# 数值分析

本节介绍数学运算中的数值求解方法，包括求极值、求根、曲线拟合、数值积分、和与积的数值计算、线性规划与非线性规划等。

## 1、数值求和与数值求积

NSum[f,{i,imin,imax}]求数值和 $\sum_{i=i\min}^{i\max} f$；

NSum[f,{i,imin,imax,di}]同上，但求步长以 di 增加时的和；

NSum[f,{i,imin,imax},{j,jmin,jmax},…]求多重数值和；

NSum[f,{i,imin,imax,di},{j,jmin,jmax,di},…] 同上，步长为 di；

NProduct[f,{i,imin,imax}] 求数值积 $\prod_{i=i\min}^{i\max} f$；

NProduct[f,{i,imin,imax,di}] 以下 2 个命令参见 Nsum[]的说明；

Product[f,{i,imin,imax},{j,jmin,jmax},…]

NProduct[f,{i,imin,imax,di},{j,jmin,jmax,di},…]

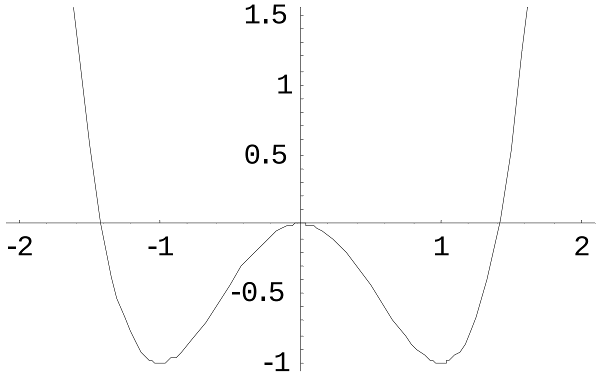
```
In[1]:= NSum[(n^2 + 100) / (n^10 + 20 n + 100), {n, 1, 1000}]

Out[1]= 0.926025

In[6]:= NSum[i + j, {i, 1, 10, 0.5}, {j, 1, 21, 2}]

Out[6]= 3448.5 + 0. i

In[7]:= NProduct[10 + i, {i, 1, 99}]

Out[7]= 3.97889×10^169 + 0. i

In[8]:= NProduct[10 + i, {i, 1, 2, 0.1}]

Out[8]= 4.63307×10^11

In[9]:= NSum[10 + i, {i, 1, 2, 0.1}]

Out[9]= 126.5
```

## 2、极值

- **FindMinimum[f,{x,x0}]**从初值x=x0开始寻找函数f的极小值
- **FindMinimum[f,{x,x0},{y,y0},…]**从初值(x0,y0,…)开始寻找寻找函数f的极小值

对于FindMinimum[]，选取不的初值可能会得到不同的极值，例如

```
Clear[f, x, y]; f[x_] := x^4 - 2 x^2;
{FindMinimum[f[x], {x, 0.5}], FindMinimum[f[x], {x, -0.5}]}
{{-1., {x → 1.}}, {-1., {x → -1.}}}
Plot[f[x], {x, -2, 2}]
```




另外，可以应用 **FindMaximum** 命令求出函数的极大值,其用法与上面相同.

## 3、方程的根

**•FindRoot[lhs==rhs,{x,x0}]** 从初值x=x0开始寻找方程的根

**•FindRoot[lhs==rhs,{x,{x0,x1}}]** 同上，但初值为(x0,x1内

**•FindRoot[lhs==rhs,{x,xstart,xmin,xmax}]** 以初值xstart求解方程，若x在区间(xmin,xmax)之外就停止计算

**•FindRoot[{eqn1,eqn2,…},{x,x0},{y,y0},…]** 求联立方程的根

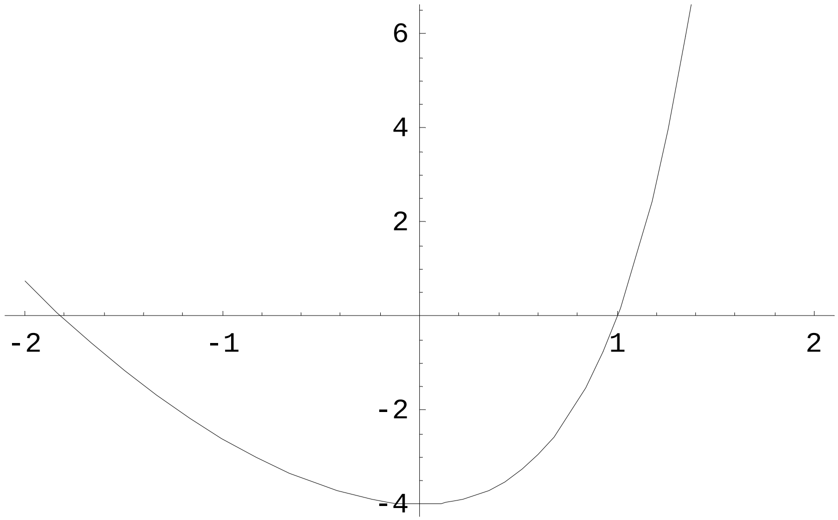
FindRoot[]使用割线法求函数的根，因此初值的选取很重要，不同的初值可能得到不同的根。

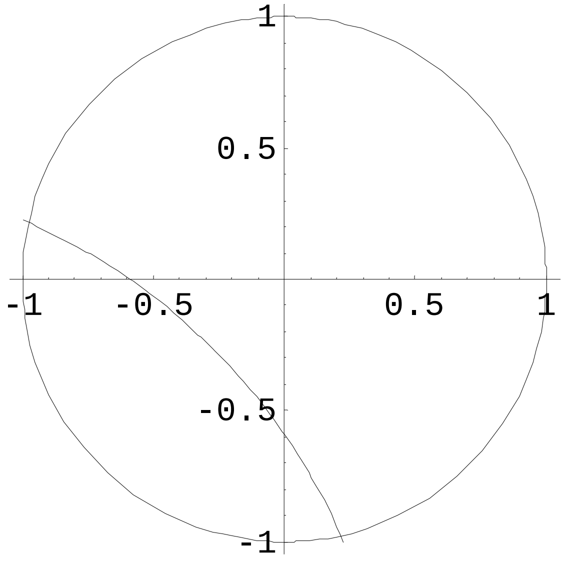
```
Clear[x, f]; f[x_] := x^2 + (1 - Exp[x])^2 - 4;
{FindRoot[f[x], {x, -1.7}], FindRoot[f[x], {x, 1}]}
```

{{x → -1.81626}, {x → 1.00417}}

```
FindRoot[{x^2 + y^2 == 4, Exp[x] + Exp[y] + Sin[x] + Sin[y] == 1},
 {x, -2}, {y, 0}]
```

{x → -1.96757, y → 0.358696}



## 4、数值积分

- **NIntegrate[f,{x,a,b}]** 函数f的数值积分
- **NIntegrate[f,{x,a,b},{y,y1[x],y2[x]]** 函数f的二重数值积分

由于大部分函数使用**Integrate[]**命令不能求出其理论积分，因此，我们只能求它的数值积分。

```
Integrate[Sin[Sin[x]], {x, 0, 1}]
```

$$\int_0^1 \mathrm{Sin}[\mathrm{Sin}[x]]\,dx$$

```
NIntegrate[Sin[Sin[x]], {x, 0, 1}]
```

0.430606

```
NIntegrate[Sin[Sin[x]], {x, 0, 1}, WorkingPrecision → 20]
```

0.4306061031

## 5、数据的插值逼近

- **Interpolation[{{x1,y1},{x2,y2},…}]** 给出通过数据点**(x1,y1), (x2,y2),…**的一个单变量近似函数
- **Interpolation[{{x1,y1,z1},{x2,y2,z2},…}]** 给出通过数据点**(x1,y1,z1),(x2,y2,z2),…**的一个双变量近似函数
- **Interpolation[{{x1,y1,…},{x2,y2,…},…}]** 多个变量近似函数

在**Mathematica**中，近似函数是由**InterpolatingFunction[]** 生成的，其具体用法参见如下的例子。

```
Clear[d, data, x, y, f, sinxy];
data = Table[{x, Exp[x]}, {x, 0, 1, 0.1}]
{{0, 1}, {0.1, 1.10517}, {0.2, 1.2214}, {0.3, 1.34986},
 {0.4, 1.49182}, {0.5, 1.64872}, {0.6, 1.82212},
 {0.7, 2.01375}, {0.8, 2.22554}, {0.9, 2.4596}, {1., 2.71828}}
f = Interpolation[data]
InterpolatingFunction[{{0., 1.}}, <>]
{f[0.15], f[0.35], f[0.75]}
{1.16183, 1.41906, 2.117}
d = Table[{x, y, Sin[x y]}, {x, 0, Pi, 0.5}, {y, 0, Pi, 0.5}];
data = Flatten[d, 1]; sinxy = Interpolation[data]
InterpolatingFunction[{{0., 3.}, {0., 3.}}, <>]
{Sin[0.4 0.6], sinxy[0.4, 0.6]}
{0.237703, 0.237718}
```

上页 下页 退出

# 6、曲线拟合

•Fit[data,funs,vars] 用变量vars,函数集合funs拟合一组数据

线性拟合的意思是：给定一列数据及拟合函数集{f1 ,f2 ,…,fn }，Fit[]命令给出该函数集的拟合函数k1f1+k2f2+…knfn，参见下面的例子。

```
d = {{1, 2}, {3, 7}, {5, 9}, {7, 15}, {9, 35}, {11, 80}, {13, 150}};
x =.; f[x_] = Fit[d, {1, x, Log[x]}, x]
-13.0035 + 23.7089 x - 65.4063 Log[x]
d = {{1, 2, 3}, {4, 5, 6}, {5, 6, 7}, {6, 7, 8}, {7, 8, 9}}; y =.;
Fit[d, {1, x, y, x y, x^2, y^2}, {x, y}]
```

$$0.9 - 0.2x - 0.1x^2 + 0.7y - 0.3xy + 0.4y^2$$

利用软件包nonlinearfit.m，可以进行非线性拟合，装入此软件后，系统会提供以下函数:

•NonlinearFit[data,model,vars,params] 利用数据data拟合函数模型model，其中vars是函数变量集合，params是要拟合的参数集合。

```
<< statistics`NonlinearFit`

clear[x, d, a, b]; d = Table[{x, 3 x + Exp[2 x]}, {x, 0, 1, 0.1}] // N

{{0., 1.}, {0.1, 1.5214}, {0.2, 2.09182}, {0.3, 2.72212},
 {0.4, 3.42554}, {0.5, 4.21828}, {0.6, 5.12012},
 {0.7, 6.1552}, {0.8, 7.35303}, {0.9, 8.74965}, {1., 10.3891}}

NonlinearFit[d, b x + Exp[a x], x, {a, b}]
```

$$e^{2.\,x}+3.\,x$$

## 7、 线性规划与非线性规划

对于线性规划，**Mathematica**提供了三个函数，分别是：

- **ConstrainedMax[f,{inequalities},{x,y,…}]** 求目标函数**f**在不等式约束条件下的极大值。
- **ConstrainedMin[f,{inequalities},{x,y,…}]** 同上，但求极小值
- **LinearProgramming[c,m,b]** 求目标函数**cx**在约束**mx≥b**和**x≥0**下取得的最小值向量**x**，即求线性规划问题

$$\begin{aligned} &\min \quad c^T x \\ &\text{s.t.} \quad mx \geq b \\ &\qquad\quad x \geq 0 \end{aligned}$$

的最小值。

```
ConstrainedMax[3 x1 + x2 + 3 x3,
 {2 x1 + x2 + x3 ≤ 2, x1 + 2 x2 + 3 x3 ≤ 5, 2 x1 + 2 x2 + x3 ≤ 6}, {x1, x2, x3}]
```

$$\left\{\frac{27}{5}, \left\{x1 \to \frac{1}{5}, x2 \to 0, x3 \to \frac{8}{5}\right\}\right\}$$

下面说明LinearProgramming[]的使用方法，请看例子。求

$$\begin{aligned} \min \quad & x1 + 6x2 - 7x3 + x4 + 5x5 \\ \text{s.t.} \quad & 5x1 - 4x2 + 13x3 - 2x4 + x5 = 20 \\ & x1 - x2 + 5x3 - x4 + x5 = 8 \end{aligned}$$

```
Clear[c, b, A]; c = {1, 6, -7, 1, 5}; b = {20, -20, 8, -8};
A = {{5, -4, 13, -2, 1}, {-5, 4, -13, 2, -1}, {1, -1, 5, -1, 1},
  {-1, 1, -5, 1, -1}};
LinearProgramming[c, A, b]
```

$$\left\{0, \frac{4}{7}, \frac{12}{7}, 0, 0\right\}$$

上页 下页 退出

对于非线性规划,可使用以下函数:

**NMaximize[函数, 变量]与NMinimize[函数, 变量]求出函数的局部极值, NMaximize[{函数, 条件}, 变量]与NMinimize[{函数, 条件}, 变量]求出函数的某些条件下的的局部极值, 请注意, 这2个函数只在V4.2后的版本中才能够使用, 以前版本中没有.**

```
(* 求出函数z=Sin[x+y]-x^2-y^2 的极大值 *)
NMaximize[Sin[x + y] - x^2 - y^2, {x, y}]

{0.400489, {x → 0.369543, y → 0.369543}}

(* 求出z=x^2+(y-0.5)^2 在y≥0 及y≥x+1 下的极小值 *)
(* 若有多个条件,若是"并且" 关系,用符号"&&" 表示 *)
(* 若是"或者" 关系,用符号"||" 表示 *)
NMinimize[{x^2 + (y - .5)^2, y ≥ 0 && y ≥ x + 1}, {x, y}]

{0.125, {x → -0.249982, y → 0.750018}}
```

```
(* 求出z=x^2+y^2 在满足条件x≥1 或者y≥1 下的极小值 *)
NMinimize[{x^2 + y^2, x ≥ 1 || y ≥ 2}, {x, y}]
```

$\{1., \{x \to 1., y \to 1.97376\times 10^{-8}\}\}$

```
(* 变量也可以用Element[变量,变量类型] 定义类型 *)
(* 或者等价形式: 变量∈变量类型    来定义        *)
(* 其中,变量类型有:Reals,Integers,Complex       *)
(* Postive, Negative等等                          *)
(* 下面是在限定x≥1 并且为整数时求极值           *)
NMinimize[{(x - 1/3)^2 + (y - 1/3)^2, x ≥ 1, x ∈ Integers}, {x, y}]
```

$\{0.444444, \{x \to 1, y \to 0.333333\}\}$

```
NMinimize[{x Sin[x], x ≤ 6 && x ≥ 1}, {{x, 2, 3}}]
```

$\{-4.81447, \{x \to 4.91318\}\}$

```
NMinimize[{x^2 + y^2 + z^2, 3 ≤ x ≤ 4}, {x, {y, 2, 5}, z}]
```

$\{9., \{x \to 3., y \to -1.71155\times 10^{-9}, z \to 2.85827\times 10^{-9}\}\}$

# 8、微分方程的数值解

- NDSolve[{eqn1,eqn2,…},y[x],{x,xmin,xmax}] 求微分方程的数值解，x的范围从xmin到xmax，其它与Dsolve[]命令相同。
- NDSolve[{eqn1,eqn2,…},{y1[t],y2[t],…},{t,tmin,tmax}]求微分方程组的数值解，其中t由tmin到tmax。

```
s = NDSolve[{y''[x] + 2 y'[x] + 10 y[x] == Sin[2 x], y[0] == 1, y'[0] == 0},
   y[x], {x, 0, 10}];
```

```
{{y[x] → InterpolatingFunction[{{0., 10.}}, <>][x]}}
```

```
f[x_] = s[[1, 1, 2]]
```

```
InterpolatingFunction[{{0., 10.}}, <>][x]
```

```
Plot[f[x], {x, 0, 10}]
```