# 常用函数与运算符

## Matlab常用函数

表格1,常用函数

|  |  |
| --- | --- |
| 函数 | 描述 |
| exp | 以e为底的指数 |
| log | 自然对数 |
| log10 | 常用对数 |
| log2 | 以2为底的对数 |
| abs | 求实数的绝对值或复数的模（注意不同于C++中还有fabs） |
| fix | 取整 |
| floor | 向下取整，floor(x)取不大于x的最大整数 |
| ceil | 向上取整，ceil(x)取不小于x的最小整数 |
| round | 四舍五入 |
| mod | 求模或有符号取余 |
| gcd | 最大公约数 |

## Matlab特殊运算符

表格2 特殊运算符

|  |  |
| --- | --- |
| 运算符 | 描述 |
| .\* | 矩阵点乘（数组乘法） |
| ./ | 矩阵点除（数组除法） |
| .^ | 数组乘方 |
| / | 右除 |
| \ | 左除 |
| ./ | 数组右除 |
| .\ | 数组左除 |

# 向量与矩阵

## 创建矩阵

表格3,创建矩阵的常用函数

|  |  |
| --- | --- |
| 函数 | 描述 |
| zeros(m,n) | 生成m\*n的全0矩阵 |
| ones(m,n) | 生成m\*n的全1矩阵 |
| rand(m,n)或randn(m,n) | 取值为0~1的随机数的m\*n矩阵 |
| diag(X) | 产生以向量X为对角线的对角阵 |
| diag(A) | 返回矩阵A的对角线元素 |
| magic(n) | 产生1~n2之间的m阶矩阵，每行、每列以及对角线元素之和相等 |
| eye(n) | n阶单位阵 |
| linspace(min,max) | 列向量，创建取值位于min和max之间的  长度为100的列向量 |
| linspace(min,max,n) | 列向量，创建取值均匀分布于min和max之间  且具有指定长度n的列向量 |
| X=min:h:max | 行向量，创建取值始于min、位于min和max之间  且步长为h的行向量 |

## 矩阵操作

### 算术操作

表格4 算术操作的常用函数

|  |  |
| --- | --- |
| 函数 | 描述 |
| sum(A) | 返回一个行向量，包含矩阵A的每列元素之和，如果A是向量，直接返回向量所有元素之和 |
| sum(sum(A)) | 矩阵A的所有元素之和 |
| mean(A) | 返回矩阵A每列的平局值，如果A是向量，直接返回向量所有元素的平均值 |
| [values,rows]=min(A)/max(A) | 返回矩阵A每列的最大/最小值以及它们在每列中的行号 |
| [m,n]=size(A) | 返回矩阵A的行数和列数 |
| length(A) | 返回向量A的长度（维度）；若A为矩阵，则返回A行数和列数中的最大值，相当于max(size(A)) |
| repmat |  |
| reshape |  |
| mat2cell |  |
| cell2mat |  |

### 逻辑操作

表格5 一些常用逻辑操作

|  |  |
| --- | --- |
| 表达式 | 描述 |
| An | 返回与A同规模的逻辑矩阵，真为1，假为0 |
| find() | ? |
| & | 元素级与 |
| | | 元素级或 |
| && | 逻辑与 |
| || | 逻辑或 |

## 对矩阵的操作函数（cholesky分解、LU分解等）

# 字符串、单元数组和结构体

# 多项式与函数

## 多项式

### 多项式的生成

y=poly2sym(P)

其中P=[aN,…a1,a0]，aK表示K次幂的系数，P是降次排列的多项式系数矩阵，得到的多项式y属于符号表达式

### 多项式的运算函数

### 函数插值与多项式拟合

1. 一维插值

Yq=interp1(X,Y,Xq,method)：X、Y为插值节点的坐标值向量，Xq为待插值点的横坐标向量，可用一维直线离散点采样的方法（如linspace定数采样或等间距采样）或者得到，返回待插值点的插值结果即纵坐标向量，method为插值方法，若未指定插值方法，默认使用线性插值。

表格6 一维插值可选的方法

|  |  |
| --- | --- |
| 参数 | 对应方法 |
| ‘linear’ | 线性插值 |
| ‘nearest’ | 最邻近插值 |
| ‘spline’ | 三次样条插值 |
| ‘cubic’或’pchip’ | 三次插值 |

1. 拟合

p=polyfit(X,Y,n)：返回最小二乘条件下降次排列的最佳拟合多项式系数矩阵，X、Y为横坐标、纵坐标数据向量，n为拟合多项式的次数，可根据次矩阵生成拟合的多项式函数。

## 函数

### 函数的表示

1. M文件
2. 函数定义

function[out1,...,outN]=fun(x1,...,xM)：out1、out2、…、outN为返回值，x1、x2、…、xN为输入参数（形参），fun为函数名

如果无返回值则可直接用function fun(x1,...,xM)定义函数；

如果输入参数可直接用function[out1,...,outN]=fun定义函数。

| nargout | nargoutchk | varargin | varargout|narginchk

1. 函数的输入输出参数

nargin/nargout——调用函数时输入/输出的参数个数

varargin/varargout——输入/输出的参数可以只用一个 varargin 表示，然后调用时输入/输出的参数都在这个元胞数组中，用 varargin{1} 即可获得第一个输入参数

nargoutchk

1. 函数调用

[out1,...,outN]=fun(x1,...,xM)，返回所有值

fun(x1,…,xN)，只返回out1

通过@fun可获取函数句柄，如g=@fun,g(x1,x2...,xN)

1. 匿名函数

fun=@(x1,...,xN)f(x1,...,xN)

fun为函数名，x1、x2、…xN为输入参数，f(x1,...,xN)为函数表达式

### 函数图像的绘制

fplot

### 函数句柄

### 匿名函数作为参数进行地址传递

求插值的函数

# MATLAB符号处理

## 基本的符号运算

### 符号变量的生成

1. x=sym(‘x’)或x=sym(‘x’,’real’)：创建符号变量或创建指定为实数的符号变量
2. S=sym(A)：如果A为字符串，则返回一个符号变量，如果A为数字或矩阵，则返回该参数的符号表示
3. syms var1 var2 … varN：定义多个符号变量
4. syms var1 var2 … varN option：其中option可以是real、unreal、integer等，将定义的所有符号变量指定为option类型
5. syms fun(var1,var2,…varN)：同时定义符号函数fun和符号变量var1、var2、…、varN

* 疑问：solve(sym(‘…’))会出现警告

### 符号变量和数值的转化

1. 符号变量转化为数值
2. subs(s)
3. subs(s,[var1,var2,…varN],[number1,number2,…,numberN])
4. subs(s,{var1,var2,…,varN},{a1,a2,…aN})：其中a1、a2可以是数字或者矩阵
5. 数值转化为符号变量
6. sym(A)：如果A为字符串，则返回一个符号变量，如果A为数字或矩阵，则返回该参数的符号表示
7. findsym(f)：确定表达式f中的符号变量

### 符号表达式的化简（主要针对多项式）

1. 合并同类项
2. collect(f)：将f中每个元素进行合并同类项
3. collect(f,u)：对指定变量u进行合并同类项
4. 展开

expand(f)

1. 嵌套格式（秦九韶算法）

horner(P)：P为符号表达式或由符号表达式组成的矩阵，降低计算的时间复杂度

1. 因式分解

factor(f)：如果f为整数，则返回此数的素数因数

1. simplify
2. simple

### 复合函数

1. compose(f,g)：返回函数f(g(y))，其中f=f(x)，g=g(y)，x、y别为f、g的默认自变量
2. compose(f,g,z)：返回函数f(g(z))，自变量为z
3. compose(f,g,y,z)：返回函数f(g(z))，指定f的自变量为y
4. compose(f,g,x,y,z)：返回返回函数f(g(z))，指定f、g的自变量分别为x、y

### 反函数

1. finverse(f)：返回f的反函数，使用默认自变量
2. finverse(f,v)：返回f的反函数，指定自变量为v

## 符号图形的绘制

四种方式：显函数，隐函数，参数方程，极坐标

方程表示的曲面的绘制？极坐标的绘制

### 符号函数（方程）曲线的绘制（ezplot、ezplot3）

1. 显函数
2. ezplot(f)：绘制函数f在默认区间上的图像
3. ezplot(f,[min,max])：绘制函数f在区间[min,max]上的图像
4. ezplot(f,[min,max],fig-handle)：在句柄为fig\_handle的窗口中绘制图像，注意区间不能省略，注：只能是窗口句柄，不能使用坐标轴句柄
5. h=ezplot(…)：绘图的同时返回ezplot曲线图形（不是坐标轴）的句柄
6. 隐函数
7. ezplot(f)：绘制方程f(x,y)=0在区间、上的图像
8. ezplot(f,[min,max])：绘制方程f(x,y)=0在区间、上的图像
9. ezplot(f,[xmin,xmax,ymin,ymax])：绘制方程f(x,y)=0在区间、上的图像
10. ezplot(f,[xmin,xmax,ymin,ymax],fig\_handle)：在句柄为fig\_handle的窗口中绘制图像，注意区间不能省略
11. 参数方程
12. ezplot(x,y)：绘制参数方程x=x(t)、y=(t)在上的图像
13. ezplot(x,y,[tmin,tmax])：绘制参数方程x=x(t)、y=(t)在上的图像
14. ezplot(x,y,[tmin,tmax],fig\_handle)：在句柄为fig\_handle的窗口中绘制图像，注意区间不能省略
15. 极坐标？
16. 三维曲线
17. ezplot3(x,y,z)：绘制参数方程x=x(t)、y=(t)、z=z(t)在上的图像
18. ezplot3(x,y,z,[tmin,tmax])：绘制参数方程x=x(t)、y=(t)、z=z(t)在上的图像
19. ezplot3(…,axes\_handles)：在句柄为axes\_handle的坐标系中绘制图像，注意ezplot只能使用窗口句柄，不能使用坐标轴句柄，且不能省略自变量区间
20. ezplot3(…,fig\_handles)：在句柄为fig\_handle的窗口中绘制图像
21. h= ezplot3(…)：绘图的同时返回ezplot3曲线图形（不是坐标轴）的句柄
22. 以方程组表示的三维曲线

### 符号函数（方程）曲面的绘制（ezmesh、ezsurf）

1. ezmesh（三维网格图）和ezsurf（三维曲面图）

两者用法相同，这里以ezmesh为例进行介绍：

1. ezmesh(f)：绘制函数f(x,y)在默认区域、上的图像
2. ezmesh(f,[min,max])或ezmesh(f,[xmin,xmax,ymin,ymax]：绘制函数f(x,y)在区域min<x<max、min<y<max或区域xmin<x<xmax、ymin<y<ymax内的图像
3. ezmesh(x,y,z)：在默认区域、绘制三维参数方程的图像
4. ezmesh(x,y,z,[min,max])或ezmesh(x,y,z,[smin,smax,tmin,tmax])：在指定区域绘制三维参数方程的图像
5. ezmesh(…,n)：指定网格数，默认n=60
6. ezmesh(…,axes\_handle)：在句柄为axes\_handle的坐标系中绘图，注意ezplot只能用窗口句柄，不能使用坐标轴句柄，且不能省略自变量区间
7. ezmesh(…,fig\_handle)：在句柄为fig\_handle的坐标系中绘图
8. h=ezmesh(…)：绘图的同时返回ezmesh网格图形（不是坐标轴）的句柄
9. ezmeshc和ezsurfc

在绘制三维网格或曲面的同时绘制等值线，用法与ezmesh和ezsurf相同。

### 等值线的绘制（ezcoutour）

1. ezcontour(f)：绘制二元函数f(x,y)在默认区域的等值线
2. ezcontour(f,[min,max])或ezcontour(f,[xmin,xmax,ymin,ymax])：在指定区域绘制,函数f(x,y)的等值线图
3. ezcontour(…,n)：指定等值线的数目
4. ezcontour(…,axes\_handle)：在句柄为axes\_handle的坐标系中绘图
5. ezcontour(…,fig\_handle)：在句柄为fig\_handle的窗口中绘图
6. h=ezcontour(…)：绘图的同时返回ezcontour等值线图形（不是坐标轴）的句柄

## 符号微积分

### 求极限

1. limit(f,x,a)：当x趋于a时f的极限
2. limit(f,a)：自变量趋于a时f的极限，自变量由findsym确定
3. limit(f)：自变量趋于0时f的极限
4. limit(f,x,a,’left’)和limit(f,x,a,’rught’)：左右极限

### 符号微分

1. diff(S)：求S的导数，自变量由findsym确定
2. diff(S,’v’)：求S对v的导数（符号变量可以不带单引号）
3. diff(S,n)：求S的n阶导数
4. diff(S,’v’,n)：求S对v的n阶导数（符号变量可以不带单引号）
5. jacobian(F,V)：求函数向量F对自变量向量V的微分（逐个求导），得到F的雅克比矩阵

### 符号积分

1. int(S)：求S的不定积分，自变量由findsym函数确定
2. int(S,v)：求S对v的不定积分
3. int(S,a,b)：求S在区间[a,b]上的定积分，自变量由findsym函数确定
4. int(S,v,a,b)：求S对v在区间[a,b]上的定积分

### 级数求和

1. symsum(s)：设findsym确定的符号变量为k（包含在s中），则该表达式表示计算从s(0)到s(k-1)的和
2. symsum(s,v)：计算表达式s由0到v-1的和
3. symsum(s,a,b)：计算自变量从a到b之间表达式s的和
4. symsum(s,v,a,b)：计算v从a到b之间表达式s的和

### Taylor级数

## 符号方程的求解

### 代数方程求解（solve、vpasolve）

1. 解析解solve
2. y=solve(eq)：求解方程eq的解，对默认自变量求解
3. y=solve(eq,var)：求解方程eq的解，对指定变量var求解
4. [y1,...,yN]=solve(eqs,vars)：求解方程组eqs=[eq1,eq2,…]，变量vars=[var1,var2,…]
5. 数值解vpasolve

vpasolve的参数及用法与solve基本相同，多出的init\_guess为方程解的初值（近似值）

1. y=vpasolve(eq)
2. y=vpasolve(eq,var)
3. y=vpasolve(eq,var,init\_guess)
4. [y1,...,yN]=vpasolve(eqs,vars)
5. [y1,...,yN]=vpasolve(eqs,vars,init\_guess)

### 微分方程求解(dsolve)

1. r=dsolve(‘eq’,’cond’,’v’)
2. [y1,y2,…]=dsolve(‘eq1,eq2,…’,’cond1,cond2,…’,’v’)

或者[y1,y2,…]=dsolve(‘eq1’,’eq2’,…,’cond1’,’cond2’,…,’v’)

其中v为指定的自变量，若不指定则用默认自变量

eq1、eq2、…表示微分方程，常用微分算子D表示微分，如Dy=dy/dv、D2y=d2y/dv2，

cond1、cond2、…表示微分方程的初值，通常表示为y(a)=b、y(a,b)=c（多元函数）、Dy(a)=b、的形式，可以不指定初值

# Matlab绘图

## 基本图形的绘制

### 一维直线、二维平面和三维空间离散点采样

对于绘制一些确定的函数，已知其表达式，绘图前可以先对自变量在一定范围内取样。

1. 一维直线离散点采样

绘制曲线（二维或三维）只有一个自变量，设为X，采样方式如下：

|  |  |
| --- | --- |
| **函数** | **描述** |
| X=linspace(min,max) | 列向量，创建取值位于min和max之间的  长度为100的列向量 |
| X=linspace(min,max,n) | 列向量，创建取值均匀分布于min和max之间  且具有指定长度n的列向量 |
| X=min:h:max | 行向量，创建取值始于min、位于min和max之间  且步长为h的行向量 |

1. 二维平面离散点（网格点）采样

使用meshgrid函数，用于产生二维或三维网格区域的散点。

1. [X,Y]=meshgrid(xgv,ygv)

X、Y是返回的矩阵，xgv、ygv为向量，可用一维直线离散点采样的方法得到。

若xgv与ygv分别为m、n维向量，则X是将向量xgv作为横向量排n行的nm矩阵，Y是将向量ygv作为列向量排n列的nm矩阵，从而产生以X为横坐标、Y为纵坐标的nm个网格点，结合矩阵的数组运算（点乘、点除、幂等）生成对应函数值，可用于绘制函数曲面。

例：绘制x^2+y^2的图像[X,Y]=meshgrid(xgv,ygv),surf(X,Y,X.^2+Y.^2)，

注意：这里使用surf(X,Y,X.^2+Y.^2)而不是surf(X.^2+Y.^2)，后者的绘制图像的横坐标和纵坐标分别为矩阵X.^2+Y.^2对应点的行号和列号，而不是以矩阵X、Y的元素值为横坐标和纵坐标。

1. [X,Y]=meshgrid(gv)

根据n维向量gv生成nn个二维平面离散点，离散点的X、Y坐标坐标均为gv的元素值。

1. 三维离散点采样
2. [X,Y,Z]=meshgrid(xgv,ygv,zgv)
3. [X,Y,Z]=meshgrid(gv)

### 二维图形绘制（plot）

1. plot(Y)

若Y为向量，则以向量索引为横坐标、元素值为纵坐标绘制图形；

若Y为矩阵，则对矩阵各列分别绘制图形，仍以各列索引为横坐标、元素值为纵坐标绘制图形；

若Y为复向量，则以实部为横坐标、虚部为纵坐标绘制图形。

1. plot(x,y)

x、y均为n维向量时，以x为横坐标、y为纵坐标绘制图形；

x为n维向量、y为mn或nm矩阵，则以x为横坐标绘制y的m个n维向量；

x、y均为mn矩阵，以x的各列为纵坐标、y的各列为横坐标绘制图形。

1. plot(x1,y1,...,xN,yN)：参数为2N个向量，将N个以x、y为横、纵坐标构成的图形绘制在同一坐标系中
2. plot(x,y,LineSpec)：指定线型、点的形状、颜色等属性

例：plot(x1,y1,'\*b-')，属性在单引号中，没有顺序要求

1. plot(axes\_handle,…)：在句柄为axes\_handle的指定坐标系中绘制图形
2. h=plot(…)：绘图的同时返回plot曲线图形句柄

### 三维图形绘制（plot3、mesh、surf）

1. plot3
2. plot3(X,Y,Z)

当X、Y、Z为长度相同的向量时，绘制一条分别以X、Y、Z为x、y、z坐标的空间曲线；

当X、Y、Z为规模相同的mn矩阵时，以每个矩阵的对应列为x、y、z坐标绘制m条空间曲线。

1. plot3(x1,y1,z1,…,xN,yN,zN)：参数为3N个向量，将N个以x、y、z为横、纵坐标构成的图形绘制在同一坐标系中
2. plot3(…,LineSpec)：指定曲线和点的属性
3. plot3(axes\_handle,…)：在句柄为axes\_handle的指定坐标系中绘制图形
4. h=plot3(…)：绘制的同时返回plot3曲线句柄
5. mesh（三维网格图）和surf（三维曲面图）

两者用法相同，这里以mesh为例

1. mesh(X,Y,Z)：绘制网格图，图像颜色由Z决定（与高度成正比）。X、Y可以为矩阵或向量，Z只能为矩阵，当三者均为矩阵时必须规模一致（一一对应，共mn个点），若X、Y分别为m和n维向量，则(X(i),Y(j),Z(i,j))为图像的各个节点（X和Y中元素两两组合，共mn个点）。
2. mesh(Z)：以Z的元素值为竖坐标，元素对应的矩阵行号和列号分别为横坐标和纵坐标绘制图像。
3. mesh(…,C)：由矩阵C指定图像颜色，C的规模应
4. mesh(...,'PropertyName',PropertyValue,...)
5. mesh(axes\_handles,...)：在句柄为axes\_handle的指定坐标系中绘制图形
6. h = mesh(...)：返回绘制的三维网格图形句柄
7. surf

### 等值线绘制（countour）

### 散点图绘制（scatter）

## Matlab句柄图形

peaks，gca（当前axes即函数曲线或曲面的句柄），gcf（当前figure的句柄），gco，cla，clf，reset，subplot,close，interp拟合曲面等

## Figure对象操作

### 创建窗口

* 1. figure
  2. figure(‘PropertyName’,’PropertyValue’)
  3. figure(h)：h需为已存在对象的句柄，否则需是数字
  4. h=figure(‘PropertyName’,’PropertyValue’)

figure类的属性字段可在帮助文档中查询

### 获取和设置figure对象的属性值

1. 获取属性值
2. get(h)/get(h,’PropertyName’)：返回所有/指定属性值
3. a=get(h)：返回包含所有属性值的结构体a
4. a=get(h,’PropertyName’)
5. 设置属性值
6. set(h,‘PropertyName’,’PropertyValue’)
7. set(h,A)：其中A为结构体，其域名为属性名，值为对应的属性值
8. set(h,pn,pv)：其中pn、pv为单元数组，pn指定属性名称，pv指定对应属性值
9. fig.PropertyName=’PropertyValue’：对figure类的对象fig的相应公共属性赋值

### 删除窗口

close

## Axes对象操作

### 坐标轴控制

1. 坐标轴的区间
2. axis([xmin,xmax,ymin,ymax])：不能缺省参数
3. axis([xmin,xmax,ymin,ymax,zmin,zmax])：不能缺省参数
4. xlim：获取当前x轴的范围
5. xlim([xmin,xmax])：设置x轴的范围为[xmin,xmax]
6. xlim(axes\_handle,[min,max])：设置指定坐标系中x轴的范围

xlim、ylim、zlim的用法相同

1. 坐标轴的刻度
2. axis equal：设置等刻度（即等分划值）坐标轴
3. axis sqare：设置等长坐标轴（即坐标轴长度相等）
4. 坐标系的标题、图例（属于axes对象吗？）
5. title()
6. legend()
7. 坐标轴标注
8. xlabel(str)
9. xlabel(str,Name,Value)
10. xlabel(ax,\_\_\_)
11. t=xlabel(\_\_\_)

xlabel、ylabel、zlabel的用法相同

1. 坐标轴网格
2. grid on/off：显示/隐藏当前坐标轴下的网格线
3. grid(axes\_handle,’…’)：显示/隐藏指定坐标轴下的网格线

### 曲面的网格控制

1. shading
2. shaping faceted：包含网格线、未光滑（默认形式）
3. shaping flat：仅去除构成曲面的网格线，未光滑
4. shaping interp：通过插值光滑曲面
5. shading(axes\_handle,’…’)： axes\_handle表示坐标轴句柄，后面单引号中的参数可以是前三种之间的任一种，例：shading(gca,’interp’)
6. hidden
7. mesh

### 图形子窗口

1. subplot(m,n,p)：将窗口分为mn个子区域，在第p个区域（从左到右、从上到下的顺序）绘制图像，如果p为向量，则绘图区域为p指定的区域的合并
2. subplot(m,n,p,’replace’)：如果指定区域已存在坐标系，则删除该坐标系并用新的坐标系代替
3. subplot(m,n,p,axes\_handle)：将句柄为axes\_handle的坐标系中置于窗口中的指定区域
4. h=subplot(…)：指定绘图子区域并返回坐标轴句柄

三维曲面渐变色（颜色随z值而变化或者是三维空间点坐标x、y、z的函数）

## 条形图、饼状图、矢量图

### 条形图(bar)

* 1. bar(Y)：若Y为矩阵，Y的每一行聚集在一起，横坐标用矩阵行号表示。
  2. bar(X,Y)：X为横坐标，其中元素可以不单调但必须互异
  3. bar(X,Y,width)：指定条形图的宽度，默认宽度为0.8
  4. bar(axes\_handle,…)：在句柄为axes\_handle的坐标系中绘图
  5. h=bar(…)：返回绘制的bar图形句柄
  6. bar3(…)：绘制三维条形图，用法与二维类似

### 直方图(hist、rose)

1. hist（是频数图？纵坐标是频数？）

### 饼状图(pie)

1. pie(X)：逆时针排序，如果X为矩阵，则按列的顺序排列
2. pie(X,labels)：标注labels是元素个数与X相同的单元数组，元素为字符串
3. pie(axes\_handle,…)：在句柄为axes\_handle的坐标系中绘图
4. h=pie(…)：返回绘制的pie图形句柄

### 罗盘图和矢量图(compass、quiver)

1. 罗盘图
2. compass(U,V)：以原点为始端，U、V分别为X、Y坐标轴分量绘制罗盘图，U、V可以是数字或维度相同的向量。
3. compass(axes\_handles,U,V)：在句柄为axes\_handles的坐标系中绘图
4. h=compass(…)：返回罗盘图的句柄
5. 矢量图
6. quiver(X,Y,U,V)：X、Y可以是维数与U、V相同的向量，X,、Y指定矢量的位置，U、V为矢量的分量
7. quiver(U,V)：矢量位置采用默认值
8. quiver(axes\_handle,…)：在句柄为axes\_handle的坐标系中绘图
9. h=quiver(…)：返回绘制的矢量图的句柄
10. quiver3(…)：三维矢量，调用方式与二维类似

## matlab图形类

## line、surface、bar、pie、quiver、scatter

## 一级分类figure、二级分类axes、三级分类图形类

## 元胞数组

## 绘图显示不同字体

## y\_str = '\fontname{宋体}位置偏差\fontname{Times New Roman}/m'

# 统计分析

## rcoplot，regress回归分析