3.2; 1.0) и (2.4; 1.7)

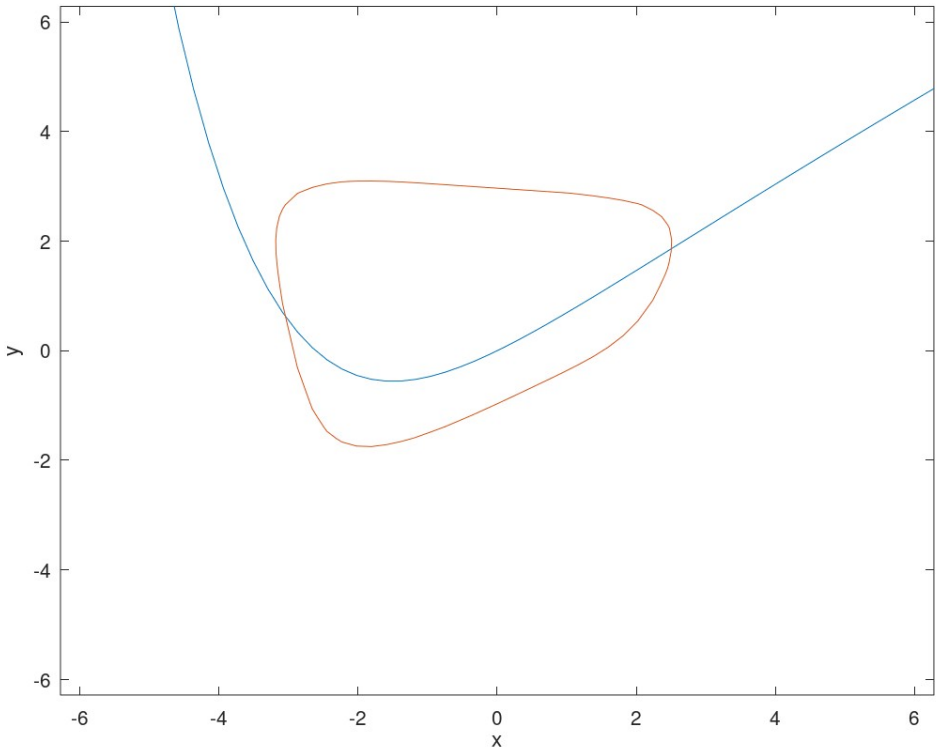

Рисунок 2

Метод Ньютона

Результаты вычислений каждого корня с различной точностью представлены в табл. 5.

Таблица 5

Значение ε	Значение $r_1 = (x_1, y_1)$	Значение $r_2 = (x_2, y_2)$	Число итераций k
0.1	(-3.0544887; 0.6375185)	(2.515406; 1.868667)	2
0.01	(-3.0540373; 0.6370857)	(2.514810; 1.868197)	2
0.001	(-3.0540373; 0.6370857)	(2.514810; 1.868197)	3
0.0001	(-3.0540370; 0.6370855)	(2.514810; 1.868196)	4
0.00001	(-3.0540370; 0.6370855)	(2.514810; 1.868196)	4
0.000001	(-3.0540370; 0.6370855)	(2.514810; 1.868196)	4

Подробные результаты вычислений одного из корней с точность 10^{-6} представлены в табл. 6.

Таблица 6

Значение k	Значение r = (x, y)	Значение $f_1(r_k)$	Значение $f_2(r_k)$	Значение $-J^{-1}(r_k) * F(r_k)$
0	(2.4; 1.7)	-0.3668150	-189.5407	(0.1385; 0.1875)
1	(2.538583; 1.887544)	-0.000000730	1.2595	(-0.0231; -0.0188)
2	(2.515406; 1.868667)	-0.000000730	1.2595	(-0.0005; -0.0004)
3	(2.514810; 1.868197)	-0.000000002	0.0008	(-0.0000003; -0.0000002)
4	(2.514810; 1.868196)	0.000000e+00	3.2423e-10	(-1.535e-13; -1.212e-13)

Метод простых итераций

$$\Phi(X)=X-\Lambda*F(X)$$

Лямбда вычисляется в программе по формуле

$\Lambda=-J^{-1}(X_0)$ где X_0 начальный вектор-приближение.

Результаты вычислений каждого корня с различной точностью представлены в табл. 7.

Таблица 7

Значение ε	Значение $r_1 = (x_1, y_1)$	Значение $r_2 = (x_2, y_2)$	Число итераций k
0.1	-3.0591599 0.6425660	2.505726 1.860926	2/2
0.01	-3.0553534 0.6384584	2.513754 1.867351	3/4
0.001	-3.0541261 0.6371778	2.514685 1.868096	5/6
0.0001	-3.0540602 0.6371095	2.514795 1.868184	6/8
0.00001	-3.0540386 0.6370871	2.514808 1.868195	8/10
0.000001	-3.0540371 0.6370856	2.514810 1.868196	10/13

Подробные результаты вычислений одного из корней с точность 10^{-6} представлены в табл. 8.

Таблица 8

Значение k	Значение r_k	Значение $\Phi(r_k)$
0	(-3.2; 1.0)	(-3.07488; 0.66267)
1	(-3.07488; 0.66267)	(-3.05916; 0.64257)
2	(-3.05916; 0.64257)	(-3.05535; 0.63846)
3	(-3.05535; 0.63846)	(-3.05438; 0.63744)
4	(-3.05438; 0.63744)	(-3.05413; 0.63718)
5	(-3.05413; 0.63718)	(-3.05406; 0.63711)
6	(-3.05406; 0.63711)	(-3.05404; 0.63709)
7	(-3.05404; 0.63709)	(-3.05404; 0.63709)
8	(-3.05404; 0.63709)	(-3.05404; 0.63709)
9	(-3.05404; 0.63709)	(-3.05404; 0.63709)
10	(-3.05404; 0.63709)	(-3.05404; 0.63709)

Выводы.

В ходе работы были исследованы и применены численные методы решения для уравнения и системы уравнений: метод Ньютона и Метод Простых Итераций:

- Построены графики, по которым найдены начальные приближения для решений и отрезки локализации.
- Написаны функции на языке *Octave* для решения данных задач
- С помощью написанных функций и найденных начальных приближений были вычислены корни уравнения/системы с различной точностью для обоих методов.
- Полученные решения, сходимость и обусловленность совпадают с теоретическими ожиданиями.

Приложение. Код программы. Уравнение.

```
function y = f(x)
  y = polyval([7.7, -2.5, -15.1], x) - 9.2 .* sin(2.7 .* x .+ 3.8);
endfunction


function y = dfdx(x)
  y = polyval([2 * 7.7, -2.5], x) - 9.2 * 2.7 .* cos(2.7 .* x .+ 3.8);
endfunction


function [x, X] = newtone(x0, e, f = @f, dfdx = @dfdx)
  x = x0;
  X = [];
  do
    x0 = x;
    x = x - f(x) / dfdx(x);
    X = [X; x];
  until (abs(x - x0) <= e);
endfunction


function x = phi(l, x, f = @f)
  x = x - l .* f(x);
endfunction


function x = dphidx(l, x, dfdx = @dfdx)
  x = 1 .- l .* dfdx(x);
endfunction


function [x, X, q, cnd] = siter(range, x0, e, f = @f, dfdx = @dfdx, phi = @phi, dphidx =
@dphidx)
  r = range(1):e:range(2);
  M = max(dfdx(r));
  m = min(dfdx(r));
  l = 2 / (M + m);
  q = abs((M - m) / (M + m));
  x = x0;
  X = [];
```

```
  do
    x0 = x;
    x = phi(l, x, f);
    X = [X; x];
  until (abs(x - x0) <= ifelse(dphidx(l, x, dfdx) < 0, (1 - q) / q, 1) * e);
  cnd = all((abs(dphidx(l, r)) .- q) <= e);
endfunction
```

Приложение. Код программы. Система Уравнений.

```
function f = f1(x, y)
  f = 9 * exp(-0.2 * y) - 4 * x - 2 * exp(-0.7 * x) + 6 * y - 7;
endfunction


function f = f2(x, y)
  f = 3 * x .^ 6 + 4 * y .^ 6 - 685 + 426 * x - 704 * y;
endfunction


function Y = F(X)
  Y = [
    f1(X(1, :), X(2, :));
    f2(X(1, :), X(2, :))
  ];
endfunction


function J = dFdX(X)
  x = X(1);
  y = X(2);
  J = [
    -4 + 2 * 0.7 * exp(-0.7 * x), -9 * 0.2 * exp(-0.2 * y) + 6;
    3 * 6 * x .^ 5 + 426, 4 * 6 * y .^ 5 - 704
  ];
endfunction
```

```
function [X, r] = newts(X0, e, F = @F, dFdX = @dFdX)
  X = X0;
  r = [];
  do
    X0 = X;
    X = X - inv(dFdX(X)) * F(X);
    r = [r; X.'];
  until (norm(X - X0) <= e);
endfunction

function X = Phi(L, X, F = @F)
  X = X + L * F(X);
endfunction

function [X, r] = siters(X0, e, F = @F, dFdX = @dFdX)
  L = -inv(dFdX(X0));
  X = X0;
  r = [];
  do
    X0 = X;
    X = Phi(L, X, F);
    r = [r; X.'];
  until (norm(X - X0) <= e);
endfunction
```