

课程说明 & Matlab 快速入门

十天美赛高强度培训

周吕文

超级数学建模

2017 年 12 月 26 日



扫码听课

Part I

课程说明

课程内容

内容列表

- ① **Matlab 快速入门**：矩阵数组运算、常用函数、控制结构、作图、数据读写和简单应用。
- ② **Latex 排版 x 2**：格式定制、数学公式、图表、文献引用。
- ③ **美赛优秀论文精析 x 6**：问题、模型、算法、程序和写作。
- ④ **作业点评 & 比赛注意事项**：Matlab/Latex 作业点评、资料检索等注意事项。

课程资料

- 幻灯片、讲议、程序、中英论文。

课程特色

问题

- 全部选用 MCM/ICM 真题，分析赛题和解题思路。

建模

- 涵盖美赛常用模型，详细讲解模型和算法原理。

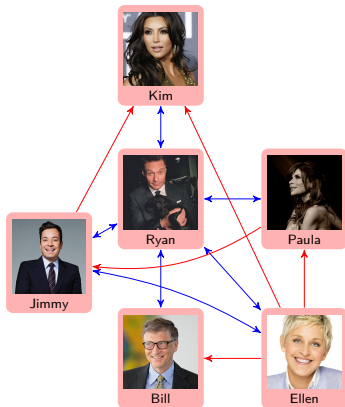
编程

- 拓展编程知识，逐句讲解特等奖论文模型和算法实现程序。

写作

- 提供特等奖原文和中英对照翻译，分析论文、摘要写作特点。

类似或相关的评价排序问题



- 2014ICM 赛题 C: 使用网络模型来评价学术的影响力.
- 2010MCM 赛题 B: 连环犯罪的地理轮廓分析.
- 1996MCM 赛题 B: 竞赛评判问题.
- 高校科研和教学水平排名.
- 电视台真人选秀节目中参赛选手的评价和排名.
- 谷歌等搜索引擎的网页排序 pagerank 算法.
- 微博/社交网络上的公众人物影响力.

are approximately 4 meters deep.

Assume a stuntman falls from 30 meters above the airbag. Gravity accelerates him from rest to a velocity v . At this point he strikes the airbag and is decelerated completely, so we have

$$\sqrt{2gd_{\text{fall}}} = \sqrt{2a_{\text{bag}}h_{\text{bag}}}$$

where d_{fall} is the fall distance, a_{bag} is the deceleration rate the stuntman experiences in the airbag, h_{bag} is the height of the airbag, and g is the acceleration due to gravity. Thus

$$a_{\text{bag}} = \frac{d_{\text{fall}}}{h_{\text{bag}}}g = \frac{30\text{m}}{4\text{m}}g = 7.5g$$

We therefore conclude:

- **When using an airbag, the stuntman experiences an average acceleration of at most $7.5g$.** This provides and upper bound on the maximum acceleration that a person can safely withstand.
- **However, with the airbag stunt the stuntman is able to land in a position that distributes forces evenly across his body.** In our stunt the stuntman will be landing in the box rig while still on the motorcycle. This will result in greater chance for injury under high deceleration.
- **We choose $5g$ as our maximum safe deceleration.** A box rig configuration which results in a higher acceleration will be rejected as unsafe.

4.2 Displacement and energy estimates, a reality check 制动距离及能量的估计, 一个实验验证

If the deceleration of the stuntman and motorcycle is constant through the boxes then we can estimate the distance required to bring him to rest. Since any deviation from constant acceleration will increase either the stopping distance or the peak deceleration, this will give us a lower bound on the stopping distance and hence a lower bound on the required dimensions of the box rig.

Suppose the stuntman enters the rig at time $t = 0$ with velocity v_0

设想一位特技演员从 30 米高度坠落到气垫上时, 重力的作用令他从静止加速到速度 v . 此时, 他撞上汽垫, 然后完全减速至 0, 因此我们有:

其中 d_{fall} 为下落距离, a_{bag} 为特技人在气垫上承受的减速度, h_{bag} 为气垫的高度, g 为重力加速度. 因此

因此我们得到以下结论:

- 当使用气垫时, 特技演员承受的最大平均 (负) 加速度为 $7.5g$. 这提供了一个人所能承受的安全加速度的上限.
- 但是, 借助气垫能够让演员在着陆时全身受力均匀. 在我们的特技表演中, 特技演员将骑着摩托车着陆到纸箱堆. 在比较大的加速度下, 这将会增加特技演员受伤的风险.
- 我们选取 $5g$ 作为最大安全加速度. 任何会导致更大加速度的纸箱堆结构都是不安全的, 将不会被使用.

如果特技演员和摩托车在纸箱中减速过程中, 加速度是一个常数的话, 我们可以估计让演员停下所需的制动距离. 然尔任何偏离恒定加速度的扰动都会增加制动距离或者加速度的最大值, 但这可以为我们提供制动距离的下限, 并由此得到所需纸箱堆在尺寸上的下限.

假设这位特技演员在 $t = 0$ 时刻飞入纸箱堆, 速

主程序

main.m

```
01 Lx = 80;          Ly = 30;          % 灌溉区域尺寸 (m)
02 nx = 801;         ny = 301;         % 网格数
03 dx = Lx/(nx-1); dy = Ly/(ny-1);     % 网格尺寸 (m)
04 flux = 0.15;       % 水源处流量 (m3/min)
05 m2cm = 100;        % 将m转化为cm
06 [pipe, npipe, nsperinkler] = layout(1); % 第一种管组放置位置
07 xi = linspace(0, Lx, nx); yi = linspace(0, Ly, ny);
08 [x, y] = meshgrid(xi,yi);           % 网格坐标
09 field = zeros(ny, nx);              % 初始化降水分布数组
10 for i = 1:npipe
11     for j = 1:nsperinkler
12         r = sqrt( (x - pipe(i).sperinkler(j).x).^2 + ...
13                 (y - pipe(i).sperinkler(j).y).^2 );
14         field = field + distr(r); % 降水分布叠加
15     end
16 end
17 field = field * flux/nsperinkler * m2cm; % 降水分布 (cm/min)
18 imagesc(xi, yi, field)
```

知识要点总结

- 模型: 掌握对平面几何图形的定量描述方法, 了解对三维几何体的定量描述方法.
- 程序: 掌握 `imread`, `image`, `convhull`, `inpolygon` 函数, 能够利用 MatLab 简单的处理一些图片.
- 写作: 掌握两种不同形式的搞要的写作格式.

报名方式

扫码报名



联系方式

- 课程交流群：362223556
- 超模公众号：supermodeling
- 客服超大鱼：210187565
- 获取优惠券：前 100 名减 200
- 课程链接：<https://ke.qq.com/course/265507>

重要提示

- 直播三个工作日后，会自动生成回放，再也不怕错过课程。

Part II

Matlab 快速入门

简介

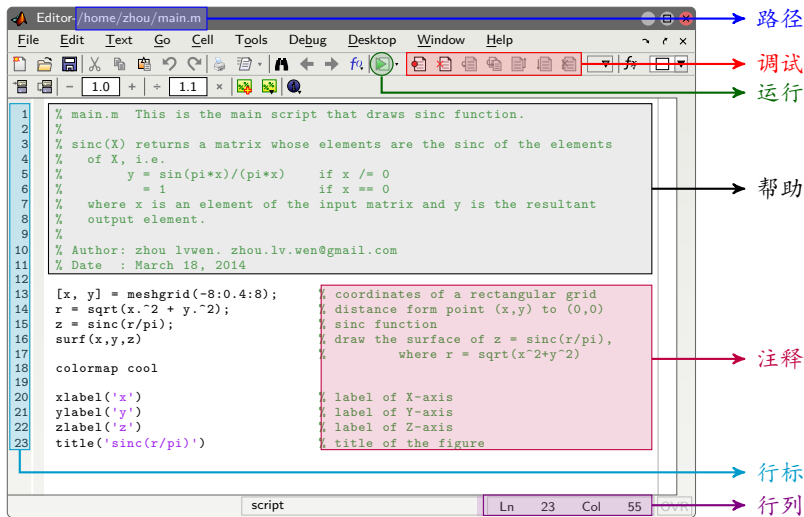
简史

- MATLAB (矩阵实验室) 是 **MAT**rix **LAB**oratory 的缩写;
- 最初由美国的 Clever Moler 教授于 1980 年开发, 初衷是为了解决“线性代数”课程的矩阵运算问题;
- 是一款由 MathWorks 公司 (1984 年成立) 出品的数学软件.

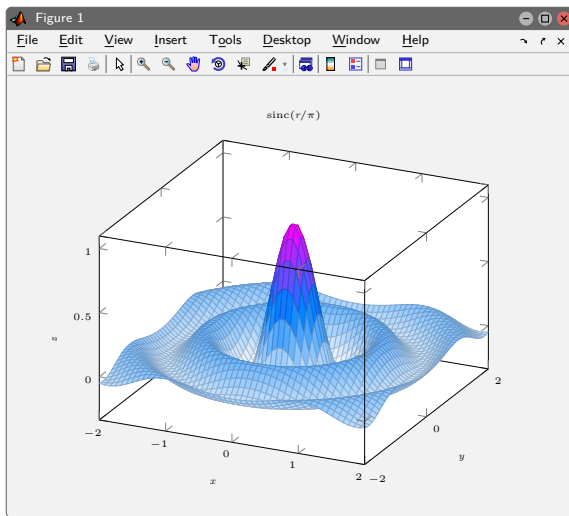
特性

- MATLAB 是一种用于算法开发, 数据可视化, 数据分析以及数值计算的高级技术计算语言和交互式环境.
- MATLAB 可用来创建用户界面及调用其它语言编写的程序.
- MATLAB 中包含众多的附加工具箱, 适合不同领域的应用.

脚本窗口



图形窗口



获得帮助

帮助文档

- 如果你知道一个函数名, 想了解它的用法, 你可以用 'help' 命令得到它的帮助文档:
`>> help functionname`
- 如果你了解含某个关键词的函数, 你可以用 'lookfor' 命令得到相关的函数:
`>> lookfor keyword`

网络资源

- Mathworks 文件交流中心: [▶ Mathworks](#)
- Github 代码托管网站: [▶ Github](#)



实数, 复数, 行向量, 列向量, 矩阵的赋值

Command Window

```
 $f_x$ >> x = 5
```

```
x =
```

```
5
```

```
 $f_x$ >> x = [1 2 3]
```

```
x =
```

```
1    2    3
```

```
 $f_x$ >> x = [1;2;3]
```

```
x =
```

```
1
```

```
2
```

```
3
```

```
 $f_x$ >> clc
```



实数, 复数, 行向量, 列向量, 矩阵的赋值

Command Window

 f_x >> x = [1 2 3; 4 5 6; 7 8 9]

x =

1	2	3
4	5	6
7	8	9

 f_x >> y = [1 2 3
4 5 6]

y =

1	2	3
4	5	6

 f_x >>

向量的一般赋值方法

Command Window

f_x >> x = [0:2]

x =

0.00 1.00 2.00

f_x >> x = [0:2]'

x =

0.00
1.00
2.00

f_x >> x = [0:0.5:2]

x =

0.00 0.50 1.00 1.50 2.00

f_x >> x = linspace(0, 2, 5)

x =

0.00 0.50 1.00 1.50 2.00

f_x >>



常用矩阵

Command Window

 f_x >> $x = \text{zeros}(2,3)$ $x =$

0.00	0.00	0.00
0.00	0.00	0.00

 f_x >> $y = \text{ones}(2)$ $x =$

1.00	1.00
1.00	1.00

 f_x >> $x = \text{eye}(2)$ $x =$

1.00	0.00
0.00	1.00

 f_x >>

固定变量

Command Window

```
 $f_x$ >> pi
```

```
ans =
```

```
3.1416
```

```
 $f_x$ >> z = i
```

```
z =
```

```
0.00 + 1.00i
```

```
 $f_x$ >> x = 1/0
```

```
x =
```

```
Inf
```

```
 $f_x$ >> 0/0
```

```
ans =
```

```
NaN
```

```
 $f_x$ >>
```



矩阵运算和数组运算

Command Window

 f_x >> A = [1 2 3; 4 5 6; 7 8 9]; f_x >> B = [1 3 5; 6 9 0; 2 4 6]; f_x >> C = A + B

C =

2	5	8
10	14	6
9	12	15

 f_x >> D = A - B

D =

0	-1	-2
-2	-4	6
5	4	03

 f_x >> clc

矩阵运算和数组运算

Command Window

 f_x >> A = [1 2 3; 4 5 6; 7 8 9]; f_x >> B = [1 3 5; 6 9 0; 2 4 6]; f_x >> E = A * B

E =

19 33 23

46 81 56

73 129 89

 f_x >> F = A.* B

F =

1 6 15

24 45 0

14 32 54

 f_x >> clc

矩阵运算和数组运算

Command Window

 f_x >> A = [1 2 3; 4 5 6; 7 8 9]; f_x >> B = [1 3 5; 6 9 0; 2 4 6]; f_x >> G = A / B

G =

0	0	0.50
-3.00	0.00	3.50
-6.00	0.00	6.50

 f_x >> H = A ./ B

H =

1.00	0.67	0.60
0.67	0.56	inf
3.50	2.00	1.50

 f_x >> clc

矩阵运算和数组运算

Command Window

 f_x >> A = [1 2 3; 4 5 6; 7 8 9]; f_x >> B = [1 3 5; 6 9 0; 2 4 6]; f_x >> I = A ^ 2

I =

30 36 42

66 81 96

102 126 150

 f_x >> J = A.^ 2

J =

1 4 9

16 25 36

49 64 81

 f_x >> clc

数组和数组行列块操作: 取值

Command Window

```
fx>> A = [1 2 3; 4 5 6; 7 8 9];
```

```
fx>> x = A(1, 3)
```

```
x =
```

```
3
```

```
fx>> y = A(2, :)
```

```
y =
```

```
4     5     6
```

```
fx>> z = A(1:2, 1:3)
```

```
z =
```

```
1     2     3
```

```
4     5     6
```

```
fx>>
```



数组和数组行列块操作: 赋值

Command Window

```
 $f_x$ >> A = [1 2 3; 4 5 6; 7 8 9];
```

```
 $f_x$ >> A(1, 3) = 0
```

A =

1	2	0
4	5	6
7	8	9

```
 $f_x$ >> A(2, :) = [6 5 4]
```

A =

1	2	0
6	5	4
7	8	9

```
 $f_x$ >> A(1:2, 1:2) = [-1 -2; -3 -4]
```

A =

-1	-2	0
-3	-4	4
7	8	9



比较和逻辑运算

Command Window

```
fx>> x = [1 2 3 4 5 6 7 8 9];
```

```
fx>> y = [1 4 3 8 6 5 7 2 9];
```

```
fx>> eq = (x==y)
```

```
eq =
```

```
      1      0      1      0      0      0      1      0      1
```

```
fx>> xy = (x>5)&(y<7)
```

```
xy =
```

```
      0      0      0      0      0      1      0      1      0
```

```
fx>> xoy = (x>5)|(y<7)
```

```
xoy =
```

```
      1      1      1      0      1      1      1      1      1
```

```
fx>>
```



比较和逻辑运算

Command Window

```
f_x>> x = [1 -2 3 -4 5 -6 7 -8 9];
```

```
f_x>> x(x<0) = 0
```

```
x =
```

```
1    0    3    0    5    0    7    0    9
```

```
f_x>> y = [1 2 3;-4 5 6; 7 8 9];
```

```
f_x>> y(y(:,1)<0,:) = 0
```

```
y =
```

```
1    2    3
```

```
0    0    0
```

```
7    8    9
```

```
f_x>>
```



数组操作函数: flipud, fliplr, rot90

Command Window

```
f_x>> A = [1 2 3; 4 5 6; 7 8 9];
```

```
f_x>> B = flipud(A)
```

```
A =
```

```
    7    8    9  
    4    5    6  
    1    2    3
```

```
f_x>> C = rot90(A)
```

```
C =
```

```
    3    6    9  
    2    5    8  
    1    4    7
```

```
f_x>>
```



数组操作函数: sum

Command Window

```
fx>> A = [1 2 3];  
fx>> sum(A)  
  
ans =  
  
6  
  
fx>> B = [1 2 3; 4 5 6; 7 8 9];  
fx>> sum(B)  
  
ans =  
  
12 15 18  
  
fx>> sum(B,2)  
  
ans =  
  
6  
15  
25  
  
fx>>
```



数组操作函数: max, min

Command Window

```
f_x>> A = [1 2 3];  
f_x>> max(A)  
ans =  
3  
f_x>> max(A,2)  
ans =  
2 2 3  
f_x>> B = [1 3 9; 4 8 6];  
f_x>> max(B)  
ans =  
4 8 9  
f_x>> max(B, [], 2)  
ans =  
9  
8
```



常用数学函数: sin, cos, tan, cot, asin, acos, atan, acot

Command Window

 f_x >> $x = 0:\pi/6:\pi;$ $x =$

0.00 0.52 1.05 1.57 2.09 2.62 3.14

 f_x >> $y = \sin(x)$ $y =$

0.00 0.50 0.87 1.00 0.87 0.50 0.00

 f_x >> $z = \text{asin}(y)$ $z =$

0.00 0.52 1.05 1.57 2.09 2.62 3.14

 f_x >>

常用数学函数: abs, sqrt

Command Window

 f_x >> $x = [-4 \ 9 \ -16 \ 25];$ $x =$ $-4 \quad 9 \quad -16 \quad 25$ f_x >> $y = \text{abs}(x)$ $y =$ $4 \quad 9 \quad 16 \quad 25$ f_x >> $z = \text{sqrt}(y)$ $z =$ $1 \quad 3 \quad 4 \quad 5$ f_x >>

常用数学函数: ceil, fix, floor, round

Command Window

```
fx>> x = [-1.6 -0.2 1.2 0.6];
```

```
fx>> y = ceil(x)
```

```
y =
```

```
    -1     0     2     1
```

```
fx>> z = floor(x)
```

```
z =
```

```
    -2    -1     1     0
```

```
fx>> g = fix(x)
```

```
g =
```

```
    -1     0     1     0
```

```
fx>> f = round(x)
```

```
f =
```

```
    -2     0     1     1
```

```
fx>>
```



基本语句

基本语句

- `for .. end`
- `if .. else .. end`
- `while .. end`
- `switch .. case .. end`

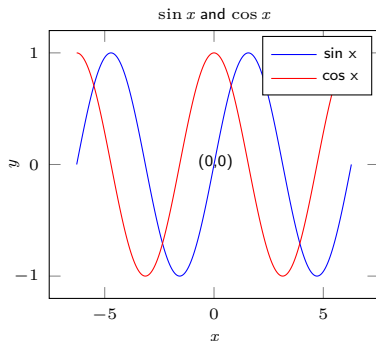
举例: 求 1-10 以内的奇数和

```
1 % sum of the odd numbers between 1 and 10
2 x = 0;
3 for i = 1:10
4     if mod(i,2)
5         x= x + i;
6     end
7 end
```



简单作图

```
1 x = -2*pi:0.1:2*pi;  
2 y1 = sin(x);  
3 y2 = cos(x);  
4 plot(x, y1, '-b');  
5 hold on  
6 plot(x, y2, '-r');  
7 xlabel('x')  
8 ylabel('y')  
9 text(0,0, '(0,0)')  
10 legend('sin x', 'cos x')
```



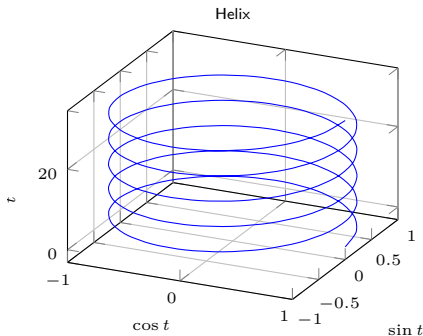
简单控制语句

- title(图形名称)
- xlabel(x 轴说明); ylabel(y 轴说明)
- text(x,y, 图形说明)
- legend(图例 1, 图例 2, ...)
- grid on / grid off / grid minor
- axis([xmin xmax ymin ymax]), xlim([xmin, xmax])



三维曲线图

```
1  t=0:pi/50:10*pi;  
2  x = sin(t);  
3  y = cos(t);  
4  z = t;  
5  plot3(x,y,z)  
6  title('Helix')  
7  xlabel('sin t')  
8  ylabel('cos t')  
9  zlabel('t')  
10 grid on
```



三维曲面图: 补充函数 meshgrid

Command Window

```
 $f_x$ >> [x, y] = meshgrid(1:3, 1:3)
```

```
x =
```

```
1    2    3
1    2    3
1    2    3
```

```
y =
```

```
1    1    1
2    2    2
3    3    3
```

```
 $f_x$ >> rsq = (x-2).^2 + (x-2).^2
```

```
rsq =
```

```
2    1    2
1    0    1
2    1    2
```

```
 $f_x$ >> r = sqrt(rsq)
```

```
r =
```

```
1.4142  1.0000  1.4142
1.0000         0  1.0000
1.4142  1.0000  1.4142
```

(1, 1)	(2, 1)	(3, 1)
(1, 2)	(2, 2)	(3, 2)
(1, 3)	(2, 3)	(3, 3)

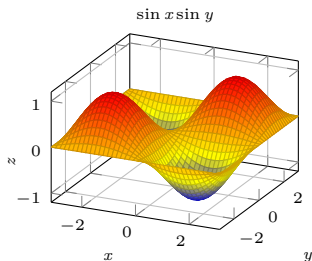
2	1	2
1	0	1
2	1	2

$\sqrt{2}$	1	$\sqrt{2}$
1	0	1
$\sqrt{2}$	1	$\sqrt{2}$



三维曲面图

```
1 [x,y] = meshgrid(-pi:0.1:pi);  
2 z = sin(x).*cos(y);  
3 mesh(x,y,z) % meshc(x,y,z)  
4 surf(x,y,z) % surfc(x,y,z)  
5 xlabel('x')  
6 ylabel('y')  
7 zlabel('z')  
8 title('sin x sin y')
```



M 函数格式

M 函数格式

```
1 function [output 1, ..] = functionname(input1, ..)
2 % comment of this function
3
4 MatLab command 1;
5 MatLab command 2;
```

举例: 求矩形面积

```
1 function area = rectarea(L, W)
2 % rectarea Area of a rectangle
3 %
4 % rectarea(l, w) calculate the area of a rectangle
5 % with a length of L and a width of W
6
7 area = L .* W
```


txt 文本数据读 |

Command Window

```
f_x>> data = load('score.txt');
f_x>> w = data(:,1);
f_x>> s = data(:,2:3);
f_x>> r = w'*s

r =

    88.29    87.19

f_x>>
```

score.txt

% w	章小天	刘强西	
0.13	65	99	% 财富
0.05	90	80	% 权势
0.17	99	65	% 美貌
0.08	90	90	% 智商
0.09	90	99	% 情商
0.05	99	65	% 文艺
0.06	95	90	% 学历
0.08	81	98	% 阅历
0.08	98	88	% 体力
0.06	90	80	% 年龄
0.02	70	98	% 人际
0.13	88	99	% 撩技



txt 文本数据读 II

Command Window

```
 $f_x$ >> fid = fopen('swords.txt', 'r');  
 $f_x$ >> t1 = fgetl(fid)  
  
t1 =  
    刘正风  
  
 $f_x$ >> t2 = fgetl(fid)  
  
t2 =  
    包惜弱  
  
 $f_x$ >> t3 = fgetl(fid)  
  
t3 =  
    东方不败  
  
 $f_x$ >> t = [t1(1) t2(3) t3(1)]  
  
t =  
    刘弱东  
  
 $f_x$ >> fclose(fid);  
 $f_x$ >>
```

swords.txt

刘正风
包惜弱
东方不败
张翠山
小龙女
天山童姥
贱宁公主
摩诃巴思

高颜超
清虚
无崖子
马夫人



txt 文本数据写

Command Window

```
f_x>> fid = fopen('bill.txt');  
f_x>> fprintf(fid, '京西商城月账单\n\n');  
f_x>> fprintf(fid, '电子 -20000');  
f_x>> fprintf(fid, '包包 %d', -1200);  
f_x>> fprintf(fid, '%s %d\n\n', '服装', -24000);  
f_x>> fprintf(fid, '账目细节\n');  
f_x>> detail = -[845 832.5 836 872.2 825 844];  
f_x>> fprintf(fid, '%-5d %5.1f\n', [1:6; detail]);  
f_x>> fclose(fid)  
f_x>>
```

bill.txt

京西商城月账单

电子 -10000

包包 -1200

服装 -24000

账目细节

1 -845.0

2 -832.5

3 -836.0

4 -872.2

5 -825.0

6 -844.0



Excel 文本数据读

Command Window

```
fx>> data = xlsread('data.xls','Sheet1','A3:C5')
```

```
data =
```

```
    95    77    77
    10    42    54
     4    26    60
```

```
fx>> data = xlsread('data.xls','Sheet1','A1:B3')
```

```
data =
```

```
NaN    81
NaN    35
 95    77
```

```
fx>> data(isnan(data)) = 0
```

```
data =
```

```
  0    81
  0    35
 95    77
```

```
fx>>
```

data.xls: Sheet1

	A	B	C
1		81	32
2		35	88
3	95	77	77
4	10	42	54
5	4	26	60
6	48	85	78
7	77	35	98
8	89	75	29
9	81	3	57
10	83	84	39
11	96	15	87
12	65	92	11



Excel 文本数据写

Command Window

```
f_x>> xlswrite('RS.xls',{ 'T','W'},'Sheet1','B1:C1')  
f_x>> XYZ = { 'X','Y','Z'; 6 6 4; 8 7 5};  
f_x>> xlswrite('RS.xls',XYZ,'A2:C4')  
f_x>> xyz = [4 2 3; 8 9 1];  
f_x>> xlswrite('RS.xls',xyz,'A6:C7')  
f_x>>
```

RS.xls: Sheet1

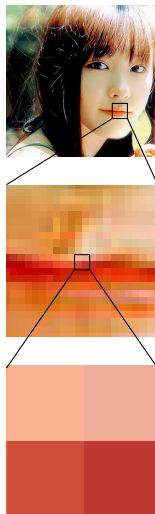
	A	B	C
1		T	W
2	X	6	8
3	Y	6	7
4	Z	4	5
5			
6	4	2	3
7	8	9	1
8			
9			
10			
11			
12			



图片数据读入

Command Window

```
f_x>> Yui = imread('Yui.jpg');  
f_x>> size(Yui)  
  
ans =  
    374    374     3  
  
f_x>> image(Yui)  
f_x>> lip = Yui(242:266, 255:279, :);  
f_x>> image(lip);  
f_x>> procheilon = lip(12:13, 12:13, :);  
  
procheilon(:,:,1) =  
    255    252  
    228    206  
procheilon(:,:,2) =  
    175    178  
     90     71  
procheilon(:,:,3) =  
    118    139  
     51     42  
  
f_x>> image(procheilon);  
f_x>>
```

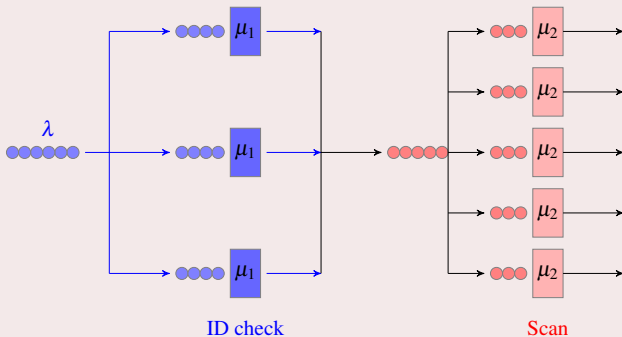


Part III

编程实例：机场安检问题

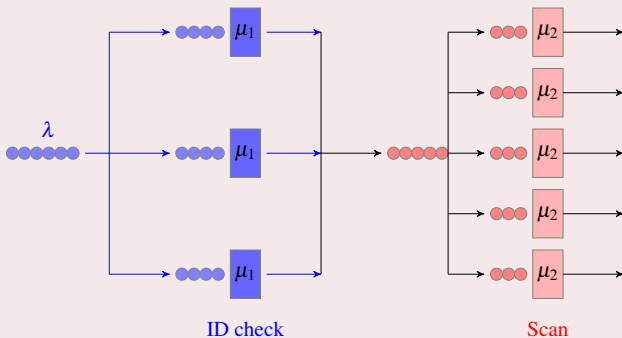
MCM 2017 D: 机场安检问题

登机流程：1. 办理登机手续 (ID Check); 2. 安检 (Scan)



- 建立模型确定旅客通过安检口的流量和瓶颈。
- 设计对现有系统的改进，减少旅客等待时间。

两并联排队系统串联



- 乘客到达时间间隔服从 λ (秒/人) 的指数分布。
- 登机手续/安检的服务时长服从 μ_1/μ_2 (秒/人) 的指数分布。

定义

- 第 i 个乘客到达时刻: $A(i)$
- 第 i 个乘客开始和结束办理登机手续时刻: $C_0(i)$ 、 $C(i)$
- 第 i 个乘客开始安检和结束时刻: $S_0(i)$ 、 $S(i)$
- 第 j 个登机手续台结束服务时刻: $E_c(j)$, 初值 $E_c = [0, 0, 0]$
- 第 j 个安检台结束服务时刻: $E_s(j)$, 初值 $E_s = [0, 0, 0, 0, 0, 0]$

递推公式

- $A(i) = A(i-1) + \text{exprnd}(\lambda)$
- $C_0(i) = \max\{A(i), \min(E_c)\}$, $C(i) = C_0(i) + \text{exprnd}(\mu_1)$
- $E_c(k) = C(i)$
- $S_0(1) = \max\{C(i), \min(E_s)\}$, $S(i) = S_0(i) + \text{exprnd}(\mu_2)$
- $E_s(j) = S(i)$

收益模型

等待时长

- 第 i 个乘客等待时长: $W(i) = S(i) - A(i)$
- 所有平均等待时长: $T = \frac{1}{n} \sum W(i)$

假设

- 服务一个乘客收益为 C_p 。
- 每个工作人员单位时间工资为 C_t 。
- 每个登机手续服务台和安检台分别需要 1 和 3 名工作人员。

收益模型

$$P = \frac{C_p}{T} - (n_c + 3n_s)C_t$$

程序实现

mms.m

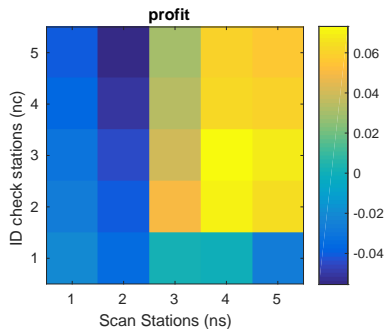
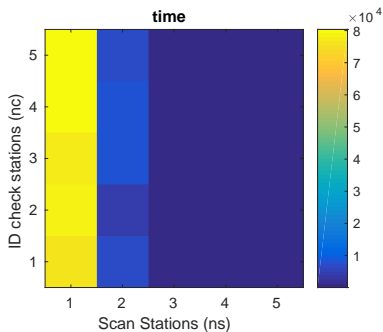
```
01 function [Wm, Pm] = mms(nc, ns)
02 lambda = 12.9; mu1 = 11.2; mu2 = 28.6; n = 10000;
03 [A, C0, C, S0, S] = deal(zeros(n,1));
04 Ec = zeros(nc,1); Es = zeros(ns,1);
05 for i = 1:n
06     if i>1; A(i) = A(i-1)+exprnd(lambda); end
07     [minC, k] = min(Ec);
08     C0(i) = max(A(i), minC);
09     C(i) = C0(i) + exprnd(mu1);
10     Ec(k) = C(i);
11
12     [minS, j] = min(Es);
13     S0(i) = max(C(i), minS);
14     S(i) = S0(i) + exprnd(mu2);
15     Es(j) = S(i);
16 end
17 Wm = mean(S - A);
18 cp = 6.60; tsa = 18.5/3600; Pm = cp/Wm - (nc+ns*3)*tsa;
```

程序实现

main.m

```
01 Wm = zeros(5); Pm = zeros(5);
02 for nc = 1:5
03     for ns = 1:5
04         [Wm(nc,ns), Pm(nc,ns)] = mms(nc, ns);
05     end
06 end
07
08 imagesc(Pm); colorbar; axis image
09 set(gca,'xtick',1:5, 'ytick',1:5, 'ydir','normal');
10 xlabel('Scan Stations (ns)');
11 ylabel('ID check stations (nc)')
```

计算结果



本幻灯片中的模型参考了 2017 年布朗大学的特等奖论文

Thank You!!!