Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №2
Задание №2 часть II по курсу «Численные методы»
Вариант №12

Студент:	С. М. Власова
Преподаватель:	И. Э. Иванов
Группа:	М8О-306Б
Дата:	
Оценка:	
Подпись:	

Москва, 2020

Задание №2.2

Реализовать метод простой итерации 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: 12

$$\begin{cases} x_1 - \cos(x_2) = 3 \\ x_2 - \sin(x_1) = 3 \end{cases}$$

1 Описание метода решения

Систему нелинейных уравнений с n неизвестными можно записать в виде

$$\begin{cases} f_1(x_1, x_2, \ldots, x_n) = 0 \\ f_2(x_1, x_2, \ldots, x_n) = 0 \\ \quad \ldots\ldots\ldots\ldots\ldots \\ f_n(x_1, x_2, \ldots, x_n) = 0 \end{cases}$$

или, более коротко, в векторной форме

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0},$$

где $\boldsymbol{x}$ — вектор неизвестных величин, $\boldsymbol{f}$ — вектор-функция

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{pmatrix}, \quad \boldsymbol{f} = \begin{pmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ \ldots \\ f_n(\boldsymbol{x}) \end{pmatrix}, \quad \boldsymbol{0} = \begin{pmatrix} 0 \\ 0 \\ \ldots \\ 0 \end{pmatrix}$$

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 $f_1(x_1, x_2) = 0$, $f_2(x_1, x_2) = 0$ на плоскости (x_1, x_2).

Метод простой итерации. При использовании метода простой итерации система уравнений приводится к эквивалентной системе специального вида

$$\begin{cases} x_1 = \phi_1(x_1, x_2, \ldots, x_n) \\ x_2 = \phi_2(x_1, x_2, \ldots, x_n) \\ \quad \ldots\ldots\ldots\ldots \\ x_n = \phi_n(x_1, x_2, \ldots, x_n) \end{cases}$$

или, в векторной форме

$$\boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{x}), \quad \boldsymbol{\phi}(\boldsymbol{x}) = \begin{pmatrix} \phi_1(\boldsymbol{x}) \\ \phi_2(\boldsymbol{x}) \\ \dots \\ \phi_n(\boldsymbol{x}) \end{pmatrix}$$

где функции $\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \dots, \phi_n(\boldsymbol{x})$ — определены и непрерывны в некоторой окрестности искомого изолированного решения $\boldsymbol{x}^{(\star)} = (x_1^{(\star)}, x_2^{(\star)}, \dots, x_n^{(\star)})^T$.

Если выбрано некоторое начальное приближение $\boldsymbol{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})^T$, последующие приближения в методе простой итерации находятся по формулам

$$\begin{cases} x_1^{(k+1)} = \phi_1(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}) \\ x_2^{(k+1)} = \phi_2(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}) \\ \quad \dots\dots\dots\dots \\ x_n^{(k+1)} = \phi_n(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}) \end{cases}$$

$k = 0, 1, 2, \dots$ или в векторной форме

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{\phi}(\boldsymbol{x}^{(k)}), \quad k = 0, 1, 2, \dots$$

Если последовательность векторов $\boldsymbol{x}^{(k)} = (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})^T$ сходится, то она сходится к решению $\boldsymbol{x}^{(\star)} = (x_1^{(\star)}, x_2^{(\star)}, \dots, x_n^{(\star)})^T$.
Достаточное условие сходимости итерационного процесса формулируется следующим образом:

Теорема 2. 1 *Пусть вектор-функция $\boldsymbol{\phi}(\boldsymbol{x})$ непрерывна вместе со своей производной*

$$\boldsymbol{\phi}'(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial\phi_1(\boldsymbol{x})}{\partial x_1} & \frac{\partial\phi_1(\boldsymbol{x})}{\partial x_2} & \dots & \frac{\partial\phi_1(\boldsymbol{x})}{\partial x_n} \\ \frac{\partial\phi_2(\boldsymbol{x})}{\partial x_1} & \frac{\partial\phi_2(\boldsymbol{x})}{\partial x_2} & \dots & \frac{\partial\phi_2(\boldsymbol{x})}{\partial x_n} \\ & \dots\dots\dots & & \\ \frac{\partial\phi_n(\boldsymbol{x})}{\partial x_1} & \frac{\partial\phi_n(\boldsymbol{x})}{\partial x_2} & \dots & \frac{\partial\phi_n(\boldsymbol{x})}{\partial x_n} \end{pmatrix}$$

в ограниченной выпуклой замкнутой области G и

$$\max_{\boldsymbol{x}\in G} ||\boldsymbol{\phi}'(\boldsymbol{x})|| \le q < 1,$$

где q — постоянная. Если $\boldsymbol{x}^{(0)} \in G$ и все последовательные приближения

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{\phi}(\boldsymbol{x}^{(k)}), \quad k = 0, 1, 2, \dots$$

также содержатся в G, *то процесс итерации сходится к единственному решению уравнения*

$$\boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{x})$$

в области G *и справедливы оценки погрешности* $(\forall k \in N)$:

$$||\boldsymbol{x}^{(\star)} - \boldsymbol{x}^{(k+1)}|| \leq \frac{q}{1-q} \cdot ||\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}||$$

$$||\boldsymbol{x}^{(\star)} - \boldsymbol{x}^{(k+1)}|| \leq \frac{q^{k+1}}{1-q} \cdot ||\boldsymbol{x}^{(1)} - \boldsymbol{x}^{(0)}||$$

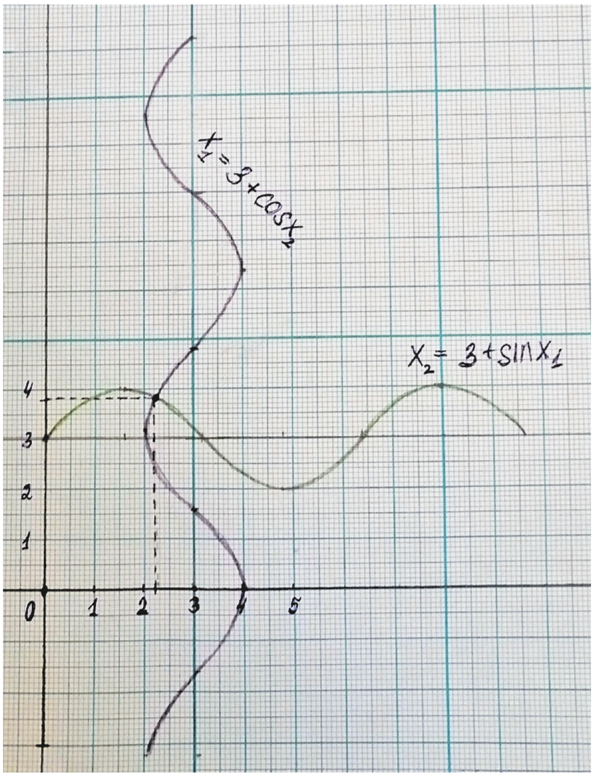
2 Моя задача

Методом простой итерации найти положительное решение системы нелинейных уравнений

$$\begin{cases} x_1 - \cos(x_2) = 3 \\ x_2 - \sin(x_1) = 3 \end{cases}$$

с заданной точностью.

Решение:
Этап I. Локализация: для выбора начального приближения применим графический способ. Построив на плоскости (x_1, x_2) в интересующей нас области кривые $f_1(x_1, x_2) = 0$ и $f_2(x_1, x_2) = 0$, определяем, что положительное решение системы находится в квадрате $2 < x_1 < 3$ и $3 < x_2 < 4$.



В качестве начального приближения возьмем середины выбранных интервалов.

$$x_1^{(0)} = \frac{(2+3)}{2} = 2.5, \quad x_2^{(0)} = \frac{(3+4)}{2} = 3.5$$

Этап II. Уточнение: преобразуем исходную систему уравнений к виду

$$\begin{cases} x_1 = \cos(x_2) + 3 \equiv \phi_1(x_1, x_2) \\ x_2 = \sin(x_1) + 3 \equiv \phi_2(x_1, x_2) \end{cases}$$

Проверим выполнение условия $\max_{\boldsymbol{x} \in G} ||\boldsymbol{\phi'(x)}|| \le q < 1$, в области $G : |x_1 - 2.5| \le 0.5$, $|x_2 - 3.5| \le 0.5$. Для этого найдем

$$\max_{x \in G} ||\boldsymbol{\phi}'(\boldsymbol{x})|| = \max_{x \in G}\{\max_{(i)} \sum_{j=1}^{n} |\frac{\partial \phi_i(x_1, x_2)}{\partial x_j}|$$

Так как

$$\frac{\partial \phi_1(x_1, x_2)}{\partial x_1} = 0, \qquad \frac{\partial \phi_1(x_1, x_2)}{\partial x_2} = -\sin(x_2),$$

$$\frac{\partial \phi_2(x_1, x_2)}{\partial x_1} = \cos(x_1), \qquad \frac{\partial \phi_2(x_1, x_2)}{\partial x_2} = 0,$$

то в области G имеем

$$|\frac{\partial \phi_1(x_1, x_2)}{\partial x_1}| + |\frac{\partial \phi_1(x_1, x_2)}{\partial x_2}| = | - \sin(x_2)| = \max_{x_2 \in [3,4]} |\sin(x_2)| \le 1.$$

$$|\frac{\partial \phi_2(x_1, x_2)}{\partial x_1}| + |\frac{\partial \phi_2(x_1, x_2)}{\partial x_2}| = | \cos(x_1)| = \max_{x_1 \in [2,3]} |\cos(x_1)| \le 1.$$

Так, если последовательные приближения $(x_1^{(k)}, x_2^{(k)})$ не покинут область G, итерационный процесс будет сходящимся.
В качестве начального приближения примем $x_1^{(0)} = 2.5, \quad x_2^{(0)} = 3.5$.

Вычисления завершаются при выполнении условия

$$\frac{q}{1-q} \cdot ||\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}||,$$

$$||\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}|| = \max_{i} |x_i^{(k+1)} - x_i^{(k)}|.$$

3 Протокол

Входные данные через командную строку я задаю значение точности ε.
Выходные данные я вывожу на экран.
Скриншот консоли:
$\varepsilon = 0.1$.

```
(base) vlasochka@vlasochka-VPCSB11FX:~/Документы/ЧМ/Lab2$ g++ Iter2.2.cpp -o iter2.2
(base) vlasochka@vlasochka-VPCSB11FX:~/Документы/ЧМ/Lab2$ ./iter2.2
Введите точность вычислений: 0.1
Метод итераций:
q = 0.989992
Положительный корень системы нелинейных уравнений:(2.21085, 3.80257)
Число итераций: 15
```

$\varepsilon = 0.01$.

```
(base) vlasochka@vlasochka-VPCSB11FX:~/Документы/ЧМ/Lab2$ ./iter2.2
Введите точность вычислений: 0.01
Метод итераций:
q = 0.989992
Положительный корень системы нелинейных уравнений:(2.21047, 3.80227)
Число итераций: 20
```

$\varepsilon = 0.0001$.

```
(base) vlasochka@vlasochka-VPCSB11FX:~/Документы/ЧМ/Lab2$ ./iter2.2
Введите точность вычислений: 0.0001
Метод итераций:
q = 0.989992
Положительный корень системы нелинейных уравнений:(2.21044, 3.80231)
Число итераций: 29
```

$\varepsilon = 0.00001$.

```
(base) vlasochka@vlasochka-VPCSB11FX:~/Документы/ЧМ/Lab2$ ./iter2.2
Введите точность вычислений: 0.00001
Метод итераций:
q = 0.989992
Положительный корень системы нелинейных уравнений:(2.21045, 3.80231)
Число итераций: 34
(base) vlasochka@vlasochka-VPCSB11FX:~/Документы/ЧМ/Lab2$
```

Можно заметить, что сходимость метода простой итерации достаточно медленная. Это можно объяснить тем, что коэффициент q чуть меньше 1 — условие сходимости метода выполняется для всех $\boldsymbol{\phi}_i(\boldsymbol{x})$, но сходимость итерационного процесса в выбранной области плохая.

$$x^{(\star)} \approx \begin{pmatrix} 2.21045 \\ 3.80231 \end{pmatrix}$$

Алгоритм находит точное решение за 34 итерации с точностью $\varepsilon = 0.00001$.

4 Исходный код

Листинг 1: Метод простой итерации

```cpp
#include<iostream>
#include<map>
#include<vector>
#include<math.h>

double Phi1(double x1, double x2)
{
    double val = cos(x2) + 3;
    return val;
}

double Phi2(double x1, double x2)
{
    double val = sin(x1) + 3;
    return val;
}


bool CheckPhi(std::vector<std::pair<double, double>>& p, double eps)
{
    double q;
    while(Phi1(p[0].first, p[0].second) < p[0].first || Phi1(p[0].first, p[0].second) > p[0].second ||
        Phi2(p[1].first, p[1].second) < p[1].first || Phi2(p[1].first, p[1].second) > p[1].second)
    {
        p[0].first += eps;
        p[0].second -= eps;
        p[0].first += eps;
        p[0].second -= eps;
    }
    return true;
}

int main()
{
    double eps, q, kaf;
    int it_count = 0;
    std::vector<std::vector<double>> X(2, std::vector<double> (2, 0));
    std::vector<std::pair<double, double>> ab(2);

```

```
    ab[0] = std::make_pair(2, 3);
    ab[1] = std::make_pair(3, 4);
    std::cin >> eps;

    q = std::max(fabs(-sin(ab[1].second)), fabs(-sin(ab[1].first)));
    q = std::max(q, fabs(cos(ab[0].second)));
    q = std::max(q, fabs(cos(ab[0].first)));

    if(q >= 1)
        std::cerr << "  Phi";
    else
        kaf = q/(1-q);
    do
    {
        it_count++;
        X[0][0] = X[0][1];
        X[1][0] = X[1][1];
        X[0][1] = Phi1(X[0][0], X[1][0]);
        X[1][1] = Phi2(X[0][0], X[1][0]);

    }
    while(kaf*std::max(fabs(X[0][1] - X[0][0]), fabs(X[1][1] - X[1][0])) >= eps);

    std::cout << "Method of simple iterations:" << std::endl;
    std::cout << "q = " << q << std::endl;
    std::cout << "The positive solution of a system of nonlineral equations:" << "(" << X[0][1] << ", " << X[1][1] << ")" << std::endl;
    std::cout << "The number of iterations: " << it_count << std::endl;

    return 0;
}
```

5 Выводы

Выполнив вторую часть второго задания второй лабораторной работы, я узнала, что метод простых итераций применим и для решения систем нелинейных уравнений. Этот метод является итерационным и базируется на последовательном приближении, согласно итерационной формуле. Чтобы метод простых итераций сходился, необходимо, чтобы выполнялся ряд условий. Одно из этих условий — выбранная эквивалентная вектор-функция должна быть непрерывна вместе со своей производной и ограничена в замкнутой области-окрестности изолированного решения системы. В случае, если начальное приближение лежит в этой окрестности и все последующие приближения не выходят за ее пределы — процесс итерации сходится к единственному решению в этой области.