

第一讲 Scilab概述

■ 1. 科学计算软件与SCILAB

SCILAB一词源于英文“Scientific Laboratory”（科学实验室）词头的合并。SCILAB是以法国国立信息与自动化研究院（INRIA）和法国桥梁学院（ENPC）的科学家为主共同开发的“开放源码”式科学计算软件。SCILAB已经得到各国研究工作者，大学教授以及教育界和工业界人士的普遍关注，并被免费地引进到他们的研究、教学以及产品研发中，目前正以每月10,000份的速度向全世界发布。

■ 2. SCILAB在世界的发展状况

科学计算软件曾在工程界的研究和发展活动中掀起了一场革命。如今，这些软件已经被广泛应用到工业工程实验室中，成为各科研领域众多学者、教师和学生必备工具。法国国立信息与自动化研究院（INRIA）无疑是这场革命的先驱者。

可以说SCILAB 软件及其工具箱已经取得了很大的成功：每月都有来自全球的近万人次登录SCILAB 网站，并下载该软件。如此国际化的成功一方面因为该软件本身的语法和基本功能完全可以和行业参照软件MATLAB相媲美，另一方面因为它是完全免费的。

■ 3. SCILAB在中国的推广

中国科学院自动化所和法国国立信息与自动化研究院（INRIA）联合创办的中法信息、自动化与应用数学实验室吸引了中法许多大学和科研单位的科学家，这些科学家紧密合作，从事了许多基础性、公益性的前沿科学研究。

经过2001,2002两届SCILAB竞赛，更多的中国大学生们了解和接受了SCILAB，竞赛作品的水平逐年提高。现在，SCILAB竞赛每年举办一届，影响力越来越大，让我们盼望着更多高水平作品和更多SCILAB使用者的出现。也希望随着SCILAB在中国的传播，自由科学软件能够逐渐为广大教育科研人员所接受和使用。

下载地址：<http://www.scilab.org/download/>

第二讲 Scilab操作入门

- Scilab的命令提示符是-->，在该提示符下可以输入命令。

在运行demo时会有暂停，此时显示的提示符>>只是等待用户按Enter键，不能输入命令。在菜单Control->stop可以暂时停止demo的执行，之后命令提示符变为-1->，此时可以输入命令。使用resume命令或者选择菜单Control->resume 可以恢复demo的执行。使用abort命令或选择菜单Control->abort可以放弃demo的执行，命令提示符重新变为-->.

■ 数值运算

基本运算

在指令界面的 '-->' 之后，即可输入运算字串。例子：

--> 1+1 //按下[Enter]

Ans =

2.

--> log(10) //按下[Enter]

Ans =

2.3025851

■ 变量与常量

- 在scilab中变量的宣告是很自由的，几乎大部分的字串皆可以成为变量。例如`x=3;a2=14;.....`但是变量开头不可以是数字。同时也不可以将Scilab内定常数更改它的数值。
- Scilab的内定常数多半以'%'开头，例如圆周率，它以'%pi'来表示。自然常数e：%e。虚数i：%i。如对这些常数重新宣告会出现错误讯息。



变量的类型

scilab的数据类型分为三大类：标量式、矩阵式和特殊数据类型。

标量式和矩阵式都包含如下几种基本类型：数值类型、布尔类型、多项式类型和字符串类型。特殊数据类型包括表(list)和函数。





数值类型

如matlab一样，scilab的变量可以不经定义直接赋值。不同的是，引用系统预定的变量需加上%，如得到常量pi的值，需使用%pi，



类似的如虚数单位%i，布尔值%t,%f等等。



布尔类型

可通过关系运算符产生布尔类型的变量。要得到布尔类型的字面常量，可以用%t,%f,%T,%F。

字符串类型

用单引号或双引号表示字符串。在scilab中，字符串被当作一种基本类型而不是字符数组来看待。因此，在用字符串构成矩阵的时候没有对字符串长度的限制。例如，

```
-->s=['this' 'is' ; 'a' 'string']
```

'this'与'a'长度不需要相同。

■ 多项式类型

- 生成多项式用poly函数。
`[p] = poly(a, "x", ["flag"])`
a是矩阵或向量(包括标量)
x 是符号变量
flag是可选参数

若a是矩阵，则不需flag参数。返回矩阵 $x \cdot E - a$ 的特征多项式，E为与

a同维的单位矩阵

若a是向量(含标量情形)，返回结果与flag有关。若flag为roots(默认值)，则返回自变量是x，以向量a的个元素为根的多项式。

例如，
`--> p = poly(0, "x", "roots")`
然后可利用返回结果运算得到新的多项式：
`--> q = 1 + 2*p + p^2`
也可直接指定系数生成多项式
`--> p = poly([1 2 1], "x", "coeff")`

■ 矩阵生成

直接输入

可以使用同Matlab一样的语法来输入矩阵，如
 $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

行分隔符用分号;或者换行符，列分隔符用逗号，或空格，
首尾用[,]括起来。数据必须是同一基本类型，但布尔类型
可作为数值类型(0和1)而与数值类型混用,数值类型可作为
多项式与多项式混用。

使用start:increment:end双目运算符生成向量

例如: --> $x = 0:0.1:10;$

■ 使用函数

可用linspace得到均匀分布的向量。

$[v] = \text{linspace}(x1, x2, [n])$

v : 从 $x1$ 到 $x2$ 均匀分布的 n 个数, n 可省, 缺省值为100。

■ eye 生成单位矩阵

$x = \text{eye}(m, n)$ 生成 $m \times n$ 维的对角线为1, 其它元素为0的矩阵

$x = \text{eye}(A)$ 生成与矩阵 A 同维的单位矩阵, 若 A 是标量, 结果为1

$x = \text{eye}()$ 生成维数不定的单位矩阵, 当它与其它维数确定的矩阵相加时自动转为相应维数的矩阵

■ 基本运算符

- 加减乘除： $+$ $-$ $*$ $/$
- 自然数的次方： e^x 在scilab上，要写成： $\exp(x)$
- 绝对值： $\text{abs}()$ ，例如： $\text{abs}(-3)$ Ans= 3.
- 矩阵转数值：在算符前加上“.”
例如： $.*$ ， $./$ ， $\text{exp}()$ 等等，例如： $3*t.*\exp(\sin(t))$
- 次方算符： $^$ (例如2的3次方23： 2^3)
- 开根号： $\text{sqrt}()$

■ 运算符：log系列

- $\log()$ ：这是以自然数 $e=2.7182818$ 为底的对数。这种对数在数学上是以 \ln 来表示之。
- $\log_2()$ ：以2为底的对数。
- $\log_{10}()$ ：以10为底的对数，也是我们最熟悉的一种对数之一。

■ 其它少见的log指令：

$\log_{1p}(x)$ ：等同于 $\log(1+x)$

$\logspace()$ ：以对数方式从 a, b 取 n 等分。

例如： $\logspace(0, 2, 300)$ ：0到2之间取300等份。

■ 三角函数

- 三种： $\sin()$, $\cos()$, $\tan()$. 至于 \cot , \sec , \csc 这三个函数 Scilab 并没有设计，不过使用： $1/\sin()$ ， $1/\cos()$ ， $1/\tan()$ 这样的格式即可解决。
- 三个反函数： $\text{asin}()$, $\text{acos}()$, $\text{atan}()$.

第三节 基本程序语句与算法实现

熟悉Scilab编程环境

Scilab基本常识 —— 四则运算

例 1 计算： $3+6-7\times 2+12\div 3$ 。

解：运行“Scilab”，在命令行输入算式：

——> $3+6-7 * 2+12/3$

// 输入算式时，“ \times ”号用“ $*$ ”号替代，
“ \div ”用“ $/$ ”符号替代，按 Enter 键，即
生成计算结果。

+、-、*、/

= (为赋值号)

==(为等于)

Modulo(a,b) (为求a除以b的余数)

第三节 基本程序语句与算法实现

熟悉Scilab编程环境

Scilab基本常识 —— 数值、变量、表达式

1.2 数值、变量和表达式

数值、变量和表达式的含义与数学上的含义基本相同。

Scilab 的数值采用十进制，精确度的默认值为小数点后 7 位。

给变量命名，第一个字母必须是英文字母。

在命令行输入：

```
--> a = 2/3
```

```
a =
```

```
.6666667
```

■ 双曲线函数

- 这是一个跟三角函数相似但内容完全不同的函数。此类函数与e有关。
- 三个： $\sinh()$, $\cosh()$, $\tanh()$

■ 三个反函数： $\operatorname{asinh}()$, $\operatorname{acosh}()$, $\operatorname{atanh}()$

第三节 基本程序语句与算法实现

熟悉Scilab编程环境

Scilab基本常识 —— 数值、变量、表达式

1.3 数组及其在计算中的应用

如果算式只是求和，Scilab 可把各个加数输入到中括号内，两数之间用空格或逗号分开，然后给这组数起个名称。例如，求 $2+3-5+8$ ，可在命令行输入：

```
-->a=[2 3 -5 8]; // 如果在命令行输入结束后，输入分号，则命令计算机储存但不显示输入结果。
```

若需要引用数组中具体的某一个分量时，可使用 $a(i)$ 来表示，如上例， $a(1)$ 的值为2

第三节 基本程序语句与算法实现

熟悉Scilab编程环境

Scilab基本常识 —— 数值、变量、表达式

上面输入数组是逐个元素输入，有些特殊的数组可用冒号“:”生成，例如输入：

```
-->c=1:0.5:4           //按 Enter，生成行向量，
```

```
c =
```

```
! 1.    1.5    2.    2.5    3.    3.5    4. !
```

这种生成方式表明，在 1 和 4 之间按间距 0.5 生成一个行向量，中间的数字 0.5 称作这个向量的步长。

例 3 (1) 令 $x = -3, -2.5, -2, \dots, 2, 2.5, 3$ ，求 x^2 值。

第三节 基本程序语句与算法实现

熟悉Scilab编程环境

Scilab基本常识 —— 数值、变量、表达式

清除Scilab临时保存的所有变量

如果在命令行输入“clear”，按“Enter”键，则清除 Scilab 临时保存的所有变量。



第三节 基本程序语句与算法实现

1、基本程序语句

任何一门程序语言都应包含的基本语句结构：

赋值语句、输入语句、输出语句、条件语句、循环语句

接下来我们学习**数学应用软件Scilab**提供的**编程语句**，来实现曾经学习过的算法。

数学应用软件Scilab

第三节 基本程序语句与算法实现

1) 赋值语句

一般格式：

变量名 = 表达式

几点说明：

A、赋值号左右不能对换

B、赋值号右边的表达式中每一个变量都必须事先赋给确定的值，同一个表达式中不能出现两个或多个“=”

第三节 基本程序语句与算法实现

1) 赋值语句

一般格式：

变量名 = 表达式

例如：在Scilab中给三个变量a、b、c分别赋值，并计算出三个数的平均数

```
a=3  
b=4  
c=5  
aver=(a+b+c)/3
```

```
a=3;b=4;c=5;  
aver=(a+b+c)/3
```

```
a=3;b=4;c=5;  
aver=(a+b+c)/3;
```

语句说明：输入“;”则命令计算机存储但不显示输入结果

第三节 基本程序语句与算法实现

2) 输入语句

键盘输入语句：input()

例如：在Scilab中计算任意学生的语文、数学、外语三门课的平均成绩，要求语文、数学、外语成绩能从键盘输入

```
a=input("chinese")
```

```
b=input("math")
```

```
c=input("english")
```

```
aver=(a+b+c)/3
```

第三节 基本程序语句与算法实现

3) 输出语句

输出语句：有print、write、disp等

例如：

```
a=input("chinese")  
b=input("math")  
c=input("english")  
print(%io(2),a,b,c)
```

说明：%io(2)表示在屏幕上输出，以上语句表示在屏幕上依次输出变量a,b,c的值

第三节 基本程序语句与算法实现

4) 条件语句

```
if  条件表达式  
    语句序列1 ;  
else  
    语句序列2 ;  
end
```

如果将条件表达式及语句序列**写在同一行**，则需要条件表达式后加上关键字“**then**”或“**,**”以示分隔；

```
if 条件表达式 then 语句序列 ; end  
或者  
if 条件表达式 , 语句序列; end
```

第三节 基本程序语句与算法实现

4) 条件语句

例：写出
求一元二
次方程的
根的程序

```
a=input("please input a:");
b=input("please input b:");
c=input("please input c:");
delta=b^2-4*a*c;
if delta<0
    print(%io(2),"方程无解");
else
    x1=(-b+sqrt(delta))/(2*a);
    x2=(-b-sqrt(delta))/(2*a);
    print(%io(2),x1,x2)
end
```

第三节 基本程序语句与算法实现

5) 循环语句

语句形式一：**for** 循环

基本语法形式如下：

for 循环变量 = 初值 : 步长 : 终值

循环体

end

第三节 基本程序语句与算法实现

5) 循环语句

如：对任意自然数 n ，编程求 $n!$ ！

```
n=input("please input n:")
```

```
k=n;
```

```
s=1;
```

```
for k=n:-1:1
```

```
    s=s*k;
```

```
end
```

```
print(%io(2),s)
```

第三节 基本程序语句与算法实现

5) 循环语句

练习：对任意自然数 n ，编程实现 n 以内所有自然数之和



第三节 基本程序语句与算法实现

5) 循环语句

语句形式二：

while 表达式

循环体

end

第四节 Scilab 运算

最小值

```
-->v=[ 11 23 73 25 49 92 28 23];  
-->min(v)//向量v中所有元素的最小值  
ans=  
11.
```

最大值

```
-->max(v)//向量v中所有元素的最大值  
ans=  
92
```

向量中所有元素的总合

```
-->sum(v) //向量v中所有元素的总合  
ans=  
324.
```

标准差

```
-->stdev([1 4 10 -5 6 9 -20]) //向量[1 4 10 -5 6 9 -  
20]的标准差  
ans=  
10.451703
```

排序

```
-->sort([1 4 10 -5 6 9 -20]) //向量[1 4 10 -5 6 9 -  
20]的元素由大排序到小  
ans=  
!10.9.6.4.1.- 5.- 20.!
```

平均值

```
-->mean([1 4 10 -5 6 9 -20]) //向量[1 4 10 -5 6 9  
-20]的平均值  
ans=  
.7142857
```

■ 向量中各元素间的运算

-->v1=[1 5 6 7]; v2=[0 2 3 5];
-->v3=v1.*v2 //向量v1 , v2中所有对应之元素相乘
v3=
!0.10.18.35.!

■ -->v4=v2./v1 //向量v2 , v1中所有对应之元素相除
v4=
!0..4.5.7142857!

■ 矩阵向量运算

■ -->size(A) //若无指定计算结果名则以ans储存其结果

■ 转置矩阵
-->A'

■ 行向量(column vector)及列向量(row vector)
-->A(:,3) //第3行的所有元素;行向量

■ 矩阵相加
-->C=A+B //矩阵A加矩阵B

C =

!	4.	7.	11.!
!	8.	14.	10.!
!	8.	4.	9.!

■ 多項式工具

- 由多项式系数来定义多项式

--> `p=poly([-9 6 -5 4 1], 'z', 'coeff')` //p是z的多项式系数

- $9 + 6z - 5z + 4z + z$

- 多项式方程式的根

--> `roots(p)` //多项式方程式p的所有根

ans=

!1.2007909!

!.0177876 + 1.1962571i!

!.0177876 - 1.1962571i!

! - 5.2363662!

■ 非线性代数方程式

■ 第一个例子

```
-->function y=fct1(x),y=x+exp(x)-10,endfunction //定义函数  
fct1
```

```
-->fsolve(1,fct1) //求解fct1=0的解，1为起始值
```

```
ans=
```

```
2.0705799
```

■ 第二个例子

-->function y=fct2(x),y=x^3+x*cos(x)-4*x,endfunction //定义函数fct2

-->fsolve(-5,fct2)//求解fct2=0的解，-5为起始值

ans=

-2.1281265

■ 微分方程式

■ $\frac{dy}{dt} = y^2 - y \sin(t) + \cos(t)$, $y(0) = 0$

```
-->function ydot=f(t,y),ydot=y^2-y*sin(t)+cos(t),endfunction
//定义微分方程式dy/dt=f(t,y)
-->t0=0;y0=0; //起始值条件
-->t=0:0.1:%pi; //计算范围t由0开始到%pi(=3.1415927)间隔
    为0.1
-->y=ode(y0,t0,t,f); //调用ode函数求解微分方程式
-->plot(t,y) //绘出y对t的图形
```


■ 內插与样条函数

-->x=[0 1 2 3 4]; y=[3 1 0 2 4]; //定义一组离散资料(x,y)

-->d=splin(x,y); //离散资料(x,y)对应之样条函数

-->xval=1.5; //欲內插之x坐标值xval

-->yval=interp(xval,x,y,d) //相对应xval之內插函数值yval
yval =
.171875

■ 数值积分

■ 例子

■ -->function y=f(x),y=x*sin(30*x)/sqrt(1-
((x/(2*%pi))^2)),endfunction //定义函数f-->intg(0,2*%pi,f)

//对函数f积分，由0积分到2p

ans=

- 2.5432596



- 谢谢！
- 相关学习补充资料请见课程网站网上学习！

