姓名：陈墨霏

学号：1120233329

班级：07112304

专业：计算机科学与技术

指导教师：孙新

关于编程实现解线性方程组的直接法的

实验报告

（第三章上机实验报告）

目录

[1. 实验目标 1](#_Toc183030528)

[2. 实验内容 1](#_Toc183030529)

[2.1 实现消元法核心算法 1](#_Toc183030530)

[2.2 开发一个图形化界面 1](#_Toc183030531)

[3. 实现过程 2](#_Toc183030532)

[3.1 消元法的一般步骤 2](#_Toc183030533)

[3.1.1 消元过程 2](#_Toc183030534)

[3.1.2 回代过程 2](#_Toc183030535)

[3.1.3 具体的各类消元法 2](#_Toc183030536)

[3.2 核心算法实现 3](#_Toc183030537)

[3.2.1 高斯LU分解（Gaussian LU）： 3](#_Toc183030538)

[3.2.2 克劳特LU分解（Crout LU）： 3](#_Toc183030539)

[3.2.3 平方根LU分解（Sqrt LU）： 3](#_Toc183030540)

[3.2.4 求解线性方程组（solve\_lu）： 3](#_Toc183030541)

[3.2.5 整体流程（solve\_linear\_system）： 4](#_Toc183030542)

[3.3 图形化界面搭建 4](#_Toc183030543)

[3.3.1 主窗口搭建过程 4](#_Toc183030544)

[3.3.2 三对角矩阵输入窗口搭建过程 4](#_Toc183030545)

[4. 输入、输出测试 5](#_Toc183030546)

[4.1 高斯消元法 5](#_Toc183030547)

[4.2 克劳特消元法 6](#_Toc183030548)

[4.2.1 一般情况 6](#_Toc183030549)

[4.2.2 追赶法 7](#_Toc183030550)

[4.3 平方根法 9](#_Toc183030551)

[5. 实验分析、结论、改进策略和心得体会 10](#_Toc183030552)

[5.1 实验分析 11](#_Toc183030553)

[5.1.1 算法实现与效率 11](#_Toc183030554)

[5.1.2 图形化界面的用户体验 11](#_Toc183030555)

[5.1.3 算法的准确性 11](#_Toc183030556)

[5.2 实验结论 11](#_Toc183030557)

[5.2.1 算法有效性 11](#_Toc183030558)

[5.2.2 界面友好性 11](#_Toc183030559)

[5.2.3 实际应用价值 12](#_Toc183030560)

[5.3 改进策略及方向 12](#_Toc183030561)

[5.3.1 粘贴输入数据、从文件导入数据 12](#_Toc183030562)

[5.3.2 导出结果到文件 12](#_Toc183030563)

[5.3.3 根据矩阵特点自动选择求解方法 12](#_Toc183030564)

[5.3.4 求解过程详细展示 13](#_Toc183030565)

[5.3.5 更加具体明确的错误提示 13](#_Toc183030566)

[5.3.6 更加符合工程开发要求的的代码撰写和文件组织 13](#_Toc183030567)

[5.4 心得体会 13](#_Toc183030568)

# 实验目标

使用Python语言实现**n阶稠密线性方程组（阶数n<300）**的消元法，支持高斯消元法（采用列主元素法）、克劳特消元法（含追赶法）、平方根法三种主要解法。

进一步的，使用pyqt6库开发一个图形化界面，支持矩阵阶数选择，支持消元方法选择，支持三对角矩阵快速输入。

# 实验内容

## 实现消元法核心算法

使用Python语言实现了求解n阶稠密线性方程组的程序，支持多种常用的数值解法，包括高斯消元法、克劳特消元法和平方根法。其中，高斯消元法采用了列主元素选取策略以提高数值稳定性；克劳特消元法适用于一般线性方程组的直接分解求解；平方根法则针对正定对称矩阵进行优化分解。这些算法为解决工程计算中的复杂线性方程组提供了高效的数值工具，并具有良好的扩展性和准确性。

## 开发一个图形化界面

使用PyQt6库开发一个图形化界面，旨在提供用户友好的操作体验以求解线性方程组。界面包括多个输入框和控件，首先，用户可以通过输入框选择矩阵的阶数，并选择对应的消元方法（如高斯消元法、克劳特法或平方根法）。此外，提供了一个下拉框来选择是否生成三对角矩阵，并通过额外的输入界面支持快速输入三对角矩阵的值。用户可以选择不同的矩阵类型和方法后，点击按钮生成相应的矩阵和求解过程，系统会自动显示解向量，并通过表格显示系数矩阵、常数向量和解向量。整个操作流程简单直观，并支持结果的实时更新和错误提示，确保用户能够轻松进行矩阵计算与求解。

# 实现过程

## 消元法的一般步骤

### 消元过程

消元法的消元过程通过逐步将方程组转化为上三角矩阵的形式，从而简化问题求解。通过行变换消除未知数的系数，逐步消去下三角部分，使得方程组的结构变得更加简洁。消元的核心思想是通过矩阵变换将原方程组转化为等价的上三角形式，使得后续的回代过程能够高效地进行。此过程体现了LU分解的思想，其中L矩阵表示通过消元过程得到的下三角矩阵，而U矩阵则是最终得到的上三角矩阵。

### 回代过程

回代过程则是在已经得到上三角矩阵的基础上，从最后一个方程开始，逐步求解出每个未知数的值。在回代过程中，每次通过已知的部分解来求解当前未知数的值，最终得到整个方程组的解。这一过程与LU分解中的U矩阵相对应，展示了如何通过分解得到的上三角矩阵逐步解出原方程组的解。LU分解的核心思想正是通过这样的矩阵分解，使得解方程的过程更加高效和简便。

### 具体的各类消元法

高斯消元法、克劳特消元法（含追赶法）和平方根法，均可以视为消元法的一般过程下，对L矩阵的对角线元素的不同取法。高斯消元法将lii取为1，克劳特消元法将lii取为ann(i-1)，平方根法将lii取为uii。因此，可以将代码进行一定的复用。

## 核心算法实现

代码在./ funclib/funclib\_elimination.py中。

### 高斯LU分解（Gaussian LU）：

列主元选择：在进行消元操作前，选择当前列的最大值所在行作为主元，通过交换行来增强数值稳定性。

生成置换矩阵P：如果发生行交换，更新置换矩阵P，用于调整矩阵的顺序。

消元过程：通过逐行消元的方式，生成上三角矩阵U和下三角矩阵L。L[i][j]表示下三角矩阵的元素，U[i][j]表示上三角矩阵的元素。

求解过程：通过solve\_lu函数，首先解方程LZ = B，然后解方程UX = Z，得到最终解X。

### 克劳特LU分解（Crout LU）：

分解过程：将矩阵A分解成一个下三角矩阵L和一个上三角矩阵U。与高斯消元不同，克劳特法直接将U初始化为单位矩阵。

逐列消元：通过逐列消元计算L和U的元素，其中L是下三角矩阵，U是上三角矩阵。

求解过程：同样使用solve\_lu函数，通过前向替代解方程LZ = B，后向替代解方程UX = Z。

### 平方根LU分解（Sqrt LU）：

Cholesky分解：适用于对称正定矩阵，将矩阵A分解为下三角矩阵L和其转置U。具体做法是通过计算L的元素，从而得到U（即L的转置）。

求解过程：与其他两种方法相同，使用solve\_lu函数，通过LZ = B和UX = Z的步骤求解线性方程。

### 求解线性方程组（solve\_lu）：

solve\_lu函数通过两步过程求解线性方程：

前向替代：解方程LZ = B，通过逐步消除L的元素得到中间变量Z。

后向替代：解方程UX = Z，通过逐步消除U的元素得到最终解X。

### 整体流程（solve\_linear\_system）：

solve\_linear\_system函数根据输入的分解方法（mode参数）选择对应的LU分解算法（Gaussian、Crout、Sqrt），并返回LU分解的结果（L、U矩阵）及线性方程组的解X。

## 图形化界面搭建

### 主窗口搭建过程

代码在./GUI/GUI\_main.py和./GUI/back\_end\_GUI\_main.py中。

1. **初始化和布局设置：**在\_\_init\_\_方法中，首先调用init\_ui设置窗口标题和尺寸，并通过create\_central\_widget创建中央小部件，set\_geometry方法将窗口定位在屏幕中央。随后，使用垂直布局QVBoxLayout和水平布局QHBoxLayout来组织界面元素。
2. **输入框和下拉菜单：**create\_input\_layout方法创建了一个包含文本输入框、下拉菜单（用于选择矩阵类型）和另一下拉菜单（用于选择求解方法）的输入区域。
3. **表格：**通过create\_scroll\_layout方法设置了三个滚动区域，分别用于显示系数矩阵、常数向量和解向量。每个滚动区域内都有一个QTableWidget来显示表格数据。init\_table方法根据用户输入的矩阵大小创建表格，并设置表格的列和行高。create\_table方法根据输入的大小动态更新表格。
4. **交互功能：**包括根据下拉菜单选择不同矩阵类型和求解方法，调用相应的计算和结果展示功能。

### 三对角矩阵输入窗口搭建过程

代码在./GUI/GUI\_tr.py和./GUI/back\_end\_GUI\_tr.py中。

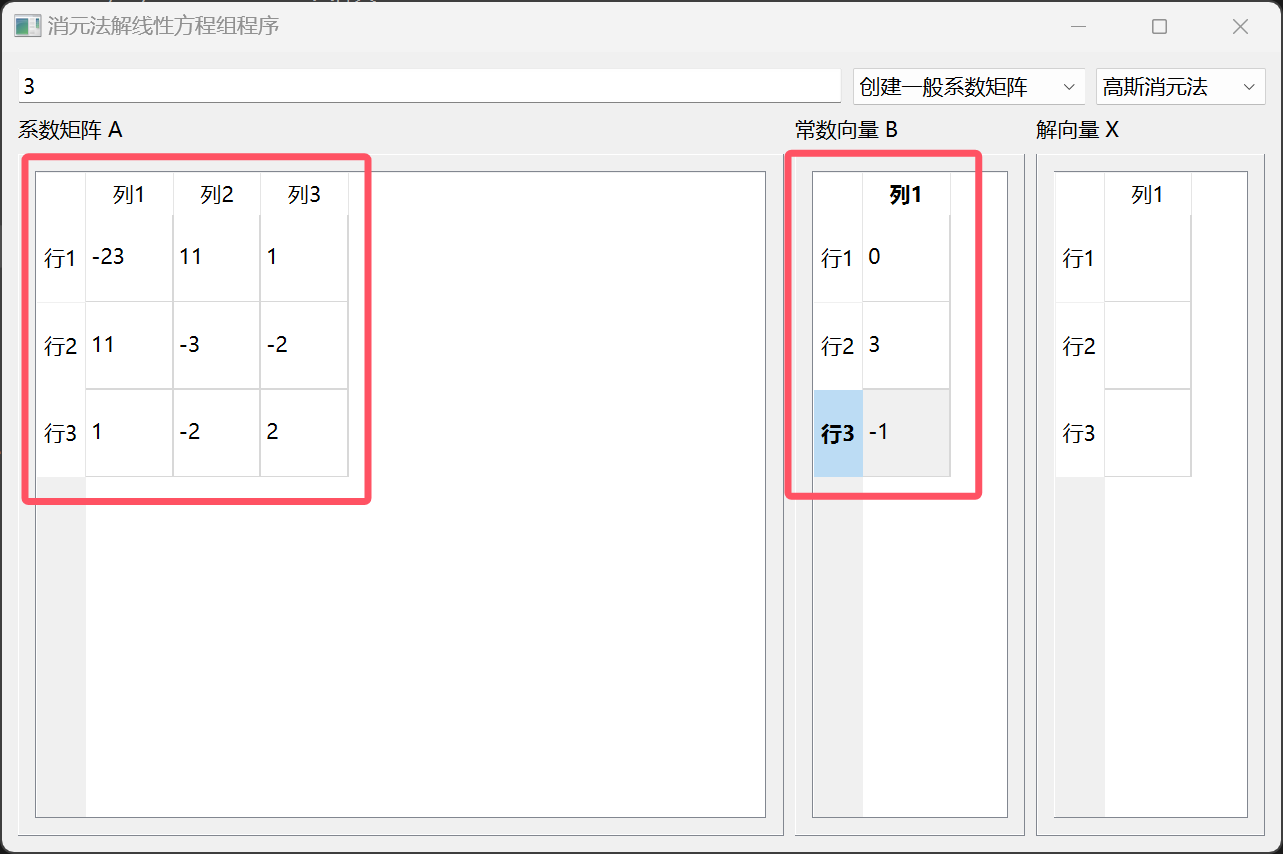
1. **初始化和布局设置：**在\_\_init\_\_方法中，窗口大小和位置根据屏幕几何信息设置，并禁用最大化按钮。接着，创建了一个垂直布局QVBoxLayout，并将其应用于窗口的中心部件。
2. **按钮和表格：**窗口包含一个按钮“创建三对角系数矩阵”，用户点击后触发create\_tridiagonal\_matrix方法。表格通过QTableWidget动态创建，显示三对角矩阵的数值。表格的行列大小为size×3，列分别代表三对角矩阵的上三角、对角和下三角元素。

# 输入、输出测试

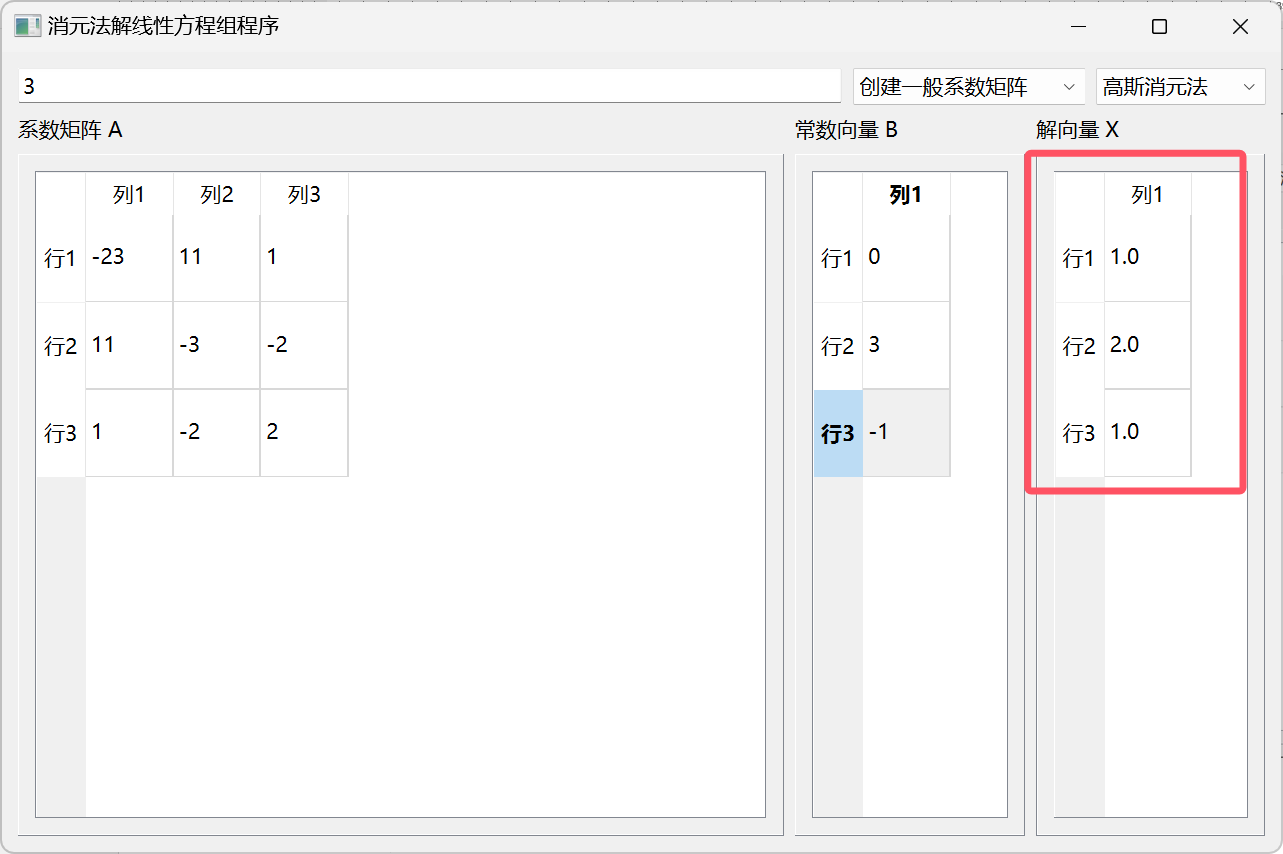
## 高斯消元法

示例采用课本P78 例3.1。

输入数据：



选择高斯消元法，求得输出结果：



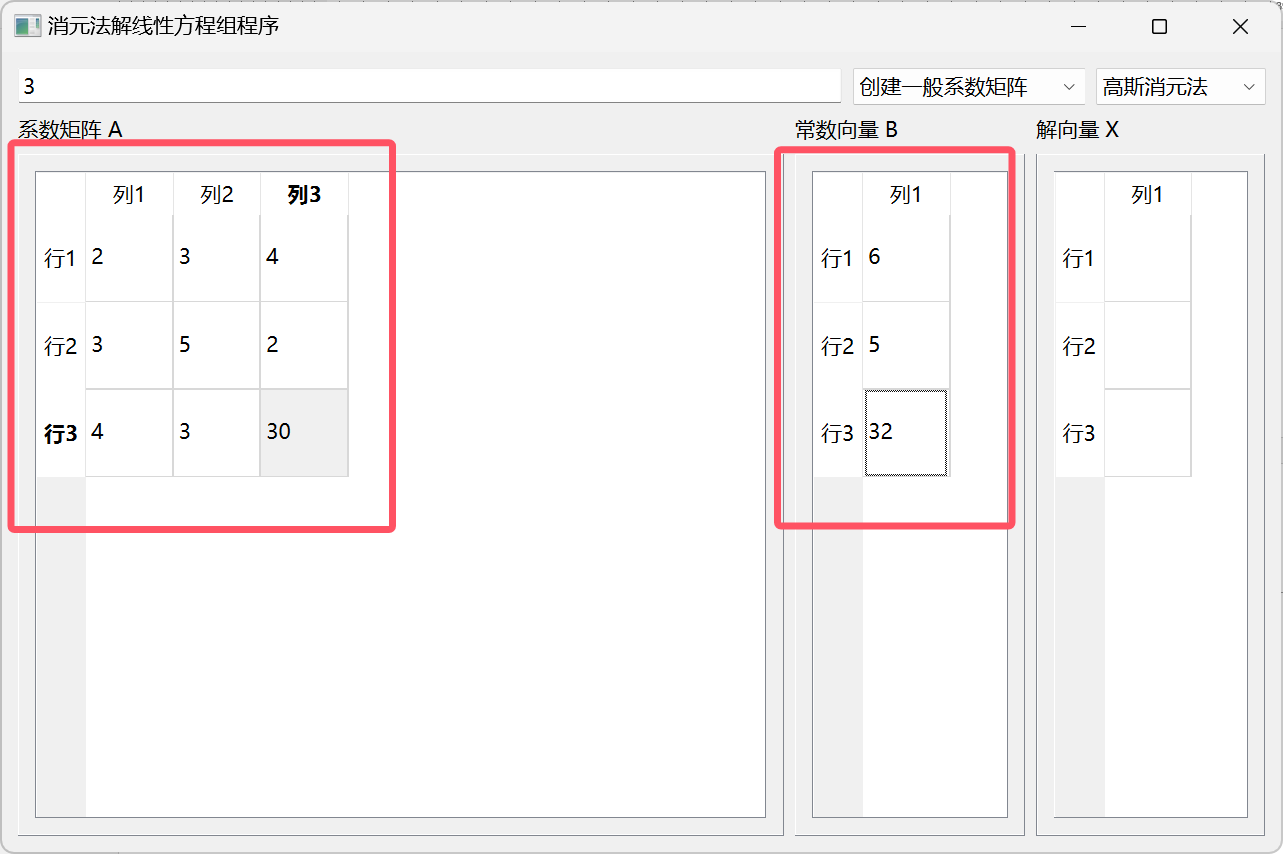
与课本结果基本一致。

## 克劳特消元法

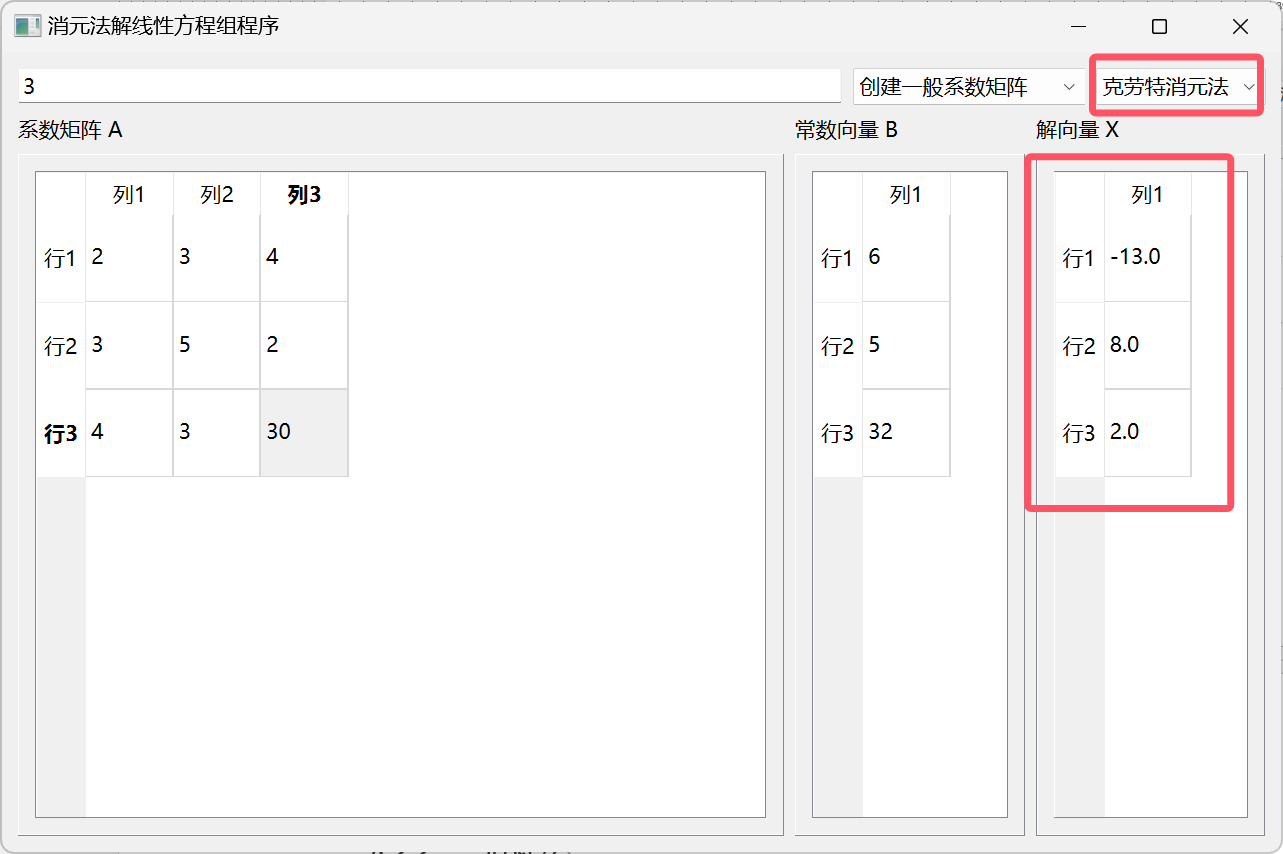
### 一般情况

示例采用课本P79 例3.2。

输入数据：



选择克劳特消元法，求得输出结果：



与课本结果基本一致。

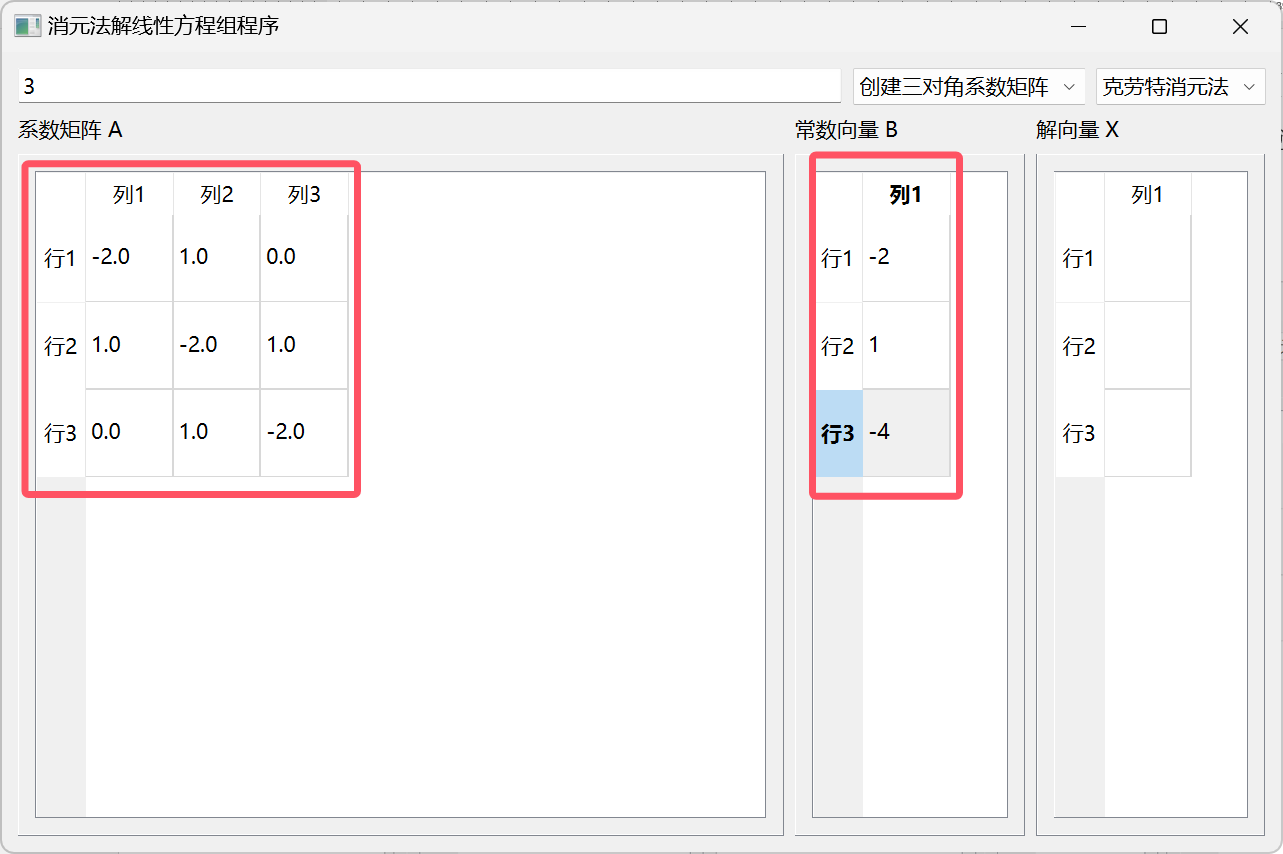
### 追赶法

示例采用课本P82 例3.4。

选择“创建三对角系数矩阵”，按**以下规则**输入数据：

三对角系数矩阵第一行当中的非零数据在第一行右对齐，最后一行当中的非零数据在最后一行左对齐。其余行的非零数据，按顺序输入即可。之后点击“创建三对角矩阵”按钮，则返回主界面，表格中自动生成三对角系数矩阵。此后，按顺序输入常数向量。





选择克劳特消元法，求得输出结果：



与课本结果基本一致。

## 平方根法

示例采用课本P80 例3.3。

输入数据：



选择平方根法，求得输出结果：



与课本结果基本一致。

# 实验分析、结论、改进策略和心得体会

## 实验分析

### 算法实现与效率

本实验成功实现了高斯消元法、克劳特消元法和平方根法三种解线性方程组的直接法。通过实际编程实现，我们发现高斯消元法在处理大规模方程组时，由于采用了列主元素选取策略，其数值稳定性较好。克劳特消元法适用于一般线性方程组，而平方根法则特别适用于正定对称矩阵，显示出对特定类型的矩阵有优化效果。

### 图形化界面的用户体验

使用PyQt6开发的图形化界面提供了直观的用户交互方式，使得非专业用户也能方便地求解线性方程组。界面的设计考虑了用户的输入便利性，包括矩阵阶数选择、消元方法选择以及三对角矩阵的快速输入。

### 算法的准确性

通过与课本中的示例进行对比，我们的算法实现结果与课本结果基本一致，验证了算法的正确性。这表明我们的程序能够有效地解决线性方程组问题，并提供准确的解。

## 实验结论

### 算法有效性

实验结果表明，所实现的三种消元法能够有效地解决线性方程组问题，且结果准确可靠。特别是对于特定类型的矩阵，如正定对称矩阵，平方根法显示出其优越性。

### 界面友好性

图形化界面的设计提高了程序的易用性，使得用户无需深入了解算法细节即可进行线性方程组的求解。界面的直观性和操作的便捷性是本实验的一个重要成果。

### 实际应用价值

本实验的成果不仅在理论上验证了消元法的有效性，而且在实际应用中，通过图形化界面的实现，使得这些算法可以被更广泛地应用于工程和科学计算中，具有较高的实用价值。

## 改进策略及方向

### 粘贴输入数据、从文件导入数据

当前，在输入数据时，需要一个一个手动输入。如果数据量稍微偏大的情况（如n>5），则输入会变得非常困难，这也使得这个程序设计的300的阶数上限变得无所适从。

因此，可以添加从含有制表符的剪贴板数据粘贴功能，就像粘贴到excel表格一样，直接粘贴数据到系数矩阵和常数向量处，以此简化手动输入过程。

另外，还可以添加从csv文件、json文件、excel文件等文件导入数据的方式，通过规定这些文件内的数据组织格式，一键导入数据。

### 导出结果到文件

可以添加一个“导出结果到文件”选项，以导出计算结果到csv文件、json文件、excel表格等文件中，以实现进一步的操作。

### 根据矩阵特点自动选择求解方法

添加一个“自动选择求解方法”选项。如针对对称正定矩阵，自动选择为平方根法。

### 求解过程详细展示

可以添加对LU分解得到的L矩阵、U矩阵的展示。

可以添加对列主元素法具体交换过程的展示。

### 更加具体明确的错误提示

当前，在计算过程中，求解出错的提示仅仅有一个“计算错误”，而无法检测是无解、有无数个解、除0，还是其他错误。通过更加具体明确的错误提示，可以帮助我们发现输入的错误。

### 更加符合工程开发要求的的代码撰写和文件组织

当前的代码存在诸多的不规范的问题。举一例子：消元法代码存在大量的复用现象。这种情况完全可以通过函数接口来解决，但是由于能力和时间限制，尚且没有进行修改和简化。

## 心得体会

由于我将事情拖到了ddl再去做，所以时间就变得非常紧迫，所以5.3实际上也是我想实现但是没时间实现的功能。

**附录：**

**程序运行方法**

在命令行中，切换到程序根目录，输入python main.py，即可运行。

命令行运行方式，需要的库如下（./requirements.txt）：

json

numpy

PyQt6