

多物理场仿真技术节选 2

再次节选了“多物理场仿真技术”公众号部分内容，相比于 1，更偏向于软件技术研发。大部分内容都能在相关教科书和网络上找到。文章更像是针对多物理场软件研发的汇总。

欢迎关注：

www.cae-sim.com

扫码关注公众号：



邓子平

一篇文章入门仿真软件性能优化

原创 cae-sim.com 多物理场仿真技术



之前有讨论过关于支付宝和工业仿真软件数据特点的对比，工业仿真软件慢的根本原因在于模型的整体性和难以拆分。

根据经验，笔者将加速仿真软件性能方法，归纳为四个方面：

1. 分治(Divide And Conquer)
2. 单一数据模型(Single Data Model)
3. 优化高频操作
(Optimize High-frequency Operation)
4. 改进编程语言使用和底层算法
(Improve Language Usage and Basic Algorithm)

1. 分治(Divide And Conquer)

1.1.分治是最容易理解的，从最简单的文件读写说起。在有限元模型中，通常我们会把网格文件保存在一个文件中。文件会包含节点，单元，边界荷载材料等所有信息。把所有信息放在一个文件的好处是容易处理，但是当文件规模达到G的级别时，比如10G，读写就会出现性能瓶颈。网格文件中通常节点和单元会占据绝大部分数据，合理的做法就是分出两个进程，一个读节点，一个读单元；进一步可以将节点单元放在不同文件中。在后处理文件中，存在众多结果数据，对于10G网格的模型，后处理器根本无法一次性装下，分开不同文件处理是必然选择。现在固态硬盘的读写速率普遍达到500M/秒以上，硬件I/O已经不是瓶颈，需要改进的是软件策略。

1.2.有朋友会说，这个分治操作太简单了。是的，因为网格数据节点和单元本身就是分开的，我们只是在文件级别进行分治，进一步比较难的是在模型数据层面进行分治。目前（2020年）大部分CAD或仿真软件都很难导入一个1G的step文件，原因就在于Step文件是采用的类似索引结构存储信息，无法并行；为了提升性能，可以写一个转换器，将其按照某种属性分类，分别保存为文件，再次读入的时候就容易并行化，特别适用于需要反复读入的情况。

1.3.在仿真软件中，数据类型主要分为业务数据，几何模型数据，渲染数据，网格数据，求解器数据，结果数据。分别说一下这些数据的特点：业务数据多为属性或者参数，控制等数据，特点是数据量小；几何数据大多用BRep或者参数表示；渲染数据多为点数据和三角形，网格数据为节点和单元，荷载，边界，工况等；求解器数据为各种大规模矩阵；结果数据在网格数据基础上，给节点或单元附上各种属性数据，或者对原始数据进行二次提取和加工。

对同一个模型大小的估计，不同数据类型和表达方式可以得出不同结论。比如计算材料学中的一个晶胞，是一个长方体，业务数据可能就只有一个点坐标；对于参数几何而言就是简单的长宽高；对渲染而言是12个三角面片(每个面两个三角形)；对网格而言，如果网格划分的很细，可能是百万的三角形；对于求解器而言，高阶单元在网格的基础上，还要乘上好几倍；对于仿真来说，模型大小主要由最终求解器数据的矩阵规模来确定，矩阵的规模也就是要求的自由度的规模。

使用分治方法之前，需要明确对象是否合适分开处理，如果能满足以下条件就可以采用：

1. 数据之间耦合度低；
2. 分开之后的数据保留各自的属性，尽量保留整体属性；
3. 分治的相关数据操作之后能再次组合在一起；
4. 分治的各个数据规模开销相当

理清了数据的特点和分治的要求之后，我们会发现分治的核心是将模型数据分离，但是这与之前提到的仿真模型是整体的概念相互矛盾，这就是仿真模型加速的难点所在：既要分治，又要保持整体性！

几何数据中，我们尽量在设计阶段将几何分开设计，比如四条相同腿的桌子，桌面和腿分开设计，不要进行布尔运算，对于腿，不要使用深度拷贝操作，尽可能使用类似实例操作，也就是创建一根腿，其它三个复制原始几何+变换矩阵，这样可以省下建模和离散渲染的开销，在数据量多的时候大幅提升性能。对原始几何数据根据属性进行分类，提取特征，在需要全局进行数据计算时首先利用属性特征进行检索。最典型的例子是在全局进行干涉检查，如果任意两个对象两两检查，算法复杂度为 N 的平方，数据量大的时候行不通，而通过类似Octree, kbtrees的结构，将所有对象分开处理，只检索对象附近的区域，会大大提升性能。这是利用空间树实现分治的一种方法，实际中可以根据业务属性分组，也就是利用分治的办法提升性能。

在求解器中，最终我们需要求解大规模线性方程组，之前介绍的Krylov子空间方法也是比较典型的分治法，即将矩阵进行降阶，分成维度更低的矩阵，再进行求和运算；而用于快速求解满秩矩阵的快速多级，本质上也是分治法，将矩阵分成不同的区域，先对不同区域内进行计算，再在区域外计算，最后组合结果，快速多级将满秩矩阵的求解复杂度保持在了线性水平。求解器的分治算法是分治方法中最难的部分，也是属于比较底层的算法。

将模型数据进行分治之后，剩下的操作就相对容易了，主要通过多线程，多进程，群组服务器，而这些操作可以自己开发，也可以利用已有的MPI工具，如OpenMP，OpenMPI。需要提一下GPU计算，在渲染数据和矩阵计算中，可以利用GPU来进行，而非CPU，目前主流仿真软件都支持GPU计算，这个之前也有专门的相关介绍，参考附录。

2. 单一数据模型(Single Data Model)

这个也比较容易理解，对于大模型，始终只保留一份数据，避免大数据的复制拷贝等操作，最好从制度上保证这种现象出现，比如核心数据只能使用指针，或者智能指针；对于临时出现的大数据，用完即销毁；对于海量小数据，自动分配大内存池和活动空间
单一数据模型说起来很容易，但实际中很难做到，主要还是对大模型的操作比较繁琐，也难调试

3. 优化高频操作

(Optimize High-frequency operation)

高频操作也是影响性能的一个重要因素。以求解器数据更新操作举例，比如一个单元被修改，常规做法是会立马通知并更新矩阵，如果修改了十个单元，那就会通知十次，对于小模型，这种操作很难觉察到性能问题，但如果是修改百万的数据，这种性能影响就非常明显了；理想的做法是修改数据后不要马上更新，而是等到需要使用数据计算之前再去统一更新。

控制高频操作的典型方法是使用缓存方法和之前介绍设计模式提到的Observer模式。即将需要频繁更新的数据进行缓存，只对缓存数据进行操作，通过Observer在适当的时候进行更新，避免全局进行频繁操作。高频操作无论是在几何，网格，还是求解器都是影响性能的一大瓶颈，而且是完全可以通过简单的技术手段避免的。

此外通过数据分级也是有效解决高频操作的方法，即根据业务逻辑，尽可能采用金字塔层级而非扁平的数据模型，这种结构有利于从架构上实现最小权力原则和高频局部数据操作，提升性能。

4. 改进编程语言使用和算法

(Improve Language Usage and Basic Algorithm)

如果做好了之前的三点，那剩下的就是需要从语言和底层算法方面改进了，比如C++中合理使用vector，map，unordered_map；避免多重循环，慎重选择复数类等等，这个需要的是长期开发积累。

此外，一些基础算法也要重视，比如自己开发面面求交算法，性能，稳定性，准确性要非常好的拿捏，以符合业务需求，用户可能只需要判断是否相交，而不需要求交线，不求交线可以提升性能，对于不可避免的高频计算，可以节省不少开销。

其实关于工业仿真软件性能的话题早有探讨，只是没有系统地分析。工业仿真软件追求的不是一种功能，黑科技，或先进的技术手段，看重的是准确性，稳定性，可靠性和性能。当然这几种指标并不是独立的，比如加密网格会提高准确性，但会降低性能，一般来说准确性是优先考虑的；检查各项数据的完整性和正确性可以提高稳定性和可靠性，但无疑也要消耗更多资源；所以在实际研发中这些指标有可能相互排斥，需要根据具体的业务来选择相应的策略。

工业软件的性能问题是一个可以一直讨论的话题，后续还将结合实际做更多探讨。

一篇文章入门求解器模型降阶方法

FEM之在求解器中使用设计模式(8)---Command和Observer模式

一篇文章入门HPC(高性能计算)

FEM之求解器加速(1)---HPC简介

BIM中仿真杂谈

原创 www.cae-sim.com 多物理场仿真技术

收录于话题

#软件研发测试工程师

15个



其实一直对建筑业不太关注，修路造房子造桥通常都是大手笔，从设计角度讲都是按照规范来，从验证角度看又是实验数据说话，仿真在设计中的地位不高。而且一提建筑给人就是傻大粗的感觉，但建筑业是个赚钱的行业，基建通常一投就是几百亿！近几年放水力度加大，建筑行业多多少少都会受益。

BIM简介

现在提建筑业就不能不提到BIM(Building Information Modeling)，BIM其实是个大话题。

百度百科对于BIM是这样定义的：

建筑信息模型（Building Information Modeling）是以建筑工程项目的各项相关信息数据作为模型的基础，进行建筑模型的建立，通过数字信息仿真模拟建筑物所具有的真实信息。它具有信息完备性、信息关联性、信息一致性、可视化、协调性、模拟性、优化性和可出图性八大特点。

BIM技术是一种应用于工程设计建造管理的数据化工具，通过参数模型整合各种项目的相关信息，在项目策划、运行和维护的全生命周期过程中进行共享和传递，使工程技术人员对各种建筑信息作出正确理解和高效应对，为设计团队以及包括建筑运营单位在内的各方建设主体提供协同工作的基础，在提高生产效率、节约成本和缩短工期方面发挥重要作用！

简单讲，就是将建筑的整个生命周期 信息化，数据化，智能化，优化从设计到施工的整个流程，降低建筑成本。更通俗一点：建模、管理、算量、演示、方案等合起来就叫BIM。

在BIM中是否需要仿真呢？答案是肯定的，只不过仿真在BIM中的地位没有那么多高，甚至连BIM链上的一环都算不上，但是并不影响我们用结构分析软件，毕竟CAE市场的1/3都是结构。

从仿真角度看，建筑结构通常需要进行如下仿真分析：

1. 最基本的结构静力分析；
2. 模态分析/ 动力学分析；
3. 对于大型建筑，流体分析；
4. 声场（隔音）、热分析（散热），电磁（手机信号传播），光分析（彩光）等。

其中的静力/动力学分析是重点，主要是各式各样的荷载，地震，风载，随机荷载等等。

仿真：

抛开工程实现，仅从有限元编程实现角度看，建筑的结构分析相对简单的，有限元单元通常都是梁，杆，刚体，弹簧，索等0维1维单元，再复杂也就壳单元，所以建模也相对简单。

软件：

市面上的大部分CAE软件都能做结构分析，但专业的BIM仿真软件似乎不多，建模的如Revit, Bentley, 仿真的韩国的Midas, 国内的PKPM。

BIM对于结构仿真是有促进作用的，以前仿真计算作为单独的一环，和设计施工脱节，现在被接到BIM链中，其优点在于，仿真信息可以最大化利用，设计CAD做了修改，可以无缝导入到仿真模块，而仿真结果也可以给研究设计施工等提供信息，大数据云端的快速发展使得仿真结果可以在任意用户端访问，想象着以后施工碰到问题甚至可以实时的进行仿真，查看结果，出施工图，而不用再回设计院来回折腾。

附录：

住建部印发《2016-2020年建筑业信息化发展纲要》

仿真软件开发工具介绍（1）--ACIS/HOOPS

多物理场仿真技术

CAD/CAE/EDA/CFD/PLM软件产品发展历史很长，涉及到的知识面也非常广：计算力学，电磁学，热，声，流体，计算图形学，并行计算，几何造型，各种数值计算算法，网格生成，GUI设计，PDE，随便一个话题都是一个非常大的领域，单独研究开发都能做上好几年。

开发仿真类的商业软件，讲究的是高效率，快速研发出产品，及早推向市场。在国内IP保护不够，研发投入不足的情况下，选择合适的开发工具就非常重要。

作为科普，**仿真软件开发工具介绍系列**将会介绍一些适用于商业开发的工具。仍然，目标是开发能向国际主流CAE产品看齐的产品。希望对国内研发企业选择开发工具有些帮助。介绍的工具有免费的也有商业的，这些工具都有成功应用的经验，对于商业开发工具，因为都是本人使用过的，评价会比较客观，从而避免打广告的嫌疑。

仿真产品开发广泛用到的两个商业组件HOOPS/ACIS。之所以放在一起介绍是因为这两个组件目前是捆绑销售。

HOOPS是Techsoft3d公司（最早由Autodesk出来的一帮人创建的公司）开发的一个图形显示产品，提供了核心的图形架构和显示功能。我们知道在各种数值计算软件产品中，需要对几何模型和有限元模型，后处理结果等进行各种编辑操作和显示，传统图形显示开发工具为OpenGL，但是OpenGL的缺陷也是非常明显的，需要开发人员有较深的图形学功底，而且功能的开发周期很长。开发人员只需要有基本的图形学和编程水平，就可以使用HOOPS快速的搭建起图像显示和图形编辑的应用程序。

缺点：作为显示工具在几何造型功能方面很弱，性能还可以提升。

ACIS是Spatial公司的产品，2000年公司被达索收购，相比HOOPS名气要大一些，ACIS利用的是B-rep即边界表示法来表示几何。ACIS是最早开发并且商业化比较成功的三维几何造型引擎。简单讲可以利用代码创建出各种复杂的三维几何，并且支持复杂的几何操作，比如布尔操作，装配，特征识别，网格划分等。ACIS的文件名后缀为*.sat和*.sab（sat为ASCII码文件，sab为二进制文件），作为事实上3D几何工业标准之一，大多数的三维CAD产品和仿真产品都支持*.sat文件的导入导出。

个人认为，ACIS最大的优点在于两个方面：稳定性和标准。对于复杂几何的生成和编辑高度依赖其稳定性，从开发角度来讲，复杂的几何出了问题，几乎是无法调试的。作为事实上的3D图形标准之一，使用了ACIS也就意味着能和大多数的软件产品兼容。

缺点：多线程支持弱，性能还有很大提升空间，显示模块弱。

作为两个组件功能互补，Spatial公司开发出了ACIS/HOOPS bridge,即AICS用来造型，HOOPS用来显示ACIS的模型。总之利用AICS/HOOPS可以快速开发出三维CAD应用程序。

这两个组件在国外有很多成功应用；在国内也有企业在用，但总体量并不大，主要原因还是在价格上偏贵。如果开发的产品附加值不高的话，采用这套系统得不偿失。也从另外一个方面说明了自主研发和核心技术的重要性：没有核心技术，基础的第三方产品都买不了。

仿真软件开发工具介绍(5)---Simmetrix

原创 www.cae-sim.com 多物理场仿真技术



经常关注本公众号和多物理场仿真技术博客的朋友应该很了解作者对工业仿真软件CAD/CAE/CFD/EDA/TCAD以及国内现状的观点。

简单讲：工业仿真软件既不是可有可无的CAD附属品，也不是高精尖的核心技术产品。从研发角度看，工业仿真软件的技术门槛一直在降低，从产品设计角度看，在产品生命周期中的地位会越来越重要。

在之前研发工具文章系列中介绍了ACIS/VTK/HOOPS/OSG以及OCC，本文再介绍一款稍重量级的研发工具：[Simmetrix](#)

Simmetrix 公司网址

<http://www.simmetrix.com>

公司成立于1997年，至今已有20多年历史，主要提供基于仿真的工具库（注意是研发工具，不是终端软件产品），主要包含了网格划分，几何重建，仿真模型处理等一系列工具。

MeshSim 网格方面：

1. 三角形、四边形或四边形表面网格的生成
2. 支持装配和非二维流形几何模型(三维实体、壳、线和嵌入表面的混合物)
3. 支持四面体和六面体网格生成
4. 网格尺寸控制(全局到模型或局部到模型实体)
5. 基于网格单元质量的控制(纵横比、倾斜等)
6. 基于曲率的自动细化，在高曲率区域生成较小的元素
7. 周期性边界条件的网格匹配，创建和模型实体对完全匹配的网格
8. 各向异性网格(创建与方向相关的网格，包括基于模型曲率的网格)
9. 支持网格加粗和网格加密
10. 专门支持CFD的边界层网格划分
11. 并行网格划分

12. 根据用户提供信息或者几何特点进行自适应网格划分

GeomSim 几何方面：

- 1.提供特征识别，模型简化，模型调整和修复功能
- 2.支持离散模型（STL）
- 3.提供直接读入主流三维CAD内核 ACIS/Parasolid/以及NX/Pro/E/Granite/SpaceClaim/SolidWorks

FieldSim

数据管理：

1. 管理有限元数值计算模型，提供查找，计算，集成，分析等功能

三维图像处理，类似于Simpleware

1. 直接将三维扫描图像转换成可用于有限元分析的网格模型

以上只是列出了Simmetrix部分功能，早年曾试用Simmetrix组件，功能稍显单薄，但主要功能覆盖了一款通用仿真软件的相当一部分核心功能，能大大减少研发的工作量和难度，作为一般仿真软件产品开发工具，是一个不错的选择。

仿真软件开发工具介绍（6）--MeshGems/MESQUITE

原创 www.cae-sim.com 多物理场仿真技术



今天推荐两款针对仿真软件研发的网格工具。

注意：是提供给开发人员的工具，不是最终仿真用户使用的产品。

MeshGems 是法国的一家公司，其最早的作者是网格算法领域的专家，目前市面上相当一部分网格专著都由其本人以及学生所著。该公司后拆分成两家，独立运营。其中之一网址：

<http://meshgems.com/>

MeshGems历史可以追溯到三十多年之前，是目前使用最广，功能性能稳定性最好，影响最大的商业网格引擎。主流商业仿真软件，比如Nastran，Abaqus，Ansys，SolidWork，LS-DYNA 以及ETA/Autodesk/PTC 仿真 产品，乃至专业网格工具HyperWork都是其用户。MeshGem几乎可以满足所有商业求解器对网格在功能，稳定性，性能方面的需求，实至名归的商用“网格一哥”。

目前MeshGems提供了主流CAD内核接口，包括Parasolid，OCC，ACIS，基于OCC的Salome，并可以导出为标准的Nastran求解器deck文件格式（笔者多年前提出的需求）。

对开发网格功能而言，不考虑成本的话，MeshGems首选。

MESQUITE

MESQUITE是美国桑迪亚国家实验室开发的一款开源网格质量改进工具，可以在已有网格基础上提供网格平滑，尺寸修改，加密等功能。对于网格优化尤其是对网格质量要求很高的求解器，MEQDUITE是一个不错的选择。

更多信息可以搜索“多物理场仿真”新浪博客。

仿真软件开发工具介绍（7）--Parasolid/OpenCascade

原创 www.cae-sim.com 多物理场仿真技术

收录于话题

#仿真几何相关

5个



介绍的仍然是开发工具，不是用户最终使用的产品。

前面介绍过ACIS，今天再介绍一下Parasolid/OpenCascade

两款都是基于Brep结构的三维几何内核。相关信息可以容易百度到，主要讲几个要点：

Parasolid 现在隶属于德国西门子，由收购UCS公司而来；

1.Parasolid最早由UCS收购公司收购Sutherland公司得到，Sutherland 之前收购了公司ShapeData，ShapeData创办人正是B-rep结构提出者，也是ACIS的开发者，所以ACIS和Parasolid同宗同源

2.Parasolid 商用，使用需要购买

3.因为Parasolid开发时间在ACIS之后，所以早期Parasolid要比ACIS强，但是后来ACIS被达索收购，加强了研发和市场投入，在仿真功能支持方面，目前ACIS要强于Parasolid（来自于本人开发体会）

4.Parasolid导出文件后缀名为x_t、.x_b，分别对应文本和二进制格式

OpenCasCade 由法国公司

Matra Datavision开发维护，简称OCC或OCCT

1.OCC开源，代码可以免费查看和使用，包括商用；

2.开发人员可以申请，为OCC贡献代码；

3.经过多年发展，目前OCC相比早期功能已经强大很多。根据个人开发经验，如果ACIS/Parasolid功能打分100，OCC可以打70分。差距主要在性能，稳定性方面，很多功能方面也有所欠缺，但是用作一般的商用软件开发也已经绰绰有余。

CFD流体求解器算法--LBM 介绍

www.cae-sim.com 多物理场仿真技术

大多数人并不知道Powerflow这个软件，tb上也找不到枪版。但该软件在汽车行业比较有名。其快速，准确，便捷的特点成为许多汽车公司仿真首选。

Powerflow 提供了流体，热，噪音分析和优化等功能。而开发公司Exa (<http://www.exa.com/>) 按软件使用时间收费，说日进斗金一点也不过，是典型的埋头闷声发财的公司。

Powerflow的核心求解器采用了LBM算法（Lattice Boltzmann Method 格子波尔兹曼方法）。

在流体中的常用控制方程是 N-S方程，但在很多情况下该控制方程很难描述或者求解，比如复杂多相流系统，多空介质流体等，而从微观的分子动力学上建立的简化模型能解决这些问题。**Boltzmann 方程可以简单的理解为N-S方程在微观水平的表达形式。其基本思想是划分网格，计算每一个网格节点上离子的状态（碰撞，迁移，速度，方向，分布，属性等）**，当然实际计算中要复杂的多。其实很多流体现象都可以用微观的分子动力学来表示，这也是格子波尔兹曼方法的理论基础。

相比于目前流体求解器广泛使用的有限体积法，LBM优点很多，最主要的原因还是精度上优于其它CFD算法，而且适合使用大规模硬件集群进行HPC计算：

1. 数据预处理和网格生成效率高；
2. 方便支持并行数据分析，处理和评价；
3. 支持解决多相流的小液滴和气泡；
4. 支持解决复杂的几何形状和多孔介质；
5. 支持复杂，耦合流动与传热和化学反应；

采用LBM开发的商业软件xflow的西班牙公司Next Limit Dynamics在2016被达索收购，同样采用LBM的PowerFlow 也被达索收购，也从侧面说明了LBM在CFD领域成功应用

目前使用LBM的开源软件有 Palabos, OpenLB:

<http://www.palabos.org/>

<http://www.numhpc.org/openlb/>

FEM之求解器---开源求解器简介

www.cae-sim.com 多物理场仿真技术



求解器是仿真软件的核心，求解器的性能直接决定了软件的水平。

笔者多年从事CAD/CAE/CFD/EDA软件的开发，接触到了较多的开源求解器，根据笔者的经验，将开源求解器分为三类：

第一类是在精度，性能方面或者实用性上靠近商业求解器，可用于实用开发的

1. Code Aster

<http://www.code-aster.org/>

2. Calculix

<http://www.calculix.de/>

3. Elmer

<http://www.csc.fi/english/pages/elmer/sources>

4. OpenFOAM

<http://www.openfoam.com/>

第二类是逊于第一类，能解决某个领域的问题，有实用价值的

1. MaxFEM

2. Z88

3. Impact

4. Meep

第三类是适合于科研教学的(数不胜数)

1. OpenFEM

2. OOFEM

3. Dune

4. Matlab FEM toolbox

5. polyde-fem

6. 飞箭

...

个人认为开源求解器主要存在两个弊端：

1. 缺乏通用性，单元类型，边界，材料等支持的类型上，一旦出现特殊的case，就无法支持。最典型的是Calculix的居然不支持几何边上加荷载，虽然从实现角度来讲并不困难。
2. 缺乏实用的前后处理器，这对很多求解器的使用和推广是个很大的限制。后处理还好说一点，前处理器涉及到建几何，导入几何，网格划分，材料，属性设置等，每一项都需要比较专业的处理工具。很多开源求解器在CAD/Mesh等前处理做了很多工作，但仍然有很多局限性。

商业软件和开源软件一个根源性的区别在于生态链的隔离！

商业软件自底向上是一套独立的系统，从几何内核，网格，优化，求解线性方程组库以及很多基础模块也都是商业库（有点像波音组装飞机，波音公司的零件来自于全球精选的供货商）无论从稳定性，功能性，可靠性都是开源软件无法比拟的。开源软件只适合学术性研究以及小型工具软件的研发，一旦复杂度和规模上来之后，开源软件无能为力。

目前国内大部分仿真软件的研发都基于开源工具，先天性不足，但商业库都不便宜。所以积累技术，拓展国外市场，逐步融入商业软件生态链也是发展国内工业仿真软件一个可行的方法。而只有真正做出质量好，解决实际工程问题的产品，才能让国内的工业软件发展走上良性循环的道路。

仿真软件开发工具介绍（8）--VKI

原创 www.cae-sim.com 多物理场仿真技术

收录于话题

#仿真研发工具

21个



今天再介绍一款重量级的仿真软件商业开发工具VKI，仍然针对的是开发人员，不是最终用户。

公司网址：

www.vki.com

之所以叫重量级，是因为VKI除了几何引擎，几乎囊括了仿真软件开发所有模块，达索，Ansys，MSC，hyperwork，西门子等仿真软件巨头也是其用户。

VKI公司总部在美国，成立于1989年，距今已有30年历史，在仿真软件方面有丰富的技术积累。VKI目前有6个组件模块，按模块销售，主要有图形显示，网格生成，有限单元管理，第三方有限元文件交互，热，结构，流体求解器（没错！求解器也可以买），后处理等。

VKI和其它商业开发工具一样，license按照软件销售比例收费，价格自然不便宜。所以使用类似VKI商业套件，开发本身需要有硬核的产品，才能对冲商业风险。

之前文中提到MeshGem，随口一说称其为“网格一哥”，没想到网上很快就有了引用。从功能上看，如果MeshGem为网格一哥，那VKI就为“网格二弟”吧。😏

快速多级子算法 (FMM) 介绍和实现

www.cae-sim.com 多物理场仿真技术



国内对FMM (Fast Multipole Method) 的介绍都比较复杂, 涉及大量的计算公式。

本文试图用最简单的语言介绍FMM的原理和实现, 并介绍使用C++开发的快速多级算法模块, 用户可自定义核函数, 收敛标准, 截断系数等参数, 可用于实际工程, 后续会介绍FMM与边界元方法 (Boundary Element Method) / 矩量法 (Method of Moment) 结合解决大规模声场, 电磁场问题。

在前面简单介绍了快速多级算法 FMM。快速多级子算法能加快解决非对称满秩矩阵, 扩展了矩量法, 边界元等方法的应用规模, 使其能解决较大规模的实际工程问题。

以下是电磁计算软件FEKO中关于多层快速多级算法 (MultiLevelFMM) 的介绍, 是对FMM的一种改进:

https://www.feko.info/product-detail/numerical_methods/mlfmm

1. 概述

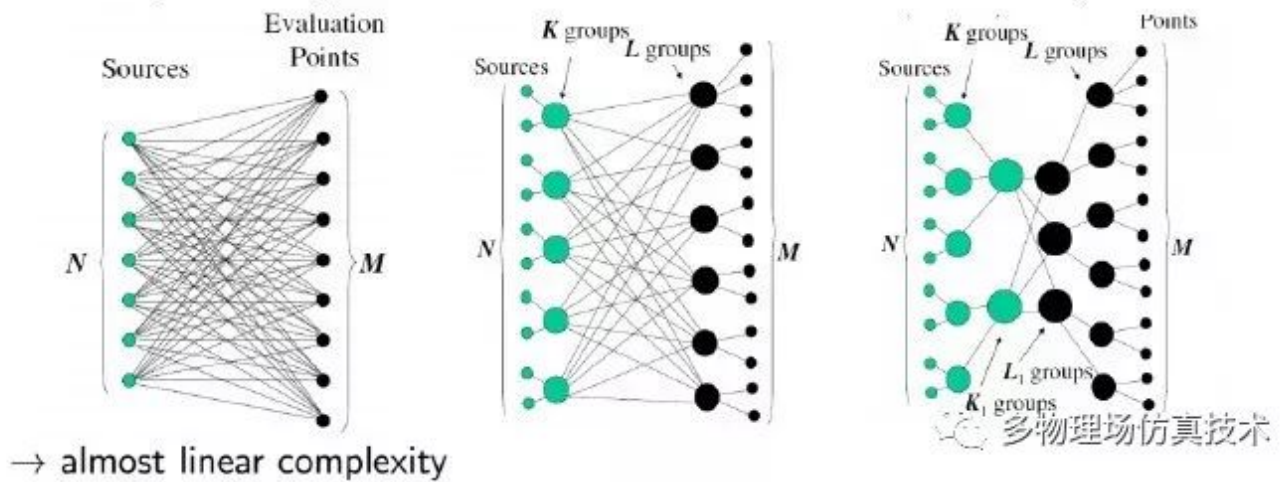
FMM算法的提出来源于多粒子系统相互作用的势场计算, 比如带电粒子或者天体之间引力等。以静电场为例, 空间中N个带点离子构成的系统, 第i个离子所在位置的静电势 $A(x_i)$ 表示为:

$$A(x_i) = \sum_{j=1}^N \frac{m_j}{r_{ij}}$$

其中

1. i不等于j;
 2. x_i 是第i个粒子所在的坐标;
 3. m_j 是第j个粒子所占的权重, 与带电量呈正比;
 4. r_{ij} 是第i个粒子和第j个粒子之间的距离;
- 按照常规计算方法, 对N个粒子实现求和问题, 计算量达到 $O(N^2)$;

在BEM MOM 等数值计算中，一次这样的求和也就是一次矩阵和向量的乘法迭代。

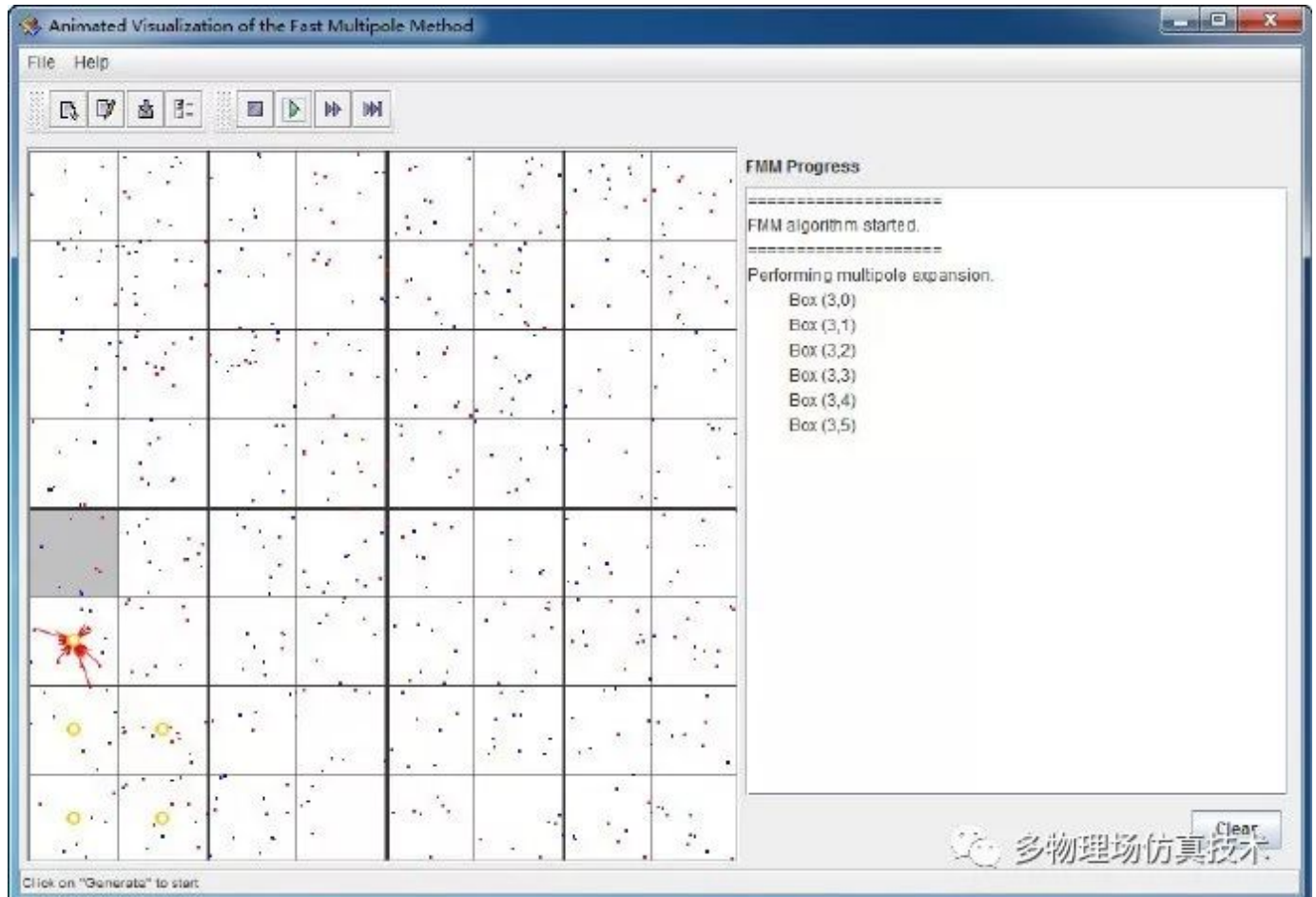


FMM的实现基本思想是以树形结构为基础，通过多级展开和局部展开，把原对象进行分层分组，将 $N \times N$ 的关系转换为少数组对象之间的关系（如上图），从而减少计算量。该算法实现的核心是如何把每个对象归纳到一组对象中，这个主要是通过动态树结构来实现的。

计算过程如下：

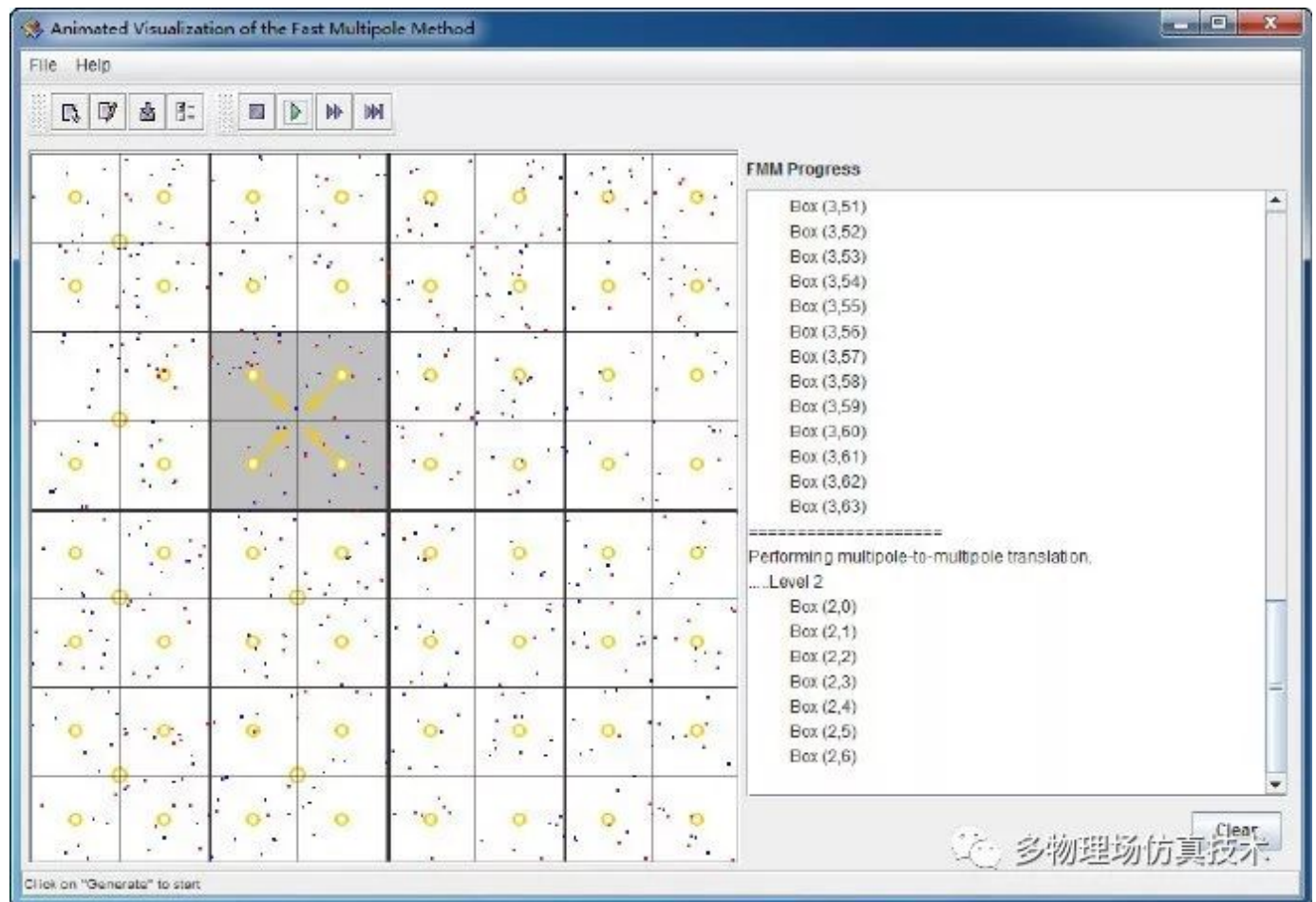
1. 多级展开：

多级展开将叶子节点内（每个小方块内）所有多级展开系数累加，即可以得到该叶子节点的多级展开系数，展开节点为叶子节点的中心。



2. 多级-多级转换 (Level-2)

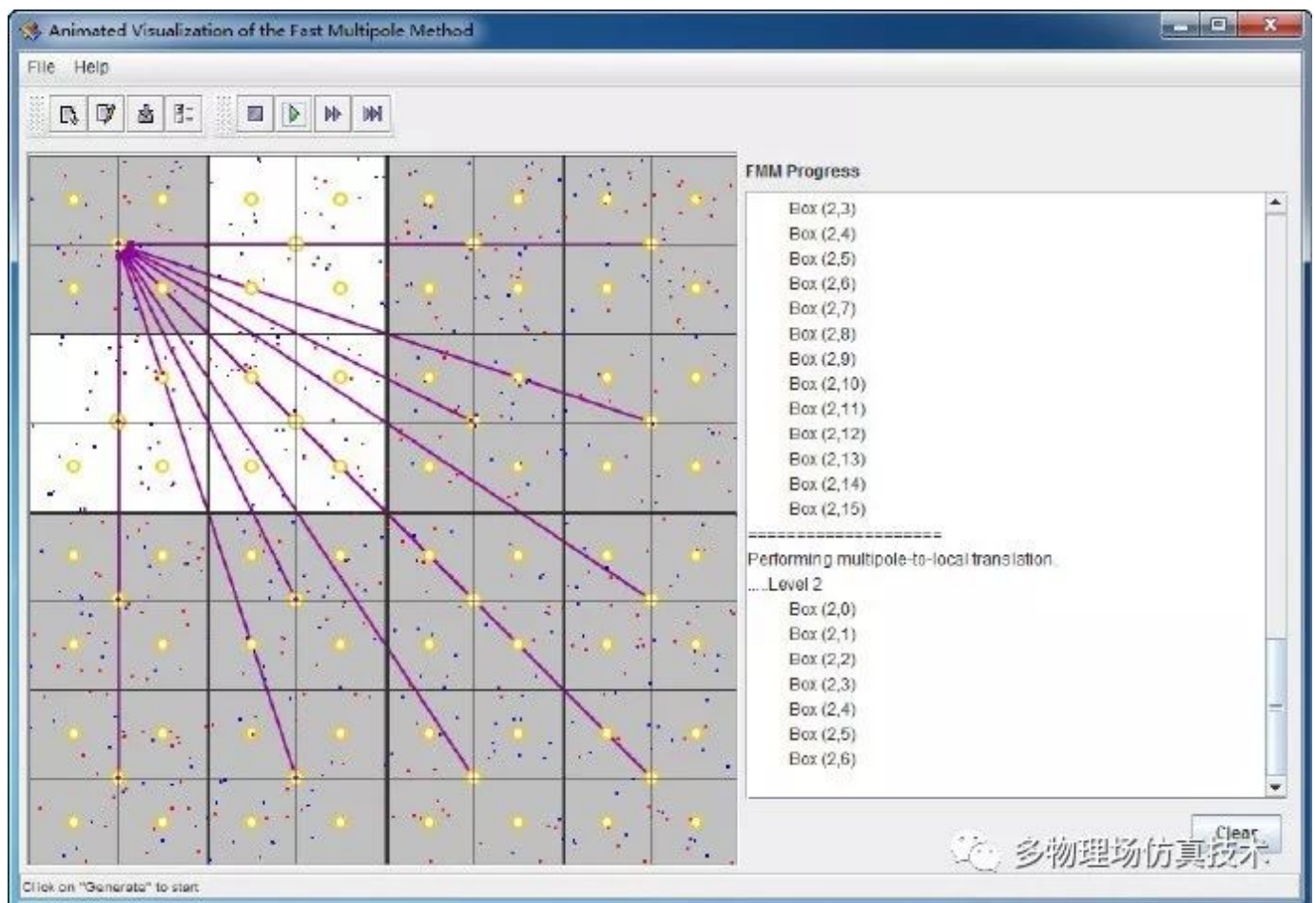
将源点和目标点系数进行转换



3.多级-局部转换 (Level-2)

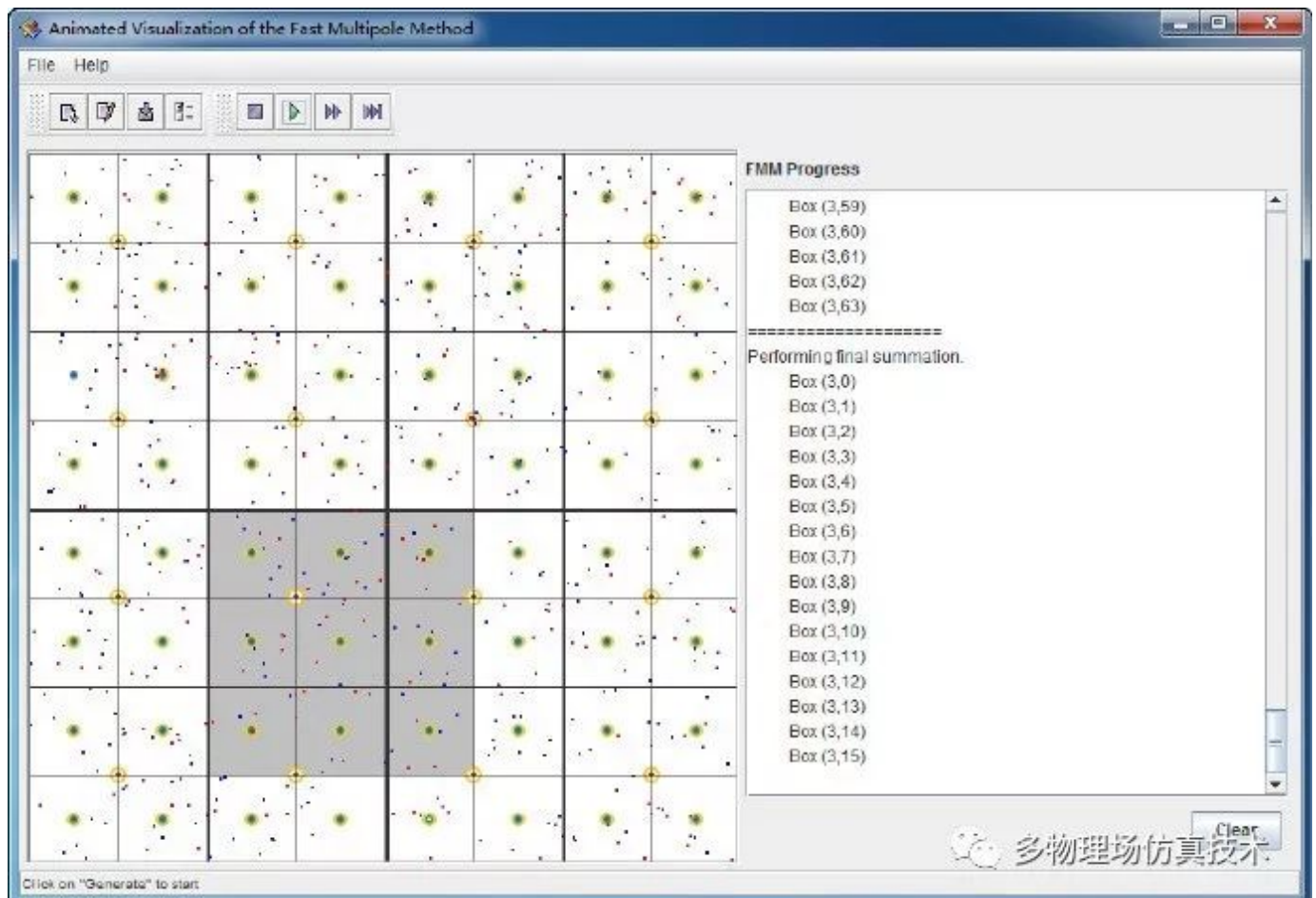
利用多级展开向局部展开系数的传递关系式，将该节点的“相互作用表”（紫色直线连接区域部分即为左上角点的相互作用表）中所有节点的多级展开系数传递，并累加到该节点的局部展开系数，局部展开点是该点的中心

4.局部-局部转换 (Level-2)



5. 多级-局部转换 (Level-3)

6. 最终求和



2. 实现

FMM基础模块，该模块采用C++开发，使用面向对象思想，定义了基础的核函数，并实现了几种简单的核函数，用户可以继承定义自己的核函数，从而应用到流体，电磁，声场等各种需要计算非对称满秩矩阵的数值计算方法中。

基本函数和数据结构：

主要函数：

```
CreateFMMTree      //递归的方法构建节点，生成四叉树或者八叉树结构;  
L2L/Local2Local    //局部到局部转换;  
M2L/Moment2Local   //多级到局部转换;  
M2M/Moment2Moment   //多级到多级转换;  
UpwardPass         //树结构向上遍历;  
DownwardPass       //树结构向下遍历;
```

下图显示了采用Laplace核函数，**一次迭代即 矩阵*向量**，FMM和高斯迭代 所用的时间比较，可以看出FMM所用时间与N几乎呈线性，而且精度很高。考虑到求解线性方程组时需要计算N次，在N达到几万以后，高斯迭代法在PC上基本求不出结果。



更多信息，可以访问“多物理场仿真技术”新浪博客



关于仿真优化工具，之前介绍了很多。这里介绍两个可用于软件研发的工具
PETSc/TAO和Dakota

TAO (Toolkit for Advacned Optimization) 早期是一个独立的软件包，后集成到PETSc中，PETSc是一个可以用于并行求解大规模方程组的科学计算包。大部分数值计算软件，包括商用软件都用到PETSc，是仿真软件求解器的一个基础开发工具。

TAO使用到了PETSc中的数据结构，可用于求解大规模的各种约束，非约束，边值等优化问题。

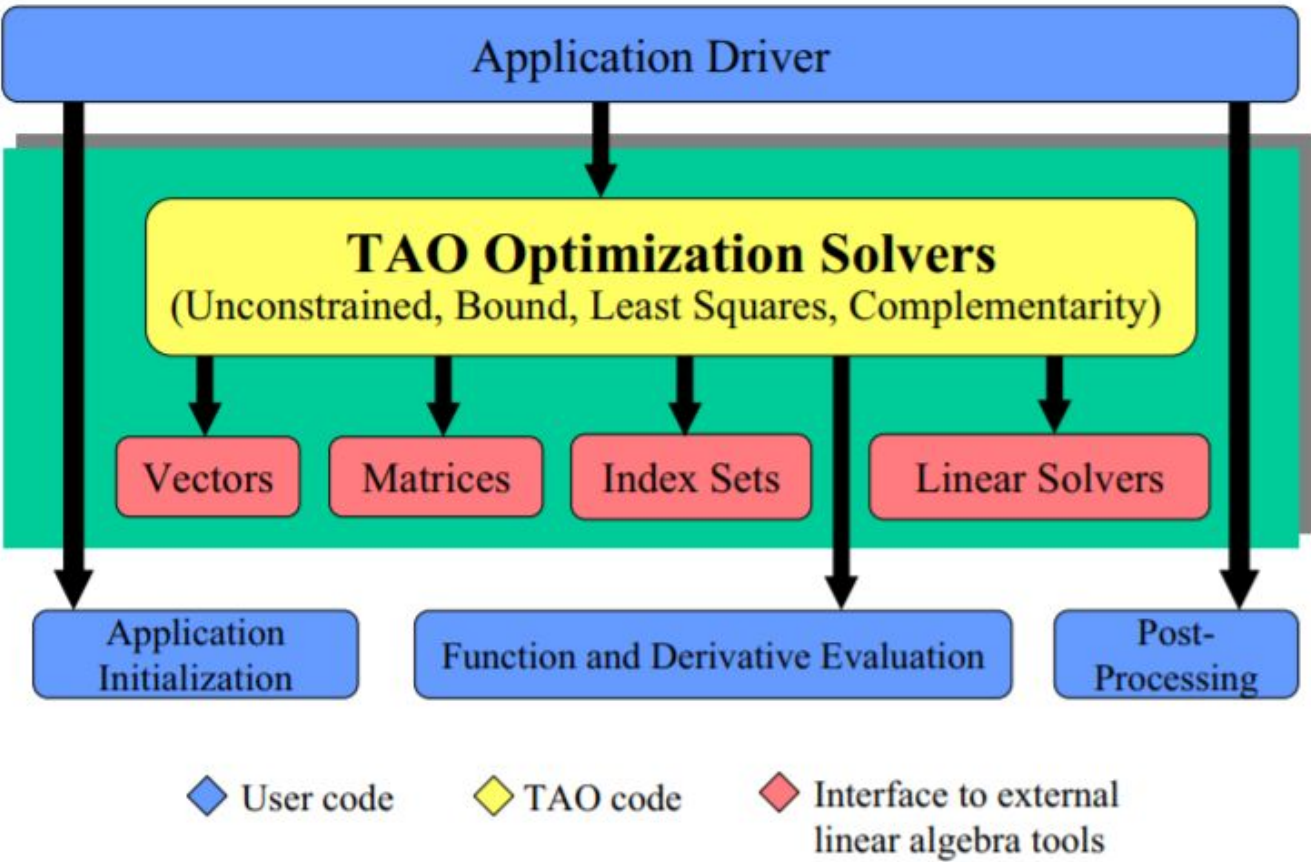


Figure 1.1: TAO Design

网址：

<https://www.mcs.anl.gov/petsc/index.html>

Dakota是另一个开源的多学科优化设计软件，经过多年发展，不仅在底层算法，流程，功能上日趋成熟，也提供了GUI，在实用性上也有了很大改进。

Dakota在算法，易用性，稳定性，文档，用例等各个方面和目前主流商业多学科优化软件iSIGHT，ModelCenter和VisualDOC相比还有一定差距，但由于开源，对于有算法基础的研发人员，可以很容易学习其中算法，并加以改进。对于有经验的工程师，也可以自己定义仿真流程，集成到实际业务逻辑中。

网址：

<https://dakota.sandia.gov/>

一篇文章入门热学有限元

原创 www.cae-sim.com 多物理场仿真技术



本文主要介绍热学有限元相关理论知识，不涉及公式积分，仍然主要面向仿真软件研发测试相关人员，也可供仿真工程师参考。

热学是研究宏观物体的热运动以及和热运动相关的各种规律和性质的科学，是人类最早研究的自然学科之一，也是多物理场仿真中的重要基础。热学有限元的基函数变量为温度，温度是标量，其自由度为1，相比结构，流体，电磁，热学在有限元理论中算是最容易理解和入门的。

1. 三种传热方式:

传导，对流，辐射（2019年发现了第四种传热方式：真空声子传热）

传导：接触良好的物体之间或者物体内部由于温度不同而产生的热交换；

对流：物体和周围介质之间发生的热交换

辐射：物体以电磁波的形式和周围发生的热交换，辐射不需要介质，真空中可以进行传播。

2. 常用物理量

比热容：又称比热容量，简称比热(specific heat)，是单位质量的某种物质，在温度升高时吸收的热量与它的质量和升高的温度乘积之比。比热容是表示物质热性质的物理量。通常用符号 c 表示。比热容与物质的状态和物质的种类有关。

导热系数：导热系数是指在稳定传热条件下，1m厚的材料，两侧表面的温差为1度（K， $^{\circ}\text{C}$ ），在1秒内，通过1平方米面积传递的热量，用 λ 表示，单位为瓦/米·度。导热系数反应了材料导热能力，和材料本身特性相关。在前处理过程中需要给对象赋值材料，材料中需要包含导热系数将分析对象赋值。

热流密度：热流密度定义为导热系数和温度梯度的乘积。

对流系数（换热系数）：对流是固体表面和周围介质之间，由于温度差引起的热交换。在电子元器件散热中是最常用。

斯蒂芬玻尔兹曼常数：用于计算辐射斯蒂芬玻尔兹曼方程的常数，该方程表明：能量传递和物体表面积成正比,与有效温度的四次方温度差成正比。

3. 热荷载和边界条件

热荷载和边界条件和热传导几种方式相对应。

温度：温度通常作为约束，可以加载到点线面上；

对流，热流：通常作为面荷载加载到实体面上，也支持线单元；

热流率：作为集中荷载，可以作用在线单元模型上

热生成率：主要用于对象产生热量的情况，比如电路芯片生热，可以作为三维实体荷载加载到对象上。

热学中的其它荷载都是上述几种基本荷载的组合或者延伸。

4.单元的选择

由于温度只有一个自由度，单一的热学分析对单元选择没有太多要求，但需要注意的是：热和其它物理场进行耦合分析时，可用于热学分析的单元可能并不适用于其它单元分析。比如0阶三维四面体单元，对于热分析没问题，而对于实体结构分析误差会很大。在进行多物理场耦合分析的时候，需要根据实际情况选择对应的单元。

5.稳态和瞬态

稳态是指温度场不随时间变化，是瞬态分析的基础，瞬态是指温度场随时间变化而变化。对于瞬态问题的求解，其核心是选择合适的时间步长。时间步长过小，更容易收敛，结果更精确，但求解时间增加，但同时增加扰动的可能；时间步长过大，求解结果误差偏大。

6. 难点

热学分析的难点和其它CAE/CFD/EDA仿真一样，在于如何建立有效的几何模型，并将几何模型转换成正确的有限元模型，即边界条件荷载等要设置正确！尤其对于复杂几何模型，不仅首先要保证几何正确，可以正确划分网格，还要保证在温度梯度大的地方有足够加密的网格，对于没有标记属性的几何，很难自动化设置参数，人工干预必不可少。

7.热固耦合

当温度发生变化时，由于物体的热胀冷缩特性，在物体内部的受力约束会在物体内部产生应力。这种应力称为热应力。热固耦合非常普遍，比如铁轨路桥伸缩缝，发动机热应力，电路板受热应力分析等等。

热固耦合和其它多物理场耦合一样，分为两种：直接耦合和间接耦合，直接耦合是用包含结构和热自由度的同一单元，一次性求解PDE得到结果。商业软件中出现的大多数大多提供了耦合单元；间接耦合则是先计算出温度场，然后将温度场作为荷载加载到结构分析中。两种分析方法各有利弊：间接法简单，求解所需资源少；直接法适合高度非线性耦合。

8.热流耦合

热流（CFD）耦合分析最常用的应用场景是散热，比如高速飞行器表面散热，电子元器件工作散热等。热流耦合一般采用间接耦合分析方法，首先考虑的是温度场，即固定温度边界条件，然后作为输入进行CFD计算。

关于热学的单元介绍，网格分析，偏微分方程PDE可以参考如下文章：
一篇文章入门多物理场有限元

说说 Generative Design

原创 www.cae-sim.com 多物理场仿真技术

收录于话题

#仿真几何相关

5个



Generative Design（生成式设计）最近两年出现频率多起来，但Generative Design并不是一个新的术语。

Generative Design 其核心就是在设计阶段考虑到所有要素，并合理优化各个要素比重，约束，目标等，使设计最大可能自动化。从算法角度看，Generative Design 本质上是一个自动优化设计问题。

以公司装修为例，设计需要根据各种约束（是否需要修改，办公室数目，个人喜好，同一团队安排在一起）等等考虑自动化给出一个最优的设计方案，保证在预算范围内大家都满意。这也让我联想到大学时参加的国际数学建模竞赛的题目：伽马刀设计。转成数学问题就是给出任意一个三维肿瘤形状，然后给出几个固定形状的高温靶点，设计一个方案使肿瘤最大程度将靶点包围（高温靶点可以烧掉肿瘤）。实际操作中医生通常根据经验手工给出方案，不仅依赖经验，治疗耗时，而且有出错的概率（烧到正常组织，或者肿瘤切除不完全）。

另外一个很好说明Generative Design的实际应用案例是拓扑优化设计，经过初始设计，迭代仿真，最后给出轻量化最优设计结果，设计结果能否制造是另外一回事。当然Generative Design 不等于拓扑优化，拓扑优化只是Generative Design的一种体现方式。Generative Design 涉及的范围和理念很广，它可以使用任何方法包括人工智能，机器学习等，不只是结构，其它流体，电磁，热，都可以是其整体优化设计的因素。

PTC公司刚刚发布的Cero7.0声称加入了 Generative Design，其内容是集成了ANSYS的仿真工具，不仅能进行应力仿真，还能进行热分析，流体分析，模态等分析，使用户在设计早期进行仿真，优化设计。在传统仿真工具中，我们可以发现仿真工具几乎没有设计

能力，模型主要靠外部导入，仿真工具即使提供的设计模块也很鸡肋，几乎排不上用场。所以单纯的仿真不能解决设计问题，或者说仿真不是设计需要考虑的唯一因素。

从软件层面看，由于各个行业入门和仿真本身的高门槛，使得这种集成未来主要依靠公司层面的合作。比如PTC和ANSYS就是最典型的合作。而在细分领域比如EDA行业，兼有设计和仿真的公司则更是形成了高度的垄断。

设计理念通常会领先于实际应用。从系统工程角度看，Generative Design的实际应用落地更多依赖于软硬件的发展以及CAD/CAE/CAM产品级的融合，也还有很多需要完善的地方，但是正如《仿真软件十年回顾和展望》一文中提到CAD/CAE无缝集成将是未来发展势所趋，也是设计仿真公司发展的方向。

说说digital twin

www.cae-sim.com 多物理场仿真技术



公众号里有朋友要我谈谈digital twins 数字双胞胎，在和朋友聊天中也提到过这个概念。

其实在技术领域，中国最不缺乏的就是技术概念！国内的销售公司，引进新的技术概念是日常工作，因为在销售过程中能带上一些新名词对谁都是好看的，项目中挂羊头卖狗肉是常有的事。对于技术人员来说，技术概念落地才有意义，通常新的技术概念涉及到产业链或者生态链，不是简单靠技术能实现的，所以我本人对很多技术概念并无太多兴趣，前面提到的Generative design也是一样。

数字化双胞胎”（Digital Twin）是指以数字化方式拷贝一个物理对象，模拟对象在现实环境中的行为，对产品、制造过程乃至整个工厂进行虚拟仿真，从而提高制造企业产品研发、制造的生产效率。

简单讲就是在实物和模型数据之间建立起关联，从设计开始，到实物制造，工作状态都能有虚拟模型对应。其实在很多行业尤其是精密制造业比如航天航空，集成电路，安防等早就实现了这种功能。最近几年建筑行业流行的BIM其实也是Digital twin的一种表现。因为建筑行业一直是“傻大粗”，数字化程度非常低，所以才有了BIM的火热。随着信息技术的发展，类似于AI技术，物联网，智能设计制造等等都会使数字双胞胎更容易落地。

所以数字双胞胎并不是新东西，概念早就有，只是技术发展的必然产物。你也可以叫它“虚拟制造”，“数字三胞胎”，“数字伴侣”，其本质上都是一样的。

FEM之在求解器中使用设计模式(1)--Bridge模式

原创 www.cae-sim.com 多物理场仿真技术

收录于话题

#设计模式 8 #求解器开发 22



首先很佩服工程出身的人能编程写代码，事实上，很多优秀的求解器不是计算机出身的人开发的。之所以想写《[在求解器中使用设计模式](#)》系列有关技术细节的文章，是因为自己在参考很多开源求解器代码时，发现很少有开发者能基于软件工程，从代码的维护性，扩展性，稳定性，移植性，兼容性方面来考虑设计。

且不说用100数组定义一个单元的未知自由度有多么耗内存，也不说如何阅读有100多个参数的函数，更不用说如何在如下代码添加新单元：判断单元类型用switch语法加上20多个case，每个case的代码有一百多行。单是看着一个简单的用线性三角单元求解拉普拉斯方程分别用Matlab, C, Fortran, VB, Java写上一遍就觉得很无语：在一件简单事情上做了太多重复性的工作。

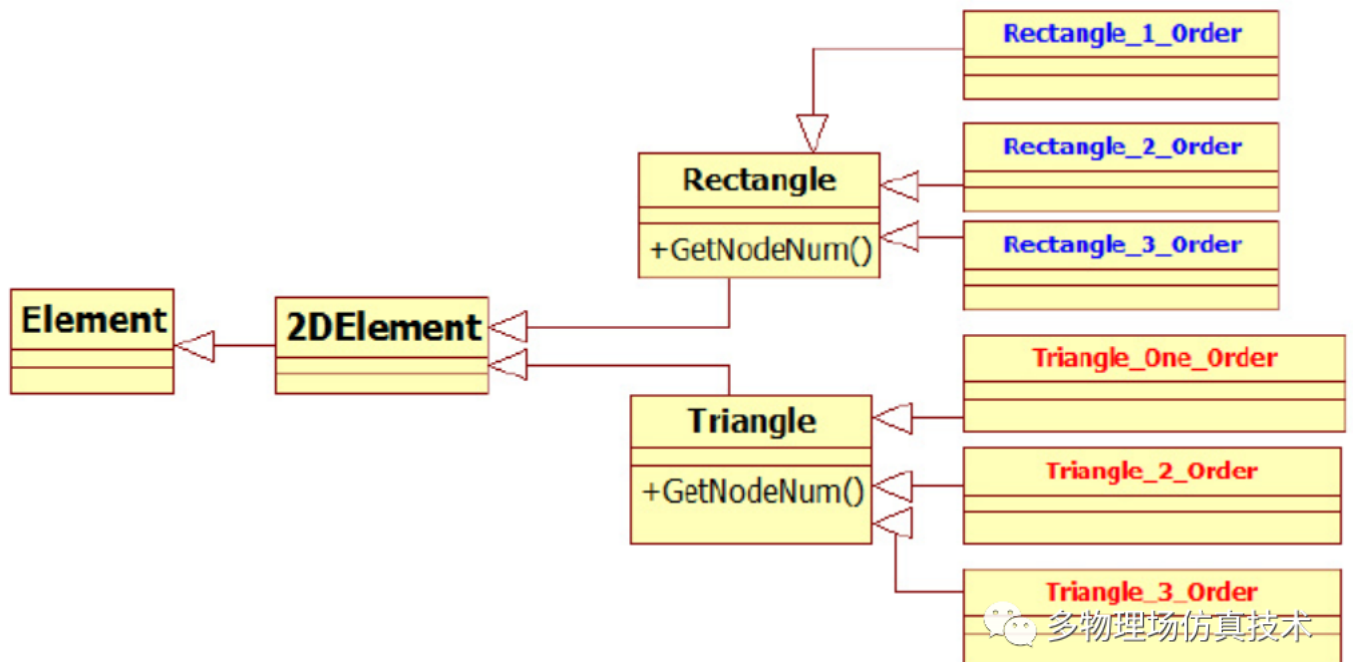
设计模式 (Design pattern) 是一套被反复使用、多数人知晓的、经过分类编目的、代码设计经验的总结。使用设计模式是为了可重用代码、让代码更容易被他人理解、保证代码可靠性。毫无疑问，设计模式于己于他人于系统都是多赢的；设计模式使代码编制真正工程化；设计模式是软件工程的基石脉络，如同大厦的结构一样。

首先介绍Bridge模式（桥接模式）：

考虑如下问题：

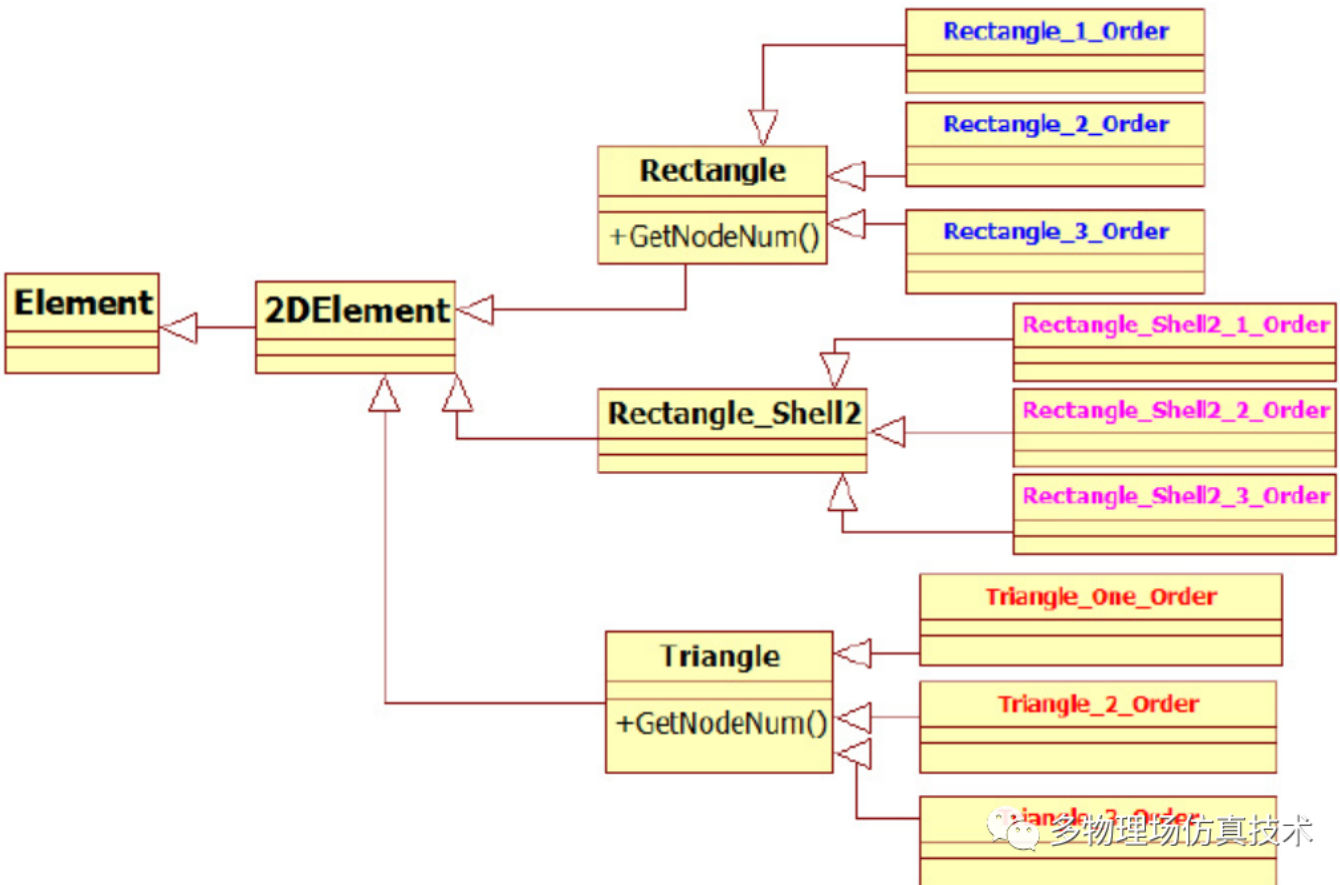
有限元单元按照阶次可以分为线性，二次，高次单元。以2D三角单元为例，线性单元（3个点），二次单元（6个点，每边中点有一个点）

类图通常设计如下图1：



多物理场仿真技术

GetNodeNum()用来获得单元的节点数目：
 以四边形Shell单元为例：
 Rectangle_1_Order()=4; 1阶 4个点
 Rectangle_2_Order()=8; 2阶8个点
 Rectangle_3_Order()=12; 3阶12个点
 现在需要添加一种新的四边形单元类型Rectangle_Shell2。
 图1变为如下图2：



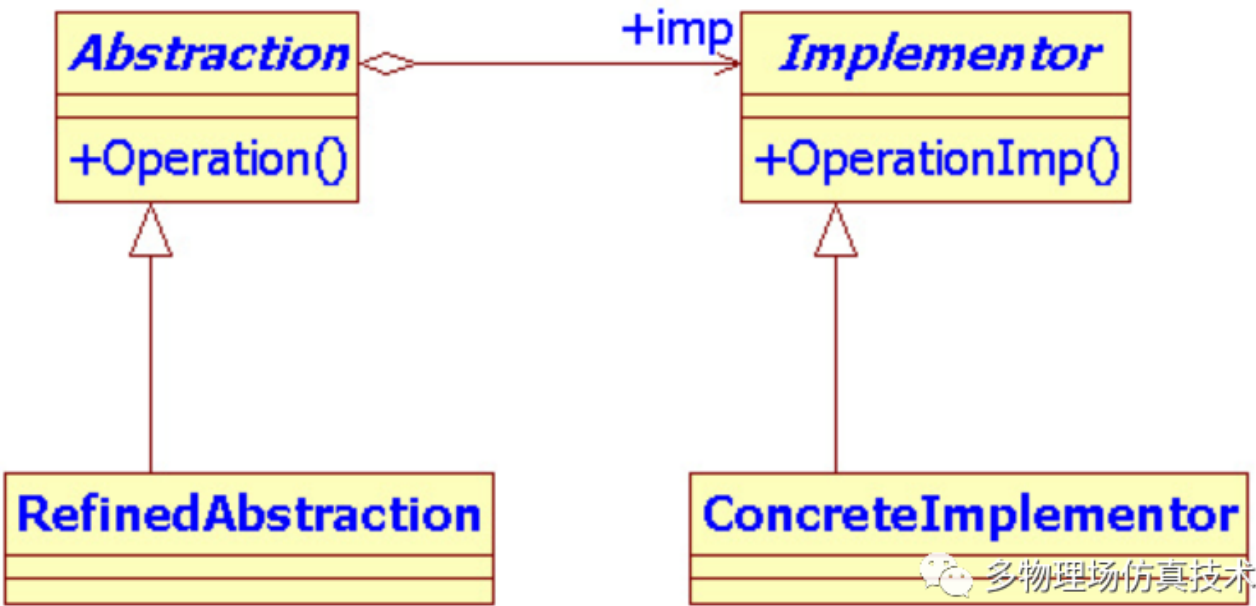
多物理场仿真技术

可以看出增加一种单元类型，要增加4个类，还不考虑更高次的单元（已有软件最高可以达到8次，Ansys中仅Plane/Shell单元就有40多种，如果全部添加则需要8*40=320个类，而且还只考虑了长方形这一种单元，显然不合理）

针对这个问题有两种解决办法：

- 1. 在每个单元内部根据阶次计算节点个数。这种办法肯定可行，但是增加了单元的复杂性，也造成了代码的重复。
- 2. 利用Bridge模式，将单元节点计算和单元抽象分离。

典型的Bridge图如下：



【GOF95】在提出桥梁模式的时候指出，桥梁模式的用意是"将抽象化(Abstraction)与实现化(Implementation)脱耦，使得二者可以独立地变化"。这句话有三个关键词，也就是抽象化、实现化和脱耦。

1.抽象化：

存在于多个实体中的共同的概念性联系，就是抽象化。作为一个过程，抽象化就是忽略一些信息，从而把不同的实体当做同样的实体对待。

2.实现化：

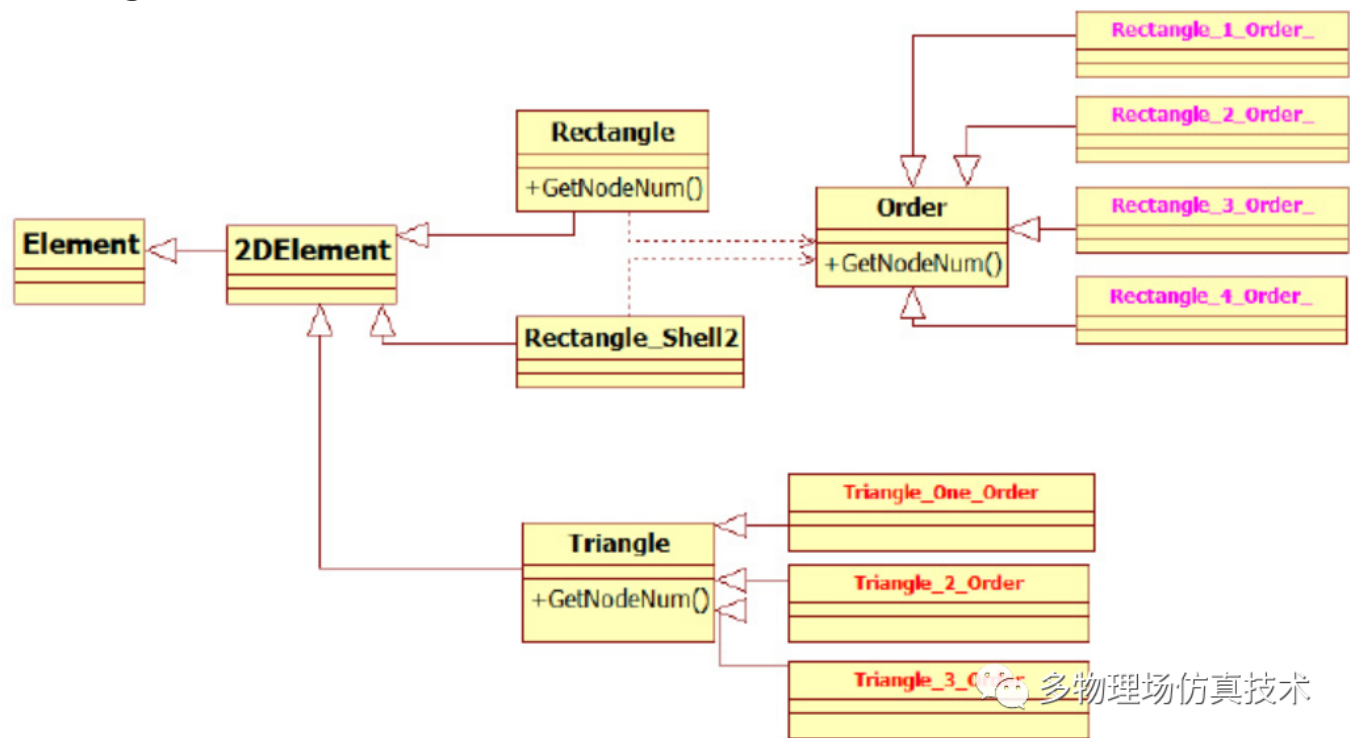
抽象化给出的具体实现，就是实现化。

3.脱耦：

所谓耦合，就是两个实体的行为的某种强关联。而将它们的强关联去掉，就是耦合的解脱，或称脱耦。在这里，脱耦是指将抽象化和实现化之间的耦合解脱开，或者说是将它们之间的强关联改换成弱关联。

将两个角色之间的继承关系改为聚合关系，就是将它们之间的强关联改换成为弱关联。因此，桥梁模式中的所谓脱耦，就是指在一个软件系统的抽象化和实现化之间使用组合/聚合关系而不是继承关系，从而使两者可以相对独立地变化。这就是桥梁模式的用意。

采用Bridge模式后，图2变成如下图3：



通过Bridge模式 实现了有限元单元和阶次的解耦，二者可以独立的变化，基于Bridge模式，可以在有限元单元中应用更多的变量，比如维度：0维到3维；高斯积分精度，从0到高阶；针对多物理场的单元：电磁，结构，热，声，流体，耦合单元等等，使得有限元单元设计更加灵活，通用性，扩展性更强。

现在网上关于设计模式资源很多，更多Bridge的具体实现可以百度

后续将会陆续介绍Composite，Builder以及Factory 等几种常用设计模式在求解器开发中的应用。

FEM之在求解器中使用设计模式(2)---Abstract Factory模式和Builder模式

原创 www.cae-sim.com 多物理场仿真技术

收录于话题

#设计模式 8 #求解器开发 22



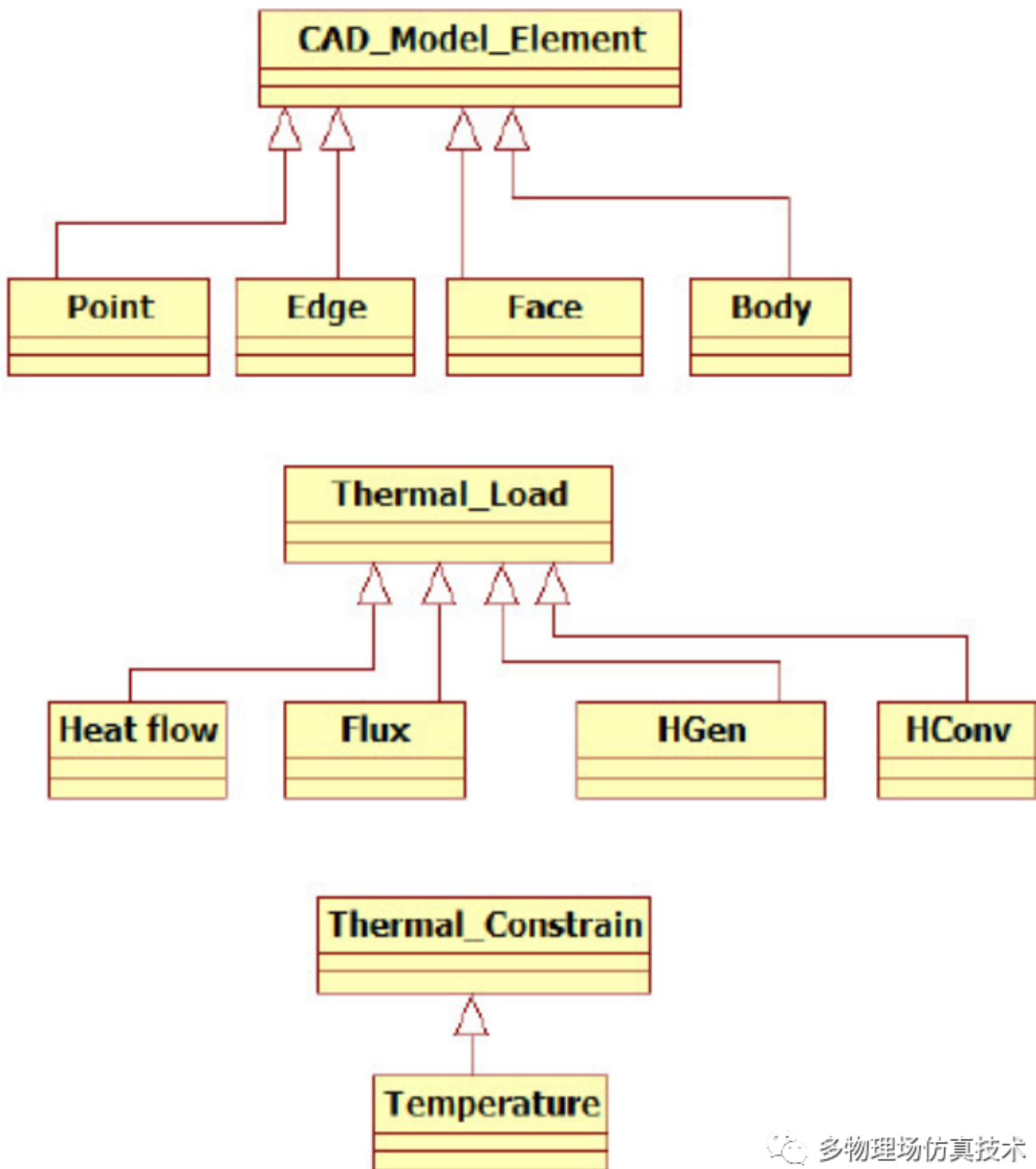
Abstract Factory模式和Builder模式是使用最广泛的两种创建型设计模式。之所以把两者放到一起，是因为两者有共同的地方，利用这两种模式可以优化对象创建的流程。

本文利用设计模式解决两个创建流程问题：

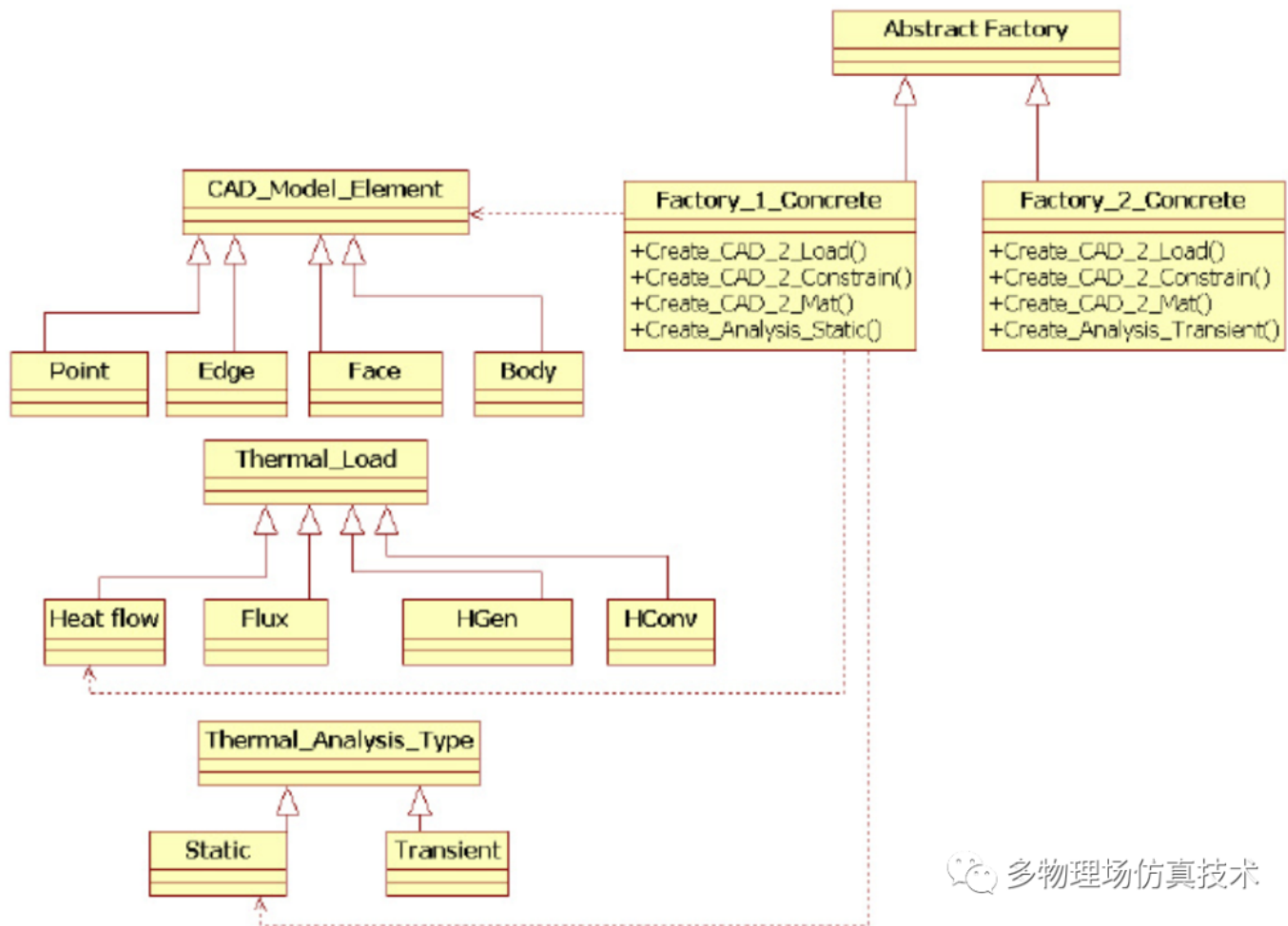
1. 单物理场网格生成前有限元模型的创建
2. 多物理场网格生成后有限元模型的创建

Abstract Factory模式的意图是 是为创建一组（有多类）相关或依赖的对象提供创建接口。

以稳态温度场为例,假设荷载，约束可以加载到任意几何上，即可以加载点，线，面，体上。那么数据结构可以定义如下：



创建过程使用Abstract Factory模式，如下：



多物理场仿真技术

使用Abstract Factory(抽象工厂)模式有如下好处:

- 1.封装了产品的创建,使得不需要知道具体是哪种产品,只需要知道是哪个工厂就行了。
- 2.可以支持不同类型的产品,使得模式灵活性更强。
- 3.可以非常方便的使用一组中间的不同类型的产品。

在上例中,用户可以根据分析类型,材料类型,几何映射边界约束和荷载构建不同的创建流程。

2. 多物理场网格生成后有限元模型的创建,以稳态温度场为例

任意一个单物理场有限元模型创建流程如下:

```

void Create_FEM_Model()
{
    //创建单元数据,网格已经生成成
    createElement_x_2_Material();

    //创建边界约束.
    BuildGeometry2Load();

    //创建荷载.
    BuildGeometry2Constrain();
}
  
```

```

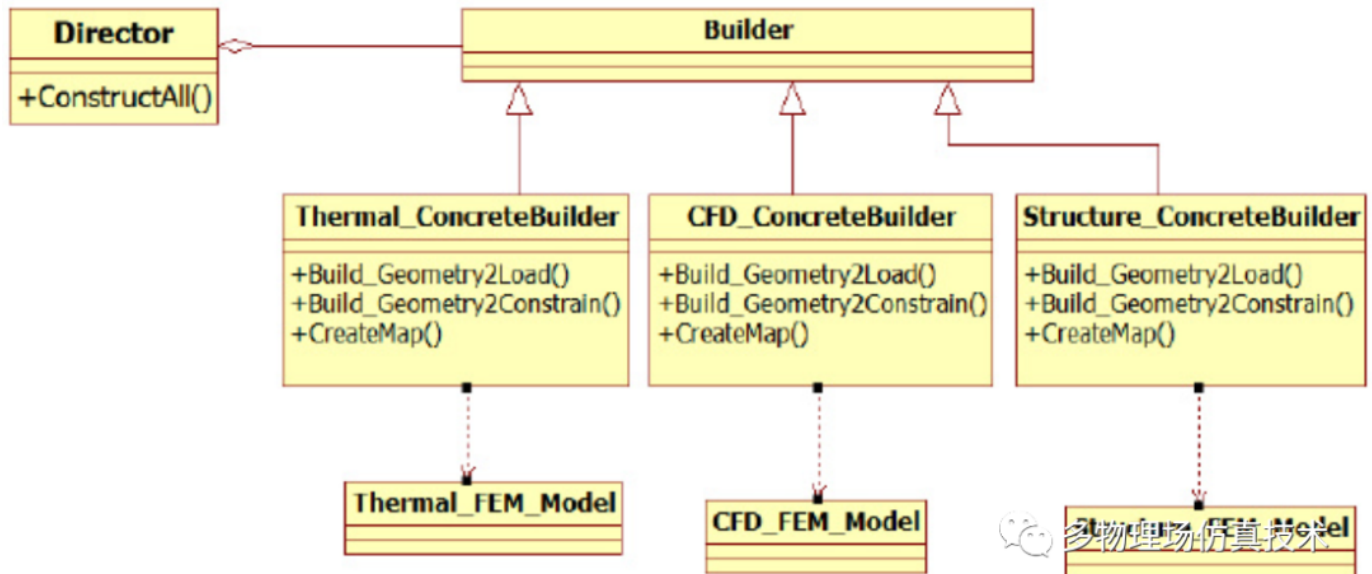
//设置分析类型.
SetAnalysisType();
}

```

理论上附属在几何上的属性在划分网格后都映射到结点(Vertex)和单元(Element)上。对于多物理场的创建也一样。

Builder模式的意图是 将一个复杂对象的构建与其表示相分离，使得同样的构建过程可以创建不同的表示。

利用Builder模式创建多物理场：



使用Builder模式的优点：

- 1、建造者模式的“加工工艺”是暴露的，这样使得建造者模式更加灵活。
- 2、解耦了组装过程和创建具体部件，使得我们不用去关心每个部件是如何组装的。

上例中对于多物理场，每种物理场的边界荷载都是不一样的，通过Builder模式可以不用关心单个物理场的实现，将创建过程和具体的创建解耦。增加了结构的灵活性和可扩展性。

FEM之在求解器中使用设计模式(3)---Composite模式

原创 www.cae-sim.com 多物理场仿真技术

收录于话题

#设计模式 8 #求解器开发 22



在求解有限元模型中会碰到这种情况：

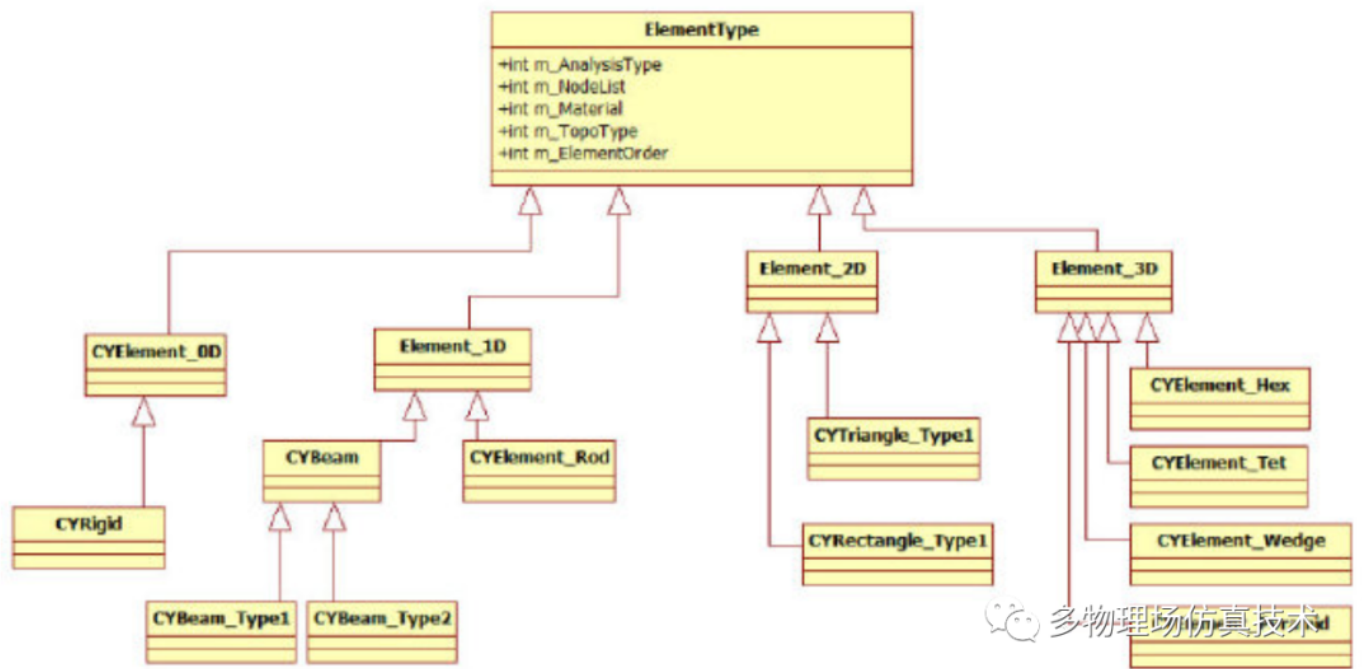
1. 一个模型有多种单元类型（比如同时存在2维单元和3维单元，2维单元又同时存在三角单元和四边形，3维单元同时存在四面体和六面体）
2. 仿真流程有多种工况，每种工况又有多个步骤，每个步骤对应的有限元模型不同
3. 一个模型有多个相同或者类似的子结构，而且子结构存在嵌套情况。

对于第一种情况，一般的做法就是将所有单元循环，判断单元类型，根据单元类型做相应处理。如果单元类型少该方法有可行性，但是当单元类型不确定，而且数量过多，使用该方法代码不仅结构累赘，难以扩展而且层次不清，封装性差。

利用Composite模式可以很好解决这个问题：

将对象组合成树形结构以表示“部分-整体”的层次结构。Composite使得用户对单个对象和组合对象的使用具有一致性。（Composite使得用户对单个单元和组合单元的使用具有一致性）。

单元结构图：



假设FE模型中同时存在4种单元：梁，杆，四面体Shell，MPC单元
创建过程如下：

按照如上建立结构后，代码实现了如下的树形结构：

所有单元

- ▣ 1D单元
 - ▣ 梁单元
 - 梁单元1
 - 梁单元2
 - ...
 - 梁单元N
 - ▣ 杆单元
 - 杆单元1--M
- ▣ 2D单元
 - ▣ 四边形Shell
 - Shell单元1--K
- ▣ 连接单元
 - ▣ MPC连接单元
 - MPC1 ---L

多物理场仿真技术

使用Composite模式后，对于调用者来说可以很清楚的看到单元类型的层次结构，由于所有单元类型接口相同，建立好此结构后，调用者可以很方便的添加，删除，查询，组合，管理任意多种类型的单元（比如添加新的类型单元，调用者只需准确定义好单元类型，实现新单元类型的接口，而不用理会有限元模型object的操作）

针对第三个问题，以集束炸弹FE模型建模为例：



多物理场仿真技术

假设一个集束炸弹有三个炸弹，每个炸弹有若干子炸弹，每个子炸弹又有多个子炸弹。对于使用者来说，即可以把三个炸弹看做一个炸弹，也可以把任意一个炸弹的子多个子炸弹看做一个炸弹组合。利用Composite在建模的时候解决了子炸弹之间的递归组合问题。

(1) Composite (组合) 模式的目的

组合模式的目的是：让客户端不再区分操作的是组合对象还是叶子对象，而是以一个统一的方式来操作。

实现这个目标的关键之处，是设计一个抽象的组件类，让它可以代表组合对象和叶子对象。这样一来，客户端就不用区分到底是组合对象还是叶子对象了，只需要全部当成组件对象进行统一的操作就可以了。

(2) 对象树

通常，组合模式会组合出树形结构来，组成这个树形结构所使用的多个组件对象，就自然的形成了对象树。

这也意味着凡是可以使用对象树来描述或操作的功能，都可以考虑使用组合模式，比如读取XML文件，或是对语句进行语法解析等。

FEM之在求解器中使用设计模式(4)---Builder模式

原创 www.cae-sim.com 多物理场仿真技术

收录于话题

#设计模式 8 #求解器开发 22



Builder(建造者模式)将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。

Builder模式和AbstractFactory模式在功能上很相似，因为都是用来创建大的复杂的对象，它们的区别是：Builder模式强调的是一步步创建对象，并通过相同的创建过程可以获得不同的结果对象，一般来说Builder模式中对象不是直接返回的。而在AbstractFactory模式中对象是直接返回的，AbstractFactory模式强调的是为创建多个相互依赖的对象提供一个同一的接口。

适用于以下情况：

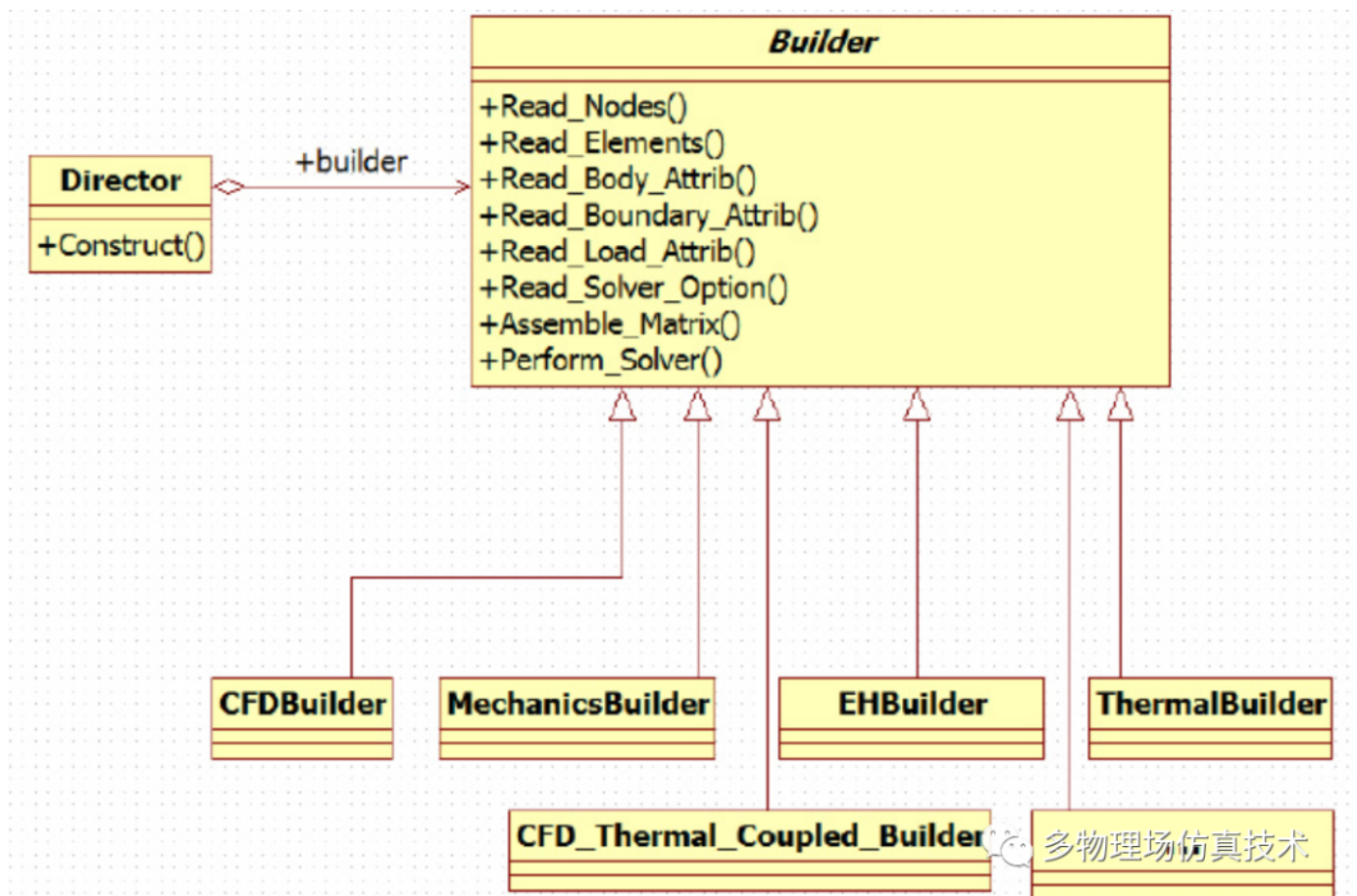
- 1)当创建复杂对象的算法应该独立于该对象的组成部分以及它们的装配方式时。
- 2)当构造过程必须允许被构造的对象有不同的表示时。

求解器的实现过程其实都大同小异，完全可以用相同的过程来组织，而对应的不同的求解器的具体实现可以封装起来。

在求解器中的步骤一般为：

1. 读入节点信息；
2. 读入网格单元信息；
3. 读入体属性（材料）；
4. 读入荷载属性；
5. 读入边界属性；
6. 读入求解器属性；
7. 组装求解矩阵和向量；
8. 求解方程组；

不同的求解类型求解方法可能不同，但是整体流程可以按照上述顺序固定下来。如果有不同的内容，可以放到其中任意一个步骤中。对于多物理场的处理，对应的UML图如下：



这样做的好处是：

1. 接口统一，在求解器调用端不必知道每个求解器内部的求解细节；
2. 每个求解器相对独立，可以独立变化；

Builder有点类似于Abstract Factory, 但是Builder强调的是顺序过程相同，即表现形式相同，而内部可以不同。PDE的组成形式千变万化，一个小的参数也可能导致整个方程性质的变化。但是万变不离其宗，使用FEM求解PDE的流程都是一样的。

从代码设计角度看，Builder模式是一个非常合适组织多物理场求解器的软件设计方法。

一篇文章入门边界元方法

原创 www.cae-sim.com 多物理场仿真技术



老规矩，仍然针对软件研发测试工程人员，不会有公式积分，科普性文章。

在数值仿真领域，有限元方法一直是首选，但在一些细分领域，其他方法可能更有效，比如有限体积法（FVM）对流体，边界元法（Boundary Element Method）对声，电磁散射。

在2005年关于SCI收录的数值方法统计中，有限元相关文献有66237篇，有限差分19531篇，边界元10126篇。

以声场为例，声音向外传播，作用区间是一个典型的无限范围，如果使用FEM，需要对整个区域划分网格，在三维空间随着计算范围的增大，四面体或者六面体网格数目会急剧增大。而采用BEM，只需要求解声场边界上的数值，大大降低了计算量，提高了计算效率。

边界元方法的核心：**只以边界上的变量为基本变量，求解边界变量，场内的未知量可以用公式根据边界变量求出。**

BEM 基于解析解，在处理某些特定领域（声场，电磁场，连续介质弹性力等无限域问题）具有精度高，降低维度等特点，同时通过与FEM相结合，能够综合两者的优势，提高计算效率和精度。

前面介绍过声场的控制方程为[赫姆霍兹方程](#)，边界元方法的第一步是要将PDE转换为边界积分方程，对于三维空间的方程，需要使用格林函数将体积分转换为面积分。

通常流程是：

1. 利用算子的基本解作为权函数，按加权余量格式得到区域上的积分方程；
2. 利用[高斯公式（格林函数）](#)建立区域内积分和边界积分的关系，从而得到区域内任意一点的通解变量表示的积分表达式；（利用格林函数可以将三维体积分转为二维面积分）
3. 将基本解的奇异点 P 趋于边界点 p ，得到边界积分方程。

这个过程简称为用加权余量法建立边界积分方程。

BEM求解流程与FEM基本类似，划分网格，建立线性方程组，求解方程，不同的只是建立的本构方程不同。因为BEM求解区域只在边界上，所以三维问题只需要划分面网格，二维问题划分线网格，使前处理工作大为简化。

既然BEM效率这么高，为什么使用范围不如FEM广泛呢？主要两点原因：

1. FEM 是一种纯数值解法，而BEM是半数值解，也就是说要以解析基本解为前提条件，有些情形比如非均匀介质，各向无规律异性等问题无法处理。

2. BEM方法最后形成的线性方程组为非对称满秩矩阵，限制了工程上的求解规模。桌面单机几千自由度就已无法解出结果，虽然通过FMM（快速多极子方法）解决了这个问题，但因为FMM算法本身实现也有一定难度，因而没有FEM应用广泛。笔者曾经计算显示当模型自由度在15000以下时，传统的BEM效率更高，而大于15000时，传统BEM计算量会急剧上升，此时FMM的计算量仍然呈线性增长。

目前，边界元在电磁和声场领域应用较多，在连续介质结构也有较多研究，商业软件有BEASY, IntegratedSoft, FastBEM等。

快速多级方法参考公众号之前文章：

快速多级算法简介

快速多级子算法（FMM）介绍和实现

FEM之在求解器中使用设计模式(5)---Adaptor模式

原创 www.cae-sim.com 多物理场仿真技术

收录于话题
#设计模式 8 #求解器开发 22

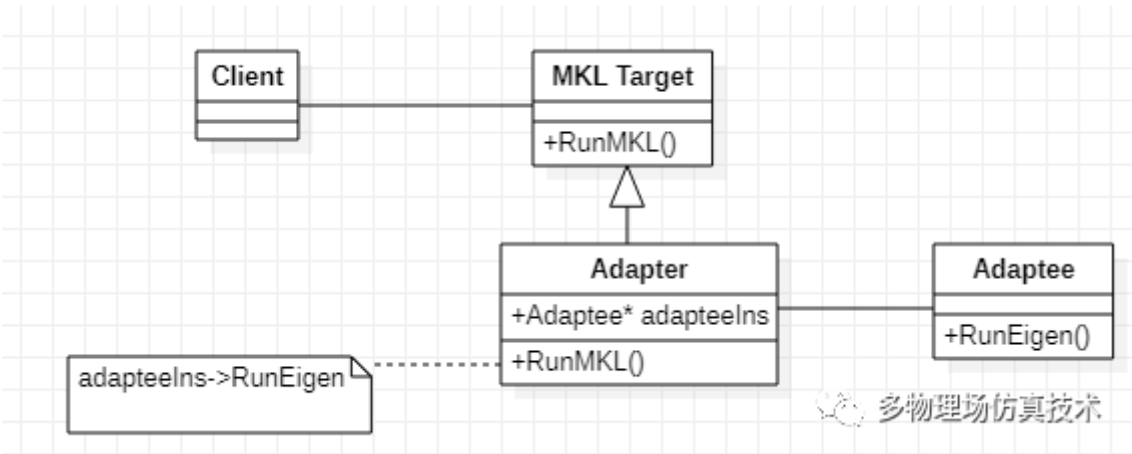


在开发求解器过程中，经常会碰到一种情况：需要换一种线性方程组求解工具，比如最开始使用入门上手的Eigen，矩阵规模上来之后需要用PETsc或者MKL。如果一开始没有考虑到这种需求，很容直接使用Eigen的数据结构，将原始的数据结构和Eigen耦合在一起；此外可能一开始不需要全部转成PETsc格式，仍然需要Eigen数据结构和某些功能；还有可能已经有第三方封装好的MKL或者PETSc或者求解器不仅自己需要，还要满足给第三方的接口。

当然这些需求可以通过面向过程的编码方式实现，但从代码的可重用以及代码维护，扩展角度看，用面向对象的方式更灵活，更易于封装和模块化。

Adaptor 适配器也是一种常见的设计模式，主要作用把一个类的接口变换成客户端所期待的另一种接口，从而使原本接口不匹配而无法一起工作的两个类能够在一起工作。

关于Adaptor最通俗的解释就是电源接口的转换，比如我们国家的电压是220伏，国外是110伏，所以国内的电子产品到国外使用就需要用转换器，将110伏转成220伏，这个转换器就可以认为是Adaptor的具体表现。



如上图所示，用户原来调用的是MKL的接口，通过adpater接口，可以调用Adaptee的Eigen接口。这样既不用改动原来Eigen接口，也无需改动MKL接口。通过adapter实现数据转化和调用，最大程度实现了解耦功能。

FEM之在求解器中使用设计模式(6)---Strategy模式

原创 www.cae-sim.com 多物理场仿真技术

收录于话题

#设计模式 8 #求解器开发 22



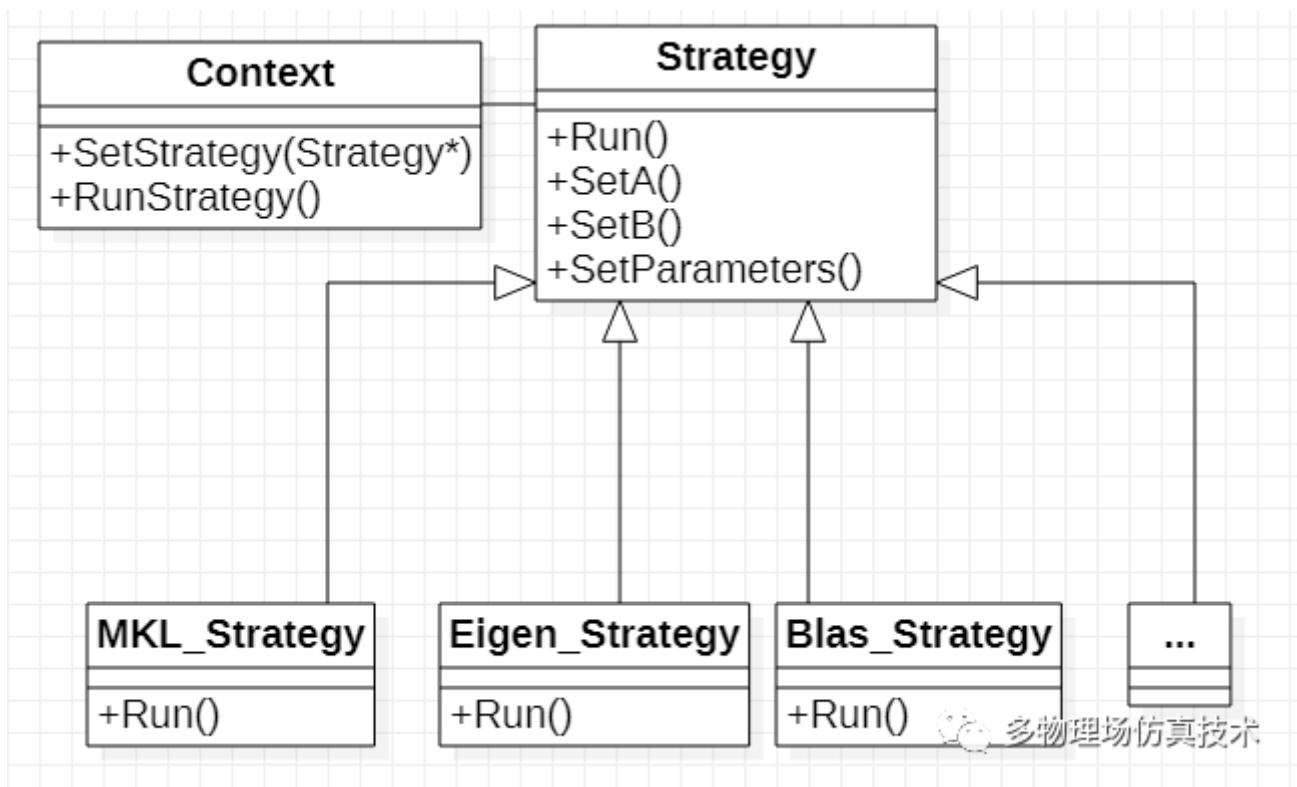
完成一项任务，往往可以有多种不同的方式，每一种方式称为一个策略，我们可以根据环境或者条件的不同选择不同的策略来完成该项任务。最常用的实现方式就是使用if else语句，但是这种方式将所有使用模块放在了同级水平，造成修改和添加功能时候很容易破坏原有结构。

策略模式(Strategy Pattern)：定义一系列算法，将每一个算法封装起来，并让它们可以相互替换。策略模式让算法独立于使用它的客户而变化，也称为政策模式(Policy)。

策略模式的优点：

- 1.由于策略类都实现同一个接口，所以使它们之间可以自由切换。
- 2.增加一个新的策略只需要添加一个具体的策略类即可，基本不需要改变原有的代码。
- 3.避免使用多重条件选择语句。

在求解器开发中，仍然以调用求解线性方程组库为例，前面介绍了多种线性方程组库，其本质都是求解大规模的 $Ax=b$ 方程，我们可以在设计之初就使用策略模式，将调用各个库封装成单独的策略，这样在调用的时候就可以根据实际情况选择合适的求解库。不仅有利于开发，也有利于维护扩展和测试。



如上图所示，我们将MKL，Eigen，Blas调用封装成独立的strategy，统一参数调用接口Run()。再需要添加新的库时，按照该接口设计即可，在程序中调用新的库时，无需改动其它接口，保证了功能的独立，不会影响已有功能。

FEM之在求解器中使用设计模式(7)---属性模式

原创 www.cae-sim.com 多物理场仿真技术

收录于话题

#设计模式 8 #求解器开发 22



属性模式是笔者提出的一种设计模式，本质上是针对求解器数据的特点，对已有设计模式的融会贯通和应用。

问题的提出：

求解器处理的单元数据有如下特点：

- 1.单元数据量大，一般都在百万千万级别以上；
- 2.单元几何数据格式固定，含有各种属性数据；
- 3.单元的属性种类多，且较多重复数据；
4. 一个模型中单元类型数有限；
5. 不会有海量的属性数据；

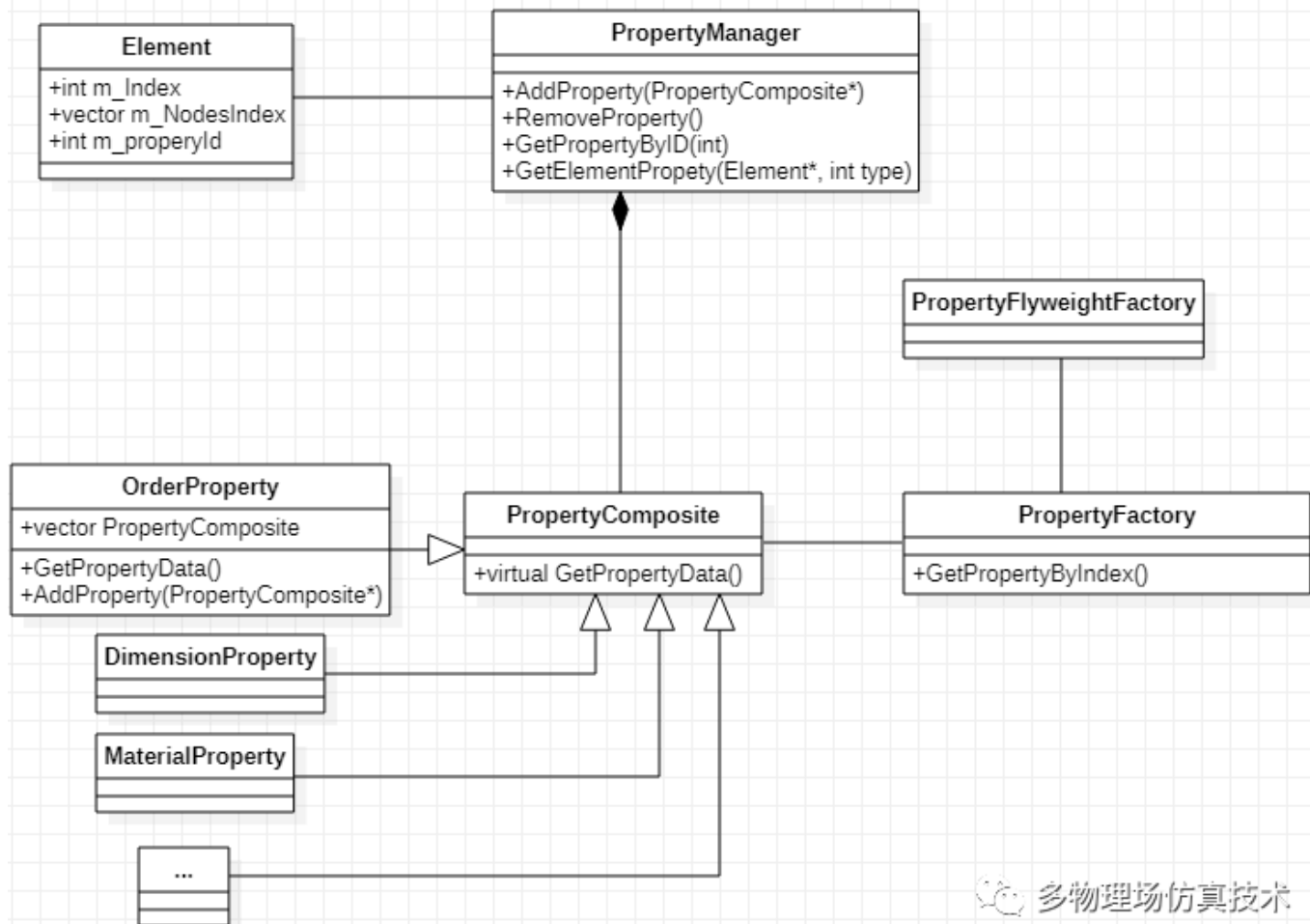
属性模式需要使用以下基本设计模式：

1. 享元(flyweight)模式；
2. 工厂模式；
3. Composite模式；
4. 单例模式

扩展可以加入：

1. 桥接模式；
2. 观察者模式

UML图如下：



说明：

1. 单元类简化为只包含**单元索引**，**属性索引**和**节点索引编号**；
2. 属性类定义为使用composite模式的类，即一个属性中可以包含多个属性，定义了每种属性的具体数据；
3. PropertyManager为使用单件模式的属性管理类，使用该类可以取得单元索引的属性类；
4. PropertyFlyweightFactory为使用享元模式的工厂类，该类使用工厂模式根据有限元数据创建所有属性类；且每种属性类唯一对应一个索引。

改进：

1. 该模式中当数据改变时，可以加入Observer模式，更新每种属性数据；
2. 属性可以根据实际情况利用Bridge模式分成固定的几大类型，比如几何，网格，荷载类型等等；

属性模式的优点：

1. 将单元内部的属性转移到外部，简化单元内部数据结构，从设计上避免大量数据的拷贝；
2. 通过外部属性的管理，将属性和单元分离，使得属性和单元在设计上可以独立变化；

3. 通常情况下，一个模型的单元类型的属性都是相同的，所以属性模式能很好的满足求解器的数据要求，尤其是非通用仿真求解器。

属性模式针对使用有限元方法的求解器数据结构管理，具有普遍适用性。

属性模式的缺点：

1. 需要管理复杂的属性类，当属性类别越来越多时，增加管理复杂度；
2. 数据量大的时候，属性数据操作会有一些性能损失，当然跟求解器计算时间相比，微不足道。

FEM之在求解器中使用设计模式(8)---Command和Observer模式

原创 www.cae-sim.com 多物理场仿真技术

收录于话题

#设计模式 8 #求解器开发 22



Command(命令)模式和Observer(观察者)模式也是软件研发中常用的设计模式。

Command模式使用场景：

- 1.系统需要将请求调用者和请求接收者解耦，使得调用者和接收者不直接交互。
- 2.系统需要在不同的时间指定请求、将请求排队和执行请求。
- 3.系统需要支持命令的撤销(Undo)操作和恢复(Redo)操作。
- 4.系统需要将一组操作组合在一起，即支持宏命令。

具体解释：我们在求解器研发中的各种数据加载，计算，流动都可以看成是由命令发出。我们可以将各种命令的发出者，具体实现者，发出模式，命令管理等分别开来，使其可以独立运行。

Observer模式使用场景：

定义对象间的一种一对多的依赖关系，以便当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并自动刷新。

具体解释：比如后处理应力数据有多种表现形式：二维XY视图，云图，报表等，当原始数据发生改变时候，这些视图接受到数据变动通知时，其表现内容可以自动更新。

用一个具体例子来说明Command模式和Observer模式的使用：

以最简单的静力分析为例，在求解器中我们将整个流程定义如下：

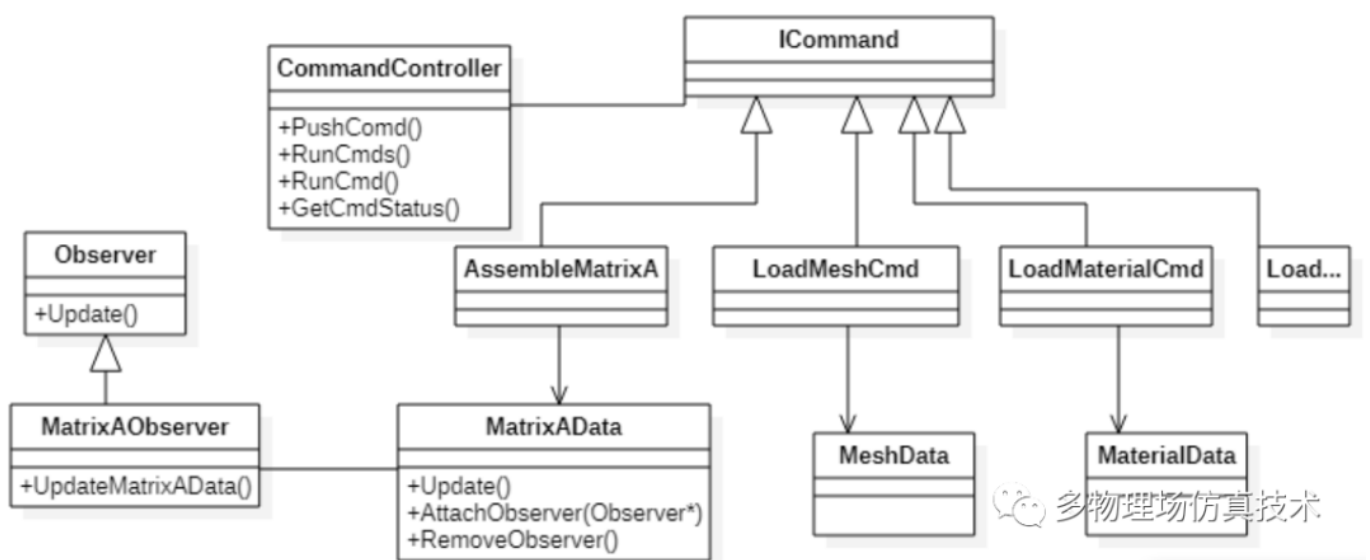
1. 加载网格数据；
2. 加载材料数据；

3. 加载边界和荷载;
4. 组装单刚矩阵;
5. 组装总刚矩阵;
6. 组装荷载;
7. 求解方程组.

需求如下:

1. 载入数据后形成总刚矩阵, 中途修改材料数据, 再次组装总刚矩阵.

我们将每一步操作都对应为一个Command, 将数据的修改使用Observer模式。具体UML如下图:



1. 所有命令从ICommand继承, 每个具体的Command对应一个步骤;
2. 加载材料数据后形成MaterialData
3. MatrixAobserver 实现 Observer 的 Update 接口, 并在总刚矩阵中设置 MatrixAObserver, 当材料数据改变时, 调用Observer接口更新刚度矩阵
4. CommandController控制所有的命令执行, 获取命令状态, 以及命令调度;

通过Command和Observer模式, 可以实现求解器中顺序调用过程, 执行状态和数据的解耦, 在不影响数据结构的情形下实现数据的动态更新。

改进:

1. 求解器中所有的流程都可以采用**Command模式**, 建立更细度的流程控制体系;
2. 完善**命令管理类**, 添加命令流程, 状态, 逻辑, 序列管理;
3. 在求解大规模线性方程组过程中, 尽可能使用**数据缓存+观察者模式**以提升求解性能, 避免无谓的流程重复性能开销。

一篇文章入门求解器开发 设计模式

原创 www.cae-sim.com 多物理场仿真技术

收录于话题

#求解器开发 22 #软件研发测试工程师 15



求解器是仿真软件的最核心模块，开发求解器是一项复杂繁琐的工作，包含了较多的数据结构，业务逻辑，专业算法和数值计算，合理的使用软件设计模式能优化求解器的架构设计，提升求解器的质量。

但现实是，由于求解器开发者大部分都是理工科出身，比如力学，流体等有较强的专业背景，而非计算机专业，因此在业务逻辑处理上较强，而在数据组织，软件架构上有所欠缺；而一般的程序员又缺乏仿真专业知识背景。

因此写了设计模式在求解器中应用系列，希望在专业业务开发和软件架构设计直接搭起一个桥梁。

本文介绍了几种常见的软件设计模式在求解器开发中的应用，并在已有设计模式基础上，针对有限元数据结构特点，提出了一种属性模式。

FEM之在求解器中使用设计模式(1)--Bridge模式

FEM之在求解器中使用设计模式(2)---Abstract Factory模式和Builder模式

FEM之在求解器中使用设计模式(3)---Composite模式

FEM之在求解器中使用设计模式(4)---Builder模式

FEM之在求解器中使用设计模式(5)---Adaptor模式

FEM之在求解器中使用设计模式(6)---Strategy模式

FEM之在求解器中使用设计模式(7)---属性模式

FEM之在求解器中使用设计模式(8)---Command和Observer模式

FEM之在求解器中使用设计模式(9)---Facade模式和Proxy模式

原创 www.cae-sim.com 多物理场仿真技术



再介绍软件开发中常用的两种设计模式 Facade（门面）模式和Proxy（代理）模式。

Facade模式个人认为是和单例一样，是比较简单也容易理解的一种模式。

Facade的含义是当外部和一个子系统沟通时，可以通过一个统一的门面对象进行。

比如客户发现了软件bug，客户可以和软件团队的架构师，高级工程师，测试工程师，产品工程师任意一个人反馈，但最好的是直接反馈给产品经理，产品经理再和团队内部的人员沟通。这就保证客户和研发部门始终只通过产品经理沟通，而产品经理就是门面。

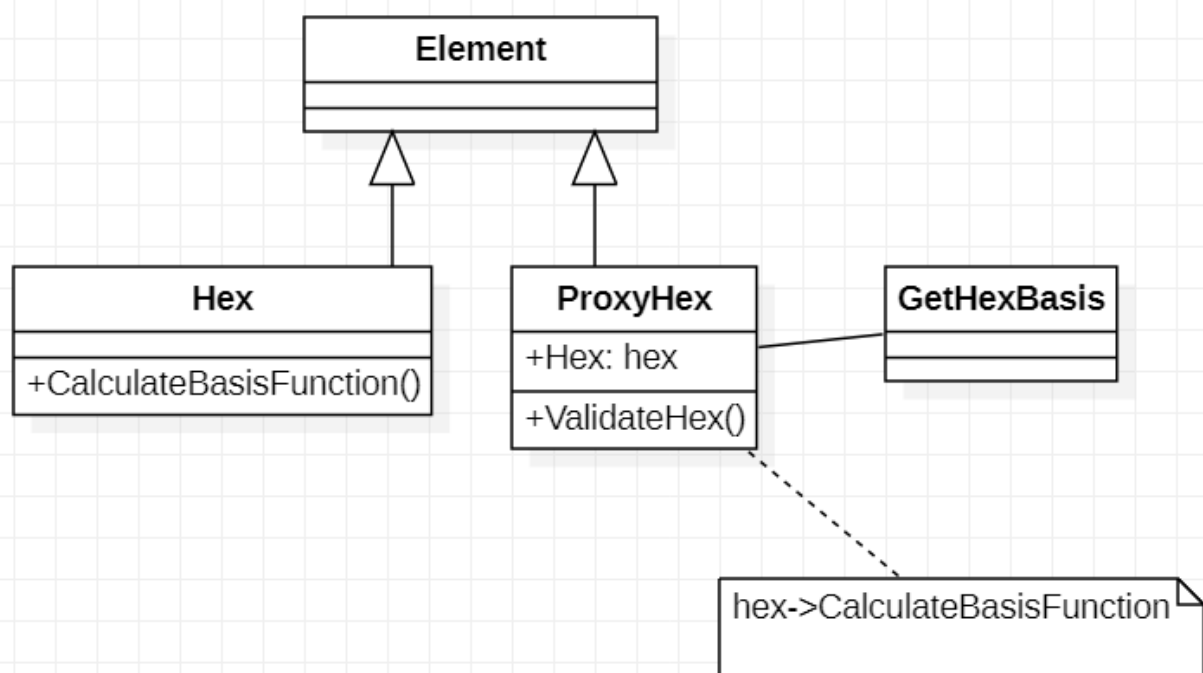
所以从这个意义看，Facade是比较容易理解和实现的模式。在求解器开发中可以尽量使用这种模式。比如在组装刚度矩阵过程中，需要频繁取网格中的各种几何和属性数据，每种数据都有自己的数据结构，这时候可以构造一个网格管理器，通过管理器的接口来存取数据，避免直接存取数据。这样做的好处是组装刚度矩阵只需要关注组装逻辑，而不用花太多精力去管理存取数据，同时也便于任务的拆分。Facade比较简单，就不画UML图了。

Proxy（代理）模式

代理模式的目的是为对象提供一种代理以控制对象的访问。

代理模式在数据管理中也是一种常见的模式，比如我们定义的网格数据：点的坐标。设计点的类通常包含三维坐标xyz，如果需要计算一个点到另一个点的距离，有两种方法，一是在点的类内部实现，也可以添加另外的工具类来实现；求解器的数据中有很多类似的结构需求。

在类的内部实现功能，类似函数多的时候，类的设计臃肿，且和类本身定义没有多大关联；如果使用工具类，缺乏对类本身细度的控制，而Proxy代理模式能很好解决这些问题。



多物理场仿真技术

上图显示了使用 ProxyHex 代理类来计算六面体 Hex 的基函数，该代理类使用 ValidateHex 函数检验 Hex 数据的有效性，如果有效则调用 Hex 的基函数的计算函数，从而简化了 Hex 类的设计。

简单聊聊Techsoft3d公司

原创 www.cae-sim.com 多物理场仿真技术



Techsoft3D公司 (www.techsoft3d.com) 对大多国人比较陌生，但对于做图形显示的人应该非常熟悉。

公司1996年在美国成立，创始团队来自于Autodesk，其主要业务集中在计算机图形显示上，最早旗舰产品为HOOPS，后来发展到云端，手机端，以及VR，OEM，计算机视觉等方向。

最早的HOOPS提供源码，为了保护源码，代码写的比较晦涩。笔者仔细阅读过代码，并没有太多技术上的瓶颈，其底层实现仍然是OpenGL和DirectX，在这之上进行了封装。

HOOPS作为商业图形渲染引擎产品，其最大的成功之处在于和ACIS的捆绑销售，ACIS之前做过详细介绍，参考附录。ACIS本身是一个几何引擎，只有简单的显示模块，无法处理工业级别的图形渲染，HOOPS+ACIS的策略使得HOOPS显得比较成功，一些主流的产品比如solidworks，spaceclaim，HFSS，CST等都使用HOOPS。从技术角度来看，开源VTK，底层OpenGL都能很好的处理ACIS的显示，尤其是VTK，在大规模图形处理方面比HOOPS要强，更适合科学计算方面的图形化显示。

2020年10月和11月，Techsoft3D分别收购了CeetronAS和VKI两家公司。这两家公司是软件SDK提供商，CeetronAS专业从事CAE领域的图形视觉开发，而VKI则提供了CAE领域从网格划分，FEM三方接口和求解器等基础开发组件。可见Techsoft3D也是想垄断CAE软件研发基础组件，致力于更精细的图形细分市场。

Techsoft3D公司如果放在中国，可能活不过3年。在当前科技领域技术“断供”，国产自主的战略环境下，对国内科技领域的高层决策者是一个最好的参考例子：只有保护好知识产权，普及知识产品权意识，让软件技术产品成为高附加值产品，才能从底层推动整个科技应用领域的发展。

希望将来国内也有越来越多类似Techsoft3d公司出现，那样国内高水准的工业软件遍地开花也就指日可待了。

仿真软件开发工具介绍（1）--ACIS/HOOPS

仿真软件开发工具介绍（8）--VKI

一篇文章入门格子玻尔兹曼方法 (LBM)

邓子平 多物理场仿真技术



“大多数人并不知道Powerflow这个软件，tb上也找不到枪版。但该软件在汽车行业比较有名。其快速，准确，便捷的特点成为许多汽车公司仿真首选。Powerflow提供了流体，热，噪音分析和优化等功能。而开发公司Exa (<http://www.exa.com/>) 按软件使用时间收费，说日进斗金一点也不过，是典型的埋头闷声发财的公司”

以上是2015年笔者写的文章中对Powerflow的介绍，两年后也就是2017年Exa公司以4亿美元被达索收购，Exa的网站也再打不开，Powerflow成为达索旗下的一款产品。

原文：

http://blog.sina.com.cn/s/blog_68c287690102vi85.html

在收购的新闻稿中，有如下描述：

"Exa加盟后，达索系统的**3DEXPERIENCE**平台将为客户提供经验证的多样化解决方案产品组合，它整合了格子玻尔兹曼(Lattice Boltzmann)流体仿真技术、Exa的全产业化解决方案以及将近350名极富经验的仿真专业人士。与传统方法相比，这套解决方案能够更快速准确地为空气动力学应用、流体声学应用、热管理应用和其他行业中日益增多的应用求解高难度流体问题"

"Exa公司总裁兼首席执行官Stephen Remondi指出：“Exa经验证的格子玻尔兹曼(Lattice Boltzmann)仿真技术、丰富的专业技能与工业专业知识是我们一直在高动态流仿真市场处于领先地位的原因。达索系统3DEXPERIENCE平台和SIMULIA产品组合的一部分将为我们所有行业的客户创造巨大价值"

Powerflow的核心求解器采用了**LBM**算法 (**L**attice **B**oltzmann **M**ethod 格子玻尔兹曼方法)

从对象尺度来看，我们一般把研究对象分为**宏观**，**介观**和**微观**。常见的有限元，有限体积，有限差分等属于宏观的范畴，在宏观几何上，把对象看作是连续介质，连续介质是很多数值计算分析方法的前提条件。

介观是介于宏观与微观之间的一种体系。处于介观的物体在尺寸上已是宏观的，因而具有宏观体系的特点；但是由于其中电子运动的相干性，会出现一系列新的与量子力学相位相联系的干涉现象，这又与微观体系相似，故称“介观”，介观和微观把对象看做独立的细小个体而非连续介质。

在流体中的常用控制方程是 N-S方程，但在很多情况下该控制方程很难描述或者求解，比如复杂多相流系统，多空介质流体等，而从介观微观的分子动力学上建立的简化模型能解决这些问题。**Boltzmann 方程可以简单的理解为N-S方程在微观水平的表达形式。其基本思想是划分网格，计算每一个网格节点上离子的状态（碰撞，迁移，速度，方向，分布，属性等）**，当然实际计算中要复杂的多。其实很多流体现象都可以用微观的分子动力学来表示，这也是格子波尔兹曼方法的理论基础。

LBM源自晶格气体自动机（LGA）方法，可以将其视为空间，时间和粒子速度均为离散的简化虚拟分子动力学模型。例如，在二维FHP模型中，每个晶格节点通过三角形晶格上的6个晶格速度连接到其邻居。以给定的晶格速度移动的晶格节点上可以有0或1个粒子。在一个时间间隔后，每个粒子将沿其方向移动到相邻节点；此过程称为传播或流传输步骤。当一个以上的粒子从不同的方向到达同一节点时，它们会按照一组碰撞规则发生碰撞并改变其速度。

LBM可以轻松解决很多传统宏观FVM无法解决或者棘手的问题：

1. 数据预处理和网格生成效率高；
2. 方便支持并行数据分析，处理和评价；
3. 支持解决多相流的小液滴和气泡；
4. 支持解决复杂的几何形状和多孔介质；
5. 支持复杂，耦合流动与传热和化学反应；

与传统的计算流体力学方法(如有限差分，有限单元法、有限体积法等)相比，格子 Boltzmann方法主要有以下优点：

- (1)算法简单，简单的线性运算加上一个松弛过程，就能模拟各种复杂的非线性宏观现象
- (2)能够处理复杂的边界条件
- (3)格子Boltzmann方法中的压力可由状态方程直接求解
- (4)编程容易，计算的前后处理也非常简单
- (5)具有很高的并行性
- (6)能直接模拟有复杂几何边界的诸如多孔介质等连通域流场，无须作计算网格的转换

目前使用LBM的商用软件xflow， powerflow等
开源软件有 Palabos， OpenLB：
<http://www.palabos.org/>

<http://www.numhpc.org/openlb/>

想进一步了解格子波尔兹曼方法，知乎上有更详细的介绍。

笔者的话：

FDTD, LBM, 离散元, 计算材料学的软件研发有相似点, 都不需要基于B-REP的三维几何和复杂的网格划分, 尤其后三者图形显示上以球, 圆柱, 立方体等基本元素为主, 而且大部分类型相同, 通过实体+视图矩阵变换可以轻松处理大规模数据, 因此在软件研发前后处理上难度大大降低; 研发核心聚焦在求解器上, 同时由于GPU, openMP等各种加速算法的门槛降低, 硬件资源费用降低以及量子计算机的迅速发展, 使得仿真效率越来越高, 以后会越来越普及。对于缺乏复杂几何网格开发的国内科研机构 and 高校, 求解器算法优势将更突出, 开发相关数值仿真软件其实是一个不错机会。

ACIS vs ANSYS

原创 邓子平 多物理场仿真技术

ACIS的发音为 /'eisis/

ANSYS的发音为 /ænses/

两者发音很接近，其实通过第一个重音元音可以简单地区分开来。在以往沟通中经常出现两者混淆的情况，最尴尬的一次是：面对面的讲如何用ACIS进行几何造型，差不多半小时沟通结束后，对方一直以为做的是ANSYS的二次开发。

ACIS是美国spatial technology公司（现在属于达索子公司）的产品，是应用于CAD系统开发的三维几何内核。

ANSYS 是目前全球排名第一的CAE软件公司，也是其公司早期旗舰产品的名称。

稍微讲一讲ACIS的历史。ACIS取自于Alan Grayer, Ian Braid导师Charles Lang，以及Ian Braid本人 + Solid 的首写字母。其中Ian Braid首先在1973年提出了Boundary Representation的概念，也就是现在通常所说的BRep结构。在这套结构中通过BODY, SHEEL, FACE, LOOP, COEDGE等概念可以完整表达三维几何的拓扑结构数据和几何数据，并很好的支持各种三维建模，修改，计算等操作。

三人最早成立了Shape Data公司，一直从事三维实体造型的工作。之后公司被另一家公司收购之后，在原有工作的基础上，开始了Parasolid的研发，1988年Parasolid被UG收购。Parasolid被收购的几乎同时，三人又和Spatial公司开始了ACIS的研发，再后面就是Spatial公司在2000年被达索收购，UG在2007年被西门子收购。另外要提一下Autodesk，Autodesk通过打官司拿到了ACIS在14版本左右的源码，后面也是独立开发发展。

如上所述，Parasolid和ACIS同宗同源，底层概念和结构都几乎完全相同，最初开发者也一样。在经过长期的演化之后，到现在两者还是走出了不同的商业和市场路线。

一篇文章入门三维几何内核

仿真软件开发工具介绍（1）--ACIS/HOOPS

仿真软件开发工具介绍（7）--Parasolid/OpenCascade

声场求解器开发之(3)---C++开发BEM求解器

www.cae-sim.com 多物理场仿真技术

本文主要介绍基于BEM（边界单元方法）方法开发用于计算声学求解器的基本知识。

1. 概要

在数值仿真领域，有限元方法一直是首选，但在一些细分领域，其他方法可能更有效，比如有限体积法（FVM）对流体，边界元法（Boundary Element Method）对声，电磁散射。

以声场为例，声音向外传播，作用区间是一个典型的无限范围，如果使用FEM，需要对整个区域划分网格，尤其在三维空间，随着计算范围的增大，网格数目会急剧增大。而采用BEM，只需要求解声场边界上的数值，大大降低了计算量，提高了计算效率。

既然BEM效率这么高，为什么使用范围不如FEM广泛呢？主要两点原因：

1. FEM 是一种纯数值解法，而BEM是半数值解，也就是说要以解析基本解为前提条件
2. 因为最后形成的线性方程组为非对称满秩矩阵，限制了工程上的求解规模。桌面单机几千自由度就已无法解出结果，虽然通过FMM（快速多极子方法）解决了这个问题，但因为FMM算法本身实现也有一定难度，因而没有FEM应用广泛。统计显示当模型自由度在15000以下时，传统的BEM效率更高，而大于15000时，传统BEM计算量会急剧上升，此时FMM的计算量仍然呈线性增长。

BEM 基于解析解，在处理某些特定领域（声场，电磁场，连续介质弹性力等无限域问题）具有 精度高，降低维度等特点，同时通过与FEM相结合，能够综合两者的优势，提高计算效率和精度。

2. 理论

前面介绍过声场的控制方程为赫姆霍兹方程，边界元方法的第一步是要将PDE转换为边界积分方程，对于三维空间的方程，需要使用格林函数将体积分转换为面积分。

通常流程是：

1. 利用算子的基本解作为权函数，按加权余量格式得到区域上的积分方程；
2. 利用高斯公式（格林函数）建立区域内积分和边界积分的关系，从而得到区域内任意一点的通解变量表示的积分表达式；（利用格林函数可以将三维体积分转为二维面积分）
3. 将基本解的奇异点P趋于边界点p，得到边界积分方程。

这个过程简称为用加权余量法建立边界积分方程。

BEM求解流程与FEM基本类似，划分网格，建立线性方程组，求解方程，不同的只是建立的本构方程不同。

因为BEM求解区域只在边界上，所以三维问题只需要划分面网格，二维问题划分线网格，使前处理工作大为简化。

BEM的计算流程与FEM相似：

1. 生成网格（三维使用面单元，二维计算使用线单元）

2. 计算单元影响矩阵
3. 组装总体影响矩阵
4. 求解线性方程组
5. 根据边界上的值计算任一点的声场压力

不失一般性，以线性四边形单元求解空间声场问题。

单元定义可以从FEM的四边形单元派生，因此可以直接使用FEM中等参单元定义的形函数，再加上声场计算所需的函数与变量。针对每个单元需要计算H和G矩阵，这两个矩阵后面用来组装整体矩阵。为了求解精度，三维实体划分网格的时候，四边形边长需要控制在波长的一定比例之内。

1. 声场线性四边形单元定义如下：

```
#include "FEM_Element.h"
#include "FEM_Quad.h"

class BEM_Quad : public FEM_Quad
{
public:
    BEM_Quad(void);
    ~BEM_Quad(void);

    void CalGeMatrix();
    void CalHeMatrix();
    void CalAngle();

    CMatrix* GetHe();
    CMatrix* GetGe();

    double GetAngle();
private:
    CMatrix* m_Ge;
    CMatrix* m_He;
    double m_Angle;
};
```

多物理场仿真技术

单元函数说明：

CalGeMatrix() 和 CalHeMatrix()分别用来计算 G矩阵和H矩阵。关于矩阵介绍参考附录参考1。

2. 材料属性：

材料可以直接使用FEM定义的部分

1. 角频率 = $2 * \pi * v$ (v为频率 单位 赫兹);
2. 声在介质中的传播速度;
3. 介质的密度;

3. 整体刚度组装流程：

```

void BEM_Model::AssembleGMatrix()
{
    for (int i = 0; i < elemNum; i++)
    {
        //遍历全局所有节点;
        for (int j = 0; j < 4*elemNum; j++)
        {
            //计算单元影响矩阵;
            elemMap[i]->InitGIData(gldata.at(j));
            elemMap[i]->CalGeMatrix();
            elemMap[i]->CalHeMatrix();

            //组装全局影响矩阵;
            AssembleGMatrix(elemMap[i]);
            AssembleGMatrix(elemMap[i]);
        }
    }
}

```

多物理场仿真技术

4. 边界条件带入总体矩阵，构建线性方程组

边界条件包括压力，速度以及声阻

5. 结果计算

求解得到边界上的声场压力和速度之后，就可以直接利用表达式计算得到空间中任意一点的压力和速度。

由于最后的HG矩阵为满秩矩阵，在本人I7四核 + 8G PC机上单元达到5000时，基本上已经算不出结果。

参考：

1. *STRUCTURE-ACOUSTIC ANALYSIS USING BEM/FEM IMPLEMENTATION IN MATLAB.* **FREDRIK HOLMSTRÖM**
2. *The Boundary Element Method in Acoustics.* **Stephen Kirkup**
3. *BEM-FEM Acoustic-Structure Interaction For Modeling and Analysis of Spacecraft Structures Subject to Acoustic Excitation.* **Harijono Djojodihardjo**

声场求解器开发之(4)---BEM声场求解器基础

原创 www.cae-sim.com 多物理场仿真技术

本文主要介绍基于BEM（边界单元方法）方法开发用于计算声学求解器的基本知识。

1. 概要

在数值仿真领域，有限元方法一直是首选，但在一些细分领域，其他方法可能更有效，比如有限体积法（FVM）对流体，边界元法（Boundary Element Method）对声，电磁散射。

以声场为例，声音向外传播，作用区间是一个典型的无限范围，如果使用FEM，需要对整个区域划分网格，尤其在三维空间，随着计算范围的增大，网格数目会急剧增大。而采用BEM，只需要求解声场边界上的数值，大大降低了计算量，提高了计算效率。

既然BEM效率这么高，为什么使用范围不如FEM广泛呢？主要两点原因：

1. FEM 是一种纯数值解法，而BEM是半数值解，也就是说要以解析基本解为前提条件
2. 因为最后形成的线性方程组为非对称满秩矩阵，限制了工程上的求解规模。桌面单机几千自由度就已无法解出结果，虽然通过FMM（快速多极子方法）解决了这个问题，但因为FMM算法本身实现也有一定难度，因而没有FEM应用广泛。统计显示当模型自由度在15000以下时，传统的BEM效率更高，而大于15000时，传统BEM计算量会急剧上升，此时FMM的计算量仍然呈线性增长。

BEM 基于解析解，在处理某些特定领域（声场，电磁场，连续介质弹性力等无限域问题）具有 精度高，降低维度等特点，同时通过与FEM相结合，能够综合两者的优势，提高计算效率和精度。

2. 理论

前面介绍过声场的控制方程为赫姆霍兹方程，边界元方法的第一步是要将PDE转换为边界积分方程，对于三维空间的方程，需要使用格林函数将体积分转换为面积分。

通常流程是：

1. 利用算子的基本解作为权函数，按加权余量格式得到区域上的积分方程；
2. 利用高斯公式（格林函数）建立区域内积分和边界积分的关系，从而得到区域内任意一点的通解变量表示的积分表达式；（利用格林函数可以将三维体积分转为二维面积分）
3. 将基本解的奇异点 P 趋于边界点 p ，得到边界积分方程。

这个过程简称为用加权余量法建立边界积分方程。

BEM求解流程与FEM基本类似，划分网格，建立线性方程组，求解方程，不同的只是建立的本构方程不同。

因为BEM求解区域只在边界上，所以三维问题只需要划分面网格，二维问题划分线网格，使前处理工作大为简化。

FEM之单元(1)---三角单元介绍

原创 www.cae-sim.com 多物理场仿真技术



本文可以当做有限元的小品文看，目的是对有限元理论进行小结。

1.一阶三角形场函数

三角形单元因生成容易，计算简单，容易加密，成为有限元分析中最常用的单元。

针对一阶单元，即一个三角形有三个点。

对于任意一个三角形，假设三顶点坐标点为 (x_1, y_1) (x_2, y_2) (x_3, y_3) ，三点的场函数分别定义为点 $F_{x1}, F_{y1}, F_{x2}, F_{y2}, F_{x3}, F_{y3}$

三角形内任意一点的场函数可以表示为

$$F_x = a_1 + a_2 * x + a_3 * y$$

$$F_y = b_1 + b_2 * x + b_3 * y$$

其中 F_x 为 x 方向的场函数， F_y 为 y 方向的场函数， $a_1, a_2, a_3, b_1, b_2, b_3$ 为常数项。

由于已知三点坐标，即在三点上也满足上式。将三点的坐标带入上式，可得6个方程。

例：

$$F_{x1} = a_1 + a_2 * x_1 + a_3 * y_1$$

$a_1, a_2, a_3, b_1, b_2, b_3$ 共6个变量，6个方程，可以求出：

$$a_1 = ((x_2 * y_3 - x_3 * y_2) * F_{x1} + (x_3 * y_1 - x_1 * y_3) * F_{x2} + (x_1 * y_2 - x_2 * y_1) * F_{x3}) / (2 * A)$$

$$b_1 = ((x_2 * y_3 - x_3 * y_2) * F_{y1} + (x_3 * y_1 - x_1 * y_3) * F_{y2} + (x_1 * y_2 - x_2 * y_1) * F_{y3}) / (2 * A)$$

$$\text{其中 } A = ((x_2 * y_3 - x_3 * y_2) + (x_3 * y_1 - x_1 * y_3) + (x_1 * y_2 - x_2 * y_1)) / 2$$

公式看起来比较繁琐，这些转换的目的只有一个：为了让场函数能用 三角形的三个顶点的坐标来表示。

将求解出来的 $a_1, a_2, a_3, b_1, b_2, b_3$ 带入到

$$F_x = a_1 + a_2 * x + a_3 * y$$

$$F_y = b_1 + b_2 * x + b_3 * y$$

中：

可以得到新的表达式：

$$F_x = N_1 * F_{x1} + N_2 * F_{x2} + N_3 * F_{x3}$$

$$F_y = N_1 * F_{y1} + N_2 * F_{y2} + N_3 * F_{y3}$$

其中

$$N1=(x2*y3-x3*y2)+(y2-y3)*x+(x3-x2)*y$$

以此类推，详细推导公式可以参考任意一本有限元书籍

总之最后的结果是：

三角形内任意一点的场函数可以用 三个顶点的坐标，场函数，以及该任意点坐标表示，这样一来，只要我们求出了顶点的场函数的值，就可以通过插值计算出三角形内任意一点的场函数值。如果是矢量，需要两个表达式，标量只要一个表达式。

2.偏微分方程概念

有限元方法的目的就是求解偏微分方程，利用有限元方法求解偏微分方程主要有两种：

1. 变分原理的Ritz方法
2. 利用加权余值中的 伽辽金法 (Galerkin weighted residual method)

一些基本概念

1.边界：

第一类边界,直接描述边界上结果，用于已知边界确切结果 比如边界上外力 $F = 10$ ，温度 $T=40$

又叫 Dirichlet (狄利克雷) 边界

第二类边界 不直接给出确切结果，用导数方式给出

又叫Neumann (诺伊曼/诺曼) 边界

第三类边界可以看成是 第一类和第二类边界的叠加

又叫Robin 边界

2.常用标记：

梯度、散度和旋度是矢量分析里的重要概念。之所以是“分析”，因为三者是三种偏导数计算形式。这里假设读者已经了解了三者的定义。它们的符号分别记作如下：

$$\text{grad } \varphi \leftrightarrow \nabla \varphi$$

$$\text{div } \mathbf{F} \leftrightarrow \nabla \cdot \mathbf{F}$$

$$\text{rot } \mathbf{F} \leftrightarrow \nabla \times \mathbf{F}$$

从符号中可以获得这样的信息：

- ①求梯度是针对一个标量函数，求梯度的结果是得到一个矢量函数。这里 φ 称为势函数；
- ②求散度则是针对一个矢量函数，得到的结果是一个标量函数，跟求梯度是反一下的；
- ③求旋度是针对一个矢量函数，得到的还是一个矢量函数。

这三种关系可以从定义式很直观地看出，因此可以求“梯度的散度”、“散度的梯度”、“梯度的旋度”、“旋度的散度”和“**旋度的旋度**”，只有旋度可以连续作用两次，而一维波动方程具有如下的形式

$$\frac{\partial^2 y}{\partial x^2} = a^2 \frac{\partial^2 y}{\partial t^2} \quad (1)$$

其中 a 为一实数，于是可以设想，对于一个矢量函数来说，要求得它的波动方程，只有求它的“旋度的旋度”才能得到。下面先给出梯度、散度和旋度的计算式：

$$\nabla \varphi = \mathbf{i} \frac{\partial \varphi}{\partial x} + \mathbf{j} \frac{\partial \varphi}{\partial y} + \mathbf{k} \frac{\partial \varphi}{\partial z} \quad (2)$$

$$\nabla \cdot \mathbf{F} = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z} \quad (3)$$

$$\nabla \times \mathbf{F} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ F_x & F_y & F_z \end{vmatrix} \quad (4)$$

多物理场仿真技术

上图截取自：

http://blog.sina.com.cn/s/blog_5701b67c0100x7fv.html

3. 不同物理场应用

平面力学问题：

场函数 F_x ， F_y 取为位移

由平面应变为位移对坐标的导数

即x方向应变 = F_x'/x

y方向应变 = F_y'/y

可知 $F_x = a^2$ ， $F_y = b^3$ 都为常数。所以一阶三角形为常应变单元，即一个单元内应变不发生变化，为了保证精度，所以在物体应变变化大的地方要加密网格。

弹性力学偏微分方程中的应力，应变通过变换也都可以用场函数来表示，推导可参考有限元书（目前市面上关于有限元大都是力学方面的）。

由物体平衡时，物体整体势能最小，将势能函数取变分可得到2D静力平面问题的矩阵表达式，表达式中包含了刚度矩阵，位移向量和节点荷载向量，根据整理的公式，就可以直接用代码实现了。

平面热传递:

因为温度是标量, 所以推导求解比力学问题要简单, 场函数定义为:

$$F = a_1 + a_2 * x + a_3 * y$$

仍然可以利用前面的推导, 得出任意一点场函数的表达式。

平面温度场方程为:

$$k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + q_v - \rho c_p \frac{\partial T}{\partial t} = 0$$

以第一类边界为例, 带入泛函计算, 最后可得泛函表达式:

$$[K]\{T\}^e + [N]\left\{\frac{\partial T}{\partial t}\right\}^e - \{p\}^e$$

公式的推导参考《有限单元法在传热学中的应用》

针对稳态, 第二项可以去掉。

平面电磁

场函数F取静电势/磁势, 电场/磁场

考虑如下二阶微分方程:

$$-\frac{\partial}{\partial x} \left(\alpha_x \frac{\partial \Phi}{\partial x} \right) - \frac{\partial}{\partial y} \left(\alpha_y \frac{\partial \Phi}{\partial y} \right) + \beta \Phi = f$$

常用的二维拉普拉斯方程, 泊松方程和赫姆霍兹方程都是上述方程的特殊形式

利用变分建立该函数的泛函, 将场函数带入泛函, 最后可推导出单个单元方程

$$[K]\{F\} - B = 0$$

下一步和静力学一样, 组装成总体刚度矩阵, 求解

在电磁计算中, 使用三角面片网格会出现伪解的现象, 而且由于是多种材料, 在处理边界时会比较困难, 于是引入了矢量单元, 将自由度赋在边而不是节点上 (edge element), 具体参考《电磁场有限元方法》

平面声学

场函数F 取声压

声学中需要求解波动方程

$$\nabla^2 p = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2}$$

右边的P为声压。对于简谐振动, 可以改写成 赫姆霍兹方程 (Helmholtz)

得出该方程的泛函, 也可以通过能量平衡原理导出泛函。取第一边界条件, 场函数带入可求得

$$[K]*F = a[B]$$

之后可求得声压和频率

平面流体

流体主要求解Navier-Stokes方程

理论上三角形单元是可以用来求解流体N-S方程的，实际上由流体的特点，有限元通常使用四边形和六面体。考虑到计算效率，目前CFD软件用的最多的还是有限体积法。

对于三角形二阶单元（每条边上有一个点，一个单元共有6个点）以及高阶，推导过程类似。工程上二阶单元已经能很好满足要求。有些商业软件提供了高阶单元（p单元），高阶单元的好处是只需少量网格，针对某些特殊case，可以在不改变网格情况下，通过升阶提升精度。

有限元是求解偏微分方程一种方法，如果想发开有限元程序，求解偏微分方程是绕不过去的槛。不过好在科学家，数学家们已经做了很多研究，不用我们自己去推导头疼的公式，只要记住公式和结论都行了，力学中二阶的四面体单元刚度矩阵有 $30 \times 30 = 900$ 个数据，实在没办法自己推导。但是作为有限元开发，了解推理过程对开发是非常有好处的。比如一般商业有限元程序中都会使用等参单元和高斯积分，积分点在单元内部而不在顶点，所以计算出的结果为高斯点的值，还有死锁，沙漏等诸如此类，这些都是与有限元算法本身特性相关。

FDTD开源软件推荐

www.cae-sim.com 多物理场仿真技术



FDTD（时域有限差分）算法用来求解麦克斯韦方程，历史比较悠久，实现相对容易，开源软件也非常丰富。

笔者推荐两个用过的FDTD的开源软件：

FDTD++

实现简单，windows上可以用VS编译运行

<http://www.fdtddxx.com/>

下载需要注册

Meep

只能在Linux运行，windows编译比较复杂，功能相对丰富

源码可以在github上获得

<https://meep.readthedocs.io/en/latest/>

FEM开源软件推荐

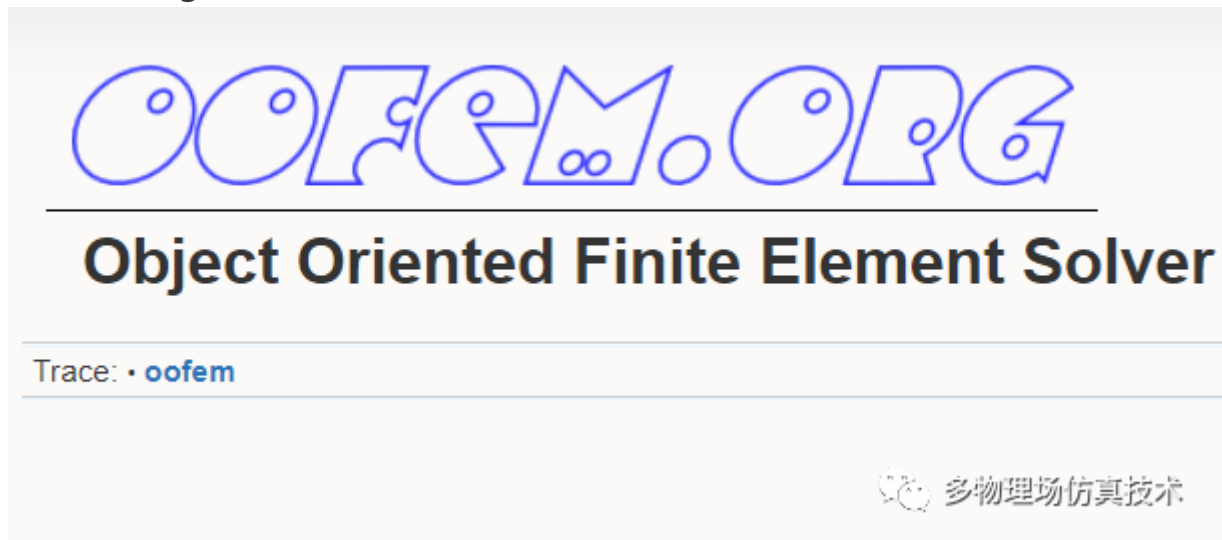
www.cae-sim.com 多物理场仿真技术

之前介绍了FDTD的开源软件，有读者反应希望推荐几个FEM的开源软件。作者仿真软件架构开发十余年，接触了大量的开源FEM工具，主流和小众的开源工具基本都摸过一遍，而且一般一两年都会跟进一次。这些软件在wiki上都能搜到。

从学习角度看，任何一款开源工具深入研究都可以。而code_aster, calculix等从功能上开已经接近商业软件，是相当不错的开源工具。

但是从软件工程，系统工程，以及研发角度看，最值得推荐的是OOFEM。主要原因是OOFEM面向对象，结构清晰，纯C++编写，便于扩展，windows上cmake生成VS工程，可编译调试，提供很多demo，虽然前后处理比较蹩脚，但影响不大。当然更新较慢，最近一次更新是2017年。

www.oofem.org



FEM之优化算法(4)---Isight在仿真软件开发中的应用

原创 www.cae-sim.com 多物理场仿真技术

收录于话题

#仿真研发工具

21个



很多的CAE软件中提供了局部网格参数功能，即可以设置几何局部区域的网格尺寸。网格是影响仿真结果的一个主要因素，网格太稀疏，计算结果不精确，网格太密，浪费计算资源。早期软件的仿真精度很大程度依靠工程师的经验。

为了解决这个问题，出现了自适应网格划分方法，其基本原理是将初始网格划分的比较稀疏，计算一次后，找出结果梯度比较大的区域，对网格进行加密后，再进行计算，继续找出梯度大的区域，如此反复，直到收敛到一个标准为止，该方法前面中有过介绍，参考附录。

该方法具有通用性，但迭代计算比较耗费时间，特别对大case，在对具体行业应用时，还可以进一步优化。通常几何模型的拓扑（即形状，比如导弹，汽车）很少变化，所以在网格参数设置时候可以根据几何形状，材料类型等实际特点一步将网格参数设置到位，可以减少人为的干预。

本文介绍利用优化软件Isight在开发行业CAE软件中的一些经验。（不是给Isight打广告，而是用起来确实比较方便，在自己用过的大case中，也有isight算不出来情况）

工具：Isight, Matlab, C++

功能：对某电子产品散热进行模拟（电子产品结构基本固定，由基板，芯片，盖板组成）

目的：找到三者的最优网格大小设置。

当几何简单时，用户很容易找到网格最优设置。当几何比较复杂时，相当多用户自己找最优解还是有一定难度的。针对如下某类型的电子产品有如下模板：

基板，盖板各有2个，芯片4个，网格使用四面体，在软件中设置好各部分网格的参数，让用户不用干预网格。

	实体名称	材料	属性
基板	J_Body_1	A	P_1
	J_Body_2	A	P_2
芯片	X_Body_1	B	P_3
	X_Body_2	C	P_3
	X_Body_3	C	P_3
	X_Body_4	D	P_6
盖板	G_Body_1	E	P_7
	G_Body_2	E	

步骤：

1. 求出精确解；
2. 设置各部分网格初始参数；
3. Isight中设置优化流程；
4. Matlab提取结果，将结果返回Isight；
5. 找到各个部分网格的最佳参数；
6. 找到材料，属性与网格参数的关系。

第6步最后会计算出网格尺寸和参数的关系表达式，这个需要大量的测试。

关于Isight软件的介绍：

ISIGHT是美国Engineous公司出品的过程集成和优化设计的软件，可以将数字技术、推导技术和设计探索技术有效融合，并把需要大量人工完成的工作由软件自动化处理实现，在多学科优化软件中的市场占有率居首位。ISIGHT软件可以集成仿真代码并提供设计支持，从而对多个设计可选方案进行评估研究，大大缩短了产品的设计周期，显著提高了效率。ISIGHT在缩短产品设计周期、降低产品成本、提高产品质量等方面，取得令人瞩目的突破。优化功能作为ISIGHT软件的重要组成部分，结合ISIGHT平台强大的集成功能和丰富多样的优化算法库，已成为航空航天、汽车、兵器、船舶、电子、动力、机械、教育研究等领域首选的设计优化解决方案。总的来说，ISIGHT软件的优化功能有集成自动化、算法多样化和结果数据可视化分析等三大优势。

一、集成自动化

一个典型的工程需要不断进行"设计 - 评估 - 改进"的循环。CAD和CAE的引入提高了这一过程的效率。CAD加快了造型、装配、出图的设计过程，CAE则减少了大量的试验，提供了有效的分析和评估工具。但是在这种设计过程中，大多数是重复性工作，中间环节容易出错。ISIGHT通过一种搭积木的方式快速集成各种仿真软件，将所有设计流程组织到一个统一的框架中，自动运行仿真软件，并自动重启设计流程，从而消除了传统设计流程中的"容易出错"问题，使整个设计流程实现全数字化和全自动化。

典型的仿真优化过程主要包括以下四个步骤：

- 软件集成（集成一个或多个仿真软件）
- 问题定义（定义设计变量、约束和目标函数）
- 设计自动化（选择优化设计算法）

-数据分析和可视化（实时监控设计分析过程）

二、算法多样化

ISIGHT提供了丰富的优化算法库，包括数值优化算法、智能优化算法，以及处理多目标优化问题的多目标优化算法，既可处理离散变量，也可探索连续的设计空间。对于目标函数存在多峰性、非线性、不可微函数，数值优化算法可能无法寻找最优解，而全局优化算法则为解决这类问题提供了新的思路。工程上，经常出现同时对多个目标进行优化，这些目标之间是相互冲突的，因此设计人员需要对多目标进行比较，权衡，使得多个目标尽可能达到最优。ISIGHT提供了多种多目标优化算法，并支持对多目标优化的pareto图的直观分析。

三、结果数据可视化分析

对于大多数设计软件，算法开始执行后需要等待算法程序结束后用户才可以查看计算结果数据，用户面对的好像一个“黑匣子”，缺乏可视化的方法来实时监控设计过程的运行状态。而ISIGHT软件不但提供了多种程序结束后分析查看数据的方法，还做到了实时监控运行过程，主要体现在以下几个方面：

1. 在优化设计过程中，提供了以图表的形式直观、实时地展现优化计算过程的探索状况。
2. 多角度把握进程结束后的设计参数和目标函数的变化及相关关系，使问题的特性更加明确。
3. 提供了针对多目标设计问题专门的后处理能力，有效地查看Pareto数据，使复杂数据的分析更加简单。

从模型可以看出，影响网格参数的有几何的形状，个数，材料，额外属性，相当于每一个步长就有 8^4 种组合。如此多的参数无法一一测试，只能利用DOE和优化算法来简化计算。

在实际应用中，优化后的网格优势很明显：即一次可以将网格划分成功，避免了自适应网格的反复迭代计算，在求解精度方面也满足要求。当case 比较简单，这种方法的优势体现不出；如果工程师经验足够丰富，也能达到相同的效果。所以该方法的价值依赖于具体的行业应用。

FEM之理论通俗有限元(2-1)---偏微分方程应用(Matlab/COMSOL)

原创 www.cae-sim.com 多物理场仿真技术



本文介绍分别用Matlab, COMSOL, 以及自己手工 求解拉普拉斯方程。

1. Matlab

命令行输入

1. pdetool

2. 启动界面，工具栏选择画长方形工具，画一个正方形，如图1-1
3. 点击菜单 Options--》Application--》Heat transfer
4. 点击菜单 Boundary--》Boundary Mode，选中图中左边一条边，温度设置为100，选中右边一条边温度设为0.
5. 点击菜单 Solve--Solve PDE 如图1-2

图1-1:

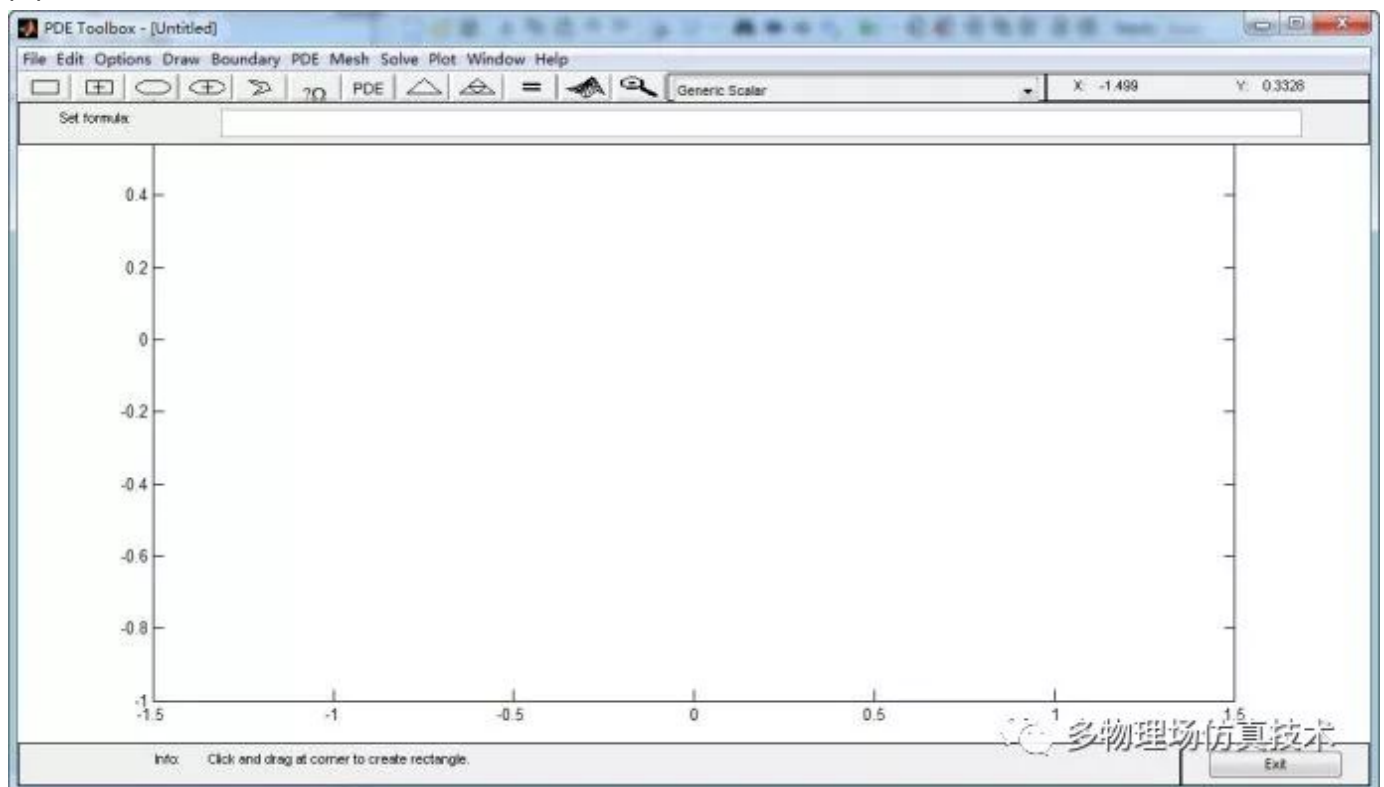
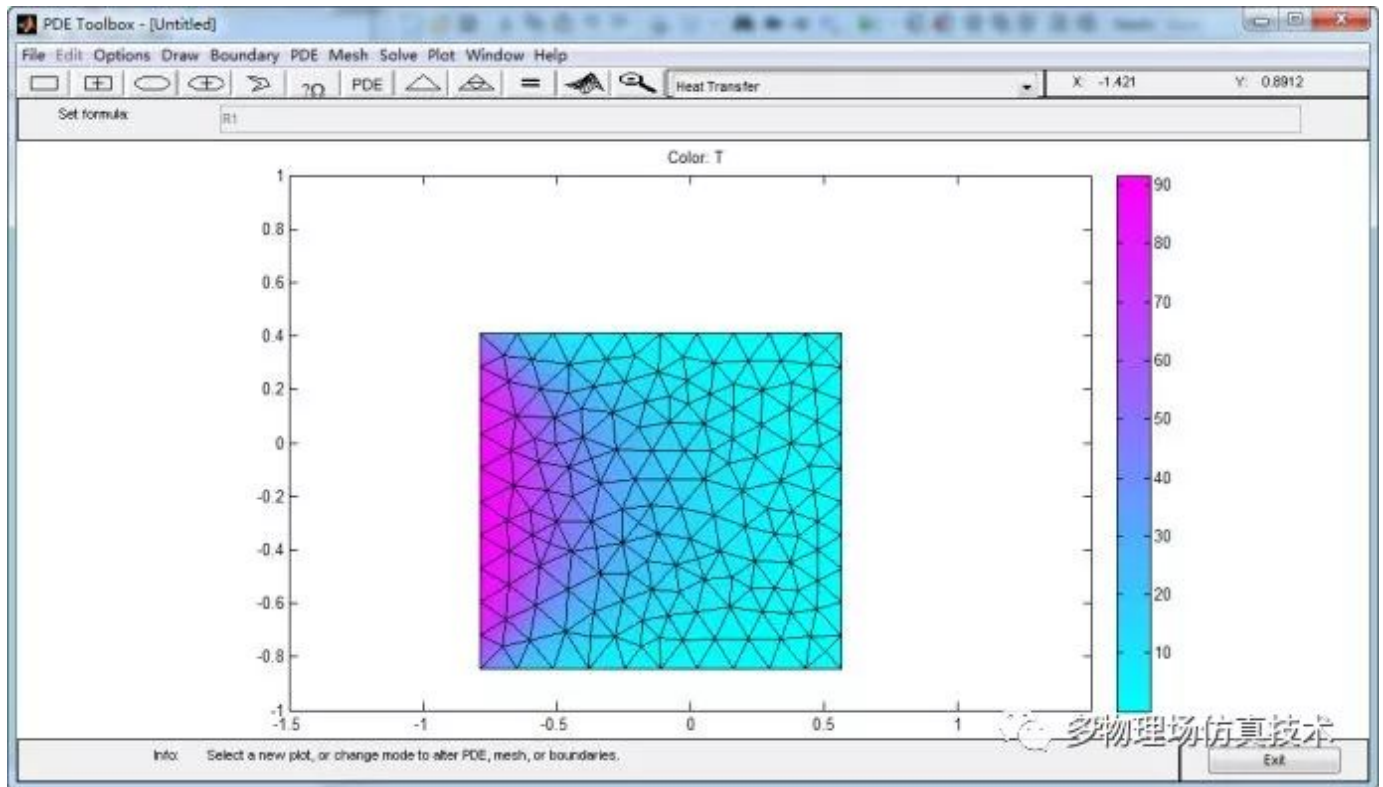


图1-2:



Matlab 解偏微分方程最大的局限性是 只能求解平面不能解三维。

2. COMSOL

1. Space 选择2D

2. 求解类型选择Heat transfer-》Heat transfer in solid, 求解类型选Stationary

3. 选中树形菜单Geometry, 右键, 选择rectangle, 然后画出一个正方形

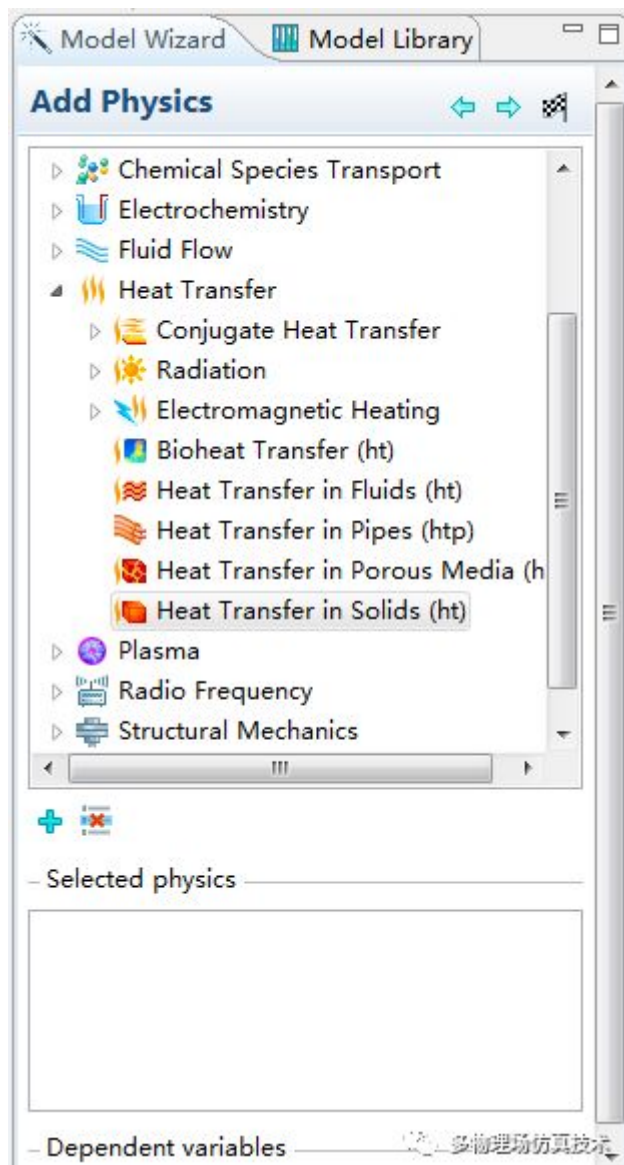
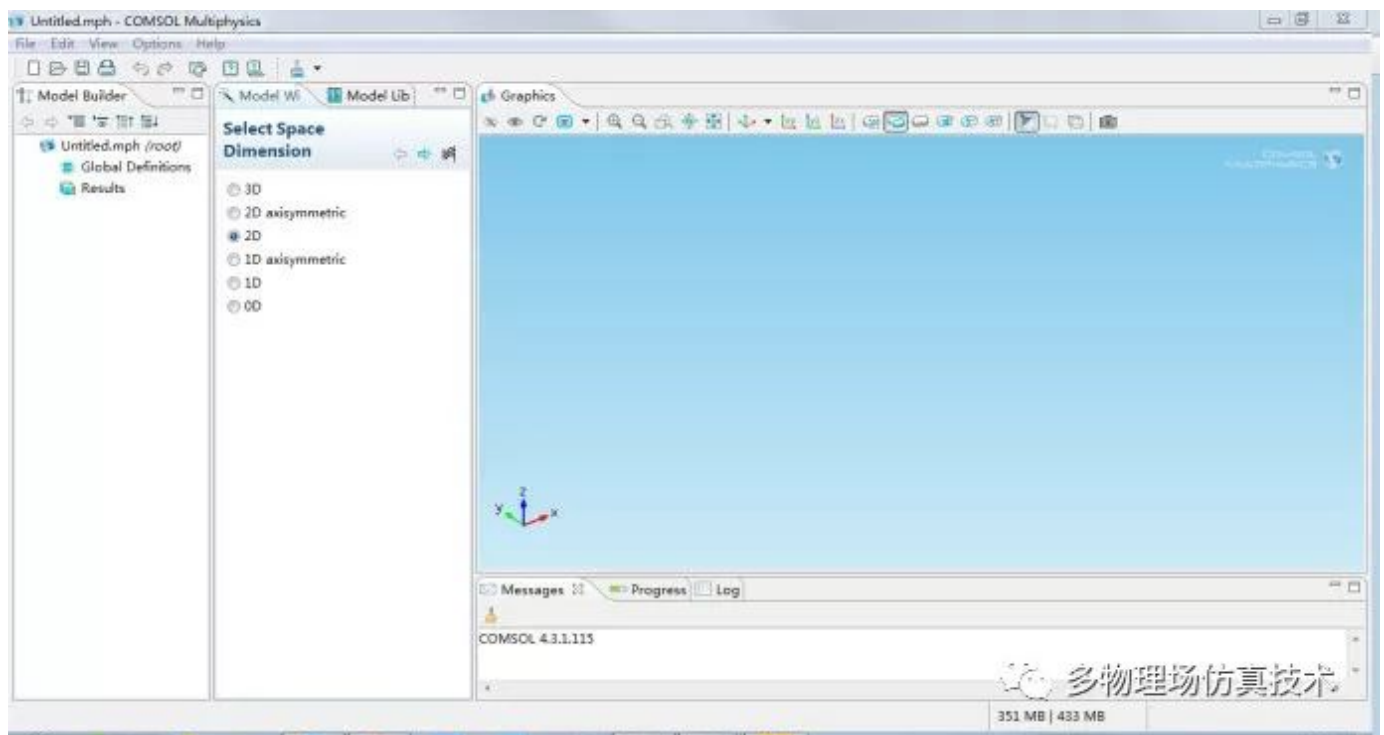
4. 选中树形菜单Material, 选中一个Build-in的材料

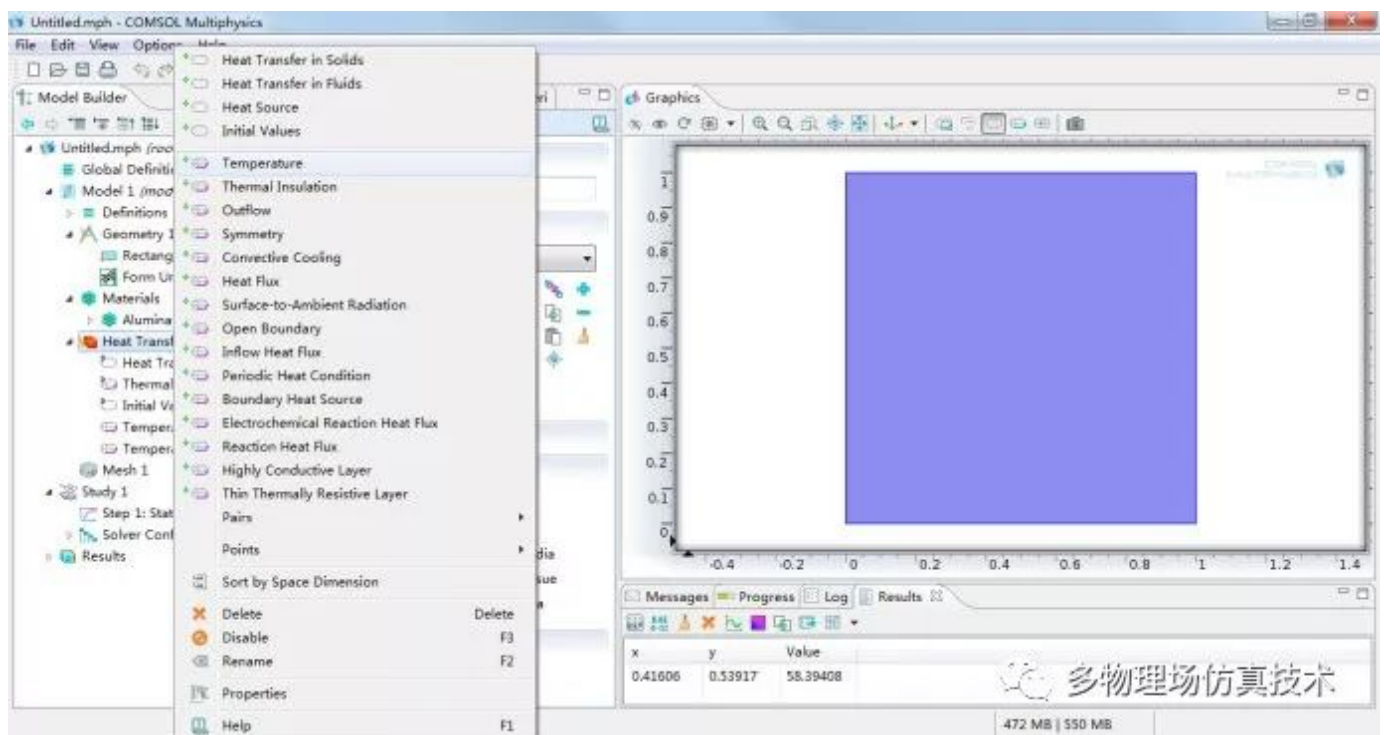
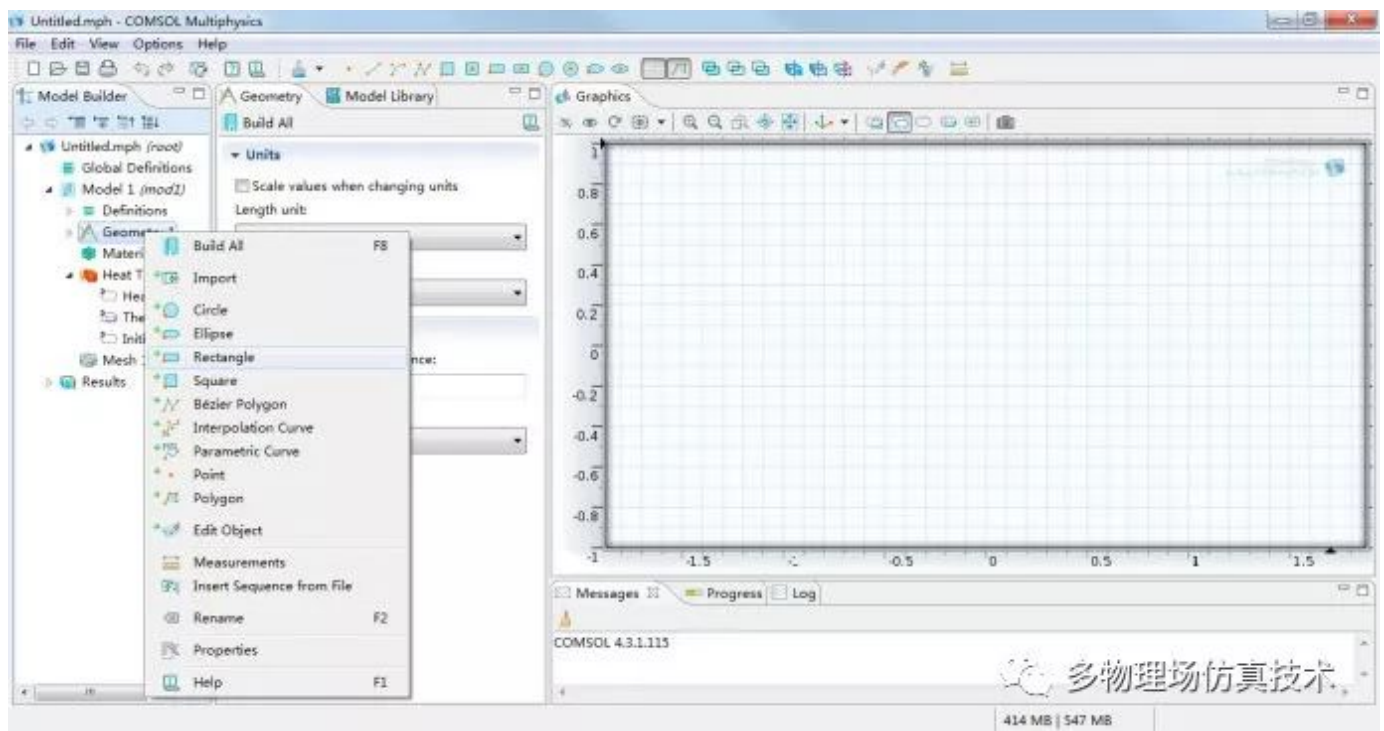
5. 选中树形菜单Heat Transfer in Solids,右键选中 Temperature, 然后选中长方形左边设置100,

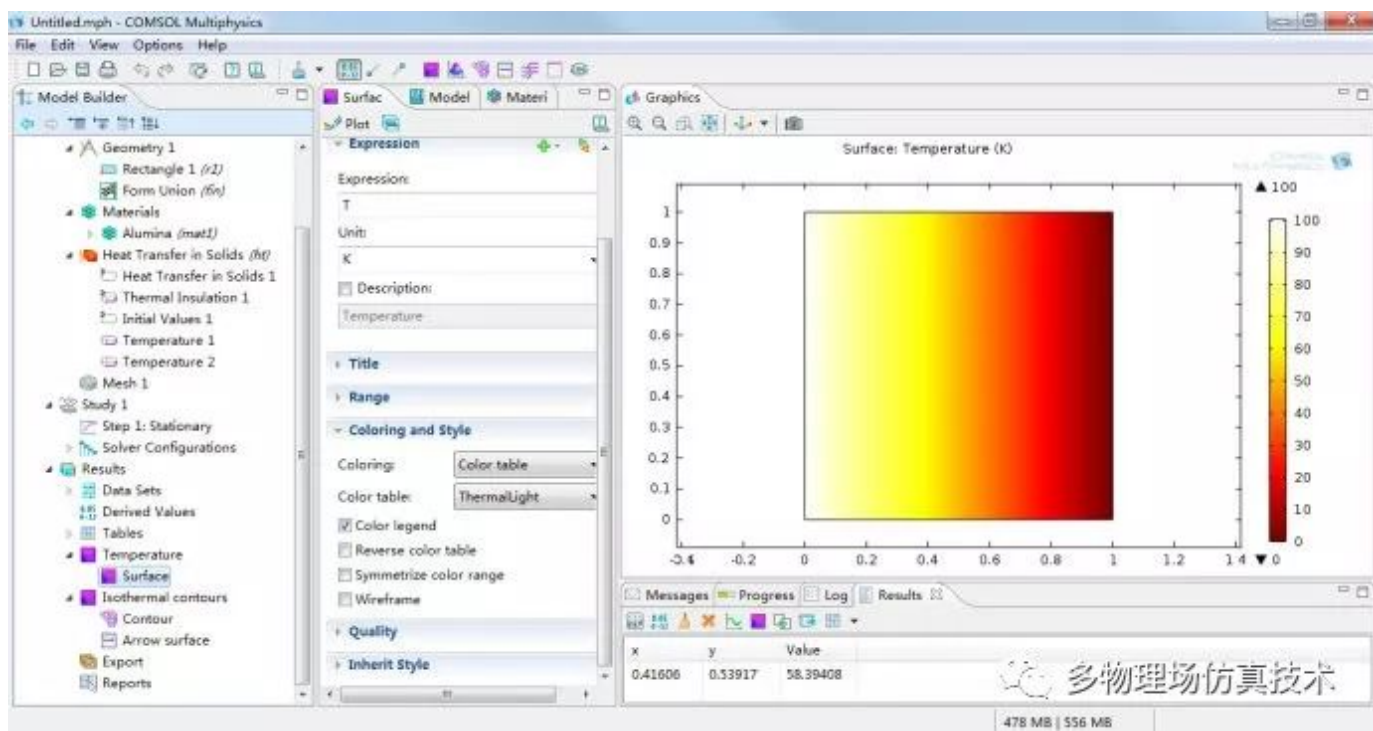
6. 重复5, 选中正方形右边设置0

7. 点击树形菜单Study1, 右键计算 Compute selected step

结果看起来和Matlab 不太一样, 是因为材料属性热导率不同, Matlab 默认边界温度为0。







3. 手工求解

先将正方形离散为三角单元，为三角单元建立形函数，然后利用伽辽金方法推导出积分公式，加入边界，建立可编程实现的有限元方程，求解线性方程，计算出结果。

三角单元离散化参考：FEM之单元(1)---三角单元介绍。

假设三角单元温度场函数为：

$$T = [S1 \ S2 \ S3] [T_i \ T_j \ T_k];$$

$S1 \ S2 \ S3$ 为 线性形函数；

$$S1 = (a1 + b1x + c1y) / 2A;$$

$$S2 = (a2 + b2x + c2y) / 2A;$$

$$S3 = (a3 + b3x + c3y) / 2A;$$

$T_i \ T_j \ T_k$ 为 三角形三个顶点上的温度；

A 为三角形面积；

T 表示了三角单元内部任意一点的温度。

下面最主要的一步是利用伽辽金方法为三角单元建立方程：

$$\int_A S^T \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) dA = 0$$

将形函数与偏微分方程乘积的积分余差为0。

其中 S^T 即为 $[S1 \ S2 \ S3]$ 形函数，其中包含了变量 x, y ，上式可以拆成两个

$$\int_A S^T \left(\frac{\partial^2 T}{\partial x^2} \right) dA + \int_A S^T \left(\frac{\partial^2 T}{\partial y^2} \right) dA = 0$$

对第一个式子进行解析，按照分步积分和链式求导法则，有如下表达式：

$$\frac{\partial}{\partial x} \left(S^T \frac{\partial T}{\partial x} \right) = S^T \frac{\partial^2 T}{\partial x^2} + \frac{\partial S^T}{\partial x} \frac{\partial T}{\partial x}$$

将该式子右边第二项移动到左边，就得到了要计算的表达式

$$\frac{\partial S^T}{\partial x} \frac{\partial T}{\partial x}$$

其中 $\frac{\partial S^T}{\partial x}$ 求导无任何问题，简单的高等数学知识；

$$\frac{\partial}{\partial x} \left(S^T \frac{\partial T}{\partial x} \right)$$

表达式 $\frac{\partial}{\partial x} \left(S^T \frac{\partial T}{\partial x} \right)$ 计算要利用一下 格林函数，将面积分转化成曲线积分，其实也是大学高等数学知识。最终的计算结果可以将积分表达式转化成用a, b, c表现的矩阵形式。

上面一步是很多教科书都没有写明的，一带而过，但这才是用伽辽金方法生成有限元方程的关键所在。

形成矩阵形式以后，就可以很方便的用编程实现了。详细推导过程可以参考《有限元分析--ANSYS理论与应用》P310--P312，作者 Saeed Moaveni，这个系列的都不错，深入浅出，通俗易懂。

下一步就是要将单个矩阵组装成整体矩阵表达式。知道了节点编号，就是对号入座就行了。假设节点数目为N，生成一个N*N的总体矩阵K，按节点编号，将单个单元标号的矩阵放入整体矩阵中，不同单元相同节点号直接相加即可，这表明了单元的连续性。之后就是边界条件，将已知边界的温度生成一个N的向量B，解方程 $Kx=B$ ，即可得到温度场分布。在解决实际问题时，由于矩阵特别大，在总刚组装以及求解线性方程组时会非常讲究，以后有时间讨论。

注：文中所使用的软件仅供学习使用。

一篇文章入门计算流体力学CFD--（上）

原创 www.cae-sim.com 多物理场仿真技术



流体力学是力学体系中的一大分支，计算流体力学（Computational Fluid Dynamic，简称CFD）是用计算机模拟计算流体力学相关内容，属于仿真计算的范畴，也是多物理场仿真中的一个大块头。

CFD内容较多，一个很小的分支就能写成一部专著。本文主要针对CFD软件研发和测试人员，尽可能覆盖CFD商业软件里的术语设置，也可以作为CFD工程师参考，后续还会根据反馈调整，因为篇幅原因，分为上下两部分。

关于OpenFOAM

OpenFOAM是一款开源的CFD软件包，是目前使用最广的开源CFD工具。OpenFOAM在CFD的地位就像是Unix/Linux在操作系统的地位一样。所以如果想要深入学习CFD，了解CFD原理以及CFD软件底层细节，OpenFOAM是个不二选择。当然，OpenFOAM比较庞大，windows支持不太友好，对于初学者来说是一个比较大的挑战。

1. 理论基础--控制方程

CFD 的需要求解的偏微分方程（Partial Differential Equation）为 Navier-Stokers（**纳维斯托克）方程**，简称N-S方程。其它方程比如欧拉方程，雷诺方程是N-S方程的简化。N-S方程由连续性方程，动量方程，能量方程推导而来，其理论基础是基于质量守恒，牛顿第二定律以及能量守恒，推导过程比较复杂范畴，从略。所以N-S是从宏观范畴研究流体，近年来流行的无网格方法比如格子玻尔兹曼则是从分子运动层面解决N-S不容易解决的问题。

完整的N-S方程包括了传导，扩散，对流，压缩（不可压缩），时间等各个要素，但在实际问题中，有些要素是可以不考虑的，所以可以对方程进行各种简化。这也是开源软件OpenFOAM将求解器设计成多个可执行程序的原因之一，一种求解器对应某种特定问题，起了解耦的作用。

FasFluid中有工具对所有求解器进行过滤功能，根据求解特点找到合适的求解器。

2. 数值求解方法

CFD的求解方法还是传统的有限差分，有限元以及有限体积。在商业和开源CFD软件中，采用的都是有限体积方法，主要原因是有限体积方法是在实现难易程度，精度，稳定性，资源消耗等各个方面比较好的折中。

3. 边界条件

PDE边界条件常分为三大类，详细介绍可参考附录文章。所有其它类型的边界只不过是这三种边界条件的细化和衍生。CFD使用的边界条件同样也是常用的三大类。

以下是CFD中常用的几类边界条件：

inlet 入口边界条件

outlet 出口边界条件

wall 墙

prescribed pressure

symmetry 对称边界

periodicity 周期性边界

OpenFOAM针对流体，对边界条件做了进一步细化：

1. 基本类型

基本类型	patch wall ↓	symmetry empty wedge cyclic
------	--	--

常用的几何类型为 patch 和 wall。通常情况下patch为多个面的组合。

2. 主要类型：

mixed 的衍生类型		
inletOutlet	依据速度的方向把速度和压力的边界条件在 fixedValue 和 zeroGradient 之间进行切换	inletValue value
outletInlet	依据速度的方向把速度和压力的边界条件在 fixedValue 和 zeroGradient 之间进行切换	outletValue value
pressureInletOutletVelocity	inletOutlet 和 pressureInletVelocity 的混合	value
pressureDirectedInletVelocity	inletOutlet 和 pressureDirectedInletVelocity 的混合	value inletValue
pressureTransmissive	压力传递边界条件	pInf
supersonicFreeStream	非直角波震压力 p_∞ ，温度 T_∞ ，速度 U_∞ 边界条件	
fixedGradient/zeroGradient 的衍生类型		
fluxCorrectedVelocity	通过进口通量计算速度分量	value
buoyantPressure	依据气压设定 fixedGradient 压力条件	

fixedValue 的衍生类型		需要指定的关键词
movingWallVelocity	改变边界面的法向值以使得通过边界的通量为 0 ⁵⁷	value
pressureInletVelocity	进口压力已知, 速度通过边界的法向通量计算	value
pressureDirectedInletVelocity	进口压力已知, 速度通过边界的 inlet - Direction 方向的通量计算	value inletDirection
surfaceNormalFixedValue	通过固定大小为边界面矢量指定法向边界条件	value
totalPressure	$p_0 = p + 0.5\rho u^2$ 固定, 速度改变的时候, 压力随之改变	p_0
turbulentInlet	通过平均波动尺度计算波动量	多物理场仿真技术 fluctuationScale

类型	描述	需要指定的数据
fixedValue	指定 ϕ 的值	value
fixedGradient	指定 ϕ 的法相梯度	gradient
zeroGradient	ϕ 的法相梯度为 0	——
calculated	ϕ 的值从其他场的值计算而来	——
mixed	依据 valueFraction 的值而变的 fixedValue 和 fixedGradient 混合形式	refValue refGradient valueFraction value
directionMixed	valueFraction 具有方向的 mixed 形式, 例如, 对于法向和切向的混合级数不同的情况	refValue refGradient valueFraction 多物理场仿真技术 value

3. 衍生类型:

其他类型		
slip	对于标量场使用 zeroGradient 条件, 对于矢量场的切向分量为 zeroGradient, 法相分量为 fixedValue	——
partialSlip	依据 valueFraction 的 zeroGradient 和 slip 的混合形式, =0 意味着 slip	多物理场仿真技术

4. 雷诺数/边界层

Re是雷诺数 (Reynolds number) 的数学符号, 一种可用来表征流体流动情况的无量纲数。公式为: $Re = \rho v d / \eta$, 其中 v 、 ρ 、 η 分别为流体的流速、密度与动力黏性系数, d 为一特征长度。雷诺数小, 流体呈层流流动状态; 雷诺数大, 意味着惯性力占主要地位, 流体呈紊流 (也称湍流, Turbulence) 流动状态。一般管道雷诺数 $Re < 2000$ 为层流状态, $Re > 4000$ 为紊流状态, $Re = 2000 \sim 4000$ 为过渡状态。在不同的流动状态下, 流体的运动规律、流速的分布等都是不同的, 因此雷诺数的大小决定了粘性流体的流动特性。

边界层是流体中由于粘性特性而出现的一种现象。如果粘性很小的流体 (如水, 空气等) 在大雷诺数时与物体接触并有相对运动, 则靠近物面的薄流体层因受粘性剪应力而使速度减小; 紧贴物面的流体粘附在物面上, 与物面的相对速度为 0。边界层通常非常“薄”, 网格

划分中需要对边界层进行非常细致的加密才能达到求解精度要求。好的网格引擎有专门的边界层网格加密功能。因为影响到求解精度，边界层的网格也是用户关心的重点。

5. 湍流（紊流，Turbulence）

湍流是流体力学的经典难题之一，由于湍流的非线性和湍流解的不规则性，湍流理论成为流体力学中最困难而又引人入胜的领域。虽然湍流已经研究了一百多年，但是迄今还没有成熟的精确理论，许多基本技术问题得不到理论解释。诺贝尔奖获得者海森堡曾经说到：“我要带着两个问题去见上帝：量子力学和湍流大涡模拟。”

目前湍流的几种主要求解方法，其它也都是在这几种方法上进行改进：

DNS （Direct numerical simulation）直接数值仿真

LES （Large eddy simulation）大涡模拟

RANS 模型（Reynolds-Averaged N-S 方程）

DES （Detached Eddy Simulation ）分离涡模拟

以上几种方法在OpenFOAM都有对应实现。

实际当中每种模型还有更具体的实现模型，比如常见的汽车风阻，k-epsilon模型可以较好的结果。

应用	模型
内流及电子冷却	Wall Treatment models
气动问题（跨声速流动）	RANS模型：Spalart-Allmaras模型
一般问题，复杂几何的外部流动	RANS模型：k-epsilon模型
一般问题，内部流动，射流，大曲率流，分离流	RANS模型：k-w模型
旋风，强旋转以及其他复杂流动	RANS模型：雷诺应力模型
介于层流与湍流之间的流动	各种转换模型
热疲劳，振动，浮力流（船舶设计）	LES模型
外部气动力，气动声学，壁面湍流	DES, DDES, IDDES模型
基础理论研究及湍流模型构建	DNS模拟
分离区域，航天	Scale Adaptive Simulation (SAS模型)

（以上图片来源于网络）。

6. 几何和网格

CFD的仿真计算和其它物理场仿真一样，需要网格数据作为输入；通常CFD计算只需要对象的面几何，也就是说只需要面表达的数据结构就够用，所以几何数据通常小于其它需要三维实体结构的数据。CFD中的几何文件只用*.STL面片格式就能够表达信息，不需要类似ACIS/Parasolid的三维拓扑结构，当然ACIS/Parasolid的Brep结构也可以用于流体计算，只不过会有大量冗余信息；如果需要附加额外属性，比如需要在指定地方加密，可以自定义格式同时建立和几何面片的映射关系。

理想流体的网格为四边形（2D）和六面体（3D），如果几何非常复杂，比如曲面曲线不规则，则需要引入三角形（2D）和四面体（3D），之前文章提到的网格质量标准，自适应网格划分也都适用CFD网格。

动网格是CFD中特有的网格方法，动网格模型可以用来模拟流畅形状由于边界运动而随时间改变的问题。边界的运行形势可以使预先定义的运动，即在计算前指定其速度或角速度；也可以是预先未定义的运动，即边界的运动由前一步计算结果决定。在使用动网格模型时，需要先定义初始网格，边界运动方式并指定参与运动的区域。

一篇文章入门热学有限元

一篇文章入门多物理场有限元（全篇）

深入解析HFSS自适应网格加密策略

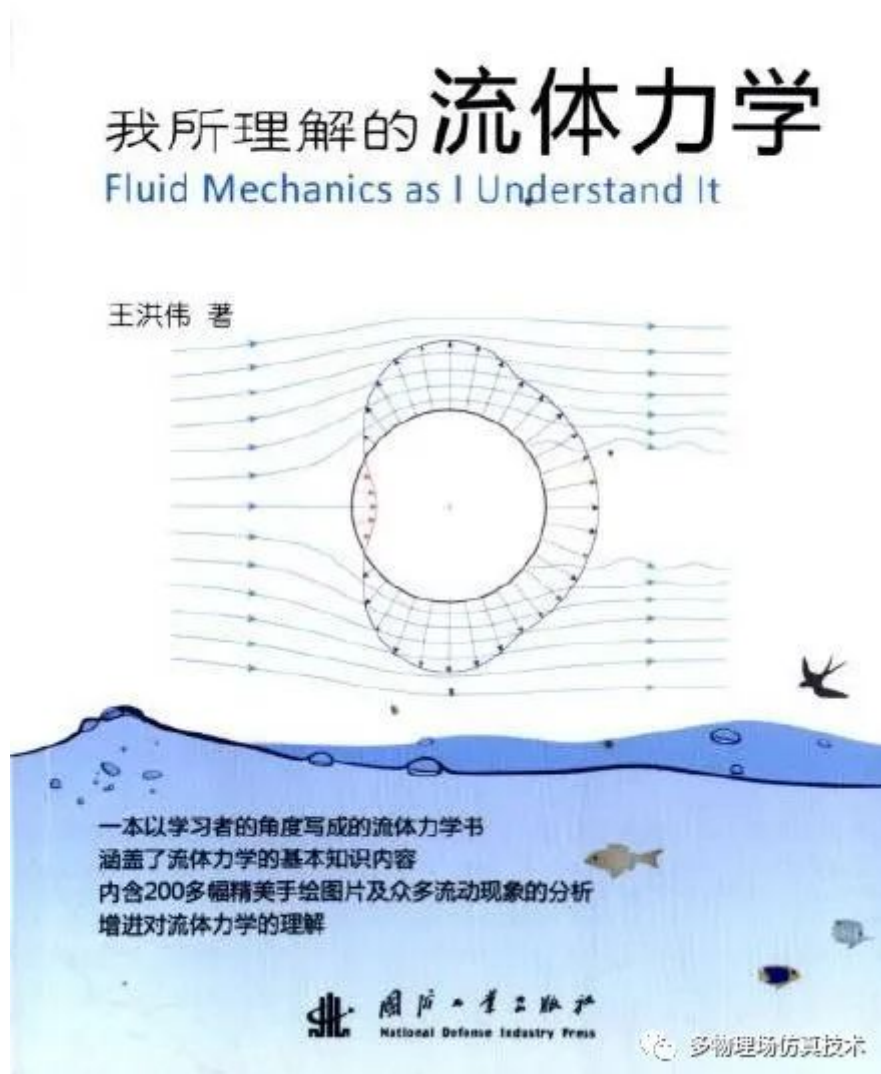
仿真软件网格功能开发

一篇文章入门计算流体力学CFD--（下）

原创 www.cae-sim.com 多物理场仿真技术



推荐一本书《我所理解的流体力学》，这本书包含了流体力学的公式理论推导，基本原理以及很多流体的有趣现象，图文并茂，兼顾专业和科普，是介绍流体力学不可多得的好书。



就在今年2020年4月26号，武汉多位车主反映，武汉鹦鹉洲长江大桥桥体出现可感知的上下晃动现象，该桥管养单位武汉市城投集团公司回应称，此次桥梁异常振动系特定风况引起，振幅在设计允许范围内；桥梁结构运行正常，安全有保障。在“一篇文章入门多物

理场有限元”一文中介绍过美国塔科马大桥早年被风吹垮塌的事故，之后的悬索桥都考虑到了强风对桥梁的影响，进行了CFD相关仿真，所以桥的质量本身是没有问题的，但是因为牺牲了刚度，导致振动位移过大带来的行车安全隐患也不容忽视，想想坐车上下颠簸达一米的时候，真是在坐过山车的感觉。

计算流体力学CFD属于流体力学的一部分，是用数值计算方法求解流体力学问题。在“一篇文章入门计算流体力学CFD上”中介绍了计算流体的一些基本概念，本文再进行一些扩展。

按照流体不同的特点，可以进行分类：

1. 理想流体和粘性流体

粘性(Viscosity)是流体内部发生相对运动而引起的内部相互作用

2. 牛顿流体与非牛顿流体

按照内摩擦剪应力与速度变化率的关系不同，粘性流体又分为牛顿流体和非牛顿流体；

3. 可压缩流体和不可压缩流体

例如空气可压缩，水不可压缩

4. 定常和非定常流动

许多流体机械在启动或者关机时候可以看做非定常流动，而正常运转时可以看做定常流动

5. 层流和湍流

可以通过雷诺系数来确定

有限体积法的离散格式：

在使用有限体积法建立离散方程时，很重要的一步是将控制体积界面上的物理量以及导数通过节点物理量插值求出，引入插值方式的目的是即使为了建立离散方程，常用的离散方式有：

中心差分格式，一阶迎风格式，混合格式，指数格式，迎风格式，二阶迎风格式，QUICK格式，压力修正法

SIMPLE不是simple

SIMPLE算法是流体中经常使用的离散算法，全名为压力耦合方程组的半隐式方法 (Semi-Implicit Method for Pressure Linked Equations)，是CFD中一种被广泛使用的求解流场的数值方法，于1972年由苏哈斯·帕坦卡与布莱恩·斯波尔丁提出。

SIMPLE算法自1972年问世以来在世界各国计算流体力学及计算传热学界得到了广泛的应用，这种算法提出不久很快就成为计算不可压流场的主要方法，随后这一算法以及其后的各种改进方案成功的推广到可压缩流场计算中，已成为一种可以计算任何流速的流动的数值方法。

改进的SIMPLE方法包括SIMPLEC (SIMPLE-Consistent) 算法，SIMPLER (SIMPLE revised) 和PISO (Pressure implicit with splitting of operators)

CFD的一些其他概念

多相流/组分传输与气体燃烧/多孔介质

物质一般具有气态，液态，固态。在通常所指的多相流动中，相可以定义为具有相同类别的物质，该类物质在所处的流动中具有特定的惯性相应并与流畅相互作用。目前有两种处理多相流的数值计算方法：欧拉-欧拉方程，欧拉-拉格朗日方程。

多物理场耦合

FSI (fluid-structure interaction) 流固耦合是CFD多物理场耦合最常见的耦合方式。CFD和热耦合也是常见的耦合方式，比如电子器件散热。

CFD/有限元/无网格

之前提到大部分商业软件以及OpenFOAM采用的都是有限体积法，少用有限元方法。其实有限元精度比有限体积更高，只是有限体积不仅满足计算要求，而且算法简单，在目前硬件软件相同条件下，有限元并不比有限体积更有优势。基于无网格的CFD发展迅速，实际应用也越来越成熟，未来无网格法和有限元计算CFD应该是趋势。

介绍CFD可是说是研发仿真“一篇文章入门系列”中最痛苦的事，CFD内容实在太多太复杂了。笔者开发OpenFOAM前后处理工具FasFluid的时候，也是被绕的云里雾里。文章中很多一句话基本上都是几本书的内容，仅供参考索引。

一篇文章入门计算流体力学CFD-- (上)

FasFluid入门教程(7)---赛车CFD分析

一篇文章入门多物理场有限元 (全篇)

推荐一款开源有限元结构仿真工具

原创 www.cae-sim.com 多物理场仿真技术



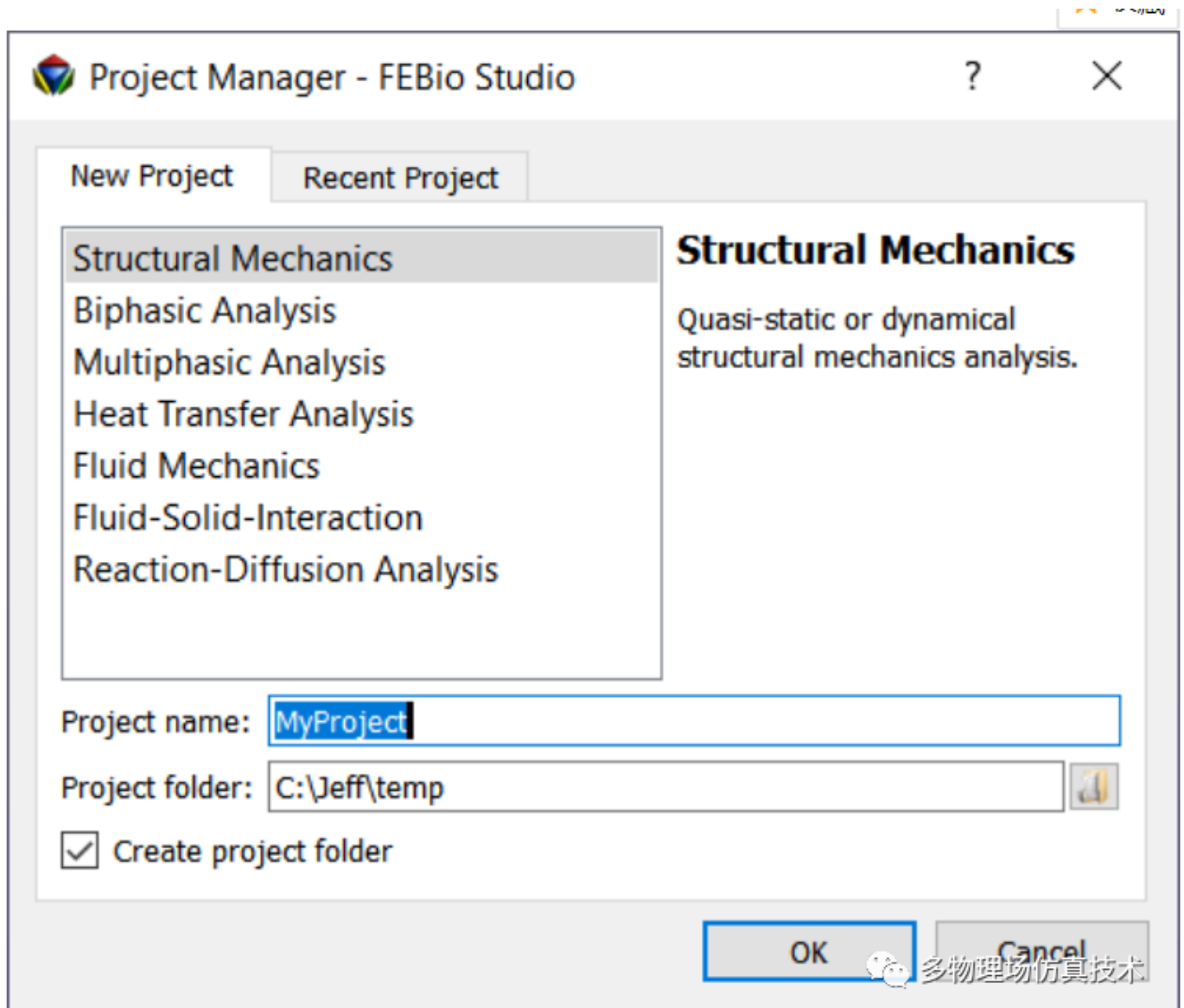
先上地址：

<https://febio.org/>

笔者大概在2012年左右接触到该软件，去年看的时候该软件还在维护开发，每年保持不小的commit。

总的来说，相比其它开源软件，febio的几大特点：

1. 代码C++编写，CMAKE可跨平台编译；
2. 工程化方面一直在改进，和COMSOL，LS DYNA，ANSYS 有部分接口；
3. 有自己的前后处理器！
4. 网格划分使用tetgen
5. 有自己的几何处理数据
6. 偏向于医疗器械生物仿真
7. 求解类型



当然，缺点也有：

1. 有些计算不靠谱，准确性性能方面还有很大提升空间；
2. 几何处理外部接口少，很难处理大和通用几何；
3. 工程化方面还有很多事情要做

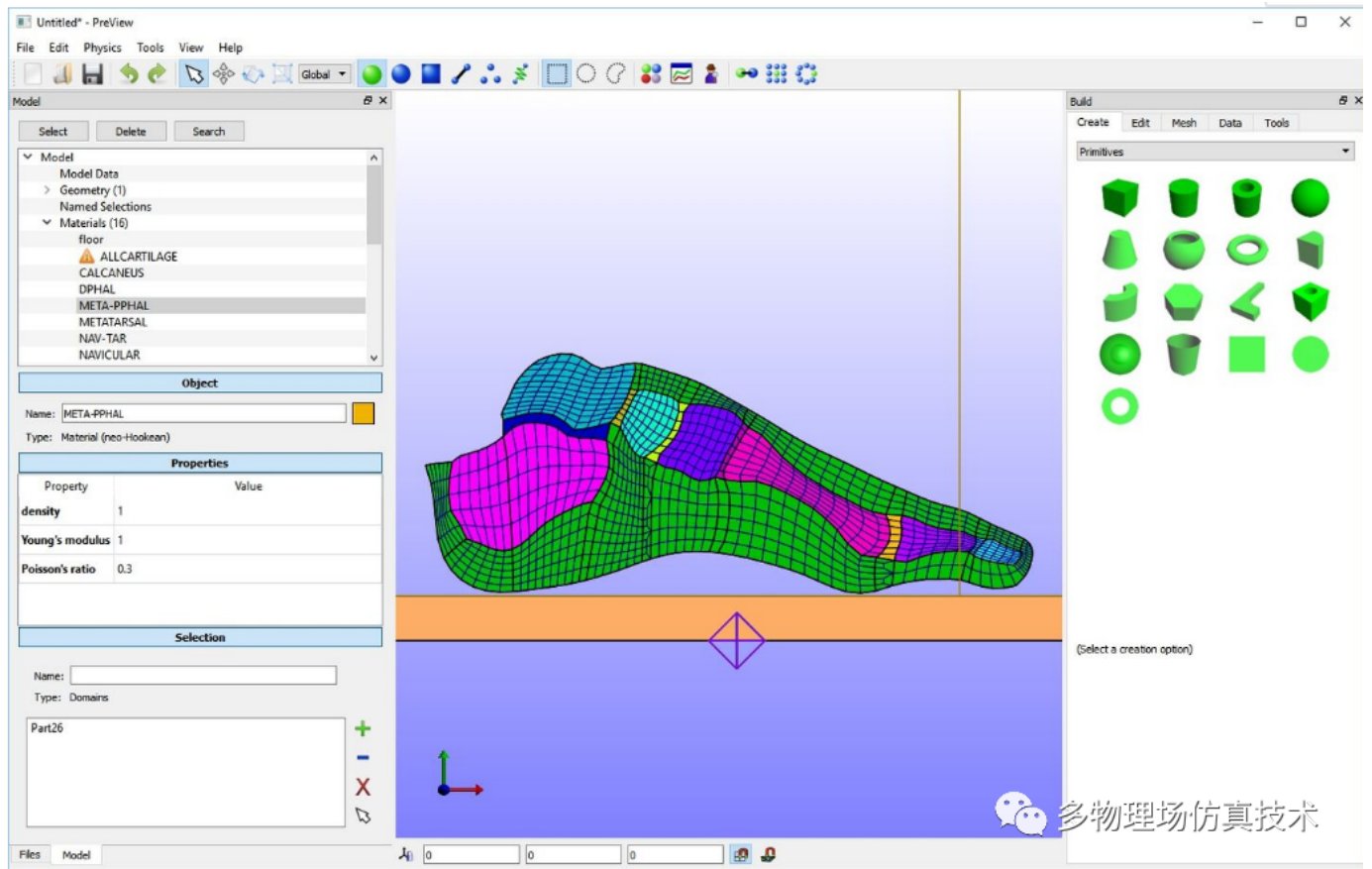
一般来讲，仿真软件开发是一个团体项目，包括求解器，前后处理，第三方面接口等，而 febio 几乎都能提供相对相对友好的支持，降低了开发复杂程度。所以从开发角度来看，febio 是一款优秀的开源结构仿真工具，特别对于想了解软件底层技术，以及想学习仿真软件架构设计，是一个相当不错的选择。

FEM 开源软件推荐

FDTD 开源软件推荐

再聊开源软件

FEM之求解器---开源求解器简介



一篇文章入门多重网格方法

原创 邓子平 多物理场仿真技术



很多人第一次看到“多重网格”的时候，可能认为是有限元方法的网格划分方法。实际上多重网格方法是一种求解大规模线性方程组的数值计算方法，和有限元方法里的网格是两码事。

求解线性方程组通常有直接法和迭代法，直接法主要以求解逆矩阵为主，精确解，在小规模（自由度小于10万）比较常用，而大规模的方程组（自由度100万-1000万）通常适用迭代方法，一般给出一个初始值和迭代矩阵，在此基础上进行迭代计算，达到收敛标准为止。而对于超大规模（自由度1000万以上）的方程组，对硬件要求较高，普遍使用模型降阶方法，目前工程领域主要使用Krylov子空间方法（之前有过介绍）和多重网格方法，尤其是Krylov方法很容易进行并行和分布式计算。

多重网格方法属于迭代方法，具有迭代方法的一系列特点，在求解大规模稀疏对称正定线性方程组时，效率要高出普通迭代方法很多。

之所以叫多重网格，是因为该方法最初应用在有限差分方法上，有限元和有限差分在网格上的区别在于：有限差分的网格是结构化网格，即横平竖直的均匀网格，网格是平铺在对象上；而有限元方法则是把对象进行离散化。所以缩小线性方程组规模时，简单跨过中点，恢复时插值取其两点之间点即可，这也是多重网格的最初由来。这种最初的取点方法由于依赖于几何空间数据，所以又叫几何多重网格(Geometrical Multi Grid, GMG)，后来发展起来的无需几何空间数据，即纯粹求解线性方程组的方法，又叫代数多重网格(Algebraic Multi Grid, AMG)，很明显AMG在处理线性方程组迭代矩阵时要比GMG复杂。

多重网格方法理论基础复杂，翻遍网络，找不出一个实际应用的例子。后面有空拿一个具体求解例子来说明问题。

在笔者看来，解决工程领域的性能问题简单归结两个原则：二八原则和分治原则。即性能瓶颈普遍存在20%的地方，而解决方法即将问题分开，集中处理这20%的问题。

在一篇文章入门仿真软件性能优化一文中介绍过改进工业软件性能的方法和思想，多重网格方法属于其中的分治法，或者也属于模型降解的一种。其核心思想是在原有的线性方程组基础上构造迭代矩阵，缩小线性方程组的规模，其目的是加快某些区域的收敛速度，收敛到一定程度后再插值恢复到原有的规模。

开源工具推荐两个：

AMGCL

<https://github.com/ddemidov/amgcl.git>

HYPRE：

<https://computing.llnl.gov/projects/hypre-scalable-linear-solvers-multigrid-methods>

近两年工业仿真软件公司并购

原创 邓子平 多物理场仿真技术



在 仿真软件十年回顾和展望(整理版)

一文中总结了2008到2019年十年间仿真软件公司的并购情况，近两年，虽然全球疫情严重，但并没有阻碍工业仿真软件领域的并购，尤其在底层工具领域，几家大公司反而加大了收购力度。

-----ANSYS-----

2021年收购航天数据服务提供商AGI（Analytical Graphics Inc.），AGI为为航空航天、电信以及国防和情报应用提供仿真和建模服务。

2019年9月收购LSTC公司（LSDYNA旗舰产品）的最终协议阶段，其为显式动力学和其它高级有限元分析技术的领先供应商。收购价约7.75亿美元。

2021年收购Phoenix Integration公司，该公司是一家领先的基于模型工程（MBE）和基于模型系统工程（MBSE）的软件提供商

-----Techsoft3D-----

2020年10月和11月，Techsoft3D（HOOPS旗舰产品）
分别收购了CeetronAS和VKI两家公司

-----Altair -----

2019年10月，收购 EDA (电子设计自动化) 软件公司Polliwog Co. Ltd.

2019 年，收购英国DEM Solutions公司，其旗下产品EDEM是散体物料仿真领域离散元方法(DEM)技术的市场领导者。

2020年初收购newFASANT。该公司拥有计算电磁学和高频电磁学的领先技术，在天线设计与天线布局、雷达散射截面（RCS）分析、车车通信（V2V）/汽车自动驾驶（ADAS）、红外成像/热特征分析等领域有突出贡献。

2020 年 5 月 7 日，收购瑞典公司 WRAP International AB 的瑞谱 (WRAP) 软件业务，瑞谱 (WRAP) 是一项世界领先软件技术，用于民用和国防无线电管理机构的频谱管理和无线网络规划。

2020 年 7 月 27 日，收购首尔领先的聚氨酯发泡仿真技术供应商S&WISE

2021年收购加拿大S-Frame工程软件公司

-----Cadence-----

Cadence在2021年收购了Numecia 和 Pointwise

-----Synopsys-----

2019年，收购汽车软件与仿真公司QTronic GmbH

2020年11月19日，收购了光散射测量和测量设备的全球供应商LightTec，Synopsys的光学设计软件工具与LightTec解决方案的结合扩大了客户获取用于光学系统的材料和介质的精确光散射数据的机会。

2021年收购 Code Dx，Code Dx 是一家屡获殊荣的应用安全风险管理解决方案提供商，可自动识别软件漏洞、更早发现漏洞，并进行优先排序和修复。

-----Dassault-----

2019年6月12日，收购美国健康护理软件公司MedidataSolutions (MDSO.US)，交易价值约58亿美元。创造有史以来最高收购价格，超过了西门子2006年并购UGS的35亿美元，和2016年并购Mentor公司的45亿美元。

2020年，达索低调收购了网格引擎公司MeshGem。

-----SIEMENS-----

2018年以后西门子的收购也是丝毫没有停下，公司太多不再表述。

参考文章

关于工业仿真软件中的图形显示引擎

原创 邓子平 多物理场仿真技术



有好几个读者建议写写工业仿真软件中的图形显示引擎，翻了写过的几百篇文章，除了一篇介绍HOOPS和VTK的文章，没有其它。

笔者用过VTK，OpenGL，HOOPS，OSG开发过产品，从实际应用情况看，各有各的优点，都能满足实际需求，倒不是说图形引擎并不重要或者简单，**而是从工业软件实际需求看**，图形渲染这块没有太多技术瓶颈，在工业软件研发中属于偏上层的应用。工业软件没有游戏电影中那些顶级特效的需求，只要想做，基本都能做出来。

如果非要做个总结，笔者给的建议如下：

1. 有自己的非常核心技术，投资不缺钱，希望快速完成产品上市，而且对产品要求质量高。推荐HOOPS。HOOPS商用，效果功能都不错。
2. 属于中规中矩的开发，有比较了解图形学的开发者，可以使用VTK
3. 产品开发周期长，希望有独立的产品技术积累，推荐OpenGL。

其它开发引擎都不太推荐直接使用，当然像OSG，OGRE有不少值得学习和借鉴的方法和思想。

对于Vulkan，也可以多留意，虽然工业软件的开发对新技术并不是特别敏感，但是从历史看，一项新技术如果发展成熟，几乎肯定是会采用的。

另外，HOOPS和VTK都隶属于美国公司，开发需要注意政策上的风险。

笔者早年在研发中曾经接手过国外的一个仿真后处理图形显示项目，该项目的源码采用JAVA编写，在渲染这块没有使用任何第三方库（你没听错，画线都是使用的最原始的bresenham算法），像面片渲染，云图显示都是自己实现算法。

其实图形引擎深耕，也有很多值得研究内容，比如体渲染，高保真显示，光线追踪，大规模场景显示，VR，AR，机器视觉等等，有兴趣的朋友可自行研究，后续文章将不再涉及图形显示方面的内容。

仿真软件开发工具介绍(2)---显示引擎VTK/OSG

仿真软件开发工具介绍（1）--ACIS/HOOPS

说说Mesh Fusion功能

邓子平 多物理场仿真技术



有朋友私信让讲讲ANSYS HFSS产品最新的Mesh Fusion技术。

网上查看了一下相关资料和视频，ANSYS说这是近十年来HFSS最大的技术创新。ANSYS2008年收购的Ansoft，也就是收购以来最大的技术创新，而ANSYS的工程师也说这是HFSS有史以来最大的一项技术突破。更有国内工程师说，这种学术界都不敢碰的技术难题被HFSS解决了。

看了Mesh Fusion的功能介绍，一句话概括：就是网格之间进行**布尔并**操作！对于前处理功能来说，这并不是什么新功能，技术实现没有太多难点，稍后详细介绍。

但是对于一款拥有网格布尔并的前处理和独立求解器的产品来讲，可以解决很多工程问题：

1. 关于数值仿真模型的优化，分治是一大难点，分治在一篇文章入门仿真软件性能优化一文中详细介绍；

利用网格并功能，大的几何模型可以分开独立划分网格，利用分治支持并行计算，尤其对于PCB，SoC，连接器等电路问题分析，大大降低了模型之间的耦合。

2. 帮助解决几何不兼容的问题，如果模型之间接口有问题，几何会不兼容，而**网格布尔并**则避开了这个问题；

3. 容易支持解决多尺度问题。通常多尺度问题的模型在同一坐标空间中划分网格非常容易出现数值精度误差；比如手机中的芯片和建筑统一模型中分析电磁波；

两个**网格布尔并**运算算法流程基本如下

1. 确定两个模型相交区域，这个信息可以通过几何模型自动传导；

2. 分别对两个模型进行网格划分；

3. 在相交区域进行网格并运算，运算也分几种情况。

面面之间完全重合，只需简单把一个模型网格映射到另一个上，在另一个面上作为硬点和硬边即可。

如果不重合则涉及到两个不同面网格之间建立映射，其中有一些规则，比如网格依小不依大；重合和不重合部分网格尺寸尽量一致；映射完成后需要调整网格质量保证平滑过渡；数值精度上要保证完全重合。

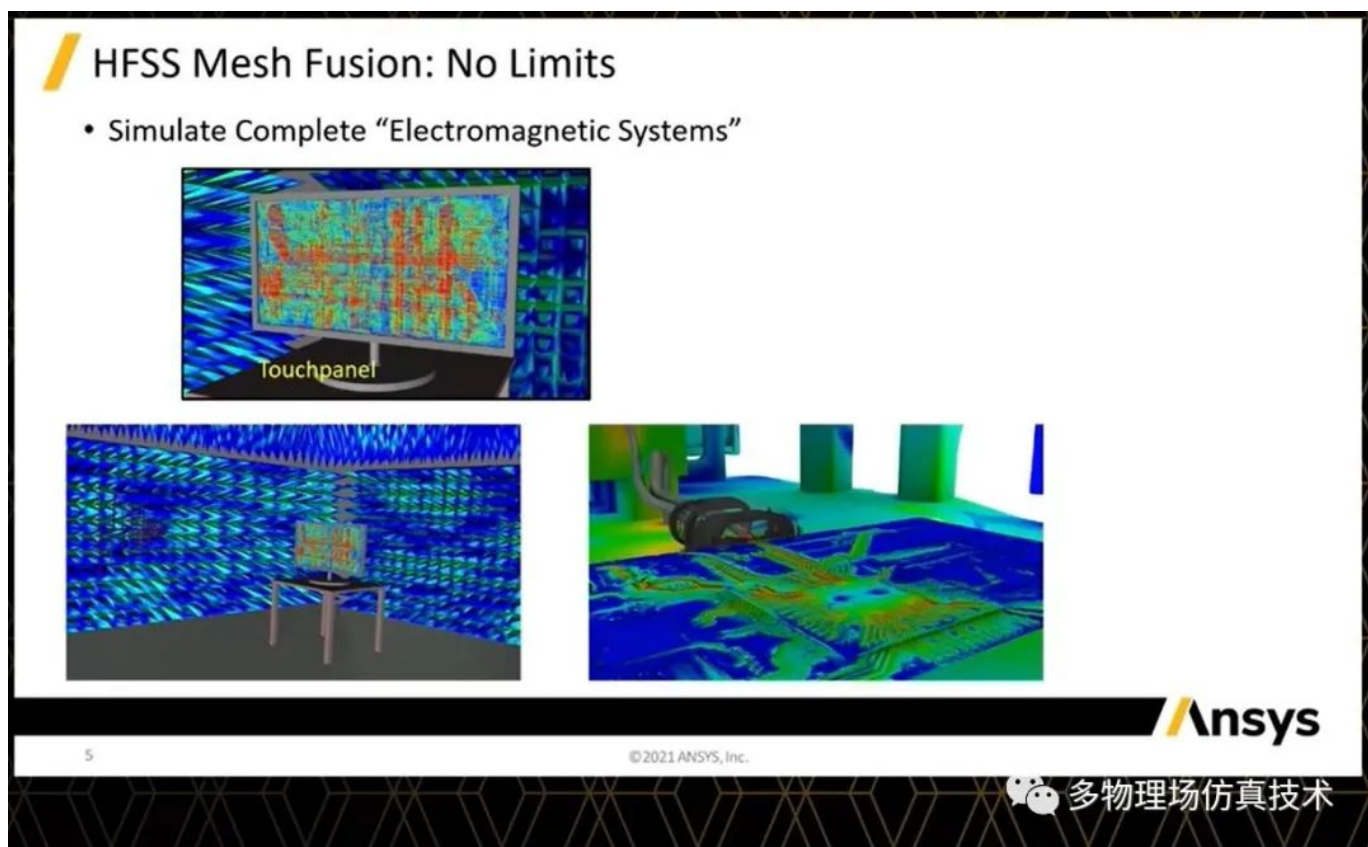
如果接触面不是单独平面而是多个不同面，则涉及到每个面建立映射以及不同平面建立映射。通常电路模型接口都是平面，少有曲线曲面，即使有曲面处理逻辑也和平面类似。

HFSS的计算网格是四面体，截面都是三角形，三角形的计算操作比较成熟，在现已有网格算法框架内都可以解决，参考如下两篇文章：

深入理解数值计算网格(4)--万能的四面体

深入理解数值计算网格(3)--非结构化网格生成算法

总的来讲，**网格布尔**并能帮助解决一些工程问题，并加速问题求解。但过分强调这种技术点是否也在表明大厂的创新前进动力不足？按照笔者的理解，软件计算效率跟上乃至超过摩尔定律才是王道。



“一亿”是“一千万”的十倍吗？

原创 邓子平 多物理场仿真技术



笔者在 一篇文章入门多物理场有限元（全篇）一文中提出过该问题。以下是原文内容：

-----1亿是1千万的10倍？-----

早期笔者使用矩量法求解线性方程组，在不使用快速多级方法，自由度达到3万时，台式机上已经无法求解出，8G的机器内存不够用。对于满秩矩阵的线性方程组，常规求解方法时间复杂度为 n^3 (n 的3次方)。

对于自由度1千万以下稀疏矩阵的求解，好的台式机基本能应付，而当自由度达到1亿的时候，简单的将硬件乘以10倍完全不能满足要求。因为计算的空间复杂度，时间复杂度并不是线性，通常是 $N\log N$ ， N^2 或者更高。当自由度达到1亿时，不仅需要对硬件核心部件CPU，内存扩容，而且在磁盘阵列，I/O，并发计算，GPU，网络，带宽等方面都提出了更苛刻的要求。

在算法方面看，减少网格密度，在物理量梯度大的地方加密，无变化的地方将网格变稀疏，可以有效减少计算量；另外优化求解算法本身，使其更加易于并行化计算。

从目前来看，计算机硬件计算能力的更新速度，跟不上指数级求解规模的增加速度。这也是量子力学发展的最大推动力！

针对该问题，本文稍微展开一下。

这里的1亿，主要指仿真中最终求解线性方程组的自由度个数，就是刚度矩阵的规模，即在方程 $Kx=b$ 中， K 的规模为1亿*1亿的方阵。

从硬件角度看，性能瓶颈首先在内存。目前台式机的内存普遍在64G以下。对于1千万的稀疏矩阵，存储勉强够用，加上计算，一般都不够用，在windows系统上如果内存不够用，会启用虚拟内存，也就是用页面交换把硬盘当内存使用，频繁读写磁盘，这时候性能会大大降低。而对于1亿自由度的矩阵，可能刚度矩阵组装都不够用。对于一亿自由度的矩阵，首先要保证内存能够存储，一般的方法是使用具有超大内存的工作站服务器。

内存问题解决后，再就是CPU性能。前面讲过，通常矩阵计算的空间复杂度，时间复杂度并不是线性，而是 $N\log N$ ， N^2 或者更高。使用单核单线程，耗费的时间是条曲线上

升趋势。虽然稀疏矩阵采用了CSR等方法压缩数据，但自由度的平方数据仍然还是海量数据。现在的CPU普遍多核多个逻辑线程。这也要求在进行迭代法求解方程组时，合理的进行矩阵分解，参考一篇文章入门仿真软件性能优化，采用分治的方法适应硬件的需求。这里需要注意的是单核CPU的性能以及CPU的数量。根据实际情况进行分治！

在满足了单机CPU运行之后，再就是集群分布式计算。采用多台计算设备，比如刀片等专业服务器，超算等等，这里需要考量的是负载均衡和网络带宽，即保证单台设备的利用率和整体进度保持一致。对于超大量的计算，最好能软件实现动态规划，对于频繁交换数据的计算，要注意网络带宽的瓶颈。

再从软件方面来看，根据笔者的研发经验，相当一部分仿真的性能问题可以通过软件以及改进算法解决。在前处理中，尽量避免软件中高频操作，大数据拷贝，大量细小数据操作等。如果必须有，对操作进行采样性能评估，保证在可接受范围内。**在求解线性方程组前，提取矩阵特征，尽可能在最细度上根据矩阵特征，同时结合模型和业务的特点，选取最合适的求解方法。**

其实目前对于数值计算方法而言，其算法都是公开透明的，商业软件之间真正较量的是业务模型的准确性，算法的稳定性以及效率。而这也恰恰是每个软件可以改进和提升的地方

2004年，ANSYS解出了1亿自由度的模型，2008年，ANSYS解出了10亿自由度的模型。即使现在来看，我们认为自由度超过1千万的模型是个大模型，所以1亿自由度的模型仍然可以认为是超大模型，且只能在超大的服务器上运行，可以想象，随着硬件的提升和算法的改进，实际应用中会有越来越多的超1亿自由度的模型求解。

回到开始的问题，“一亿”是“一千万”的十倍吗？显然不是。

最后还是以业内的一句话作为结束语：

“如果你的仿真还没有受到硬件的限制，那说明你的仿真还没有入门”。