

实验一 MATLAB 语言与编程基础

1.1 实验目的

- 1、熟悉 MATLAB 软件环境，掌握运行一个 MATLAB 软件 M 文件设计的基本步骤，包括 编辑、调试和运行。
- 2、掌握 MATLAB 语言程序设计的基本框架，能够编写的 MATLAB 程序(M 文件)。
- 3、了解程序调试的思想，能找出并改正 MATLAB 程序中的语法错误。

1.2 MATLAB 知识要点

1.2.1 编写、保存并运行第一个 MATLAB 语言脚本文件

(1)启动 MATLAB

进入上机环境，从系统桌面上双击 MATLAB 图标 ，进入 MATLAB 软

件环境，如图 1 所示。软件各窗口区的分布及其作用参见教程 0.4 节 MATLAB 简介部分。大家可以参照表 0.2 熟悉一下 MATLAB 几个通用命令的使用。

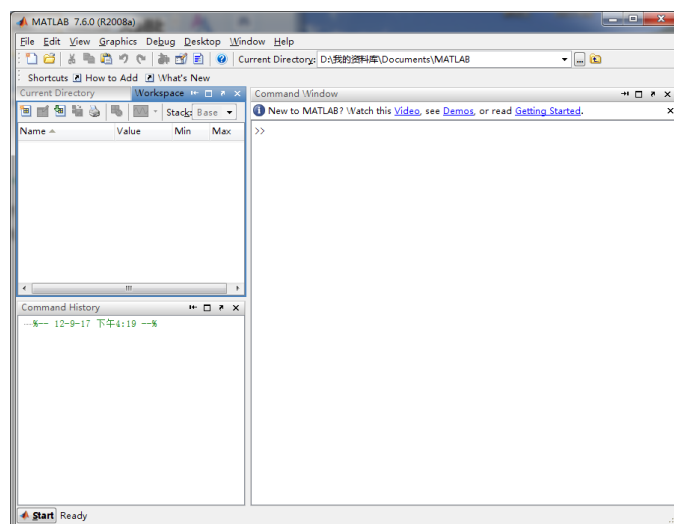



图 1 MATLAB 软件环境

(2)进入编程环境

通过点击工具栏上的“New M-File”按钮或者按如图 2 所示通过依次点击 File --> New --> M-File 打开编程环境，如图 3 所示。

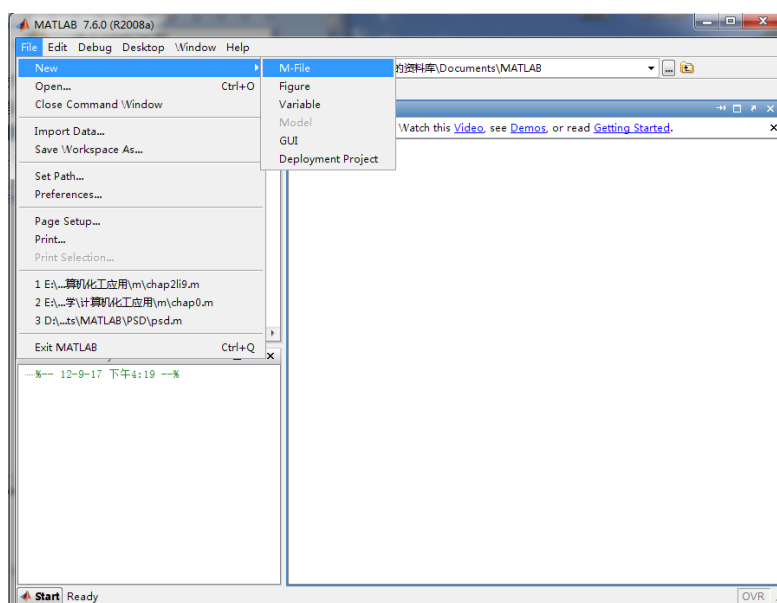


图 2 新建 M 文件

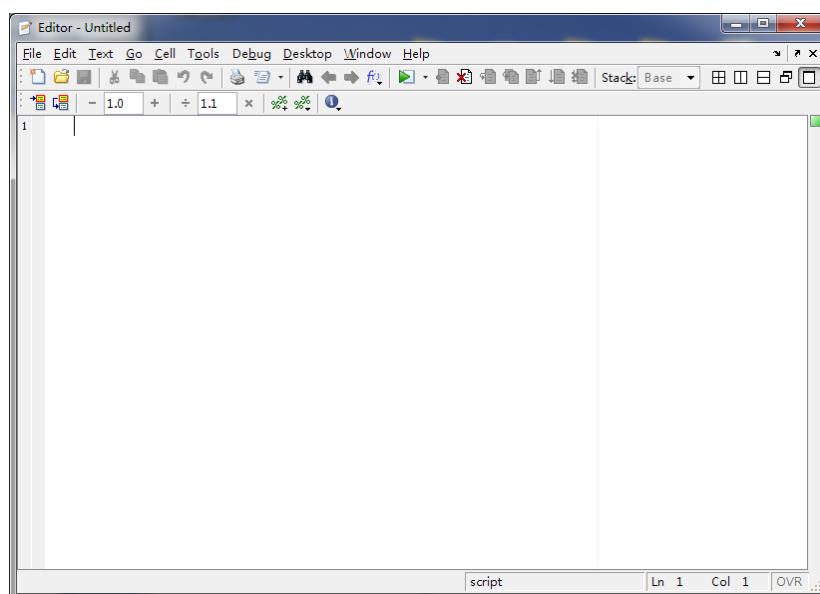


图 3 M 文件编辑窗口

(3)输入和编辑源程序

在图 3 所示的程序编辑窗口输入并编辑如下源程序：

```

M=2000;
D=0.5;
T=283:10:323;
miu=10.^(-10.2158+1.7925e3./T+1.7730e-2*T-1.2634e-05*T.^2);
rhou=21.6688*0.2740.^(-(1-T/647.13).^0.28571);
deltP=M^1.8*miu.^0.2./(20000*D^4.8*rhou)

```

输入完成后效果如图 4 所示。

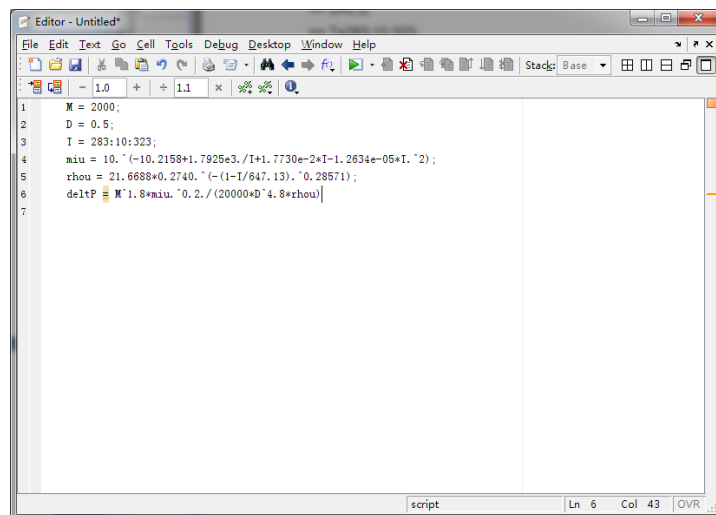



图 4 输入源代码后的程序编辑窗口

(4)保存源程序

点击工具栏上的存盘图标  或菜单栏 File --> Save 保存源程序，文件名为 First，如图 5 所示。注意：可以不用加文件扩展名.m，系统会自动添加。

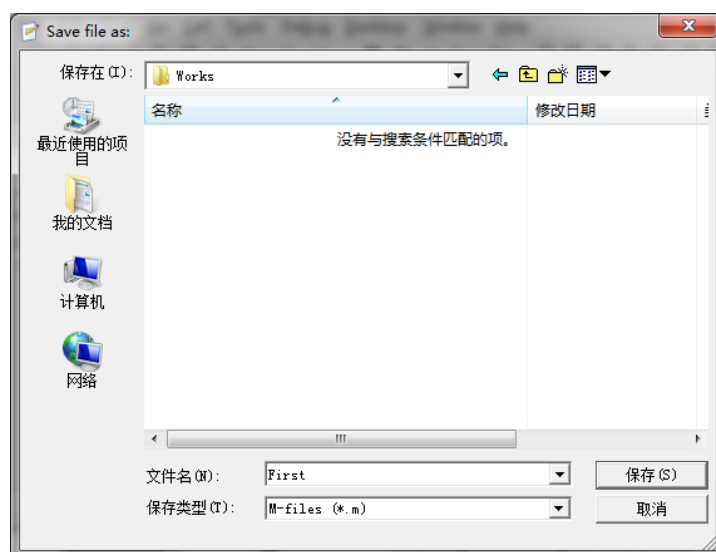


图 5 保存文件

(5)运行程序

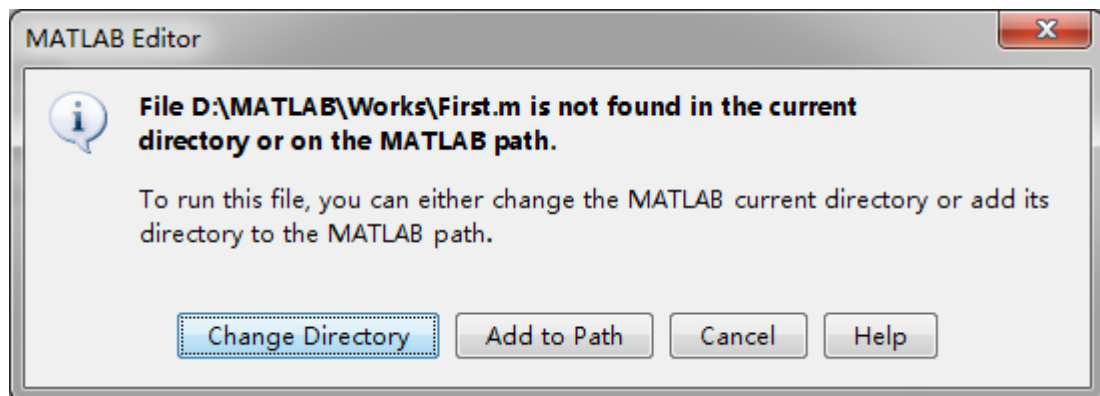



图 6 改变当前路径

点击工具栏上的运行按钮  或 F5 键运行程序，可能会弹出图 6 所示窗口，点击第一个按钮 **Change Directory**，改变当前路径。运行结果会将显示在命令窗口，如图 7 所示。

此外大家可以在当前路径窗口看到多了一个文件名为 **First.m** 的文件。文件类型为 **M-File**。

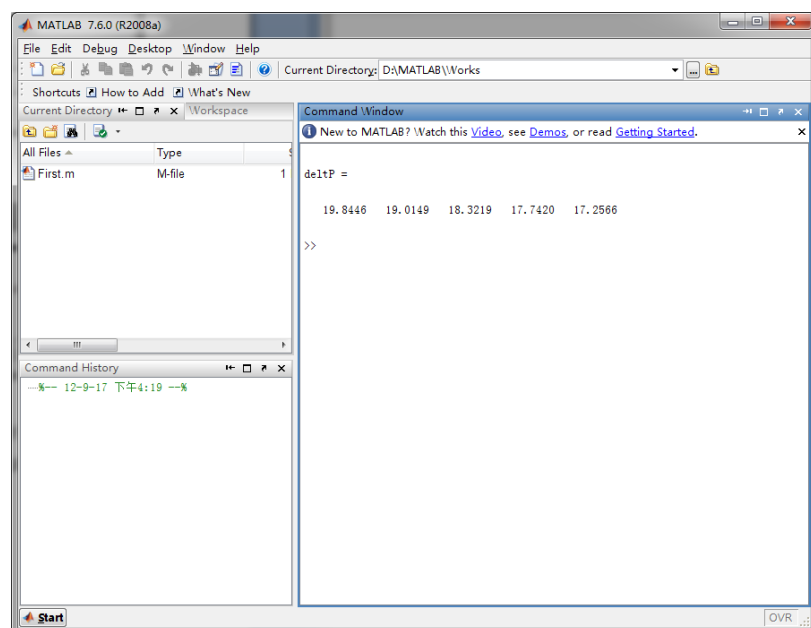



图 7 程序运行结果

1.2.2 打开并调试第一个 MATLAB 语言脚本文件

在上一节，我们应该掌握了编写、保存并运行一个 MATLAB 程序的基本步骤。

下面我们介绍打开并调试一个 MATLAB 程序的基本步骤。





(1)打开一个 MATLAB 程序

点击工具栏上的 Open File 图标  或通过菜单栏 File 的 Open...选项打开 First.m 文件。

(2)设定断点

源程序左边有行号，在行号的右边有短横线，点击某行的短横线就可以在当前行号设定一个断点，如图 8 所示，在第 1 行设定断点。

(3)运行程序

点击工具栏上的运行按钮  或 F5 键运行程序，可以发现程序在断点处暂停，如图 9 所示。注意：绿色箭头  所指的行目前并没有执行，将在下一步执行。可以对照软件的变量空间窗口（Workspace）观察，当绿色箭头指在源程序第 1 行时   `M = 2000;`，变量空间窗口并没有变量 M 生成。

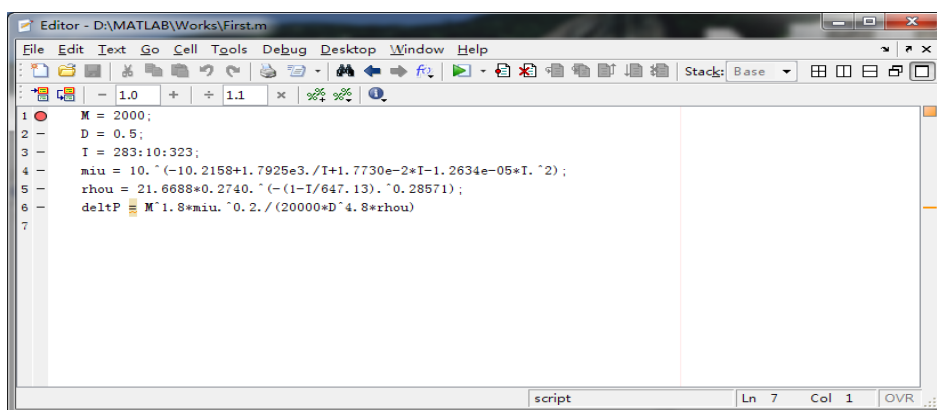


图 8 在第 1 行设定断点

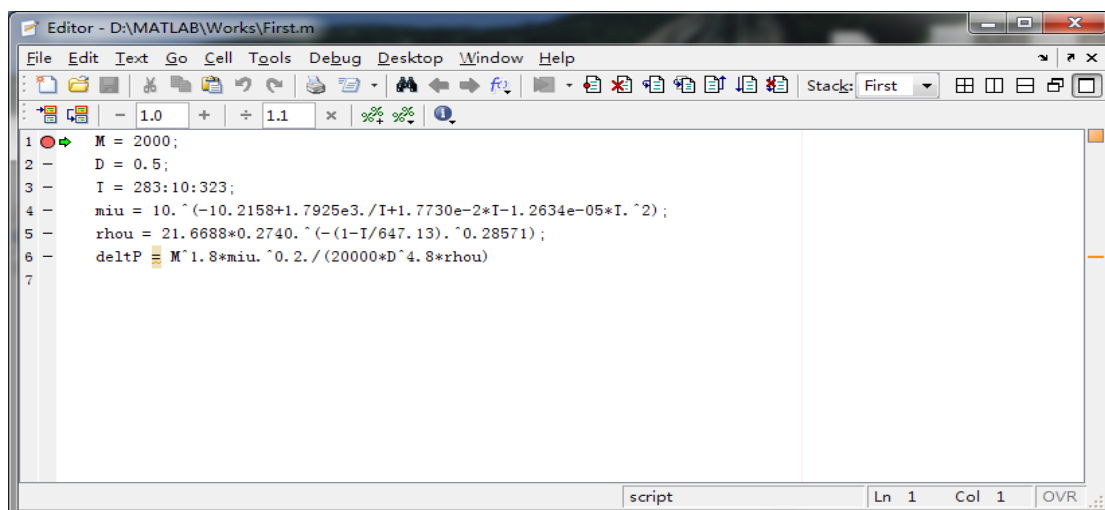




图 9 程序在断点出暂停

(4)单步执行命令

点击工具栏上的 Step 按钮  或 F10 键单步执行程序。可以通过程序编辑窗口观察到绿色箭头  指在程序的第 2 行。也就是说第 1 行命令已经执行完毕，下一步将执行第 2 行命令，如图 10 所示。

随着第 1 行命令的执行，变量空间也发生变化，可以发现在变量空间多了一个变量 M，如图 11 所示。但是由于第 1 行命令尾部“;”的作用，第 1 行命令的执行结果并没有在命令窗口显示出来。

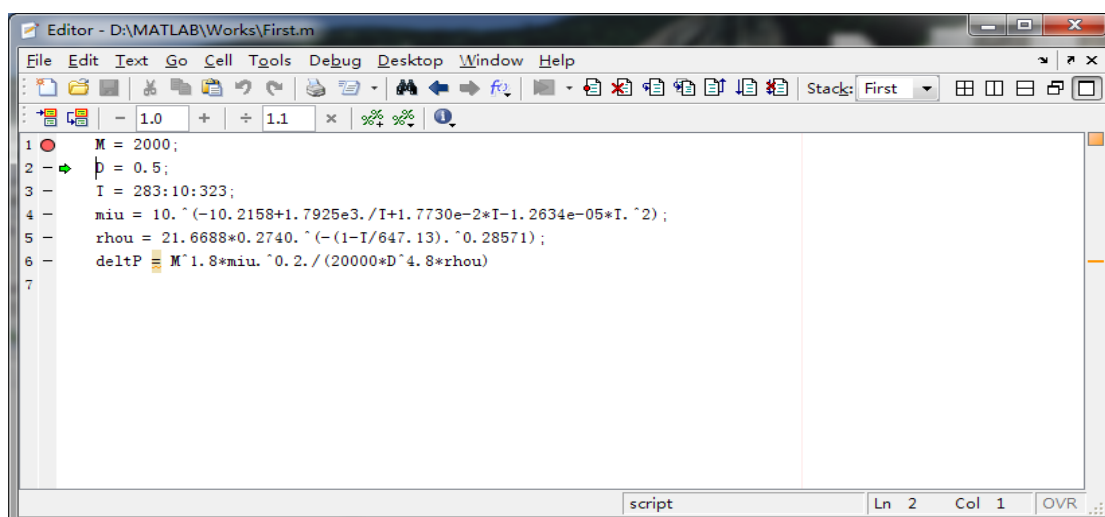


图 10 单步执行 1 次后的程序编辑运行界面

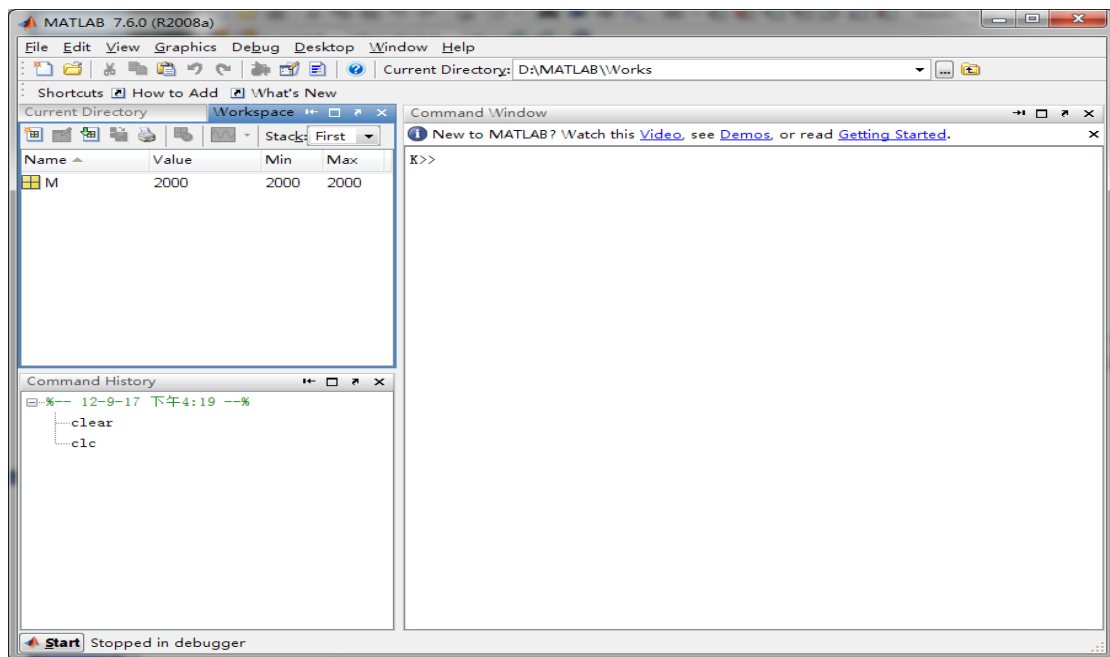



图 11 单步执行 1 次后的变量空间

(5)继续单步执行命令

继续通过 F5 键单步执行后面的命令，并对照变量空间观察程序执行的效果。可以发现每执行 1 行命令就有相应保存运算结果的变量生成。

(6)执行最后一行命令

最后一行命令生成变量 `deltP`，并在命令窗口显示运行结果。同时运行界面的绿色箭头变为向下 ，表示当前程序执行完毕，但是并未退出当前程序的运行环境，如图 12 所示，继续单步运行，退出当前程序。

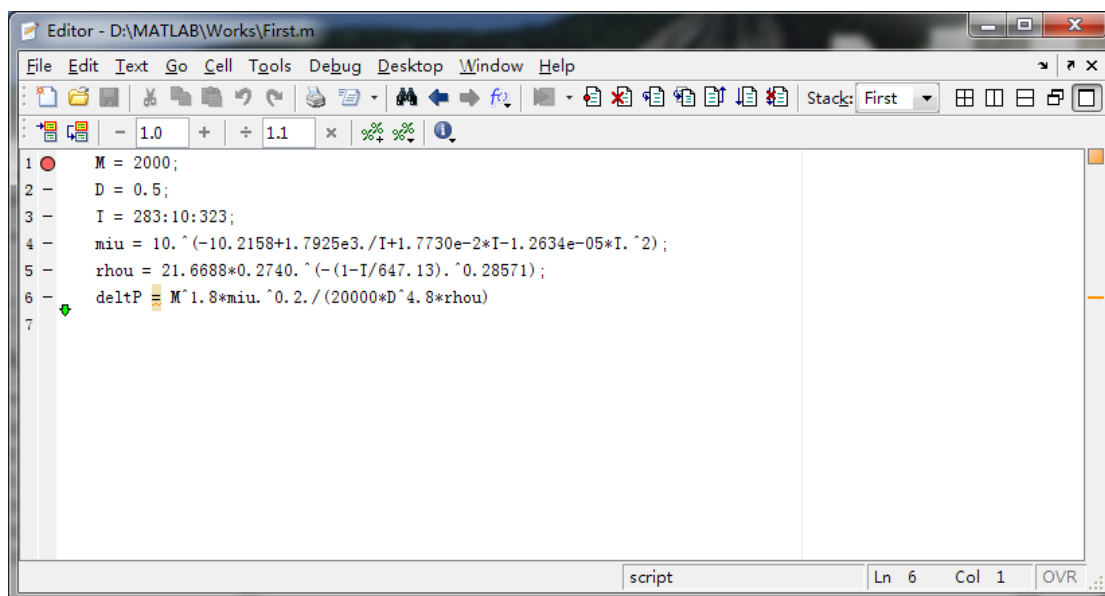


图 12 执行完最后一行命令的运行界面

1.2.3 编写、保存并运行第一个 MATLAB 语言函数文件

(1)编写并保存 MATLAB 语言函数文件

将前面的 MATLAB 脚本文件修改为函数文件，该函数文件可以用于计算不同管径、流量和温度下的压降。函数保存为 `dP.m`，文件的保存参照第（一）节第 4 小节部分。

源代码如下：

```

function deltP = dP(M,D,T)
miu = 10.^(-10.2158+1.7925e3./T+1.7730e-2*T-1.2634e-05*T.^2);
rhous = 21.6688*0.2740.^(-(1-T/647.13).^0.28571);
deltP = M^1.8*miu.^0.2./(20000*D^4.8*rhou);

```

(2)函数的运行

由于本函数定义有输入参数，并且输入参数没有缺省值，所以在程序运行前需要指定输入参数。

如果直接按第（一）节介绍的方法，直接点击运行按钮，命令窗口会提示错误信息，如图 13 所示。

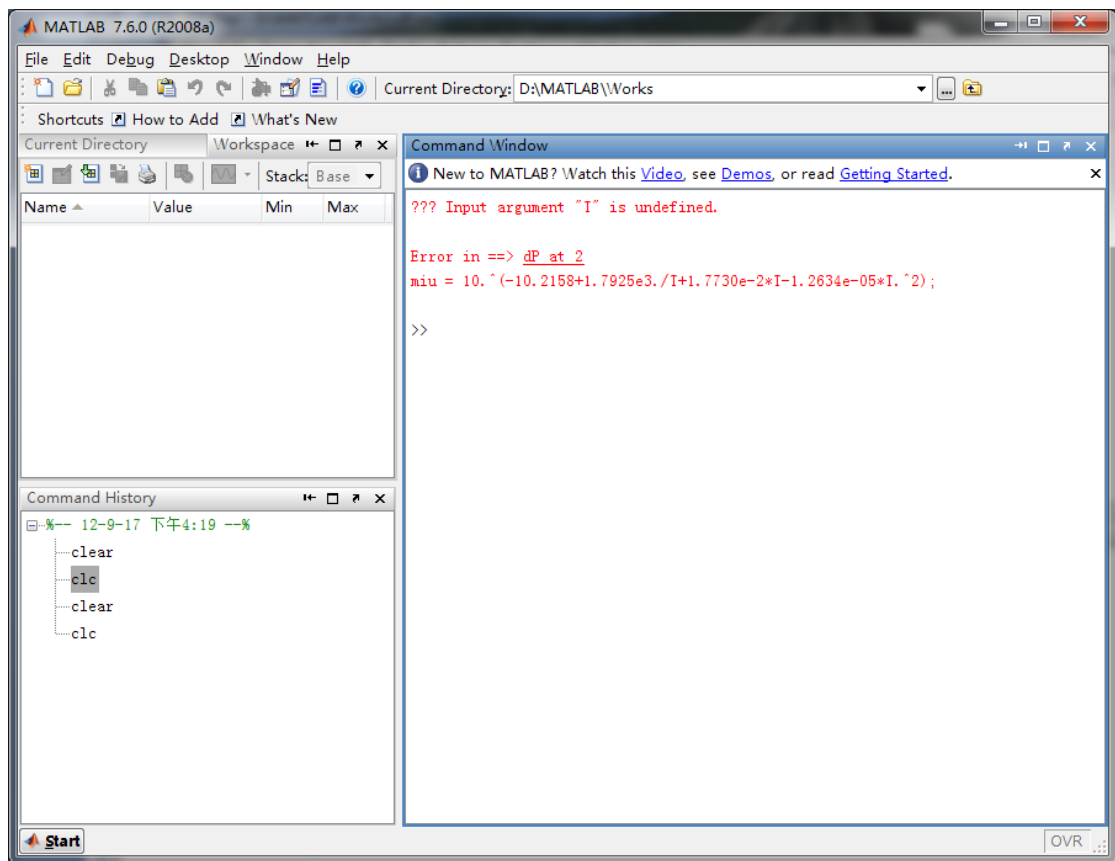


图 13 未指定输入参数时的错误提示

正确的做法是在命令窗口输入如下命令：

```
deltP = dP(2000,0.5,[283:10:323])
```

然后回车。运行效果如图 14 所示。

(3)函数的单步执行

对照第（一）节程序的运行效果可以发现，在函数运行结束后，变量空间只有变量 `deltP`，而没有其它保存中间运行结果的变量。

通过函数的单步执行可以观察在程序执行过程中变量的变化情况。按照前面介绍的方法在函数第一行命令（对源程序第 2 行）处设定断点，然后在命令窗口输入如下命令后回车：

```
deltP = dP(2000,0.5,[283:10:323])
```

运行效果如图 15 所示。

函数的输入参数已在变量空间生成并赋值。

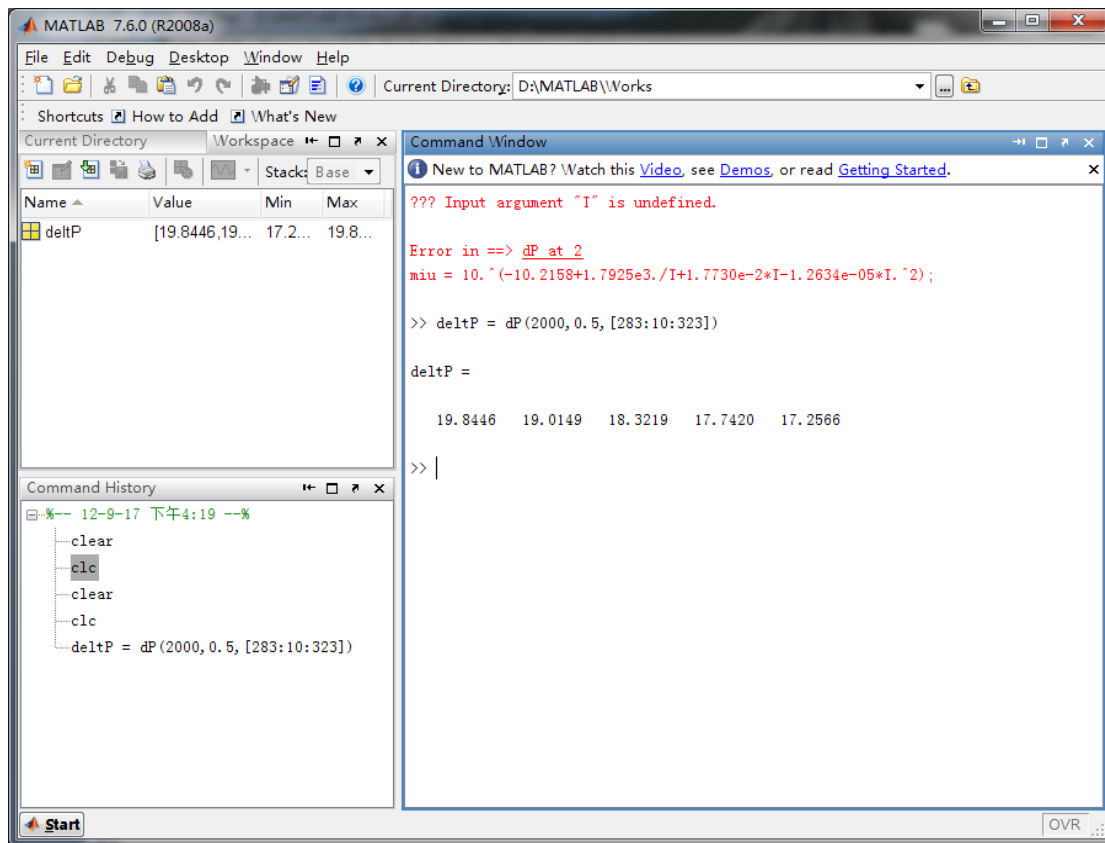


图 14 函数运行方式及效果

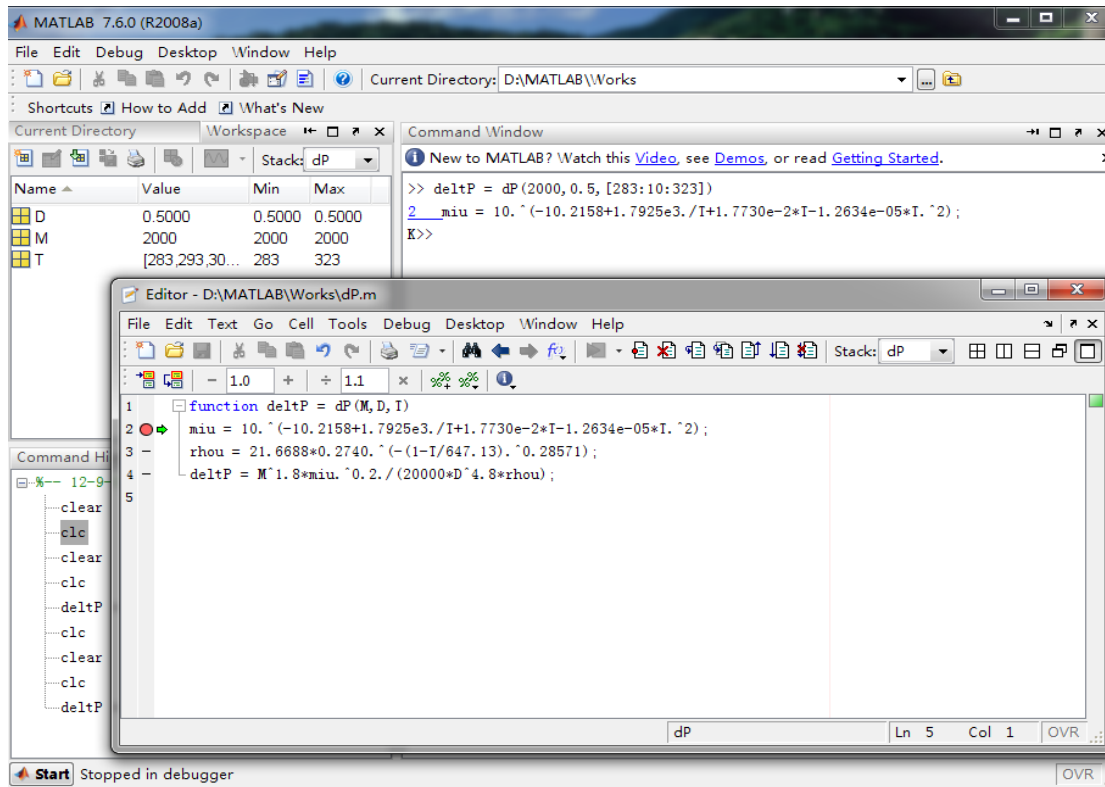


图 15 函数单步运行效果

(4)函数的最后一行命令执行完毕与退出函数的区别

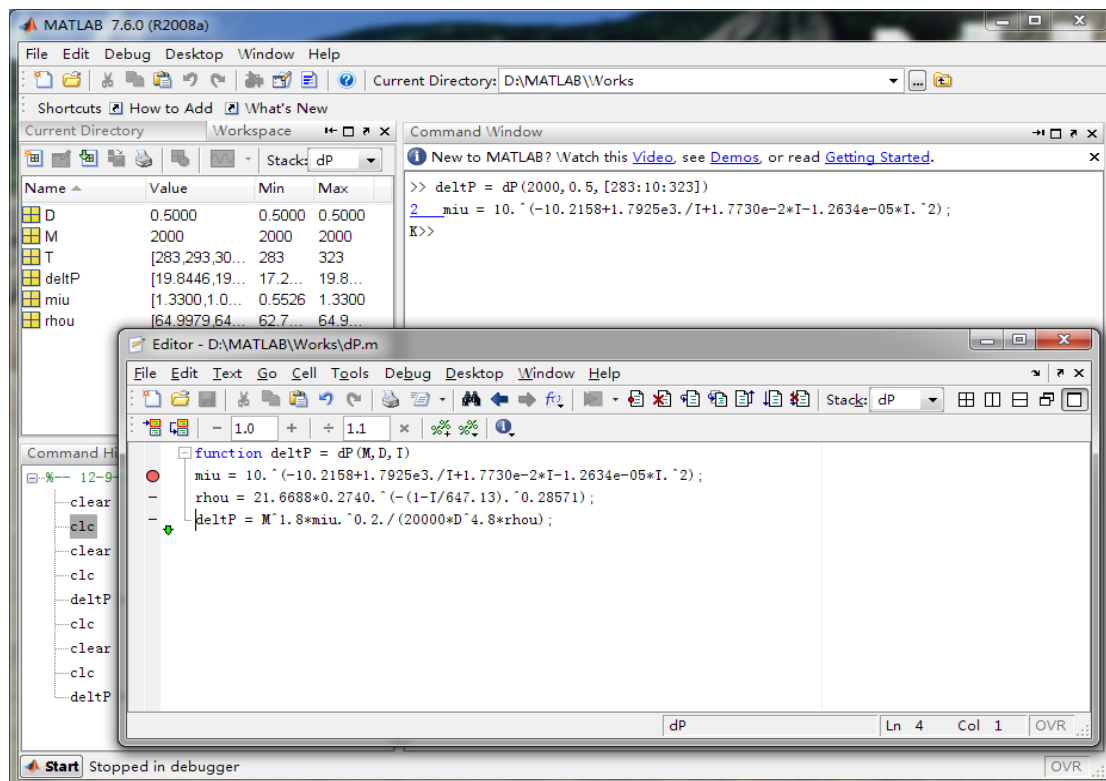


图 16 函数最后一行命令执行完毕

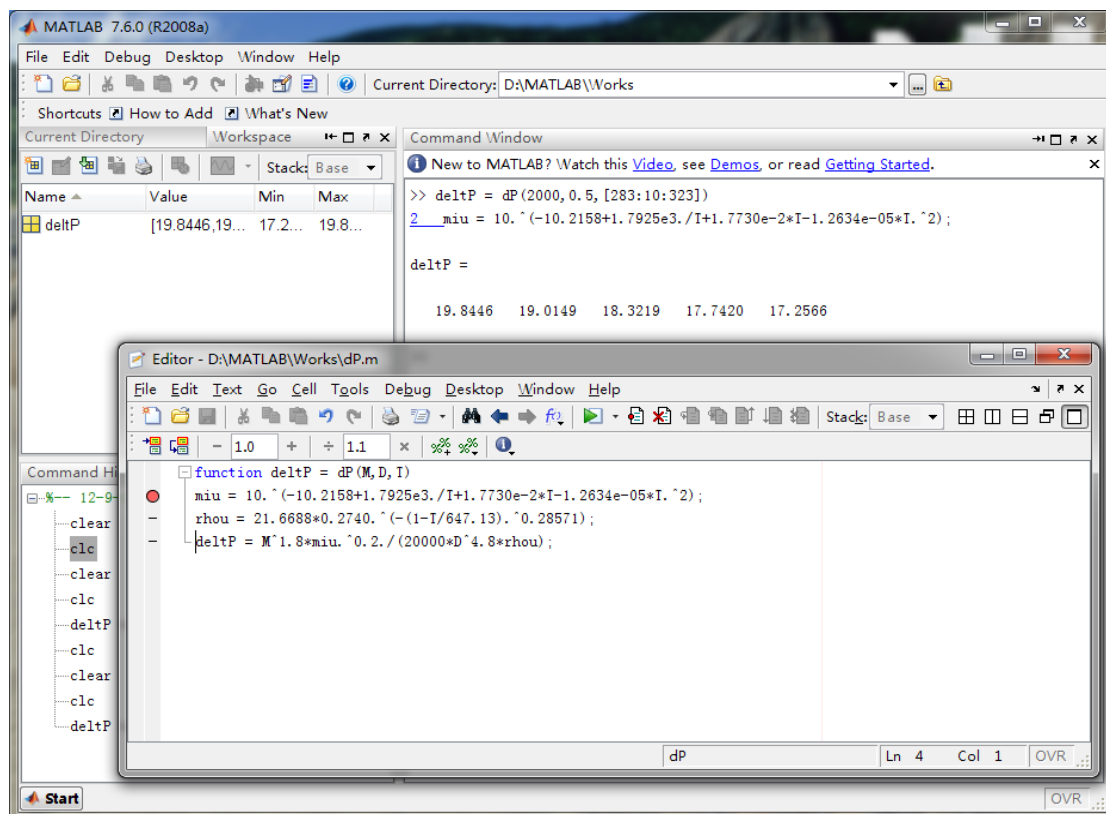


图 17 退出函数后的系统界面

比较图 16 与图 17 可以发现，当函数的最后一行命令执行完毕后，函数中的变量在变量空间依然存在；而退出函数后函数中的变量已经在变量空间消失。

1.3 实验内容

1.3.1 计算 10000 以内所有素数的和

- 1) 利用 MATLAB 的 `isprime` 函数判断一个数是否为素数(在命令窗口键入 `doc isprime` 可以了解该函数的详细使用方法)，分别采用 `for` 和 `while` 循环编程计算；
- 2) 不采用循环结构，进行计算。采用 `tic` 和 `toc` 函数计时，并比较与循环结构计算所需时间的差别。

1.3.2 毕达格拉斯三元组的查找

毕达格拉斯三元组是指三个自然数满足 $a^2+b^2=c^2$ ，已知仅存在一个三元组满足 $a+b+c=1000$ 的条件，请编写一个 MATLAB 程序查找 a, b 和 c 的值。与你的同学比较计算时间的差异。请自行学习 `meshgrid` 函数，通过这一函数提高你的程序效率。

1.3.3 根据斜率画不同颜色的直线

通过键盘输入函数 (`input` 函数) 获取直线的横坐标和纵坐标，程序根据坐标计算直线的斜率绘制不同类型的直线。当直线斜率大于 1 时，在图中画出以红色实线表示；当直线斜率等于 1 时，在图中以黑色虚线表示；当直线斜率小于 1 时，在图中以蓝色实线表示。要求图中直线能够直观反映其斜率。