Лабораторная работа №1 «Среда имитационного моделирования» (5 баллов)

1. Обзор пакета Scilab

Scilab – мощная интерактивная система автоматизации инженерных, научных и математических расчетов, построенная на расширенном представлении и применении матричных операций.

Установка пакета sci-lab: https://wiki.scilab.org/

Некоторые возможности системы:

- В области математических вычислений:
  - матричные, векторные, логические, условные операторы;
  - символьные вычисления;
  - полиномиальные и рациональные функции;
  - элементарные и специальные функции;
  - полиномиальная арифметика.
- В области реализации численных методов:
  - решение дифференциальных уравнений;
  - численное интегрирование;
  - поиск корней нелинейных алгебраических уравнений;
  - оптимизация функций нескольких переменных;
  - одномерная и многомерная интерполяция;
  - решение задач математической статистики.
- В области программирования:
  - свыше 500 встроенных математических функций;
  - интерфейс к Fortran, Tcl/Tk, C, C^{++}, Java, LabView.
- В области визуализации результатов расчетов и графики:
  - возможности создания и редактирования двухмерных и трехмерных графиков;
  - проведение визуального анализа данных.

- Обработка сигналов
- Параллельная работа
- Статистика
- Работа с компьютерной алгеброй

Scilab имеет схожий с MATLAB язык программирования, в составе имеется утилита, позволяющая конвертировать документы Matlab → Scilab.

Программа доступна для различных операционных систем, включая GNU/Linux и Microsoft Windows.

Scilab состоит из 3-х частей:

- интерпретатор
- библиотека функций (Scilab-процедуры)
- библиотека Fortran и C процедур

Упражнение 1.

```
--> v=[1 2 3 4]
v =
    1.    2.    3.    4.
--> m=[1, 2;  3, 4]
m =
    1.    2.
    3.    4.
--> sin(v)
ans =
   0.8414750   0.9092974   0.1411200  -0.7568025
--> 3*v
ans =
    3.    6.    9.    12.
```

Математические выражения в Scilab

Математические выражения состоят из чисел, констант, переменных, операторов, функций и спецзнаков. Числа могут быть целыми, дробными, с фиксированной и плавающей точкой. Примеры: -3 2.453 123.12e-3. Последнее число – это 123.12*10-3, т.е. 0,12312. Для разделения целой и десятичной части числа используется точка. Числа могут быть вещественными и комплексными. Кроме того, в Scilab существуют так называемые системные переменные и символьные константы:

- %i – мнимая единица (%i=);
- %pi – число π=3,1415927;
- %e – число e=2,71828184;
- %eps – 2.22d-16;
- %inf – значение машинной бесконечности;
- ans – переменная, хранящая результат последней операции;
- %nan – указание на нечисловой характер данных (not-a-number).

В памяти компьютера переменные занимают определенное место, называемое рабочим пространством (Workspace). Для очистки рабочего пространства используют функцию clear:

- **clear** – уничтожение определений всех переменных;
- **clear x** – уничтожение определения переменной x;
- **clear a, b, c** – уничтожение определений нескольких переменных.

Упражнение 2.

```
--> v1=[2 4 6 8];  v2=[1, 4, 12, 24];  p=v1/v2,  t=v1.*v2,  r=v1./v2, h=v1.\v2
p =
    0.3826323
t =
    2    16    72    192
r =
    2.   1.   0.5    0.3333333
h =
```

```
 0.5    1.    2.   3.
-->(2*1+4*4+6*12+8*24)/(1^2+12^2+4^2+24^2)
 ans  =
   0.3826323
```

Функции – это имеющие уникальные имена подпрограммы, выполняющие определенные преобразования над своими аргументами и при этом возвращающие результаты этих преобразований.

Функции (макросы) в Scilab похожи на те, что встречаются в других языках программирования. Функции могут иметь аргумент, сами являться аргументом другой функции, быть членом списка, участвовать в операциях сравнения, вызываться рекурсивно. Функция начинается со слова function и заканчивается словом endfunction.

Первая строка функции может быть следующей:

function var=my_name(x1,...,xk),

где var - имя переменной, а xi - входные переменные.

Упражнение 3.

Наберите пример функции, вычисляющей сумму положительных элементов в массиве v

```
  function g=f(v)
  s=0;  n=length(v);
    for i=1 : n
       if  v(i)>0  then
          s=s+v(i);
       end
    end
  g=s;
  endfunction
--> x=[1 2 5 -3 7 -9 12];  t=f(x)
t =
27
```

Операторы и функции

Некоторые специальные символы:

a*(: , j)** – j-й столбец матрицы ***a;

a*(i , :)** – i-я строка матрицы ***a;

***a*(j : k)** – элементы a_j, a_{j+1}, …, a_k;

a*(:)** – записывает все элементы массива ***a в виде столбца;

***a*(m , :) =[ ]** – удаляет из матрицы строку **m**;

a*'** – транспонированная матрица ***a;

***a*.'** – транспонирование массива;

prod(*a*) – произведение элементов массива;

prod(*a*, dim) – произведение элементов столбцов (**dim=1**) или строк (**dim=2**) ;

sum(*a*) – сумма элементов массива;

sum(*a*, dim) - сумма элементов столбцов (**dim=1**) или строк (**dim=2**).

Матричные операции линейной алгебры

- **det(*a*)** – возвращает определитель квадратной матрицы ***a***;
- **rank(*a*)** – возвращает ранг матрицы;
- **norm(*a*)** – возвращает норму матрицы ***a***;
- **b=orth(*a*)** – возвращает ортонормальный базис матрицы ***a***;
- **inv(*a*)** – возвращает матрицу, обратную матрице ***a***;
- **spec(*a*)** – возвращает вектор собственных значений матрицы ***a***.

Упражнение 4.

Решить систему линейных уравнений (д.з)

$$\begin{aligned} 2x_1 + x_2 + x_4 &= 8 \\ x_1 - 3x_2 + 2x_3 + 4x_4 &= 9 \\ -5x_1 - x_3 - 7x_4 &= -5 \\ x_1 - 6x_2 + 2x_3 + 6x_4 &= 0 \end{aligned}$$

Упражнение 5.

Решить уравнение $7x^3 + 45x^2 + 12x + 23 = 0$
И построить график функции $y = 7x^3 + 45x^2 + 12x + 23$ на отрезке [-8; -5]

-->x=-8:0.1:-5; plot(x, f(x)); xgrid()

Библиотека функций распределения

Как построить создать последовательность случайных
величин?

С помощью команды **rand.**
Синтаксис
rand (m1,m2,.. [,key])
rand(x [, key])
rand()
rand(key)
rand("seed" [,n])
rand("info")

Параметры
mi : целые числа
key : символьная переменная, принимающая значение "uniform"и "normal". Значение "uniform" соответствует равномерному распределению, а "normal" - Гауссовскому.

x : матрица. Во внимание принимается только ее размер.

Примеры использования.
1) rand(m1,m2) Образование случайной матрицы из m1 строк и m2 столбцов (m1 на m2).
2) rand(m1,m2,..,mn) Образование случайной матрицы размером m1 на m2,.. на mn.
3) rand(A) Образование случайной матрицы такого же размера, какого была матрица A. Матрица rand(A) комплексная, если матрица A была комплексная. s=1:4; a=rand(s) a = ! .9184708 .0437334 .4818509 .2639556 !
4) rand() без аргументов дает случайное скалярное число, случайным образом изменяющееся при следующем вызове.
5)rand("normal") или rand("uniform") позволяют задать распределение случайных чисел.
6)rand("info") возвращает значение переменной key.

Как построить последовательность случайных величин с заданным распределением?

С помощью команды **grand**, которая тоже является генератором случайных чисел, но более сложным.
В отличии от команды rand, команда grand может создавать последовательности случайных чисел из различных распределений.
Синтаксис
Y=grand(m, n, dist_type [,p1,...,pk])
Y=grand(X, dist_type [,p1,...,pk])
Y=grand(n, dist_type [,p1,...,pk])
S=grand(action [,q1,....,ql])
Параметры
m, n : целые числа, определяющие желаемый размер матрицы **Y**
X : матрица. Используется только данные о ее размере (задает m на n)
dist_type : символьная переменная, указывающая, распределение случайной величины какого типа мы создаем. Принимает определенные значения ('bin', 'nor', 'poi', ...)
p1, ..., pk : параметры (действительные или целые), необходимые для полного определения распределения типа **dist_type**
Y : результирующая матрица из случайных величин размера **m** на **n**
action : символьная переменная, указывающая на базу генератора (ов). Значение параметра **action**, равное '**setgen**' меняет текущую базу генератора, 'getgen' извлекает имя текущего базового генератора, 'getsd' извлекает начальное число для генерации случайных чисел текущего генератора и т.д.
q1, ..., ql : параметры (обычно строка) необходимые для определения параметра action.
S: результат применения команды (обычно строка или столбец действительных векторов)

С помощью команды **grand** можно получить следующие распределения случайных чисел:
• Бета -распределение
• Биномиальное распределение (два варианта)
• Распределение hi-квадрат (два варианта)
• Экспоненциальное распределение
• F- дисперсионное распределение (Фишера) (два варианта)
• Гамма распределение
• Нормальное распределение Гаусса-Лапласа
• Мультивариативное Гауссово распределение
• Геометрическое распределение
• Распределение Маркова
• Полиномиальное распределение

• Распределение Пуассона
• Распределение, состоящее из перестановки из n-элементов
• Равномерное распределение (четыре варианта)

Подробный синтаксис этих команд смотрите с помощью команды **help grand**.
В help grand приводятся параметры команды, отвечающие за разные варианты генератора случайных чисел.

Пример:

Создадим и построим экспоненциальное распределение случайных величин со средним арифметическим, равным Av.

```
m=100;
n=1;
Av=2;
```
// Av -величина среднего значения для построенной последовательности случайных величин
```
Y=grand(m,n,'exp',Av);
plot2d(1:100,Y);
```
// Черная линия
```
Av2=5;
Y2=grand(m,n,'exp',Av2);
plot2d(1:100,Y2,5);
```
// Красная линия

Упражнение 6.

Построить матрицу с количеством строк 10 и столбцов 2, переменные заполняются случайными числами с равномерным распределением.
Нанести полученные сгенерированные случайные числа на график в виде точек
```
plot(x(:,1),x(:,2),'.')
```

Упражнение 7.

Сгенерировать 1000 случайных чисел с любым известным распределением и построить их гистограмму, математическое ожидание, дисперсию.

Упражнение 8.

Сгенерировать случайную точку равномерно распределенную в квадрате со стороной *a*.