常微分方程（组）数值解

例：已知一阶常微分方程$\frac{dy}{dt}+2y=e^t, y(0)=\frac{1}{3}$，易得解析解$y(t)=\frac{1}{3}e^t$

数值解是通过将微分方程转化为差分方程，取一部分点进行计算，画出部分图像的一种计算机求解微分方程的方法，本例计算$t\epsilon[0,1]$上的图像，具体如下

首先，在$t\epsilon[0,1]$上等间隔Δt取点$0=t_1<t_2<\cdots<t_n=1$

其中$t_{i+1}=t_i+\Delta t, i=1,2,\cdots,n-1$

根据导数定义，t_i时刻的导数为$\left.\frac{dy}{dt}\right|_{t=t_i}=\lim\limits_{\Delta t\to 0}\frac{y(t_i+\Delta t)-y(t_i)}{\Delta t}$，当$\Delta t$取一个比较小的定值时，比如$\Delta t=0.001$，则$t_i$时刻的导数可近似为

$$\left.\frac{dy}{dt}\right|_{t=t_i}\approx\frac{y(t_i+\Delta t)-y(t_i)}{\Delta t}=\frac{y(t_{i+1})-y(t_i)}{\Delta t}$$

记t_i时刻的函数值$y(t_i)$为y_i，那么$\left.\frac{dy}{dt}\right|_{t=t_i}\approx\frac{y_{i+1}-y_i}{\Delta t}$，带入微分方程化简得到

$$y_{i+1}=(1-2\Delta t)y_i+\Delta te^{t_i}, i=1,2,\cdots,n-1$$

注意：e^t上也要带t_i以及化简的时候Δt一定不能漏

初始条件$y(0)=y(t_1)=y_1=\frac{1}{3}$

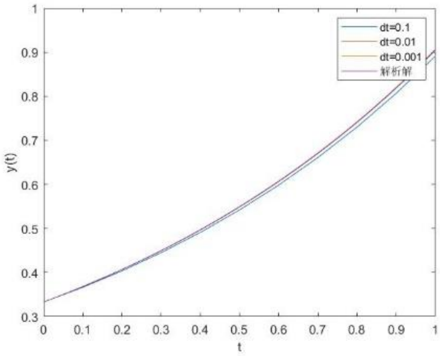
最终就是要求解差分方程组$\begin{cases} y_{i+1}=(1-2\Delta t)y_i+\Delta te^{t_i}, y_1=\frac{1}{3} \\ t_{i+1}=t_i+\Delta t, t_1=0, t_n=1 \end{cases}$

严格意义上是一个差分方程组，但是实际编程时，一般时间序列是人为给定的，所以真正要计算的其实就是关于y_i的那个差分方程。

在编程过程中，注意三个序列，以$\Delta t=0.1$为例

数值解递推过程

i	1	2	3	4	5	6	7	8	9	10	11
t_i	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
y_i	1										



一阶微分方程数值解与解析解对比

显然，当Δt逐渐缩小时，数值解更接近解析解，数值解和解析解误差在减小。

```
ode1.m
function [y,t]=ode1(ts,te,dt)
%ts是初始时刻,dt时间间隔,te为结束时刻
    t=ts:dt:te;%定义时间序列
    n=length(t);%计算时间序列的维数
    y=zeros(size(t));%初始化函数值向量
    y(1)=1/3;%赋初值
    for i=1:n-1
        y(i+1)=(1-2*dt)*y(i)+dt*exp(t(i));
        %差分方程别写错，dt不能漏，exp中带ti
    end
end
```

```
test_ode1.m
clear;close all;clc;
dt=[0.1 0.01 0.001];%3种时间步长
ts=0;te=1;%初始时刻和结束时刻
for i=1:3
    [y,t]=ode1(ts,te,dt(i));
    plot(t,y)
    hold on
end%循环画图
y2=1/3*exp(t);
plot(t,y2);%解析解
xlabel('t')
ylabel('y(t)')
legend('dt=0.1','dt=0.01','dt=0.001','解析解')
```

例：已知二阶常微分方程$\frac{d^2y}{dt^2}+2\frac{dy}{dt}+y=\cos t, y(0)=0, y'(0)=\frac{3}{2}$，易知解析解为$y(t)=te^{-t}+\frac{1}{2}\sin t$

对应二阶导数项，可近似为$\frac{d^2y}{dt^2}=\frac{y_{i+2}-2y_{i+1}+y_i}{(\Delta t)^2}$，然后带入化简。

注意：二阶微分方程已知两个初始条件$y(0), y'(0)$

那么$y(0)=y(t_1)=y_1$，但是$y'(0)=\frac{y(0+\Delta t)-y(0)}{\Delta t}=\frac{y_2-y_1}{\Delta t}$，所以$y_2=y_1+\Delta t y'(0)$，二阶微分方程对应二阶差分方程，同样初始条件需要 2 个。

后续编程和一阶方程类似，这里就不多介绍了，数模国赛中一般考一阶微分方程（组）居多。

对于二阶微分方程，可通过引入参量$x=\frac{dy}{dt}$，化为一阶微分方程组。

因为$\frac{dy}{dt}=x$，所以$\frac{d^2y}{dt^2}=\frac{d}{dt}\left(\frac{dy}{dt}\right)=\frac{dx}{dt}$，所以二阶微分方程可化为

$$\begin{cases} \dfrac{dx}{dt}+2x+y=\cos t \\ \dfrac{dy}{dt}=x \end{cases}$$

初始条件 $y(0)=0, x(0)=y'(0)=\dfrac{3}{2}$

对于常微分方程组，转化为差分方程组，则将导数项全部带入差分形式，即

$$\begin{cases} \dfrac{dx}{dt}=\dfrac{x_{i+1}-x_i}{\Delta t} \\ \dfrac{dy}{dt}=\dfrac{y_{i+1}-y_i}{\Delta t} \end{cases}$$

最终是化成一个一阶差分方程组。

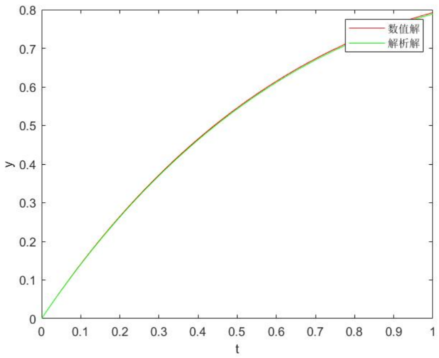
当然，针对线性微分方程组，可以写成矩阵微分方程的形式

记 $X=\begin{pmatrix} x \\ y \end{pmatrix}, \dfrac{dX}{dt}=\begin{pmatrix} \dfrac{dx}{dt} \\ \dfrac{dy}{dt} \end{pmatrix}, A=\begin{pmatrix} -2 & -1 \\ 1 & 0 \end{pmatrix}, U=\begin{pmatrix} \cos t \\ 0 \end{pmatrix}$，那么

$$\frac{dX}{dt}=AX+U$$

记 $X_i=\begin{pmatrix} x_i \\ y_i \end{pmatrix}$，则 $\dfrac{dX}{dt}=\dfrac{X_{i+1}-X_i}{\Delta t}$，即

$$X_{i+1}=(A\Delta t+E)X_i+U\Delta t$$



二阶微分方程（矩阵微分方程）解析解与数值解对比

ode2.m

```
clear;close all;clc;
dt=0.01;%时间步长
t=0:dt:1;%定义时间序列
n=length(t);%计算时间序列的维数
X=zeros(2,n);
%这里最好定义2行n列，矩阵形式计算是按列算的，每一个时刻对应一列
X(:,1)=[3/2;0];
A=[-2,-1;
    1 0];%微分方程矩阵
U=[cos(t);zeros(size(t))];
E=eye(2);%单位阵
for i=1:n-1
    X(:,i+1)=(A*dt+E)*X(:,i)+dt*U(:,i);
end
y=X(2,:);%数值解
y1=t.*exp(-t)+1/2*sin(t);%解析解
```

去年国赛 A 题波浪能最大输出功率设计就考察过，牛顿第二定律写出的是二阶运动学方程，引入的参变量就是速度，将两个二阶微分方程构成的方程组转化为 4 个一阶微分方程构成的方程组。

前面所使用的将 $\frac{dy}{dt} \approx \frac{y_{i+1} - y_i}{\Delta t}$ 的方法称为欧拉法，是最基本的微分方程数值解算法，除此之外，还有 45 阶龙格库塔法（matlab 内置函数 ode45）。

针对一阶常微分方程 $\frac{dy}{dt} = f(t,y)$

欧拉法：$y_{i+1} = y_i + \Delta t f(t_i, y_i)$

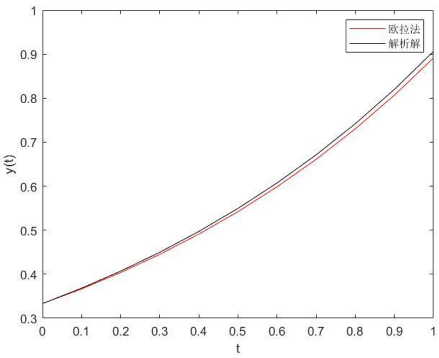
龙格库塔法：

$$\begin{cases} y_{i+1} = y_i + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = f(t_i, y_i) \\ k_2 = f\left(t_i + \frac{\Delta t}{2}, y_i + \frac{\Delta t}{2} k_1\right) \\ k_3 = f\left(t_i + \frac{\Delta t}{2}, y_i + \frac{\Delta t}{2} k_2\right) \\ k_4 = f(t_i + \Delta t, y_i + \Delta t k_3) \end{cases}$$

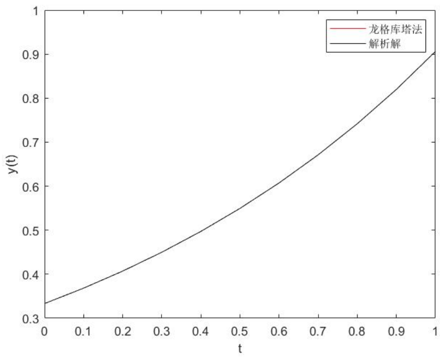
因为 $\frac{dy}{dt} = f(t,y)$，所以 k_1 就是该微分方程的解在 (t_i, y_i) 这一点的导数，k_2 是该微分方程的解在 $\left(t_i + \frac{\Delta t}{2}, y_i + \frac{\Delta t}{2} k_1\right)$ 这一点的导数，以此类推，将 4 个导数进行加权平均，不同于欧拉法直接将 $f(t_i, y_i)$ 作为导数。

例：已知一阶常微分方程 $\frac{dy}{dt} + 2y = e^t, y(0) = \frac{1}{3}$，易得解析解 $y(t) = \frac{1}{3} e^t$

比较欧拉法和龙格库塔法，$\Delta t = 0.1$



欧拉法与解析解比较



龙格库塔法与解析解比较

显然，龙格库塔法精度更高，matlab 中 ode45 函数就是用 45 阶龙格库塔法求解一阶微分方程数值解的。