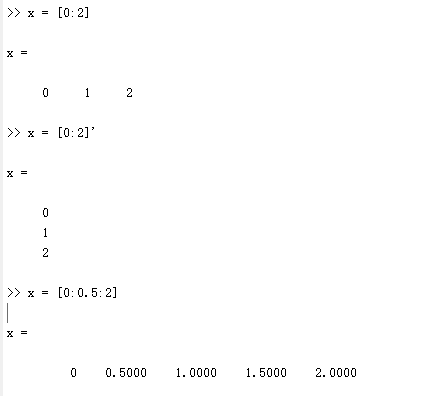
又开新坑了。。

**MATLAB学习笔记**

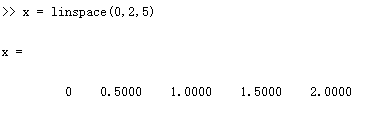
基本操作

**一.矩阵**

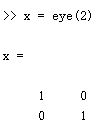
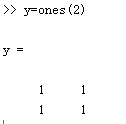
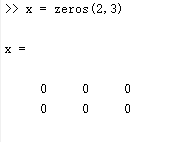
1.矩阵的定义



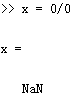
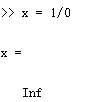
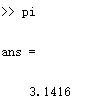
也可以用

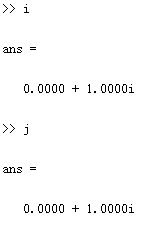


全0，全1，单位阵：

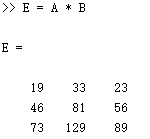
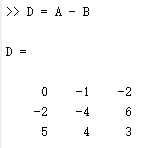
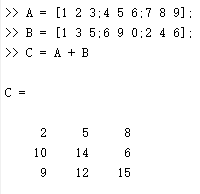


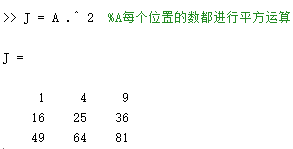
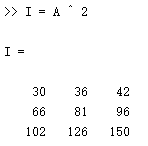
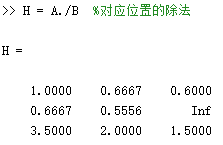
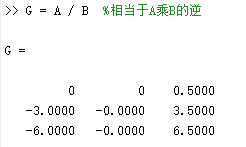
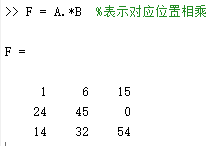
固定变量（pi，Inf，i，j，NaN等）





2.矩阵的运算

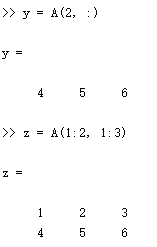
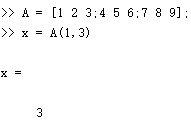




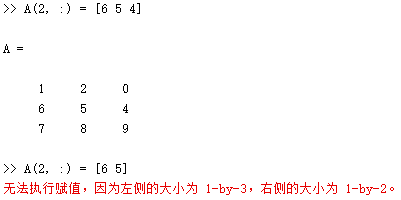
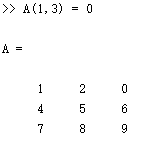
如果ax = b，则x = a\b是矩阵方程的解。

如果xa = b, 则x = b/a是矩阵方程的解。

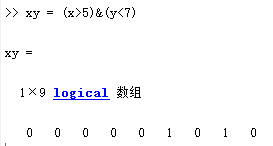
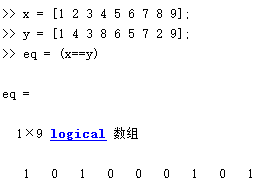
3.取矩阵特定的值

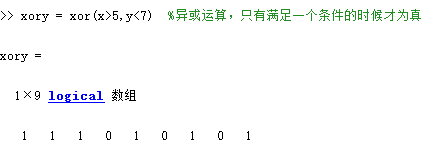
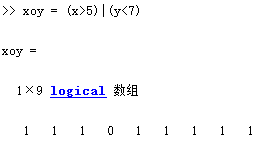


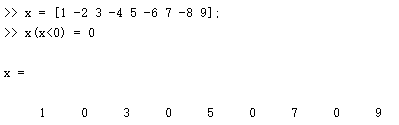
4.赋值

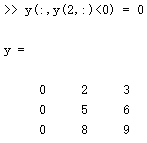
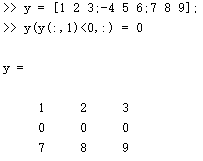


5.比较和逻辑运算



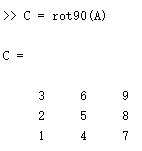
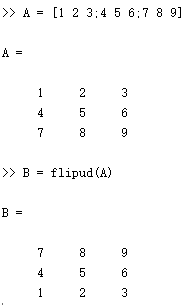




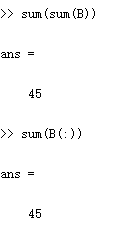
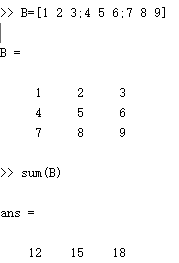
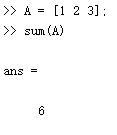


列向量y(:,num)；行向量x(num,:)

数组操作函数

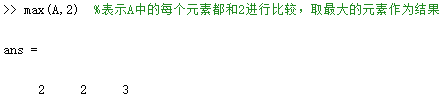
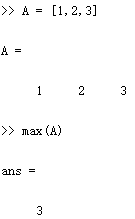


求和

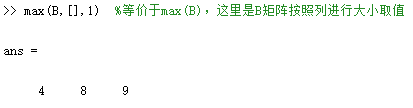


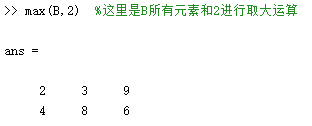
sum(B,1)是行相加，sum(B,2)是列相加，sum(sum(B))或者sum(B(:))可以求整个矩阵所有元素的和。

比大小

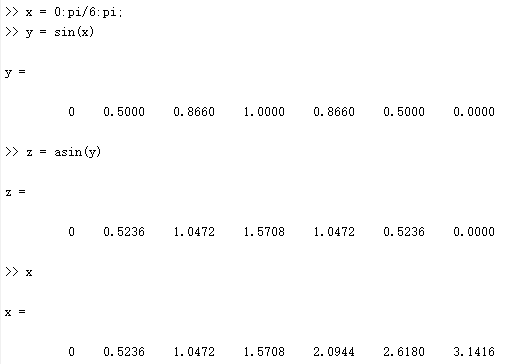


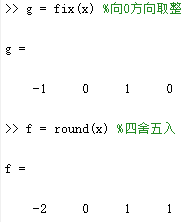
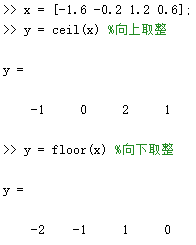
注意：  

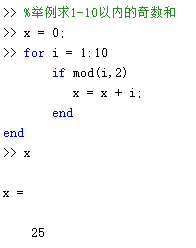


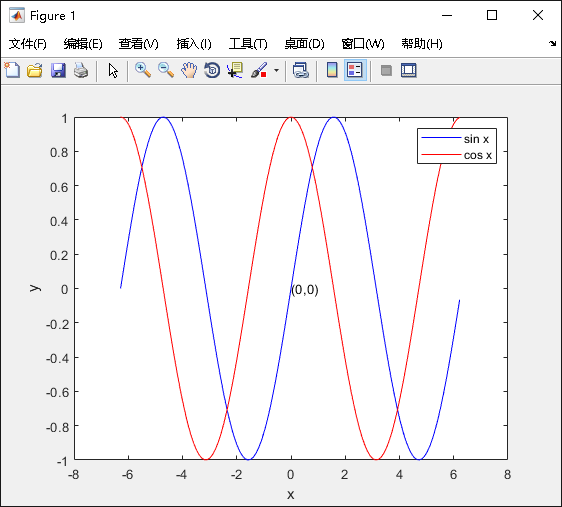
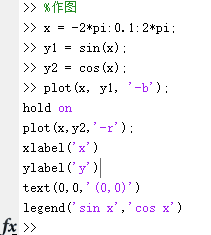
6.三角和反三角，绝对值，开方：

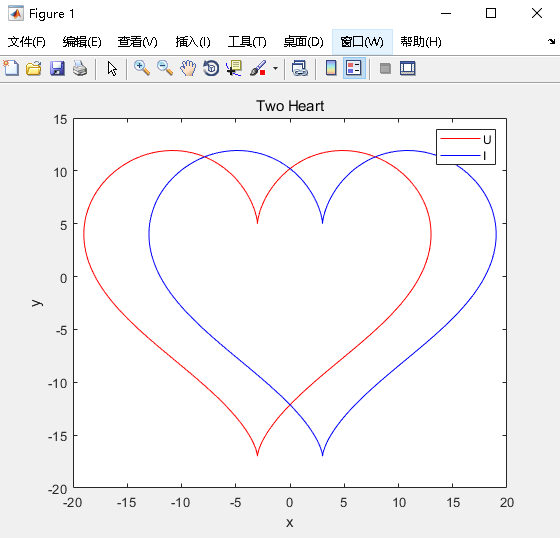
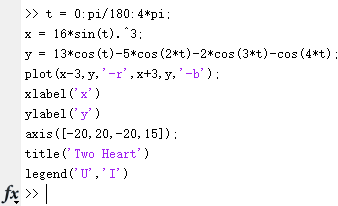




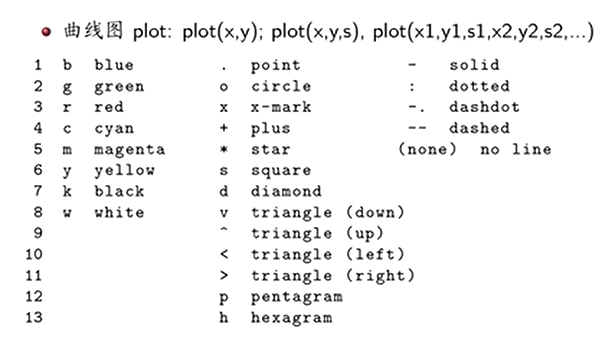
**二.基本语句**



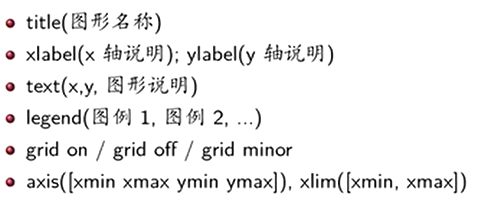




关于曲线图plot函数的一些参数说明：

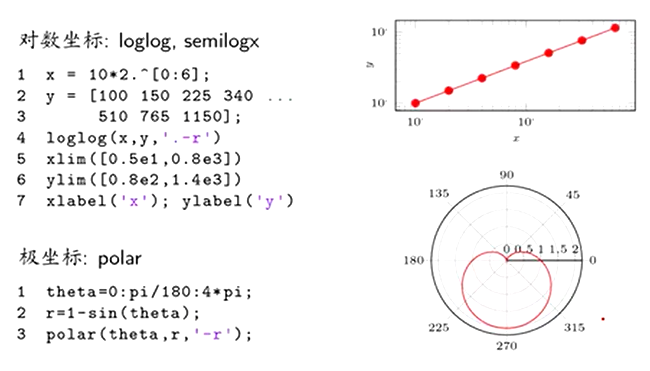


plot可以同时画多组曲线。



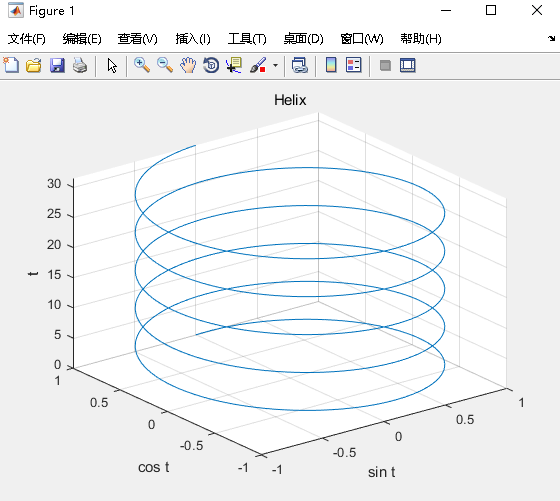
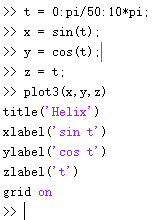
gird on打开网格，grid off关闭网格，grid minor绘制非常细的网格。

其他坐标系的绘图方法：

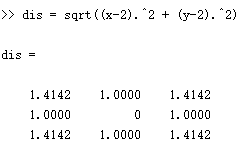
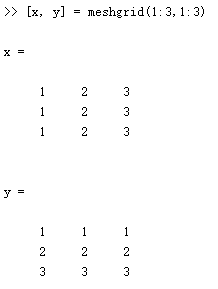


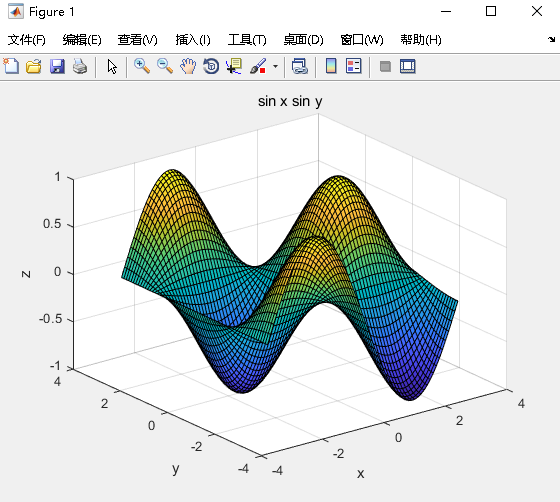
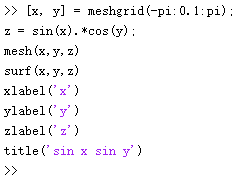
三维曲线作图：

只在plot后面加上3表示画三维图



meshgrid函数用来给出方网格的坐标





其中mesh和surf都是绘图，但surf相比可以绘出有颜色的曲面。

**三.函数**

function [output\_name 1,output\_name 2……] = function\_name(input\_name 1,input\_name 2……)

%commond of the funcution

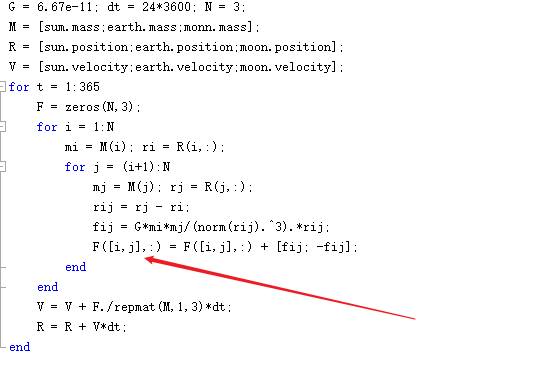
sentence 1;

sentence 2:

……

注意函数要在脚本里写，直接在命令行写是不能执行的。

下面举一个三个天体运动方程的例子：



红箭头的地方表示我看不是很懂233333

**四.MATLAB的文件操作**

1.文件路径

files = dir(‘文件名’)

之后通过files(3).name，files(3).date等可以查看文件名，创建日期等等。不从files(1)，files(2)开始是因为前两步是路径，只会显示.。

2.读取文本

数据读取：（注意如果有文本可以先在文本里用百分号注释掉）

data = load(‘文件名.txt’)

文本读取：

①.txt的文件读取

fid = fopen(‘文件名.txt’,’r’) %r是只读的意思

t1 = fgetl(fid) %t1用于获取第一行的内容，通过循环可以读取整个txt文件的所有文本

最后别忘了fclose(fid);

②..xls的文件读取（Excel）

data = xlsread(‘data.xls’,’Sheet1’,’A3:C5’) %1.文件名，2.Sheet 3.读取范围

Excel是空的部分系统会补上NaN，可以通过data(isnan(data)) = 0将NaN转换成0.

3.读取图片

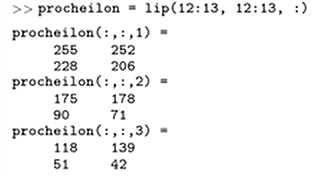
Yui = imreand(‘Yui.jpg’);

size(Yui) %输出像素，像素，三个像素值

image(Yui) %用于显示图片

part = Yui(242:266,255:279,:);

image(part); %显示图片上抠下来的部分图



procheilon用于获取像素点，第一层是R（红），第二层是G（绿），第三层是B（蓝）。

可以用image(procheilon)观察此图。对于黑白图只有一层，但如果有三层数值都相同也是黑白图。

4.读取视频

obj = VideoReader(‘Videos/……’) %路径+文件名

frame = read(obj,25);imshow(frame); %显示出图像的某一帧

如果打算对视频做处理，可以对视频的帧数做一个循环，每一帧进行同样的操作。

（对于视频里的声音处理这里到时候再补充23333）

文本操作：

①.txt文件的操作

text = regexp(t1,’,’,’split’) %表示在t1中用,将各个部分分开(split)

[n1,n2,n3] = text{:} %将分隔出来的东西赋给n1,n2,n3

字符串转化成数值函数：str2num(字符串名)

文本书写：

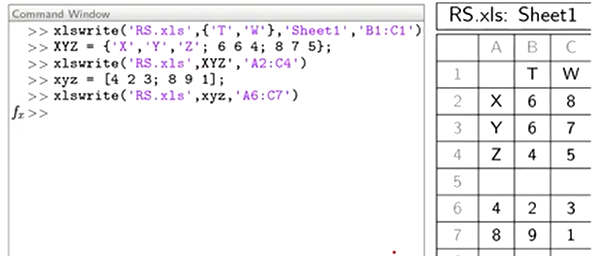
fid = fopen(‘文件名.txt’,’w’) %新建文本

fprintf(fid,’书写内容\n’)

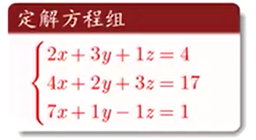
注意，matlab里面占位符也适用，常用占位符有%s字符，%d数字等等，占位符前面加-左对齐，加数字表示占位等等等等……这里不再赘述。

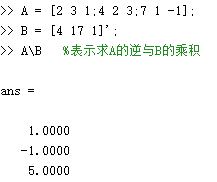
②..xls的文件书写（Excel）

xlswrit(‘文件名.xls’,{‘书写内容1’,’书写内容2’}，‘Sheet1’,’B1:C1’)

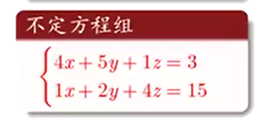


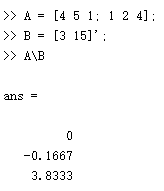
**五.线性方程组求解**



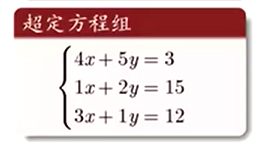


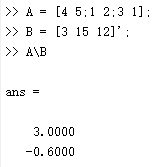
这里之所以可以解方程，是因为A\*X = B，经过变换以后X = A‘\*B。所以只需要计算A\B，结果就是方程的解。

’ 



对于不定方程组，软件只会给出一组解，不过是对于某一变量是0的情况，比如这里x是0.





对于无解情况，软件对x,y运用最小二乘法求解。

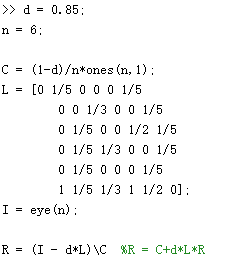
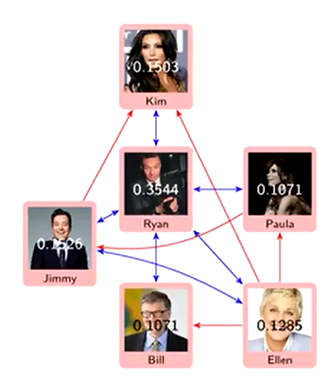
实例：PageRank算法：

指标形式：

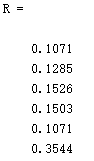
矩阵形式：

d是一个概率值，在这里一般是0.85。

举个例子：

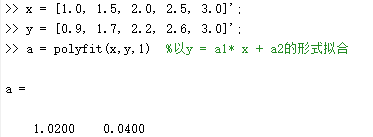


结果：



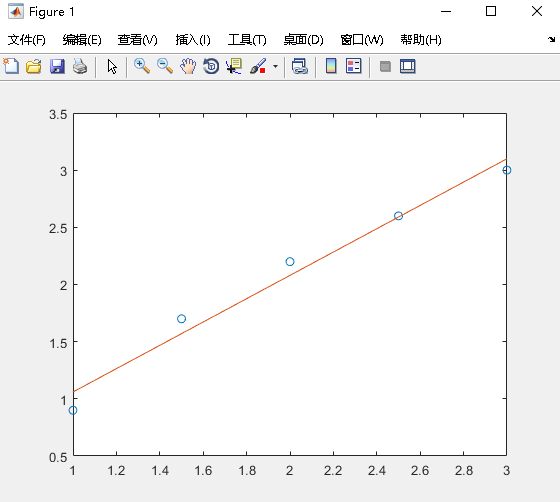
**六.简单建模方法介绍**

1.拟合

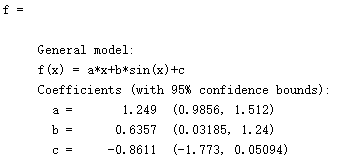


拟合之后的绘图：

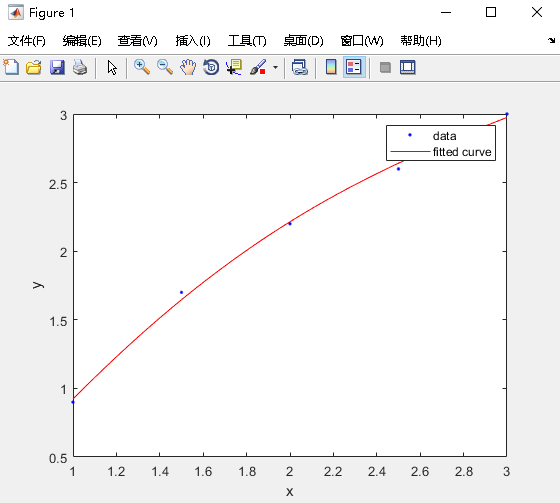




以其他形式拟合：  





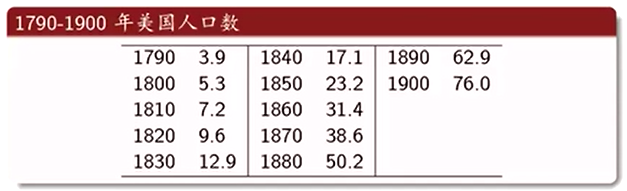
举例：指数增长模型

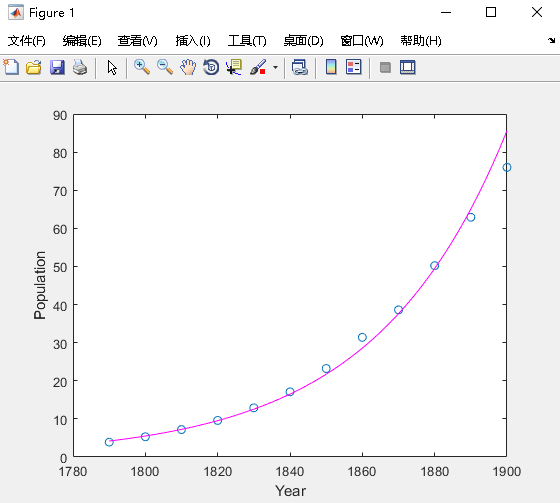
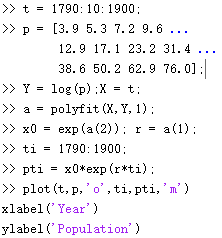
x(t) = x0e^rt

lnx(t) = rt + lnx0

Y = a1t + a2 %a1就是r，a2就是lnx0

人口模型见姜启源书上1790-1900年间的统计数据





2.插值和线性回归：

线性回归regress

插值interp

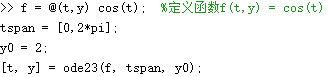
[B,Bint,R,Rint,Stats] = regress[X,Y]

（）

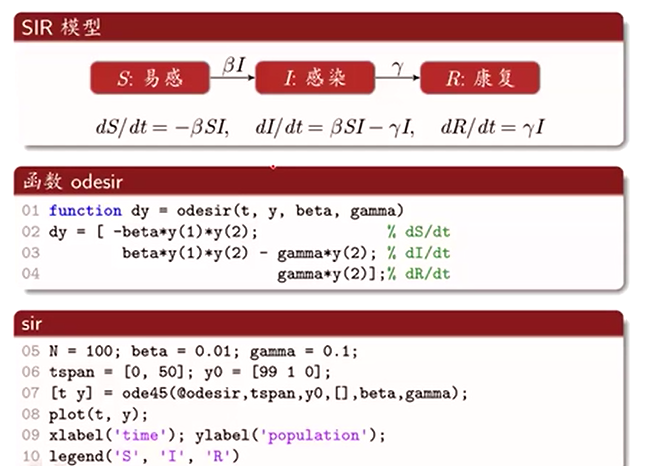
3..Runge-Kutta法求解微分方程

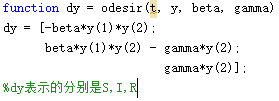
一阶微分方程：

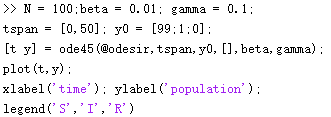
y’ = cost

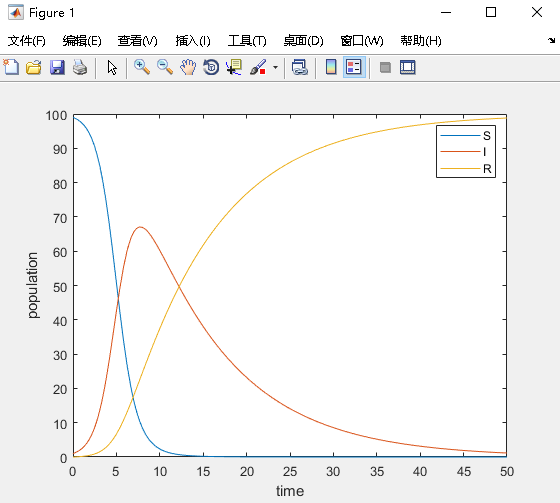


举例：传染病模型









此模型也适用于新闻传播一类

**九.启发式算法**

启发式算法是基于直观或经验的局部优化算法，不保证一定是最优解。

1.禁忌搜索（Tabu Search）：对局部领域搜索的一种扩展，是一种全局逐步寻优算法，是对人类智力过程的一种模拟。

2.模拟退火（Simulated Annealing）：通过模拟物理退火过程搜索最优解的方法。

3.遗传算法（Genetic Algorithms）：通过模拟自然进化过程搜索的最优解的方法。

4.神经网络（Neural Networks）：模仿动物神经网络行为特征，进行分布式并行信息处理的算法数学模型。

5.蚁群算法（Ant Algorith）：模仿蚂蚁在寻找食物过程中发现路径的行为来寻找最优化路径的几率型算法。

模拟退火：



初始解的生成：

1.通常以一个随机解作为初始解，并保证理论上能够生成解空间的任意解。

2.也可以是一个荆条选过的较好的解，这种情况下，初始温度应当设置的较低。

3.初始解不宜太好，否则很难从这个解的邻域中跳出。

邻解生成函数：

1.邻解生成函数应尽可能保证产生的候选解能够遍布解空间。

2.邻域应尽可能的小：能够在少量循环步中允分探测，但每次的改变不应引起太大的变化。

等温步数的确定：

（Metropolis抽样稳定准则）

降温法：

1.经典模拟退火算法T(t) = T0/log(1+t)

2.快速模拟退火算法T(t) = T0/1+t

3.常用的模拟退火算法降温方式T(t+dt) = aT(t) (0.8<=a<=0.99)

花费函数COST（不要太复杂）

一般又目标函数来构造

终止条件：

1.理论上温度要降为0时才终止退火算法，但实际上温度较低时，尝试的接受概率就几乎为0了。

2.设置终止温度的阈值，或设置外循环迭代次数。

3.算法搜索到的最优值连续若干步保持不变。

模拟退火算法实例：

旅行商问题（TSP）

Simulated\_Annealing.m主程序

route = randperm(numberofcities); %randperm±íÊ¾¶ÔÀ¨ºÅÄÚµÄÊýÉú³ÉÒ»¸öËæ»úÅÅÐò

temperature = 1000; cooling\_rate = 0.95;

Titerations = 1;

previous\_distance = totaldistance(route); %¼ÆËãÂ·¾¶×Ü³¤

while temperature > 1.0

temp\_route = perturb(route, 'reverse'); %perturbÊÇ²úÉúÁÚ½âµÄº¯Êý£¬ÒÔ·´×ªµÄ·½Ê½½øÐÐÈÅ¶¯£¬Éú³ÉÒ»¸öÐÂµÄ½â

current\_distance = totaldistance(temp\_route); %Â·³¤

diff = current\_distance - previous\_distance;

if(diff<0)||(rand < exp(-diff/(temperature))) %rand±íÊ¾Ëæ»úÉú³É´Ó0µ½1µÄÊý

route = temp\_route; %½ÓÊÕµ±Ç°½â

previous\_distance = current\_distance;

Titerations = Titerations + 1;

end

if Titeration >= 10

temperature = collong\_rate\*temperature;

Titeration = 0;

end

distancematrix.m程序

function dis = distancematrix(city) %cityÓÐ34\*2¸öÔªËØµÄ½á¹¹Ìå

numberofcities = length(city); %lengthÓÃÓÚ·µ»Ø¾ØÕóÖÐÐÐºÍÁÐ×î´óµÄÊý

R = 6378.137; %µØÇò°ë¾¶£¬ÓÃÓÚÇóÁ½¸ö³ÇÊÐµÄÇòÃæ¾àÀë

for i = 1:numberofcities

for j = i+1:numberofcities

dis = zeros(34,2);

dis(i, j) = distance(city(i).lat, city(i).long,...

city(j).lat, city(j).long, R);

%distanceº¯ÊýÓÃÓÚÇóÇòÃæÉÏÁ½µãµÄ¾àÀë£¬Ê×ÏÈÒª¸ø³öÇòÃæÉÏÁ½µãµÄ¾­Î³¶È£¬ÔÙ¸ø³öÇòÃæµÄ°ë¾¶

dis(j, i) = dis(i, j);

end

end

totaldistance.m程序

function d = totaldistance(dis, route)

d = dis(route(end),route(1)); %Ê×Î²Á½¸ö³ÇÊÐµÄ¾àÀë

for k = 1:length(route)-1

i = route(k);

j = route(k+1);

d = d + dis(i,j);

end

perturb.m程序

function route = perturb(route\_old, method)

route = route\_old;

numbercities = length(route);

city1 = ceil(numbercities\*rand); %0µ½34Ö®¼äµÄËæ»úÊý£¬²¢È¡Õû

city2 = ceil(numbercities\*rand); %Í¬ÉÏ

switch method

case 'reverse' %reverse·¨»»Á½¸öÊý×ÖºÍÖ®¼äµÄÄÚÈÝ

cmin = min(city1, city2);

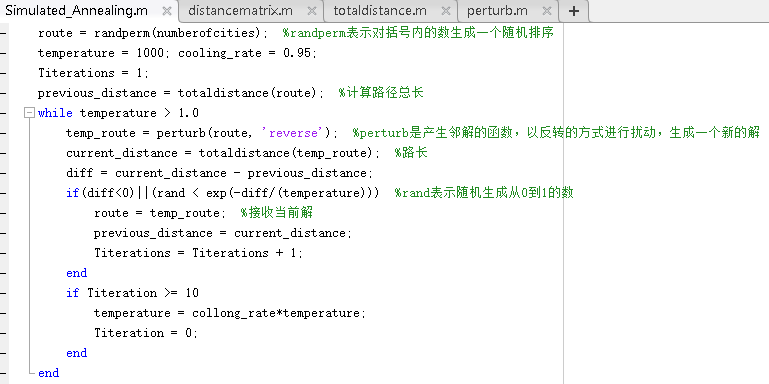
cmax = max(city1, city2);

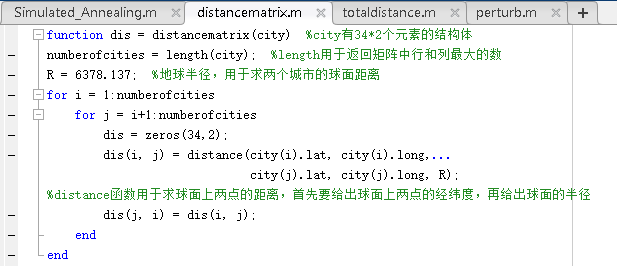
route(cmin:cmax) = route(cmax:-1:cmin);

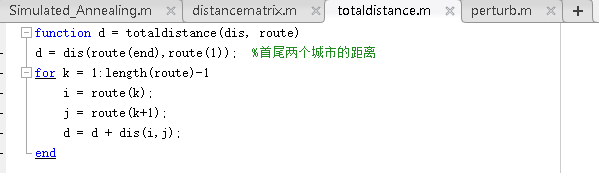
case 'swap' %swapÖ»»»Á½¸öÊý×Ö

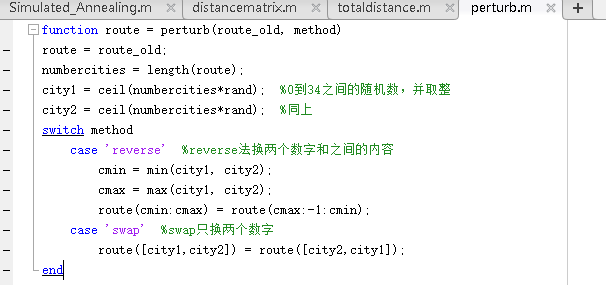
route([city1,city2]) = route([city2,city1]);

end









遗传算法：

遗传算法的基本思想（伪代码）：

set initial generation = 0 %第0代

probability of mutation = α %变异概率

probability of performing crossover = β %杂交概率

construct a population of n randomly-generated individuals Pk; %种群

while not termination do:

evaluate: compute fitmess(i) for each individuals in Pk

%评价函数，个体适应度

select: select m members of Pk insert into Pk+1

%挑出适应度高的加到下一代

crossover: produce αm children by crossover and insert into Pk+1

%杂交

mutate: produce βm children by mutate and insert into Pk+1

%变异

update generation: k = k + 1

end while

return the fittest individual from Plast

编码：

对解进行编码，分为以下四种：

1.二进制编码

2.格雷编码



3.实数编码

4.符号编码

适应度函数（评价函数）：

通常由目标函数改造而得到。

选择：

轮盘赌：又称比例选择算子，个体i被选中的概率pi与其适应度成正比。

两两竞争：从父代中随机地选取两个个体，比较适应值，保存优秀个体，淘汰较差的个体。

排序选择：根据各个体的适应度大小进行排序，然后基于所排序号进行选择。

交叉：

相当于基因染色体部分互换

单点交叉

两点交叉

变异：

相当于某一点的数值发生改变

单点变异

换位变异

举例：TSP问题算法求解

Genetic\_Algorithm.m文件

popSize = 100; %ÖÖÈº¹æÄ£

max\_generation = 1000; %³õÊ¼»¯×î´óÖÖÈº´úÊý

Pmutation = 0.16; %±äÒì¸ÅÂÊ

for i = 1:popSize %³õÊ¼»¯ÖÖÈº

pop(1,:) = randperm(numberofcities); %Éú³É¶ÔnumberofcitiesÊýÁ¿µÄpopSize¸öËæ»úÅÅÐò

end

for generation = 1:max\_generation %Ö÷Ñ­»·¿ªÊ¼

fitness = 1/totaldistance(pop,dis);%¼ÆËã¾àÀë£¨ÊÊÓ¦¶È£©£¬ÒòÎª¾àÀëÔ½Ð¡Ô½ºÃËùÒÔÕâÀï¶¨ÒåÊÊÓ¦¶ÈÊÇ¾àÀëµÄµ¹Êý

[maxfit, bestID] = max(fitness);

bestPop = pop(bestID,:);

pop = select(pop, fitness, popSize, 'competition'); %Ñ¡Ôñ£¬ÕâÀïÑ¡ÔñÁ½Á½¾ºÕù

pop = crossover(pop); %ÔÓ½»£¨½»²æ£©

pop = mutate(pop,Pmutation); %±äÒì

pop = [bestPop; pop]; %¾«Ó¢±£»¤

end

popDist = total\_distance(pop,dis); %¼ÆËã¾àÀë

[minDist, index] = min(popDist); %ÕÒ³ö×î¶Ì¾àÀë

optRoute = pop(index,:); %ÕÒ³ö×î¶Ì¾àÀë¶ÔÓ¦µÄÂ·¾¶

select.m文件

function popselect = select(pop, fitness, nselect, method)

popSize = size(pop,1); %sizeº¯ÊýÓÃÓÚµÃµ½¾ØÕóµÄÐÐºÍÁÐ£¬1ÓÃÓÚµÃµ½ÐÐ£¬2ÓÃÓÚµÃµ½ÁÐ

switch method

case 'competition'

i1 = ceil(popSize\*rand(1,nselect));

i2 = ceil(popSize\*rand(1,nselect));

I = i1.\*(fitness(i1) >= fitness(i2)) + ...

i2.\*(fitness(i1) < itness(i2));

case 'roulette'

p = fitness/sum(fitness);

cump = cumsum(p);

%ÀûÓÃ²åÖµ£ºyi = ÏßÐÔ²åÖµ(x, y, xi)

I = interp1([0 cump],1:(popSize+1),...

rand(1,nselect),'linear');

I = floor(I);

end

popselect = pop(I);

crossover.m文件

function children = crossover(parents)

[popSize, numberofcities] = size(parents);

children = parents;

for i = 1:2:popSize

parent1 = parents(i+0,:); child1 = parent1;

parent2 = parents(i+1,:); child2 = parent2;

InsertPoints = ceil(numberofcities\*rand(1,2)); %Ñ¡³öÁ½¸ö½»²æµã

for j = min(InsertPoints):max(InsertPoints)

if parent1(j)~=parent2(j) %Èç¹û¶ÔÓ¦Î»ÖÃ²»ÖØ¸´

child1(child1==parent2(j)) = child1(j);

child1(j) = child2(j);

child2(child2==parent1(j)) = child2(j);

child2(j) = child1(j);

end

end

children(i+0,:) = child1;

children(i+1,:) = child2;

end

mutation.m文件

function children = mutation(parents, probmutation)

[popSize, numberofcities] = size(parents);

children = parents;

for k=1:popSize

if rand<probmutation %ÒÔÒ»¶¨¸ÅÂÊ±äÒì

InsertPoints = ceil(numberofcities\*rand(1,2));

I = min(InsertPoints);

J = max(InsertPoints);

switch ceil(rand\*4) %swap,slide,flip

case 1 %µ¥´¿»»Î»

children(k,[I,J]) = parents(k,[J I]);

case 2 %Ò»¸öÊýÓëÒ»¸öÇø¼ä»»Î»

children(k,I:J) = paarents(k,[I+1:J I]);

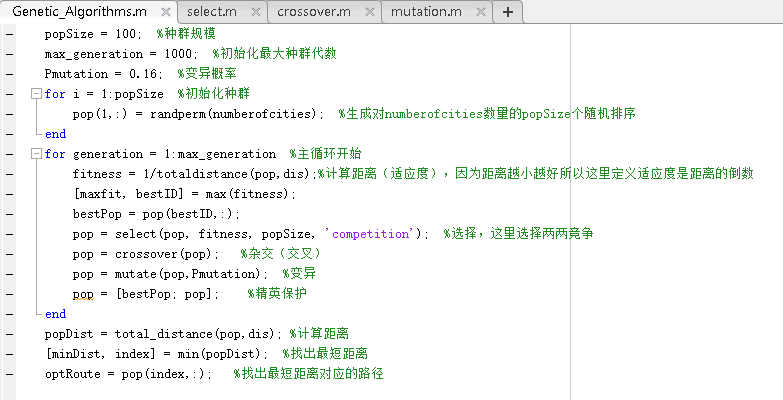
otherwise %Ò»¸öÊýÓëÒ»¸öÇø¼äµ¹¹ýÀ´»»Î»

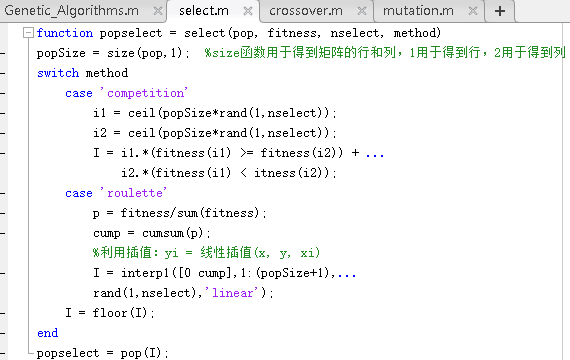
children(k,I:J) = parents(k,J:-1:I);

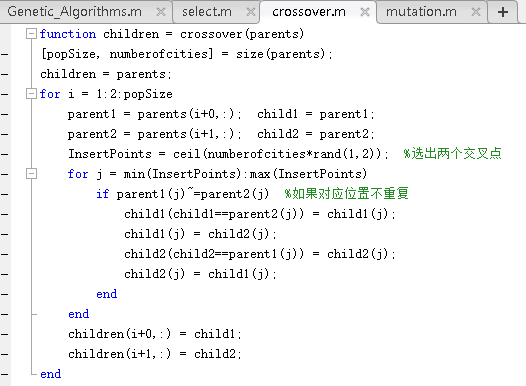
end

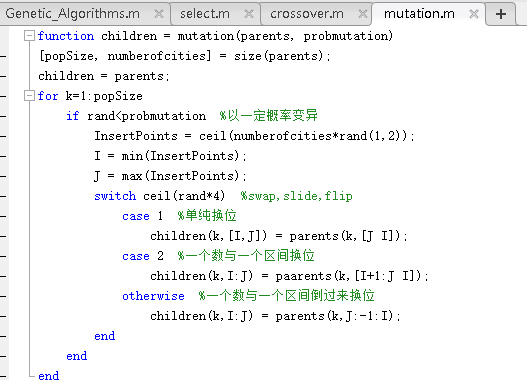
end

end









模拟退火算法和遗传算法可以用来解决同样类型的问题。

元胞自动机：

本质：离散的动力学系统

解释：定义在有限时间和空间，并且元胞的状态有限，而且元胞自动机的举止行为具有动力学特征。

一维元胞自动机有256种规则

规则：状态转移函数

举例1：森林大火的元胞自动机程序

n = 300;

Plight = .000005; Pgrowth = .01;

UL = [n 1:n-1]; DR = [2:n 1];

veg = zeros(n,n);

imh = image(cat(3,veg,veg,veg)); %¿ÉÊÓ»¯±íÊ¾É­ÁÖµÄ¾ØÕó

% veg = {empty = 0 burning = 1 green = 2}

for i = 1:3000

sum = (veg(UL,:)==1) + (veg(:,UL)==1) + (veg(:,DR)==1) + (veg(DR,:)==1);

%¸ù¾Ý¹æÔò¸üÐÂÉ­ÁÖ¾ØÕó£ºÊ÷ = Ê÷ - ×Å»ðµÄÊ÷ + ÐÂÉúµÄÊ÷

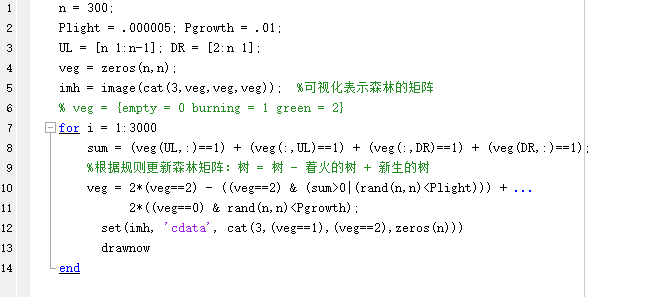
veg = 2\*(veg==2) - ((veg==2) & (sum>0|(rand(n,n)<Plight))) + ...

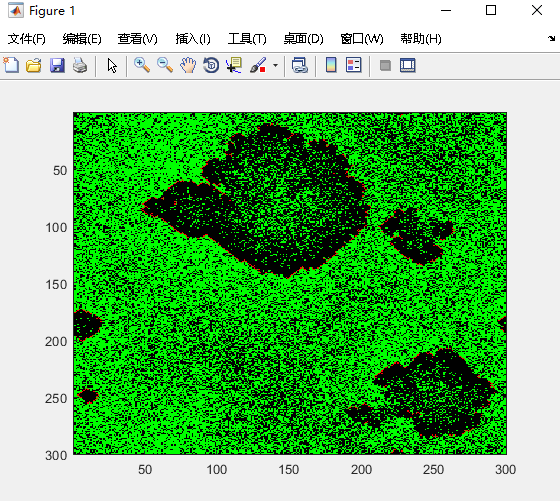
2\*((veg==0) & rand(n,n)<Pgrowth);

set(imh, 'cdata', cat(3,(veg==1),(veg==2),zeros(n)))

drawnow

end





举例2：交通问题

单车道交通模拟：

flux函数得到平均流量（主函数）

function flux = ns(rho,p,L,tmax) %rho = 0.2;p = 0.2;L = 72;tmax = 72

ncar = round(L\*rho);

x = sort(randperm(L,ncar)); %1µ½LÖÐËæ»ú³é³öncar¸öÊý£¬²¢ÅÅÐò

vmax = 5;

v = vmax \* ones(1, ncar); %³õÊ¼»¯ËùÓÐ³µËÙ¶ÈÎªvmax

for t = 1:tmax

v = min(v+1, vmax); %¼ÓËÙ¹æÔò

gaps = gaplength(x,L,ncar);

v = min(v, gaps-1); %·ÀÖ¹Åö×²

v = max(v - (rand(1,ncar)<p),0); %Ëæ»ú¼õËÙ binornd

x = x + v; %Î»ÖÃ¸üÐÂ

x(x>L) = x(x>L) - L;

flux = flux + sum(v)/L; %Æ½¾ùÁ÷Á¿

end

flux = flux / tmax; %Ê±¼äÆ½¾ù

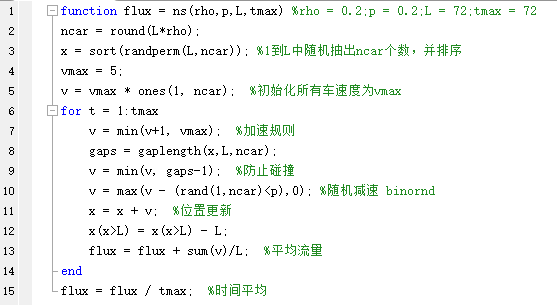
gaps函数用于计算两车之间的距离

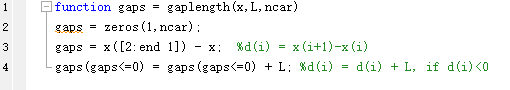
function gaps = gaplength(x,L,ncar)

gaps = zeros(1,ncar);

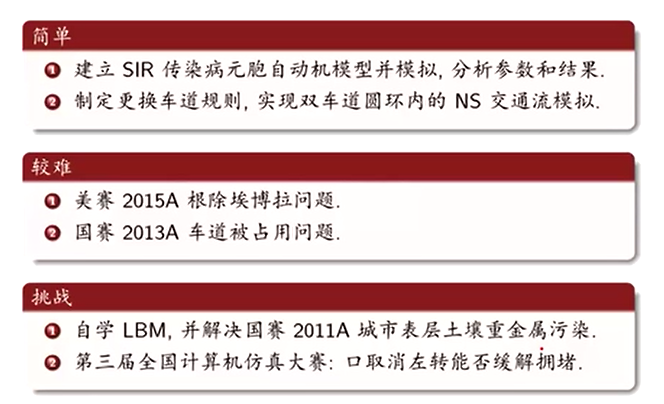
gaps = x([2:end 1]) - x; %d(i) = x(i+1)-x(i)

gaps(gaps<=0) = gaps(gaps<=0) + L; %d(i) = d(i) + L, if d(i)<0





exercises：



评价模型：

1.层次分析法

2.模糊综合评判

引入：设头发数量n = 1，为秃子

若n = k为秃子，则n = k + 1也是秃子

评价要素： 因素集U = {u1,u2,…}

评价集V

预测模型：

1.拟合

polyfit/fit拟合函数

时间序列：预测对象按照时间顺序排列而成的序列。

时序预测：根据时序过去变化的规律，推测今后趋势。

时间序列的变化形式：

长期趋势变动Tt 季节变动St 循环变动Ct 不规则变动Rt

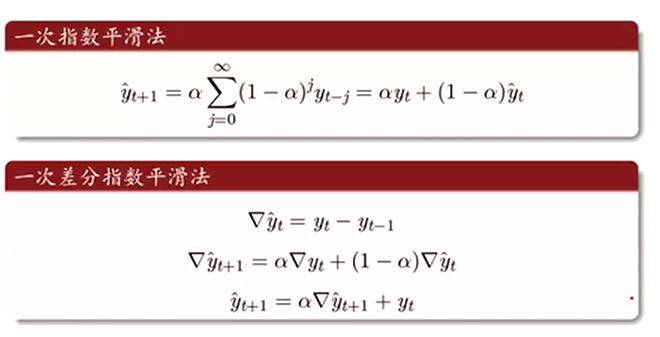
模型：加法模型，乘法模型，混合模型。

移动平均法：适合平稳增长或下降的数据。

例如已知9个月的数据去预测第10个月的，取定N = 4，每隔4个月求一次均值去预测下一个月；利用公式

S = norm(yhat(1:end-1) – y(N+1:end))/sqrt(T - N)可以求标准差。

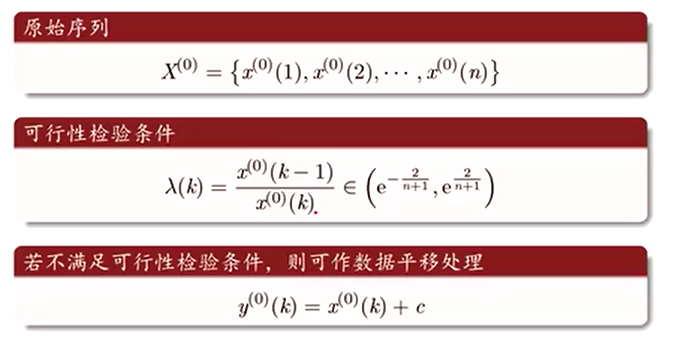
其中norm是用于求向量范数，例如A = [3 4] ,norm(A) = 5.

平滑法：

灰色预测：

GM(1,1)预测模型：模型是1阶微分方程，且只含一个变量。

1.得到原始序列



2.对数据进行可行性检验

3.对数据进行修正

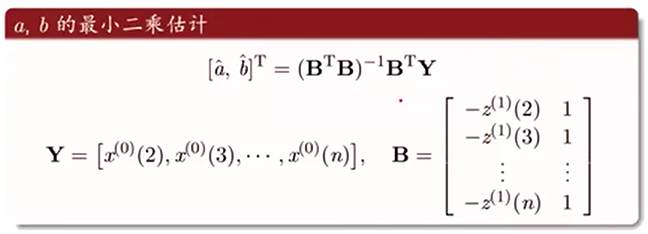
对于n组数据，我们需要进行n-1次可行性检验。若不满足可行性检验则需对数据进行平移处理。

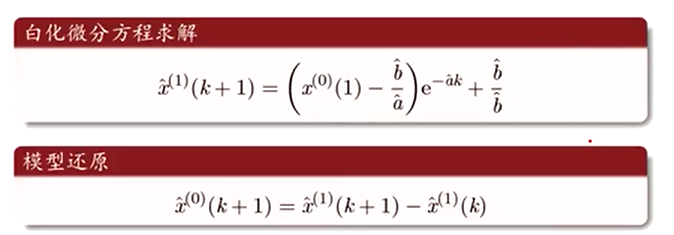
4.得到依次累加生成序列X(1)

5.得到均值生成序列Z(1) z(1)(k) = x(1)(k)/2+x(1)(k-1)/2

6.灰微分方程：x(0)(k) + az(1)(k) = b, k = 2,3,…,n

7.白化微分方程：dx(1)/dt + ax(1)(t) = b，a和b在灰微分方程基础上用最小二乘法进行估计。





残差检验：e(k) < 0.2 ,rho(k) < 0.2

