

有趣的盒子：代码

R.H Wang

January 2024

```
^^I\begin{mybox1}
^^I^^I设 $A, B$ 都是域 $F$ 上的 $n$ 级矩阵. 证明: 如果 $AB \pm BA = A$ , 且 $B$ 是幂零矩阵, 那么 $A=0$ .
^^I\end{mybox1}
```

设 A, B 都是域 F 上的 n 级矩阵. 证明: 如果 $AB \pm BA = A$, 且 B 是幂零矩阵, 那么 $A = 0$.

0.1 代码盒子

第一个 Mac 风格:

```
^^I\begin{macbox}{这是标题}
^^I^^I\begin{lstlisting}[style=python4]
^^I^^I^^Idef main(): # 主函数
^^I^^I^^Ipool = multiprocessing.Pool(processes=2) # 定义2个大小的进程池
^^I^^I^^Ifor item in range(10): # 创建10个进程
^^I^^I^^Iresult = pool.apply_async(func=work, args=(item,)) # 非阻塞形式执行进程
^^I^^I^^Iprint(result.get()) # 获取进程返回结果
^^I^^I^^Ipool.close() # 执行完毕后关闭进程池
^^I^^I^^Ipool.join() # 等待进程池执行完毕
^^I^^I\end{lstlisting}
^^I\end{macbox}
```

这是标题

```
def main(): # 主函数
    pool = multiprocessing.Pool(processes=2) # 定义2个大小的进程池
    for item in range(10): # 创建10个进程
        result = pool.apply_async(func=work, args=(item,)) # 非阻塞形式执行进程
    print(result.get()) # 获取进程返回结果
    pool.close() # 执行完毕后关闭进程池
    pool.join() # 等待进程池执行完毕
```

```
^^I\begin{macboxd}{这是标题}
^^I^^I\begin{lstlisting}[style=python3]
^^I^^I^^Idef main(): # 主函数
^^I^^I^^Ipool = multiprocessing.Pool(processes=2) # 定义2个大小的进程池
^^I^^I^^Ifor item in range(10): # 创建10个进程
^^I^^I^^Iresult = pool.apply_async(func=work, args=(item,)) # 非阻塞形式执行进程
^^I^^I^^Iprint(result.get()) # 获取进程返回结果
^^I^^I^^Ipool.close() # 执行完毕后关闭进程池
^^I^^I^^Ipool.join() # 等待进程池执行完毕
^^I^^I\end{lstlisting}
^^I\end{macboxd}
```

这是标题

```
def main(): # 主函数
    pool = multiprocessing.Pool(processes=2) # 定义2个大小的进程池
    for item in range(10): # 创建10个进程
        result = pool.apply_async(func=work, args=(item,)) # 非阻塞形式执行进程
    print(result.get()) # 获取进程返回结果
    pool.close() # 执行完毕后关闭进程池
    pool.join() # 等待进程池执行完毕
```

style=python3,总共定义了python2 python2 python3 pythonx四个风格的可以选用. 以及可以选取matlab和R.

0.2 代码块

MATLAB code

```
1 % Euler method for the ODE model
2 % u'(x)=x^2+x-u, x in [0,1]
3 % Initial condition: u(0)=0 ;
4 % Exact solution: u(x)=-exp(-x)+x^2-x+1.
5 clear all; clf
```

```

6      h=0.1;
7      x=0:h:1;
8      N=length(x)-1;
9      u(1)=0;                                % initial value
10     fun=@(t,u) t.^2+t-u;                    % RHS
11
12     for n=1:N
13         u(n+1)=u(n)+h.*fun(x(n),u(n));
14     end
15
16     ue=-exp(-x)+x.^2-x+1;                    % exact solution
17     plot(x,ue,'b-',x,u,'r+', 'LineWidth',1)
18     legend('Exact','Numerical','location','North')
19     %title('Euler method','fontsize',12)
20     set(gca,'fontsize',12)
21     xlabel('x','fontsize',16), ylabel('u','fontsize',16,'Rotation',0)

```

R code

```

1      # parameter constraints
2      trans = function(b0){
3          b1 = b0
4
5          # probability
6          b1[1:2] = exp(b0[1:2])/(1+exp(b0[1:2]))
7
8          # variance
9          b1[5] = b0[5]^2
10
11         # AR(1), AR(2) coefficients
12         XX6 = b0[6]/(1+abs(b0[6]))
13         XX7 = b0[7]/(1+abs(b0[7]))
14         b1[6] = XX6 + XX7
15         b1[7] = -1*XX6*XX7
16
17         return(b1)
18     }

```