CAE软件研发的一些思考(5)--系统的开发求解器

原创 www.cae-sim.com 多物理场仿真技术
收录于合集 #工业软件杂文系列 23个

文章最早写于2015年，展开后用来指导了一些公司和研究所的求解器研发。在当时来看能满足一些小规模的研发。考虑到现在硬件快速发展，以及求解规模，求解内容复杂度增加，以及国内公司研发升级，开发人员增多，文章内容已经不再满足实际需求，不过对于求解器开发入门仍然具有一定参考意义。

目前国内大部分求解器，以及开源求解器的开发仍然停留在小作坊式的水平，开发出来的程序能实现基本功能，但在稳定性，扩展性，测试性，维护性等方面差强人意，很难达到商业应用的要求。

严格意义上来讲，求解器的开发也属于软件开发的范畴，理应用软件工程的思想来指导，但由于求解器本身有其特殊性，开发流程也不适合完全按照一般软件开发流程来做。

本文结合自己开发经验，讨论一下如何系统的开发求解器。求解器开发可以分为三个阶段：

1.原型开发；

2.迭代开发；

3.维护开发；

1. 原型开发

这阶段主要完成以下任务：

1.1.技术选型；

确定要实现的功能，使用的开发语言，开发环境和工具。目前大部分求解器开发使用C/C++/Fortran语言

1.2 实现基本功能；

要能对最简单的例子进行计算，并得到正确的结果。需要做的工作：

- 能生成标准求解器的输入文件，比如Nastran，Ansys，HFSS，Fluent等的求解器输入文件，例子的计算结果要与这些标准求解器计算的结果做比较。
- 标准求解器输入文件的解析器。用来解析输入文件，作为开发求解器的输入数据。
- 比较标准求解器的计算结果和开发的求解器结果。

这阶段的主要目的是保证算法的正确性。开发时为了提高效率，可以借助Matlab软件: 用Matlab完成原型的开发，直到计算结果正确。在此基础上再将Matlab翻译成 C++/Fortran。这样在早期可以将精力集中在算法验证上。需要注意的是尽量进行模块化开发。

1.3. 完成求解器原型；

这里需要介绍一下Matlab软件，基本介绍看百度，主要说一下Matlab混合编程。Matlab有工具是可以把M文件翻译成C++的，不推荐。主要介绍如何把C++/Fortran文件编译成Matlab文件。这个功能很有用，当进行模块化开发的时候，C++/Fortran完成模块功能，然后编译成Mex文件，作为Matlab的模块使用。这样可以逐步将Matlab翻译成C++/Fortran，提高开发效率。

小结：

1>需要开发一种标准求解器文件的解析器。

2>需要熟练使用标准CAE软件进行仿真，熟悉求解器输入文件和计算结果

4>开发的求解器要能正确计算经典的Benchmark例子

原型开发决定了开发的可行性，如果这阶段的任务无法完成，需要加强研发的投入。

2. 迭代开发

这阶段主要完成以下任务：

1. 完善新功能

在完成原型的基础上，添加新功能，比如支持新的单元类型，支持新的荷载边界，处理更复杂的模型等。

2. 保证计算准确性基础上，进一步提高求解器的质量

可靠性：正确的模型，都能给出可靠的计算结果；

鲁棒性：任何例子都能给出正确的反馈；

稳定性：大规模计算时，程序能保持稳定；

效率：计算速度，内存消耗。考虑GPU，并行计算；

3. 完善求解器的前处理和后处理：

有限元模型检查；

网格质量检查；

仿真结果分析；

4. 创建更多经典的Benchmark例子进行测试。

小结：

迭代开发阶段的主要目的是完善求解器，建立规范化的开发流程：

1> 确定技术选型，比如线性方程组库的使用，并行计算，GPU等

2> 完善前处理和后处理

3> 建立更多经典Benchmark例子，例子的选择需要 有经验的工程师的参与

4> 确定求解器输入文件格式

5> 定期发布版本以供测试

3. 维护开发

这阶段主要完成任务有：

1. 测试实际工程的例子，处理实际工程中所碰到的问题

实际工程的模型要远比经典模型复杂，求解器需要更多的功能支持计算实际的模型。

2. 建立回归测试机制

回归测试是求解器开发中非常重要的一环，通常求解器修改后，需要验证是否对以前的case有影响，这就需要建立回归测试机制，通常用一种脚本语言（Python,Perl）开发回归测试程序。每次修改代码后，运行程序，比对修改后与修改前的计算结果。

可以看出要开发出高质量的求解器，既要熟悉求解器本身的算法，又要了解软件开发流程，更要熟悉软件工程中的架构，复用，重构，模块等思想，对开发人员提出了更高的要求。