РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ
(лабораторные работы - 6 часов)

Целью выполнения лабораторных работ является исследование существующих и разработка новых алгоритмов решения нелинейных уравнений и систем нелинейных уравнений.

Теоретические требования. Необходимо подготовить конспект, содержащий описание следующих численных методов:

1) метод бисекции (метод деления пополам);
2) метод ложного положения (метод ложной позиции или метод регул falsi);
3) метод Ньютона;
4) метод хорд;
5) метод фиксированной точки;
6) методы решения уравнений с множественными корнями;
7) методы решения системы уравнений.

Практические требования. Решить задания и составить отчет.

Задание I. Задано уравнение

$$f(x) = x - 2e^{-x}, \ x \in \mathbb{R}. \qquad (1)$$

A. Определите корни уравнения (1):

1) используя метод бисекции, $a = 0$ и $b = 1$, выполните три итерации;
2) используя метод хорд, начальные условия $x_1 = 0$, $x_2 = 1$, выполните три итерации;
3) используя метод Ньютона, начальное условие $x_1 = 1$, выполните три итерации.

B. Метод Стеффенсена (самостоятельное изучение теории) можно использовать для решения уравнений вида $f(x) = 0$, но в отличии от метода Ньютона не используются производные функции. Процесс решения начинается выбором начального приближения x_i, которое принимается как первое решение. Следующее приближение x_{i+1} расчитывается следующим образом:

$$x_{i+1} = x_i - \frac{f^2(x_i)}{f(x_i + f(x_i)) - f(x_i)}.$$

Напишите в MATLAB функцию для решения нелинейного уравнения методом Стеффенсена. Назвовите функцию как **SteffensenRoot**, определяя ее как **Xs=SteffensenRoot(Fun, Xest)**, где **Xs** выходной аргумент функции - численное решение. Входным аргументом является анонимная функция – имя функции Fun, функция служит для расчета $f(x)$ для x, **Xest** – начальное приближение, используемое при численном решения. Итерации прекращаются, если относительная ошибка меньше 10^{-6}. Число итерации должно быть ограниченно так, чтобы избежать зацикливания ($N = 100$). Если желаемая точность не достигнута за 100 итераций, вычисления должны быть остановлены и выдан коммуникат об ошибке.

C. Используйте функцию **SteffensenRoot** для решения уравнения (1).

D. Сравните по точности полученное решение с решением, полученным при использовании языка программирования **Python**.

Задание II. Поверхность панели вулканизируется излучаемой нагревателем энергией. Температура вулканизации определяется процессами излучения и конвективным теплообменом. Если излучение рассматривается как рассеянное, следующая система нелинейных одновременных уравнений позволит определить J_h, T_h, J_c, T_c:

$$\begin{cases} 5.67\times10^{-8}T_c^4+17.41T_c-J_c=5188.18, \\ J_c-0.71J_h+7.46T_c=2352.71, \\ 5.67\times10^{-8}T_h^4+1.865T_h-J_h=2250, \\ J_h-0.71J_c+7.46T_h=11093, \end{cases} \tag{2}$$

где J_h и J_c – излучение тепла нагревателем и вулканизируемой поверхностью; T_h и T_c – соответсвующие излучению температуры.

A. Покажите, что следующие итерационные функции могут быть использованы для решения системы нелинейных уравнений итерационным методом фиксированной точки:

$$T_c=\left[\frac{J_c-17.41T_c+5188.18}{5.67\times10^{-8}}\right]^{1/4}, \tag{3}$$

$$T_h=\left[\frac{2250+J_h-1.865T_h}{5.67\times10^{-8}}\right]^{1/4}, \tag{4}$$

$$J_c=2352.71+0.71J_h-7.46T_c, \tag{5}$$

$$J_h=11093+0.71J_c-7.46T_h. \tag{6}$$

B. Решите систему нелинейных уравнений итерационным методом фиксированной точки, используя итерационные функции (3) – (6). Используйте следующие начальные условия: $T_h=T_c=298K$, $J_c=3000\frac{W}{m^2}$, $J_h=5000\frac{W}{m^2}$. Выполнить 100 итерации. Постройте графики сходимости численного решения для каждой переменной, принимая $\tilde{T}_c=481K$, $\tilde{J}_c=6222\frac{W}{m^2}$, $\tilde{T}_h=671K$, $\tilde{J}_h=10504\frac{W}{m^2}$.

Задание III.

A. Создайте и исследуйте собственный метод решения нелинейного уравнения $f(x)=0$, $x\in\mathbb{R}$. Этот метод должен быть основан на идеях метода бисекции и метода регул фальси (falsi). При разработке примите следующее: решение находится в интервале $[a,b]$, первая итерация выполняется методом бисекции, дальнейшие итерации выполняются двумя методами и выбирается «лучшее» приближение. Выполните кодирование в MatLab:

Xs=**BiRegRoot**(Fun, a, b, ErrMax),

где переменная Xs содержит решение, входной аргумент Fun – имя функции, при помощи которой расчитывается $f(x)$ для x (это фиктивное название для функции, импортируемой в функцию **BiRegRoot**), a и b – границы интервала, ErrMax – максимальная расчетная относительная ошибка:

$$\mathrm{ErrMax} = \left| \frac{x_{NS}^{(n)} - x_{NS}^{(n-1)}}{x_{NS}^{(n-1)}} \right|, \quad x_{NS} = \frac{a+b}{2}.$$

При кодировании алгоритма необходимо учесть:

- проверка интервала, в котором может находиться решение;
- ограничение количества итераций – для избежания зацикливания;
- оценка достижения точности;
- графическое отображение результатов; коммуникаты об ошибках.

B. Выберите самостоятельно задачу (физика, химия). Сформулируйте математическую модель, сводя ее к нелинейному уравнению, определите параметры и переменные. Используйте разработанную функцию **BiRegRoot** для решения задачи. Сравните полученное решение с результатами, полученным при использовании встроенной функции **fzeros**.

Задание IV. Решите численным методом уравнение

$$2x^3 - 4x^2 - 4x - 20 = 0, \quad x \in [3,4].$$

Предложите алгоритм и обоснуйте его выбор. Докажите, что полученное численное решение совпадает с решением, получаемым при использовании функции MatLab – **roots**.

Задание V. Напишите программу в **Python** для решенения нелинейного уравнения $f(x) = 0$, $x \in \mathbb{R}$ (выбрать уравнение согласно варианту, см. Таб. 1). Для решения используйте метод простых итераций (**MyIteration**) и метод Ньютона (**MyNewton**). Функции **MyIteration** и **MyNewton** должны реализовывать алгоритмы численного приближения решения выбранного уравнения. Для расчета значений функции $f(x)$ и ее производной используйте анонимные функции **MyFunction** и **MyDerivative**. Выполните вычисления с двойной точностью. Результаты вычислений представьте в виде таблице, записаной в текстовом файле.

Задание VI. Напишите программу в **Python** для решенения системы нелинейных уравнений $f(\mathbf{x}) = 0$, $\mathbf{x} \in \mathbb{R}^n$ методом Ньютона (выбрать систему уравнений согласно варианту, см. Таб. 2). Выполните вычисления с двойной точностью. Результаты вычислений представьте в виде таблице, записаной в текстовом файле.

Таблица 1. Нелинейные уравнения к заданию V.

Вариант	Уравнение	Вариант	Уравнение
1	$x - \cos\left(\frac{x}{3}\right) = 0$	16	$\sin(1 - 0.2x^2) - x = 0$
2	$x + \ln(4x) - 1 = 0$	17	$e^x - e^{-x} - 2 = 0$
3	$e^x - 4e^{-x} - 1 = 0$	18	$x - \sin\left(\frac{1}{x}\right) = 0$
4	$xe^x - 2 = 0$	19	$e^x + \ln(x) - x = 0$
5	$4(x^2 + 1)\ln(x) - 1 = 0$	20	$1 - x + \sin(x) - \ln(1 + x) = 0$
6	$2 - x - \sin\left(\frac{x}{4}\right) = 0$	21	$(1 - x)^{1/2} - \cos(1 - x) = 0$
7	$x^2 + \ln(x) - 2 = 0$	22	$\sin(x^2) + \cos(x^2) - 10x = 0$
8	$\cos(x) - (x + 2)^{\frac{1}{2}} + 1 =$	23	$x^2 - \ln(1 + x) - 3 = 0$
9	$4\left(1 + x^{\frac{1}{2}}\right)\ln(x) - 1 = 0$	24	$\cos\left(\frac{x}{2}\right)\ln(x - 1) = 0$
10	$5\ln(x) - x^{\frac{1}{2}} = 0$	25	$\cos\left(\frac{x}{5}\right)(1 + x)^{1/2} - x = 0$
11	$e^x + x^3 - 2 = 0$	26	$3x - e^{-x} = 0$
12	$3\sin\left(x^{\frac{1}{2}}\right) + x - 3 = 0$	27	$4\left(1 + x^{\frac{1}{2}}\right)\ln(x) - 10 = 0$
13	$0.1x^2 - xln(x) = 0$	28	$x - \frac{1}{3 + \sin(3.6x)} = 0$
14	$\cos(1 + 0.2x^2) - x = 0$	29	$0.25x^3 + \cos(\frac{x}{4}) = 0$
15	$3x - 4\ln(x) - 5 = 0$	30	$2 - x = \ln(x)$

Таблица 2. Системы нелинейных уравнений к заданию VI.

Вариант	Система уравнений	Вариант	Система уравнений
1	$\begin{cases} tg(xy+0.4)=x^2 \\ 0.6x^2+2y^2=1, x,y>0 \end{cases}$	16	$\begin{cases} \sin(x+y)-x=1 \\ x^2+y^2=1 \end{cases}$
2	$\begin{cases} \sin(x+y)-1.6x=0 \\ x^2+y^2=1, x,y>0 \end{cases}$	17	$\begin{cases} tg(xy+0.2)=x^2 \\ 0.9x^2+2y^2=1 \end{cases}$
3	$\begin{cases} tg(xy+0.1)=x^2 \\ x^2+2y^2=1 \end{cases}$	18	$\begin{cases} \sin(x+y)-1.1x=0 \\ 0.8x^2+2y^2=1 \end{cases}$
4	$\begin{cases} \sin(x+y)-1.2x=0.2 \\ x^2+y^2=1 \end{cases}$	19	$\begin{cases} \sin(x+y)-1.2x=0.1 \\ x^2+2y^2=1 \end{cases}$
5	$\begin{cases} tg(xy+0.3)=x^2 \\ 0.9x^2+2y^2=1 \end{cases}$	20	$\begin{cases} tg(xy)=x^2 \\ 0.5x^2+3y^2=1 \end{cases}$
6	$\begin{cases} \sin(x+y)-1.3x=0 \\ 0.8x^2+2y^2=1 \end{cases}$	21	$\begin{cases} \sin(x+y)+2x=1 \\ 2x^2+y^2=1 \end{cases}$
7	$\begin{cases} \sin(x+y)-1.5x=0.1 \\ x^2+y^2=1 \end{cases}$	22	$\begin{cases} tg(xy+0.4)=3x^2 \\ 6x^2+0.2y^2=1, x,y>0 \end{cases}$
8	$\begin{cases} tg(xy)=x^2 \\ 0.7x^2+2y^2=1 \end{cases}$	23	$\begin{cases} \sin(2x+y)-1.6x=0 \\ x^2+2y^2=1, x,y>0 \end{cases}$
9	$\begin{cases} \sin(x+y)+1.2x=1 \\ x^2+y^2=1 \end{cases}$	24	$\begin{cases} tg(xy+1)=x^2 \\ 3x^2+2y^2=1 \end{cases}$
10	$\begin{cases} tg(xy+0.2)=x^2 \\ 0.6x^2+2y^2=1 \end{cases}$	25	$\begin{cases} tg(xy+0.1)=2x^2 \\ 0.6x^2+2y^2=1, x,y>0 \end{cases}$
11	$\begin{cases} \sin(x+y)=1.5x-1 \\ x^2+y^2=1 \end{cases}$	26	$\begin{cases} \sin(x+y)-1.5x=0.1 \\ x^2+y^2=1 \end{cases}$
12	$\begin{cases} tg(xy+0.3)=x^2 \\ 0.9x^2+2y^2=2 \end{cases}$	27	$\begin{cases} tg(xy+0.1)=x^2 \\ x^2+2y^2=1 \end{cases}$
13	$\begin{cases} tg(xy+0.1)=2x^2 \\ 0.6x^2+2y^2=1, x,y>0 \end{cases}$	28	$\begin{cases} tg(xy+0.5)=x^2 \\ x^2+y^2=1 \end{cases}$
14	$\begin{cases} \sin(x+2y)-1.2x=0 \\ x^2+y^2=1, x,y>0 \end{cases}$	29	$\begin{cases} \sin(x+y)+1.5x=1 \\ x^2+y^2=1 \end{cases}$
15	$\begin{cases} tg(xy+0.5)=x^2 \\ x^2+y^2=1 \end{cases}$	30	$\begin{cases} \sin(x+y)-1.1x=0 \\ 0.8x^2+2y^2=1 \end{cases}$