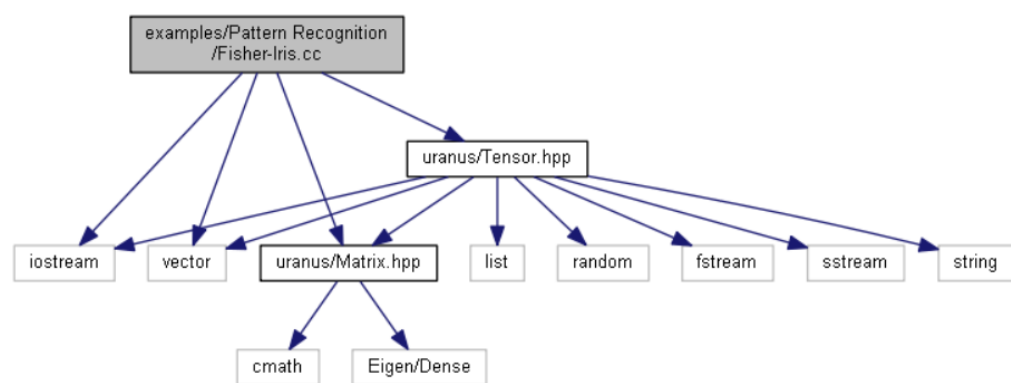
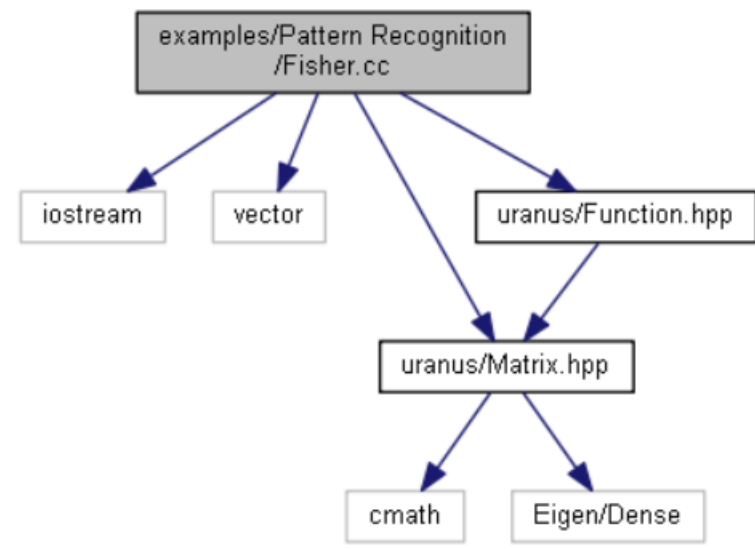
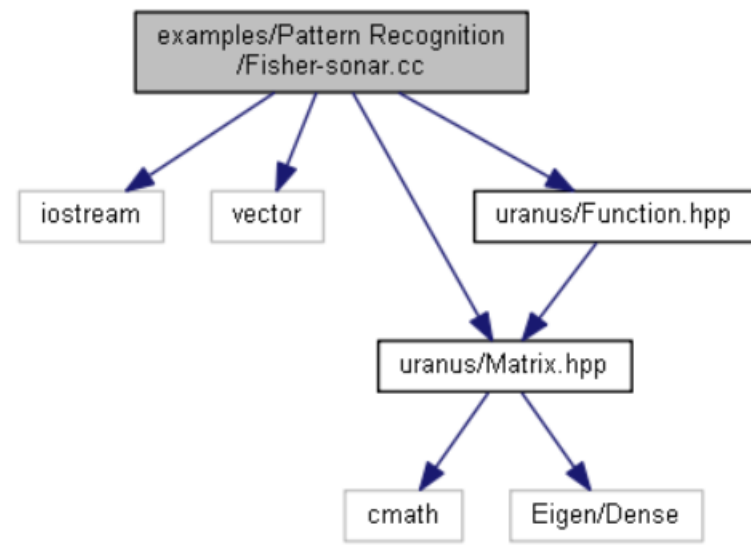


Uranus 详细设计说明书

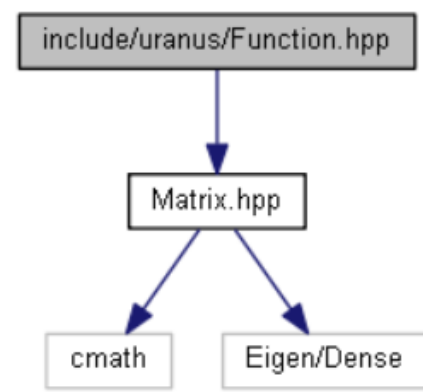
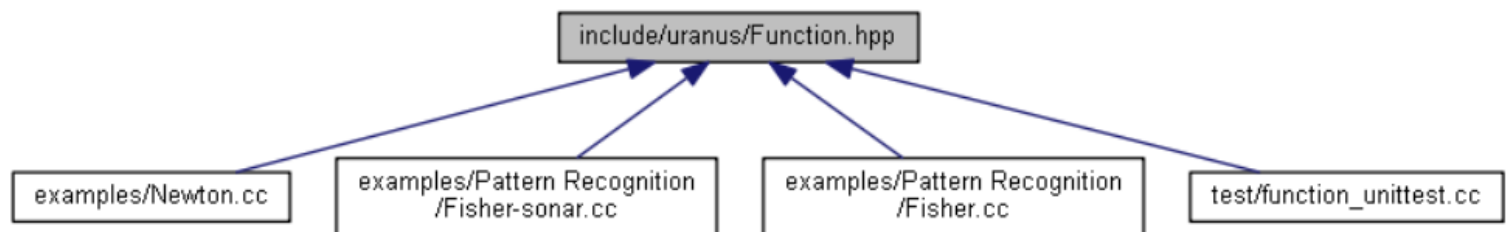
一、 算法设计

Fisher 线性判别算法结构图：

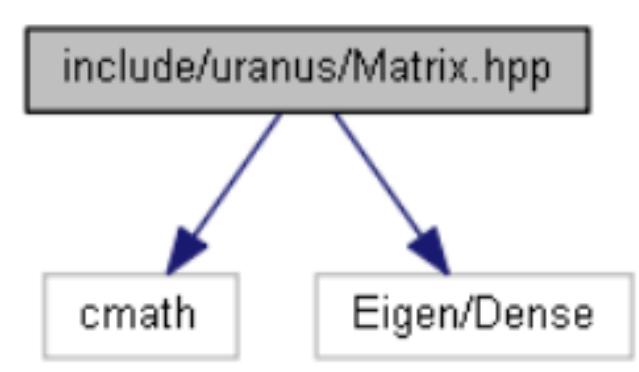
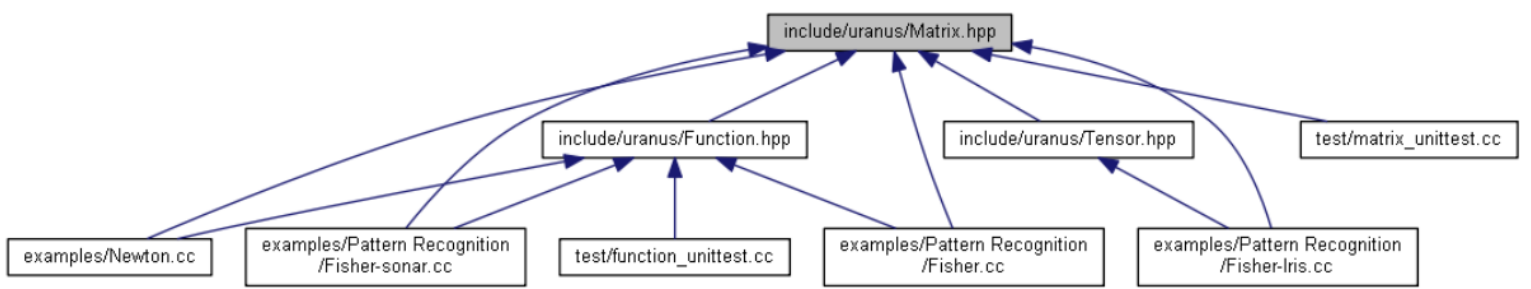




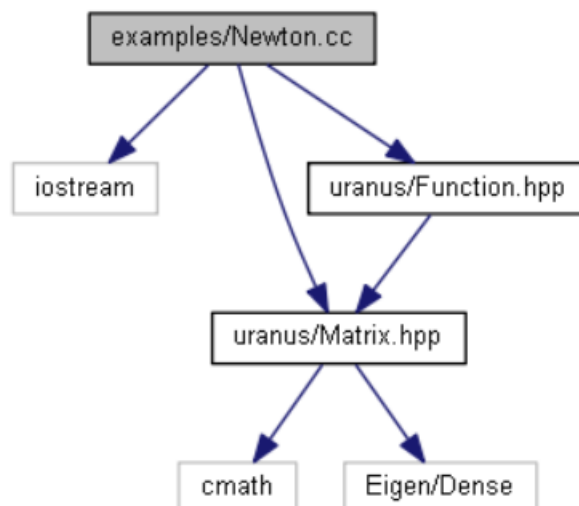
方法调用模块算法结构图：

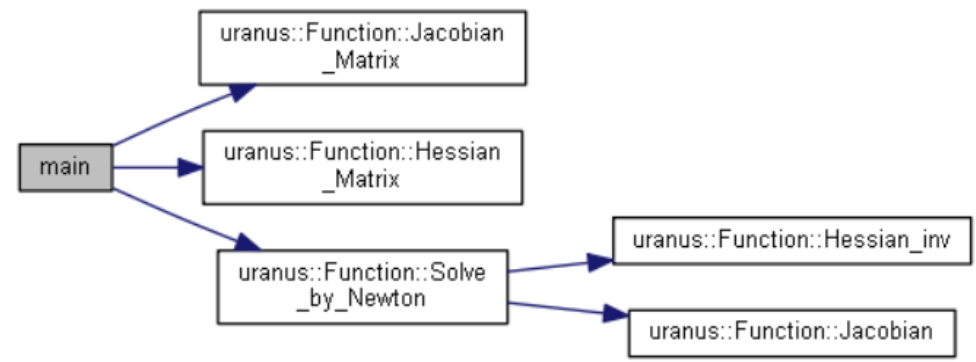


矩阵运算算法结构图：

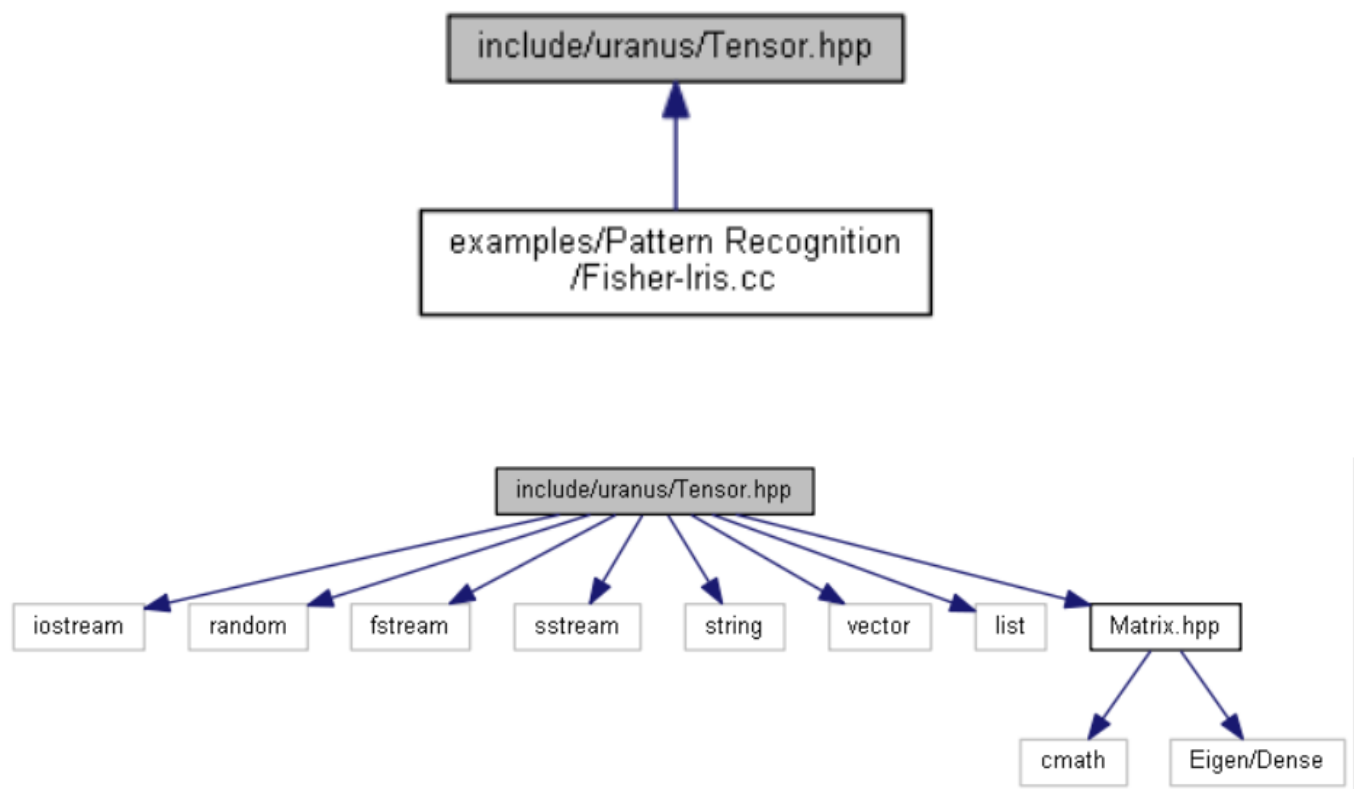


牛顿法算法结构图：



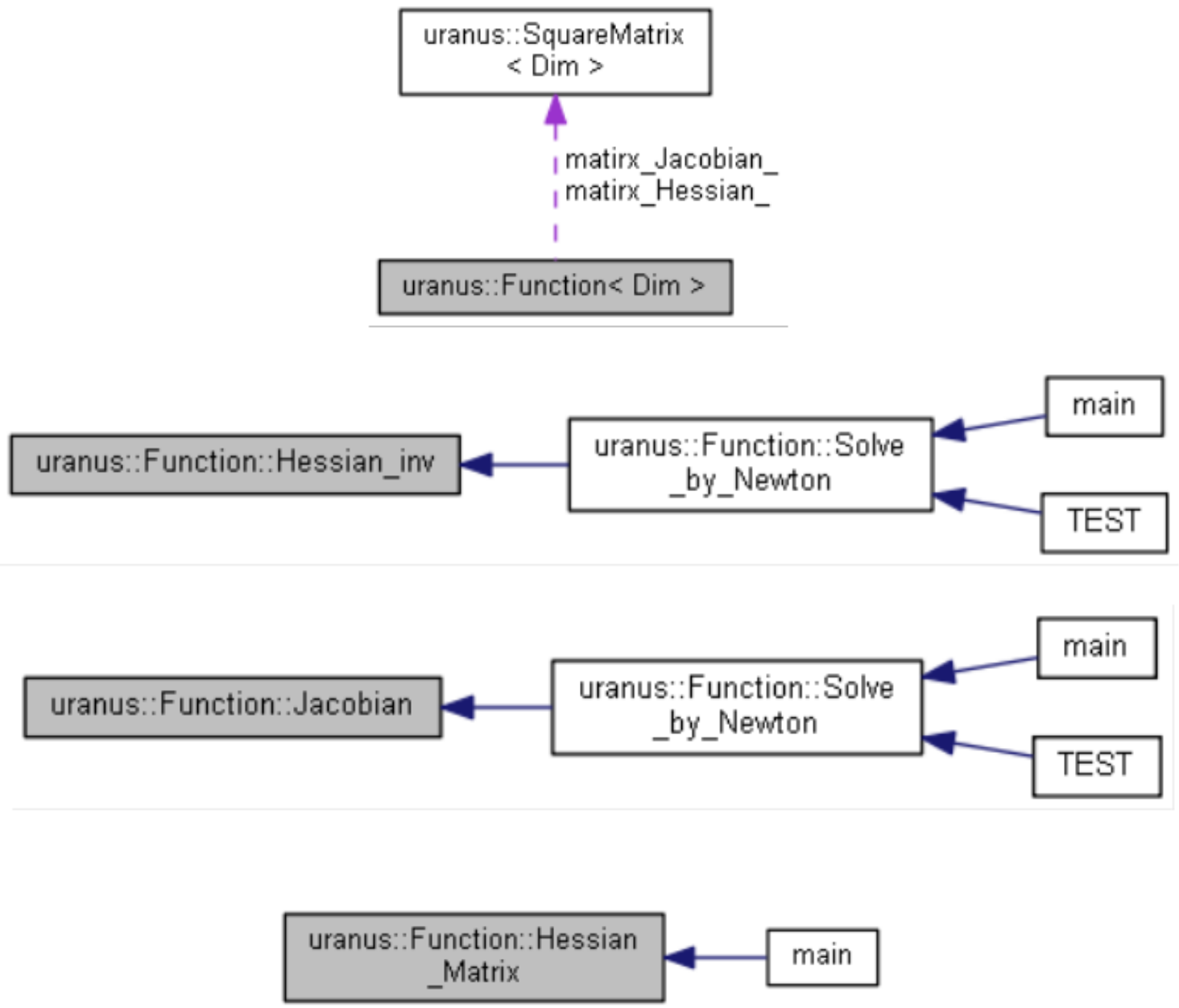


张量求取算法结构图：





雅可比矩阵与海森矩阵算法结构图：

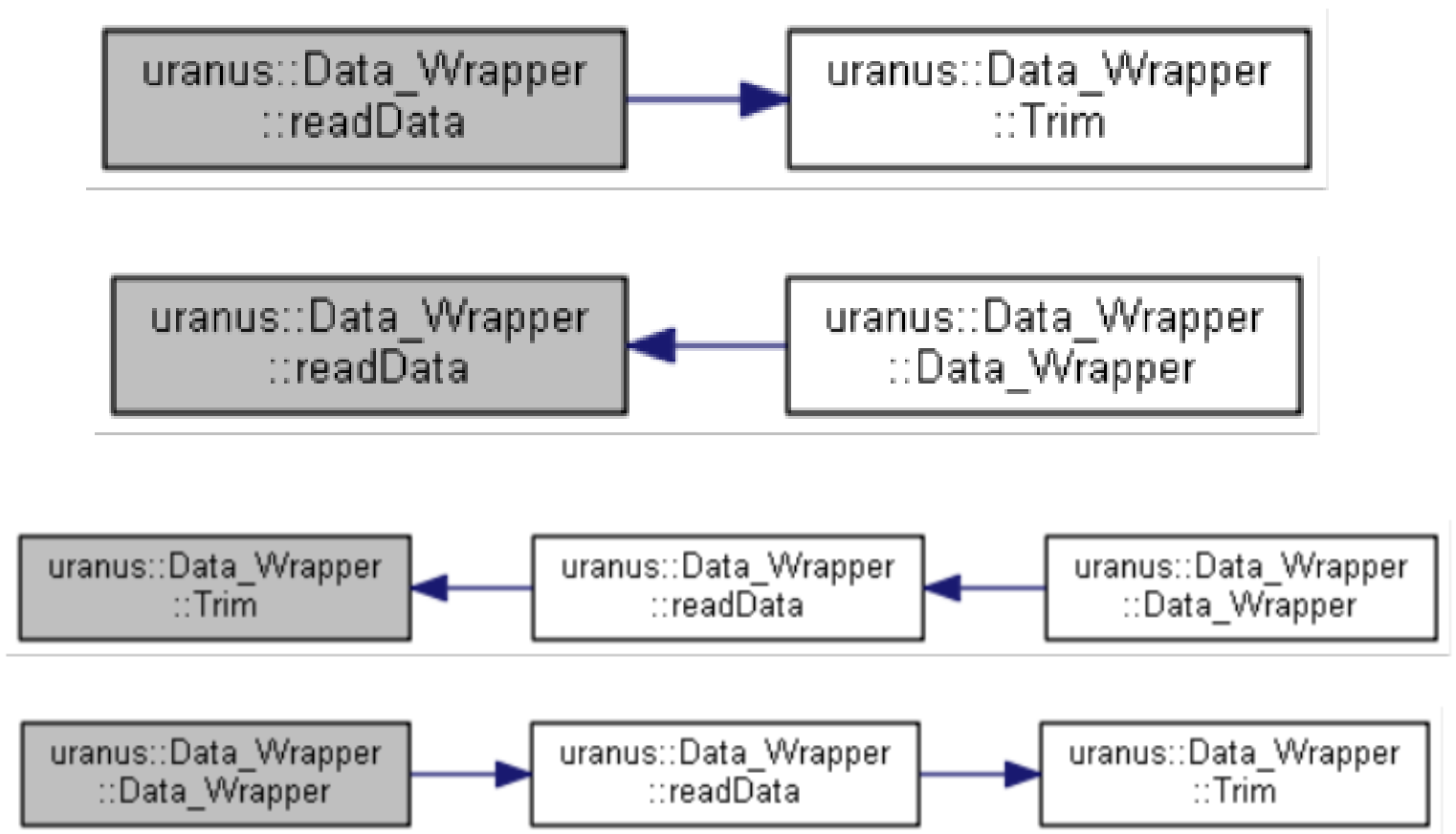


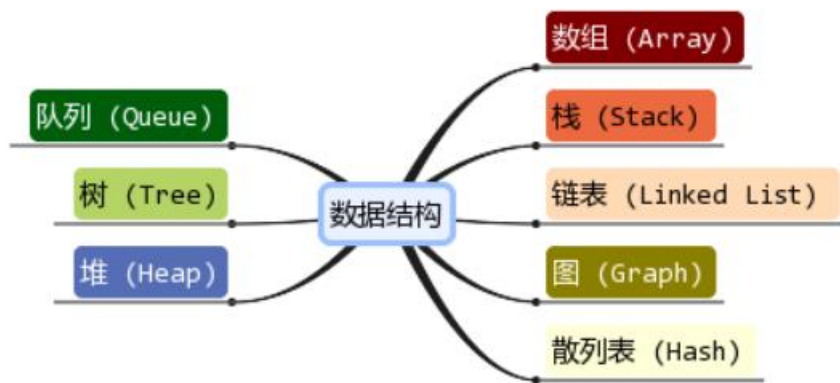
二、 数据结构设计

输入数据：整型、浮点型。

输出数据：根据输入数据类型，返回对应的数据类型。

数据结构图：





Uranus 库为最优化库，其中包含了牛顿法、fisher 线性判别、k 近邻、k-nn、k-means 聚类等优化方法。

不同的优化算法根据算法的特点，拥有不同的数据结构，uranus 库中，大多采用数组、链表、矩阵、队列存储所需数据。

三、交互设计

由于此次设计的程序为一优化库，用户使用只需调用函数库，以及输入变量与参数，并且本库的使用一般用于其他代码中，并不需要 GUI 界面以及频繁的人机交互，只需给出使用文档让用户了解，并了解如何使用即可。因此本库并没有设计人机交互界面，只给出方法的文档以及库的使用说明。

概要设计说明书

1 引言

1.1 编写目的

基于本专业学习的内容，根据学习过的课程，例如模式识别和最优化理论，为了方便同学们能直接使用各类算法，本次项目就将这两门课程所学习到的一些算法集成到一个库函数里，这样以后使用这类算法时就可以直接利用函数调用，而不需要重新编写代码了。

1.2 项目背景

本次项目是作为软件工程课程的大作业，鉴于本学习学习的模式识别和最优化理论课程，于是就将项目定位于编写这两门课程相关的函数库。

1.3 参考资料

《软件工程》，张秋余等人，西安电子科技大学出版社，2014 年 12 月第 1 版

《模式识别》，张学工，清华大学出版社，2010 年 8 月第 3 版

2 任务概述

2.1 目标

Uranus 函数库包括模式识别和最优化理论中的各类常用的基础算法和模型，例如 Fisher 线性判别、k 均值聚类、牛顿法等等，并且保证其算法的性能，方便用户调用和使用函数。

2.2 运行环境

· 主机：常用的笔记本电脑、台式电脑等。

- 操作系统：各类操作系统，包括 Windows、Linux 等。
- 编译器：各种 C、C++语言编译器。

2.3 需求概要

用户对函数库的要求是，调用时方便，源代码可读性强、算法的性能很好。

2.4 限制描述

项目设计应该是模块化的，即将整个函数库分成各个函数模块去编写实现，如果多个函数模块用到同一种函数，则可以把函数模块分成各个子函数模块编写实现。

每个函数模块编写的框架和规范是一样的，包括函数调用的形式、变量和子函数的命名方式等。

保证各个函数的源代码中有注释说明，方便用户阅读理解程序。

3 总体设计

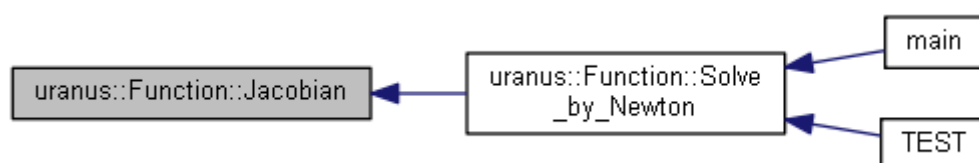
3.1 模块设计

调用模块：用户调用函数的形式和规则。

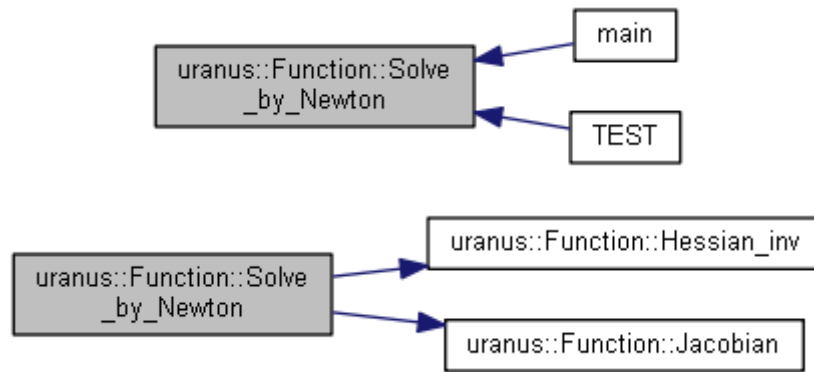
函数模块：各个算法的具体代码。

3.2 基本设计概念和处理流程

函数总体调用图：



函数分步调用图：



4 接口设计

4.1 用户接口

用户的调用函数。

4.2 内部接口

函数库调用函数模块。

函数模块使用用户输入的数据。

函数模块调用子函数模块。

5 运行设计

5.1 运行模块的组合

不论是管理员还是用户运行函数库，都需要将调用模块和函数模块结合在一起使用。

5.2 运行控制

调用函数时，需要使用函数库所规定的调用函数形式。

查看函数模块源代码时，需要使用编程语言的相关函数来查看。

5.3 运行时间

函数库基本上用户在使用此函数库，管理员很少使用，只是更新、添加相关模块。

各个函数模块的运行时间基本上可以达到此算法普遍的运行时间。

5.4 运行效果

各个函数模块所得到的结果都可以达到此类算法的普遍运行结果，满足用户对算法性能的需求。

6 出错处理设计

6.1 出错输出信息

当用户调用函数出错时，函数库会根据用户使用时出错的情况，将错误信息、错误对应的部分、此处错误要求的规范等信息都会输出给用户。

6.2 出错处理对策

当用户调用函数出错时，用户必须修改相关代码，才能处理此处错误。

7 维护设计

函数库比较小，而且各个算法模型都分成各个函数模块编写，每次维护时只需要对相应的函数模块进行维护就行。

概要设计说明

一、 目的

本测试报告为 Uranus 库项目的测试报告，目的在于总结测试阶段的测试情况以及分析测试结果，描述系统是否符合需求并对测试质量进行分

析。作为测试质量参考文档提供给用户、测试人员、开发人员、项目管理者 and 需要阅读本报告的其他质量管理人员阅读。

二、 测试概述

2.1 测试对象

Uranus 库：一个用于解决数学和优化问题的小型库。

2.2 项目背景

为了致敬 1801 年 9 月高斯用数学方法预测并发现了谷神星 (Ceres)，谷歌将自己开发的数学库命名为 ceres-solver。

Ceres Solver 是一个开源 C++ 库，用于建模和解决大型复杂的优化问题。它是一个功能丰富，成熟且高性能的库，自 2010 年以来一直在 Google 生产。Ceres Solver 可以解决两种问题。

- 具有边界约束的非线性最小二乘问题。
- 一般无约束优化问题。

希望我们实现一个小型数学库作为 AI5281L 软件工程这门课程的大作业，同时希望该库能够用于解决本学期其他学院选修课程的相关问题：

- AI5276L 最优化及其应用
- AI5278L 模式识别

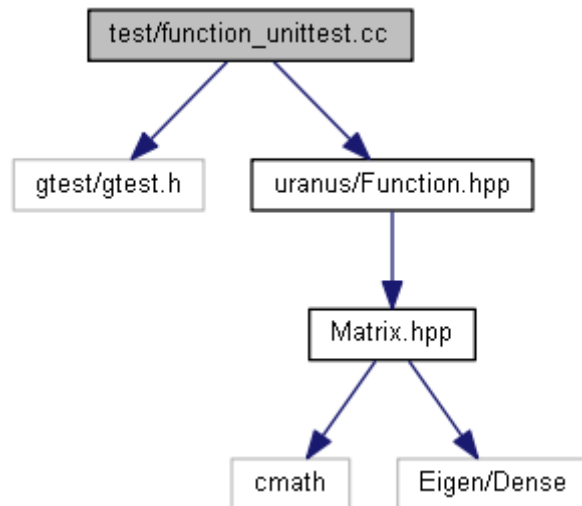
我们的项目以“海王星（天王星）”为名，因为 Uranus 也是一颗“被发现”的星体。

海王星是太阳系中距离太阳最远的行星，在 1846 年 9 月 23 日被发现，是唯一利用数学预测而非有计划的观测发现的行星。天文学家利用天王星轨道的摄动推测出海王星的存在与可能的位置。

三、 测试方案

3.1 Function_unittest.cc

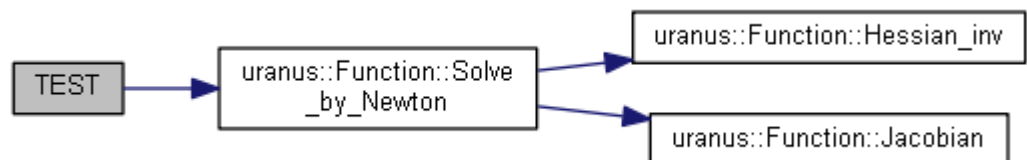
3.1.1 Function_unittest.cc 的引用(Include)关系图



3.1.2 函数

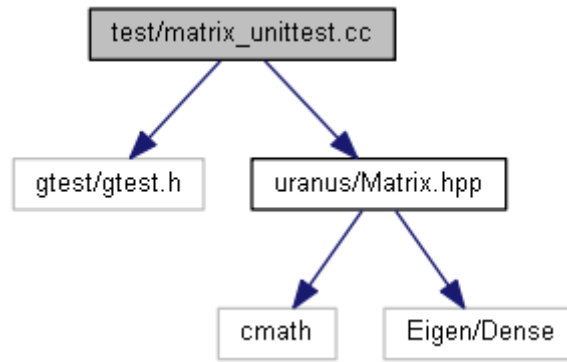
TEST (Function, Newton)

3.1.3 函数说明



3.2 Matrix_unittest.cc

3.2.1 Matrix_unittest.cc 的引用(Include)关系图



四、测试结论

4.1 功能性

系统正确的实现了各函数库的基本功能，实现了数据内容的多语言功能。实现了基础数据库的计算。

4.2 易用性

现有系统的可靠性控制不够严密，很多控制是通过页面控制实现的，如果页面控制失效，可以向数据库插入数据，引发错误。

现有系统的容错性不高，如果系统出现错误，返回错误类型为找不到页面错误，无法回复到出错前的状态