数值微分计算

数值微分问题

- 数值微积分在高等数学课程中没有介绍，但可能广泛用于实际问题
  - 未知函数，只知道一些离散的数据点
- 本节主要内容
  - 数值微分算法
  - 中心差分方法及其MATLAB实现
  - 二元函数的梯度计算

一阶数值微分算法

- 导数的定义

$$y'(t) = \lim_{\Delta t \to 0} \frac{y(t+\Delta t) - y(t)}{\Delta t}$$

- 前向差分公式

$$y'_i \approx \frac{\Delta y_i}{\Delta t} = \frac{y_{i+1} - y_i}{\Delta t}$$

- 后向差分公式

$$y'_i \approx \frac{\Delta y_i}{\Delta t} = \frac{y_i - y_{i-1}}{\Delta t}$$

- 算法精度 $o(\Delta t)$

中心差分算法高精度公式

- 微分公式

$$y_i' \approx \frac{-y_{i+2} + 8y_{i+1} - 8y_{i-1} + y_{i-2}}{12\Delta t}$$

$$y_i'' \approx \frac{-y_{i+2} + 16y_{i+1} - 30y_i + 16y_{i-1} - y_{i-2}}{12\Delta t^2}$$

$$y_i''' \approx \frac{-y_{i+3} + 8y_{i+2} - 13y_{i+1} + 13y_{i-1} - 8y_{i-2} + y_{i-3}}{8\Delta t^3}$$

$$y_i^{(4)} \approx \frac{-y_{i+3} + 12y_{i+2} - 39y_{i+1} + 56y_i - 39y_{i-1} + 12y_{i-2} - y_{i-3}}{6\Delta t^4}$$

- 算法精度 $o(\Delta t^4)$

中心差分算法的MATLAB实现

- 函数调用 $[d_y, d_x] = \mathtt{diff\_ctr}(y, \Delta t, n)$
- 函数清单

```
function [dy,dx]=diff_ctr(y,Dt,n)
y1=[y 0 0 0 0 0 0]; y2=[0 y 0 0 0 0 0]; y3=[0 0 y 0 0 0 0];
y4=[0 0 0 y 0 0 0]; y5=[0 0 0 0 y 0 0]; y6=[0 0 0 0 0 y 0];
y7=[0 0 0 0 0 0 y];
switch n
   case 1, dy=(-y1+8*y2-8*y4+y5)/12/Dt;
   case 2, dy=(-y1+16*y2-30*y3+16*y4-y5)/12/Dt^2;
   case 3, dy=(-y1+8*y2-13*y3+13*y5-8*y6+y7)/8/Dt^3;
   case 4, dy=(-y1+12*y2-39*y3+56*y4-39*y5+12*y6-y7)/6/Dt^4;
end
dy=dy(5+2*(n>2):end-4-2*(n>2)); dx=([2:length(dy)+1]+(n>2))*Dt;
```

例3-56 数值微分逼近

- 对函数 $f(x)=\dfrac{\sin x}{x^2+4x+3}$
  - 生成数据，用数值微分法求1～4阶导数
  - 并与其导数的解析解比较精度
- 输入函数，并求解析解，并代入x向量得出精确解

```
>> h=0.05; x=0:h:pi; syms x1;
   y=sin(x1)/(x1^2+4*x1+3);
   yy1=diff(y); f1=subs(yy1,x1,x);
   yy2=diff(yy1); f2=subs(yy2,x1,x);
   yy3=diff(yy2); f3=subs(yy3,x1,x);
   yy4=diff(yy3); f4=subs(yy4,x1,x);
```

与解析解的精度比较

- 比较不同阶的导数

```
>> y=subs(f,x1,x); [y1,dx1]=diff_ctr(y,h,1);
   subplot(221), plot(x,f1,dx1,y1,':');
   [y2,dx2]=diff_ctr(y,h,2);
   subplot(222), plot(x,f2,dx2,y2,':')
   [y3,dx3]=diff_ctr(y,h,3);
   subplot(223), plot(x,f3,dx3,y3,':');
```

```
   [y4,dx4]=diff_ctr(y,h,4);
   subplot(224), plot(x,f4,dx4,y4,':')
```

- 分析误差

```
>> norm(double((y4-f4(4:60))./f4(4:60)))
```

二元函数梯度计算

- 多变量函数 $f(x_1, x_2, \ldots, x_n)$
- 梯度是其对各个自变量偏导数构成的向量

$$\boldsymbol{v} = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots, \frac{\partial f}{\partial x_n}\right]$$

- 函数 gradient()的调用格式 $[f_x, f_y] = \texttt{gradient}(z)$
- 计算梯度 $f_x = f_x/\Delta x$， $f_y = f_y/\Delta y$
- 其中 Δx 和 Δy 分别为 x 和 y 生成网格的步距

例3-57 梯度的数值计算

- 已知 $z = f(x,y) = (x^2 - 2x)\mathrm{e}^{-x^2-y^2-xy}$
  - 生成数据
  - 用数值方法求梯度并分析误差
- MATLAB求解语句

```
>> syms x y; f=(x^2-2*x)*exp(-x^2-y^2-x*y);
   [x1,y1]=meshgrid(-3:.2:3,-2:.2:2);
   z1=double(subs(f,{x,y},{x1,y1}));
   [zx1,zy1]=gradient(z1); zx1=zx1/0.2;
   zy1=zy1/0.2; contour(x1,y1,z1,30);
   hold on; quiver(x1,y1,-zx1,-zy1), hold off
```

误差曲面绘制

- 和理论值相比，绘制误差曲面

```
>> zx=diff(f,x); zy=diff(f,y);
   zx2=double(subs(zx,{x,y},{x1,y1}));
   zy2=double(subs(zy,{x,y},{x1,y1}));
   subplot(211), surf(x1,y1,abs(zx2-zx1));
   axis([-3 3 -2 2 0,0.15])
   subplot(212); surf(x1,y1,abs(zy2-zy1));
   axis([-3 3 -2 2 0,0.15])
```

加密网格再重新测试

- 将网格加密一倍

```
>> [x2,y2]=meshgrid(-3:.1:3,-2:.1:2);
   z2=double(subs(f,{x,y},{x2,y2}));
   [zx3,zy3]=gradient(z2);
   zx3=zx3/0.1; zy3=zy3/0.1;
   zx2=double(subs(zx,{x,y},{x2,y2}));
   zy2=double(subs(zy,{x,y},{x2,y2}));
```

- 绘图比较

```
>> subplot(211), surf(x2,y2,abs(zx3-zx2));
   axis([-3 3 -2 2 0,0.06])
   subplot(212); surf(x2,y2,abs(zy2-zy3));
   axis([-3 3 -2 2 0,0.06])
```