《数学实验》第12讲

主要内容：

微分方程数值解：

微分方程简介、ODE、Euler法、实验

数值积分函数：

思想简介、quad积分、应用实例

微分方程模型与ODE函数

一阶常微分方程的原型系统是

$$\frac{dy}{dt} = f(t, y)$$

其中，y = y(t) 是未知函数，

f(t, y) 是给定的二元函数.

初值条件 y(t0) = y0

初值问题
$$\begin{cases} \dfrac{dy}{dt} = f(t, y) \\ y(t_0) = y_0 \end{cases}$$

常微分方程组

定义

$$\begin{cases} \dfrac{dy_1}{dt} = \alpha y_1 + \beta y_2 + g_1(t) \\ \dfrac{dy_2}{dt} = \gamma y_1 + \delta y_2 + g_2(t) \end{cases}$$

其中$\alpha, \beta, \gamma, \delta$为已知系数，$g_1(t), g_2(t)$ 为 t 的已知函数，

初值 $y_1(t_0) = y_{1,0}, y_2(t_0) = y_{2,0}$ ，称为常微分方程组。

求微分方程(或方程组)的数值解，就是要寻找解函数在一系列离散节点$t_1 < t_2 < \ldots\ldots < t_n < t_{n+1}$ 上的近似值 $y_1, y_2, \ldots, y_n$ 。

MATLAB求解微分方程

Runge-Kutta (不要求)自适应步长算法：

ode23	显式的单步Runge-Kutta低阶（2阶到3阶）解法。适用具有一定难度对精度要求不高，或者f(t,y)不平滑（非连续）的问题
ode45	显式的单步Runge-Kutta中阶（4阶到5阶）解法。适用对精度有一定要求的非难度问题

MATLAB求解微分方程【自适应步长算法(ode23)】

求解步骤:

(1)用函数文件定义一阶微分方程(或方程组) 右端函数

(2) 用MATLAB命令ode23()求数值解。

基本语法:

[T, Y] = ode23('F',Tspan,y0)

其中： ‘F’ 是包括函数文件名字的字符串或函数句柄。

Tspan = [t0，tN]是常微分方程求解区域;

y0是表示初始条件;

求解常微分方程组:

基本语法:

[T, Y] = ode23('F',Tspan,y0)

T：求解区域内离散数据、

Y：求解区域内离散数据的对应数值解。

‘F’：常微分方程组右端项组成的列向量.

y0：列向量（初值）.

注:

数值解向量Y 的每一列对应于一个未知函数的数值解.

Tspan可以指定具体节点. 例:linspace(a,b,step)

示例：马尔萨斯模型
$$\begin{cases} \dfrac{dN}{dt} = rN \\ N(0) = N \end{cases}$$

$N(t)$表示人口数量，r为人口变化率 。

以1994 年我国人口为12亿为初值，取人口变化率r =0.015，求解常微分方程

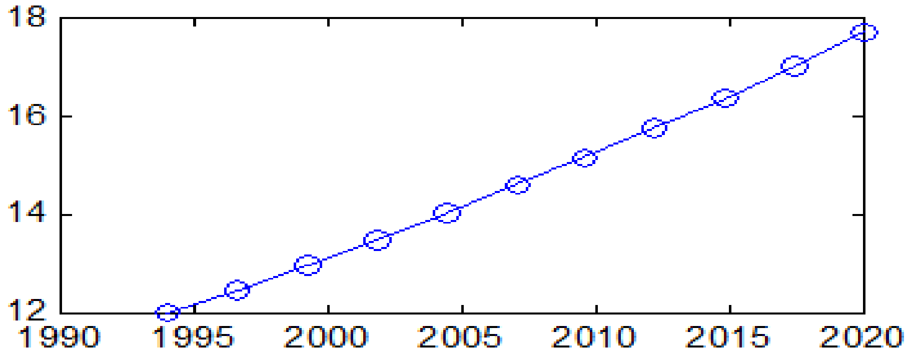
$$\begin{cases} \dfrac{dN}{dt} = 0.015N \\ N(1994) = 12 \end{cases}$$

编写程序fun1.m:

```
function dfun=fun1(t,N)
dfun=0.015*N;
```

命令窗口输入:

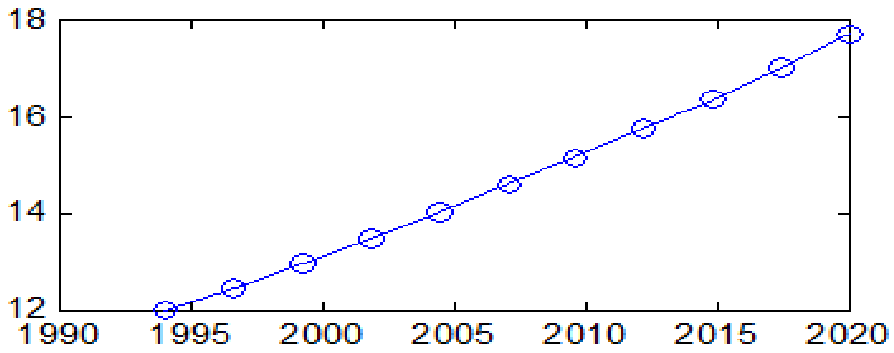
```
[t,N]=ode23(@fun1,[1994,2020],12)
plot(t,N,'o')
```



编写一个程序文件实现求解：

```
function test
[t,N]=ode23(@fun1,[1994,2020],12)
plot(t,N,'o')

function dfun=fun1(t,N)
dfun=0.015*N;
```



解微分方程的Euler法:

- $\dfrac{dy}{dt}=f(t,y)$ 在 $t=t_0$ 点处原问题未知解的泰勒

级数展开式为

$$y(t)=y(t_0)+(t-t_0)y'(t_0)+\frac{(t-t_0)^2}{2}y''(t_0)+\cdots$$

近似的： $y(t)=y(t_0)+(t-t_0)f(t_0,y_0)$

于是上述公式对 $y(t_1)$ 的数值逼近为

$$y_1\approx y_0+hf(t_0,y_0),h=t_1-t_0$$

$y(t_1)$ 的逼近解可由点(t_0, y_0) 处的曲线 $y(t)$ 斜率进行外插得到

可以由同样的过程计算y_2, 即

$$y_2 \approx y_1 + hf(t_1, y_1)$$

依次类推，一般地，有

$$y_{k+1} \approx y_k + hf(t_k, y_k)$$

（迭代公式）

示例：使用欧拉法进行手工计算（了解即可）

$$\begin{cases} \dfrac{dy}{dt} = t - 2y \\ y(0) = 1 \end{cases}$$

来测试欧拉法。其**精确结果为**:

$$y=\frac{1}{4}[2t-1+5e^{2t}]$$

下表给出了在*h*=0.2时此问题求解的前几步手工计算结果，并且列出了精确值和误差。

			欧拉法	精确解	误差
k	t_k	$f(t_{k-1}, y_{k-1})$	y_k	y	y-
0	0.0	NA	1.0000	1.0000	0
1	0.2	-2.000	0.6000	0.6879	-0.0879
2	0.4	-1.000	0.4000	0.5117	-0.1117
3	0.6	-0.400	0.3200	0.4265	-0.1065

欧拉法的matlab实现（要求）

上例我们可以通过迭代公式编成代码，并使用循环反复求值。

```
y(1)=1                 %初始条件
for j=2:n
  y(j)=y(j-1)+h*(t(j-1)-2*y(j-1));
end
```

```
h = 0.1; tn = 1;
t=(0:h:tn)';
n=length(t);
y=1*ones(n,1);
for k=2:n
  y(k)=y(k-1)+h*feval(t(k-1),y(k-1));
end
plot(t, y, 'sb-')

function dfun=feval(t, y)
  dfun=t - 2*y;
end
```

实验：微分方程数值解（注意方程组和方程区别）

1、实验目的

了解微分方程初值问题的数学模型，掌握使用MATLAB求解常微分方程的数值方法，学会归纳不同参数和仿真之间的规律。

2、实验的基础

MATLAB求常微分方程初值问题

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

数值方法是先创建函数文件,用以描述微分方程右端二元函数,然后用ode23()求出数值解。

2、实验的基础

常微分方程组初值问题 $\begin{cases} \dfrac{d\vec{y}}{dt} = \vec{f}(t,\vec{y}) \quad t \geq t_0 \\ \vec{y}(t_0) = \vec{y}_0 \end{cases}$

一阶常微分方程组初值问题数值求解方法

[T, y] = ode23(‘ F ’, Tspan, y_0)

其中, F是函数文件,表示微分方程右端函数

Tspan = [t_0 Tfinal] ——求解区域;

y_0 ——初始条件

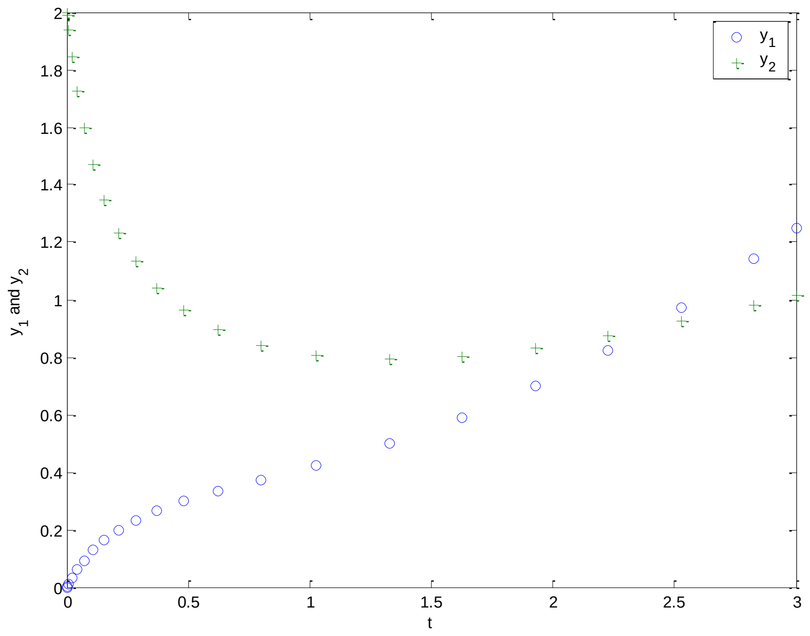
注: 函数F(t,y) 必须返回列向量.

数值解y 的每一列（每一列为一个函数的解）对应于列向量T中的数值解

微分方程数值解

示例：求解常微分方程组初值问题

$$\begin{cases} \dfrac{dy_1}{dt} = -e^{1-t}y_1 + 0.8y_2,\ y_1(0) = 0 \end{cases}$$



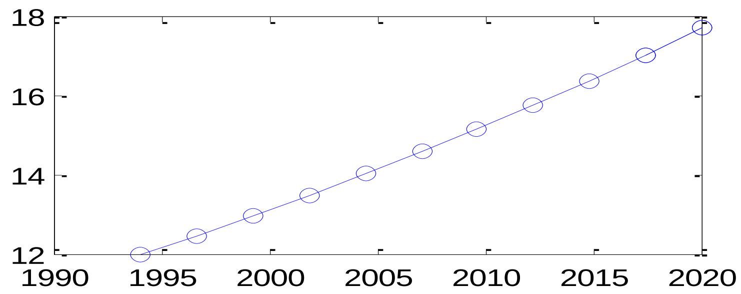

```
y0=[0;2];
tn=[0,3];
[t,y]=ode23(@fun2,tn,y0)
plot(t,y(:,1),'o',t,y(:,2),'+')
xlabel('t');ylabel('y_1 and y_2');
legend('y_1','y_2')

function dfun=fun2(t,y)
dfun=[-y(1)*exp(1-t)+0.8*y(2);
                    y(1)-y(2)^3];
end
```

例1: 马尔萨斯模型,以1994 年我国人口为12亿为初值，求解常微分方程．

$N(t)$表示人口数量,取人口变化率r =0.015,微分方程

$$\begin{cases} \dfrac{dN}{dt} = 0.015N \\ N(1994) = 12 \end{cases}$$



编辑窗口 ➔

```
function z=fun1(t,N)
z=0.015*N;
```

命令窗口 ➔

```
ode23('fun1',[1994,2020],12)
[T,N]=ode23('fun1',[1994,2020],12)
```

例2: 微分方程数值解实验：捕食者与被捕食者问题

一、实验内容

假设有足够多的青草供野兔享用，而狐狸仅以野兔为食物。x为野兔数量，y表示狐狸数量。假定在没有狐狸的情况下，野兔增长率为100%.如果没有野兔，狐狸将被饿死，死亡率为100%。狐狸和野兔相互作用的关系是，狐狸的存在使野兔受到威胁，且狐狸越多野兔增长受到阻碍越大，设野兔数量的负增长系数为0.015.而野兔的存在又为狐狸提供食物，设狐狸数量的增长与野兔的数量成正比，比例系数为0.01.数学模型为

$$\begin{cases} \dfrac{dx}{dt} = x - 0.015xy \\ \dfrac{dy}{dt} = -y + 0.01xy \end{cases} \qquad \begin{aligned} x(0) &= 100 \\ y(0) &= 20 \end{aligned}$$

计算$x(t)$，$y(t)$ 当$t\in[0，20]$时的数据。绘图并分析捕食者和被捕食者的数量变化规律。以x为横坐标，y为纵坐标绘制图形，分析生态循环的周期和衰减情况。

微分方程数值解

二、实验目的

了解微分方程初值问题的数学模型，掌握使用MATLAB求解常微分方程的数值方法，学会归纳不同参数和仿真之间的规律。

三、实验原理

创建MATLAB的函数文件，用于描述微分方程组右端项

```
function z=fox(t,y)
z(1,:)=y(1)-0.015*y(1).*y(2);
z(2,:)=-y(2)+0.01*y(1).*y(2);
```

使用MATLAB命令ode23()调用函数文件fox.m，求得向量Y(t)=[*x*(t),*y*(t)]函数的数值结果.Y的第一列数据是第一函数x(t)的数值，Y的第二列是第二函数*y*(t)的数值。利用数值结果绘出两个函数的图形，观察分析不同的初始值导致的不同数值结果，用以寻找规律。

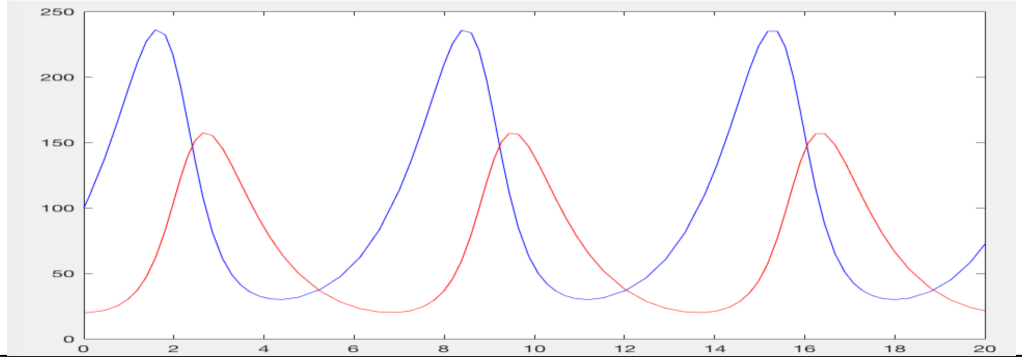
微分方程数值解

四、实验程序：求微分方程数值解并绘解函数图形

```
Y0=[100; 20];
[t,Y]=ode23('fox',[0,20],Y0);
x=Y(:,1);y=Y(:,2);
figure(1),plot(t,x,'b',t,y,'r')
figure(2),plot(x,y)
```

五、实验结果及分析

当初值取$x(0)$=100, $y(0)$=25时，仿真程序计算输出结果表明，野兔数量最小值和最大值分别为X_{min}=29.9517,X_{max}=236.0533；狐狸数量最小值和最大值分别为Y_{min}=20.0000,Y_{max}=157.2538。野兔数量和狐狸数量变化规律如图1所示



------兔子数量；------狐狸数量

例3：求解描述振荡器的经典Van der Pol微分方程.

$$\begin{cases} \dfrac{d^2y}{dt^2} - u(1-y^2)\dfrac{dy}{dt} + y = 0 \\ y(0) = 1;\ y'(0) = 0 \end{cases}$$

令x(1)=y； x(2)=dy/dt,
则：
dx(1)/dt=dy/dt=x(2)；
dx(2)/dt=d(dy/dt)/dt = d^2y/dt^2
= u*(1-x(1)^2)*x(2)-x(1).

```
   x0=[1;0];
   tn=[0,20];
   [t,y]=ode45(@vdpol,tn,x0);
   plot(t,y(:,1),t,y(:,2),'--')
   xlabel('t');ylabel('y_1 and y_2');
   legend('y_1','y_2')

function dfun=vdpol(t,x)
    u=2;
   dfun=[x(2);u*(1-x(1)^2)*x(2)-x(1)];
end
```

一、数值积分

对于积分 $I(f)=\int_a^b f(x)dx$

如果知道$f(x)$的原函数$F(x)$，则由Newton-Leibniz公式有

$$\int_a^b f(x)dx = F(x)\Big|_a^b = F(b)-F(a)$$

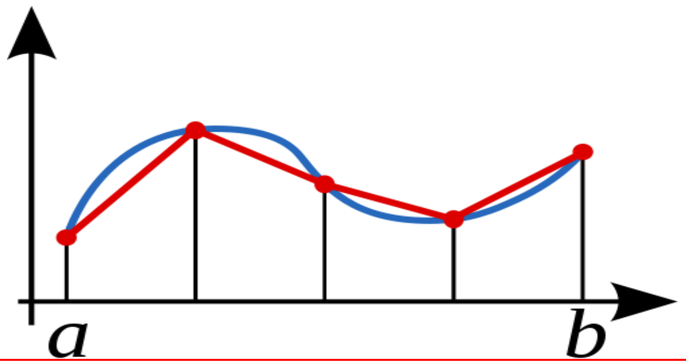
问题： （1）$f(x)$的解析式根本不存在,只给出了$f(x)$的一些数值;

（2）$f(x)$的原函数$F(x)$求不出来,如$F(x)$不是初等函数;

（3）$f(x)$的表达式结构复杂,求原函数较困难

数值积分的基本思想： 细分积分区间，以简单代复杂， 求小面积的和，得到定积分的近似值。

特点: (1)无需寻求原函数。(2)以简代繁。



二、MATLAB进行数值积分的主要函数

函数名	说明
trapz	梯形法求解积分
quad	基于变步长simpso法求积分
quadl	高精度Lobatto积分法
dblquad	矩形区域二重数值积分

函数quad的使用

基本语法： q = quad(fun,a,b,tol)

1. fun 被积函数文件名或函数句柄
2. a, b 积分下限,积分上限
3. tol 积分精度

示例：

$$\int_0^2 \frac{1}{x^3-2x-5}dx$$

编写求解程序：

```
function testmain_1
Q = quad(@myfun1,0,2)
%定义函数
function y = myfun1(x)
y = 1./(x.^3-2*x-5);
```

运行结果：

Q = -0.4605

数值积分

函数dblquad的使用

基本语法： q = dblquad(fun,a,b,c,d,tol)

1. fun 被积函数文件名或函数句柄
2. a, b 内积分下限,内积分上限
3. c, d 外积分下限，外积分上限
4. tol 积分精度

示例：

$$\int_0^1 dy \int_1^2 (x^2 + y)dx$$

编写求解程序：

```
function testmain_2

Q = dblquad(@myfun2,1,2,0,1)

%定义函数

function z = myfun2(x,y)

z = x.^2+y;
```

运行结果：

Q = 2.8333

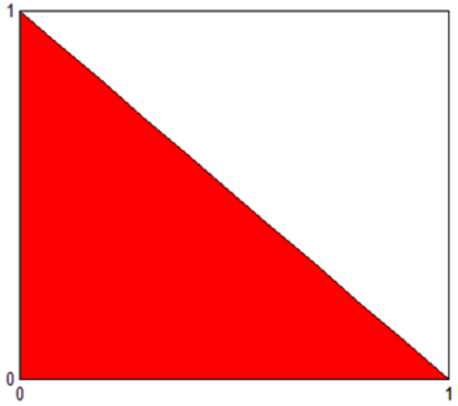
三、思考

(1) 上述例子的精确积分?

(2) 非矩形区域的二重积分?

$$\int_0^1 dx \int_0^{1-x} (x^2 + y)dy$$

```
fun=@(x,y) x.^2+y;
ymax=@(x) 1-x;
quad2d(fun,0,1,0,ymax)
ans=0.25
```



更多例子：

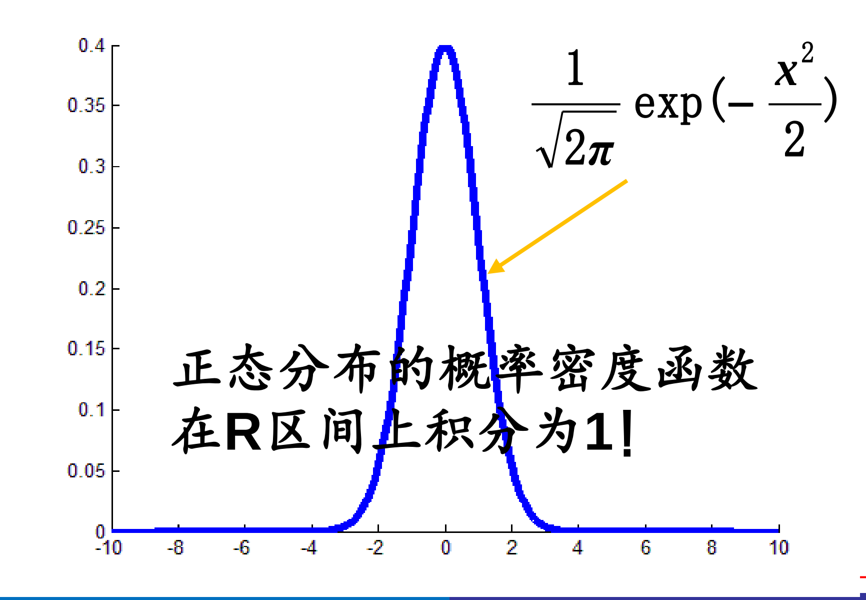
例1

$$\int_0^2 \frac{1}{x^3-2x-5}dx$$

```
syms x
f=1./(x.^3-2*x-5);
int(f,0,2)
```

Warning: Explicit integral
could not

```
function 
Q = quad((
%定义函数
function y
y = 1./(x
```



例2

$$\int_a^b \frac{1}{\sqrt{2\pi}}\exp(-\frac{x^2}{2})dx$$

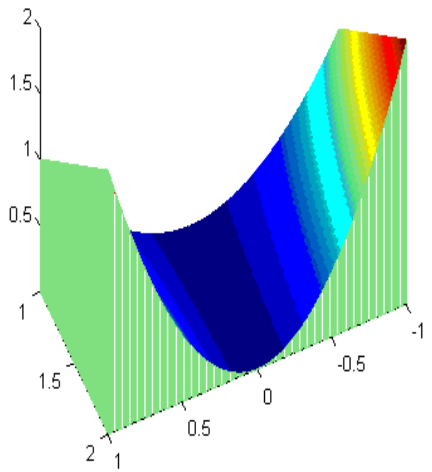
```
syms x;
f=1/(sqrt(2*pi))*exp(-x.^2/2);
int(f,-900,900)
```

➔原函数形式复杂

```
function testmain3
g=quad(@fun3,-900,900);
function z=fun3(x)
z=1/(sqrt(2*pi))*exp(-x.^2/2);
```

ans= 1

例4：曲顶柱体的体积 $\int_{-1}^{1}\left[\int_{1}^{2} x^2 y dy\right]dx$




```
function testmain4
q=dblquad(@fun4,-1,1,1,2)
function z=fun4(x,y)
z=x.^2.*y;
```

```
syms x y
f=x.^2.*y;
int(int(f,y,1,2),x,-1,1)
```

思考:

(1) 变上限积分的计算?

(2) 对于非矩形区域的二重积分如何计算?

补充材料(自学)

1. 求解微分方程组例子
2. 欧拉法解微分方程例子
3. 梯形积分trapz例子

示例：使用欧拉法求解如下问题

$$\begin{cases} \dfrac{dy}{dt} = t - 2y \\ y(0) = 1 \end{cases}$$

调用：

demoEuler(0.2)

看结果如何?

在步长h=0.2,h=0.1,h=0.05下，方程精确解与使用欧拉法得到的数值解的比较

```
function [t,y]=demoEuler(h)
tn=2;y0=1;
[t,y]=odeEuler('rhs',tn,h,y0);
yex=(2*t-1+5*exp(-2*t))/4;
fprintf(‘t y_Euler y_exact error\n');
for k=1:length(t)
fprintf('%9.4f %9.6f  %9.6f
%10.2e\n',t(k),y(k),yex(k),y(k)-yex(k))
end
fprintf('\nMax error = %10.2e  for
h=%f\n',norm(y-yex,inf),h);


function dydt=rhs(t,y)
dydt=t-2*y;
```

数值积分---应用实例

示例：梯形法 trapz

```
x=0:pi/10:pi;
y=sin(x);
trapz(x,y)
```

```
x=0:pi/100:pi;
y=sin(x);
trapz(x,y)
```

ans =
    1.9835

ans =
    1.9998

sin(x)在[0,pi]上积分的精确解：

```
syms x
int(sin(x),0,pi)
```

精确积分结果应该是2，对于可以进行符号积分求解精确结果的函数，使用数值方法意义并不大。

但是当被积函数不能进行符号积分时，使用数值积分方法可以快速得到比较可靠的计算结果。

学到了什么?

微分方程数值解：
微分方程简介、ODE、Euler法、实验

数值积分函数：
思想简介、quad积分、应用实例

补充材料