S T A P M A T 程序介绍

STAP程序最早为K. J. Bathe教授配合其教材采用Fortran IV所编写的教学程序；清华大学航天航空学院张雄教授在此基础上用Fortran 90语言进行了改写，并删除了过时的一些Fortran语言特性，增加了一些新的模块；为配合课程教学需要，清华大学航天航空学院赵雷洋博士生和刘岩副教授将其改为Matlab版本。

1. 程序结构介绍

结合计算类课程的特点，相比较Fortran版本和C++版本的STAP程序，Matlab版本程序的定位是尽可能降低二次开发难度，使得同学们的关注点主要集中在算法方面。因此，本程序在一定程度上牺牲了运行效率，以模块化为主要特征，规范程序结构，以便于同学们理解、应用以及开发程序。

程序根目录下包括SRC和Data两个文件夹，其中SRC文件夹下存放的是程序调用的各种函数，而Data文件夹下存放的是输入输出文件。STAPMAT的输入文件与STAP90的输入文件只有两点区别，一是无需严格按照固定列输入，只要变量之间隔开一个空格即可；二是MODEX变量扩展了不同的求解器，除了STAP90中的0（数据检查）和1（变带宽存储的LDLT分解求解）之外，还提供了2（MATLAB中稀疏矩阵的左除求解），同学们也可以按照需要自己进一步扩展求解器。

下面将按照程序运行顺序具体介绍每个模块的内容。

1.0 函数的帮助信息

高度的模块化使得程序中出现频繁的函数调用，为了便于大家理解，在每个函数文件的开头都标明了该文件中主函数的帮助信息。

MATLAB的函数文件中允许存在多个函数，其中与文件名相同的函数被称为主函数，其余函数均是为主函数所服务的。在函数文件中，仅主函数可以被其他文件所调用，因此主函数的帮助信息非常重要。在本程序中，主函数的帮助信息主要有三大部分，分别是函数调用情况、函数被调用情况以及程序编写信息。

函数调用情况，即主函数中调用了那些函数，这些函数又是在那里定义的。一般来讲，函数定义信息的结构为“路径+文件名 - 函数名”，当调用的函数是本函数文件中的辅助函数时，省去路径，只剩下“文件名 - 函数名”，如“Solve.m - WriteDis()"；当调用的函数文件与当前文件在同一目录下，函数定义信息为“./文件名-函数名”，如"./LDLTFactor.m - LDLTFactor()"；当调用的函数文件与当前文件位置差异较大时，往往采用以SRC文件夹为根目录的表示方法，如"SRC/Mechanics/GetStress.m - GetStress()"。

函数被调用情况，即本文件中的主函数是为哪个函数服务的。该函数定义信息与函数调用的定义信息类似，只不过不再标出函数名（因为出于结构清晰的考虑，仅主函数会调用其他文件中的函数），如"SRC/Mechanics/GetStiff.m"。

程序编写信息并不影响同学们对程序的理解，但对于多人合作开发的程序，注明作者和时间有利于团队准确统计各自贡献和快速定位责任主体，当然采用分布式版本控制系统GIT能够更好地做到这一点。

1.1 BasicData模块

由于MATLAB在函数调用过程中普遍使用数值传递而非指针传递，因此本程序使用了全局”数值类”的方法。Class作为MATLAB中的面向对象数据结构，有着很多复杂的用法，但我们这里仅用其存储特定的变量，可以简单的认为是一个结构体。通过将声明的类的对象定义为全局变量，实现各个函数中的变量调用，避免了复杂的参数传递过程。

ControlData.m定义的是ControlData类，由此声明的cdata对象，存储了多个需要在不同函数中使用的、用于控制程序流程和关键信息的变量。不过该对象中多为标量，总体占据内存不大。

SolutionData.m定义的是SolutionData类，由此声明的sdata对象，存储了包括坐标、节点、单元、刚度阵等多个力学变量。由于这些变量所占内存较大，为了提高效率，一般要求使用前进行内存的预分配（如使用zeros()）函数。并且在具体使用时，先将对象中的变量复制到局部变量中，对局部变量进行操作，最后再将局部变量复制到对象中，这样也会显著提升运行效率。注意，MATLAB中的矩阵是按列存储的，所以需要注意控制变量的存储结构。

1.2 Initiation模块

为了对应输入文件的结构，本程序会通过该模块下的ReadFile.m 读入输入文件中的控制信息、节点信息和载荷信息，并通过WriteParasOut.m 输出到OUT文件中。在读取完前两行信息（标题和控制参数）后，调用ReadFile.m 中的辅助函数进行变量的初始化，然后读取节点信息，随即计算求解时方程的个数。之后读取输入文件中的载荷信息，也随即计算出对应的载荷列向量。

1.3 Mechanics模块

由于不同单元类型中，单元信息和材料信息是不同的，因此在本模块中实现材料信息和单元信息的读取，并计算刚度阵。

该模块有一个总体控制函数GetStiff.m，该函数被主程序stapmat所调用，并根据之前读取的单元类型，负责调用不同单元的函数。同时，在Mechanics文件夹下的其他函数，都是是针对任意类型单元的通用函数，往往被其子文件夹中的函数所调用。

1.3.1 Truss模块

Truss模块定义了桁架单元的相关函数，总共有三个函数文件。

TrussStiff.m 和ReadTruss.m 是在计算刚度阵时调用的，TrussStiff()函数为 总体控制函数，实现了单元信息的初始化、材料和单元信息的读取，以及单元刚度阵的计算和集成。

TrussStress.m 是在求解位移后调用的，实现了节点应力的计算。

1.4 Solver模块

与其名字相对应的，该模块主要负责求解。当程序运行到此模块时，已得到了总体刚度阵和载荷列向量，只需要求解线性代数方程组即可得到位移向量。本程序目前提供了两种求解方法，一种是针对变带宽存储的刚度阵的LDLT分解求解，一种是MATLAB提供的大型稀疏矩阵的左除求解。

LDLT分解求解主要调用了两个函数，分别是分解函数LDLTFactor()和求解函数ColSol()。先对刚度阵进行LDLT分解，然后针对每一种载荷情况进行求解，能够有效节省分解所用的时间。

稀疏矩阵求解，需要先将变带宽存储的刚度阵转换为稀疏矩阵，使用的是Solve.m文件中的辅助函数Stiff2Sparse()。转换完之后，针对每一种载荷情况进行左除求解。尚未验证该求解效率是否比LDLT分解求解效率高。

2. 二次开发技巧

2.1 开发所修改的程序部分

关于STAP90程序的二次开发主要有两类，分别是添加单元和添加求解类型。

由于程序中将单元类型影响到的部分分割到了Mechanics的子文件夹中（如 SRC/Mechanics/Truss)，因此添加单元时，只需要增加一个子文件夹，并实现对应的功能（材料和单元信息读取，刚度阵计算，应力计算），并在GetStiff.m文件和GetStress.m文件中稍作修改即可。

添加求解类型（如特征值求解和时间积分），只需要改动Solve.m文件，并实现相应的功能即可。

2.2 刚度阵计算的检验

在添加单元过程中，最容易出现的问题就是刚度阵计算错误。通过手动计算少量单元的刚度阵，并与程序中的刚度阵对比，逐一排查程序内容，有利于快速修正计算刚度阵的代码。由于组集刚度阵的函数不用修改，因此一般问题会出现在形函数的数值和节点顺序上。