

西北工业大学  
NORTHWESTERN POLYTECHNICAL UNIVERSITY

西北工业大学航空学院  
**MATLAB 软件与应用**  
课程大作业报告

报告名称:	MATLAB 课程总结与实例练习
学 院:	航空学院
班 级:	01011704 班
姓 名:	冯铮浩
学 号:	2017300281
联系方式:	13815233646
任课老师:	宋科老师

西安·西北工业大学

2020 年 6 月 12 日

## 摘 要

本文为《MATLAB 软件与应用》课程的期末结课报告，主要针对本学期所学课程内容进行总结，并对 MATLAB 软件在专业课与科研工作中的应用进行简要介绍，并针对一道凸包构建与面积计算的生活实际问题进行 MATLAB 编程求解与结果分析。主要详细内容如下。

首先，对课程的主要内容（章节）和重点章节的内容要点进行了总结与回顾。第 1 章为 MATLAB 概述及系统环境，第 2 章为 MATLAB 矩阵及其运算，第 3 章为 MATLAB 数据可视化，第 4 章为 MATLAB 程序设计，第 5 章为 MATLAB 数值计算，第 6 章为 MATLAB 符号运算，第 7 章为 MATLAB GUI 设计，第 8 章为 MATLAB Simulink 仿真。

其次，对本学期的六次上机作业过程进行了总结。依次进行了编程出错汇总、解题经验和教训分析，以及对在自主实践过程中发现的问题与解决问题的新命令、新思路、新方法等进行了汇总说明，辅以相关结果展示与资料参考。

接着，结合实际，对课程知识点在专业课或科研工作中的应用进行了回顾与总结。运用到 MATLAB 的专业课主要有大学物理实验、概率论与数理统计、边界元素法、计算方法、飞行动力学（下）等多门科目。在科研工作与大学生创新训练项目开展过程中，本人亦使用了 MATLAB 算法工具箱进行工程优化设计。

然后，利用 MATLAB 解决了实际中的一个问题，即以“篱笆围羊群”为背景的凸包搜索与面积计算问题。在描述问题并说明求解思路后，采用自主算法，利用 MATLAB 对该问题进行程序设计与结果分析，并进一步由本学期课程所学知识，查阅资料，对该问题的 GUI 用户交互界面进行了设计，程序代码见附录。

最后，对本学期线上课程学习进行了总结与未来展望。

**关键词：**《MATLAB 软件与应用》课程；课程总结；应用回顾；实例分析；MATLAB

## 目 录

<b>1 课程内容与要点总结 .....</b>	<b>4</b>
<b>2 上机训练过程总结 .....</b>	<b>6</b>
2.1 编程出错汇总.....	6
2.2 解题经验和教训.....	10
2.3 探索创新点汇总.....	10
<b>3 课程的实际应用 .....</b>	<b>12</b>
3.1 MATLAB 在专业课程中的应用 .....	12
3.2 MATLAB 在科研工作中的应用 .....	14
<b>4 MATLAB 程序实例 .....</b>	<b>15</b>
4.1 问题描述.....	15
4.2 求解思路与实现流程.....	15
4.2.1 核心算法实现.....	16
4.2.2 GUI 交互模块设计 .....	18
4.3 程序编写说明.....	18
4.4 程序运行结果.....	19
<b>5 课程学习感悟 .....</b>	<b>20</b>
<b>6 参考文献 .....</b>	<b>21</b>
<b>附录 .....</b>	<b>21</b>

# 1 课程内容与要点总结

本学期 MATLAB 软件与应用课程的讲解章节与内容要点如下：

## (1) 第 1 章——MATLAB 概述及系统环境

内容要点提炼：

- 了解 MATLAB 的发展概述及其特点；
- 熟悉 MATLAB 软件界面（菜单、命令行窗口、工作区 Workspace、m 文件窗口等）；
- 熟记 MATLAB 基础操作与重要指令，包括
  - 命令行基本操作（如 clear/clear all/clf/clc/edit）；
  - 数据显示格式的设置（如 format long/short e/long e/hex/bank/rat）；
  - 常用标点符号（如 ; / : / ... / %）；
  - 常用文件格式（如 .m/.mat）；
  - 常用的系统预设变量（如 pi/inf/eps）；
  - 基本运算符（如 + - \* / \ ^）；
  - 常用的命令符（如帮助文档 help/doc、查询变量 whos）。

## (2) 第 2 章——MATLAB 矩阵及其运算

内容要点提炼：

- 熟悉常量、变量与数据类型（赋值即定义原则，数据的储存与导入）；
- 掌握矩阵类型与多种定义方式（灵活运用， ; : 等符号，空矩阵）；
- 掌握矩阵的基本操作与运算，包括
  - 矩阵置空（用于删除某行或列，空和等于零不同），矩阵合并；
  - 矩阵元素的访问、提取，如何操作矩阵的一部分片段；
  - 矩阵单下标和全下标的相互转换和内在规则；
  - 矩阵基本运算，特别注意 \* / ^（普通运算符）与 .\* ./ .^（点运算符）的区别；
  - 其他矩阵分析常用函数（如 max/min/mean/sum）；
- 熟悉字符串、元胞与结构定义与操作，包括
  - 字符串（和多行字符串）基本定义，字符串内容中的单引号由两

个连续的单引号来表示；

- 元胞数组、结构数组的概念、基本定义方式和常用函数。

### (3) 第 3 章——MATLAB 数据可视化

内容要点提炼：

- 掌握二维数据绘图模块，包括
  - 掌握二维曲线绘图 `plot` 的基本用法，包括常用参数的设置方法与图形细节控制（线型、颜色、标记点、图片标题、坐标名称、图例、文本等）；
  - 掌握一图多线(`hold on`)，一窗多子图(`subplot`)，双纵坐标(`plotyy`)的重要绘图命令的基本使用；
- 掌握三维数据绘图模块，包括
  - 三维曲线/曲面绘图命令（如 `plot3/mesh/surf`）的基本使用与区分；
  - 一些特殊的三维绘图函数；
  - 图形效果的修饰（即对图像进行色彩、光照、视角、裁剪、空间变换等方面的处理，使图形更加美观）；
  - 图形句柄绘图（体现了面向对象的程序设计概念，为底层绘图，充分体现出 MATLAB 的开发性）。

### (4) 第 4 章——MATLAB 程序设计

内容要点提炼：

- 掌握三种基本程序设计结构，包括
  - 顺序结构（包括输入 `input`，输出 `disp` 等函数）的应用；
  - 选择/分支结构（`if` 语句、`switch` 语句、`try` 语句）的运用；
  - 循环结构（`for` `while` 两种循环结构）的运用；
- 掌握重要流程控制命令（`break/continue/pause/keyboard` 等）；
- 函数文件的书写和调用方法（注释的功能）；
- 函数递归调用、全局变量（`global`）的定义；
- 重视各程序模块与命令的综合运用；

### (5) 第 5 章——MATLAB 数值计算

内容要点提炼：

- 掌握线性方程组求解两种方法（右除号： $x=A\backslash b$  或  $x=\text{inv}(A)*b$ ）；
- 掌握多项式基本操作（定义方式，roots/polyval/polyfit 等函数使用）；
- 掌握求解非线性方程组方法与技巧（solve/fsolve/roots 命令的运用）；
- 掌握插值与拟合常用命令，包括
  - 插值模块：一维插值命令 interp1 的使用方法；
  - 拟合模块：曲线拟合 polyfit 函数；
- 熟悉数值积分 quad/trapz 等函数的基本使用与参数设置。

## (6) 第 6 章——MATLAB 符号运算

内容要点提炼：

- 掌握符号变量/表达式的定义方法，熟练运用 sym/syms 命令；
- 掌握基本符号运算（四则运算、化简命令 subs/simplify/expand）等；
- 掌握符号程（组）及其求解方式，熟练运用 solve 命令，配合 subs 和 double 可间接获得数值解；求解符号微分方程，运用 dsolve 命令；
- 掌握符号微积分等命令（如 limit/diff/int.taylor）基本用法；

## (7) 第 7 章——MATLAB GUI 设计

内容要点提炼：

- 了解 GUI 基本创建方式、文件管理、设计流程；
- 对几个上课所讲的设计实例进行分析；

## (8) 第 8 章——MATLAB Simulink 仿真

内容要点提炼：

- 了解 Simulink 的基本操作与模块选取及封装；
- 针对老师上课所讲小球弹跳能量衰减问题进行 Simulink 仿真测试。

# 2 上机训练过程总结

本部分对本人在六次 MATLAB 上机作业中出现的問題、总结的经验以及在编程实践探索过程中使用的一些新命令、新思路、创新点进行说明。

## 2.1 编程出错汇总

### 1. MATLAB 软件与应用（第 4 次上机作业：实验内容 4）

**题意描述：**编写一个函数 M 文件，其功能如下：没有输入量时，画出单位圆；输入量是大于 2 的自然数  $N$  时，绘制正  $N$  边形，图名应反映显示多边形的真实边数；输入量是“非自然数”时，给出“出错提示”。此外，函数 M 文件应有 H1 行、帮助说明和程序编写人姓名。

**出错原因：**后续回顾检查时，发现实际编程方法与题意表述有偏差。题意要求编程实现函数 M 文件，但是在上机过程中仅实现了 M 脚本文件的编写，即跳过了函数调用环节。虽然能够实现功能一致，但是编程思路与细节有很大不同。

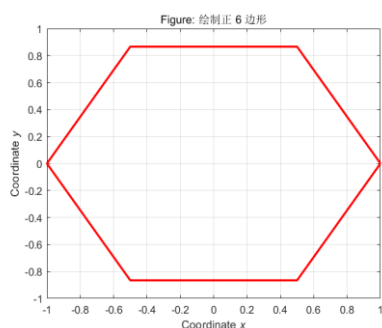
**改正说明：**现将改正后的函数源代码与原始代码进行对比，如表 1 所示。

表 1 MATLAB 源程序订正（第 4 次上机作业：实验内容 4）

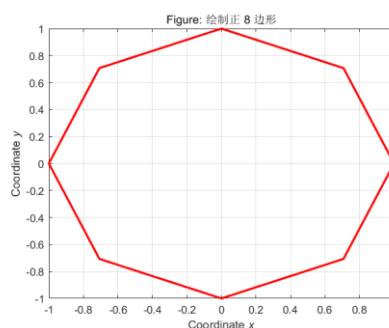
程序编号	P1	预期实现功能	根据输入数据绘制正 N 边形（函数模块）
原始文件	Program_2017300281_Course_004_004.m	修改函数 M 文件	① Polygon_Main.m ② Polygon_Painting.m
<pre> %% MATLAB软件与应用（第4次上机MATLAB程 序源代码：实验内容4） %% 授课老师：宋科老师 %% 姓名：冯铮浩 %% 学号：2017300281 %% 学院：航空学院 %% 完成时间：2020年4月15日  % 程序初始化 clear all clc format long %% Problem_004（实验内容4，计算K值） while (1)     % 从键盘输入（使用input函数）一个参数N     Para=input('请输入参数N（如需退出程序，请输 入-1）：');     fprintf('\n');     if (Para== -1)         fprintf('程序运行结束！\n');         fprintf('\n');         break;     end     if (~isempty(Para)) % Case 2:输入是大于2的自然数N时，绘制正N边形         if ((Para&gt;2) &amp;&amp; (fix(Para)==Para))             fprintf('输入值为大于2的自然数，绘制 正%d边形如图：\n',Para);             fprintf('\n');             theta=0:((2*pi)/Para):(2*pi);             [x,y]=pol2cart(theta,1);             plot(x,y,'r-','LineWidth',2);             xlabel('Coordinate \itx'),ylabel('Coordinate \ity');         end     end end </pre>		<pre> function Polygon_Main     % Function Name: Polygon_Main 主要实现功     能：对第4次上机实验内容4绘制正N边形的函数M文     件进行调用     % 程序编写人姓名：冯铮浩     % 学号：2017300281      % 程序初始化     clear all;clf;clc     format long      % 主程序编写     while (1)         % 从键盘输入（使用input函数）一个参数N         Para=input('请输入参数N（如需退出程序，         请输入-1）：');         fprintf('\n');         if (Para== -1)             fprintf('程序运行结束！\n');             fprintf('\n');             break;         else             Polygon_Painting(Para);         end     end end  function Polygon_Painting(Para)     % Function Name: Polygon_Painting 主要实现功     能：根据不同输入量绘制正N边形     % 输入： Para——正N边形的边数（应为大于2     的自然数N）     % 输出：带有正N边形的图案图窗     % 函数程序编写人姓名：冯铮浩 </pre>	

<pre> % 添加对应多边形边数的标题 title(['Figure: 绘制正 ',num2str(Para),' 边形']); grid on; box on; else fprintf('错误：输入量非大于2的自然 数！请重新输入！\n'); fprintf('\n'); end else % Case 1:没有输入量时，绘制单位圆 fprintf('没有输入量，绘制单位圆如图：\n'); fprintf('\n'); theta=0:(pi/60):(2*pi); x=cos(theta); y=sin(theta); plot(x,y,'b-','LineWidth',2); xlabel('Coordinate \itx'),ylabel('Coordinate \ity'),title('Figure: 绘制单位圆'); grid on; box on; end end </pre>	<pre> % 学号：2017300281 % 核心条件语句实现 if (~isempty(Para)) % Case 2:输入量是大于2的自然数N时，绘 制正N边形 if ((Para&gt;2) &amp;&amp; (fix(Para)==Para)) fprintf('输入值为大于2的自然数，绘制 正%d边形如图：\n',Para); fprintf('\n'); theta=0:((2*pi)/Para):(2*pi); [x,y]=pol2cart(theta,1); plot(x,y,'r-','LineWidth',2); xlabel('Coordinate \itx'),ylabel('Coordinate \ity'); % 添加对应多边形边数的标题 title(['Figure: 绘制正 ',num2str(Para),' 边形']); grid on; box on; else fprintf('错误：输入量非大于2的自然数！ 请重新输入！\n'); fprintf('\n'); end else % Case 1:没有输入量时，绘制单位圆 fprintf('没有输入量，绘制单位圆如图：\n'); fprintf('\n'); theta=0:(pi/60):(2*pi); x=cos(theta); y=sin(theta); plot(x,y,'b-','LineWidth',2); xlabel('Coordinate \itx'),ylabel('Coordinate \ity'),title('Figure: 绘制单位圆'); grid on; box on; end end </pre>
--	--

对于部分输入测试量 ( $N=6,8$ )，绘制的正  $N$  边形图形如图 1 所示。



(a) Case:  $N=6$



(b) Case:  $N=8$

图 1 绘制的正  $N$  边形显示 (Case:  $N=6,8$ )



经过调试，两组程序均成功通过测试。另外，程序还实现了循环主程序功能，如需退出程序，输入-1 即可，否则将继续询问获取输入参数，更新绘制图形。

由上表中两组程序对比，可观察到第二组运用函数 M 文件实现功能，模块更加独立与清晰，充分体现出使用 MATLAB 函数的优势。

## 2. MATLAB 软件与应用（第 3 次上机作业：习题 12）

**问题描述：**绘制一条曲线，采用搜索法根据线型属性获取曲线句柄，并修改曲线线型。

**出错原因：**首次查看问题时，对于题意描述中的“搜索法”存在一定疑惑，并对于曲线句柄操作不太熟悉，编写程序存在一定困难。

**改正说明：**经过对上课课件的推敲，使用 MATLAB 的 Help Document，以及网络资料的学习，最终解决了相关问题。首先绘制多条曲线（这里我设置为三条线型各不相同的三角函数曲线）；然后采用指令 `fig_line = get(gca, 'Children')`，获得所有包含曲线的句柄集合；随后利用循环语句，搜索线型为指定原始类型（如虚线）的曲线，记录所有满足搜索条件的曲线序号，并返回搜索目标曲线句柄到 `fig_target_Line`；最后，将句柄 `fig_target_Line` 对应曲线线性（虚线）修改为指定目标类型（如直线），完成曲线句柄修改。

## 3. MATLAB 软件与应用（第 5 次上机作业：习题 3）双三次插值问题

**问题描述：**设节点  $(x, y, z)$  中的  $x = -3:0.5:3, y = x, z = 7 - 3x^3e^{-x^2-y^2}$ ，作  $z$  在插值点  $x = -3.9:0.5:5, y = -4.9:0.5:4.5$  处的二元样条插值、双三次插值结果。

在实际编程实现中，发现双三次方法（cubic）插值网格外插节点出现 NaN。书本上例题 5.20 亦出现类似问题但未指明。

**原因分析：**在 Help Document<sup>[1]</sup>中，有关 `interp2` 外插功能说明如图 2 所示。

`Vq = interp2( __, method, extrapolval)` also specifies `extrapolval`, a scalar value that is assigned to all queries that lie outside the domain of the sample points.

If you omit the `extrapolval` argument for queries outside the domain of the sample points, then based on the `method` argument `interp2` returns one of the following:

- Extrapolated values for the 'spline' and 'makima' methods
- NaN values for other interpolation methods

图 2 插值指令 `interp2` 指令输出结果

由此可见，`interp2` 的外插功能（`extraval`）仅限于二元样条插值（`spline`）与修改的 Akima 立方 Hermite 插值（`makima`），其余插值方法不支持二维外插节点。

**解决策略：**可使用修改的 Akima 立方 Hermite 插值（makima）进行外插操作，多种插值方式的输出结果如图 3 所示。

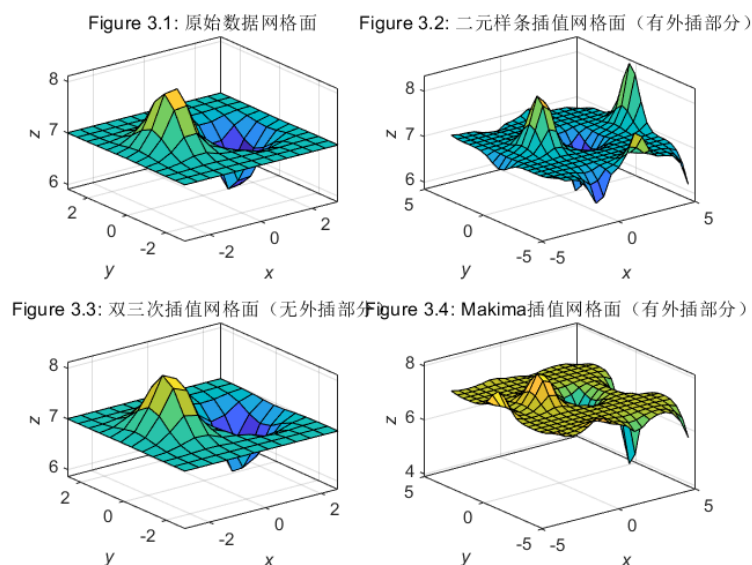


图 3 多种二维插值方法图形可视化结果

## 2.2 解题经验和教训

上机作业训练中，自主总结三条经验和教训如下：

1. 首先必须读清题干，明确各项题目要求，特别是容易忽视的编程方式要求，如编写 M 文件与 M 函数文件就有明显差异，未来需要特别注意；
2. 对于概念模糊的问题，需要通过多种方式，自主解决，如复习课件、利用网络资源，并需要勤于使用 MATLAB 自带的 Help 功能，充分利用软件优势。
3. 解题方法可能有多种，不能拘泥于一种思路而停滞不前，需要勤于动手实践，探索更好、更高效的实现方式，将每道题目的知识点与应用面弄清、弄透。

## 2.3 探索创新点汇总

在本学期 MATLAB 课程的上机练习与课下应用中，由于 MATLAB 软件版本更新升级周期较短，许多函数、命令均处在不断地调整优化过程中，故本部分列举一些我自己在编程解决上机作业时探索的新方法、创新点。

### 1. MATLAB 软件与应用（第 5 次上机作业：习题 2）新命令 vpasolve

**问题描述：**分别采用 solve、roots 求解非线性方程  $x^4 - 3x^3 + 2x^2 - 2x + 5 = 0$ 。

**改进策略：**第一种采用 solve 命令求解非线性方程的程序可以编写如下，syms x; s1 = solve(x^4-3\*x^3+2\*x^2-2\*x+5==0); s1 = double(s1); 在较新 MATLAB 版本中，亦可采用 vpasolve 命令，不仅可以在数值计算中起到与 solve 指令相同的效果，并且更加适合数值计算，而 solve 作为一种符号运算命令，并非万能，对于单方程一般可求解，对于复杂方程组求解有时会失效。

## 2. 自查并帮助班级同学解决一项课件中求解非线性方程（组）fsolve 命令问题。

**问题描述：**编写函数 myeq001.m 的非线性方程组后，调用 fsolve 指令时，发现 y = fsolve('myeq001',[1 1],1)报错，具体如图 4 所示（版本 MATLAB R2019b）：

```
>> y=fsolve('myeq001',[1,1],1)
无法执行赋值，因为此类型的变量不支持使用点进行索引。

出错 fsolve (line 341)
options.FinDiffType = optimget(options,'FinDiffType',defaultopt,'fast',allDefaultOpts);
```

图 4 使用 fsolve 指令报错信息

**解决策略：**fsolve 的基本指令 x = fsolve('fun', X0, options)，第一种解决方法，可将 options 去除或置空，即保持默认，比较简便，如图 5 所示。

```
>> y=fsolve('myeq001',[1,1])

Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>

y =

    0.6488    0.5226
```

图 5 修改后 fsolve 指令输出结果

第二种解决方法，可对 options 进行单独设置。fsolve 本质上是一种基于迭代的数值求解指令。例如根据官方帮助文档<sup>[2]</sup>，options 为 Optimization options 的简写，可设置各种数值求解算法（比如梯度下降法、有限差分法等），并设置迭代最大次数，以提升精度与收敛稳定性，例如以下指令

```
options = optimoptions('fsolve','FiniteDifferenceType','central');
```

即采用中心有限差分格式进行数值求解。

#### 4. MATLAB 软件与应用（第 6 次上机作业：习题 5）Taylor 展开式命令

**问题描述：**将函数  $\sin^2 x$  展开成  $x$  的泰勒级数，取前 4 项写出展开式。课本第六章 6.4.2 节给出的 Taylor 展开调用格式为

$$\text{taylor}(f, v, n, v_0)$$

其中,  $f$  表示待展开的符号函数;  $v$  表示展开的符号变量;  $n$  表示展开级数的项数, 省略时  $n$  默认为 5;  $v_0$  表示函数  $f$  在  $v = v_0$  处展开, 省略时  $v_0 = 0$ , 即在 0 处展开。

在实际调试过程中, 发现该指令已经失效, 可能是由于版本较低命令已更新。

**解决策略：**

- ① 方式一：直接利用新版本 Taylor 展开指令（取 10 阶精度截断）

$$f\_taylor = \text{taylor}(f, x, 'Order', 10)$$

- ② 方式二：调用 MuPAD 引擎计算展开<sup>[3]</sup>（取 10 阶精度截断）

$$f\_MuPAD = \text{evalin}(\text{symengine}, 'series((\sin(x))^2, x=0, 8)')$$

具体截取精度可根据要求调整确定, 另外可采用 `pretty` 指令优化结果显示。

### 3 课程的实际应用

#### 3.1 MATLAB 在专业课程中的应用

MATLAB 作为一款高级数学计算软件, 以其丰富智能的函数库、工具箱 (Toolbox)、帮助文档, 简约清晰的交互界面、优秀的数值算法以及强大的图形可视化能力, 而备受工程师青睐。自本科入学以来, 我在众多专业课程中均使用 MATLAB 作为核心工具实现相关需求。现按时间顺序将部分应用实例列举如下:

##### 1. 《大学物理实验》课程

**应用类型：**实验数据处理。

**应用知识：**最小二乘法、多项式曲线拟合, 图形可视化。

**MATLAB 代码长度：**50 行左右。

**应用简述：**在“放电法测电阻”实验测量得到电量  $Q$  随时间  $t$  的变化曲线 (如图 6 所示) 后, 需要对图线进行“化曲为直”处理, 得到“ $\ln Q - t$  曲线”, 再利用最小二乘法进行直线拟合, 利用 MATLAB 的多项式拟合指令, 或采用 APP 中的 Curve Fitting 工具箱可以方便绘制图线并分析斜率, 误差等数据, 如图 7 所示。

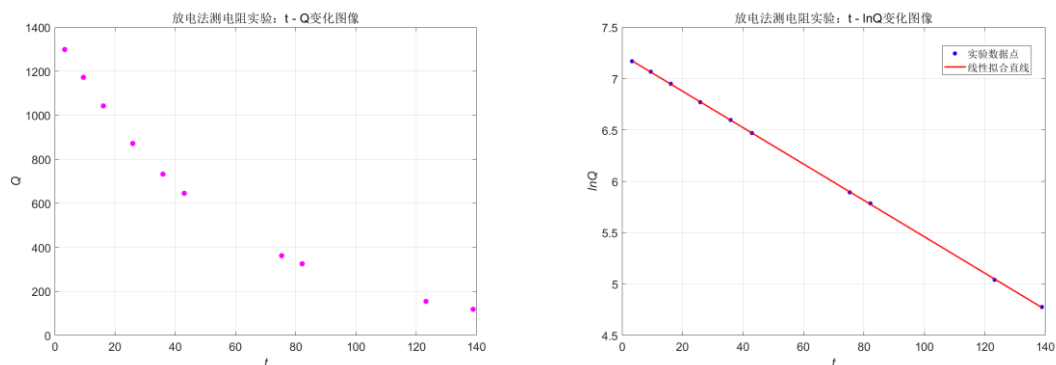
(a) 原始  $Q-t$  数据点(b)  $\ln Q-t$  数据点与拟合直线

图7 大学物理实验“放电法测电阻”MATLAB 数据图形处理

## 2. 《概率论与数理统计》课程

**应用类型:** 定理验证。

**应用知识:** 随机数指令（如 `binopdf`, `poisspdf`）；图形可视化（如直方图 `hist`）。

**MATLAB 代码长度:** 150 行左右。

**应用简述:** 利用 MATLAB 软件对概率论与数理统计课程中所学的泊松定理、大数定律和中心极限定理进行随机仿真模拟，并通过图形将模拟结果直观、动态地演示出来，从而对这两个定理所反映的本质内容给出直观的解释和说明。例如，对中心极限定理的 MATLAB 验证设计流程图与部分结果分别如图 8,9 所示。



图8 中心极限定理仿真验证算法 MATLAB 实现流程图

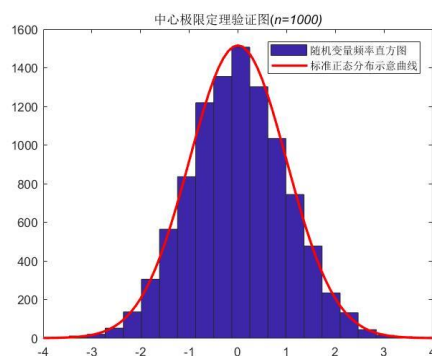


图9 概率论与数理统计“中心极限定理”MATLAB 验证结果图形可视化

### 3. 《边界元素法》课程

**应用类型：**程序设计、综合应用。

**应用知识：**程序设计、矩阵运算、数值计算、图形可视化、文件输入输出等。

**MATLAB 代码长度：**1000 行左右。

**应用简述：**利用 MATLAB 软件编写二维圆柱声散射计算程序，总共 3 个源程序，分别完成圆柱周线布置网格收敛性分析、不同波数  $k$  情形的声场空间指向性分布曲线绘制对比，以及二维声场强度分布云图计算，可配置在较高版本的 MATLAB 的工作目录下直接运行，查看输出结果。每个源程序均有结果输出模块，直接写入.dat 文件，可直接打开 Tecplot 模板进行数据导入与图形可视化，显示声场空间指向性分布曲线、二维声场强度分布以及数值结果与理论解的对比等信息，便于进行数据定性、定量分析。程序结果与分析受到该课程老师肯定。

实践证明，MATLAB 的矩阵计算能力非常强大，同时软件内部集成众多科学计算函数，如三类 Bessel 函数，需要时仅需几行指令便可快速实现，十分方便，而本人利用 C 语言和 FORTRAN 语言编写相同模块，虽然运行效率提高，但是程序复杂程度与不稳定性均显著增加。这也进一步反映出 MATLAB 的优势。

此外，在《计算方法》、《飞行动力学（下）》等课程中，我自主使用 MATLAB，并运用课堂所学知识，充分发挥其函数丰富、矩阵运算强大、图形可视化便捷的特点，完成了相关课程编程要求。同时在丰富的数学建模竞赛题目中，我作为编程队员，几乎每一道题目均使用 MATLAB 来解决部分任务，大大提升了解决问题的效率，也取得了较好的效果。具体应用类型、应用方式不再描述。

### 3.2 MATLAB 在科研工作中的应用

在航空学院“顶峰计划”本科生科研项目与大学生创新项目中，我主要参加的项目与课题均与计算流体力学（CFD）与气动优化（Aerodynamic Optimization）相关。特别是在气动优化设计的工作中，我利用 MATLAB 自带的智能算法工具箱，包括神经网络工具箱（Neural Network Toolbox）、遗传算法工具箱（Genetic Algorithm），进行翼型的优化设计。下面展示的是我对超声速 Licher 双翼翼型进行基于 RBF 神经网络的气动代理模型建模以及遗传算法迭代优化的部分结果展示，且均用 MATLAB 实现，并取得了较好效果，如图 10,11 所示。

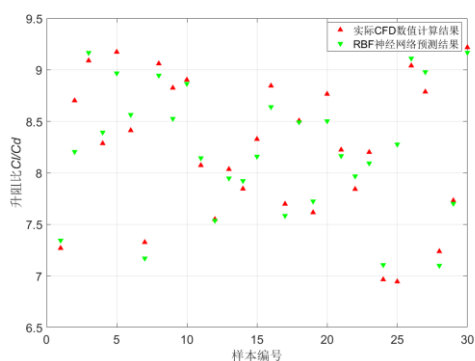


图 10 RBF 径向基神经网络代理模型预测结果与原始 CFD 数据对比验证图

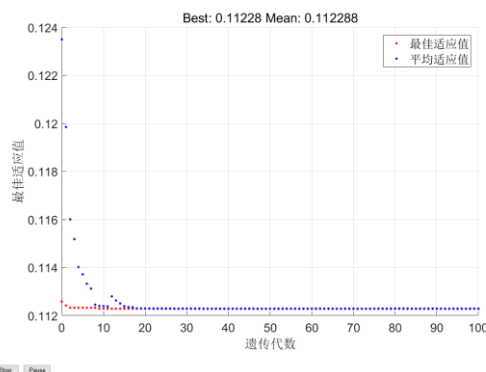


图 11 MATLAB 遗传算法迭代进化结果图

## 4 MATLAB 程序实例

下面对一道实际生活问题进行问题描述、求解思路分析，并使用 MATLAB 编程实现，进行程序说明，最后进行结果分析工作。

### 4.1 问题描述

在日常生活中，与“凸包”有关的情景与问题很多。例如找到将一群羊完整围在一个区域的封闭篱笆边界，即可认为是以羊群个体为节点，寻找一个以某些羊为节点的多边形凸包，使得羊群被完全包围。凸包（Convex Hull）更是一个计算几何（图形学）中的重要概念，在图论研究分析中有重要作用。

以羊群为对象，考虑下面的问题：在  $100m \times 100m$  的草原（可抽象为二维平面）上，随机分布 100 只羊，现需要寻找包围羊群的最小篱笆边界（选取羊群中的部分个体为篱笆节点），同时求出篱笆包围的区域总面积。

### 4.2 求解思路与实现流程

首先进行问题的数学模型抽象与简化。羊群为实际背景的研究对象，在抽象数学模型中，可简化成给定二维平面上的点，则问题转化为在平面上随机产生 100 个点，然后作出闭包曲线，并输出该凸多边形面积。

点集  $Q$  的凸包是指一个最小凸多边形，满足  $Q$  中的点或者在多边形边上或者在其内，示意图如图 12 所示。



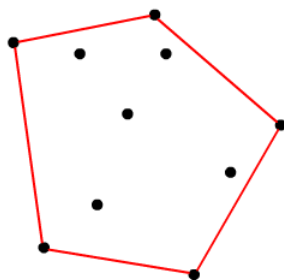


图 12 多点集合凸包构造示意图

求解凸包比较成熟的凸包算法是 Graham 扫描法<sup>[4]</sup>和 Jarvis 步进法<sup>[5]</sup>。

MATLAB 亦自带有凸包 **convhull** 函数库，具体功能详见文档<sup>[6]</sup>，主要函数如下：

```
k = convhull(P) % 计算矩阵 P 中点的二维或三维凸包
k = convhull(x,y) % 计算列向量 x 和 y 中点的二维凸包
k = convhull(x,y,z) % 计算列向量 x、y 和 z 中点的三维凸包
k = convhull(____,'Simplify',tf) % 指定是否删除不影响凸包面积或体积的
                                     顶点。默认情况下，tf 为 false
[k,av] = convhull(____) % av 可计算凸包的面积或体积
```

这里不采用 MATLAB 自带现成函数，而是选择自编 MATLAB 程序实现算法。

下面对寻找凸包并计算凸包面积的算法思路与实现流程进行说明，并指出编程过程中的注意点。同时，根据课堂所学知识，利用 MATLAB 实现多边形凸包的 GUI 界面设计，方便参数修改与结果观察。

#### 4.2.1 核心算法实现

首先针对问题进行凸包计算的核心算法实现。采用模块化编程思想，利用 MATLAB 分步实现各指定任务，并进行模块数据交互与流程设计。

##### ➤ 模块一：程序初始化，生成 100 个随机数据点坐标，调用 **rand** 函数。

首先进行程序初始化。利用 **rand** 函数生成 100 个 [0,1] 区间的数字，并乘以 100，作为节点的  $x$  坐标， $y$  坐标生成方法类似。同时，记录 100 个数据点编号。

##### ➤ 模块二：绘制原始平面点集图像，绘制 100 个随机数据点。

利用 **plot** 指令实现 100 个原始随机数据点坐标的图形可视化。

##### ➤ 模块三：搜索凸包核心算法设计

- **步骤 1：**首先寻找最小  $x$  坐标点，易知该点必为凸包点，并以此点作为凸包起点。可采用循环控制，利用“打擂台”思想，让各个  $x$  坐标与当前最小值（初始可取一个较大值，如 999999 或 INF）比较，不断更新当



前坐标的最小值与其对应节点编号，最终得到最小  $x$  坐标点及其编号。

- **步骤 2:** 设置最小  $x$  坐标点为第一个凸包点（起点  $S$ ）开始，搜索下一个凸包点，坐标为  $(x_1, y_1)$ 。以  $i$  为指针扫描除起点外的所有坐标点，设当前点为  $T$ ，坐标为  $(x_2, y_2)$ 。构造过  $S, T$  两点直线，并化成一般形式，即

$$l_{ST} : Ax + By + C = 0$$

其中， $A = y_2 - y_1$ ， $B = x_1 - x_2$ ， $C = (x_2 - x_1) \cdot y_1 - (y_2 - y_1) \cdot x_1$ 。

创建  $ST$  直线完毕后，判断剩下的 98 个点（除起点、终点外，利用禁忌表标记）是否满足均在直线的同一侧，若成立，则当前  $i$  遍历点即为凸包点；若不成立，则继续使  $i$  指针遍历下一个点。利用直线一般式方程左边项的正负，即可知其他点是在直线的同一侧还是异侧。具体判断时，仍可采用循环控制，初始化累乘变量为 1，将直线外的节点坐标代入一般式方程，并将其结果累乘，若判断出现结果变号，则判断出现当前节点与已扫描节点在  $ST$  直线异侧，可直接使用 `break` 跳出循环，开始扫描下一个点，从而大幅提升程序效率。

- **步骤 3:** 按照上述步骤穷举  $i$  变量，通过扫描必可获得下一个凸包点，现将此凸包点作为新的起点，初始化变量后，进行下一次搜索，依次类推。
- **步骤 4:** 若搜索得到的下一凸包点为初始起点（ $x$  坐标最小点），则退出搜索循环，完成寻找凸包操作。最后实现图形可视化。

**注意点：**如何判断所有其他点均在这条直线  $ST$  一侧为本模块实现关键，同时需要实时保存与更新凸包点集，注意次序，以方便图形可视化。

➤ **模块四：凸包面积计算，需要将凸包多边形分解成若干个三角形，分别求解面积并叠加。**

对于凸包面积计算模块，可利用三角形面积计算海伦-秦九韶公式

$$S = \sqrt{p(p-a)(p-b)(p-c)}, \text{ 其中 } p = \frac{a+b+c}{2}.$$

对于已经求解出的凸包，由于扫描的连续性，故其存放顺序应为依次有序的（按顺时针或者逆时针一个方向排列）。考虑使用起始两个凸包点构成的直线为三角形公共底边，利用循环指令，依次将剩余的凸包点作为三角形顶点，利用海

伦-秦九韶公式计算三角形面积，并设置累加器进行累加，则得到凸包所围面积。

### 4.2.2 GUI 交互模块设计

GUI 界面不仅能够集成程序算法，完成所需的任务，还能提供交互功能，进行参数修改。在完成核心算法设计后，将各模块转化为子函数，即可成为 GUI 内嵌模块函数，再添加数据交互与字符提示，便可完成 GUI 界面设计。

首先进行 GUI 界面功能按键布置。本文设计界面提供三个主要模块，分别为凸包构建模块、结果输出模块及图形可视化窗口。其中凸包构建模块添加输入点数文本框，可用键盘输入所需生成平面随机数据点个数。具体布置如图 13 所示。



图 13 GUI 界面布置示意图

接着，对各个按钮、文本框对应的 Callback 函数进行算法编写，并使用全局变量（global）进行模块数据共享，经过模块检查调试后，GUI 设计完成。

## 4.3 程序编写说明

运行环境：Windows 10 64 位操作系统；软件版本：MATLAB R2019b

本凸包设计与面积计算程序及其 GUI 交互界面设计，均在 MATLAB R2019b 环境下调试编写，共包含两个 .m 文件主程序与一个 .fig 图形文件，具体名称为：

### 1. Program\_Convex\_Polygon\_Closure\_Design.m

求平面上 100 个点的凸多边形闭包及该多边形面积（非函数算法设计）。

### 2. Program\_Polygon\_Painting.m 与 Program\_Polygon\_Painting.fig

利用 MATLAB 实现多边形凸包构造结果 GUI 设计，运行后生成图形窗口。

以上源程序可配置在高版本 MATLAB 直接运行，代码见附录，多处有注释。

## 4.4 程序运行结果

首先实现基本算法，凸包搜索与面积计算的程序运行示意图如图 14 所示。

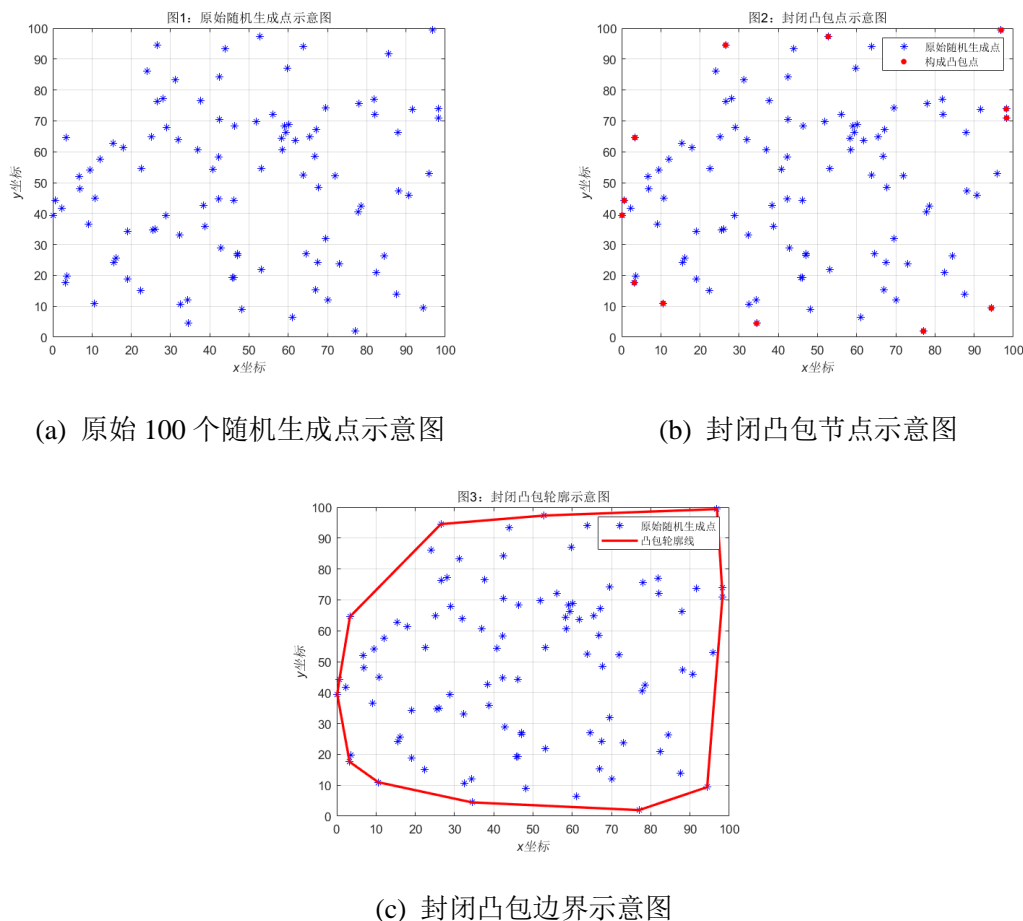
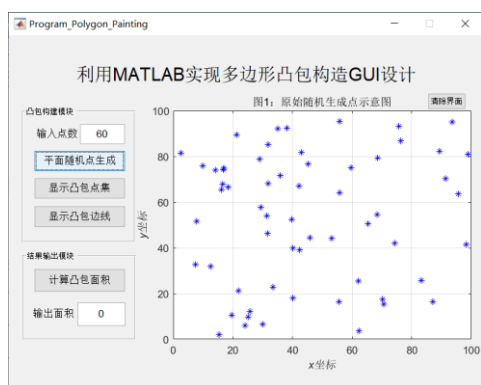


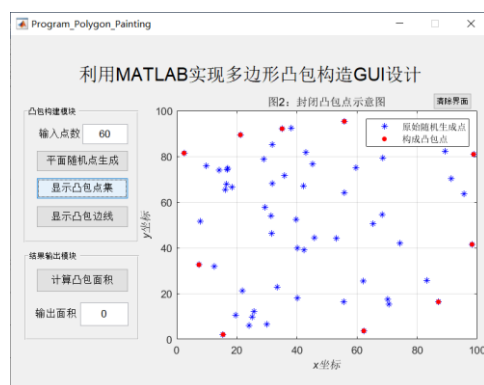
图 14 凸包搜索与面积计算问题程序运行结果示意图（100 个平面随机点）

输出凸包面积为 8214.79。观察上图，发现凸包节点与边界均满足要求，经过多次调试以及与函数输出结果对比，可以判断设计算法较为可靠，结果正确。

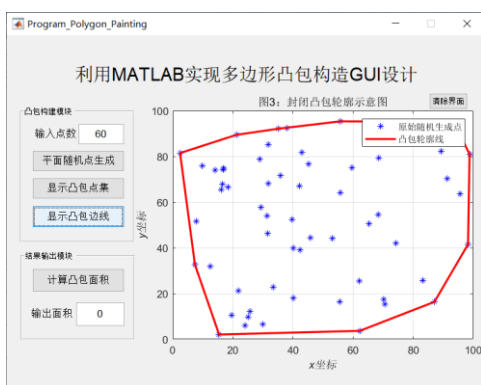
接着运行 GUI 界面，在凸包构造模块中“输入点数”后的框内直接用键盘键入随机点个数，这里选择键入 60，表示期望在平面上生成 60 个随机坐标点，并构造其凸包。再依次点击“平面随机点生成”“显示凸包点集”“显示凸包边界”三个命令按钮，可以观察凸包建立与在右侧坐标轴内的图形可视化过程，如图 15 所示。接着可以点击结果输出模块中的“计算凸包面积”按钮，则“输出面积”后的文本框中可显示当前构造凸包的面积。在 GUI 界面运行时，可随时点击右侧坐标轴右上角的“清除界面”按钮，将界面坐标轴和输入输出框初始化。



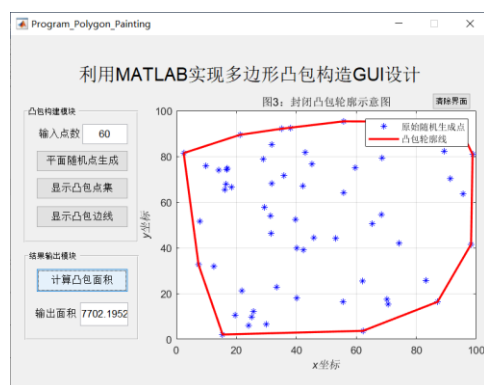
(a) 输入点数 60，点击“平面生成点”界面



(b) 继续点击“显示凸包点集”界面



(c) 继续点击“显示凸包边界”界面



(c) 继续点击“计算凸包面积”界面

图 15 凸包搜索与面积计算问题程序 GUI 运行结果示意图（输入 60 个平面随机点）

## 5 课程学习感悟

在本学期《MATLAB 软件与应用》的线上课程学习中，通过宋老师系统的课堂理论讲授与自主实习环节，特别是在六次上机作业实战训练的过程中，我充分感受到 MATLAB 软件的无穷魅力与应用价值。

MATLAB 是“矩阵实验室”的简称，其在工程学科的科学计算领域具有广阔的应用，特点鲜明，内容丰富，需要我在今后的课程学习、科研工作中继续发掘其潜力，用好这款优秀的软件，使其成为我学习之路上的“良师益友”。

在撰写课程总结报告的过程中，我回顾总结、查阅资料、思考并解决问题的能力得到了大幅的提升，也使得一份踏实、负责、精益求精的科研态度在我心中油然而生。但是，报告中的内容细节还有总结不够充分，探索还不够深入。在今后的学习生活中，我也会更加注重将课堂学习与亲身实践相结合，培养能力与兴趣，为未来的科研之路打下更加坚实的基础！非常感谢宋老师对我的辛勤指导！

## 6 参考文献

- [1] <https://ww2.mathworks.cn/help/matlab/ref/interp2.html>
- [2] <https://ww2.mathworks.cn/help/optim/ug/fsolve.html>
- [3] <https://ww2.mathworks.cn/help/symbolic/evalin.html>
- [4] <https://www.cnblogs.com/wpbing/p/9456240.html>
- [5] [https://blog.csdn.net/m0\\_37846371/article/details/76229526](https://blog.csdn.net/m0_37846371/article/details/76229526)
- [6] <https://ww2.mathworks.cn/help/matlab/ref/convhull.html>

## 附录

本报告中全部 MATLAB 源程序代码、实现功能及输出文件说明如下：

程序编号	P1	实现功能	凸包搜索与面积计算程序（非函数实现）
程序名称	Program_Convex_Polygon_Closure_Design.m		
<pre>%% Program: Finding the closure and area of convex polygon of n points on a plane %% 求平面上n个点的凸多边形闭包及该多边形面积（非函数算法设计）  %% 数据生成及初始化 clear all clc clf n=100; x=100*rand(1,n); % 在[0,100]之内随机生成x坐标点（可修改） y=100*rand(1,n); % 在[0,100]之内随机生成y坐标点（可修改）  %% 绘制原始平面点集图像 % 绘制Figure 1： 原始随机生成点示意图 % 绘制生成的100个随机点 plot(x,y,'b*','MarkerSize',6); xlabel('\itx坐标'),ylabel('\ity坐标'),title('图1： 原始随机生成点示意图'); grid on box on  %% 寻找凸包核心算法设计 % 设置凸包点集标志，1为满足条件点，0为非满足条件点 p_flag=zeros(1,n); xmax=10000; % 寻找最小x坐标点，此点必定是凸包点，以此点作为凸包起点</pre>			

```

for i=1:n
    if (x(i)<xmax) && (p_flag(i)==0)
        pj=i; % 记录最小x坐标点的点序号
        xmax=x(i); % 更新最小的xmax值
    end
end
p_flag(pj)=1;

flag_1=0; % 设置搜索停止判断标志：从起点开始搜索下一个点，判断是否已经找到凸包，
程序退出标志为下一个点是起点
st=pj; % 起点编号
cnt=1; % 记录凸包点数，初始状态已有一个点加入
pp=zeros(1,n); % pp数组记录按顺序排列（顺时针或逆时针）的凸包点集，方便后续画图
pp(1)=pj; % 设置最小x坐标点为第一个凸包点（起点）

while (flag_1~=1) % 若不满足退出条件，则进行while循环

    % 提取起点的x,y坐标为(x1,y1)
    x1=x(st);
    y1=y(st);

    % 设置i变量遍历所有点坐标,寻找下一个在凸包内的点
    for i=1:n

        if (i~=st) % 如果下一个点不在凸包内
            % 两点构造直线，提取当前点的x,y坐标为(x2,y2)
            x2=x(i);
            y2=y(i);
            % 直线方程已化简为Ax+By+C=0的形式
            A=(y2-y1);
            B=(x1-x2);
            C=((x2-x1)*y1-(y2-y1)*x1);

            tot=1; % 关键点1：tot用以判断是否该直线为凸包边线
            % 由于将在落在直线两边的点坐标代入直线方程后，Ax+By+C的值异
            % 号
            % 利用该特点，设置累乘器tot，逐次代入这些点坐标，只要检索到
            tot<=0，则可排除该点为凸包点；
            % 若其他98个点依次代入后tot始终为正，则该点为凸包点，标志为
            flag_2=1

            flag_2=1; % 判断编号点是否为凸包点的标志
            for j=1:n % 遍历所有点集检验是否满足凸包条件
                if (j~=st) && (j~=i) % 排除构造直线的两个点，因为其使得Ax+By+C=0将

```

使tot=0，无法判断结果

```

        xj=x(j);
        yj=y(j);
        t=(A*xj)+(B*yj)+C;
        % t=t*100;
        % 考虑由于tot每次乘的量若很小的情况，或者结果非常接近0
        的情况时，可乘以一个放大系数（可选择）
        tot=tot*t;
        if (tot<=0)
            flag_2=0;
            break; % 剪枝操作，提高程序效率
        end
    end
end

    % 若成功，即找到凸包点
    if (flag_2==1)
        en=i; % 记录当前点编号
        p_flag(en)=1; % 标记该点已经进入凸包集合
        cnt=cnt+1; % 凸包点个数加1
        pp(cnt)=en;
        st=en; % 关键点2：将找到的新的下一个凸包点作为新的起点，循环再寻找下一个凸包点
        if (en==pj) % 关键点3：凸包封闭标志，也即程序退出标志为：寻找到的新的凸包点回到了起点
            flag_1=1; % 完成凸包封闭操作，flag_1置1，退出整个迭代程序
            break;
        end
    end
end

end

    end % 结束i（编号）循环
end % 结束while循环

%% 绘制封闭凸包（点）示意图
% 绘制Figure 2：封闭凸包点示意图
% 完成凸包求解，记录各凸包点的x,y坐标（无序）——显示凸包点
ppx=zeros(1,cnt-1); % 各凸包点x坐标（无序）
ppy=zeros(1,cnt-1); % 各凸包点y坐标（无序）
cntt=0;
for k=1:n
    if (p_flag(k)==1) % 若该点为凸包点
        cntt=cntt+1;
    end
end

```

```

        ppx(cntt)=x(k);
        ppy(cntt)=y(k);
    end
end
% 新建图层
figure;
hold on
% 绘制生成的100个随机点
plot(x,y,'b*','MarkerSize',6);
% 绘制凸包点集
plot(ppx,ppy,'r.','MarkerSize',16);
% 添加图例、坐标信息
legend('原始随机生成点','构成凸包点');
xlabel('\itx坐标'),ylabel('\ity坐标'),title('图2： 封闭凸包点示意图');
grid on
box on

% 绘制Figure 3： 封闭凸包轮廓示意图
% 完成凸包求解，记录各凸包点的x,y坐标（按照顺时针或逆时针方向）
ptx=zeros(1,cnt);
pty=zeros(1,cnt);
for i=1:cnt
    ptx(i)=x(pp(i));
    pty(i)=y(pp(i));
end
% 新建图层
figure;
hold on
plot(x,y,'b*','MarkerSize',6);
plot(ptx,pty,'r-','LineWidth',2);
legend('原始随机生成点','凸包轮廓线');
xlabel('\itx坐标'),ylabel('\ity坐标'),title('图3： 封闭凸包轮廓示意图');
grid on
box on

%% 计算凸包面积
S=0; % 记录凸包面积
x11=x(pp(1)); % 以第一个点为三角形公共顶点
y11=y(pp(1));
for k=2:(cnt-2)
    x22=x(pp(k));
    y22=y(pp(k));
    x33=x(pp(k+1));
    y33=y(pp(k+1));

```



```

% 两点之间距离公式
a=sqrt((x11-x22)*(x11-x22)+(y11-y22)*(y11-y22));
b=sqrt((x11-x33)*(x11-x33)+(y11-y33)*(y11-y33));
c=sqrt((x22-x33)*(x22-x33)+(y22-y33)*(y22-y33));
p=(a+b+c)/2; % 计算三角形半周长
dS=sqrt(p*(p-a)*(p-b)*(p-c)); % 利用海伦-秦九韶公式计算三角形面积
S=dS+S;
end

% 结果输出：100个随机点的封闭凸包的面积
fprintf('100个随机生成点的封闭凸包的面积为 %.2f \n',S); % 结果输出

```

程序编号	P2	实现功能	凸包搜索与面积计算 GUI 界面设计
程序名称	Program_Polygon_Painting.m		
<pre>%% 利用MATLAB实现多边形凸包构造结果GUI设计 %% GUI界面初始化 function varargout = Program_Polygon_Painting(varargin) % PROGRAM_POLYGON_PAINTING MATLAB code for Program_Polygon_Painting.fig % Last Modified by GUIDE v2.5 10-Jun-2020 19:45:46 % Begin initialization code - DO NOT EDIT  gui_Singleton = 1; gui_State = struct('gui_Name',      mfilename, ...                   'gui_Singleton',  gui_Singleton, ...                   'gui_OpeningFcn', @Program_Polygon_Painting_OpeningFcn, ...                   'gui_OutputFcn',  @Program_Polygon_Painting_OutputFcn, ...                   'gui_LayoutFcn',  [], ...                   'gui_Callback',    []);  if nargin &amp;&amp; ischar(varargin{1})     gui_State.gui_Callback = str2func(varargin{1}); end  if nargout     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:}); else     gui_mainfcn(gui_State, varargin{:}); end % End initialization code - DO NOT EDIT  % --- Executes just before Program_Polygon_Painting is made visible. function Program_Polygon_Painting_OpeningFcn(hObject, eventdata, handles, varargin)</pre>			

```

% This function has no output args, see OutputFcn.

% Choose default command line output for Program_Polygon_Painting
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Program_Polygon_Painting wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Program_Polygon_Painting_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

%% 核心模块实现
%% 模块0：界面清除与还原
% --- Executes on button press in Clearance_Screen.
function Clearance_Screen_Callback(hObject, eventdata, handles)
axes(handles.axes1); % 将axes1作为当前坐标轴，函数图形在当前坐标轴中绘制
cla reset;
zero=num2str(0); %转化为字符串
set(handles.Input_Number,'String',zero); % 将Input_Number空间的string置零
set(handles.Output_Convex_Area,'String',zero); % 将Output_Convex_Area空间的string置零
guidata(hObject,handles); % 更新结构体

%% 模块1：输入随机点个数（文本框）
function Input_Number_Callback(hObject, eventdata, handles)
input=str2num(get(hObject,'String')); % 检查输入是否为空；如果为空,则默认显示为0
if (isempty(input))
    set(hObject,'String','0')
end
guidata(hObject,handles);

% --- Executes during object creation, after setting all properties.
function Input_Number_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%% 模块2：生成平面随机点
% --- Executes on button press in Point_Generator.

```

```

function Point_Generator_Callback(hObject, eventdata, handles)

global n x y; % 定义全局变量，方便参数传递

a=get(handles.Input_Number,'String'); % 提取输入随机点个数
n=str2num(a); % 将字符串转化为数值
x=100*rand(1,n); % 在[0,100]之内随机生成x坐标点（可修改）
y=100*rand(1,n); % 在[0,100]之内随机生成y坐标点（可修改）
% 绘制Figure 1：原始随机生成点示意图
% 绘制生成的100个随机点
axes(handles.axes1); % 将axes1作为当前坐标轴，函数图形在当前坐标轴中绘制
cla reset;
plot(x,y,'b*','MarkerSize',6);
xlabel('\itx坐标'),ylabel('\ity坐标'),title('图1：原始随机生成点示意图');
grid on
box on

%% 模块3：绘制凸包点集
% --- Executes on button press in Display_Convex_Enveloping_Points.
function Display_Convex_Enveloping_Points_Callback(hObject, eventdata, handles)
% 寻找凸包核心算法设计
% 设置凸包点集标志，1为满足条件点，0为非满足条件点

global n x y cnt pp; % 定义全局变量

p_flag=zeros(1,n);
xmax=10000;
% 寻找最小x坐标点，此点必定是凸包点，以此点作为凸包起点
for i=1:n
    if (x(i)<xmax) && (p_flag(i)==0)
        pj=i; % 记录最小x坐标点的点序号
        xmax=x(i); % 更新最小的xmax值
    end
end
p_flag(pj)=1;

flag_1=0; % 设置搜索停止判断标志：从起点开始搜索下一个点，判断是否已经找到凸包，
程序退出标志为下一个点是起点
st=pj; % 起点编号
cnt=1; % 记录凸包点数，初始状态已有一个点加入
pp=zeros(1,n); % pp数组记录按顺序排列（顺时针或逆时针）的凸包点集，方便后续画图
pp(1)=pj; % 设置最小x坐标点为第一个凸包点（起点）

while (flag_1~=1) % 若不满足退出条件，则进行while循环

```

```

% 提取起点的x,y坐标为(x1,y1)
x1=x(st);
y1=y(st);

% 设置i变量遍历所有点坐标,寻找下一个在凸包内的点
for i=1:n

    if (i~=st) % 如果下一个点不在凸包内
        % 两点构造直线,提取当前点的x,y坐标为(x2,y2)
        x2=x(i);
        y2=y(i);
        % 直线方程已化简为Ax+By+C=0的形式
        A=(y2-y1);
        B=(x1-x2);
        C=((x2-x1)*y1-(y2-y1)*x1);

        tot=1; % 关键点1: tot用以判断是否该直线为凸包边线
        % 由于将在落在直线两边的点坐标代入直线方程后,Ax+By+C的值异
号
        % 利用该特点,设置累乘器tot,逐次代入这些点坐标,只要检索到
tot<=0,则可排除该点为凸包点;
        % 若其他98个点依次代入后tot始终为正,则该点为凸包点,标志为
flag_2=1

        flag_2=1; % 判断i编号点是否为凸包点的标志
        for j=1:n % 遍历所有点集检验是否满足凸包条件
            if (j~=st) && (j~=i) % 排除构造直线的两个点,因为其使得Ax+By+C=0将
使tot=0,无法判断结果
                xj=x(j);
                yj=y(j);
                t=(A*xj)+(B*yj)+C;
                % t=t*100;
                % 考虑由于tot每次乘的量若很小的情况,或者结果非常接近0
的情况时,可乘以一个放大系数(可选择)
                tot=tot*t;
                if (tot<=0)
                    flag_2=0;
                    break; % 剪枝操作,提高程序效率
                end
            end
        end
        end

        % 若成功,即找到凸包点

```

```

        if (flag_2==1)
            en=i; % 记录当前点编号
            p_flag(en)=1; % 标记该点已经进入凸包集合
            cnt=cnt+1; % 凸包点个数加1
            pp(cnt)=en;
            st=en; % 关键点2：将找到的新的下一个凸包点作为新的起点，循环再寻找下一个凸包点
        if (en==pj) % 关键点3：凸包封闭标志，也即程序退出标志为：寻找到的新的凸包点回到了起点
            flag_1=1; % 完成凸包封闭操作，flag_1置1，退出整个迭代程序
            break;
        end
    end

end

end % 结束i（编号）循环
end % 结束while循环

% 绘制Figure 2：封闭凸包点示意图
% 完成凸包求解，记录各凸包点的x,y坐标（无序）——显示凸包点
ppx=zeros(1,cnt-1); % 各凸包点x坐标（无序）
ppy=zeros(1,cnt-1); % 各凸包点y坐标（无序）
cntt=0;
for k=1:n
    if (p_flag(k)==1) % 若该点为凸包点
        cntt=cntt+1;
        ppx(cntt)=x(k);
        ppy(cntt)=y(k);
    end
end

% 设置绘图坐标轴
axes(handles.axes1); % 将axes1作为当前坐标轴，函数图形在当前坐标轴中绘制
cla reset;
hold on
% 绘制生成的100个随机点
plot(x,y,'b*','MarkerSize',6);
% 绘制凸包点集
plot(ppx,ppy,'r.','MarkerSize',16);
hold off
% 添加图例、坐标信息
legend('原始随机生成点','构成凸包点');
xlabel('\itx坐标'),ylabel('\ity坐标'),title('图2：封闭凸包点示意图');

```

```

grid on
box on

%% 模块4：绘制凸包边线
% --- Executes on button press in Display_Convex_Enveloping_Boundary.
function Display_Convex_Enveloping_Boundary_Callback(hObject, eventdata, handles)

global n x y cnt pp; % 定义全局变量

p_flag=zeros(1,n);
xmax=10000;
% 寻找最小x坐标点，此点必定是凸包点，以此点作为凸包起点
for i=1:n
    if (x(i)<xmax) && (p_flag(i)==0)
        pj=i; % 记录最小x坐标点的点序号
        xmax=x(i); % 更新最小的xmax值
    end
end
p_flag(pj)=1;

flag_1=0; % 设置搜索停止判断标志：从起点开始搜索下一个点，判断是否已经找到凸包，
程序退出标志为下一个点是起点
st=pj; % 起点编号
cnt=1; % 记录凸包点数，初始状态已有一个点加入
pp=zeros(1,n); % pp数组记录按顺序排列（顺时针或逆时针）的凸包点集，方便后续画图
pp(1)=pj; % 设置最小x坐标点为第一个凸包点（起点）

while (flag_1~=1) % 若不满足退出条件，则进行while循环

    % 提取起点的x,y坐标为(x1,y1)
    x1=x(st);
    y1=y(st);

    % 设置i变量遍历所有点坐标,寻找下一个在凸包内的点
    for i=1:n

        if (i~=st) % 如果下一个点不在凸包内
            % 两点构造直线，提取当前点的x,y坐标为(x2,y2)
            x2=x(i);
            y2=y(i);
            % 直线方程已化简为Ax+By+C=0的形式
            A=(y2-y1);
            B=(x1-x2);
            C=((x2-x1)*y1-(y2-y1)*x1);

```

```

        tot=1; % 关键点1: tot用以判断是否该直线为凸包边线
        % 由于将在落在直线两边的点坐标代入直线方程后,  $Ax+By+C$  的值异
号
        % 利用该特点, 设置累乘器tot, 逐次代入这些点坐标, 只要检索到
tot<=0, 则可排除该点为凸包点;
        % 若其他98个点依次代入后tot始终为正, 则该点为凸包点, 标志为
flag_2=1

        flag_2=1; % 判断i编号点是否为凸包点的标志
        for j=1:n % 遍历所有点集检验是否满足凸包条件
            if (j~=st) && (j~=i) % 排除构造直线的两个点, 因为其使得  $Ax+By+C=0$  将
使tot=0, 无法判断结果
                xj=x(j);
                yj=y(j);
                t=(A*xj)+(B*yj)+C;
                % t=t*100;
                % 考虑由于tot每次乘的量若很小的情况, 或者结果非常接近0
的情况时, 可乘以一个放大系数 (可选择)
                tot=tot*t;
                if (tot<=0)
                    flag_2=0;
                    break; % 剪枝操作, 提高程序效率
                end
            end
        end

        % 若成功, 即找到凸包点
        if (flag_2==1)
            en=i; % 记录当前点编号
            p_flag(en)=1; % 标记该点已经进入凸包集合
            cnt=cnt+1; % 凸包点个数加1
            pp(cnt)=en;
            st=en; % 关键点2: 将找到的新的下一个凸包点作为新的起点, 循环再寻
找下一个凸包点
            if (en==pj) % 关键点3: 凸包封闭标志, 也即程序退出标志为: 寻找到的
新的凸包点回到了起点
                flag_1=1; % 完成凸包封闭操作, flag_1置1, 退出整个迭代程序
                break;
            end
        end
    end

end

```

```

    end % 结束i（编号）循环
end % 结束while循环

% 绘制Figure 3： 封闭凸包轮廓示意图
% 完成凸包求解，记录各凸包点的x,y坐标（按照顺时针或逆时针方向）
ptx=zeros(1,cnt);
pty=zeros(1,cnt);
for i=1:cnt
    ptx(i)=x(pp(i));
    pty(i)=y(pp(i));
end

% 设置绘图坐标轴
axes(handles.axes1); % 将axes1作为当前坐标轴，函数图形在当前坐标轴中绘制
cla reset;
hold on
plot(x,y,'b*','MarkerSize',6);
plot(ptx,pty,'r-','LineWidth',2);
hold off
legend('原始随机生成点','凸包轮廓线');
xlabel('\itx坐标'),ylabel('\ity坐标'),title('图3： 封闭凸包轮廓示意图');
grid on
box on

%% 模块5： 计算凸包面积
% --- Executes on button press in Calculation_Convex_Area.
function Calculation_Convex_Area_Callback(hObject, eventdata, handles)

global x y cnt pp; % 定义全局变量

S=0; % 记录凸包面积
x11=x(pp(1)); % 以第一个点为三角形公共顶点
y11=y(pp(1));
for k=2:(cnt-2)
    x22=x(pp(k));
    y22=y(pp(k));
    x33=x(pp(k+1));
    y33=y(pp(k+1));
    % 两点之间距离公式
    a=sqrt((x11-x22)*(x11-x22)+(y11-y22)*(y11-y22));
    b=sqrt((x11-x33)*(x11-x33)+(y11-y33)*(y11-y33));
    c=sqrt((x22-x33)*(x22-x33)+(y22-y33)*(y22-y33));
    p=(a+b+c)/2; % 计算三角形半周长
    dS=sqrt(p*(p-a)*(p-b)*(p-c)); % 利用海伦-秦九韶公式计算三角形面积

```



```
S=dS+S;
end

c=num2str(S); %转化为字符串
set(handles.Output_Convex_Area,'String',c); % 保存结构体，将结果赋值给
Output_Convex_Area空间的string属性，显示结果
guidata(hObject,handles); % 更新结构体

%% 模块6：输出凸包面积（文本框）
function Output_Convex_Area_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function Output_Convex_Area_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```