

# Лабораторная работа №1 по курсу «Радиотехнические устройства и системы»

## Введение в GNU/Octave

Кузнецов В.В., ассистент кафедры ЭИУ1-КФ

10 сентября 2013 г.

## 1 Цель работы

Целью семинара является ознакомление с базовыми принципами проведения вычислений системы численной математики GNU/Octave.

Система GNU/Octave — это высокоуровневый язык программирования, предназначенный прежде всего для численных расчётов. Он предоставляет удобный интерфейс командной строки для численного решения линейных и нелинейных задач, а также для выполнения других численных экспериментов. С помощью GNU/Octave можно решать задачи в том числе генерации и обработки сигналов. GNU/Octave — кроссплатформенное приложение. Установить GNU/Octave для Linux можно в один клик через пакетный менеджер, а для Windows её можно бесплатно скачать с сайта разработчика <http://octave.sourceforge.net>.

Octave работает в режиме командной строки. Octave позволяет выполнять операции с действительными и комплексными числами, матрицами, решать системы линейных уравнений, обрабатывать данные, строить графики и диаграммы. Синтаксис команд Octave близок к языку C и повторяет среду Matlab.

В ходе выполнения лабораторной работы необходимо ознакомиться с принципами выполнения вычислений в системе GNU/Octave. Для этого необходимо выполнить задания, приведённые ниже.

При подготовке руководства использовались материалы с сайтов <http://mydebianblog.blogspot.com> (на русском языке) и [http://en.wikibooks.org/wiki/Octave\\_Programming\\_Tutorial/Getting\\_started](http://en.wikibooks.org/wiki/Octave_Programming_Tutorial/Getting_started) (на английском языке).

## 2 Применение Octave для расчётов

### 2.1 Запуск и выход из программы

В состав пакета входит интерактивный командный интерфейс (интерпретатор). Интерпретатор Octave запускается из терминала ОС Linux или из его порта в Windows. После запуска Octave пользователь видит окно интерпретатора (см. рис. 1, могут быть небольшие различия в зависимости от используемой версии и ОС). Чтобы запустить GNU/Octave нужно для Linux набрать в терминале `octave`, а для Windows щёлкнуть по ярлыку, создаваемом на рабочем столе после установки программы.

Чтобы выйти из программы в командной строке нужно набрать `quit`. После того как команда напечатана, нужно нажать Enter.

```
GNU Octave, version 3.6.4
Copyright (C) 2013 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.  For details, type 'warranty'.

Octave was configured for "x86_64-suse-linux-gnu".

Additional information about Octave is available at http://www.octave.org.

Please contribute if you find this software useful.
For more information, visit http://www.octave.org/get-involved.html

Read http://www.octave.org/bugs.html to learn how to submit bug reports.

For information about changes from previous versions, type 'news'.

octave:1>
```

Рис. 1. Командная строка Octave

Чтобы получить справку о команде или функции Octave из командной строки нужно набрать `help имя_функции`. Например:

```
octave:1> help sqrt
'sqrt' is a built-in function

-- Mapping Function:  sqrt (X)
  Compute the square root of each element of X.  If X is negative, a
  complex result is returned.  To compute the matrix square root, see
  *note Linear Algebra::.

  See also: realsqrt, nthroot
```

Additional help for built-in functions and operators is available in the online version of the manual. Use the command `'doc <topic>'` to search the manual index.

Help and information about Octave is also available on the WWW at <http://www.octave.org> and via the [help@octave.org](mailto:help@octave.org) mailing list.

Выводится информация о функции `sqrt`, которая вычисляет квадратный корень. Вся справочная информация по команде `help` выводится на английском языке.

## 2.2 Комментарии и вывод результата вычислений

В окне интерпретатора пользователь может вводить как отдельные команды языка Octave, так и группы команд, объединяемые в программы. Если строка заканчивается символом ; (точка с запятой) результаты на экран не выводятся. Если же в конце строки символ отсутствует, результаты работы выводятся на экран (см. рис. 1.2). Текст в строке после символа % (процент) или # (решётка) является комментарием и интерпретатором не обрабатывается. Комментарии можно вводить и в командном режиме. Рассмотрим несколько несложных примеров.

## 2.3 Арифметические операции

Рассчитаем сколько будет, если сложить 2 и 3. Для этого введём в командной строке Octave следующие команды:

```
octave:1> 2+3  
ans = 5
```

Запись числе с плавающей точкой производится в GNU/Octave так же как и в языке С. Например число  $0.03 = 3 \cdot 10^{-2}$  можно записать как 0.03 и как 3e-2. И получим в результате 5. В переменной **ans** хранится результат последней операции. Доступны и все прочие арифметические операции. Синтаксис аналогичен языку С.

В GNU/Octave можно выполнять и операции с логарифмами, комплексными числами, тригонометрическими функциями. Наример:

Логарифмирование:

$$\frac{\log_{10} 100}{\log_{10} 10} \quad (1)$$

```
octave:1> log10(100)/log10(10)  
ans = 2
```

Тригонометрические функции (тангенс), округление до целого в меньшую сторону (**floor**):

$$\left\lfloor \frac{1 + \tan 1.2}{1.2} \right\rfloor \quad (2)$$

```
octave:3> floor((1+tan(1.2)) / 1.2)  
ans = 2
```

Корень квадратный **sqrt** и возведение в степень (операция **^**):

$$\sqrt{3^2 + 4^2} \quad (3)$$

```
octave:4> sqrt(3^2 + 4^2)  
ans = 5
```

Экспоненты (функция **exp**), комплексные числа (число  $i = \sqrt{-1}$  — мнимая единица) и число  $\pi$ . С комплексными числами можно производить те же действия, что и с обычными (вещественными) числами:

$$e^{i\pi} \quad (4)$$

```
octave:5> exp(i*pi)
ans = -1.0000e+00 + 1.2246e-16i
```

В Octave также можно объявлять переменные и составлять из них выражения. Синтаксис здесь такой же, как и в языке C. Регистр букв имеет значение. Применяются те же операции и скобки, как и в языке C.

## 2.4 Пакетный и терминальный режимы работы Octave

Возможны два варианта решения любой задачи в Octave:

1. Терминальный режим. В этом режиме в окно интерпретатора последовательно вводятся отдельные команды.

2. Программный режим. В этом режиме создаётся текстовый файл с расширением `.m`, в котором хранятся последовательно выполняемые команды Octave. Затем этот текстовый файл (программа на языке Octave) запускается на выполнение в среде Octave.

Рассмотрим пример решения задач в обоих режимах.

Рассчитаем резонансную частоту LC - контура по формуле, известной из курса электротехники:

$$f = \frac{1}{2\pi\sqrt{LC}} \quad (5)$$

Где  $L$  — индуктивность катушки, Гн;

$C_k$  — ёмкость конденсатора, Ф;

$f$  — частота, Гц

Чтобы решить нашу задачу в терминальном режиме введём последовательно после того как Octave запустится и выдаст приглашение, подобное рис.1, команды, показанные на рис. 2. На рисунке приведён вывод для Linux, отличие для системы Windows заключается только в способе запуска программы.

В переменной `ans` хранится результат последней операции, если команда не содержит знака присваивания. Следует помнить, что значение переменной `ans` изменяется после каждого вызова команды без операции присваивания.

Теперь рассмотрим, как решить эту же задачу в программном режиме. Вызовем любой текстовый редактор, например `kwrite` в системе Linux, но подойдёт даже обычный Notepad для Windows, в окне которого последовательно введём следующие команды, приведённые в листинге 1.

Листинг 1. Файл LC-kontur.m

```
1 #!/usr/bin/octave -qf # Первая строка --- тип интерпретатора
2 L=10e-6; # Объявим переменную L. Индуктивность катушки, Гн
3 C=47e-12; # Объявим переменную C. Ёмкость конденсатора, Ф
4 f=1/(2*pi*sqrt(L*C)) # Считаем резонансную частоту и
5 # выводим её на экран
6 #raise # Раскомментировать строку для Windows. Задерживает выполнение
   скрипта
7 # до тех пор пока пользователь не нажмёт Enter.
```

Сохраним введённые команды в виде файла `LC-kontur.m`. Скрипты Octave обычно имеют расширение `*.m`. Теперь эту программу необходимо запустить на выполнение. В системе Linux нужно сначала сделать файл исполняемым любым способом, например из командной строки или файлового менеджера. Чтобы дать скрипту права на исполнение и затем запустить его нужно выполнить в каталоге, где находится файл скрипта следующие команды:

```

vvk@linux-bmx0:~> octave
GNU Octave, version 3.6.4
Copyright (C) 2013 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type 'warranty'.

Octave was configured for "x86_64-suse-linux-gnu".

Additional information about Octave is available at http://www.octave.org.

Please contribute if you find this software useful.
For more information, visit http://www.octave.org/get-involved.html

Read http://www.octave.org/bugs.html to learn how to submit bug reports.

For information about changes from previous versions, type 'news'.

octave:1> L=10e-6;
octave:2> C=47e-12;
octave:3> f=1/(2*pi*sqrt(L*C))
f = 7.3413e+06
octave:4> quit
vvk@linux-bmx0:~>

```

Рис. 2. Расчёт резонансной частоты LC-контура в командной строке Octave

```

chmod a+x LC-kontur.m
./LC-kontur

```

В результате выполнения команд в окне терминала напечатается вычисленное значение резонансной частоты  $f = 7.3413e+06$ .

## 3 Матрицы и массивы

При обработке сигналов нам понадобится выполнять операции с одномерными массивами чисел с плавающей точкой (float). Одномерные массивы в Octave являются частным случаем матриц. Им соответствуют матрица-строка и матрица-столбец (вектор). Двухмерные матрицы — это двухмерные массивы. Индексация массивов начинается с единицы, в отличие от языка C.

### 3.1 Объявление матриц

Вектор — это матрица - строка, либо матрица - столбец. Векторы являются аналогами одномерных массивов в языке C.

Чтобы объявить массив нужно перечислить все его элементы через запятую , и заключить их в квадратные скобки [].

```

octave:1> x = [1, 3, 2]
x =

```

```
1 3 2
```

Чтобы задать вектор-столбец элементы в квадратных скобках нужно перечислить через точку с запятой ; :

```
octave:2> x = [1; 3; 2]
x =
```

```
1
3
2
```

Чтобы объявить матрицу, нужно перечислить каждый элемент строки через запятую, а строки разделить точкой с запятой ; .

```
octave:3> A = [1, 1, 2; 3, 5, 8; 13, 21, 34]
A =
```

```
1 1 2
3 5 8
13 21 34
```

## 3.2 Последовательности

Часто нужно сгенерировать последовательность чисел от начального значения до конечного с некоторым шагом. Например, последовательность чисел от 1 до 10 с шагом 3. Для этого служит операция : (двоеточие). Последовательности задаются в следующем формате:

`начальное_значение:шаг:конечное_значение`

Последовательности представляются в GNU/Octave матрицей-строкой. Следующая операция сгенерирует последовательность чисел от 1 до 10 с шагом 2:

```
octave:6> 1:3:10
ans =
```

```
1 4 7 10
```

Для генерации матрицы, строки, содержащего последовательность из  $N$  чисел от начального до конечного ещё служит функция `linspace`. Например следующий вызов генерирует матрицу-строку, содержащую 10 чисел в равномерно распределённых в интервале от 0 до 2:

```
octave:7> linspace(0,2,10)
ans =
```

```
0.00000 0.22222 0.44444 0.66667 0.88889 1.11111 1.33333 1.55556
1.77778 2.00000
```

### 3.3 Операции над матрицами

К матрицам и массивам в Octave можно применять все операции, известные из курса линейной алгебры: транспонирование, умножение на число, сложение, вычитание, умножение матриц, вычисление обратной матрицы.

Рассмотрим операцию транспонирования. Транспонирование — это замена строк на столбцы. Если транспонировать матрицу-строку, то получим матрицу-столбец и наоборот. Для этого после матрицы нужно поставить знак апострофа `'`.

```
octave:4> A'  
ans =
```

```
1  3 13  
1  5 21  
2  8 34
```

Матрицу можно умножать и делить на число. При этом каждый элемент матрицы делится или умножается на число:

Например:

```
octave:5> x = [1, 3, 2]  
x =  
1  3  2  
octave:6> 2*x  
ans =  
2  6  4
```

Матрицы в Octave можно складывать так же как и обычные числа при помощи операции `+`. При этом первый элемент складывается с первым, второй со вторым и т.д. Этим пользуются для сложения сигналов.

### 3.4 Поэлементные операции над матрицами

В GNU/Octave, как и в Matlab, можно применять поэлементные операции над матрицами. Используется поэлементное сложение, умножение, деление.

Рассмотрим пример. Разделим поэлементно матрицу  $A$  на матрицу  $B$ :

```
octave:1> A = [1, 6, 3; 2, 7, 4]  
A =
```

```
1  6  3  
2  7  4
```

```
octave:2> B = [2, 7, 2; 7, 3, 9]  
B =
```

```
2  7  2  
7  3  9
```

```
octave:3> A ./ B  
ans =
```

```
0.50000  0.85714  1.50000  
0.28571  2.33333  0.44444
```

В результате исполнения этих команд первый элемент матрицы  $A$  разделится на первый элемент матрицы  $B$ , второй — на второй, и так далее.

### 3.5 Индексация массивов и матриц

Индексация служит для обращения к отдельному элементу массива, как в языке C. Знаком операции индексации являются круглые скобки `()`.

Рассмотрим пример. Создадим массив (вектор-строку) из трёх элементов и обратимся ко второму элементу. Отличие от языка C здесь заключается в том, что нумерация элементов начинается с единицы, а не с нуля.

```
octave:1> x = [1.2, 5, 7.6, 3, 8]
x =

    1.2000    5.0000    7.6000    3.0000    8.0000
```

Теперь посмотрим чему равен второй элемент массива.

```
octave:2> x(2)
ans = 5
```

Чтобы отсечь часть элементов массива служит операция `1:3`. Например, чтобы получить массив, содержащий элементы исходного массива с первого по третий нужно использовать следующие команды:

```
octave:3> x(1:3)
ans =

    1.2000    5.0000    7.6000
```

Чтобы узнать длину массива (то есть количество элементов в массиве), следует использовать команду `length`.

```
octave:6> length(x)
ans = 5
```

Длина массива  $x$  равна 5 элементам.

### 3.6 Построение графиков

Для построения графиков в Octave служит функция `plot(x,y)`. В качестве аргументов ей передаются два вектора (матрица-столбец, либо матрица-строка), содержащие точки по осям  $X$  и  $Y$  соответственно. Количество элементов в векторах должно быть одинаковым.

Для примера построим график функции  $\sin(t)$  на отрезке от 0 до  $2\pi$  по 20 точкам. Команды, вводимые в окно Octave приведены ниже.

```
GNU Octave, version 3.6.4
octave:1> t=0:(2*pi/20):2*pi # генерируем вектор из 20 точек от 0 до 2*pi
t =

Columns 1 through 9:
```



```
0.00000    0.31416    0.62832    0.94248    1.25664    1.57080    1.88496
2.19911    2.51327
```

```
Columns 10 through 18:
```

```
2.82743    3.14159    3.45575    3.76991    4.08407    4.39823    4.71239
5.02655    5.34071
```

```
Columns 19 through 21:
```

```
5.65487    5.96903    6.28319
```

```
octave:2> s=sin(t) # вычисляем таблицу функции sin()
s =
```

```
Columns 1 through 10:
```

```
0.00000    0.30902    0.58779    0.80902    0.95106    1.00000    0.95106    0.80902
0.58779    0.30902
```

```
Columns 11 through 20:
```

```
0.00000   -0.30902   -0.58779   -0.80902   -0.95106   -1.00000   -0.95106   -0.80902
0.58779   -0.30902
```

```
Column 21:
```

```
-0.00000
```

```
octave:3> plot(t,s); # строим график
octave:4>
```

Если ввести в Octave указанные команды, то откроется графическое окно (см. рис. 3), в котором будет построена синусоида.

Как видно из простейших примеров, у Octave достаточно широкие возможности, а по синтаксису он близок к Matlab.

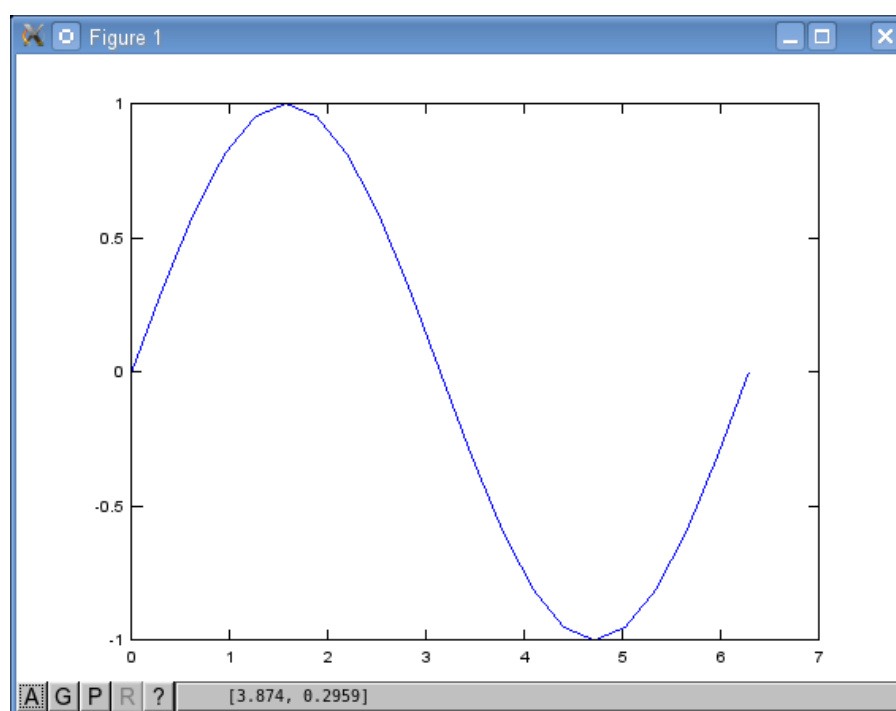


Рис. 3. Графическое окно системы Octave