

矩阵向量模板类的项目设计思路和测试说明

October 5, 2021

1. 项目设计思路:

1. **Matrix.h**的实现:

- I **Matrix.h**头文件构建了模板类 **Matrix<T>** 及其派生出的模板类 **RowVector<T>** 和 **ColVector<T>**。因为方阵和向量在结构上有所不同，因此对于每个类，都有不同的构造函数、拷贝构造函数和析构造函数。
- II 在 **Matrix** 类内，设计了 **is_square()** 函数来判断矩阵是否是方阵，并且重载了操作符 **=** 和 **()** 以方便矩阵的赋值操作。这些都是两个派生类可以访问使用的。
- III 在类外重载了 ***** 运算以便在不同情况下适应矩阵之间的乘法运算和向量之间的内积运算，并且将它们相应地作为模板类的友元。
- IV 对于 ***** 运算无法进行的情况，程序有相应的报警函数 **abort()** 并且强制退出;对于下标越界的情况，也会在程序内进行报警和强制退出。
- V 对于某些多次在文件中使用的代码(如数据复制)，采用统一的函数(如 **copy_Mem()**)以便统一修改。

2. **main.cpp**的实现:

- I 使用 **random generator**，用户仅需输入矩阵和向量的大小，程序将自动生成随机的矩阵。
- II 对于随机生成的矩阵和向量，分别进行矩阵 ***** 矩阵，列向量 ***** 行向量，行向量 ***** 列向量，行向量 ***** 行向量，列向量 ***** 列向量的运算并输出结果。
- III 如果发生输入内容错误(输入的参数个数小于程序运行所需参数个数)，有相应的函数 **exitAbnormal()** 进行报警和强制退出。

3. **Makefile** 的实现: 管理 **main.cpp**的编译工作。使用 **make** 命令直接对其进行编译，并生成可执行文件 **test**。

4. **run** 的实现: **run** 文件通过 **for** 循环来控制程序运行的次数以便控制测试产生的数据数量。

2. 测试说明:

1. 测试输入:

- I 运行 **make** 命令, 自动编译 **main.cpp** 并生成可执行文件 **test**;
- II 运行 **bash run** 命令后, 在同一行后输入九个正整数(用空格间隔), 它们的含义如下。第一个参数: 程序 **main.cpp** 运行的次数; 第二、三个参数: 矩阵 **Matrix1** 的行数和列数; 第四、五个参数: 矩阵 **Matrix2** 的行数和列数; 第六、七个参数: 行向量 **Rowvector1** 和 **Rowvector2** 的维数; 第八、九个参数: 列向量 **Colvector1** 和 **Colvector2** 的维数。(注意: 若为保证矩阵乘法和向量内积有意义, 根据其数学原理, 第三个参数应和第四个参数相同, 第六至第九个参数都应相同。)

2. 测试输出:

- I 每一次运行 **main.cpp**, 首先输出的是随机生成的 **Matrix1**, **Matrix2**, **Rowvector1**, **Rowvector2**, **Colvector1** 和 **Colvector2** 的值;
 - II 输出对 **Matrix1** 是否是矩阵的判断;
 - III 输出 **Matrix1 * Matrix2**, **Colvector1 * Rowvector1**, **Rowvector1 * Colvector1**, **Rowvector1 * Rowvector2** 和 **Colvector1 * Colvector2** 的值。
3. 注意事项: 因为用自己编写的 **C++** 矩阵乘法程序来验证本程序是毫无意义的, 所以我用 **Matlab** 对测试数据进行了检验, 都与 **Matlab** 的结果相同。此外, 为了便于同时进行手工验算, 当前 **main.cpp** 的具体模板类型是 **int**, 产生的是 **[0, 10]** 的随机整数。如要修改具体的模板类型, 只需对宏定义和随机数均匀分布模板类两行代码进行修改即可。