Практическое занятие №8

Закрепление навыков работы в SciLab

Варианты заданий:

1	Абдуллина Динара Рамиловна, Максиева Ляйсан Мансуровна
2	Гайфуллин Руслан Юнусович, Зайнуллин Динар Рафисович
3	Иванов Егор Михайлович, Ягафарова Гульназ Ильдаровна
4	Клявлина Алсу Иргатовна, Сафина Лилия Ришатовна
5	Юсупова Нелли Рафисовна, Субхангулова Алия Маратовна
6	Тимиряев Расим Радикович, Абдуллина Дина Ураловна
7	Лутфуллин Габдулла Шагитович, Мавзютов Ильсур Ирекович
8	Гаязов Алмаз Линарович, Вагапов Шамиль Рамилевич, Шакиров Артур Маратович

Сценарии.
Файл сценария – это список команд, сохраненный на диске. Для удобства работы со сценариями в комплекте SciLab имеется текстовый редактор (пункт меню Editor). Сценарии сохраняются в виде файлов с расширением sce. Загрузить сценарий на исполнение можно двумя способами:
- с помощью пункта меню Execute-Load into SciLab встроенного редактора
- с помощью команды exec(‘Путь к файлу сценария’)

При выполнении данной лабораторной работы вы можете использовать файлы-сценарии или выполнять все традиционным способом.

Практическое задание.

В вашем варианте дана функция $f(t)$ и точка $x0$.

1. Разложите заданную функцию $f(t)$ в ряд Тейлора в окрестности точки $x0$. Найдите 10 первых коэффициентов ряда.
2. Напишите функцию, реализующую вычисление приближенного значения заданной вам функции суммированием первых n членов ряда Тейлора.
3. Постройте на одном графике заданную функцию вместе с приближенной.
4. Вычислите ошибку представления заданной функции своими приближениями. Постройте на отдельном графике ошибку в зависимости от x.

Варианты

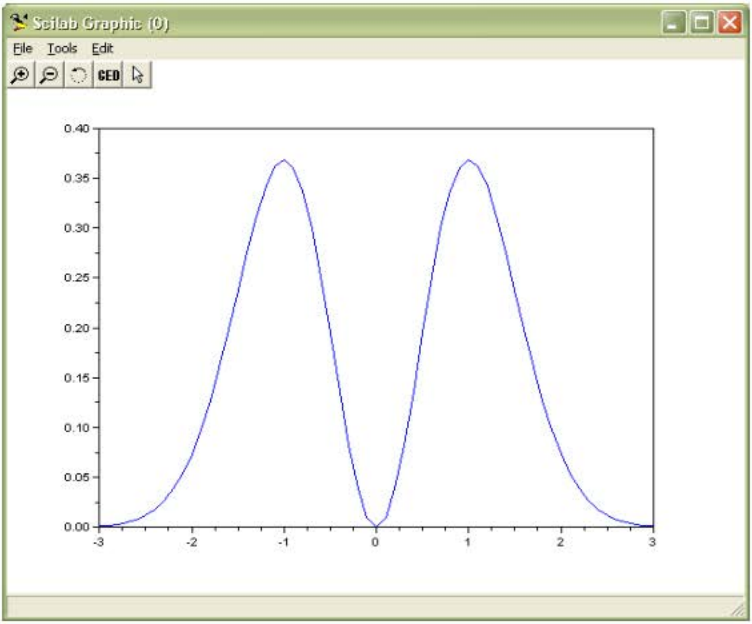
№	Функция f(t)	Точка x0
1	$f(x)=\exp(-x^2)$	0
2	$f(x)=\cos(\lvert x\rvert)$	1
3	$f(x)=x^8+3x^2-2$	2
4	$f(x)=\exp(-x)+\cos(5x)$	3
5	$f(x)=\sin(x^2)$	π
6	$f(x)=tg(x)$	0
7	$f(x)=ctg(x)$	$\pi/2$
8	$f(x)=1-2\cos(x)$	0
9	$f(x)=8x\cdot tg(x)$	0
10	$f(x)=x\exp(-x^2)$	0

11	$f(x)=\frac{\sin(x)}{x}$	0
12	$f(x)=\sin^2(x)+\cos^2(2x)$	1
13	$f(x)=\exp(-x^2+x)$	
14	$f(x)=\sin^2(x^2)$	2
15	$f(x)=\cos(x)\exp(-x^2)$	π

Пример выполнения работы.

Пусть дана функция $f(x)=x^2\exp(-x^2)$ и точка x0=1.

1. Создадим файл сценария test.sce. Все описываемые далее команды будем писать в файле сценария и загружать его на выполнение.
2. Определим функцию
   function y=f(x), y=(x^2)*exp(-x^2), endfunction;
   и построим ее график
   t=-5:0.1:5;
   ff=f(t);
   plot(t,ff)



3. Для нахождения производных введем дискретизацию некоторой окрестности исходной точки x0. При этом точка x0 должна располагаться не посередине выбранной окрестности, а ближе к ее левому краю.
   ```
   x=0:0.1:5
   ```
   Заметим, что при таком выборе дискретизации исходная точка x0 имеет номер 11, то есть x(11)=x0.

4. Вычисляем значения функции в узлах дискретизации
   ```
   y=f(x);
   ```

5. Находим значения первой производной в узлах дискретизации
   ```
   dy1=diff(y,1);
   dy2=diff(y,2);
   dy3=diff(y,3);
   for k=1:28
   z(k)=(dy1(k)-dy2(k)/2+dy3(k)/3)/0.1;
   end
   disp(z(11))
   ```

6. Вычисляем значения второй производной
   ```
   dz1=diff(z,1);
   dz2=diff(z,2);
   dz3=diff(z,3);
   for k=1:25
   w(k)=(dz1(k)-dz2(k)/2+dz3(k)/3)/0.1;
   end
   disp(w(11))
   ```

7. Вычисляем значения третьей производной

```
dw1=diff(w,1);
dw2=diff(w,2);
dw3=diff(w,3);
for k=1:22
v(k)=(dw1(k)-dw2(k)/2+dw3(k)/3)/0.1;
end
disp(v(11))
```

8. Вычисляем значения четвертой производной

```
dv1=diff(v,1);
dv2=diff(v,2);
dv3=diff(v,3);
for k=1:19
q(k)=(dv1(k)-dv2(k)/2+dv3(k)/3)/0.1;
end
disp(q(11))
```

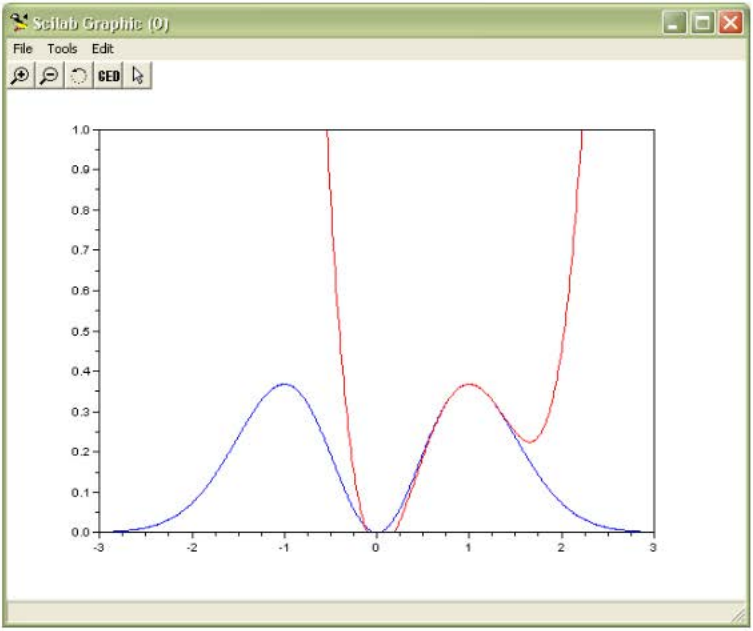
9. Формируем ряд Тейлора

```
x0=x(11);
A=y(11);
B=z(11);
C=w(11)/2;
D=v(11)/6;
E=q(11)/24;
function y=g(x), y=A+B*(x-x0)+C*(x-x0)^2+
D*(x-x0)^3+E*(x-x0)^4   , endfunction;
```

10. Строим график функции g(x)

```
gg=g(t);
plot(t,gg)
```

Получаем

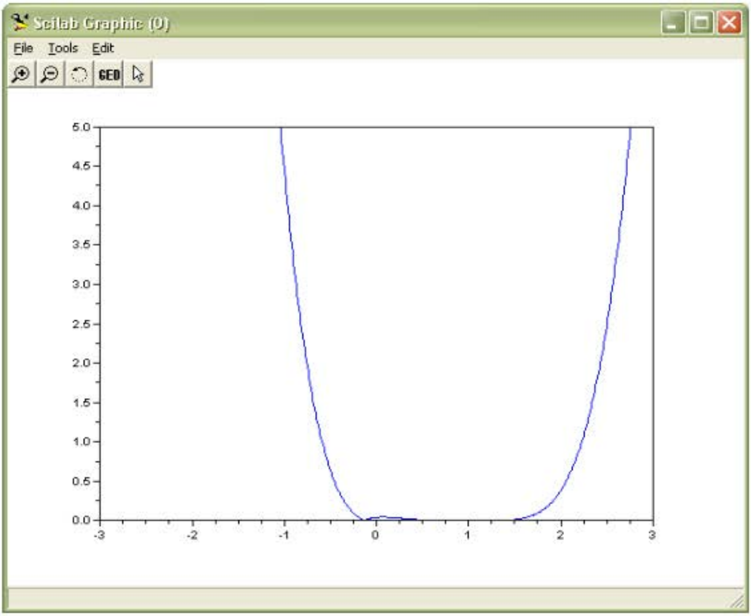


11. Введем функцию ошибки приближения рядом Тейлора

```
function e=err(x), e=abs(f(x)-g(x)), endfunction;
```

12. Построим график функции ошибки

```
ee=err(t);
plot(t,ee)
```



Решение нелинейных уравнений в Scilab

Предметная область

Решение нелинейного уравнений f(x)=0 в Scilab осуществляется с использованием функции fsolve. Решение ищется в окрестности предполагаемого значения x0. Для его определения проводится локализация решений по предвариетльно построенному графику f(x). Результаты выводятся в командное окно или в строку заголовка графика.

Контрольные вопросы

1. Задание функции пользователя.
2. Локализация решений уравнения.
3. Решение нелинейного уравнения с использованием функции fsolve.
4. Вывод полученных решений уравнения.
5. Локализация решений системы из двух уравнений.
6. Решение системы из двух уравнений.
7. Вывод полученных решений системы уравнений.

Задание к работе

Задача 1. Решение нелинейного уравнения.

- Создать программу решения нелинейного уравнений в редакторе scipad.
- В программе определить функцию f1(x).
- Вывести y1=f1(x) в виде XY графика. По нему определить приближенно корни уравнения y1(x)=0. Если корни на графике не просматриваются, то изменить пределы изменения аргумента и повторить операции.
- Для каждого корня найти точное значение, используя функцию fsolve. Перед расчетами задать приближенное значение корня x0.
- Сформировать строку с результатами и вывести ее в заголовок окна графика.

Задача 2. Решение системы из двух нелинейных уравнений.

- Создать программу решения нелинейных уравнений в редакторе scipad..
- В программе определить функции f1(x), f2(x), f3(x)=f2(x)-f1(x).
- Вывести y3=f1(x) в виде XY графика. По нему определить приближенно корни уравнения y3(x)=0. Если корни на графике не просматриваются, то изменить пределы изменения аргумента и повторить операции.

- Для каждого корня найти точное значение, используя функцию fsolve. Перед расчетами задать приближенное значение корня x0.
- Сформировать строку с результатами и вывести ее в заголовок окна графика.

Варианты заданий

№	f1(x)- полином 3-ей степени с коэффициентами a				f2(x)
	a3	a2	a1	a0	
1	0	-1	4	-1	0.2exp(x)-20
2	0	2	-2	-15	40|cos(x)|
3	0	1	4	-1	10ln(x+5.5)
4	0	9	-8	-70	100|sin(x)|
5	0	-4	4	50	70cos(x)
6	.1	-5	4	40	60exp(|0.1*x|)-100
7	.2	-3	2	30	20sin(2x)
8	.3	-6	1	50	exp(|x|)sin(2x)

Методические указания

При решении нелинейного уравнения оно формируется из функций задания, как f1(x)=0. При решении системы из двух нелинейных уравнений из функций задания формируется уравнение f3(x) = f1(x) - f2(x) = 0.

Локализация корней. Уравнение или система уравнений может иметь несколько корней, каждый из которых ищется отдельно. При этом для каждого корня надо задать диапазон аргумента, в котором он находится (только один!).

Это делается путем локализации корня. Для этого надо просчитать значения функций в заданном интервале и построить их графики. Начальное значение для решения одного уравнения - точка пересечения графиком функции оси X. График выводится процедурой, в которой аргументы - переменная x и анализируемая функция. С помощью grid on график делается с координатной сеткой:

plot(x,f1(x)) ;xgrid();

Начальное значение для решения системы из двух уравнений - точка взаимного пересечения графиков функций. Графики выводятся процедурой, в которой для каждого графика следует группа параметров:

plot(x,f(x),x,f2(x)); xgrid();

Функция fsolve. Используется для нахождения корня нелинейного уравнения. Формат этой функции:

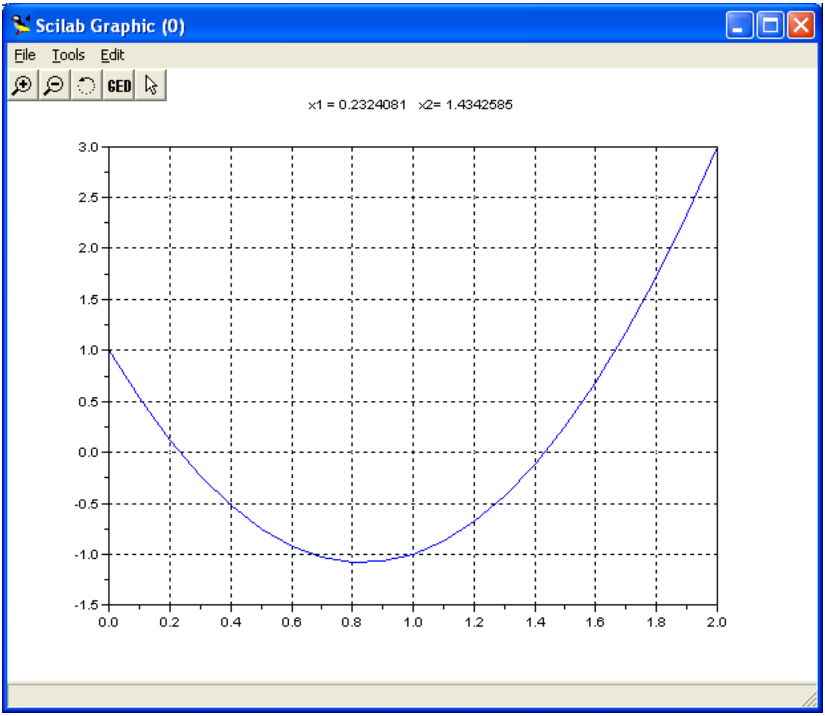
<имя результата>=fsolve(x0, f1)

Пример 1

Листинг программы

```
// Решение нелинейного уравнения
function y1=f1(x);      // Функция f1
        y1=x+3*(x-1)^2-2;
endfunction
x=0:0.1:2;
plot(x,f1(x)); xgrid;   // Графики
x0=0.2;
x1=fsolve(x0,f1)        // Корень 1
x0=1.4;
x2=fsolve(x0,f1)        // Корень 2
rezult='x1 = '+string(x1)+'   x2= '+string(x2);
title(rezult);
```

В программе ищутся 2 корня. Их приближенные значения 0.2 и 1.4 определены при пробном прогоне программы. Окончательный результат в окне графики.



Пример 2

Листинг программы

```
// Решение системы нелинейных уравнений
function y1=f1(x);              // Функция f1
      y1=x+3*(x-1)^2-2;
endfunction
function y2=f2(x);              // Функция f2
      y2=x-1;
endfunction
function y3=f3(x);               // Функция f3
      y3=f1(x)-f2(x);
endfunction
x=0:0.1:2;
plot(x,f1(x),x,f2(x)); xgrid;   // Графики
x0=0.4;
x1=fsolve(x0,f3)                 // Корень 1
x0=1.5;
```

```
x2=fsolve(x0,f3)                    // Корень 2
rezult='x1 = '+string(x1)+'   x2= '+string(x2);
title(rezult);
```

В программе ищутся 2 корня. Их приближенные значения 0.4 и 1.5 определены при пробном прогоне программы. Окончательный результат в окне графики.