

# Octave 学习

吴艺<sup>Ⓔ</sup> 1601110035<sup>\*</sup>

2016 年 10 月 21 日

## 目录

<b>1</b>	<b>Octave 简介</b>	<b>3</b>
<b>2</b>	<b>Octave 的使用</b>	<b>3</b>
2.1	安装及进入界面 . . . . .	3
2.2	简单计算 . . . . .	4
2.3	分号和隐藏结果 . . . . .	4
2.4	内建函数 . . . . .	5
2.5	取消一个命令 . . . . .	6
2.6	help . . . . .	6
<b>3</b>	<b>矩阵和向量</b>	<b>8</b>
3.1	矩阵的构建 . . . . .	8
3.1.1	方括号 . . . . .	8
3.1.2	冒号 . . . . .	8
3.2	矩阵的操作 . . . . .	9
<b>4</b>	<b>作图</b>	<b>11</b>
4.1	基本绘图: plot . . . . .	11
4.2	持续操作: hold on . . . . .	11
4.3	创建子图: subplot . . . . .	12
4.4	3D 绘图: plot3 . . . . .	13
4.5	绘制曲面: meshgrid, surf . . . . .	13
<b>5</b>	<b>脚本文件与函数</b>	<b>15</b>
5.1	控制语句 . . . . .	15
5.1.1	if...else 语句 . . . . .	15
5.1.2	for 语句 . . . . .	16
5.1.3	while 语句 . . . . .	16

---

<sup>\*</sup>北京大学数学科学学院, 科学与工程计算系, 邮箱: wuyichongyt@sina.cn

目录	2
5.2 函数 . . . . .	16
6 与 MATLAB 的区别	17
7 总结	20

## 1 Octave 简介

Octave 是一款用于数值计算和绘图的软件,它属于 GNU 计划,因此是开源的。和 MATLAB 类似,它尤其精于矩阵运算,能够批量处理相关问题并很好的将数据可视化。它是一门基于 C++ 的解释性语言,相比于编译性语言,转化为机器语言的速度更慢,但其代码功能实现上的优势弥补了这一缺陷,大大提高了有关工作的效率。

最初,Octave 是作为一款本科生化学课程的辅助程序由 Octave Levenspiel 等人开发的,它的名字也由此而来。真正完整的发展是由 John W.Eaton 于 1992 年开始的,1993 年 1 月 4 日公布第一个 alpha 版本,1994 年 2 月 17 日 1.0 版本公布,目前的 4.0.0 版本于 2015 年 5 月 29 日公布。

Octave 与 MATLAB 有着极大地相似性,事实上,在它的开发过程中就视与 MATLAB 的兼容性为一项要求,不兼容将被视作 Bug。因此,Octave 可以认为是在不侵权的前提下对 MATLAB 的软件“克隆”。那么,学习 Octave 的过程中,抓住它与 MATLAB 的异同是关键,尤其是对于有 MATLAB 基础的学习者来说。

## 2 Octave 的使用

### 2.1 安装及进入界面

安装十分简单,在命令行中输入 `sudo apt-get install octave` 即可,安装的是 4.0.0 版本。值得一提的是从 3.8.0 版本开始,Octave 支持图形用户界面 (GUI),安装完毕后在命令行输入 `octave` 将自动进入 GUI,若想在命令行界面中使用,应输入 `octave -no-gui`。启动后如下

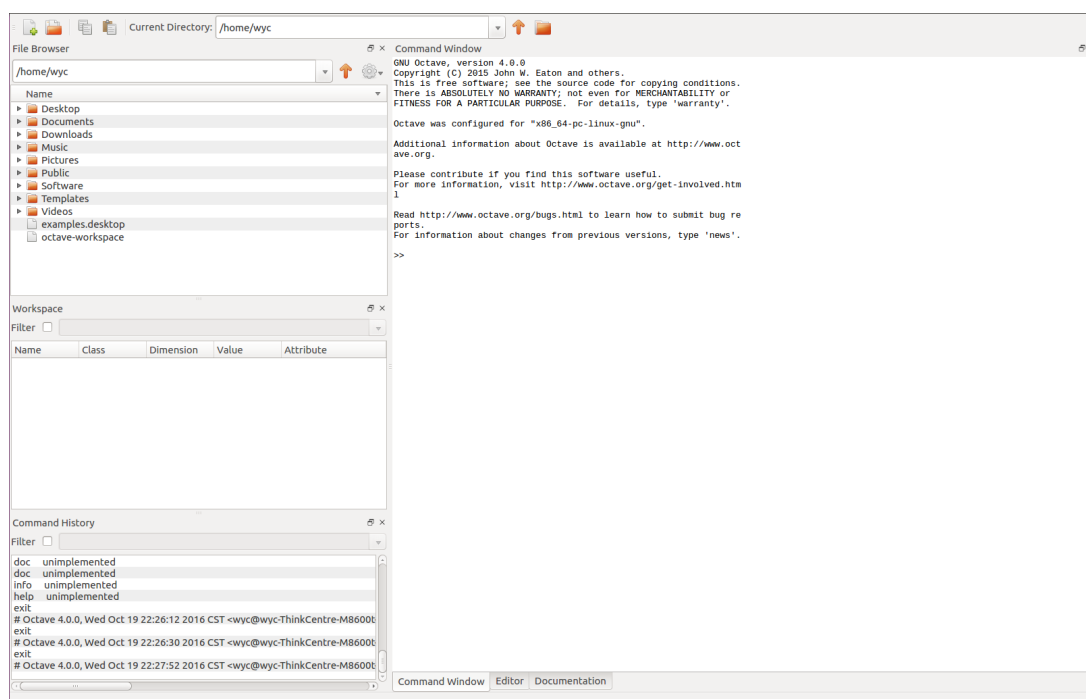


图 1: 图形用户界面

```
wycIsWYC ~:octave --no-gui
GNU Octave, version 4.0.0
Copyright (C) 2015 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-pc-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'.

>> █
```

图 2: 命令行界面

## 2.2 简单计算

Octave 最简单的使用方式就是像使用计算器一样在命令提示符下输入相应的计算式, 它能识别通常的计算表达式。各种计算符号的优先级与常规的一致, 比如括号有最大优先级, 其次为乘方, 其次为乘除运算, 最后为加减运算。

```
>> ((2+2)*5)^2-100
ans = 300
>> █
```

图 3: 简单计算

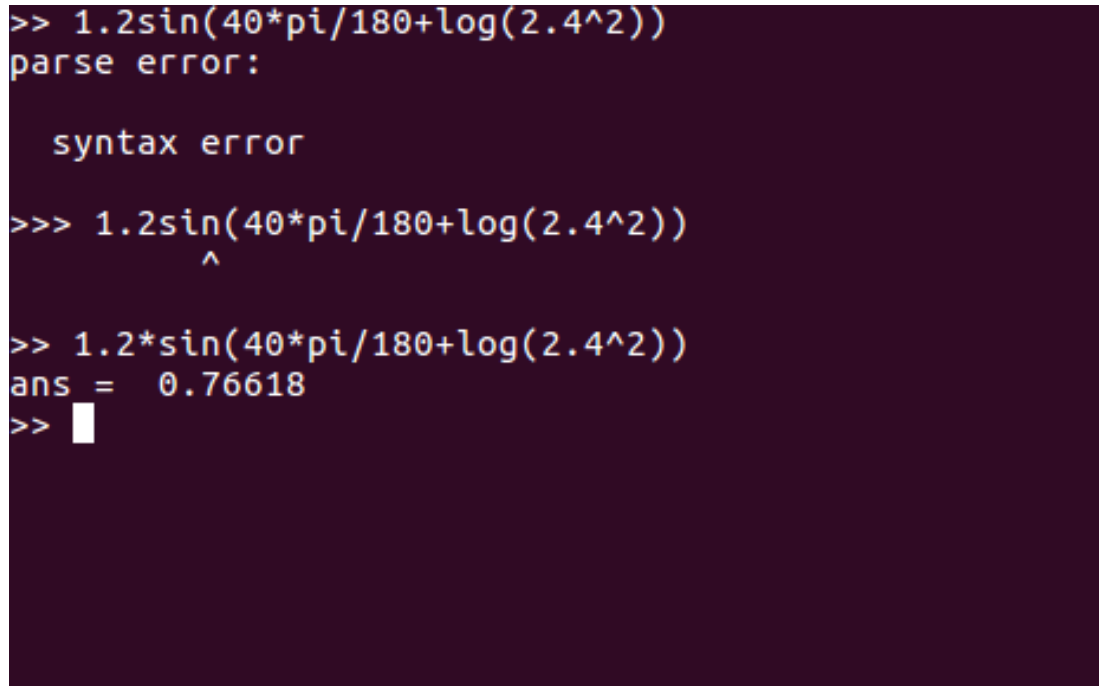
## 2.3 分号和隐藏结果

分号在通常的编程语言例如 C++ 中被用来表示程序块或者单个语句的结束。然而在 Octave 中分号的作用并非如此。在之前的例子中, 我们输入一个 Octave 命令结尾处没有加上分号, 那么 Octave 会将语句执行的结果随即显示出来。而如果我们在一行语句的末尾添上了分号, Octave

将不会显出相应的结果。这样做有时更利于我们观察运行情况，而不会被中间的一些无关紧要的数据所干扰。

## 2.4 内建函数

除了这些最基本的计算符,Octave 还提供了一系列的常用数学函数。通过调用这些函数，可以方便有效的完成我们的工作。与 C++ 调用函数一样,Octave 通过输入函数名和括号中的输入参数来调用函数, 例如计算  $1.2 \sin(40^\circ + \ln(2.4^2))$



```
>> 1.2sin(40*pi/180+log(2.4^2))
parse error:

syntax error

>>> 1.2sin(40*pi/180+log(2.4^2))
      ^

>> 1.2*sin(40*pi/180+log(2.4^2))
ans = 0.76618
>> 
```

图 4: 内建函数的使用

在刚才的例子中, 需要注意的是: 在计算表达式中一个明确的乘号是必不可少的, 比如在 1.2 和 sin 间, 这和我们手写的习惯不同。下面给出一些常用的基本函数

cos	余弦函数 (弧度制)
sin	正弦函数 (弧度制)
tan	正切函数 (弧度制)
exp	指数函数 ( $e^x$ )
log	以 $e$ 为底的指数函数
log10	以 10 为底的指数函数
sinh	双曲正弦函数
tanh	双曲正切函数
cosh	双曲余弦函数
acos	反余弦函数
acosh	反双曲余弦函数
asin	反正弦函数
asinh	反双曲正弦函数
atan	反正切函数
atan2	双参数形式的反正切函数
atanh	反双曲正切函数
abs	绝对值函数 (复数取模)
sign	符号函数
round	四舍五入
floor	近似为比它小的最大整数
ceil	近似为比它大的最小整数
fix	向 0 方向近似
rem	求余数

表 1: 基本数学函数

## 2.5 取消一个命令

如果你发现输入的一个命令执行了很长的时间都没有结束, 中止该程序的执行就变得又必要。你可以在当前命令行下输入 Ctrl-C, 程序将被中止并返回到命令提示界面。至于究竟是你程序自身的 Bug 还是其他原因导致命令的执行缓慢, 就需要你去进一步探索。

## 2.6 help

Octave 的功能十分强大, 它有许多命令以及内建函数。想要一次性全部掌握是不太现实的。如果你刚接触某个命令或者函数, 还不了解它的使用方式, Octave 本身的帮助系统会很有用, 它能让你第一时间清楚相关信息。例如我们想要知道函数 sqrt 的介绍, 最基本的使用帮助系统的方式就是

```
>> help sqrt
'sqrt' is a built-in function from the file libinterp/corefcn/mappers.cc

-- 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
mailing list.
>> █
```

图 5: help 的使用

这个函数比较简单，对于一些复杂的函数，通过 help 可以清晰的了解它的使用方法。

如果你不知道你需要的某个函数的具体名称，有个方法可以帮助你找到这样的函数是否存在。通过命令行输入 `help -list`, Octave 将给出一个其帮助的主题列表。你可以从中找到你想要的功能。

```
*** operators:

!   #   %   &   (   **   ,   .'   ...   .^   ;   =   >=   ]   ||
!=  #{  %{  &&  )   +   -   .*   ./   /   <   ==   [   ^   ~
"   #}  %}   '   *   ++  --  .**  .\   :   <=   >   \   |   ~=

*** keywords:

break          endfunction      persistent
case           endif          return
catch          endparfor      static
continue       endswitch     switch
do             endwhile      try
else           for            until
elseif        function       unwind_protect
end            global        unwind_protect_cleanup
end_try_catch  if             varargin
end_unwind_protect otherwise    varargout
endfor        parfor         while

*** builtins:

EDITOR          getpwuid
EXEC_PATH       gettrusage
F_DUPFD         getuid
F_GETFD         givens
-- less -- (f)orward, (b)ack, (q)uit
```

图 6: help -list 结果

## 3 矩阵和向量

要想在 Octave 中高效的进行计算，就得利用好它的矩阵运算机制。遇到问题时，尽量使用矩阵（向量也是矩阵）代替循环，这样可以节省大量运算。

### 3.1 矩阵的构建

#### 3.1.1 方括号

构建矩阵的方法有很多。其中最直接简单的方法就是在一个方括号 `[]` 中给出其元素。在方括号中由空格或者逗号隔开的一组数据被定义为行向量；而由分号或者回车隔开的一组数据被定义为列向量。

```
>> row = [0,1,2]
row =

    0    1    2

>> col = [0;1;2]
col =

    0
    1
    2

>> █
```

#### 3.1.2 冒号

冒号也是很重要的一个工具，在构建矩阵以及提取矩阵元素中都有着很大的作用。

```
>> mat = zeros(5);
>> mat(1,:) = 1:5
mat =

    1    2    3    4    5
    0    0    0    0    0
    0    0    0    0    0
    0    0    0    0    0
    0    0    0    0    0

>> mat(:,1) = 1:2:9
mat =

    1    2    3    4    5
    3    0    0    0    0
    5    0    0    0    0
    7    0    0    0    0
    9    0    0    0    0

>> █
```

单独的一个冒号表示整行或整列； $a:b$  表示一个以  $a$  为首项，1 为公差，最后一项不大于  $b$



的等差数列（这是  $a < b$  的情况，反之类似）； $a:b:c$  表示以  $a$  为首项， $b$  为公差，最后一项不大于  $c$  的等差数列。

一些常用的构建矩阵的方法如下

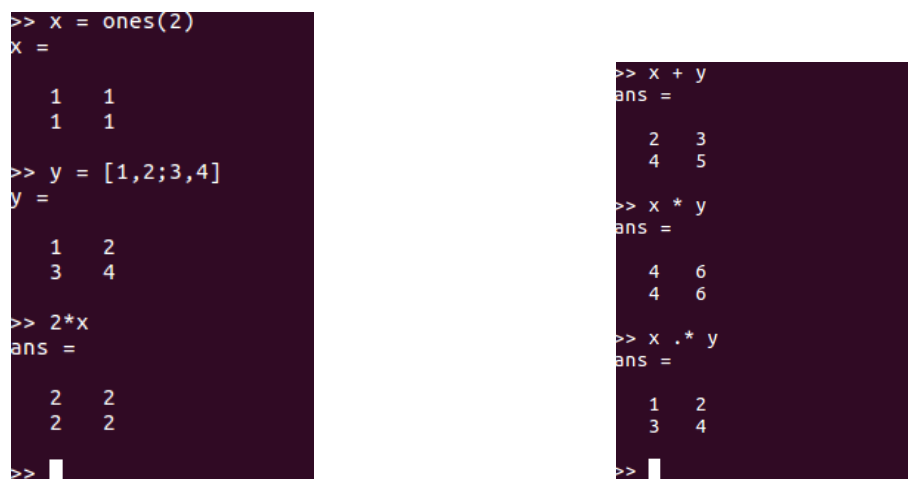
eye	创建单位矩阵
zeros	创建全零矩阵
ones	创建全一矩阵
linspace(x1,x2,N)	创建一个 $N$ 个元素的向量，均匀分布于 $[x_1, x_2]$
logspace(x1,x2,N)	创建一个 $N$ 个元素的向量，指数分布与 $[10^{x_1}, 10^{x_2}]$
rand	创建随机数矩阵
diag	创建一个对角矩阵，或者提取一个矩阵的对角元

表 2: 矩阵的构建

### 3.2 矩阵的操作

矩阵中的元素通过括号  $()$  提取，而第一个元素的编号为 1，而不是像 C 或者 C++ 那样从 0 开始。

在 C++ 中如果你想进行相同的计算，例如每个元素乘以 2，你需要使用 for 循环来对每个元素操作。在 Octave 中虽然也可以使用 for 循环来实现，但是 Octave 本身有简单的向量操作方式。对于加减以及数乘运算，相应的就是向量各分量上的运算。而矩阵的乘法必须要求矩阵的形状相符合。另外，在运算符号前加一个  $.$  代表对应元素的运算。



```

>> x = ones(2)
x =
    1    1
    1    1
>> y = [1,2;3,4]
y =
    1    2
    3    4
>> 2*x
ans =
    2    2
    2    2
>>

>> x + y
ans =
    2    3
    4    5
>> x * y
ans =
    4    6
    4    6
>> x .* y
ans =
    1    2
    3    4
>>

```

而除法  $A/B$  则代表着求解方程  $XB = A$ ，这自然要求  $B$  可逆。矩阵转置则只需在矩阵后加上一个单引号。

一些常用的操作矩阵的方法如下

---

inv	求矩阵逆矩阵
det	求矩阵特征值
trace	求矩阵的迹
eig	求矩阵的特征向量和特征值
rank	求矩阵的秩
null	求矩阵零空间的基
rref	对一个增广矩阵进行 Gauss 消去
lu	求矩阵的 LU 分解
qr	求矩阵的 QR 分解
svd	求矩阵的 SVD 分解
pinv	求矩阵的广义逆

---

表 3: 矩阵的操作

## 4 作图

Octave 的作图是通过调用开源软件 GNUPLOT 来实现的。

### 4.1 基本绘图：plot

最基本的 2 维画图命令是 `plot(x,y)`，其中 `x,y` 分别为横轴和纵轴数据。我们可以加上一些参数使线条样式更多样化

颜色	点的类型
w 白色	. 点
c 青色	o 圆圈
r 红色	x x 形
g 绿色	* 星号
b 蓝色	+ + 号

表 4: 线条样式参数

再之后，`title`，`xlabel`，`ylabel` 分别为图片添加标题和 `xy` 轴名称

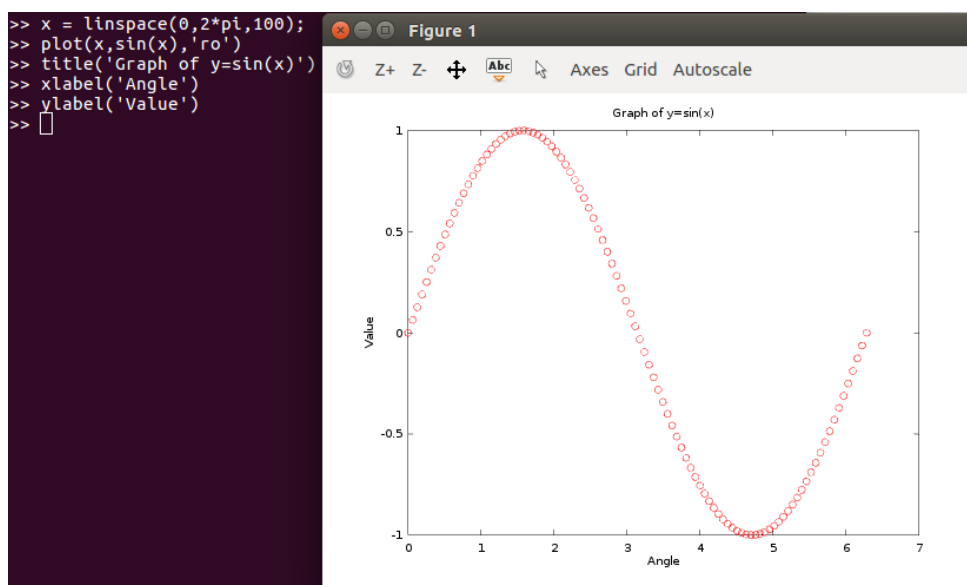


图 7: plot 基本操作

### 4.2 持续操作：hold on

如果你想在当前图形下继续进行操作，你可以通过使用 `hold on` 命令来实现。使用该命令后，我们可以实现在同一幅图上呈现由多个 `plot` 命令绘制的线条。

而多幅图片可以通过 `figure` 命令来控制，在命令行中输入 `figure` 那么下一个 `plot` 命令将会在新创建的窗口中绘制（如果既不 `hold on` 也不 `figure`，那么新的 `plot` 会取代之前的图形）。

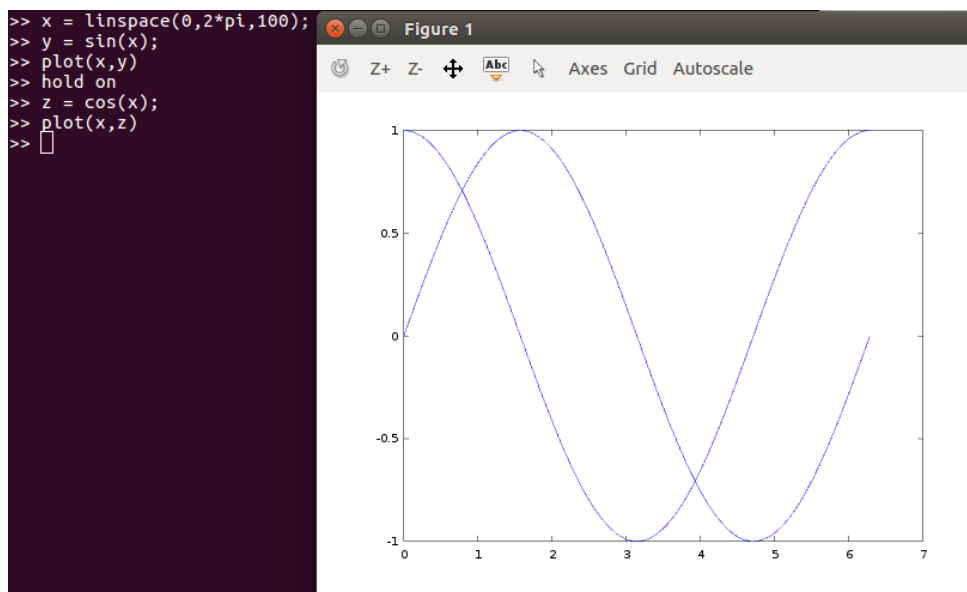


图 8: hold on 指令

### 4.3 创建子图: subplot

要在同一个窗口中创建多幅子图要通过 subplot 命令实现。该命令能够将窗口分离成一系列的子窗口, 其基本语法为

$$\text{subplot}(\text{row}, \text{columns}, \text{num})$$

其中 num 参数指定当前绘图在子窗口中的序号, 序号按照从上到下, 从左到右的次序递增。接下来用一个例子来说明 subplot 的应用

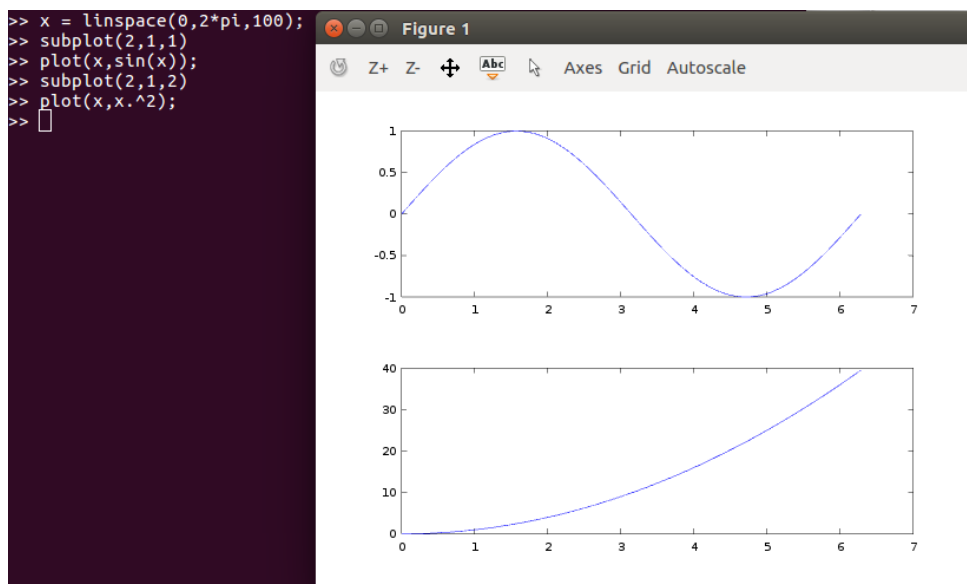


图 9: subplot 指令

#### 4.4 3D 绘图: plot3

Octave 同样可以进行 3D 数据的可视化, 最简单的 3D 绘图实现是由 plot3 命令完成的, 与 plot 命令类似, 该命令通过接受输入的 x,y 和 z 轴的数据, 绘制出点或者线。接下来的例子画出了一个螺旋线, 使用参数式的函数

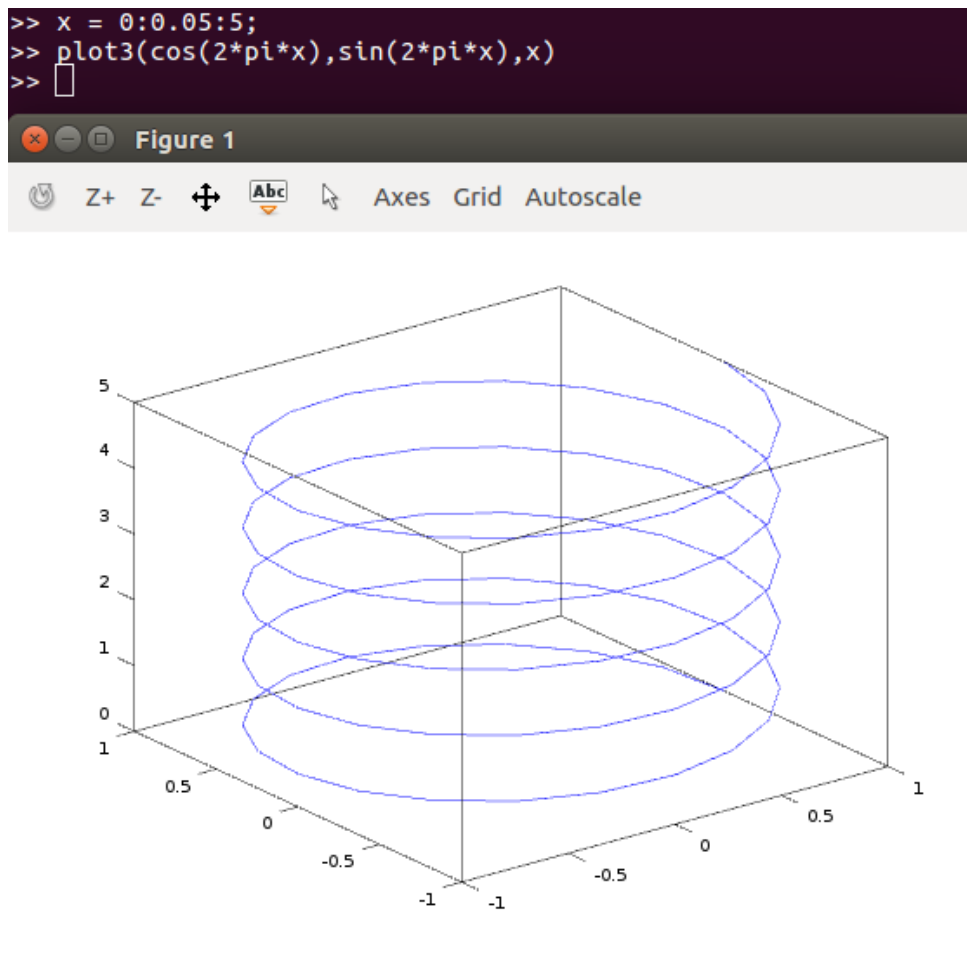


图 10: 3D 绘图

#### 4.5 绘制曲面: meshgrid, surf

除了三维的线与点之外, 我们还需要绘制曲面。例如一个函数

$$z = f(x, y) = (x-3)^2 - (y-2)^2, x \in [2, 4], y \in [1, 3]$$

为了绘制曲面, 我们首先要生成 xy 平面上的网格, 这通过 meshgrid 命令实现. 之后求得对应网格上的函数值, 利用 surf 指令生成曲面

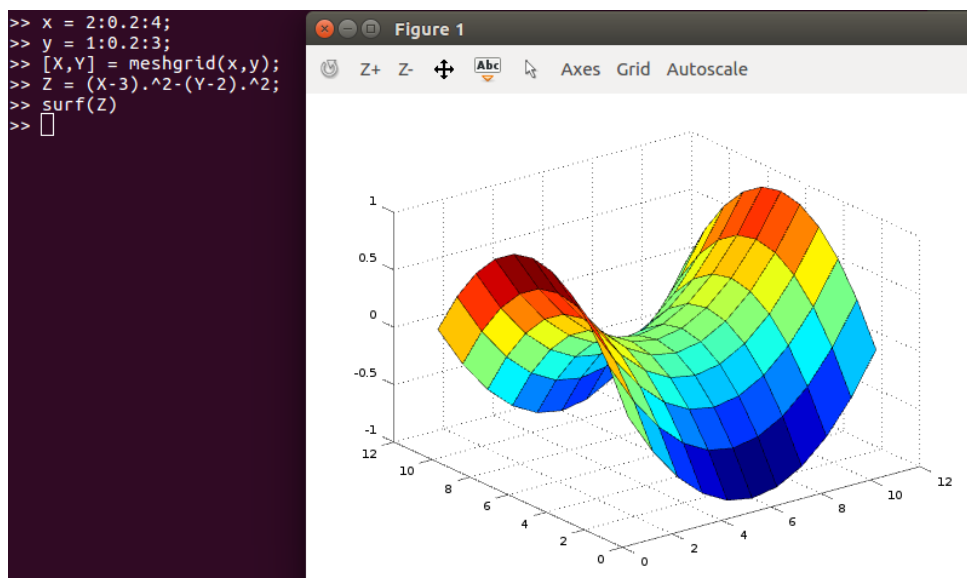


图 11: 曲面的绘制

## 5 脚本文件与函数

如果你想实现的目标比较复杂,那么直接通过终端输入及修改指令会很不方便。你可以将这一系列的指令存入一个 Octave 脚本之中。这种包含 Octave 指令的文本文件是 Octave 程序的基本形式。当你在 Octave 中执行这样的脚本的时候,其效果与将这些指令一行行输入 Octave 中是一样的。而且当你的一系列要输入 Octave 的指令不是很拿的准的时候,在一个脚本中修改这些指令会比在 Octave 终端中重新调出及修改指令要简单方便许多。Octave 的脚本是普通的文本文件,它们需要一个.m 的后缀。因此,它们被称为 M 文件。除去后缀的文件名部分是你执行该命令时需要向 Octave 终端输入的部分。

你可以在任何的文本编辑器(如,emacs,vim,notepad)中创建并编辑一个脚本文件。在 Octave 中输入命令 `edit` 将在新窗口中调出文本编辑器 emacs。要运行某个脚本时,例如 `run.m`,只需在相关路径下输入脚本名 `run` 即可。另外,% 后的一行语句代表注释,将被 Octave 解释器忽略。

Octave 假定一个脚本文件的头几行注释是该脚本的描述,这几行注释也是你使用 `help` 命令时获得的信息,因此在你每个脚本的头几行写上有关该脚本的帮助信息是一个很好的习惯。

```
1 %这是注释,给出相关信息
2 x = linspace(0,2*pi,100);
3 %下面是创建子图
4 subplot(2,1,1)
5 plot(x,sin(x));
6 subplot(2,1,2)
7 plot(x,x.^2);
8
```

图 12: 脚本文件

### 5.1 控制语句

复杂的脚本文件通常需要用各种各样的控制语句。与常用的语言类似,Octave 也提供了标准的控制语句。需要再次提醒的是,尽量用矩阵形式的计算代替循环语句,这有利于提高运行效率。

#### 5.1.1 if...else 语句

如果你想在程序中有条件的执行一些操作,就需要 `if` 这样的条件执行语句。Octave 中 `if` 语句的一般用法是

```
if expression
```

```
    statements
elseif expression
    statements
```

### 5.1.2 for 语句

for 循环是编程语言中另一个常用的结构, 它将一定次数的重复执行一段代码。虽然我们多次强调在 Octave 中你应该多使用矩阵的计算而不是 for 循环, 因为通常 for 循环会慢很多。然而有的时候 for 循环是不可避免的。该语句的语法是

```
for variable=vector
    statements
end
```

### 5.1.3 while 语句

如果你不知道需要执行多少次循环, 而是知道当某条件满足时结束循环。Octave 中的 while 语句能实现该功能

```
while expression
    statements
end
```

## 5.2 函数

通过脚本文件, 再加上内建函数的帮助, 我们能实现一些目标。但是当遇到要多次重复用到某一段代码的功能时, 就需要用户自定义函数。自定义函数能够让你在命令行、其他函数中和脚本中调用, 使你的代码更加简介清晰。

在 Octave 函数中参数是通过值传递的而不是通过 reference 传递并能返回多个返回值。Octave 函数如同脚本一样, 是写入一个纯文本文件中的, 但函数文件的第一行遵循如下的格式

```
function [output1,output2,...]=name(input1,input2,...)
```

即说明了函数的调用方式。

每个函数都写在了不同的 M 文件中, 而且该 M 文件名必须与函数名一致。例如名为 myfunc() 的函数必需被定义在名为 myfunc.m 的 M 文件中。每个函数接受一定的参数输入并输出一定的数值。

当你需要重复性的执行一定表达式, 这表明你需要将这的操作封装为一个函数。封装为函数之后 Octave 将更加易用, 增加了代码的可读性并可以供他人使用。另外, 定义函数时, 也应写好其注释内容, 这将作为 help 的信息, 便于你之后理解以及他人的使用。

下面是一个计算角度制下正弦值的函数



```
1 function s=sind(x)
2 % sind(x) calculates sine(x) in degrees
3 % Input: x, the degree, real number
4 % Output: sine(x*pi/180)
5 s = sin(x*pi/180);
6 endfunction
7
8
9
10
11
12
13
14
15
```

图 13: 函数示例

- Line1 该行告诉 Octave 本文件定义的是一个函数而不是脚本。该行表明定义了一个名为 `sind` 的函数, 而且该函数接受一个输入参数并返回一个返回值。
- Line2,3,4 这几行为注释行, 说明了该函数的功能, 输入参数以及输出值。和脚本中的一样, 函数中这部分注释将被 Octave 帮助系统识别为有关该函数的说明, 并在用户使用 `help sind` 时被返回。
- Line5 该行是实现函数的真正功能。它将获取 `x` 的值并将计算结果赋给返回值 `s`。
- Line6 Octave 中函数的结尾需要以 `endfunction` 结束。通常 Octave 不需要 `return` 语句 (当然你可以使用该语句从函数体中间跳出) 来终止函数体。

## 6 与 MATLAB 的区别

Octave 成为一个系统开发项目后, 一直试图兼容 MATLAB, 甚至将这作为一个目标。但是这种兼容不是无原则的模仿。Octave 的开发者大多也都是 MATLAB 的高手, 他们实现一个和 MATLAB 兼容功能的时候, 都会充分考虑是否值得实现, 以及考虑怎么提高这个功能的性能。特别的, Octave 的开发者本着实用的奥卡姆剃刀原则, 瞄准可能使用到 Octave 的研究人员群体, 大胆的放弃实现一些 MATLAB 的功能, 确保 Octave 够用就好。再有, Octave 主要运行在类 UNIX 系统下, 兼顾了类 UNIX 用户的习惯, 在某些功能上给出了多种选择 (比如注释)。以下是 Octave 和 MATLAB 的主要区别。有些是 Octave 的个性和特性, 高于同功能的 MATLAB 实现的性能。由于 MATLAB 和 Octave 都处于开发之中, 以下所列的各种不同也许很快就不存在了, 请注意。

- 1. 递加、递减运算符 Octave 支持递加、递减运算符, 如 `x++`, `y--`, 这与 C 风格相同。
- 2. 程序块的结束 Octave 既支持和 MATLAB 兼容的 `end` 关键字, 也支持独有的关键字“`endblock`”, 如 `endfunction`、`endif` 等。Octave 之所以采用这种形式, 是觉得 MATLAB 因为使用同一个 `end` 而导致函数结构混乱, 分不清函数的层次。

- 3. 注释 Octave 既支持和 MATLAB 兼容的%，也支持 linux 系统通用的 # 作为注释的引导符。
- 4. 字符串 Octave 使用 “”（双引号）和 ‘’（单引号）都是可以的。

```
>> x = 0;
>> if x==0
x++;
endif
>> x
x = 1
>> str1 = 'abc'
str1 = abc
>> str2 = "abc"
str2 = abc
>> str1 == str2
ans =

     1     1     1
>> 
```

- 5. 函数嵌套 Octave 的作者认为 MATLAB 的函数嵌套弊大于利，于是在 Octave 中基本上可以认为不支持嵌套。
- 6. 核心函数 MATLAB 的大部分常用的核心函数（即那些不包含在 toolbox 中的函数）都已经在 Octave 中实现了。没有实现的函数，大多是专用于某项功能的函数且这项功能 Octave 并没有实现（GUI、ActiveX 等）。
- 
- 7. 绘图 Octave 的画图后台是强大的 Gnuplot，有人认为绝对不会弱于 Matlab，而且输出格式要远多于 Matlab，公式显示也要强大很多。但有人说 Octave 绘图速度比 Matlab 慢。
- 8. GUI 目前，Octave 没有实现与 MATLAB GUI 的兼容。因为 Octave 的用户大多并不需要一个 GUI，或者即使有 GUI，他们也不用。所以他们并不怎么对开发一个与 MATLAB 的 GUI 类似的东西感兴趣。
- 9. Simulink Octave 没有支持 Simulink。主要原因是开发人员认为 Simulink 模型是落后于研究的，并且缺少灵活性，在实际的研究环境中基本不会用到。这也表明了 Octave 开发者和使用者的群体：主要是面向前沿研究的，需要个性化定制数值计算的工具。
- 10. 工具箱 Octave 是一个社区项目，工具箱都是由爱好者资源提供的。所以在和 MATLAB 工具箱的兼容性上是不怎么考虑的。

- 11. 函数的书写 Octave 允许在命令行中写函数。这是 MATLAB 不支持的。
- 12. 费用方面 Octave 是完全免费的（并且是开源的），而 Matlab 是商业软件，价格很昂贵（当然，这在当前国情下不是问题）。商业版的优势是有非常完善的服务，即使没有购买正版，也可以在 MathWorks 官方网站上获得很多非常有价值的资源。
- 13. 占用空间 Octave 比较小，安装程序只有几十兆；而 Matlab 非常庞大，最新版的安装程序大约 8G，即使只安装最基本的系统，至少也要几百兆以上。Matlab 之所以那么庞大，是因为有大量的面向各种应用领域的工具箱，Octave 无法相比的。

## 7 总结

这次对 Octave 的学习，由于其与 MATLAB 的相似性，一方面可以试做是对 MATLAB 的基本知识的回顾与总结，另一方面则是对它们的比较。通过学习，我们看到，Octave 的大部分操作与 MATLAB 是兼容的，可以说 MATLAB 上的指令在 Octave 上基本都得到了很好的“克隆”。在尽可能兼容 MATLAB 的同时，Octave 的开发者们也注入了他们自己的一些心得，这在某些程度上方便了使用者。但要看到的是，Octave 作为一个开源软件，它的安装程序很小，比起有着多年商业使用经验，已经非常成熟的 MATLAB 来说，还是单薄了些。想要更加全面的功能的话，MATLAB 无疑还是首选，而如果使用需求并不大，Octave 足以替代 MATLAB。