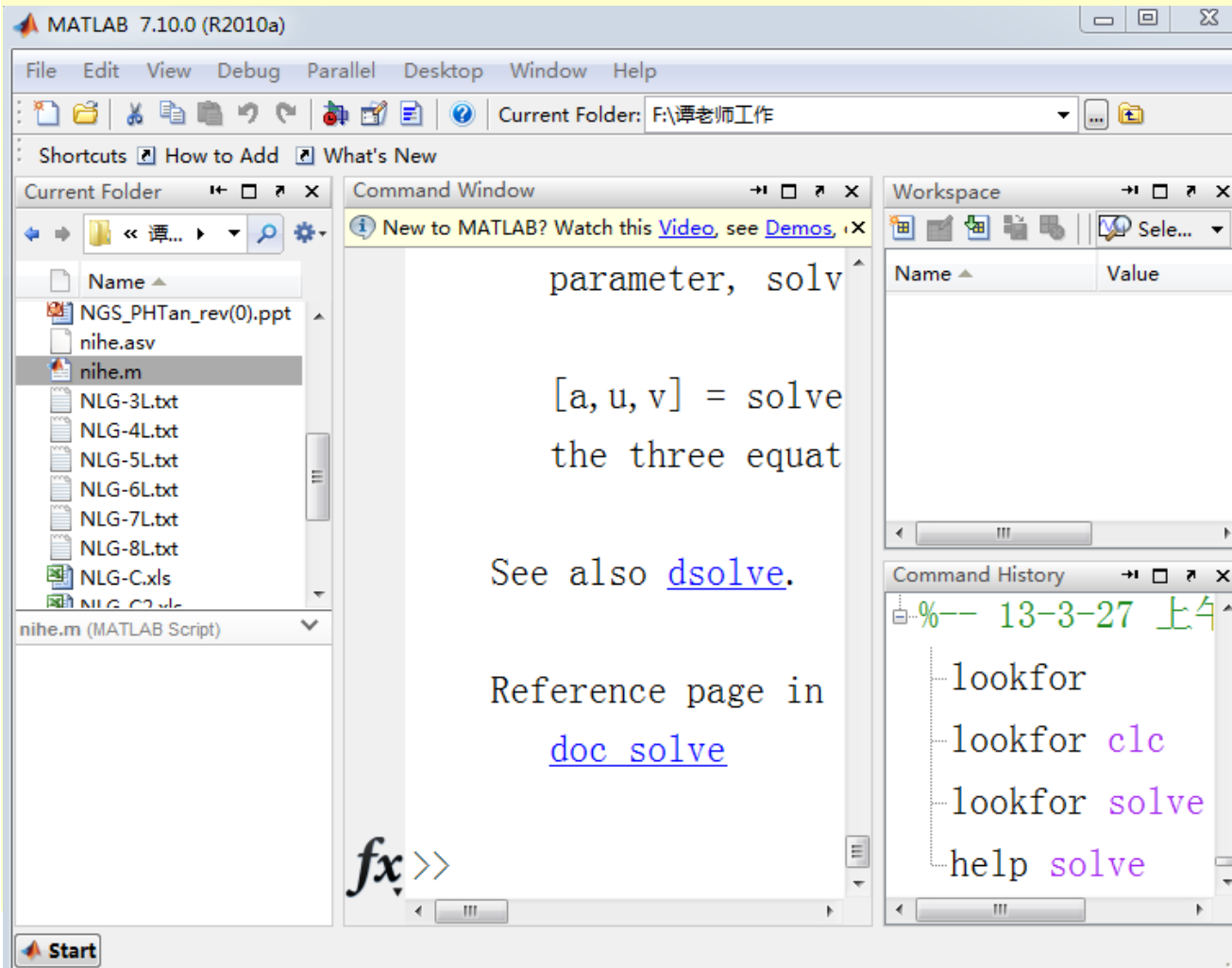


MATLAB用法简介

报告人：李晓莉
河北大学物理科学与技术学院
xiaolixiaoli2012@semi.ac.cn

1.1 MATLAB的操作界面

MATLAB (Matrix Laboratory) 是一个功能强大的优秀数学软件；
也称科学技术计算语言 (The Language of Technical Computing) 。



操作界面
四个窗口

Command
Window

Current
Folder

Command
History

Workspace

1.2 指令窗口的功能

运算提示符:	>>
计算器功能:	>>2*(300-2*i)^2+1/sqrt(5+2*i)
变量赋值:	>>A=52.738, a=3.2477e-2 (a=3.2477x10 ⁻²)
对赋值变量运算:	>>B=A+a
定义符号变量 :	>>syms a b c x
进行符号运算:	>>f=a*x^2+b*x+c >>solve(f)
执行指令:	>>clc, clear, clear all, clf
执行程序:	>>open, load, save
运行帮助程序:	>>help, lookfor

变量命名规则

- 1、变量名区分大小写
- 2、变量名以英文字母开始，第一个字母后可以使用字母、数字、下划线，但不能使用空格和标点符号
- 3、变量名长度不得超过31位，超出的部分将被忽略
- 4、某些常量也可以作为变量使用，如i在MATLAB中表示虚数单位，但也可以作为变量使用

专用变量

Pi	圆周率
eps	浮点数的相对误差 2.2204e-016
Inf	大于 10^{308} 的数据 如输入 10^{1024} 或 5/0
NaN	不定值如 0/0
Realmin	最小正实数 如2.2251e-308
Realmax	最大正实数 如1.7977e+308

例：以x表示从0到5间隔为0.5的所有数，计算 $\sin x$, $\sin 4x$, 并画出图形。

```
>> x=0:0.1:6;                                %建立矢量X
>> A=[x; 4*x];                                %建立两行的矩阵A
>> B=sin(A);                                  %计算A的正弦函数值
>> plot(x, B(1, :), x, B(2, :)) %画两条正弦曲线
```

plot用法

plot(x1,y1,x2,y2,...) 画多条曲线

plot(x,y,'color style marker') 指定颜色,线型,标志

颜色: 青 c, 洋红 m, 黄 y, 红 r, 绿 g, 蓝 b, 白 w, 黑 k

线型: 实线 -, 短划线 --, 点线 ., 点划线 -.

标志: 点 ., 加号 +, 圆圈 o, 星号 *, 叉号 x,

四方形 s, 钻石形 d, 五角形 p, 六角形 h,

三角形 (向上 v, 向下 ^, 向左 <, 向右 >)

>> plot(x,B(1,:), 'r:+') |

是画一条红色的点状曲线, 数据点用+号标志

1.3 数据存储与10种显示方式

数据运算与存储的格式都是16位十进制数，范围在 10^{-322} - 10^{308} 。

short (默认)	3位整数4位小数的定点数，超出范围用short e显示
Long	2位整数14位小数的定点数，超出范围则用long e显示
short e	5个数字的浮点数，其中1位整数4位小数
long e	16个数字的浮点数，其中1位整数14位小数
short g	显示5个数字，自动选择定点数或浮点数
long g	显示15个数字，自动选择定点数或浮点数
hex	16位十六进制数
Bank	银行用两位小数的定点数
+	显示数字的符号是正、负或零
rational	分数近似

例1 数字234.7896和45.6985678945128 在不同格式下的显示结果

	234.7896	45.6985678945128
short	234.7896	45.6986
long	2.3478960000000000e+002	45.69856789451280
short e	2.3479e+002	4.5699e+001
long e	2.3478960000000000e+002	4.569856789451280e+001
short g	234.79	45.699
long g	234.7896	45.6985678945128

编程时 `format long;`

例2 两种显示格式short和long的对比 及计算误差产生

程序如下

```
format long;
```

```
A=0;
```

```
for k=1:300, A=A+1/3; end
```

```
A
```

```
for k=1:300, A=A-1/3; end
```

```
A
```

```
B=sqrt(A)
```

运行结果是

```
A = 99.999999999999966
```

```
A = -6.994405055138486e-015
```

```
B = 0 +8.363255977870393e-008i
```

谢谢！

2.1 数据变量的类型与运算

- 标量scalar: 单个数据
- 矢量vector: 一组或一系列数据, $A(1)$
- 矩阵matrix: $m \times n$ 的二维数据, $B(m, n)$
- 列阵array: k 维数据, $M(1, m, n, \dots)$
- 数据网格meshgrid: 也是列阵, 画图专用
- 基元列阵cell array: 多种格式的数据列阵
- 结构列阵structure array: 有域名的多格式数据列阵, 用来设置解方程条件

2.2 矩阵的输入与标识

```
>> A=[2,4,5; 9, 7,1]    %同行的数据用逗号或空格隔开  
A = 2     4     5       %行与行之间的数据用分号隔开  
     9     7     1       %全部数据用方括号括起来
```

```
>>zeros(2,4)    生成2行4列的零矩阵  
>>ones(3,5)     生成3行5列的全部元素为1的矩阵  
>>rand(3,50)    生成3行50列的均匀分布的随机数矩阵
```

合并: 用[]把合并的对象括起来 [A, A]; [A; A];
删除: 令被删除对象为空 A(2,:)=[]; A(:,3)=[];

A(i, j)
A(:, j)
A(i, :)
A(2:4, j)
A(i, 2:4)
A(end, j)
A(i, end)
A(k)

2.3 矩阵的运算

不同的运算有不同的算符。差别是个圆点

- 数组运算:两个矩阵的对应元素之间的算术运算。 $A^2=A.*A$
- 矩阵运算:两个矩阵按矩阵法则的运算。 $B=A*A$
- 外部指令对矩阵内部元素的运算:
 1. 对矩阵内每个元素单独运算: 如求矩阵的各元素正弦值 $\sin(A)$
 2. 对矩阵的一列或一行元素作运算: 如求各列的和 $\text{sum}(A)$
 3. 对整个矩阵运算: 如求矩阵的逆 $\text{inv}(A)$

	转 置	加	减	乘	右 除	左 除	幂
矩阵运算算符	A' (取共轭)	+	-	*	/	\	^
数学运算算符	$A.'$ (不取共轭)	+	-	.*	./	.\	.^

2.4 编程

程序编辑器：编写，存取，路径

程序文件：
分两类

	脚本文件	函数文件
格式	无	开头function
功能	不指定输入输出变量	要指定
内存	与指令窗口共用	独立函数空间
调用	光用文件名	文件名及输入变量
子程序	无	有子函数文件

流程控制

1. forend循环结构
2. while.....end循环结构
3. ifelse.....end分支结构
4. switch....case.....otherwise.....end分支结构

2.5 作图功能演示

数据图: 二维-平面曲线,极坐标图,对数图,扇形图,直方图,.....
三维-表面图,网格图,瀑布图,假彩色图,.....
四维-切片图,复函数图,空间标量场,平面矢量场,.....
六维-空间矢量场

函数作图:普通函数,自适应函数法,复函数图,隐函数图,.....

编辑: 加注文字与标题, 改变视角与观察位置, 旋转与伸缩,
改变线型与色彩.....

动画方法:电影动画, 实时动画, 输出电影, 动画效果的指令.....

输出格式: fig,bmp,eps,jpg,tif等十几种。

二维曲线作图指令

plot	画一条或多条曲线	comet	画二维彗星曲线
bar	直方图	loglog	双对数坐标曲线
compass	原点出发的复数矢量图	pcolor	假彩色图
contour	在x - y 面上的等高线图	polar	极坐标图
errobar	误差棒图	quiver	矢量场图
ezplot	符号函数二维曲线	rose	统计频数扇块图
feather	沿 x 轴分布的复数矢量图	semilogx	x 轴对数坐标曲线
fplot	数值函数二维曲线	semilogy	y 轴对数坐标曲线
fill	平面多边形填色	stem	火柴杆图
hist	统计频数直方图	stairs	阶梯图

例1 用四张子图分别画出 $0 - 2\pi$ 范围内 $\sin(x)$, $\cos(x)$, $2*\sin(x)*\cos(x)$, $\sin(x)/\cos(x)$

- `clc;clear;`
- `x=linspace(0,2*pi,30); y=sin(x); z=cos(x);`
- `u=2*sin(x).*cos(x); v=sin(x)./cos(x);`
- `subplot(2,2,1),plot(x,y),axis([0 2*pi -1 1]),title('sin(x)')`
- `subplot(2,2,2),plot(x,z),axis([0 2*pi -1 1]),title('cos(x)')`
- `subplot(2,2,3),plot(x,u),axis([0 2*pi -1 1]),title('2sin(x)cos(x)')`
- `subplot(2,2,4),plot(x,v),axis([0 2*pi -20 20]),title('sin(x)/cos(x)')`

例2 Logistic模型

$$x_{n+1} = \mu(x_n - x_n^2) \quad (0 < \mu < 4, 0 < x < 1)$$

- `u=2.6:0.001:4;`
- `x=0.6;`
- `for j=1:150,`
- `x=u.*(x-x.^2);`
- `end`
- `for i=1:100`
- `x=u.*(x-x.^2);`
- `plot(u,x,'r.')`
- `hold on;`
- `end`

例3 两种编程方法的比较

- `clc;clear;`
- `E = linspace(0,1,40000);`
- `Ex = 0.6;r = 10^(-2);V = 1;sumG = 0;`
- `for k = 1:30000`
- `sumG = sumG + 1./(E - 1/k + 1i*r);`
- `end`
- `Gxx = imag(1./((E - Ex + 1i*r)-V^2.*sumG));`
- `plot(E, Gxx)`
- `clc;clear;`
- `Ex = 0.6;r = 10^(-2);V = 1;A = zeros(1,10000);E1 = zeros(1,10000);`
- `for h = 1:10000`
- `E = 1*(h-1)/9999;sumG = 0;`
- `for k = 1:30000`
- `sumG = sumG + 1/(E - 1/k + 1i*r);`
- `end`
- `Gxx = imag(1./((E - Ex + 1i*r)-V^2*sumG));A(1,h) = Gxx;E1(1,h) = E;`
- `end`
- `plot(E1, A)`

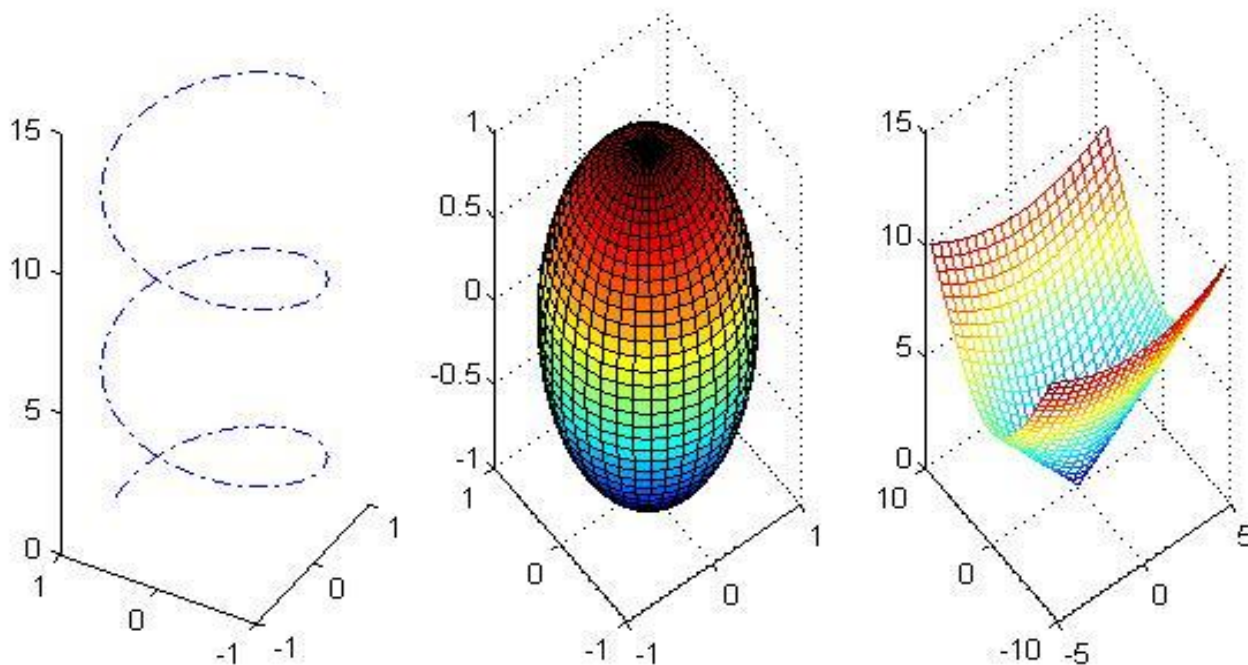
$$A = \sum_1^{30000} \frac{1}{E - \frac{1}{k} + ir}$$
$$G_{xx} = \text{Im}\left(\frac{1}{E - E_x + ir} - V^2 A\right)$$

三维曲线作图指令

空间曲线：三维参数方程（`plot3 (x, y, z)`）

旋转体：专用指令如`sphere`；`cylinder`；`ellipsoid`

空间曲面：用数据网格处理数据，用指令作图



三维空间曲面作图步骤和作图指令

步骤

- 构造数据网格:meshgrid
- 建立画图函数
- 作三维图

画空间曲面的指令:

mush 网线图

mushc 网线图再加等高线

surf 表面图

mushz 网线图再加基准平面

surfc 表面图再加光照

例 空间曲线、旋转体和空间曲面

```
subplot(1,3,1),  
t=0:0.01:15;x=sin(t);y=cos(t);z=t;  
plot3(x,y,z,'-.')  
view(-60,18)  
subplot(1,3,2),  
sphere(40)  
subplot(1,3,3)  
a = -5:0.5:5;b = -9:0.5:9;  
[X,Y] = meshgrid(a,b);  
Z = sqrt(X.^2+Y.^2);  
mesh(X,Y,Z)
```

谢谢！

3.1 数据处理

1、插值，拟合，曲线拟合工具箱

2、求方程零点

3、FFT

(DFT——离散傅里叶变换

FFT——快速傅里叶变换

FFT是DFT的快速算法

指令：fft, ifft, fft2, ifft2, fftn,
ifftn)

插 值

已知数据点，求对应的函数，常用方法

插值——函数通过数据点

拟合——要求函数偏离数据的误差最小

多项式插值法，方法有拉格朗日插值多项式，牛顿插值多项式，
埃尔米特插值多项式。

分段插值

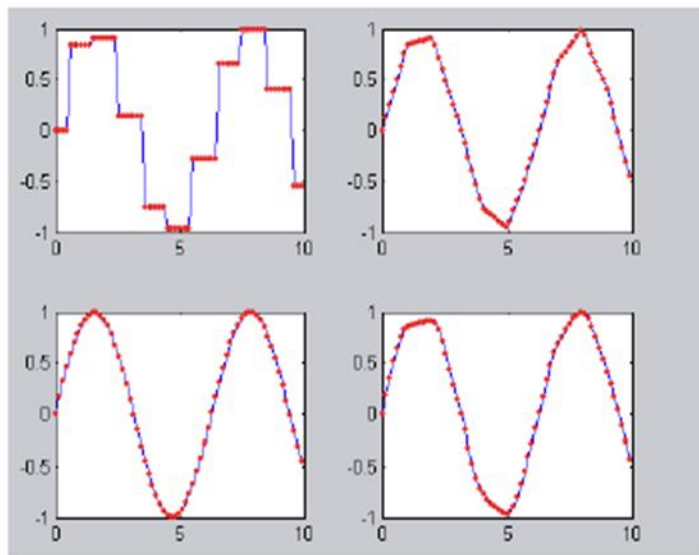
样条插值

指令		选项	
interp1	一维插值	nearest	线性最近项插值
interp1q	快速一维插值	linear	线性插值
interp2	二维插值	spline	三次样条插值
interp3	三维插值指令	cubic	立方插值
interpN	N维插值指令		

插值举例

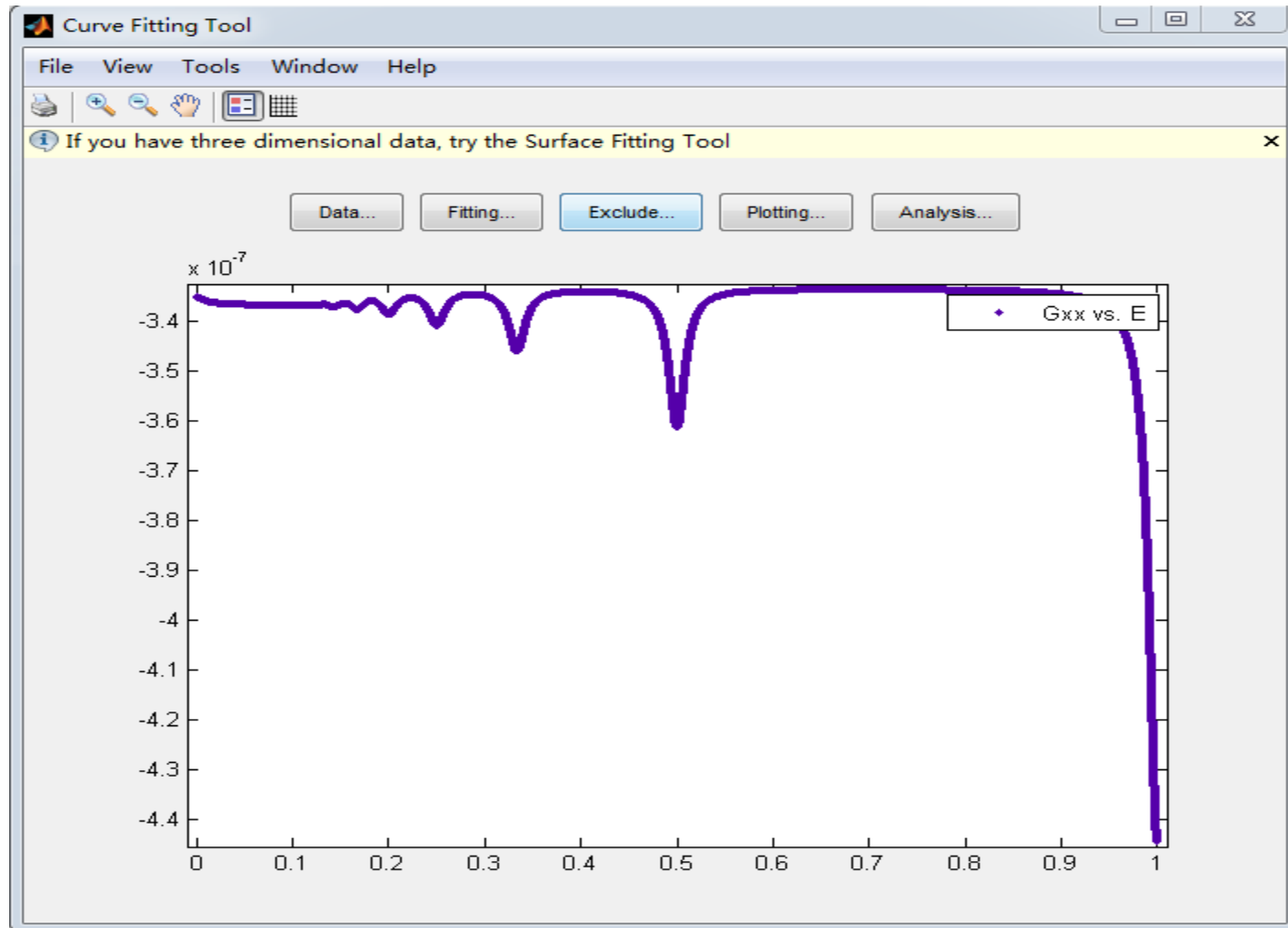
```
x=0:10; y=sin(x); %原始数据
xi=0:.15:10;      %插值点
yi1=interp1(x,y,xi,'nearest');
yi2=interp1(x,y,xi,'linear');
yi3=interp1(x,y,xi,'spline');
yi4=interp1(x,y,xi,'cubic');
subplot(2,2,1)
plot(xi,yi1,xi,yi1,'r.')
subplot(2,2,2)
plot(xi,yi2,xi,yi2,'r.')
subplot(2,2,3)
plot(xi,yi3,xi,yi3,'r.')
subplot(2,2,4)
plot(xi,yi4,xi,yi4,'r.')
```

本例中，以正弦函数的10个值作为原始数据（小圆圈表示）。再用四种方法算出41个插值点并画曲线



拟合

输入指令 **cftool**



求方程零点

solve 主要解决多项式求根

对于非线性方程数值求解，采用

fzero函数，可以用来求单变量非线性方程的根。

调用格式为：**x=fzero('fname',x0,tol,trace)**

其中**x0**为搜索的起点，一个函数可能有多个根，

但**fzero**函数只给出离**x0**最近的那个根。

fsolve函数，可以用来求非线性方程组的根。

调用格式为：**X=fsolve('fun',X0,option)**

其中**X0**是求根过程的初值，**option**为最优化工具箱的选项设定。

求方程零点举例1

myfun_qiugen.m

```
function q=myfun_qiugen(p)
```

```
x=p(1);
```

```
y=p(2);
```

```
z=p(3);
```

```
q(1)=2*cos(x)+sqrt(y)-log(z)-7;
```

```
q(2)=2^x+2*y-8*z+1;
```

```
q(3)=x+y-cosh(z);
```

myfun_qiugen_call.m

```
x=fsolve('myfun_qiugen',[0.5+0.5i,0.5+0.5i,0.5+0.5i])
```

运行结果: $x = -0.0145 + 1.6010i \quad 1.0567 - 1.7481i \quad 0.4442 - 0.3262i$

$$\begin{cases} 2\cos(x) + \sqrt{y} - \ln(z) = 7 \\ 2^x + 2y - 8z = -1 \\ x + y - \cosh(z) = 0 \end{cases}$$

求方程零点举例2 $f(x)=x-10x+2=0$ 在 $x_0=0.5$ 附近的根

funx.m

```
function fx=funx(x)
```

```
fx=x-10*x+2;
```

funx_call.m

```
z=fzero('funx',0.5)
```

运行结果: $z = 0.2222$

另一种编程方法: `funx=inline('x-10*x+2');z=fzero(funx,0.5)`

inline用来定义内联函数，可以直接定义函数的表达式，不需要使用函数的m文件（即 **function** 输出变量 = 函数名（输入变量））

谢谢！

3.2 数值积分与微分

1、符号运算工具箱（该工具箱不是基于矩阵的数值分析，而是使用字符串来进行符号分析与运算。功能主要包括：符号表达式的运算，符号表达式的复合、化简，符号矩阵的运算，符号微积分、符号函数画图，符号代数方程求解，符号微分方程求解等。）

2、数值微分的算法

3、数值积分的算法

符号运算工具箱举例

diff	符号变量微分	limit	求极限
taylor	泰勒展开	int	符号变量积分
symsum	求级数和	simple	化简表达式

int是计算积分的，主要求解不定积分

```
S1 = '6*x^3-4*x^2+b*x-5'; S2 = 'sin(a)'; S3 =  
    'sqrt(x)';  
int(S1), int(S2), int(S3)
```

数值微分的算法

泰勒展开公式

$$f(x+h) \approx f(x) + hf'(x)$$

前差公式

$$f'(x) = \frac{f(x+h) - f(x)}{h}$$

后差公式

$$f'(x) = \frac{f(x) - f(x-h)}{h}$$

中心差分(平均值)

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$$

用后差公式推

$$f''(x) = \frac{f'(x+h) - f'(x)}{h}$$

二阶导数公式

$$= \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

数值积分的算法

曲线下面积

$$I = \int_a^b f(x)dx = \lim_{n \rightarrow \infty} \sum_{k=1}^n f(\xi_k) \frac{b-a}{n}$$

矩形积分公式之一

$$L_n = h \sum_{k=0}^{n-1} f_k, \quad h = \frac{b-a}{n}$$

矩形积分公式之二

$$R_n = h \sum_{k=1}^n f_k, \quad h = \frac{b-a}{n}$$

梯形公式(两式平均)

$$T_n = h \sum_{k=1}^{n-1} f_k + \frac{h}{2}(f_0 + f_n), \quad h = \frac{b-a}{n}$$

指令（分矩阵指令和函数指令）

差分diff

梯度gradient

梯形积分trapz

累计梯形积分cumtrapz

函数积分quad, quadl

函数二重积分dblquad

函数三重积分triplequad

矩阵指令举例

diff的使用

$Y = \text{diff}(X)$ ，如果输入一个长度为 n 的一维向量，则该函数将会返回长度为 $n-1$ 的向量，向量的值是原向量相邻元素的差 $[X(2)-X(1) \ X(3)-X(2) \ \dots \ X(n)-X(n-1)]$

输入 $x = [-8 \ 6 \ 7 \ 4 \ 9 \ 5]$; $y = \text{diff}(x)$

得到 $y = \ 14 \ 1 \ -3 \ 5 \ -4$

如果输入一个 $m*n$ 的矩阵，则按行实施，结果是 $(m-1)*n$ 的矩阵，就是用下一行减去上一行

输入 $X = \text{magic}(4)$, $Y = \text{diff}(X)$

得到

$X =$

16	2	3	13
5	11	10	8
9	7	6	12
4	14	15	1

$Y =$

-11	9	7	-5
4	-4	-4	4
-5	7	9	-11

矩阵指令举例

diff的使用

```
x=0:0.1:1;
```

```
f=@(x)1+2*x-3*x.^2+4*x.^3-5*x.^4+6*x.^5;
```

```
(f=inline('1+2*x-3*x.^2+4*x.^3-5*x.^4+6*x.^5'));
```

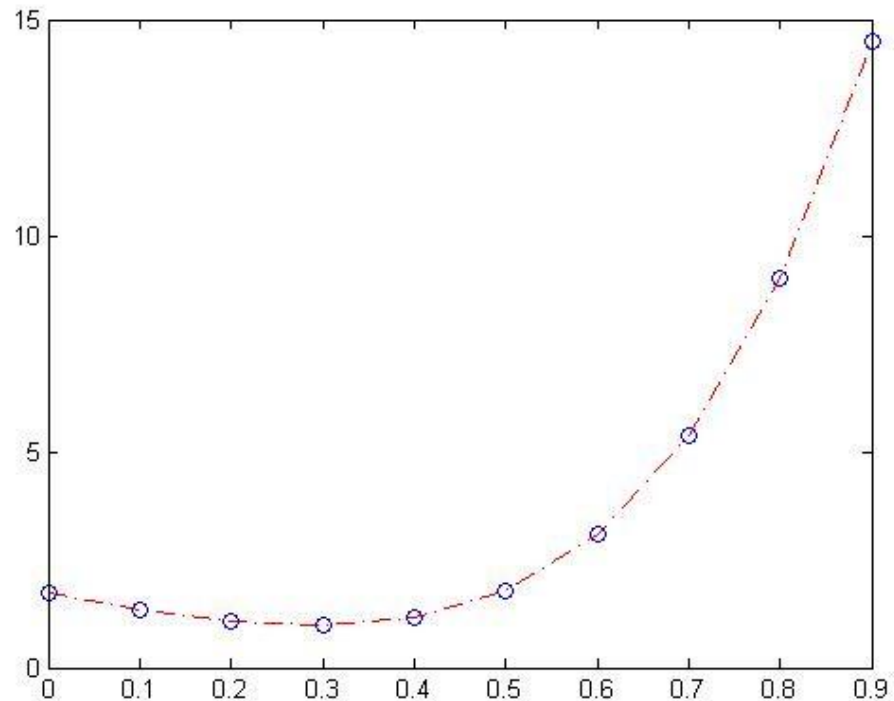
```
y=f(x);
```

```
d=diff(y)/0.1;
```

```
n=length(x);
```

```
plot(x(1:n-1),d,'r-.');
```

```
hold on;
```



函数指令举例

quad的使用

`q = quad(fun,a,b)`, 对fun函数在a、b之间求定积分, 积分精度缺省值为`1e-6`

输入 `F = @(x)1./(x.^3-2*x-5); quad(F,0,2)`

或者 `F = inline('1./(x.^3-2*x-5)'); quad(F,0,2)`

得到 `ans = -0.4605`

谢谢！

3.3 解常微分方程

龙格——库塔法解

$$\begin{cases} y' = f(x, y), & (a \leq x \leq b) \\ y(x_0) = y_0 \end{cases}$$

由中值定理, $\frac{y(x_{i+1}) - y(x_i)}{h} = y'(x_i + \theta h) \quad (0 < \theta < 1)$
得 $y(x_{i+1}) = y(x_i) + hk_{ave}$
平均斜率为 $k_{ave} = f(x_i + \theta h, y(x_i + \theta h))$ 。

由此可见, 为了提高精度, 可以多取几点的斜率值作加权平均当作平均斜率 k_{ave} , 这就是龙格库塔法基本思想.

ODE指令汇总

在求解常微分方程数值解方面，MATLAB具有丰富的函数，统称为solver

ode45	解非刚性微分方程,中等精度,使用Runge-Kutta法的四、五阶算法。
ode23	解非刚性微分方程，低精度,使用Runge-Kutta法的二、三阶算法。
ode113	解非刚性微分方程，Adams-Bashforth-Moulton PECE法。
ode23t	解中等的刚性微分方程，使用自由内插法的梯形法则。
ode15s	解刚性微分方程，使用可变阶次的数值微分(NDFs)算法。
ode23s	解刚性微分方程，低阶方法，使用修正的Rosenbrock公式。
ode23tb	解刚性微分方程，低阶方法，使用TR-BDF2方法。

指令用法说明

一般调用格式为

```
[T, Y] = ode45(odefun, tspan, y0)
[T, Y] = ode45(odefun, tspan, y0, options)
[T, Y, TE, YE, IE] = ode45(odefun, tspan, y0, options)
sol = ode45(odefun,[t0,tf],y0,...)
```

另外，要编写函数文件 `odefun.m`

还要设定求解条件

```
options = odeset('name1', value1, 'name2', value2,    )
options = odeset(oldopts, 'name1', value1,    )
options = odeset(oldopts, newopts)
odeset
```

ODE指令编程举例1

求解一阶常微分方程

例 $y' = (y + 3t)/t^2$, $y(1) = -2$, $1 < t < 4$

```
odefun=@(t,y) (y+3*t)/t^2; %定义函数
```

```
tspan=[1 4]; %求解区间
```

```
y0=-2; %初值
```

```
[t,y]=ode45(odefun,tspan,y0);plot(t,y) %作图
```

ODE指令编程举例2

求解一阶常微分方程组

例 $y_1' = y_2 y_3$, $y_2' = -y_1 y_3$, $y_3' = 0.51 y_1 y_2$,

$y_1(0) = 0$, $y_2(0) = 1$, $y_3(0) = 1$, $0 < t < 12$

编写rigid.m

```
function dot = rigid(t, y)
```

```
dot = zeros(3,1);
```

```
dot(1) = y(2)*y(3);dot(2) = -y(1)*y(3);
```

```
dot(3) = -0.51*y(1)*y(2);
```

编写ex1204.m

```
[T,Y] = ode45(@rigid,[0 12],[0 1 1]);
```

```
plot(T,Y(:,1),'-',T,Y(:,2),'-.',T,Y(:,3),'.'
```

ODE指令编程举例3

求解高阶常微分方程

关键是将高阶转为一阶， $F(y,y',y''...y^{(n-1)},t)=0$ 用变量替换， $y_1=y, y_2=y', \dots$ ，注意odefun方程定义为列向量， $dy=[y(1),y(2),\dots]$

编写yjs.m

$$\frac{d^2x}{dt^2} = 4 \quad \text{初始条件为} \quad \left. \frac{dx}{dt} \right|_{t=0} = 2, \quad x|_{t=0} = 1.$$

```
function dy = yjs(t, y)
```

```
dy = zeros(2,1); dy(1) = y(2); dy(2) = 4;
```

编写ex_yjs.m

```
[T,Y] = ode45(@yjs,[0 12],[1 2]);
```

```
plot(T,Y(:,1),'-',T,Y(:,2),'-.')
```

谢谢！

3.4 解偏微分方程

椭圆型 $-\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right) \varphi(x, y, z) = S(x, y, z)$

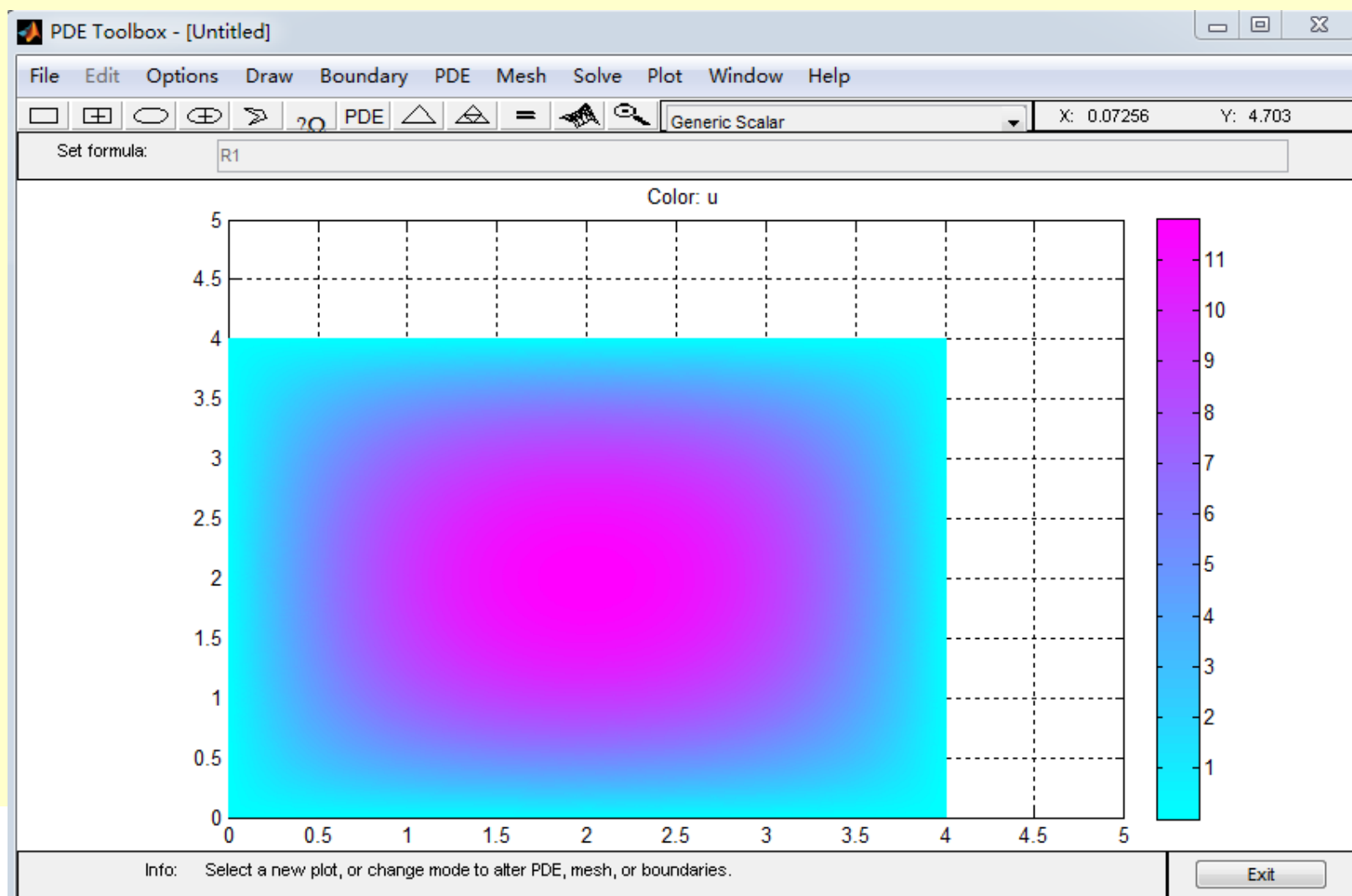
抛物型 $\left(\frac{\partial}{\partial t} - a^2 \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right) \varphi(x, y, z, t) = S(x, y, z, t)$

双曲型 $\left(\frac{\partial^2}{\partial t^2} - a^2 \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right) \varphi(x, y, z, t) = S(x, y, z, t)$

定解条件：初始条件与边界条件

偏微分方程求解工具

pdetool



偏微分方程求解方法

热传导方程 $u_t = a^2 u_{xx}$

$$\frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} \approx a^2 \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{(\Delta x)^2}$$

即 $u(x, t + \Delta t) \approx u(x, t) + \frac{\Delta t}{(\Delta x)^2} a^2 [u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)]$

令 $x = i\Delta x$, $t = j\Delta t$, $i, j = 0, 1, 2, \dots, n$, $r = \frac{\Delta t}{(\Delta x)^2} a^2$, 得

$$u_{i,j+1} = (1 - 2r)u_{i,j} + r[u_{i+1,j} + u_{i-1,j}]$$

初始条件 $u(x, 0) = \varphi(x)$ 就是 $u_{i,1} = \varphi_i$,

偏微分方程求解举例

例 细杆传热问题

$$\begin{cases} u_t = a^2 u_{xx} \\ u(0, t) = 0, \quad u(l, t) = 0 \\ u(x, t = 0) = \varphi(x) \end{cases}$$

其中 $0 \leq x \leq 20$, $a = 10$ 且

$$\varphi(x) = \begin{cases} 1, & (10 \leq x \leq 11) \\ 0, & (x < 10, x > 11) \end{cases}$$

偏微分方程求解举例

```
clear;clc;
x=0:20;a2=10;r=a2*0.02;u=zeros(21,101);
u(10:11,1)=1;%存放初始条件
for j=1:100
    u(2:20,j+1)=(1-
2*r)*u(2:20,j)+r*(u(1:19,j)+u(3:21,j));
    plot(u(:,j));axis([0 21 0 1]);pause(0.1)
end
surf(u)
```

谢谢!