

序

言

从2013年开始到2021年年底，公众号“多物理场仿真技术”围绕多物理场仿真研发技术，涵盖了CAD CAE CFD EDA HPC CAO TCAD等领域，文章累计达到400余篇，绝大部分为原创的和技术相关的文章。

国内工业软件技术薄弱，市场几乎被国外产品垄断，写技术文章的初衷也是想尽个人微薄之力推动国内工业软件研发技术的发展。

文章内容全部选自公众号“多物理场仿真技术”，没有内容但是给出链接和标题的文章都可以在公众号中搜索到。

很多文章内容来自于作者研发和工作经历，如有错误或缺点，还望指正，其中不少文章写作时间较早，如今回头看有些观点不一定正确，还望多多包涵。

关于笔者，15年工业软件研发经历，现仍然从事工业软件研发，技术咨询工作。

联系方式：

Email: fasbridge@foxmail.com

QQ: 2412004802

欢迎访问网站：

www.cae-sim.com

邓子平

记 录

1.仿真软件十年回顾和展望(整理版).....	3
2.深入理解数值计算网格(全篇).....	11
3.工业软件“断供”是一把“达摩克利斯剑”吗.....	46
4.写给仿真软件研发的“一篇文章入门”系列（终）	48
5.我所理解的“工业软件”.....	51
6.中国工业仿真软件，缘何不能“弯道超车”（更新）	55
7.工业仿真软件可以和支付宝一样快吗？	60
8.中国没有世界级的工业软件是一种耻辱吗？	62
9.给工业仿真软件研发打分.....	64
10.工业仿真软件的基础技术.....	67
11.工业软件研发有多“难”.....	69
12.通俗有限元方法理论介绍.....	72
13.仿真软件网格功能开发.....	81
14.CAE 软件研发的一些思考(1).....	87
15.FEA 精度之网格加密(3)--HFSS 网格质量分析.....	90
16.SimSolid 技术分析.....	94
17.深入解析 HFSS 自适应网格加密策略.....	96
18.一篇文章入门三维几何内核.....	98
19.（非）线性方程组求解库大全.....	103
20.一篇文章入门多物理场有限元（全篇）	107
21.大规模线性方程组解法简介.....	117
22.一篇文章入门求解器开发设计模式.....	121
23.一篇文章入门求解器模型降阶方法.....	123
24.一篇文章入门时域有限差分方法（FDTD）	126
25.一篇文章了解云端仿真.....	131
26.FEM 之在求解器中使用设计模式(7)---属性模式.....	134
27.一篇文章入门无网格方法(1).....	137
28.一篇文章入门计算机辅助优化.....	139
29.关于工业仿真软件中的图形显示引擎.....	147
30.深入剖析三维几何内核(2)--功能介绍.....	149
31.CAE 软件研发的一些思考(3)--CAE 软件架构设计.....	153
32.从技术角度聊聊“系统工程”.....	162
33.一篇文章入门前处理器开发.....	164
34.开发一个 ABAQUS 多少钱？	171
35.少一点“国产自主”，多一点“合作共赢”.....	172
36.一篇文章了解计算机辅助工程 CAE.....	174
37.哈工大被禁用 MATLAB 传递出什么信号.....	179
38.仿真软件开发工具介绍大全（1/29/2021 更新）	181
39.如何构建世界级的中国工业仿真软件.....	184
40.一些常用软件的几何内核.....	188
41.工业仿真软件研发底层技术介绍系列.....	190
42.CAE 软件研发的一些思考(5)--系统的开发求解器.....	193
43.随笔：系统的开发工业仿真软件.....	196

仿真软件十年回顾和展望(整理版)

原创 www.cae-sim.com 多物理场仿真技术



这篇文章写于2019年7月，在Ansys收购LS DYNA之前一个月，从技术角度分析了近十年来全球工业仿真软件的发展。鉴于最近关注公众号的朋友数量暴增，老文章再发，老朋友可忽略。

站在2019年的下半年，对仿真行业过去的十年做一个回顾，同时希望能从中找出一些线索和趋势，指引未来十年的发展。

术语

GPU--Graphics Processing Unit 图形处理单元

FEM--Finite Element Method 有限单元方法

FVM--Finite Volume Method 有限体积方法

PDE--Partial Differential Equation 偏微分方程

BEM--Boundary Element Method 边界元方法

DEM--Discrete Element Method 离散元方法

MOM--Method OF Moment 矩量法

EDA --Electronics Design of Automation 电子设计自动化

CAD --Computer Aided Design 电脑辅助设计

CAE --Computer Aided Engineering 电脑辅助工程

CAM-- Computer Aided Manufacture 电脑辅助加工

CAPP --Computer Aided Process Planning 电脑辅助工艺规划

AI -- Artificial Intelligence 人工智能

ML --Machine Learning 机器学习

FPGA --Field Programmable Gate Array 可编程门阵列 多物理场仿真技术

公司收购最能反映技术发展趋势，我们按照被收购公司类型，回顾一下过去十年，仿真领域比较大的并购事件。

一.综合解决方案

西门子于2012年11月12日宣布以6.8亿欧元收购了比利时软件公司LMS International。LMS公司是汽车、航空航天和其它先进制造业公司的工程创新合作伙伴。LMS公司可以帮助用户将更好的产品更快地投入市场，并将卓越的技术和效率转化为其战略竞争优势。

二.流体

1.2011年，Altair宣布成功收购ACUSIM Software, Inc. (ACUSIM)。ACUSIM软件公司成立于1992年，总部位于美国加利福尼亚州的山景城（Mountain View）。ACUSIM是领先的强大的可拓展性及高精度的计算流体力学（CFD）求解器解决方案领域的开发者。

2. 2016年，西门子与CD-adapco公司达成股份购买协议,以9.7亿美元的价格收购CD-adapco。CD-adapco是一家全球的工程仿真软件公司，提供的软件解决方案涵盖广泛的工程学科，包括流体力学(CFD)、固体力学(CSM)、热传递、粒子动力学、反应物质流、电化学、声学 and 流变学等。

3. 2016年，达索系统收购Next Limit Dynamics公司，Next Limit Dynamics是全球高度动态流体场仿真领域的领导者，其2015年营收约为160万欧元。其解决方案适用于航空航天与国防、交通运输与汽车、高科技、能源等行业。Next Limit Dynamics旗舰产品为使用格子玻尔兹曼的xflow。

4. 2017年9月28日，达索系统宣布，与产品工程仿真软件全球创新企业Exa公司(Exa Corporation)（纳斯达克：EXA）签署明确的合并协议，达索系统将正式收购位于马萨诸塞州柏林顿的Exa公司。这家达索系统的子公司将在此后10个工作日内邀约收购全部已发行的和流通的Exa普通股，每股价格24.25美元，收购完成后以现金支付，Exa的完全稀释股权价值大约为4亿美元。此次收购达索将EXA采用格子玻尔兹曼方法的旗舰产品powerflow纳入麾下，进一步巩固了其在流体无网格领域的技术和市场优势。

三. 生物，材料，复合材料

1. 2013年，ANSYS收购复合材料的瑞士小公司EVEN。

2. 2014年，达索系统通过收购 Accelrys 创建新的品牌 BIOVIA，Material Studio专用于微观材料仿真设计。

3. 2016年，Synopsys收购了英国三维重建软件simpleware

四. 电磁仿真，EDA领域：

1.2008年3月31日，全球仿真技术及产品优化设计软件供应商ANSYS公司宣布，与Ansoft公司双方签署一项最终协议，据此ANSYS将以收购总价约8.32亿美元（含现金

和股票)收购Ansoft公司, Ansoft旗下的旗舰产品HFSS, SIWave, Maxwell等悉数纳入到ANSYS旗下, 在多物理电磁仿真以及EDA领域迈出了扎实的一步。

2.2011年ANSYS收购了Apache Design Solutions, 填补其在集成电路解决方案领域的空白。

3.2014年6月,Altair公司与EMSS达成并购协议,收购FEKO软件产品以及EMSS在美国、德国和中国的三家分公司。

4.2016年 达索系统以约2.2亿欧元收购电磁(EM)和电子仿真领域技术领先德国企业CST。

5.2016年11月16日, 西门子以45亿美元收购EDA软件公司Mentor。

西门子以45亿美元收购Mentor, 让EDA软件公司再次进入人们的视野, 随着电子领域的持续升级, EDA公司的作用更加突显, 2018年中兴和2019年的华为断供事件更是把三大EDA软件公司Synopsys新思, Cadence楷登和Mentor推到了风口浪尖。前面早就讲过硬件断供只是明枪, 很容易躲开, 软件断供才是狠招。简单讲比如断供操作系统, 虽然电脑还能用, 但后续支持升级就没有了, 无法应对出现新的软硬件, 新的电脑病毒和漏洞。产品质量无法保证暂且不说, 产品上市时间会被拉长, 时间线上会被竞争对手甩在后面。

2017年2月2日, 海克斯康宣布以8.34亿美元收购老牌CAE公司MSC软件公司(“MSC”), MSC是全球领先的计算机辅助工程(CAE)方案供应商, 包括进行虚拟产品和制造过程开发的模拟软件。2016年MSC销售收入在2.3亿美元左右, 可谓卖了个好价钱。但也表明, 传统的CAE软件公司已经越来越不适应现代技术的发展, 软件公司必须在数字化设计方面转型, 在实际生产环境中的测量数据将与仿真分析紧密结合, 提供高附加值的产品和服务, 仅靠着卖软件的时代已经一去不复返了。

在计算仿真领域, 不得不提到另外两家独特的公司: COMSOL和MathWork。

成立于瑞典 的斯德哥尔摩COMSOL在2010年之前还无声无息, 最初的COMSOL只是在MATLAB的一个toolkit上改进的结构分析工具, 但在2010年之后, 随着COMSOL4.0版本的发布, COSMOL凭借着多物理场仿真的旗号, 直接求解PDE, 以及相对完善的前后处理功能, 迅速崛起, 在CAE领域可谓独树一帜。和大多数外国公司一样, 早年COMSOL在国内主要使用代理商, 后来自己成立办事处。创立于1984年的Mathwork, 作为一家从大学出来的公司, 一直靠着两大产品Matlab和Simulink, 2018年收入营业额已经比肩CAE巨头ANSYS。MATLAB提供了大量的数学函数以及各个行业的工具箱, 对于高级工程师是相当不错的选择。笔者从2001年接触MATLAB6.0版本至今, 仍然是解决技术问题的帮手。

作为3D软件的巨头，Autodesk在仿真领域也有不少动作，除了收购algor，moldflow，PlassoTech等，在仿真业务整合，云端也做了不少努力，但种种客观原因，未来想发展业务大概率通过收购其它公司完成。

展望未来十年，可能发生的收购或并购：

1. LSTC公司 其LS-DYNA作为显示动力学仿真的鼻祖和标杆，极有可能被ANSYS纳入麾下；
2. Synopsys新思，Cadence楷登会成为新的收购热点；
3. ANSYS不排除被更大的巨头收购；
4. 更多掌握核心技术的底层细分领域的小公司会被巨头收购（图形引擎，网格引擎，几何内核等）
5. VASP被ANSYS或者Simens收购；
6. 国内企业比如华为会尝试收购欧洲小的仿真企业。

下面试着从技术角度回顾和展望仿真软件的发展和趋势。

1、多物理场仿真耦合

多物理场耦合仿真的终极目标是提供对真实世界的精确模拟。虽然目前很多软件号称能解决多物理场仿真问题，但实际上获取准确的数值解仍然相当困难：理论上复杂的多物理场耦合现象难以用统一的PDE描述，因此数值计算上理论上就存在不确定性；而工程上会对模型进行简化，导致求解误差偏大；不同的物理场之间还涉及到模型交互，几何网格兼容性，强弱耦合，多尺度，相变，交叉学科等各种问题。所以进行准确的多物理场耦合分析仍然是仿真软件未来最具挑战性的工作之一。过去十年中，从事流体，电磁，材料仿真软件的公司是收购的重点，未来的收购也还会发生在这些领域，在这些领域还有很多没有解决的工程问题，软件发展潜力巨大。

2、仿真和硬件关系更加密切

FEM方法早在70多年前提出，因为需要大量的计算工作一直发展比较缓慢，在计算机出现之后才出现质的发展。现在，自由度超过千万的刚度矩阵组装，其次和非齐次线性方程组求解时间在普通台式机上仍然耗时，因此仿真软件对硬件有天然的依赖，半导体领域的摩尔定律同样适用于仿真软件，所以每次硬件出现新的革新，都会很快应用到仿真软件上。从早期的台式机多核多进程多线程，到后来的刀片服务器，网络分布式计算，再到GPU计算，以及现在的上百万核，千万核的超级计算机，AI芯片，将来可能应用的量子计算机，都是在用硬件方法加速仿真中的大规模计算。

硬件发展的另外一个趋势是越来越小，仿真可以和硬件结合，类似于现在的FPGA编程。自动驾驶是一个典型的例子，自动驾驶实际上就是一个实时的仿真系统，摄像头，传感器实时收集数据，处理器根据收集的数据进行驾驶路线计算和预测，并采取相应的策略，在

这个过程中依赖于数据的传输和对数据仿真处理。如果将碰撞仿真软件固化在汽车硬件里，汽车在行驶过程中能实时仿真与其它车碰撞的结果，从而让自动驾驶在遇到危险时采取更有利的预防和避险措施，其意义不言而喻。

随着硬件的加速，一些传统的数值计算方法使用可能会出现改观。比如求解满秩矩阵的困难的MOM，BEM等方法。目前的算法主要利用快速多级等进行加速，但精度上还是会有损失。

在下一个十年结束的时候，希望在量子计算领域有所突破。如果几十亿自由度的线性方程组求解能在普通机器上秒级以内完成，将极大促进生产力发展，说人类文明将跨进一大步一点都不夸张！

3、CAD/CAE/CAM/CAPP无缝结合，验证测试与仿真融合

传统的CAD/CAE/CAM/CAPP相互独立，会造成很多问题，比如信息孤岛，数据不兼容，仿真结果无法验证等，甚至出现试验指导仿真的情况。近年来，软件厂商在解决这些问题方面做了很多努力，但距离理想还相差很远。传统的CAD与CAE分离，简单讲设计一个单位，仿真一个单位，加工一个单位，最直接的问题就是数据兼容性问题。相信很多仿真工程师都有过被文件格式转换，几何清理折磨的经历，究其原因就是CAD/CAE流程没有打通。这种分离不仅是数据上，而且也反映在业务上。比如因为加工需要，CAD设计会有各种倒角，而仿真则不需要。很多CAD软件在其平台上引入了仿真功能，反映的就是这个趋势。在一个平台上完成CAD/CAE/CAM验证，设计，自动化，也就是CAD/CAE/CAM/CAPP的无缝结合。

仿真软件给出的仿真结果总会存在误差，需要由试验来确定，如何将实验数据和仿真数据匹配，指导实际业务，也是很工业界看重的一个问题，这也催生出了相应的软件产品，比如西门子收购的比利时LMS，是一家提供仿真，测试，咨询服务公司，西门子看中的就是测试与仿真的综合能力。而另一家比利时的Dynamic Design Solutions，也是一家提供FEM分析软件测试的公司（又是比利时，多物理场仿真软件OOFELIE也属于比利时的OpenEngineering公司），几乎所有的汽车设计厂商都是其客户。未来传感器，测量工具效能的提升会让仿真与实验结合的更紧密。

4、设计与优化紧密结合

有人说仿真的目的就是优化。优化是很多CAE软件产品中的一个模块，比如AltairOptiStruct，Ansys Design Explorer，HFSS Optimetrics，Abaqus ATOM以及独立的优化软件Isight，Tosca等，Matlab等科学计算软件中也提供了专门的优化工具箱。虽然优化工具很多，但真正应用起来让其发挥作用并不简单，首先每种优化算法有其特点和适用场景，比如遗传算法适用全局优化，牛顿适用局部优化，神经网络适用输入数据丰富情形，梯度迭代更适合有连续表达式的函数；其次业务场景对优化也有诸多限制，比如桥梁，虽然理论上可以设计优化出各种样式的桥，但实际设计施工中还是老老实实按照规范来；再者优化算法同样受限于硬件，大部分优化使用迭代算法，利用每次迭代仿真结果选择相应策略，而很多FEM自适应网格方法，FVM计算本身就是一个迭代收敛的过

程，如果加上优化迭代，计算时间会无法接受；最后优化中涉及到许多变量，这些变量可能涉及不同的阶段（有的变量属于几何，有的属于仿真，有些属于NC），场景，从而分配不同的影响因子，灵敏度。所有这些不确定因素加起来导致仿真软件优化发挥作用很困难，目前仿真优化通常应用于几何拓扑确定，业务流程固化，参数较少的应用场景。在仿真优化这块还有很多潜力可挖掘，未来十年，将会有更多的和业务结合和优化应用出现，而软硬件的发展会对优化的应用发展起到很大的促进作用。

5、云端化

仿真软件云端化是互联网技术发展必然的结果，云端化和仿真技术没有太大关系，主要是商业模式和使用方式的转变，目前有不少公司试水把产品移植到云端，比如三维设计软件OnShape，3D软件公司Autodesk（欧特克）早早在云端布局，云端平台Fusion360整合了旗下的多款CAD/CAE/CAM产品。互联网巨头亚马逊直接在其云端开发部署了开源CFD软件包OpenFOAM，用户可直接登录使用。另外B/S架构的仿真平台通过浏览器提供前后处理，而大规模求解仿真放在云端也是一个不错的选项，国外发展较早的有德国的www.simscale.com，国内有www.simright.com，目前云端化存在两个大问题。一个是安全性问题，如何保证数据安全，对公司讲，数据就是核心资产，另外如何保证商务，法律上的安全也是需要考虑的一个问题，比如涉密信息，即使保证了数据安全，也不可能放在共有云，公司企业内部私有云倒是一个不错的选项。另外一个问题是进行云端仿真计算时，存在性能瓶颈，对于大型项目，远程图形渲染，实时计算，交互，数据传输等还不能和桌面机器相比，当然随着硬件的发展，这些以后都不是问题。

对大多数用户来说，使用云的好处是降低仿真软件的使用成本，以前几十万几百万的才能使用的软件，通过云可以将价格门槛下降到很低，这也是在国内普及正版软件一条可行的路。

6、网格

传统的数值计算方法都需要高质量的网格数据作为输入，网格生成在前处理中占了很多时间。网格质量的好坏直接决定的仿真的好坏，网格生成技术作为仿真领域的一项支柱，开发难度大，且需要实际工程打磨，要想开发一套完善的网格引擎，没有五年十年的功夫是不可能的。某些情况比如多物理场，流体边界层，复合材料，EDA行业，装配体的网格的生成更为复杂，对于未知场的求解，往往需要网格加密多次求解，增加了仿真复杂度。好的网格依赖于几何输入质量以及网格算法的健壮稳定性，没有好的网格生成机制，数值计算难为无米之炊。网格算法较为成熟稳定，在这个领域很难有颠覆式的发展，未来网格底层生成算法可以和实际工业应用更加紧密结合，优化网格生成效率。

由于高阶数值计算复杂性，目前大部分商业软件网格都是线性单元（0阶）或者二次单元（一阶）。实际上高阶网格（也就是常说的P单元）在处理复杂几何上更具优势，适当的设置网格参数，只需更少数量的网格，无需迭代，就能获得更好的数值计算解，期待未来在高阶领域有更多实际应用。

无网格方法是近年来兴起的一种数值计算方法。流体领域的粒子方法(Particle Method) 和LBM格子玻尔兹曼已经相当成熟，尤其是后者，达索系统早早将市场上的LBM软件收到旗下，垄断目前市场。Altair与今年推出的SimSolid号称也不需要网格，市场验证需要假以时日。要想在无网格应用上有所突破，底层理论和硬件都需要提供更多的支持。

瑞士一家公司提出了一种基于MAR (Medial Axis Representation) 方法，号称利用CAx取代CAD/CAE，取代传统的网格划分流程，具体效果还需要市场检验。

未来，网格仍然是仿真软件前处理的重要一环，网格生成性能会进一步提升，新算法会引入让网格生成趋向智能化。

7、数值计算方法

在数值计算方法领域，主要还是以FEM为主，电磁FEM，FDTD，MOM，流体FVM，以及无网格法。其它比如FDM，BEM，DEM，谱方法等在实际应用中大规模使用，还需要更多的努力。未来数值计算方法在理论上突破已经非常困难，但在以下几个方面仍然还有发展空间：

- 1，同一分析空间中不同数值计算方法的混合使用；
- 2，PDE产生新的解析解方法所带来的新的数值解法；
- 3，半解析解和数值解法的进一步组合使用；
- 4，数值计算方法和业务场景实际工程应用的进一步耦合；
- 5，高阶网格数值计算方法；

前面讲过很多次，软件只是表现，软件和工程经验的结合才是核心技术，发展数值计算方法，只有和实际工程应用结合才能使软件产生更大的价值。

8、人工智能以及机器学习引入

人工智能最近几年火的一塌糊涂，但是真正静下心来仔细研发会发现，落地应用的项目其实集中在少数几个领域，比如语音识别，图像识别，聚类分析，专家系统等等，在仿真领域应用有限。这也是由仿真领域的特点决定的：

- 1，人工智能依赖于大规模的训练数据输入，对于仿真而言，很多应用场景中，仿真一次的数据量就非常大，多次仿真不仅时间长，而且冗余数据多，生成的有效训练数据有限，无法提供AI训练以及深度学习所需的有效输入数据；以设置最优网格参数为例，至少两个位的训练次数才会出现有效解，而求解器一次求解时间可能就是几个小时，相比AI，实际应用中自适应网格划分更迅速和准确；
- 2，目前的硬件设施无法满足AI训练，或者训练成本太高，对于1中即使是对局部网格寻找最优参数解，训练所花实际成本仍然很高，性价比差；
- 3，未来通过机器视觉进行特征识别，从而可以在几何清理，设置局部网格参数等方面实现自动化是比较可行的。

9、业务驱动仿真

传统印象中，仿真应用流程是设计，仿真，再设计，测试，再设计，生产。而业务驱动仿真很可能改变这种印象。以风力发电叶片为例，叶片的生成，运输，安装，卸载成本都很高，在运作的时候如果叶片出现故障（撞击，损坏），主要以现场修复为主。如何精准的评估修复效果，利用传感器，扫描仪等装置生成实时模型，收集真实工况数据，对比原始模型，测试模型，进行实时仿真，评估CFD，机械强度等各个指标，仿真满足要求后即可上线。仿真在整个过程中起主导作用，之后还可以定时进行仿真检查，收集实时数据反馈给原始仿真计算模型，优化产品流程。

从使用者角度看，仿真软件将不再是仿真工程师的专利，完善的业务流程，固化的仿真逻辑，CAD/CAE/CAM的高度结合，可以让任何一个普通工程师用好仿真。想象一下在建筑工地上，工人发现施工与设计图纸出现冲突，需要改图，传统的做法需要层层上报，甚至反馈到设计院，设计院实地考察，给出修改方案，来回折腾。而以后可能工人只需在pad上修改图纸，实时仿真一下，将结果反馈给上级，通过则继续施工。

过去的十年中，几乎每年都有一些技术热点，比如量子计算，物联网，AR，VR，人工智能，区块链，边界计算，深度学习。但是仔细探究很多技术只不过是新瓶装旧酒，有些是背后有资本推手，有些是从实验室走向市场，有些是得益于硬件的发展。对于国内工业仿真软件发展而言，要想有所建树，技术方面加大研发投入力度，潜心踏实积累技术，参与实际工程迭代，业务上开拓国外市场，走产品服务路线，仍然是唯一的出路。

深入理解数值计算网格(全篇)

原创 邓子平 多物理场仿真技术



为了方便阅读，将所有文章放在了一起，并且点击标题可以单独阅读。

1. 网格介绍

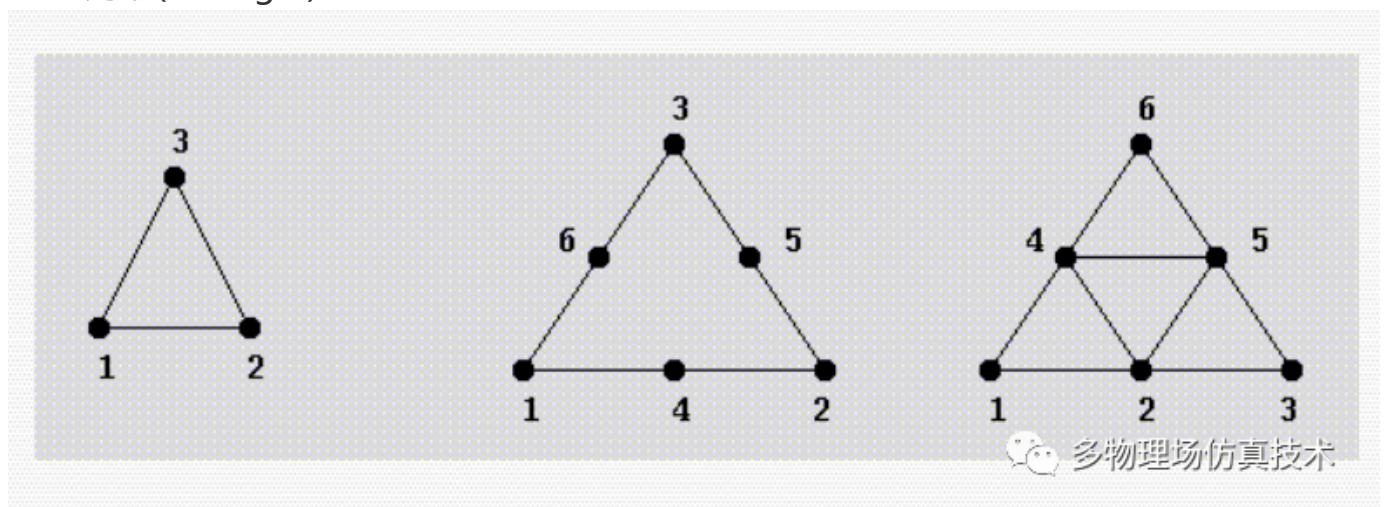
网格(Mesh)在数值计算中有着举足轻重的作用。主流的数值仿真方法诸如有限元，有限体积，有限元，边界元都是以网格为计算对象。而差分法等，时域有限差分等也是以网格(Grid)点为计算对象。

什么是好网格？

网格的好坏直接决定了仿真计算能否成功，以及正确性，精度，性能。简单的说就是尽可能用最少的网格，最真实的反应物理量的变化规律。

常见的网格种类：

1. 三角形(Triangle)



以上图形从左往右分别为

左> 0阶一次单元，3个顶点，3条边

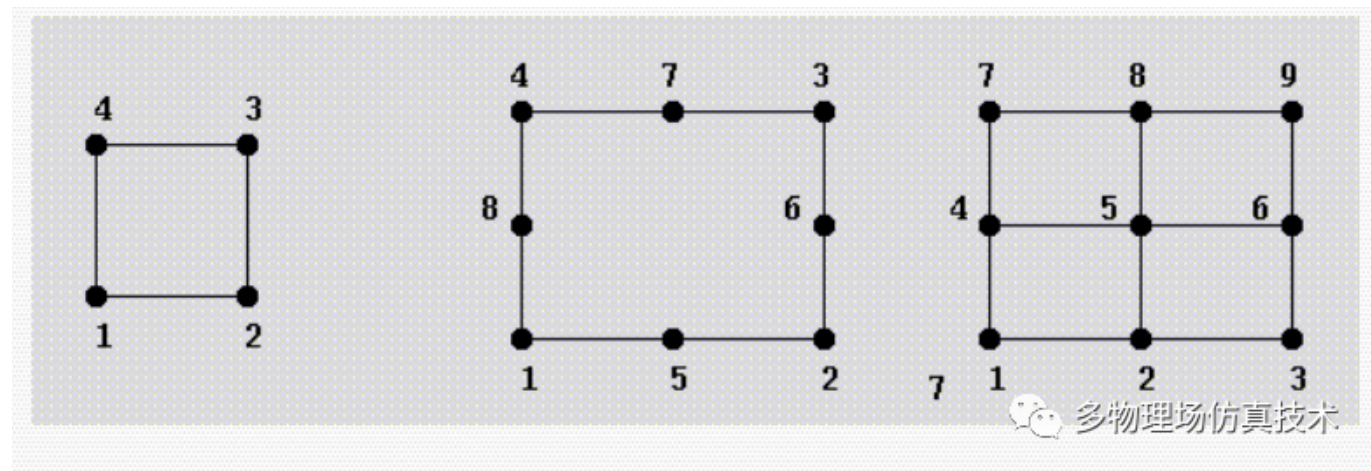
中> 1阶二次单元，6个顶点，3条边

多物理场仿真技术

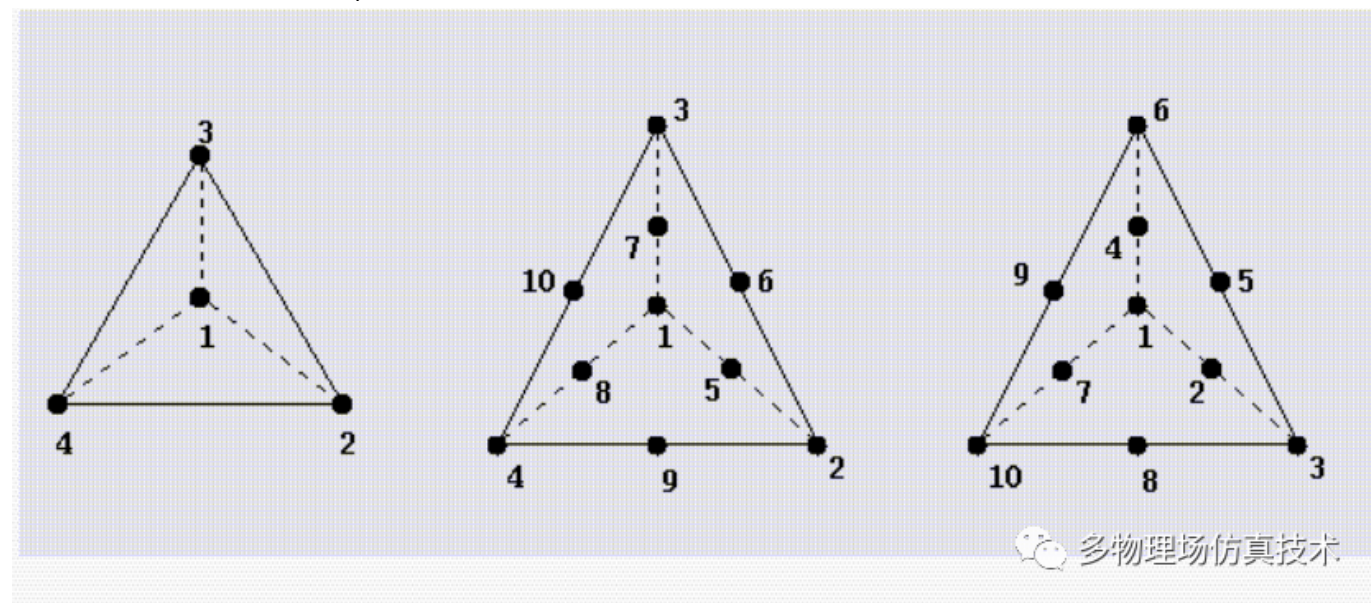
右> 拉格朗日1阶二次单元, 6个顶点, 6条边

以下单元类型可以类推

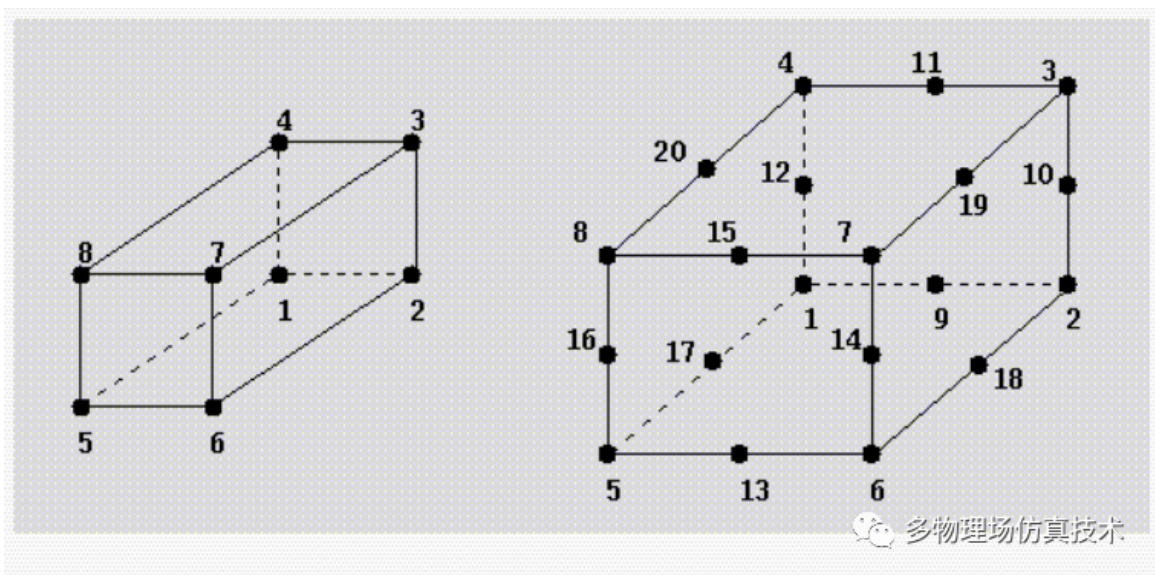
2. 四边形(Quadrilateral)



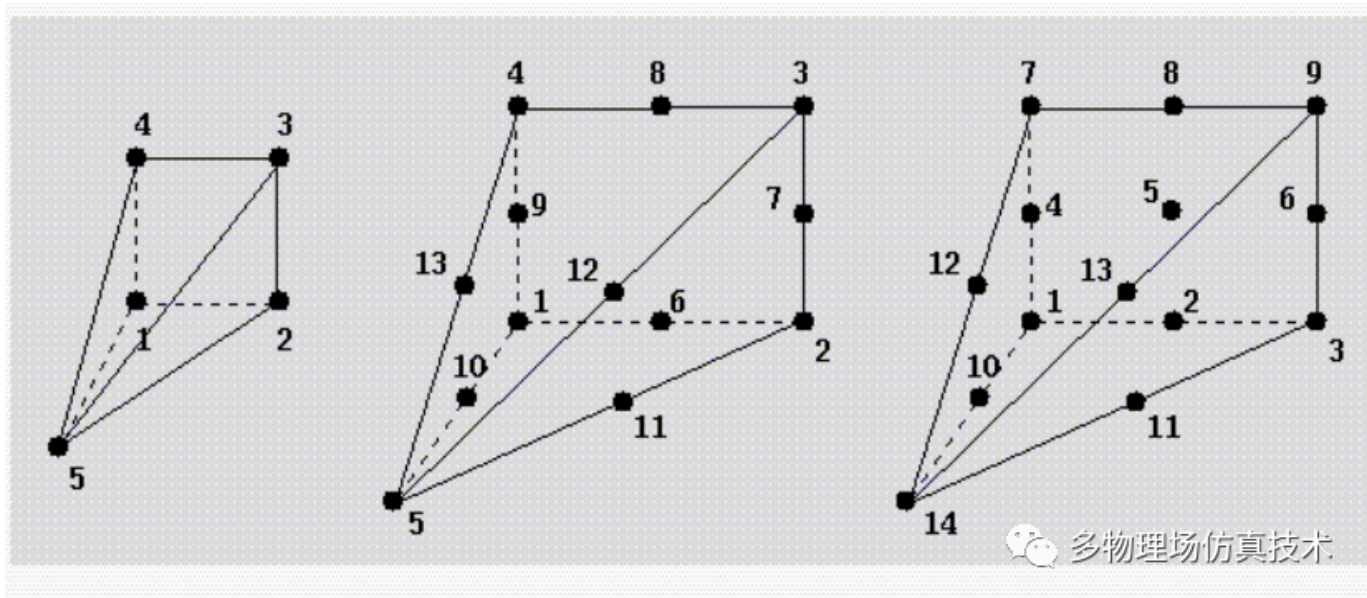
3. 四面体(Tetrahedron)



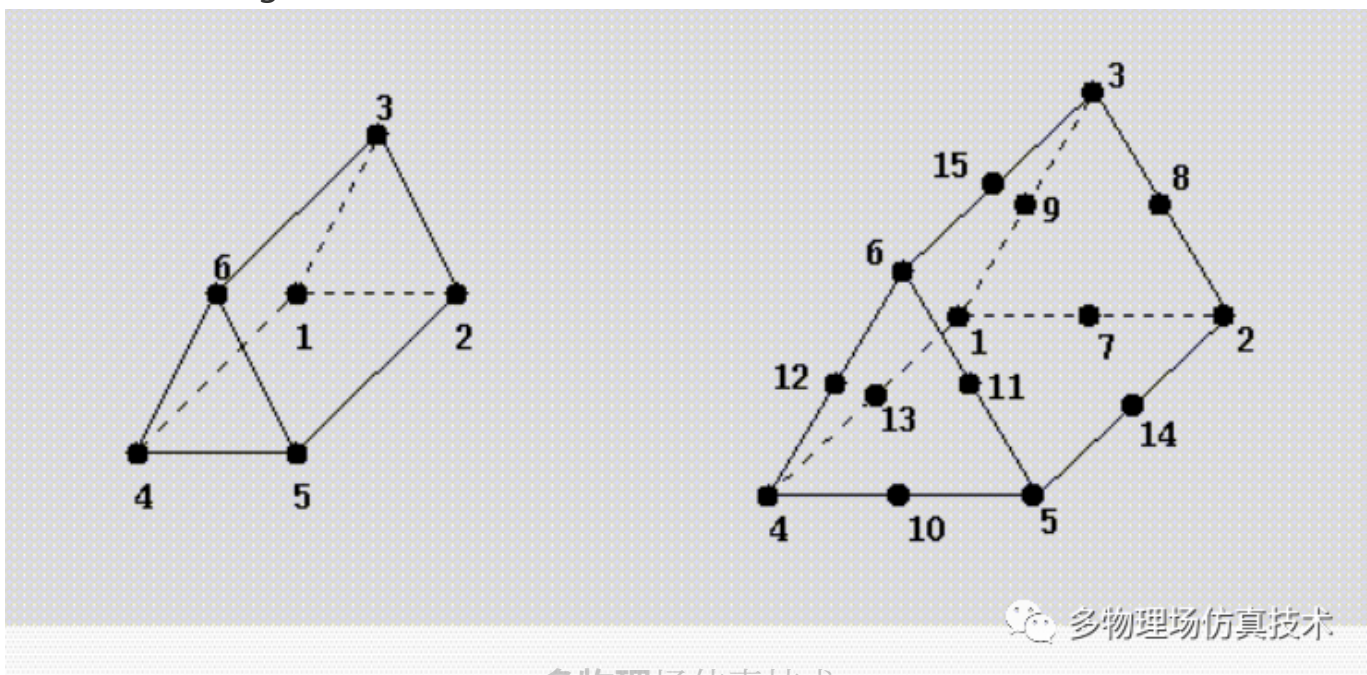
4. 六面体(Hexahedron)



5. 金字塔单元(Pyramid)



6. 楔形单元(Wedge)



网格的分类：

网格对应的英文有两个Mesh和Grid，Grid 的基本意思为格子，即四四方方非常规整的网格。

对于Mesh，我们通常分为**结构化网格**和**非结构化网格**。理解很简单，除了**四边形**和**六面体**是结构化网格，其它都是非结构化网格。

对于Grid，通常情况下我们用作有限差分方法的网格。

高阶单元：

除了上述我们提到的**0阶1次单元**以及**1阶2次单元**，在数值计算上还有更高阶的单元。高阶单元在数值计算上具有更高的精度，当然也会带来更大的计算量。在所有物理场的数值计算中，通常情况下**1阶2次单元**具有较好的性价比，即在精度和计算时间上有比较好的平衡。

以电磁矢量棱边3维四面体单元为例，其基函数自由度DOF与阶次的关系为：

$DOF = (n+1)(n+3)(n+4)/2$; n 为阶的数值

$n=0$ 时， $DOF=6$

$n=1$ 时， $DOF=20$

$n=2$ 时， $DOF=45$

$n=5$ 时， $DOF=216$

以电磁矢量棱边3维六面体单元为例，其基函数自由度DOF与阶次的关系为：

$DOF = 3(n+1)(n+2)(n+2)$

$n=0$ 时， $DOF=12$

$n=1$ 时， $DOF=54$

$n=2$ 时， $DOF=144$

$n=5$ 时， $DOF=882$

也就是说对于5阶的六面体单元，其一个单元的刚度矩阵可达 882×882 ，对于开发验证，复杂度远远高于常用的0阶和1阶单元。

几何和网格

在网格之前，我们通常拿到的都是几何数据，比如BREP结构，STL面片结构，或者参数化的几何结构。

几何离散成的三角形称之为“面片”，非网格，大部分显示引擎底层都使用三角形来渲染对象，因此2维、3维几何都需要三角化（称之为“面片化”），比如一个长方体共有6个面，每个面需要离散成两个三角形，总共12个三角形。这12个三角形我们称之为12个“面”

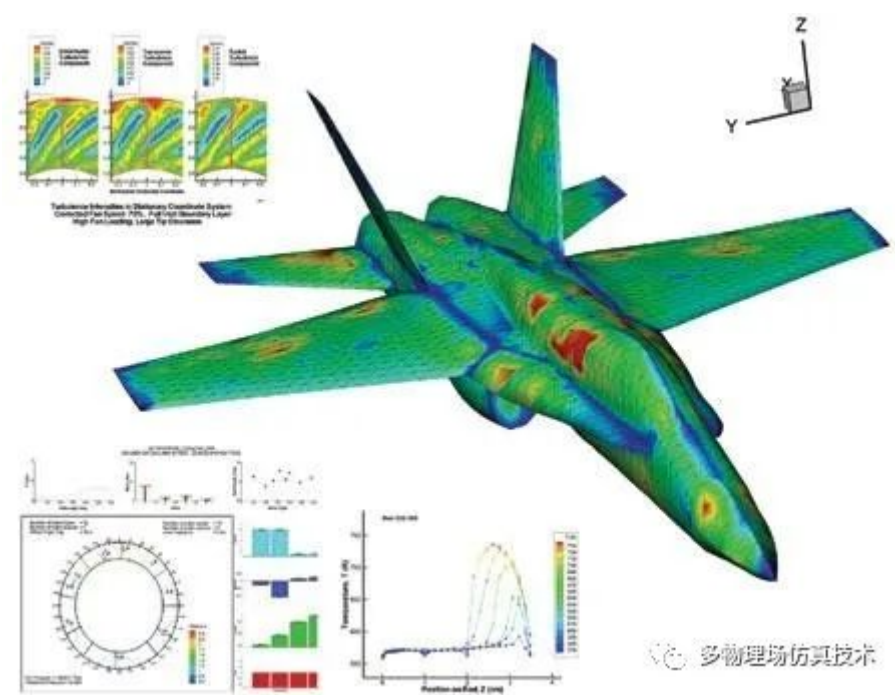
片”，而非网格！一般情况下“面片”质量很差。“面片”有两个功能：一是用来做显示渲染数据；二是可以作为网格划分的输入数据，用来生成面网格。
对于隐式曲线曲面，需要设置合适的离散参数，从网格端进行解析。

有限元方法中的单元和网格

网格是几何上的概念，单元则是数值方法中的概念。
弹簧只能在一个方向上发生变形，是典型的1维单元；同理壳单元（shell）需要XY两个方向来定义，是2维单元；四面体，六面体是3维单元，也称为实体单元；对象可以看做质点的为0维单元，比如称之为“定楼神球”的调谐质量阻尼器。

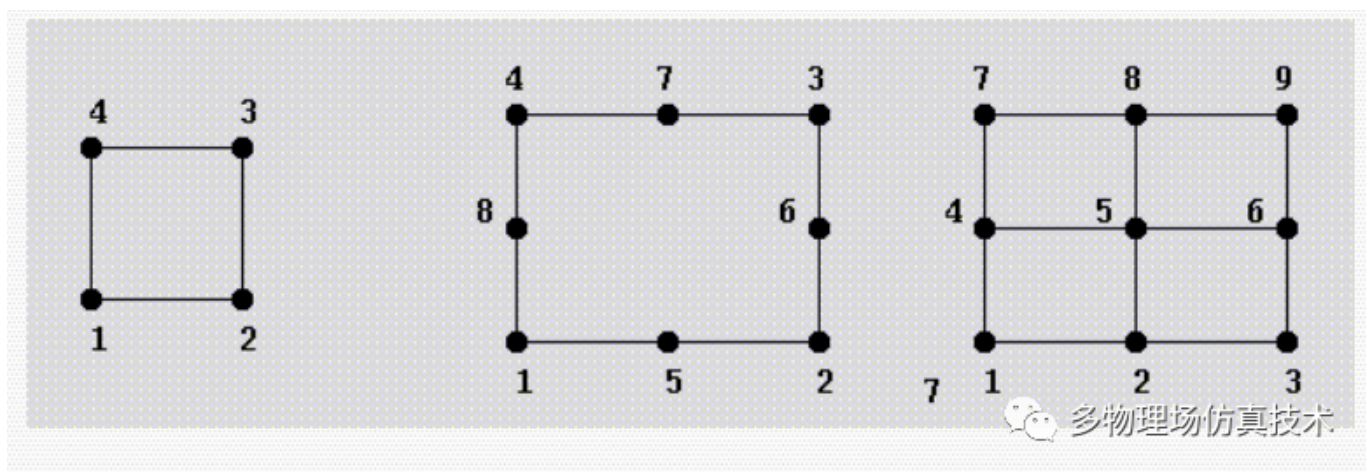
网格生成算法

网格生成算法无论是在理论还是实际应用中都比较成熟，后续将会详细介绍。在这个领域很难有颠覆式的发展，未来网格底层生成算法可以和实际工业应用更加紧密结合，优化网格生成效率。

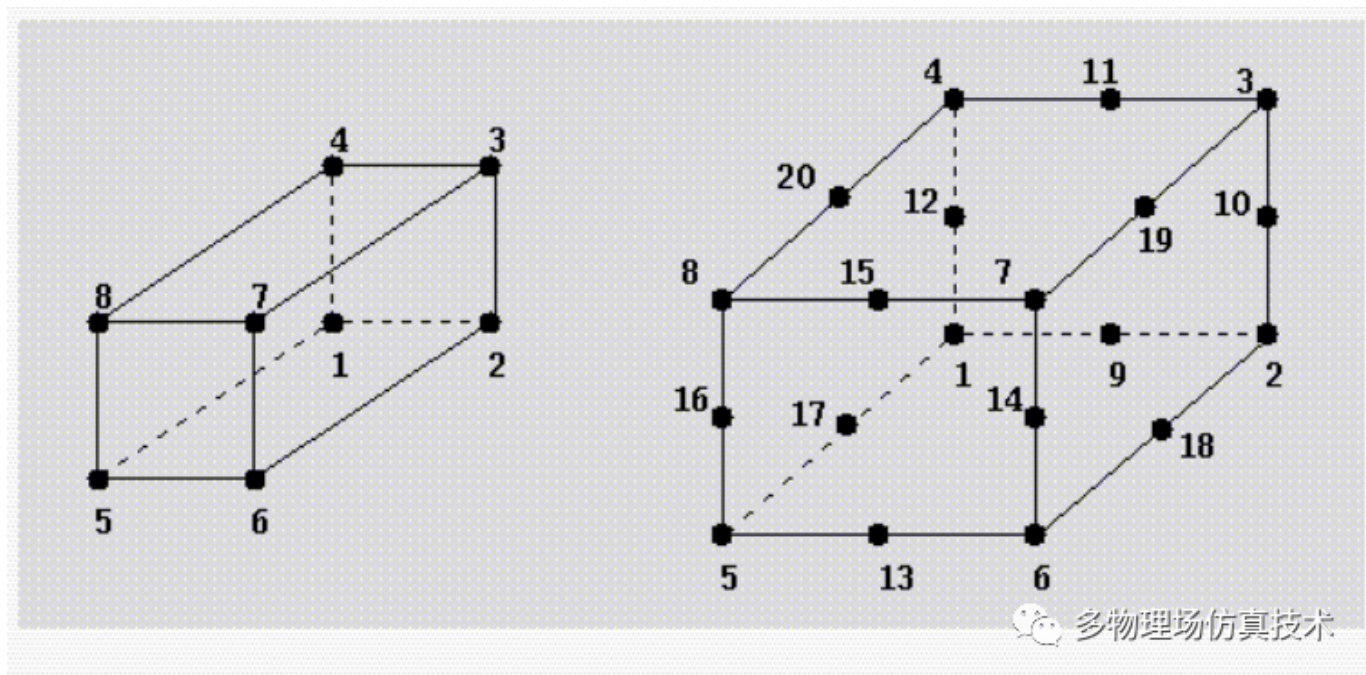


2.结构化网格生成算法

结构化网格主要指四边形和六面体



常用四边形网格



常用六面体网格

在实际应用中，一方面结构化网格在形函数基函数计算，等参变换，处理模型，模型编辑上有优势，比如网格分割，组合，沿固定方向加密，还可以方便拆成非结构化网格；另一方面，对于不太规则几何的几何体，自动生成结构化网格比较困难，需要大量的人工干预。所以在通用有限元分析中一般优先推荐结构化网格，另外CFD分析中，六面体网格天然适用流体的有限体积计算方法，也是首选。

1. 最简单的结构化网格方法是扫掠。对于三维，选中一个面，划分四边形，然后沿另一维度扫掠生成六面体。这种方法需要用户手工参与，且对几何有一定要求，扫掠生成的网格质量一般都不错。除了单向扫掠，实际算法中会有多轴扫掠。扫掠的改进方法即类似于在扫掠方法渐进的波前法，又叫plastering方法。

2. 结构化网格生成算法中，有一种间接生成方法，即首先创建非结构化网格，再合并。比如二维网格中先产生三角形，然后三角形两两组成一个四面体。六面体则是先生成四面体，然后组装六面体。这种方法又叫Morph算法。

3. 由于结构化网格有比较好的边界特性，所以多块(MultiBlock)划分也是常用的方法。即首先将几何区域分成子区域，在子区域里分别划分网格，再进行合并。比如生成六面体，一般会将空间进行八叉树划分，在每个八叉树子节点内进行划分，但这种方法需要处理边界，优化边界网格的质量。

4. Whisker Weaving是一种基于拓扑连接的自动六面体生成算法。首先在边界上生成四边形，然后以四边形为输入采用波前法推进在三维模型内部建立六面体的对偶表达，根据对偶表达生成六面体。

全自动的六面体网格生成仍然是网格生成的难点。

考虑到结构化网格生成的工程复杂性，本文仅对生成算法做基本介绍。通用前处理软件以及各种仿真软件中都会提供结构化网格功能，但全自动的网格质量一般都不会太好，需要一定的手工操作。

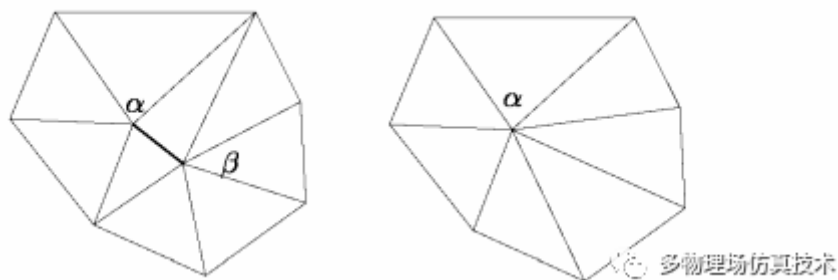
网格优化

网格生成之后，最后需要优化网格质量，常见的几种质量优化方法：

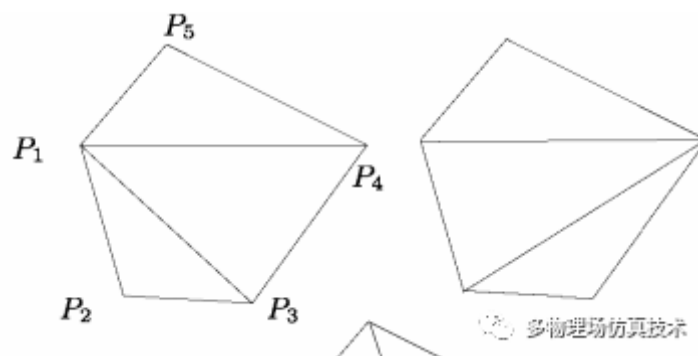
1. 移动顶点位置

移动顶点位置，让和顶点关联的单元质量都能改进

2. 移除顶点(通常说的edge collapse)



3. 交换边



4. 交换面(同交换边)

5.分解边

即将一条长边分解成两条短边

最后说一下网格的并行计算，网格并行主要有两点：一是几何模型的分解或者网格模型的分解；二是网格的拼接。这两点做好后，网格大规模并行计算基本没什么问题。现在的分布式计算和并行计算工具已经非常成熟，可以拿来直接使用。

3.非结构化网格生成算法

通常说的非结构网格主要指非四边形和六面体网格，包含三角形，四面体，楔形，金字塔等，在实际应用中最常用的还是三角形和四面体。本文也主要介绍三角形和四面体的生成算法。

非结构化网格自动生成主要包含三种方法：

1.Delaunay method

(关于此方法的汉语翻译特别杂，不建议用中文翻译)

2.Advancing-Front method (波前法)

3.Spatial decomposition based method

(基于空间分解方法)

1.Delaunay method

生成三角形最常用的方法是Delaunay方法，也是所有网格生成算法中的最“基础方法”，在很多开源软件中都有其方法的实现。

Delaunay三角网格定义：平面上的点集P是一种三角剖分，使得P中没有点严格处于剖分后中任意一个三角形外接圆的内部。简单讲，给出一堆离散点，Delaunay网格是满足所有网格质量最好的网格。

Delaunay方法目前有很多改进方法，只介绍其中最基础的一种方法:逐点插入法

生成算法：

1.给出一堆离散点，初始化三角形链表

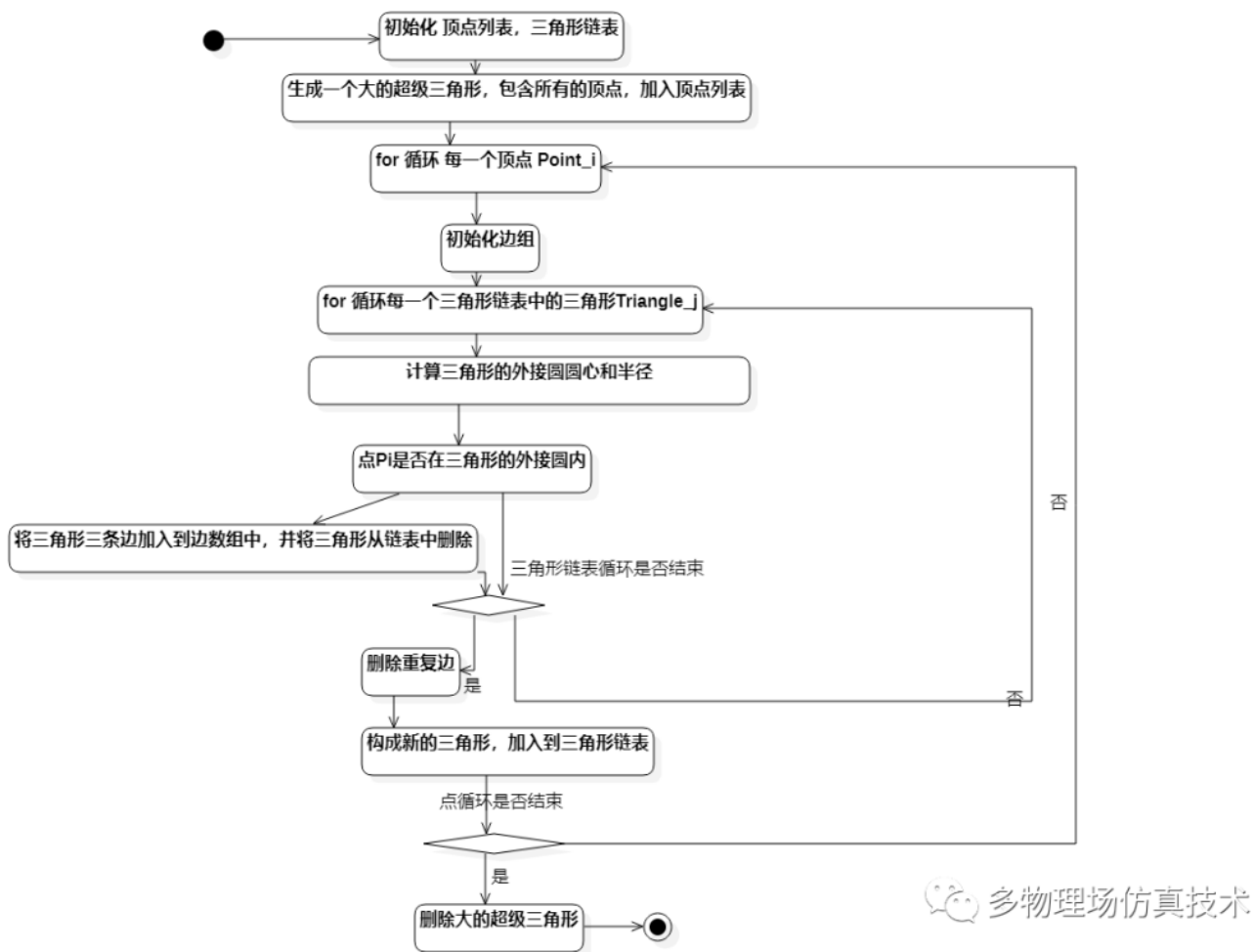
2.给出一个大三角形包含所有离散点

3.计算三角形链表中的每一个三角形的外接圆圆心和半径

4.如果点在已知三角形外接圆内，则把三角形的三条边加入到边数组中，在三角形链表中删除该三角形

5.删除所有重复的边，重复步骤3

6.用点和边数组中的每条边都组合成一个三角形，加入到三角形链表中，重复步骤2



Delaunay网格插入点UML活动图

作为基础的网格生成算法，Delaunay不仅支持二维，而且也支持三维。在三维计算中，只需将**外接圆**的判断改为**外接球**的判断即可。

2.Advancing-Front method（波前法）

Advancing-Front 方法又叫波前法，前沿推进法等。

其核心思想是沿着原始的网格边界生成网格，然后逐步推进生成，类似于波浪向未划分网格区域前进，直到所有区域被网格填满。

波前法思路清晰，尤其适用于已知边界的情况，其网格属性在生成过程中可以动态调整，控制好，最终生成的网格质量也比较好，是生成非结构化网格的基础算法之一，同时Advancing-Front 方法的经过改进，也适用于四边形和六面体等结构化网格生成。

经典的波前法对二维从一组边开始，对三维从一组三角形面开始。以二维为例，网格生成策略一方面逐个单元逐个单元生成，创建插入新点和新单元，同时新的单元和之前的单元保持连通性。算法的一个关键点是如何插入新的点，新的单元需要满足网格尺寸和单元质量要求，波前法的一个优势是启发式（heuristic）算法，可以生成高质量，梯度可控的

网格。相比其它算法，边界的完整性可以得到保证。其不足是在三维方法时，有可能出现不收敛的情况，即有些区域很难完全填充网格。

波前法的典型流程：

1. 定义网格输入数据和参数，包括整几何区域，边界，网格尺寸，网格梯度
2. 离散边界
3. 从边界任意位置开始，逐步插入点，形成单元，迭代指导求解区域被网格填满

在这个过程中，如何插入点和形成单元是算法最核心的部分，其涉及到：

1. 前进方向的单元选择
2. 前进分析以及定义最优点需要考虑的各种可能性
3. 单元构建有效性分析
4. 选择合适的数据结构

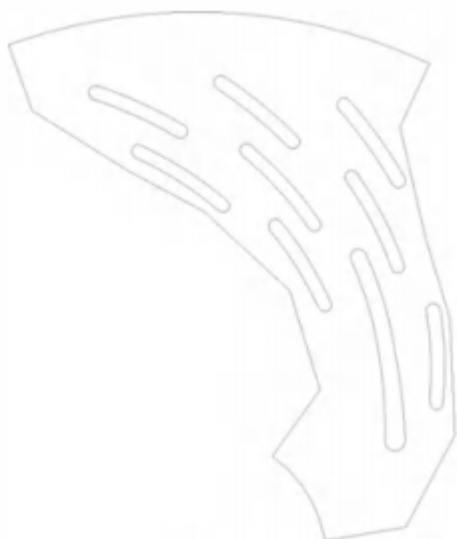
相比较其它方法，波前法在三维实现上要远远高于二维。三维需要考虑更多的因素和特殊情况，同时在算法效率上，三维的要求也更高。

3.Spatial decomposition based method (基于空间分解方法)

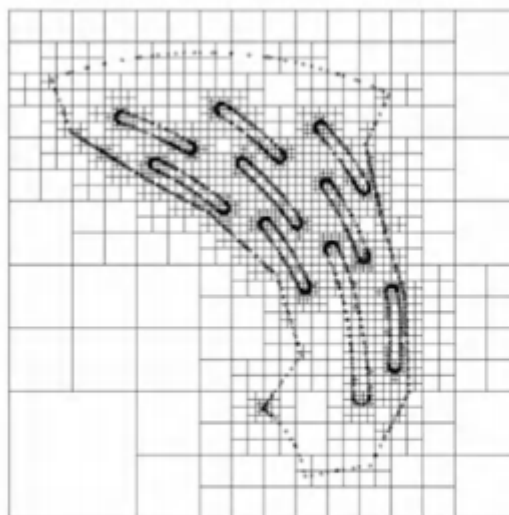
基于空间分解方法相对简单，主要用到空间树结构，即二维四叉树和三维八叉树。以平面四叉树为例：

1. 根据几何对象，创建合适的树边界
2. 根据对象点的分布，创建四叉树结构
3. 在每个四叉树内部，利用四叉树的边和点，生成网格
4. 调整网格，使其满足网格质量要求和梯度分布

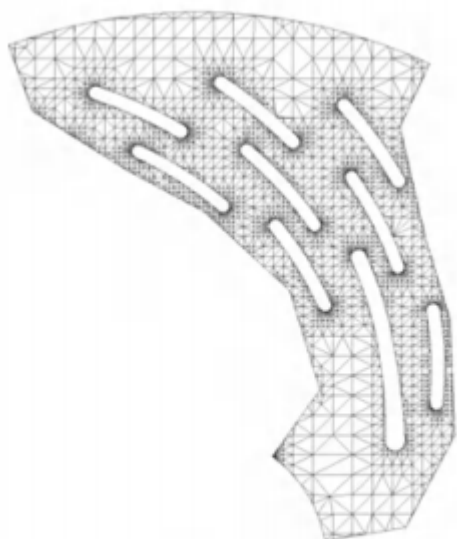
一个典型的例子如下：



i)



ii)



iii)



iv)

多物理场仿真技术

对于使用四叉树和八叉树方法生成网格，两个核心点：

1. 动态生成不平衡的树，即每个同级的树结点数目并不相同；
2. 在树结点内部选择合适的算法生成初始网格

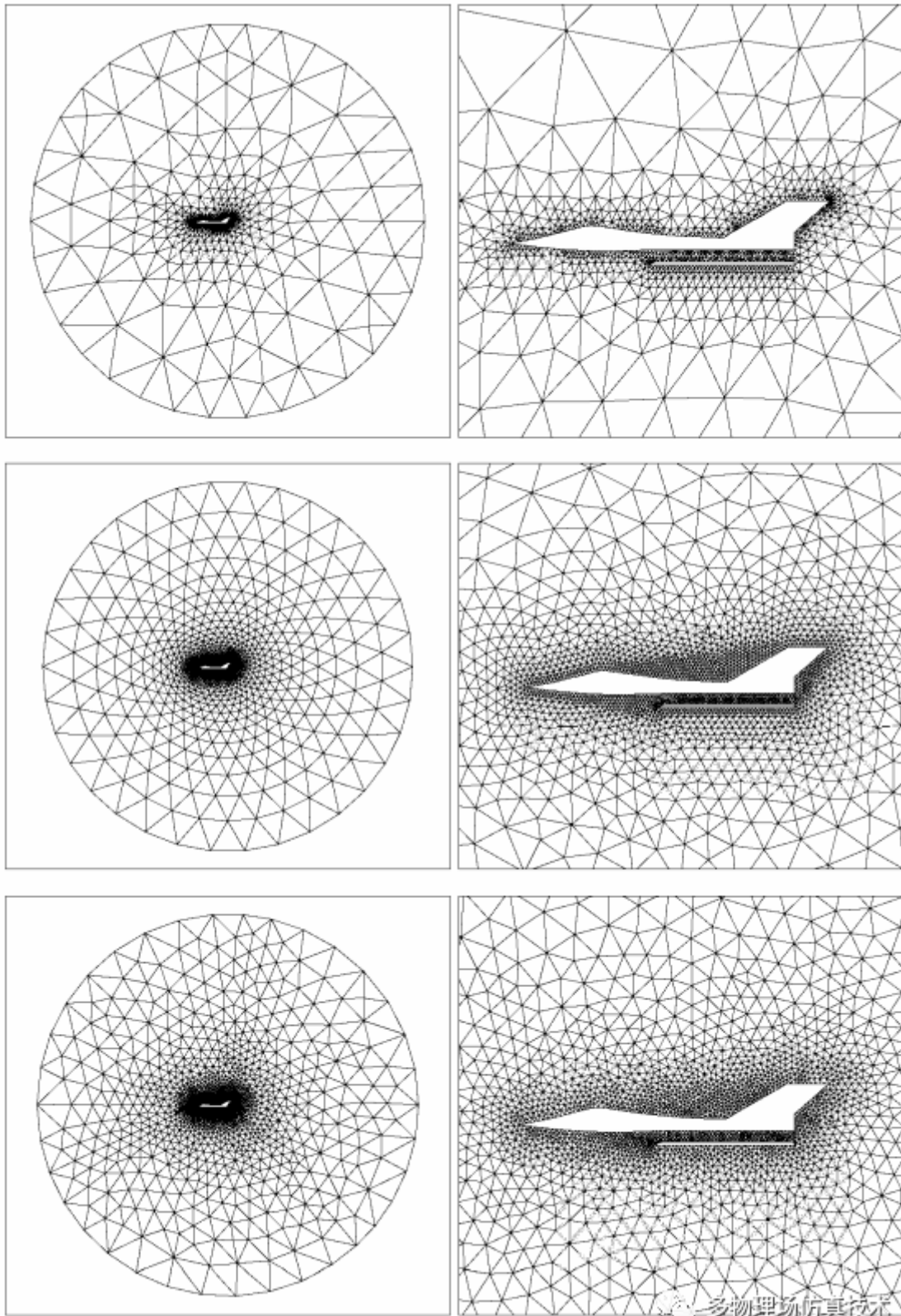
笔者在几何，图形，网格，求解器开发中，都较多的使用到了八叉树结构，八叉树结构可以说是一种基础性的数据结构和方法。

三种方法生成的网格比较：

method	np	ne	Q_M	Q_{worst}
quadtree	1,246	2,171	1.25	1.88
advancing-front	2,557	4,795	1.1	1.61
Delaunay	2,782	5,528	1.16	1.82

多物理场仿真技术

np 为点的数目， ne 为单元的数目， Q_m 为网格的质量， Q_{worst} 为最差的网格质量



图片从上往下方法分别是：
四叉树,波前法,Delaunay

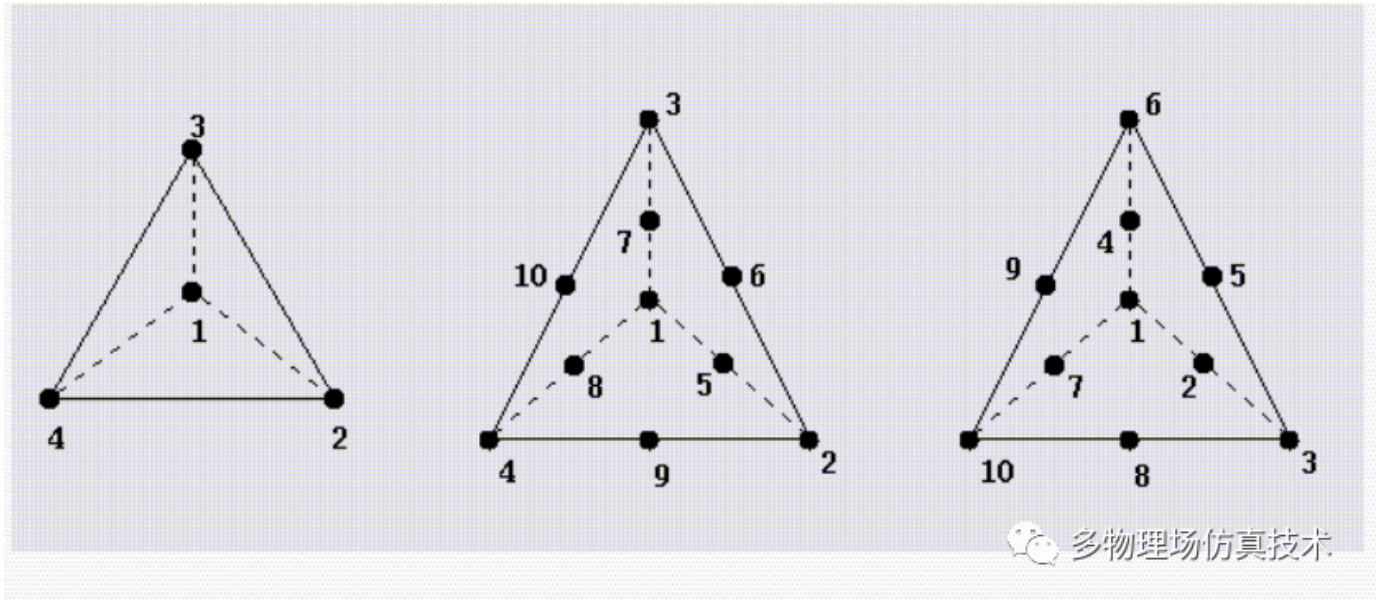
图片皆来自于书 Mesh Generation (Application to Finite Elements), 考虑到网格参数设置的差异, 重点关注网格质量和分布

在实际应用中, 三种方法并不是独立存在, 经常会有方法混用的情况, 比如在CFD网格中, 不同区域采用不同的网格策略, 可能效率和稳定性更好。

4.万能四面体

在三维分析中经常用到四面体，六面体，金字塔等三维实体网格，其中四面体因为其容易自动化生成，适应性好，数值计算方便等优点，在多物理场三维有限元分析中得到广泛应用。

几何特点



通常我们拿到四面体的是四个点的坐标，利用点可以计算出边，面的数据，在计算中要保持数据的一致性，即点，边，面的顺序要有固定规律。

以面为例，通常按照右手规则，三点顺序连接成面的法向量要一致朝外。

四面体计算中经常要用到任意四面体的体积和四面体角度

以上指标的计算方法C++源码可在 www.cae-sim.com 下方资源下载



FasTCAD

TCAD工具



FasMaterial

计算材料软件包



FasDICOI

CT DICOM文件浏

[直接下载地址](#)

[源码资源下载](#)

[四面体体积, 角度计算](#)

多物理场仿真技术

网格质量评估

除了常见的最大边，最小边，最大角，最小角等。还有以下四种评估指标：

1.Aspect ratio：取得最大的边长A，取得最短的高度H（高度定义为一个顶点到对面的距离）。最大边长除以高度：A/H即为Aspect Ratio

2.坍塌比：

计算公式： $(H/\text{Sqrt}(A))/2.14$

H为顶点到对面的高度，

A为顶点对面的三角形面积，

坍塌比范围为0到1,1是正四面体，0为完全坍塌

3.二面角

六条边，每条边关联两个面的夹角，总共有六个角

通常取最大最小的二面角评估网格质量

4.扭曲度：

单元体积与相同大小外接球条件下的理想四面体体积之比

计算公式：实际体积/理想体积

正四面体值为1

常见的四面体几何错误包括：

1.自由点，边，面，

2.重复的点，边，面和实体

3.四点共面

4.相邻两点距离或Aspect ratio超过数值计算误差

对于高阶单元，四面体网格的几何和质量计算方法相同。

网格质量是如何影响仿真精度的

几乎所有书籍解释就是网格质量差，会造成误差大，计算精度不够。

但其实网格真正对仿真精度的影响来自两方面：网格密度和网格分布。

看一个场景：假设现在网格已经最优，达到最理想状态，我们将其中两个相邻网格加密，使其网格质量变差，但其实仿真精度并不受影响，只是网格分布变差，网格密度增加。换句话说，除去数值计算误差，网格质量并不直接影响仿真精度！而是差质量的网格造成网格整体密度不够和分布不合理！

自由度

在网格技术中，我们经常提到自由度，其实自由度是有限元的概念，而非网格，自由度是在单元上的一种属性。一个自由度表示一个计算变量。比如热分析单元每个顶点上只有一个自由度，即温度；而位移则至少有XYZ三个自由度，考虑到旋转，则有更多自由度；电磁场至少有电场XYZ，磁场XYZ六个自由度。

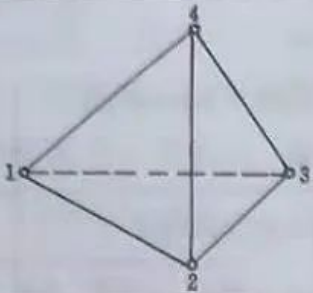
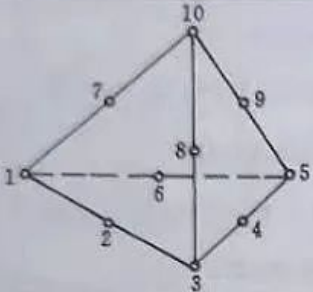
此外对于相同拓扑的网格类型，也可以根据自由度的个数来定义不同的单元，比如四面体单元在ANSYS中至少有Solid185，Solid187，Solid92等。

形函数

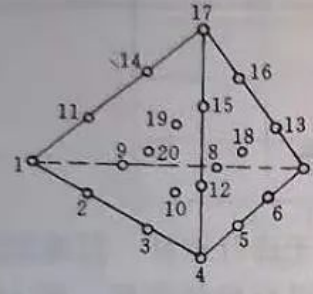
求解应力应变，以节点位移为形函数参数的单元：

一次线性单元和二次曲线单元

其中L为体积坐标，三维单元的体积坐标可以参考任意一本有限元理论方面的书籍。

4 结点 线性四面体元		u, v, w	$N_1 = L_1$ $N_2 = L_2$ $N_3 = L_3$ $N_4 = L_4$ $(L_i \text{ 为体积坐标})$	计算简单, 可适应复杂 几何形状, 单元内应力 为常量,精 度较低
10 结 点二次四面体元		u, v, w	$N_1 = L_1(2L_1 - 1), \quad N_2 = 4L_1L_2$ $N_3 = L_2(2L_2 - 1), \quad N_4 = 4L_2L_3$ $N_5 = L_3(2L_3 - 1), \quad N_6 = 4L_3L_1$ $N_7 = 4L_1L_4, \quad N_8 = 4L_2L_4$ $N_9 = 4L_3L_4, \quad N_{10} = L_4$	边界可为 曲面,可适 应有尖角的 复杂几何

三次高阶四面体单元

单元名称	单元 (母单元) 形态	自由度	形 函 数	单元特性
20 结 点三次四 面体元		u, v, w	$N_1 = \frac{1}{2}(3L_1 - 1)(3L_1 - 2)L_1$ $N_2 = \frac{9}{2}L_1L_2(3L_1 - 1)$ $N_3 = \frac{9}{2}L_1L_2(3L_2 - 1)$ $N_4 = \frac{1}{2}(3L_2 - 1)(3L_2 - 2)L_2$ $N_5 = \frac{9}{2}L_2L_3(3L_2 - 1)$ $N_6 = \frac{9}{2}L_2L_3(3L_3 - 1)$ $N_7 = \frac{1}{2}(3L_3 - 1)(3L_3 - 2)L_3$ $N_8 = \frac{9}{2}L_3L_1(3L_3 - 1)$ $N_9 = \frac{9}{2}L_3L_1(3L_1 - 1)$ $N_{10} = 27L_1L_2L_3$ $N_{11} = \frac{9}{2}L_1L_4(3L_1 - 1)$ $N_{12} = \frac{9}{2}L_2L_4(3L_2 - 1)$ $N_{13} = \frac{9}{2}L_3L_4(3L_3 - 1)$ $N_{14} = \frac{9}{2}L_1L_4(3L_4 - 1)$ $N_{15} = \frac{9}{2}L_2L_4(3L_4 - 1)$ $N_{16} = \frac{9}{2}L_2L_4(3L_4 - 1)$ $N_{17} = \frac{1}{2}(3L_4 - 1)(3L_4 - 2)L_4$ $N_{18} = 27L_2L_3L_4$ $N_{19} = 27L_1L_3L_4, \quad N_{20} = 27L_1L_2L_4$	边界可为 曲面,可适 应带尖角的 复杂几何 形状

以上图片来自于 朱伯芳《有限单元法原理与应用》

电磁矢量四面体单元

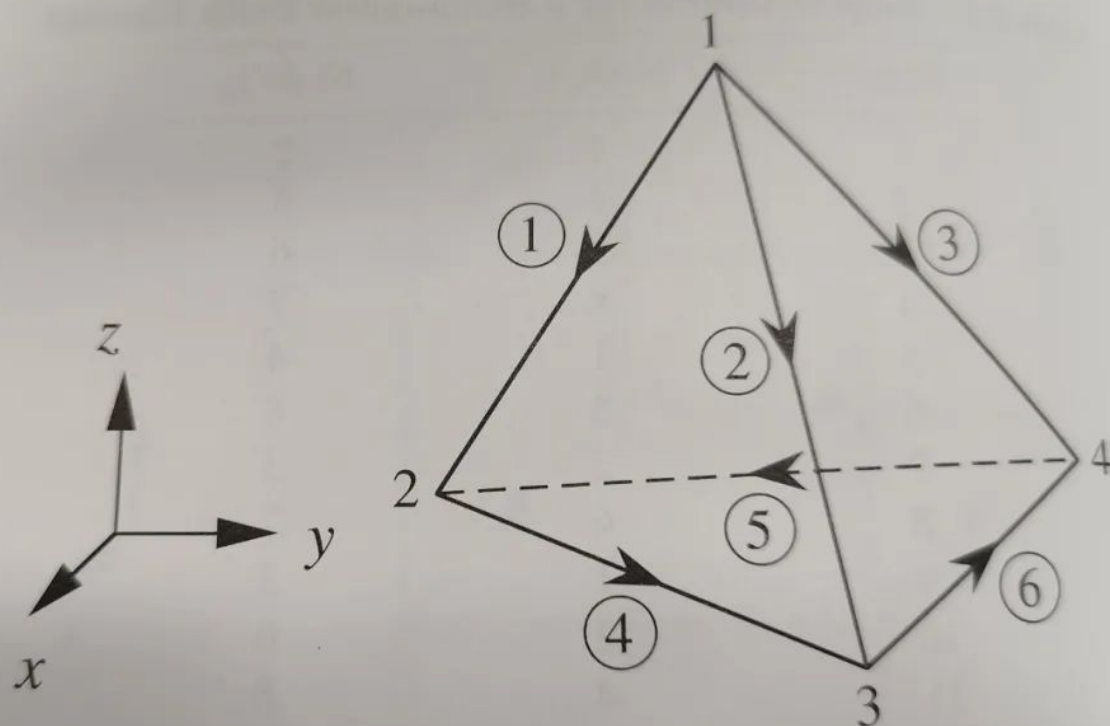


Fig. 8.12 Tetrahedral element.

多物理场仿真技术

电磁矢量四面体单元高阶形函数

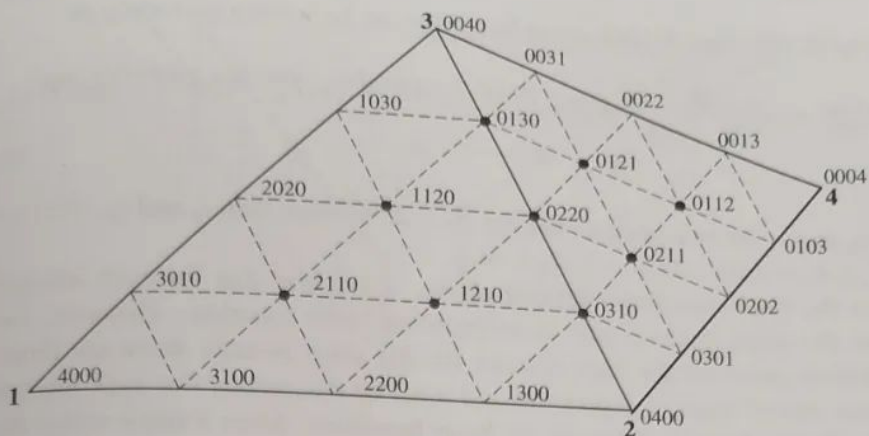


Fig. 8.28 Interpolation points on a second-order tetrahedral element for vector basis functions associated with $\mathbf{W}_{23}$ (interior points are omitted for clarity).

B. Tetrahedral Elements. For a tetrahedral element, the zeroth-order vector basis function associated with the edge connected to nodes i_1 and i_2 is given by

$$\mathbf{W}_{i_1 i_2} = \xi_{i_1} \nabla \xi_{i_2} - \xi_{i_2} \nabla \xi_{i_1} \quad (8.113)$$

where i_1 and i_2 are defined in Table 8.2. To construct higher-order vector basis functions, we arrange the interpolation points in a pyramid format and label each point using four integers (i, j, k, ℓ) , where $i, j, k, \ell = 0, 1, \dots, p+2$, and p denotes the order of the element. This arrangement is illustrated in Fig. 8.28. The p th-order vector basis functions are then given by

$$\begin{aligned} \mathbf{N}_{ijkl}^{12} &= \alpha_{ijkl}^{12} \hat{P}_i^{p+2}(\xi_1) \hat{P}_j^{p+2}(\xi_2) P_k^{p+2}(\xi_3) P_\ell^{p+2}(\xi_4) \mathbf{W}_{12} \\ &\quad k, \ell = 0, 1, \dots, p; i, j = 1, 2, \dots, p+1 \\ \mathbf{N}_{ijkl}^{13} &= \alpha_{ijkl}^{13} \hat{P}_i^{p+2}(\xi_1) P_j^{p+2}(\xi_2) \hat{P}_k^{p+2}(\xi_3) P_\ell^{p+2}(\xi_4) \mathbf{W}_{13} \\ &\quad j, \ell = 0, 1, \dots, p; i, k = 1, 2, \dots, p+1 \\ \mathbf{N}_{ijkl}^{14} &= \alpha_{ijkl}^{14} \hat{P}_i^{p+2}(\xi_1) P_j^{p+2}(\xi_2) P_k^{p+2}(\xi_3) \hat{P}_\ell^{p+2}(\xi_4) \mathbf{W}_{14} \\ &\quad j, k = 0, 1, \dots, p; i, \ell = 1, 2, \dots, p+1 \\ \mathbf{N}_{ijkl}^{23} &= \alpha_{ijkl}^{23} P_i^{p+2}(\xi_1) \hat{P}_j^{p+2}(\xi_2) \hat{P}_k^{p+2}(\xi_3) P_\ell^{p+2}(\xi_4) \mathbf{W}_{23} \\ &\quad i, \ell = 0, 1, \dots, p; j, k = 1, 2, \dots, p+1 \\ \mathbf{N}_{ijkl}^{42} &= \alpha_{ijkl}^{42} P_i^{p+2}(\xi_1) \hat{P}_j^{p+2}(\xi_2) P_k^{p+2}(\xi_3) \hat{P}_\ell^{p+2}(\xi_4) \mathbf{W}_{42} \\ &\quad i, k = 0, 1, \dots, p; j, \ell = 1, 2, \dots, p+1 \\ \mathbf{N}_{ijkl}^{34} &= \alpha_{ijkl}^{34} P_i^{p+2}(\xi_1) P_j^{p+2}(\xi_2) \hat{P}_k^{p+2}(\xi_3) \hat{P}_\ell^{p+2}(\xi_4) \mathbf{W}_{34} \\ &\quad i, j = 0, 1, \dots, p; k, \ell = 1, 2, \dots, p+1. \end{aligned} \quad (8.114)$$

多物理场仿真技术

图片来自于金建铭 《The Finite Element Method in Electromagnetics》

四面体和六面体之争

之前看过很多关于六面体和四面体的争论，大意上是说六面体要优于四面体，在网格选择上尽可能使用六面体。

从有限元理论上讲，六面体确实只需比四面体更少的网格，就能达到更高的精度。但是对于实际工程中的模型，一方面自动化六面体生成和加密都比较困难，需要更多的手工操作，耗时耗力；另一方面六面体在很多计算中有更多限制和要求，比如自锁沙漏等现象，要人为加密网格或者缩减积分，需要一定的工程经验；最后各个领域的要求不太一样，一般CFD流体计算中六面体偏多，在无法生成六面体的地方需要混合四面体网格，热仿真对网格拓扑要求不高，电磁很少用到六面体，一般的争论在结构仿真中，而结构仿真大多一般用1,2维单元，体单元在局部位置分析较多。

综上所述，其实并不存在六面体和四面体哪个更好的问题，只是对于实际的工程问题，基于工程师的经验，计算资源，仿真类型，需要选择合适的单元。

主流开源，商业网格引擎都提供了对四面体的支持。包括Gmsh, Netgen, tetgen, phg, CGAL, Vkl, meshgem, Simmetrix等等。

另外有过一份关于全球公开的四面体网格开源和商业软件的总结，并不是最新。

复制链接到浏览器中查看：

<http://www.robertschneiders.de/meshgeneration/software.html>

5. 网格参数

网格参数是网格的一个重要因素，包括网格生成策略，网格生成参数，网格属性，以及网格质量评价参数。除了之前介绍的网格生成算法，网格参数决定了网格数据的最终质量，密度，梯度等指标，也直接决定了网格能否划分成功。这也是数值计算网格和几何三角化最大差别所在。

1. 网格生成参数

网格生成参数涉及到网格生成时采用的参数，
每一个参数都有以下方面的内容：

1. 参数内容--在网格划分中需要取得的具体数据
2. 参数属性--参数的属性提供了网格的各种属性，这些属性给网格划分中各种策略处理提供依据
3. 优先级--优先级则是在和其它参数相互冲突时，如何处理

以我们最常见的网格尺寸为例。

1. 参数内容：用来指定使用的网格尺寸大小；

2. 参数属性：指定是整体尺寸，还是局部尺寸；是绝对尺寸还是相对尺寸；尺寸的单位；是否是指定几何对象或特定属性的尺寸等等
3. 优先级：一般来说，局部尺寸优先级高于整体尺寸，网格尺寸优先级高于指定的网格数量。

所有的软件网格划分参数都是以这些为基础，在上层应用中给出部分接口和参数设置，或者根据经验进行优化组合后暴露给用户少量接口，比如各种软件中的smart mesh size。

最常见的网格参数为

1. 整体网格尺寸

整体网格尺寸指定方式通常有：

- 1.1.直接指定全部整体网格尺寸；
- 2.1.指定模型对角线百分比，单一几何模型，如果不给出任何参数，通常取模型包围盒对角线一定比例为整体网格尺寸参数；
- 3.1.系统会根据模型几何数量，大小，给定一个默认参数，在此基础上，用户可以进行修改；
- 4.1.分级设置，按照尺寸比例按照区间分级设置，比如1-5级。

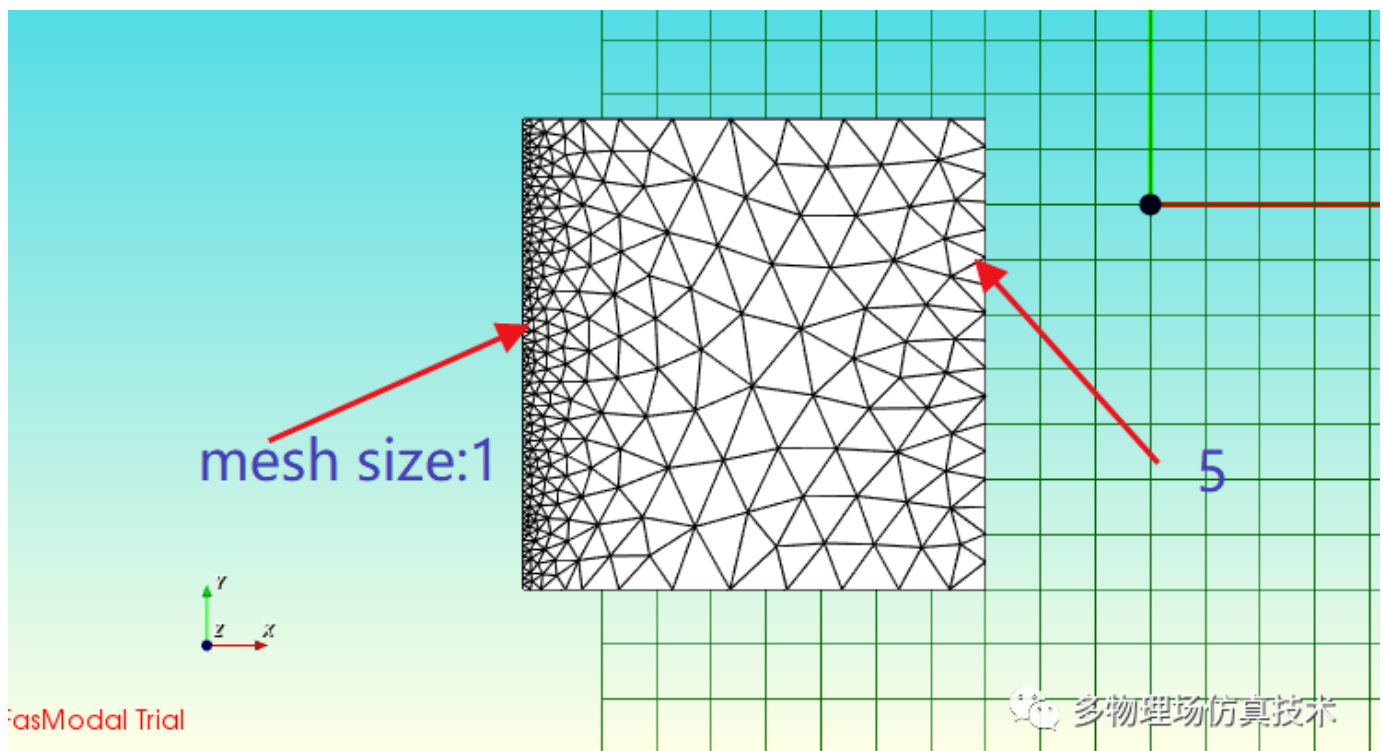
2. 局部网格尺寸

局部网格尺寸用来指定某些特定区域和几何上的网格尺寸，比如边界，激励，荷载，或者物理场梯度大的地方。局部网格尺寸的好处在于避免全局网格加密，在保证精度的前提下减少网格数量

3.增长率

网格增长率表明了相邻网格尺寸变化程度，是控制网格质量的一个重要指标。增长率为1表面网格尺寸没有变化。增长率一般控制在2，不会超过4，不同类型网格设置稍微有些偏差。

当我们设置的局部尺寸通常会小于全局尺寸，在不同尺寸网格之间存在一个过渡，这个过渡要尽量平滑，避免突变，如图。



FasModal中的局部网格尺寸设置

4.阶数控制

高阶网格需要增加新点，并且涉及到网格重编号。

5.弦高

用来指定曲线曲面离散的最大弧长与高度之比

6.跨度角

指定曲面生成的单元之间的面面角度(二面角))

7. 网格层数控制

在CFD，复合材料等领域，需要明确的设置边界上的网格层数，比如流体的边界层

8. 设置硬点 (fixed/hard point)

我们往往希望在某些特点的区域布置网格，但又不希望在几何上进行操作，可在网格中布置硬点，所谓的硬点即一定会出现在网格单元的节点列表里。

除了直接布点，还可以在边，面上有规律的设置点，比如在边上呈等比增加设点，也就是通常所说的布种子(seeding)。

9.网格加密

网格加密是在已有网格的基础上进行细化操作，涉及到网格修改，编辑，删除等操作，参考深入理解数值计算网格(8)--自适应迭代网格

2.网格数据结构

网格数据结构中除了基本的单元设计，还有一个非常重要的数据结构：

网格管理结构类

该数据结构中存储了网格的拓扑信息，还有相邻的拓扑连接信息，比如单元的相邻点线面和相邻单元，点的相邻单元，面的相邻单元，每个单元和点线面上的材料属性，边界荷载信息，历史记录，还有网格合并，加密，自由度计算等，是网格功能开发中的重点，后续在商业开发中详细介绍。

3.网格质量评价

网格质量分为**单个网格质量**和**整体网格质量**

单个网格质量控制包括：

最大角度

最小角度

最小边

最大边

Aspect ratio --最长边与最短边之比（不同类型单元有不同计算方法）

扭曲比

二维和三维，三角形和四边形，四面体和六面体分别有不同的评价指标

Aspect Ratio是比较通用的网格质量评价指标。

整体网格质量

包括数量，密度，单个网格质量的统计分类，与物理场梯度映射程度等等。这些都有比较成熟的统计方法。

4.网格策略

主要涉及到采用什么样的方法生成网格，这个一方面和模型几何有关，和仿真业务也紧密相关。

比如对于规则的长方体，可以采用映射法直接划分六面体，对于异形几何，只能采用自由划分四面体。一般只有通用有限元软件中会涉及到网格策略的选择，专业软件中会把网格策略设置好，只在高级选项中出现，一般情况无需用户干预。

6.理解高阶单元

“由于高阶数值计算复杂性，目前大部分商业软件网格都是线性单元（0阶）或者二次单元（一阶）。实际上高阶网格（也就是常说的P单元）在处理复杂几何上更具优势，适当的设置网格参数，只需更少数量的网格，无需迭代，就能获得更好的数值计算解，期待未来在高阶领域有更多实际应用。”

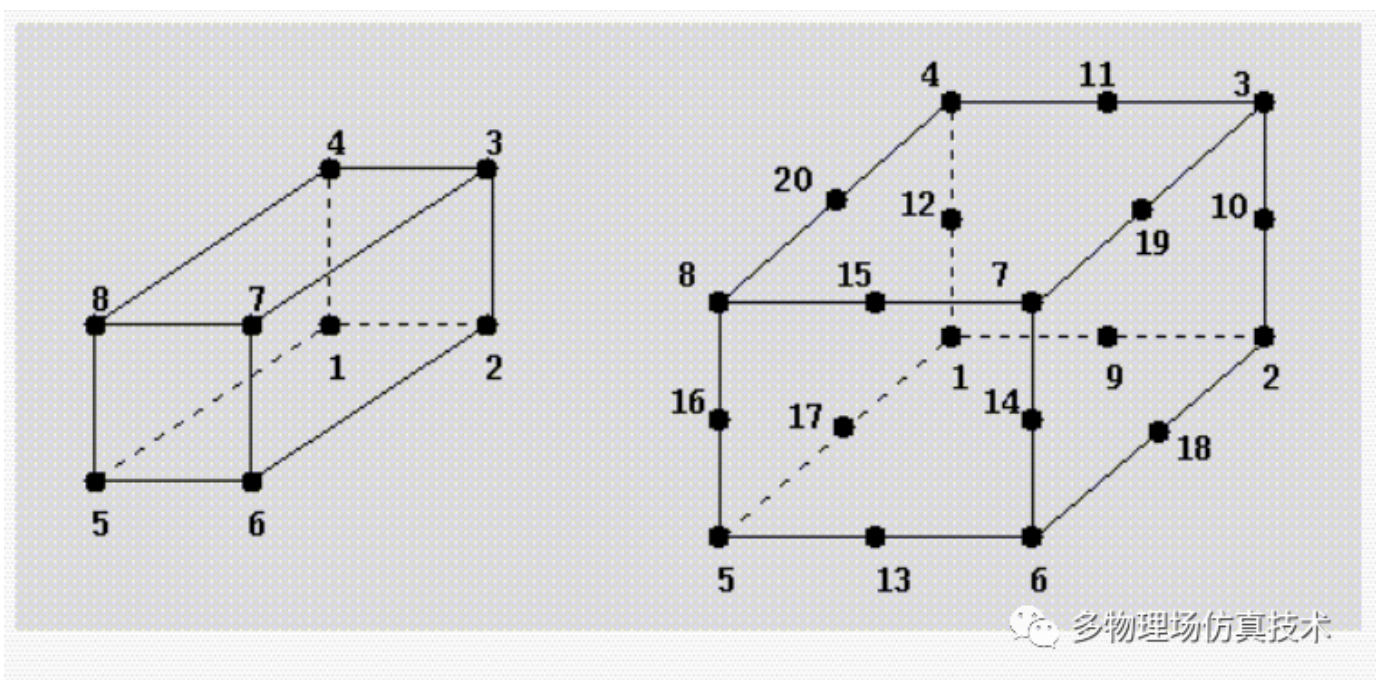
关于高阶单元，之前的文章有过很多介绍。当网格密度偏小，计算精度不够的时，可以通过两种方法提高精度，一种是加密网格（h单元方法），另一种是提升网格的阶次(p单元)，不再多描述。

下面主要介绍高阶单元的一些特点和实现

1.单元划分和生成

0阶单元，1阶单元的网格几何信息并没有改变，也就是说原来的点，边，面的位置没有发生改变。区别在于高阶单元的边或者面上点的增加。

对于网格划分来讲，难度并没有变化，区别只是在已有单元的边上再加一点。如果单元编号约定规则，把线性单元变成高阶单元在求解器阶段进行也可以。但考虑到模块解耦和功能独立性，一般还是把高阶网格的创建放在网格生成阶段。

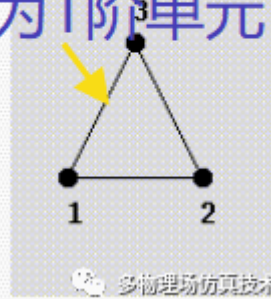


左为线性六面体单元，右为曲线六面体单元

在实际应用中，发现有些软件将线性单元（即三角形单元中仅有三个顶点）**描述为1阶单元**，有些描述为0阶单位，在沟通中造成了歧义。

为了避免歧义，统一将顶点仅在边两端的单元（三角形三个顶点）称为**线性单元**，将边中间有点的单元称为**曲线单元**。

有些软件定义为0阶单元，有些定义为1阶单元

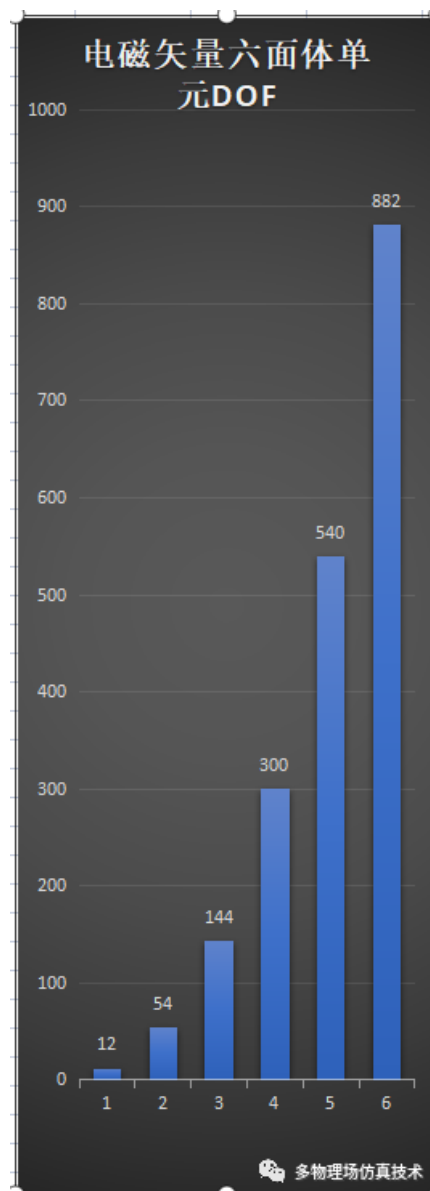


2. 高阶单元计算精度和计算资源

高阶单元之所以能提高精度，简单理解就是高阶单元使用的多项式变量次数更高，表达式更丰富，有能力更精准的反应单元内部物理量的变化。

大部分物理场的本构方程为二阶偏微分方程，或者可以转为二阶偏微分方程。高阶单元的基函数和形函数偏导后仍然能保证单元内部的非线性！

精度的提高的代价是计算量的快速增加，尤其是矩阵规模的增加



电磁六面体单元自由度数目随阶数变化

3.高阶单元实现

高阶单元实现有两种，一种是直接使用点插值的基函数方法，另一种是使用结构化组合基函数。

两种方法都基于线性单元，区别点在于第一种方法由单元上的一系列点来定义，高阶单元由增加插值点构成，因此在各个维度都有较好的独立性，且生成的刚度矩阵在预条件处理上有优势。多项式的每个表达式的系数有较好的物理含义，并且在边界上有较好的表达。不同阶数的单元有相同的统一表达式，代码更方便实现。

结构化组合基函数是由在低阶基函数的基础上乘以或加上新的函数。这样做的好处是在同一模型中可以使用不同阶的基函数，实现上灵活，但也增加了复杂度。

在实际研发中，更多的是采用第一种方法插值基函数的高阶单元。

混合单元

有限元等数值方法是在全局进行计算，所以可以在网格不容易生成的地方采用高阶单元，其它地方使用低阶一般单元，发挥各自的优点。

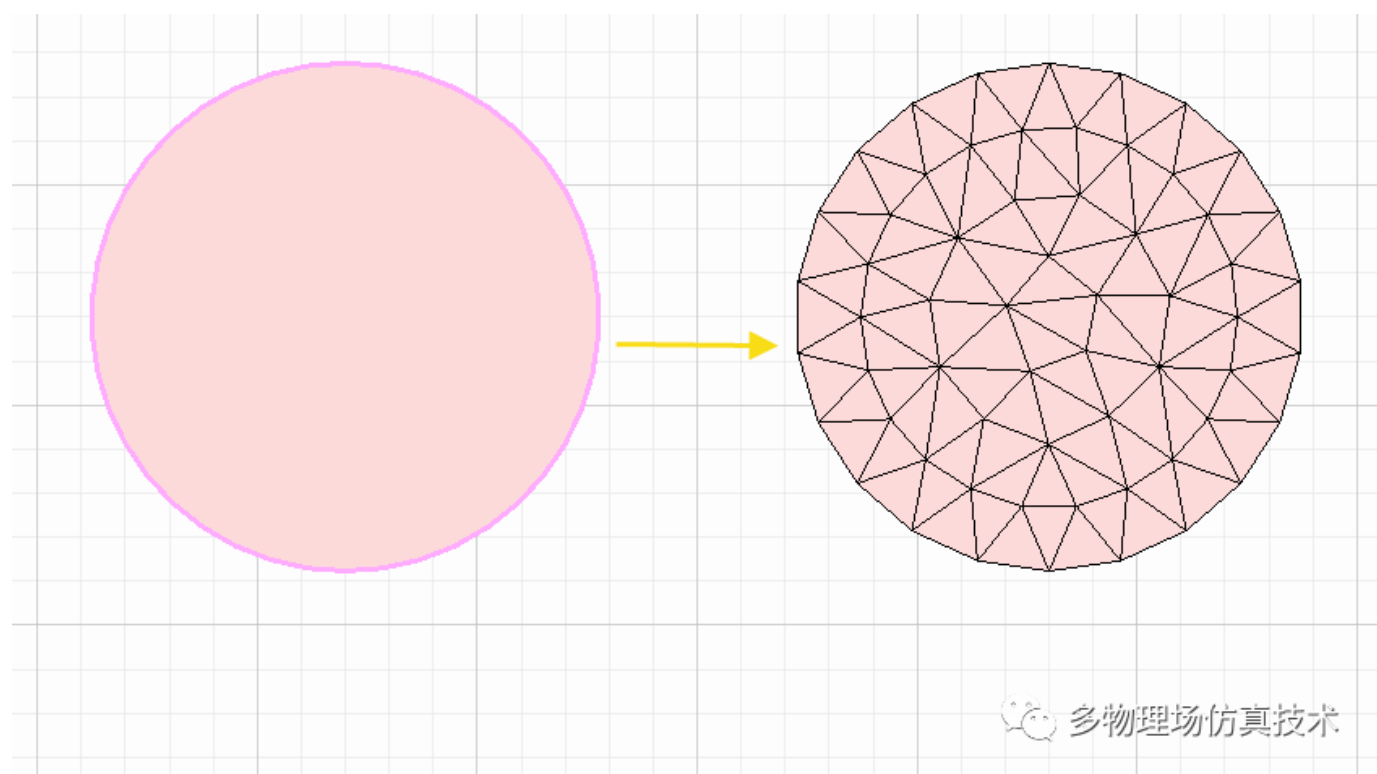
大部分有限元软件都支持高阶单元，在工程应用中，碰到网格划分困难，网格质量差，使用低阶单元收敛困难，且计算资源没有瓶颈的情况下可以尝试使用高阶单元。

7.几何与网格

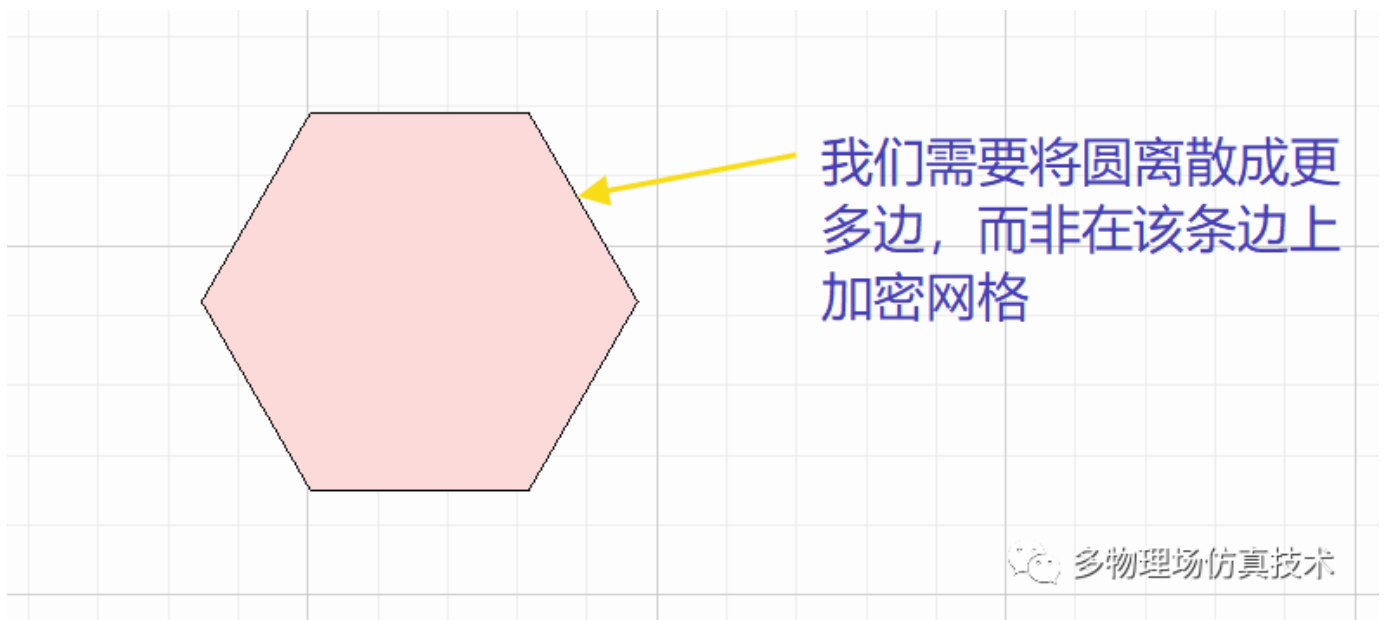
我们知道网格的输入为几何数据，几何数据可以是参数化数据，BREP结构数据，或者其它任意形式定义的数据。

1.网格加密和几何

在划分网格的时候，需要将其离散化分成多段。如图，圆被离散成多条直线。如果在网格加密过程中，需要继续对圆弧进行加密时，没有原始几何信息，我们将只能加密在22条直线上，而无法将圆离散成更多的边。实际应用中，对圆弧网格尺寸加密是要求将其离散的更密集，而非在离散的边上加密。



多物理场仿真技术



2. 共形网格与几何

在三维接触碰撞，CFD，复合材料，多物理场耦合领域分析中，需要将不同属性，不同材料对象放在一起分析。在网格上需要将之前对象属性保留。这就会碰到一个难题：如何处理不同对象接触部分的网格。

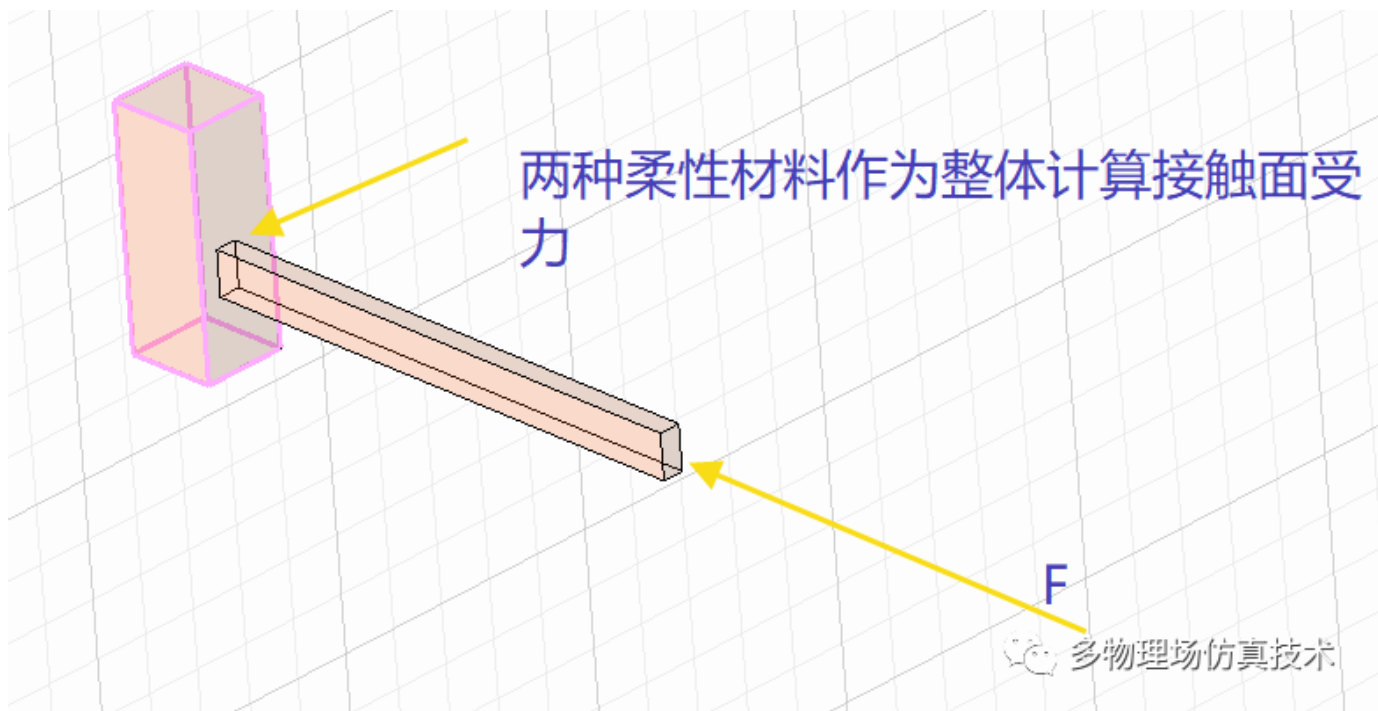
通常做法有以下几种：

- 1.将各个对象作为独立个体，将其中一个作为主要分析对象，其它对象简化；
- 2.各个对象分别划分网格，设置不同属性，将可能接触部分的网格标识出，让求解器去处理；
- 3.在可能接触部分进行网格平滑，保证点和边能完全重合，但不保证接触对象完全一对一；
- 4.划分点，边，面完全一对一的网格，也就是通常所说的共形网格 (conformal mesh)

其中第四点共性网格由于保留了物体原有属性，且能把所有对象在同一整体刚度矩阵中分析而成为大多数仿真软件的选择。

在几何上，设置不同的材料，属性是很容易的，但在共性网格划分上，由于通常是先生成面网格，在面网格基础上再生成体网格，造成了各种属性的传递困难。

以材料为例，当两种不同材料物体接触一起分析时，划分面网格阶段，我们需要区分每个面在哪种材料上，按照一般规则，我们可以定义面的法向量的方向为材料的方向，这也就需要在几何中非常明确规定实体上的面的方向，而不能毫无规律！对于接触部分，需要在几何中明确计算出具体位置，然后传递给网格引擎。



3. 几何清理

做一个简单类比，求解器是发动机，网格是汽油，几何则是原油。网格生成则是炼油的过程，高质量的原油可以炼出好的汽油。

为了生成高质量的网格，干净正确的几何输入是必要前提，好的几何也有助于后期网格的调整，编辑和优化。

相信很多工程师有清理几何的痛苦经历。设计生产的CAD几何模型往往不适用直接仿真，比如不必要的文字，几何不连续，细小边，细缝，物体干涉，倒角，圆孔，自由对象，重复对象，不同格式之间的转换带来的几何拓扑错误，容差错误等等。拓扑错误，几何不封闭，方向错误，连接顺序相反也是常见的几何错误。

4. 虚拟拓扑

虚拟拓扑是针对几何到网格过程中繁琐的操作进行的一种简化操作。

场景1：需要在实体的一个面上某一小块区域A而非整个面进行边界或荷载设置，通常做法需要创建新的面几何把这一块区域覆盖。另外一种做法可以将区域A设置为虚拟拓扑，直接传给网格引擎，让网格引擎单独处理。

场景2：需要在很多个小的面上进行荷载和边界设置，常规做法是需要一个个细小的面选择，然后进行设置，可以将这些面设置成一组，作为一个单独面使用，大大简化操作。

针对虚拟拓扑，可以设置过滤条件，让很多手工操作半自动化或者全自动化，尤其是大模型，能减轻工程师的负担，也减少操作失误的几率。

5. 几何曲线曲面处理

曲线曲面的网格划分

在几何中，曲线曲面通常分为三种：参数，显式和隐式。参数形式往往有严格的方程定义；显式几何明确给出点的位置或面片；隐式往往以表达式的显示出现。

仍然以圆为例

参数形式为：

$$x=a+r*\cos\theta$$

$$y=b+r*\sin\theta$$

参数表达式很容易满足连续性要求，满足1,2阶偏导要求，在网格端很方便拿到参数。

我们可以显式给出圆周上16个点，点顺序相连，给出圆的近似表达。

显式几何的好处是网格无需做过多处理，缺点是需要明确几何的正确性，比如16段圆弧才能满足仿真精度，但几何只给出8段，那么在网格端再如何处理也是无法达到精度要求。

隐式形式：类似公式 $x*x+y*y=A$

隐式表达的好处是给出任意一点坐标，我们能利用表达式方便判断出它的相对位置。隐式几何精准将几何信息传递给网格，对网格处理上也提出了更多要求，在曲线曲面离散上，要取到几何的本身属性以及控制参数，另外还要考虑网格参数，如果出现冲突，要选择合适的算法处理，实际中只有在特定场合才使用。

6. BREP结构和网格

主流的三维几何内核表达为BREP结构，利用这种结构，网格很容易拿到BREP结构中的几何和拓扑信息，方便进行后续网格划分。

为了显示需要，一般几何内核通常也具有三角化的功能，几何离散成的三角形称之为“面片”，非网格，大部分显示引擎底层都使用三角形来渲染对象，因此2维、3维几何都需要三角化（称之为“面片化”），比如一个长方体共有6个面，每个面需要离散成两个三角形，总共12个三角形。这12个三角形我们称之为12个“面片”，而非网格！一般情况下“面片”质量很差。“面片”有两个功能：一是用来做显示渲染数据；二是可以作为网格划分的输入数据，用来生成面网格。

也就是说，网格划分既可以直接拿几何的数据结构，也可以拿三角化之后的数据，可以根据实际需要做选择。

7. 变形网格(morphing)+高阶单元

morphing技术是网格划分中的一种常见方法。即初次划分网格后，几何发生变化，网格也跟随发生变化，但网格只是点坐标跟随变动，而拓扑(连接性)不发生变化。morphing在参数化几何设计中非常有用，由于不需要重新划分网格，提高了设计效率。借助于高阶

网格单元，可以在几何设计阶段快速完成高精度仿真，也是generative design的基础性技术之一。

8.自适应迭代网格

在前处理网格划分中，如果网格数量偏少，在物理场变化大的地方，网格内部不足以表达这种变化，从而导致计算偏差。类似于用多边形模拟圆形，线段数量越多，越接近圆形。

网格加密通常有以下几种标准：

1. 根据几何特征，比如在曲面，曲线，孔边等加密。这种加密依赖于几何拓扑信息，但并不准确，比如在尖角处网格加密反而得出错误的结果。
2. 根据业务特点，比如流体在边界层，电磁仿真在金属部分需要加密，在初次划分网格的时候就可以进行设置；
3. 根据两次仿真所得物理参数结果，比如在应力/压力/速度/温度变化梯度大的地方；
4. 根据两次仿真业务数据进行加工，比如HFSS根据S参数误差，能量误差，CFD中根据压力，速度误差。

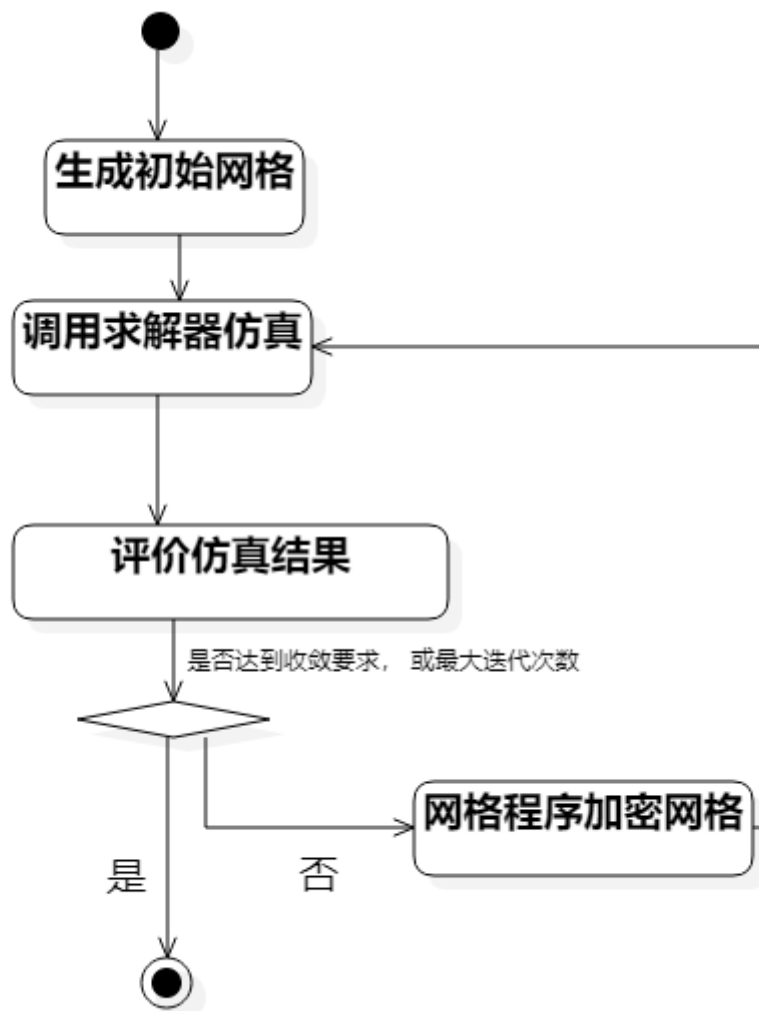
1.加密策略

加密策略涉及到什么时候加密，如何加密，加密效果如何评价等等。

按照加密的阶段，可以分为两种，**一种是在仿真之前加密，一种是在仿真计算后加密。**

仿真前加密，是利用对象已有的特性，包括几何特性和仿真属性，几何特性比如孔边，大曲率曲线曲面，仿真属性比如接触单元，有电流经过的金属，明确确定物理场会发生突变的单元，确定的荷载，边界条件位置。仿真前加密一般基于已有仿真经验，加密单元的位置比较准确，但是仿真结果的准确度很难确定。

仿真后加密就是一般性自适应网格加密，UML活动图如下：



多物理场仿真技术

其中如何“评价仿真结果”是很重要的一环，一般的做法是“根据两次仿真所得物理参数结果，比如在应力/压力/速度/温度变化梯度大的地方”，也有根据单元内部数值计算误差的统计误差作为评估标准。每种物理场的评价指标和方法都不尽相同，但总的原则都是相似的。

2. 加密算法和参数

在网格加密流程中，求解器会给出需要加密的单元。这就涉及到两个问题：

1. 如何使用加密算法
2. 加密尺寸如何确定

给出需要加密单元，不考虑效率，其实最理想的方法是重新划分网格，在给出需要加密尺寸的地方设置局部网格参数，但这种方法比较耗时，而且如果加密单元少的话，比较浪费资源。

常用的做法是在已有网格的基础上进行加密，直观的方法是将原有网格拆分成更小。这种做法理论可行，但实际应用中会碰到很多问题，首先是网格质量无法保证，以三角形为例，一个等边三角形，正中间加点分成三个三角形，每个三角形的最大角度成为120度的钝角三角形；其次网格数量很难控制，二维三角形单元必须分为三个，而三维四面体则必

须分为9个，如果加密网格尺寸只要加密到原尺寸的0.8。这种直接拆分法无法满足需求；最后直接拆分网格，导致相邻单元的尺寸梯度突变，反而不符合物理场的平滑过渡的要求。

加密尺寸可以由求解器给出，如果求解器只给出了加密单元，而没有加密尺寸。就意味着，加密尺寸需要在网格自适应迭代过程探索，增加了网格加密和求解迭代的次数。

在给定加密单元和尺寸的前提下，通用的方法还是类似波前法，将所在区域挖空，然后根据加密尺寸局部参数要求，重新生成新的网格，新网格生成后，再对周围网格进行优化微调。在该过程中，涉及到了网格的删除，编辑，再生成，再编号等一系列操作，开发难度要高于初始生成网格。

3.收敛问题

收敛问题涉及网格划分中经常碰到另外一个问题：

网格是否越密越好？

答案是否定的

网格不可能无限度的加密，因为过度加密即使没有加密上的误差，也会引起数值计算上的误差。比如建筑的仿真尺度以米为单位，无限加密到毫米尺度时，如果原始单位为米，则会出现数值计算上的误差，也就是说网格加密并不是因为物理场本身的变化，而是数值误差引起的变化。

如果网格已经满足了仿真精度的要求，再加密网格并不会增加仿真结果的精度，除了徒增加计算资源外，还会造成计算收敛结果的震荡。

网格收敛同样也要符合模型的计算规则，比如针对结构有限元中的沙漏和自锁现象，网格加密的力度要比较激进，而非普通的渐进式的加密。

网格加密的收敛是比较适合用AI算法进行，因为实际应用中，每种产品的结构基本上不会有太大的变化，这就使得网格的参数能较好的进行统计计算；网格的加密也是迭代生成，天然适合AI算法的训练；当所有网格参数训练完善时，自适应加密网格也就不需要了，直接在初始网格阶段敲定所有参数。

4.商业软件的现状

很早以前网格的划分依赖于工程师的经验，很多时候需要在算出结果后重新划分网格，来回折腾。现如今自适应加密网格已经成为商业仿真软件的标配，如果你使用的软件还没有使用到自适应网格加密，可以试试其它软件。

9.商业开发

了解公众号的朋友都知道，本公众号既不是流量自媒体，也不做一般的科普和工业软件使用教程。**公众号的所有文章都是围绕工业软件研发底层技术而谈！**

网格生成作为工业仿真软件的一项重要的基础性功能，必然也是研发的重点和难点。在前期介绍网格的基础上，本文就简单聊一聊网格相关功能的商业开发。

有朋友可能会有疑问，商业开发和一般的开发区别在哪里？

其实商业开发的难点主要在于解决实际工程问题：一方面基本的功能都要有，另一方面要能满足工程中各种稀奇古怪的需求，有些可能是反常识的工程问题。在一般功能的基础上，对网格生成的性能，稳定性，兼容性都有更高的要求。

比如我们常用的几何内核为ACIS和Parasolid，网格引擎需要能直接读入ACIS和Parasolid的数据，避免二次导入的数据损失；现在的模型越来越大，设计越来越复杂，网格要求有更精准的参数控制来处理大模型，满足性能要求，生成符合实际需求的网格；针对逆向工程和AI建模等新趋势，网格模块要能处理扫描出的大量无用数据和脏数据，有能力进行数据清洗等预处理工作；此外网格智能加密和稀疏，支持混合单元，网格拆分组合，网格修复，虚拟拓扑，并行网格划分等也一直是应用领域的刚需。

从功能需求看，网格的开发分为两部分：

1. 网格引擎开发；
2. 业务应用开发；

网格引擎提供底层网格划分技术，包括标准几何导入识别，实现网格划分算法，网格优化，生成结果网格数据，导出标准中间网格格式等等，是基础性功能研发。目前这块整体看开发资源相对较少，开源的比如getgen，tetgen，gmsh，CGAL都有提供，可以作为学习和参考的资料。综合性的功能比较全面的商业组件选择余地小。

业务应用开发在网格引擎的基础上，实现业务的实际需求，比如自动设置合理材料，边界等信息，最终生成求解器需要的网格文件。这块直接和业务挂钩，是大部分网格开发的重点。

很多专业软件因为对网格的需求比较集中和明确，因此开发往往同时包括了网格引擎和应用业务开发。**这也是未来仿真软件开发的趋势，即根据自己业务需求开发网格功能，掌握和开发底层网格生成算法！**

本来打算在商业开发中讲讲网格引擎的设计和实现，以及业务应用开发。发现内容太多，泛泛讲没有深度，也没有太多指导意义，后续有空再聊。

从十多年的工业软件研发经历看，网格作为基础性功能，技术更新十分缓慢。更新主要集中在性能比如多进程，分布式计算，几何接口扩充，实际仿真业务需求等。可以预见，未来也不会有太多更新。相比求解器，几何内核，网格功能相对独立，且内容比较集中，没有太高的瓶颈和工作量，更多是算法上的实现和完善（注意：只是相比几何内核和求解器）

从国外大环境看，各大厂商也都在加紧布局工业软件的底层核心业务，换句话说，以后在全球都很难买到底层应用模块，甚至使用可能都会比较困难。

目前国内工业软件发展如火如荼，想在这块有所作为，网格功能要早早介入底层开发，进行长期技术上的积累。

相比之前的网格入门介绍，“深入理解数值计算机网格”系列介绍了更多工业仿真软件中的网格细节，但是相比网格功能算法的具体实现，实际应用，仍然只是皮毛。后续有空会更多介绍网格引擎的设计，算法实现，网格和业务的结合以及在工业实际中的应用。

一篇文章入门网格划分

深入解析HFSS自适应网格加密策略

如何构建世界级的中国工业仿真软件

文章已于修改

工业软件“断供”是一把“达摩克利斯剑”吗

原创 邓子平 多物理场仿真技术



最近在和同行沟通中，谈及最多的问题就是如果国内工业软件断供怎么办？有人更是声称：“断供”是一把摆在国内企业的“达摩克利斯剑”。

众所周知，国内使用的各种

CAD,CAE,EDA,CAM,CFD,TCAD等工业软件包括数值计算MATLAB之类的软件都是以欧美国家为主的公司开发的，据统计这个比例高达90%以上，这些软件已经渗透到了国内技术研发的方方面面，从高校到企业，科研单位以及军工企业。而且种种迹象表明，未来各种工业软件“断供”的可能性会越来越高。

如果美国工业软件“断供”全面发生，可以预见到以下后果：

1. 国内的软件分销商迅速消亡，软件分销商的作用是将国外的软件卖到国内，因为工业软件的特殊性，价格都非常昂贵。分销商在推动工业软件在国内的普及方面还是有积极作用的，但是也因为忽悠的事情做的太多，没有技术研发投入，一旦正式“断供”，这些分销商将失去价值。
2. 国内整体科技研发进度将变缓，很多企业科研单位将迎来阵痛。工业软件覆盖范围很广，航天航空，能源，交通，医疗，制造，等各行各业。华为，哈工大已经有所经历，事实上，软件断供在华为，哈工大并没有全面发生，国外可选的软件仍然很多，尤其是欧洲的软件企业。
3. 最坏的情况是欧洲国家企业也断供，这也是最不愿意看到的。欧洲在工业软件研发方面有很强的实力，德国西门子，法国达索，但是一旦欧美联合起来断供（欧洲收购的美国软件公司仍然属于美国软件），国内将面临无工业软件可用的局面，尤其在工业软件仿真领域。
4. 在无软件可用的情况下，只能寻求开源软件和国内软件。对国内的软件厂商其实是一个不错的机会。但是反过来如果“断供”取消，国外软件再次廉价入市的时候，国内的软件厂商是否有竞争的实力，也是需要考虑的问题。

一个很关键的问题：以上情况发生的几率有多大？

笔者认为，美国的“断供”其实是个昏招，损人不利己。软件的研发成本主要在固定资产和人，所以成本几乎是固定的，多销售一个license就多一份收入，简单说软件卖给国内就是纯利润；在当前环境下，部分软件断供，马上就会有新的软件顶上来，即使全面断供美国软件，欧洲各国也不一定唯美国是瞻，毕竟工业软件并不属于高精尖限制性技术。再退一步即使欧洲也断供，国家层面也不会不管，无非是多花点钱购买。即使全面断供，只要有投入有市场，国内的工业软件就会慢慢赶上来。从外商角度讲，断供不仅自己损失收入，还会培养一批竞争对手。

所以，“断供”的过程实际上也是各方博弈的一个过程。和上世界五六十年代的技术封锁还是有本质的区别。

长期看，中国发展上升趋势不可避免，对于国外厂商来说，与其压制，不如携手合作。对国内来说，要实现最大化利益，仍然需要加强对工业软件的投入和扶持力度。毕竟在当前大环境下，独立自主，自力更生是永远不会错的。

回到正题：工业软件“断供”是一把“达摩克利斯剑”吗？

笔者认为不是，但对于国内软件分销商来说，“断供”的的确确是一把可以随时落下的“达摩克利斯剑”。



写给仿真软件研发的“一篇文章入门”系列（终）

原创 邓子平 多物理场仿真技术



“一篇文章入门”系列不再更新，点击[绿色标题](#)可打开链接。

一篇文章入门系列主要针对工业仿真软件研发测试人员，也可作为相关从业人员和学生，以及对仿真软件开发感兴趣的读者。

所有文章来自于作者长年从事工业仿真软件研发过程中的学习，认识和理解，不一定完全正确，有错误的地方，欢迎您指正，如果您有好的想法和建议也欢迎来信或留言。

仿真软件开发工具介绍大全（10/29/2020更新）

主要介绍了仿真软件主流开发工具以及底层开发库，包括几何，网格，图形图像，GUI，测试，可持续交付等工具，仿真软件较全面的参考。

1.一篇文章入门计算热学

介绍热学相关基础内容

2.一篇文章入门多物理场有限元

多物理场有限元方法科普文，适合刚开始了解多物理场，对有限元知识有一定了解的读者

3.一篇文章入门三维几何内核

介绍主流三维几何内核原理，了解CAD三维建模底层知识。

4.一篇文章入门仿真网格划分

全面介绍了仿真网格相关知识，开发，测试和产品工程师都值得一读。

5.一篇文章入门计算流体力学（上）

6.一篇文章入门计算流体力学（下）

计算流体力学CFD相关入门理论知识

7.一篇文章入门计算电磁学(3/27/2020)

计算电磁学相关入门理论知识

8.一篇文章入门高频电路电磁有限元（上）

9.一篇文章入门高频电路电磁有限元（下）

EDA行业针对电路的有限元方法介绍

10.一篇文章入门HPC

仿真高性能计算相关入门理论知识

11.一遍文章入门前处理开发

分析了Cubit的前处理开发相关内容

12.一遍文章入门仿真后处理

后处理开发相关简介

13.一篇文章入门计算机辅助优化

介绍优化算法在仿真行业的应用

14.一篇文章入门大规模方程组求解

求解大规模线性方程组的常用方法以及各种商用开源工具库

15.一篇文章入门计算机辅助工程CAE

笔者根据自己的经验和理解，简要介绍计算机辅助工程CAE

16.一篇文章入门边界元法

BEM(Boundary Element Method) 边界元方法简介

17.一篇文章入门求解器开发设计模式

介绍了软件研发中常用设计模式在求解器中的应用，并根据求解器数据的特点，提出了一种新的属性模式。

18.一篇文章入门计算声学

计算声学相关入门理论知识

19.一篇文章入门求解器模型降阶方法

求解器中求解大规模线性方程组的一类方法介绍

20.一篇文章入门时域有限差分方法 (FDTD)

电磁的FDTD时域有限差分方法介绍

21. 一篇文章入门离散元方法

简要介绍离散元(Discrete Element Method)方法相关知识

22.一篇文章入门格子玻尔兹曼方法

介绍近年来CFD中无需网格的格子玻尔兹曼方法

23.简要介绍计算光学

简要介绍计算光学相关内容

24.一篇文章入门仿真软件性能优化

仿真软件开发中如何有效提升软件性能

25.一篇文章了解云端仿真

云端仿真相关内容介绍

26.关于计算材料学

计算材料力学简介

27.一篇文章入门无网格方法(1)

介绍数值仿真中的无网格方法

28.一篇文章入门多重网格方法

求解大规模线性方程组的一种方法

所有文章尽量避免各种复杂的数学公式表达式，用通俗易懂的方式介绍仿真软件研发内容，涉及了我们经常提到的

CAD/CAE/EDA/CFD/TCAD/CAPP/CAM等工业软件领域，是国内研发工业软件不可多得的科普系列文章。

转发请注明出处

我所理解的“工业软件”

原创 www.cae-sim.com 多物理场仿真技术

收录于话题
#工业软件杂文系列

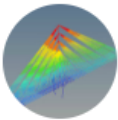
17个



最近几年，“工业软件”一直在国内升温，中兴，华为断供事件更是加剧了“工业软件”在国内的热度，在因新冠病毒疫情中美交恶的背景下，国内高校的MATLAB禁用事件似乎将“工业软件”的讨论推到了一个高峰。国内的很多媒体也惊呼：国内流行的各种软件，从操作系统，到各种设计软件，仿真软件，科学软件，几乎都是美国开发的！而敏感的投行似乎嗅到政策方面的信息，接触的一些投资人也在咨询相关信息。

从商业角度看，这次MATLAB禁用问题不大：卖家有卖和不卖的权利，买家也有买和不买的权利。但问题就在于卖家长期给你又好用又便宜的货，突然没有征兆的断货，让买家措手不及，实属非常不厚道！典型的店大欺客。

笔者一直以来的观点是：国内长期依赖国外软件的状况不可能一直持续，早晚有摔跟头的时候，只是时间和摔跟头严重程度的问题。



多物理场仿真技术

2014-11-8 23:09 来自 微博 weibo.com

几乎每个从事IT研发的人开始都会有个梦想：开发国产的**操作系统**（非Linux改良）。但随着从业经历增长，会逐渐发现这个梦想多么的不靠谱，也会越来越麻木：反正Windows用的挺好，开发那玩意儿干嘛。如果有一天全国上下windows机器全部蓝屏，敌人飞机导弹乱飞的时候，oh，梦想，但愿杞人忧天~~😂

阅读 7146 推广

1

1

 多物理场仿真技术

“工业软件”通常包括设计，验证，仿真，优化，加工，运营，管理等全套软件，是现代工业的基础性工具。离开工业软件，工业尤其是制造业就只能停留在20世纪五六十年代的水平（那个年代所有的图纸全部手画）



笔者关于广义上“工业软件”的理解有以下几点：

1. “工业软件”是工具，是辅助，而不是主角。看过讲述中国第一颗原子弹电影《横空出世》的人应该对里面一个场景印象深刻：一大群由士兵荷枪实弹保护的科学家在大房间里把算盘打得震天响。常常为了一个公式，算盘要打好几天，最长的一次是三个月！如果有人带着一台装有类似MATLAB, Mathematica数学软件的电脑穿越回去，一个公式几分钟就能算出来，而且还不会出错！但是没有软件，原子弹照样造了出来，而现在的核试验已经不需要实际引爆原子弹，借助工业仿真软件和高性能计算机，核爆可以在电脑上进行模拟！
2. 我们所接触到的工业软件只是帮助设计，而真正设计的主角还是人。比如我们需要设计一架飞机，软件只是按照人的意图去实现，仿真也是按照用户的模型来计算，软件并不会告诉你如何设计一架好的飞机。而真正涵盖工业知识的软件，比如核

爆，水下潜艇冲击爆炸之类的仿真软件，是不会拿出来销售的！再比如我国的高铁动车发达，同样我们也可以开发出专业动车设计软件，但是涉及核心知识机密，无法商业化。

3. 工业软件的价值在于提高工作效率以及稳定性可靠性。其实前面讲的两点也在说这个。工业软件的难度其实是远远高于互联网的，互联网所谓的上亿数量级别的并发在真正的工业软件面前都是小儿科，超大规模的模型不仅硬件上需要超级计算机，而且对算法的稳定性，可靠性，容错，都有非常高的要求。错之毫厘谬以千里的事在工业软件可谓司空见惯。1996年坠毁的阿利亚娜5火箭，其失败原因就在于直接移植阿丽亚娜4的代码，没有对软件进行足够的回归测试。

CAD--Computer Aided Design

CAE--Computer Aided Engineering

EDA--Electronic Design Automation

CAO--Computer Aided Optimization

CFD--Computational Fluid dynamics

TCAD--

Technology Computer Aided Design

CAM--Computer Aided Manufacture

CAPP--

Computer Aided Process Planning

下面再讲讲狭义上我们经常可以接触到的“工业软件”，对应于

CAD/CAE/EDA/CAO/CAM/CFD/TCAD/CAPP等等：

1. 工业软件需要参与实际工程应用。既然是“工业”软件，就一定要能解决工业界的实际问题。否则即使求解器做的再好也只是纸上谈兵。
2. 工业软件一般始于技术驱动，成长于用户迭代，成功于商业模式。技术是工业软件的根本，这个不用多说，工业软件的技术门槛较高，融合多门学科，开发周期也比较长，前期投入大。硅谷的很多商用工业软件在早期都有风投介入。在和用户的迭代，解决实际问题过程中不断完善软件，最终通过各种license付费策略赚钱。在IP保护好的环境下，这种模式比较可行，也容易成功。再不好也可以选择被大公司收购。
3. 开发工业软件要积极融入全球生态圈，避免闭门造车。全球化经济的背景下，应该谋求全球合作，利用已有资源，但是核心技术包括业务只能自己开发。
4. 低端的工业软件没有任何市场。在制造业，我们通常有高端，中端和低端。但在工业软件市场，只有高端，没有低端。换句话说不会有人买低端的软件。

5. 国内工业软件虽然喊的震天响，但是少有人入行。其根本原因还是在于产品附加值低，无法获得市场，就如同早期的华为手机和苹果，在同等价格下，用户一般会买苹果。手机有广大用户做市场，工业软件的市场相比之下用户少之又少，而其中仿真的份额又只有设计的5%左右。门槛高，投入大，周期长，即使出来也难比成熟产品，自然商业利益无法保证。
6. 各种软件断供对国内相关公司真的是一个不错的机遇。因为长期趋势非常明显，美国在科技上利用自己的优势，对中国的打压将是全方位的。在商用技术领域，我们的工业软件需要有自己的核心产品，困难固然有，摸石头过河的也不少，长期看，机遇和风险并存。阿里早年提出去“IOE”化，即去掉IBM的小型机、Oracle数据库、EMC存储设备实际上也是去美国化产品，在当时遇到的阻力和风险也非常大，但是现在来看也比较成功。

关于作者：

超过15年

CAD/CAE/EDA/CFD/BIM/TCAD/CAO/HPC等工业软件研发经历，现仍主要从事相关软件架构研发工作。

中国工业仿真软件，缘何不能“弯道超车”（更新）

原创 www.cae-sim.com 多物理场仿真技术



前言

这篇文章最早动笔是在2017年，初衷是针对在2016年中兴第一次被制裁的时候，不少媒体包括自媒体高呼在高科技领域中国需要“弯道超车”，摆脱对美国科技依赖。只是没预料到后来到2018年中兴事件愈演愈烈，成为整个科技界乃至国家的热点事件。文章发表后，不少读者反馈说文章观点太消极，没有正能量，打压大家的热情和积极性。事实上笔者从事软件研发15年有余，深刻理解和体会工业软件研发的艰巨性，文章只是将自己的客观感受表达出来。从技术角度看，中国开发软件完全没问题，只是受制于体制，人才，市场等约束，才导致国内工业软件被国外公司一统天下的局面。

到2019年，出现了美国制裁华为的事件，“弯道超车”更是被频频提及。在仿真软件方面，如果美国动真格将所有美国开发的软件禁用，实际上只是便宜了欧洲和俄罗斯的公司。欧洲和俄罗斯在工业软件方面并不弱，只是没有美国公司销售做的厉害，况且仿真软件也不在“禁运”之列，对华为真正的影响也就是阵痛，和接下来加强和欧洲俄罗斯公司合作以及加强自身软件研发。

-----正文-----

在当今中国的各个领域，“弯道超车”是个经常被提及的口号，然而常识和交规告诉我们，弯道不仅不能超车还得减速，超车极其危险。仔细解读“弯道超车”，其实还暗含了三个前提条件：1.两车的差距不能太远；2.车本身动车要动力十足；3.弯道的路况要好。但是很多行业都不具备这些条件。多年以后我们回头看现在的“弯道超车”和我们现在看大跃进时的“亩产万斤”如出一辙。

工业仿真软件，这里主要指计算机辅助工程CAE（Computer Aided Engineering）软件，包括通常意义上的CAD，CAE，CFD，EDA，TCAD等。对CAE行业稍微了解的人就知道，目前中国CAE软件市场完全被外资产品占据，大到Ansys，海克斯康（2017年收购MSC，以前笔者分析过收购原因），Altair，西门子，达索，Cadence，小到Comsol，Autodesk，ESI，

Synosys, Midas, Livemore等（这里的大小以市场份额来算）。关于国内为什么没有CAE软件的原因很多人做过分析，有些分析相当中肯，但是很多没有点到痛处，同时也缺乏指导性。

工业仿真软件，在中国长期以来被人误读很深，因为产品设计缺乏自主创新，市场上也缺乏相应国内产品，工业仿真软件要么被认为是可有可无的附属品，要么被夸张为高精尖的核心技术产品。在上文中，提到了一个软件项目成功与否可以从软件技术，工程技术，市场价值和策略三个方面来考量

- 1.软件技术，开发软件所涉及的一系列技术，涉及到的技术有计算机图形学，几何造型，网格技术，用户交互，核心求解器，高性能计算，优化算法以及其他底层IT技术等等。
- 2.工程技术，工程技术是指在工程实践中积累的经验，比如飞机的设计参数，隐形材料的配方，某种先进的工艺流程。
- 3.市场，与商业相关的都可以理解为市场，比如投入多少，收入多少，营销，销售，市场份额等。

软件方面：

不像快速变化的APP，互联网软件，当前大部分工业仿真软件技术的核心算法和底层技术在上个世纪已经成熟，比如有限元方法四五十年前都提出；FDTD在上世纪六七十年代提出；流体中的FVM也差不多在上世纪七八十年代，多年来在湍流和大涡模型改进；电磁领域仍然是围绕求解Maxwell方程展开；大多数三维实体建模软件的内核ACIS/Parasolid 核心概念和实现也是在上世纪七八十年代，虽然版本一直在更新，但在理论算法上很难有大的突破。而对应的网格技术，线性方程组算法，并行计算，用户交互，性能问题，HPC基本也都是循序渐进的提升，少有颠覆式的创新。这里要提一下开源工具，笔者用过很多开源仿真工具，其本质上是某些理论的实现，并无太多创新，很多真的就是工作量的事情，工业仿真软件的重点并不是解决某一类技术问题，工业仿真软件考虑的是稳定性，准确性，可靠性，可维护性，性价比，用户体验等。大多数开源工具在这些方面达不到要求。

工程技术方面：

一贯的观点是：软件是工具，利用软件积累起来的工程技术经验才是核心技术。离开时了工程实践和试验，软件只是一个外壳。看另一个极端的例子，大家都知道中国的高铁动车技术发达，如果把高铁动车的设计仿真资料，流程，工艺融合在一款软件里，在动车设计领域必定是一款畅销的软件产品，但这样的软件无法走向市场，更不可能走向国外市场，因为这种核心技术不可能外泄。从这个意义软件上讲，凡是能用的软件都是工具，而不是所谓的“核心技术”，核心技术是需要长期积累，即使转让也要长期消化。

市场方面：

欧美发达国家有着成熟的知识产权保护体系，软件从开发，销售，维护，都有着严格的IP保护，软件产品也有系统的盈利模式，基本不存在盗版一说。一般工业仿真软件最初都是技术性驱动，到后来会变成业务性驱动，业务性驱动稳定之后会引入充分的市场竞争，整个市场会趋

于良性发展，简单讲就是各个厂商会在盈利的同时力图保持技术的先进，从而推动整个行业往前发展。以Mentor为例，每年仅支付几何图形软件开发工具的License费用高达5, 6百万美元(注意：不是软件，是软件开发工具)，这在国内是想都不敢想的。因为实在太贵了，这些开发工具在国内基本无人问津，国内要么用盗版，要么用开源，所以没有坚实的底层技术积累，上层应用也就无从谈起了。

以几款软件为例，后面的百分比是一个大概的影响因素比例：

1.操作系统window软件：

软件技术：技术垄断，软件本身就是其核心技术，体量大开发周期长（50%）

工程技术：无

市场：垄断市场，一旦有相关产品出现，立马降价或者免费，维护其垄断地位,收入有保证，围绕操作系统的附加产值高（50%）

2.三维通用CAD软件：

软件技术：研发周期长，底层技术稳定（30%）

工程技术：依托行业背景，出现大量的融入行业背景技术的的专业CAD（20%）

市场：市场体量大，厂商众多，竞争充分，投入周期长，投入大（50%）

3.PKPM：

软件技术：（10%）。

工程技术：行业背景强，融入设计规范（60%）

市场：市场体量大，壁垒高，容易形成垄断。（30%）

4.Nastran

软件技术：技术积累时间长（20%）

工程技术：NASA起家，拥有大量的实际工程经验（60%）

市场：在某些领域（比如航空航天）处于垄断地位。（20%）

5.STAR-CD

软件技术：技术积累时间长，有很多独特的算法（20%）

工程技术：拥有大量的计算案例模型（50%）

市场：CFD领域拥有重要的一席之地（30%）

6.HFSS：

软件技术：求解器，网格划分，建模技术构成其核心技术（30%）

工程技术：拥有大量的计算和试验数据对比，求解器以其稳定性和准确性成为行业事实上的标准（40%）

市场：市场龙头（40%）

7.COMSOL:

软件技术：利用数值计算方法求解各物理场PDE，大而全（50%）

工程技术：（20%）

市场：偏向于研究型领域市场（30%）

8.Ansys

软件技术：早期老牌CAE，技术积累时间长（20%）

工程技术：拥有大量实际案例和客户工程应用经验（40%）

市场：推广早，通过各种并购占据市场第一份额（30%）

9.LS-DYNA

软件技术：显式动力学鼻祖（20%）

工程技术：提出了大量模型应对实际工程（40%）

市场：成为碰撞，冲击动力学分析事实上的首选工具（40%）

10.Algor

软件技术：早期大型通用CAE软件（20%）

工程技术：（20%）

市场：被Autodesk收购，市场份额有限（60%）

从上面几款软件可以看出，工业仿真软件的重头还是工程技术和市场，软件的因素只占一小部分，这也决定了工业仿真软件与普通软件的根本区别：技术驱动和业务驱动并存，不太可能有颠覆式的创新，也不太可能通过新的商业模式产生质的改变；市场需求稳定，很难出现爆炸式的增长。

2019年，美国政府针对华为的制裁一浪高过一浪，不仅从硬件上终止供货，而且开始禁止美国CAE/EDA软件公司与华为的来往。如果说硬件制裁是明着当头一棒，那软件制裁则是致命的背后一刀：在国内CAE/EDA设计仿真软件的研发要远远落后于硬件，但是对于新产品研发，仿真软件是绝对必不可少的一步。脱离仿真，不仅极大的影响研发效率，而且还可能因为实验无法通过而导致产品研发失败，仿真是验证产品早期设计唯一可行的方法，利用仿真，可以替代绝大部分实验，节约成本和减少研发周期。

以手机仿真为例，需要进行信号完整性，电源完整性，电磁干扰仿真，结构仿真：电磁仿真的标杆软件HFSS，ADS，CST，其中HFSS为Ansys（美国）旗下产品，ADS为安捷伦（美国）

产品，CST为德国；热分析标杆软件有FLOTHERM（美国Mentor公司），Icepak（美国Ansys）；散热分析有Fluent（美国Ansys公司），Star-CD被西门子（德国）收购；非线性结构分析软件Abaqus公司在美国被达索收购；跌落碰撞分析标杆为LS-Dyna（美国LSTC公司）产品；另外一些专用的仿真处理工具：HyperWork（美国Altair），Enight（美国Ansys），ANSA（希腊）。

一款好的工业仿真软件，最大的价值不在于软件本身，而在于软件所涵盖的工业背景 and 知识，以及基于虚拟模型对真实世界准确的模拟仿真。而这种固有的特点决定了，软件的价值来源于长期的知识积累，不太可能依靠一种新兴的技术就完成质的飞越，市场上大部分工业仿真软件，诞生时间基本都在十年，二十年以前，而后来的发展起来的仿真工具比如COMSOL，RADIOSS，以及CAD软件中集成的仿真工具接受度很低，就很能说明问题。

早在制裁中兴之初，我就提到制裁华为是早晚的事情，但是华为体量大，产品线远比中兴丰富，战略眼光也远在中兴之上，制裁会对华为造成影响，但不会像中兴那样伤筋动骨。如果说美国制裁中兴还有那么一点站得住理由，那制裁华为则完全是出于战略上的考虑：压制中国科技公司及其在全球的影响。美国可以接受中国的发展，但绝不能接受中国在某些行业上在全球范围内主导话语权，可以预见：美国对于中国的压制只会越来越多，力度越来越大。国内的CAE/EDA公司众多，但实力普遍弱小，如果能抓住机遇，潜心打磨产品，未尝不是一个机会。

所以，工业仿真软件“弯道超车”基本就是无稽之谈。所以要发展工业仿真软件，唯一可行的道路就是：踏踏实实积累技术，让用户和市场检验完善产品，切实解决实际工程问题，一口一口啃下市场份额，尤其是国外成熟的软件市场，否则，发展CAE软件就是一句噱头，更谈不上“弯道超车”。

工业仿真软件可以和支付宝一样快吗？

原创 www.cae-sim.com 多物理场仿真技术



支付宝和工业仿真软件似乎没有什么交集，但在网上多次看到类似的观点和言论：工业软件有那么难吗，让阿里开发一个不就行了？工业软件性能太差，运行时间太长，不能让阿里优化一下？

用过支付宝的人都有体验，在交易时，基本在秒级以内完成。有人会问类似双十一高峰，为什么这些操作都能在毫秒级完成，而工业仿真软件即使面对千万量级的数据，使用服务器，也要运行数天之久，工业仿真软件能否像支付宝一样，在秒级内完成仿真计算呢？

支付宝的快和工业软件的慢，其根源在于两者处理的数据和业务逻辑大相径庭。支付宝的每笔数据都是独立的，哪怕千亿级别的数据，不同地区的交易采用各自独立的服务器，数据和数据之间没有关联，很容易采用分治的策略，可以根据数据的特点，比如区域，年龄段，时间，交易额，规模等作出相应的分类，通过队列，映射，缓存等软硬件方法，针对不同的属性数据做出相应的对策，加速数据的处理操作。

举个简单例子，要处理无数个小球，这些小球有其属性，包括位置分布，重量，颜色，大小都不相同，可以根据这些属性还有数量，规模等进行分类，分解成不同的任务处理，不同的任务可以分配合适的资源。而仿真软件的数据就像是小球都被挂在渔网的节点上一样，小球通过网构成了一个整体，触动一个小球，会对所有小球产生影响。所以工业仿真软件处理的模型不管逻辑上还是数据上都是一个整体。

为了加速仿真，仿真软件在算法上也采用了很多加速方法，除了底层基本算法的改进外，最主要的方法也是采用分治的思想，即把模型分解成小的部分，同时处理，比如几何视图操作上的八叉树，将数据放在树结构中，避免不必要的计算；通过模型降阶(Model Order Reduction)方法降低求解模型的耦合性；再比如快速多级，也是典型的分治策略求解大规模满秩线性方程组。但是这种分治只能在高纬度上进行，最终分治的数据规模还是很大，算法复杂度仍然最低保持在 $N \cdot \log(N)$ 。另外对于大部分模型求解，需要采用迭代算法，这也是性能消耗的大头。所以相当时间内，不管硬件软件怎么发展，工业仿真软件性能瓶颈始终存在。当然如果量子计算出现颠覆性的突破，另当别论。

回到开头的问题，虽然支付宝改变了大家的生活方式，但公司本质上属于软件+金融公司，靠服务流量赚钱，在工业软件方面没有积累，没有人才，也没有动力去做。工业软件的工业+软件，两者都必不可缺。所以希望支付宝来做工业软件，纯属一厢情愿。

工业仿真软件的性能也是软件开发中除了功能以外的重中之重，后续会更详细介绍相关内容：

工业仿真软件研发性能之道

中国没有世界级的工业软件是一种耻辱吗？

原创 邓子平 多物理场仿真技术



在2021年华为中国生态大会上，华为董事、CIO陶景文发表了“加速数字化转型 共同繁荣数字生态”的主题演讲，很多媒体以“中国没有世界级的工业软件是一种耻辱”做了报道。

华为CIO陶景文：中国没有世界级的工业软件，是一种耻辱！

多物理场仿真技术

其中原文如下：

“比如说在数字经济这个行业里头，我们的“芯”就是芯片，我们现在来看，华为设计出来芯片，但造不出来。就算自己有生产线可以造的时候，我们的材料，我们的基础的东又跟不上。那我们的“魂”在我们的ICT和数字经济行业是什么呢？我觉得至少有那么几个东西，数据库，操作系统，工业软件啊。我们有这么大的工业规模，中国没有世界级的工业软件，我认为可能是一种耻辱。”

首先媒体的报道脱离了上下文背景，篡改了原话，导致完全不同的含义；如果说中国没有世界级工业软件是一种耻辱，那中国没有的世界级的东西多得去了，是不是都是耻辱？

稍微了解工业软件发展历史的人和国内工业软件现状的人都知道，工业软件产品需要长期的资金投入，原始技术积累，大量的用户迭代和工程实践，良好的知识产权保护，并基于一定的生态链系统。长期以来，国内这些条件都不具备，这是客观事实，并没必要否认，压根也谈不上耻辱。

早期对于很多有开发打算的企业，长期投入研发和购买软件费用相比，当然买要划算的多；此外，闭门造车，不融入生态链系统，直接结果就是产品功能差，解决不了实际问题，没有用户陪玩，毫无竞争力；对于有意向的资本，投入周期长，利润低，进入门槛高

等等，亏损是大概率，这也使得工业软件在资本投资上一直是冷板凳。此外，工业软件一直体量不大，和动辄亿万流水交易和上亿用户的互联网相比，用户基本都能数得过来；最后工业软件早期基本都来自于军工，也是比较敏感的存在。

所以简单讲，中国之所以缺乏世界级的工业软件，是受限于经济条件，科技发展水平以及需求的客观结果。

中国国内的工业软件基本被国外产品垄断，如果不出意外，这种趋势还会一直延续下去。但是从2017年开始，特朗普政府对中国科技企业一步步加紧的制裁政策，让国内清醒的认识到，不管是学校科研用的MATLAB软件，还是芯片设计用的EDA软件，都有可能某一天在国内被禁用。也正是在2021年5月的政府工作报告中，首次把工业软件纳入了急需解决的卡脖子技术，这也说明国家已经把工业软件提高到了战略的高度上，这是一件非常有意义的里程碑事件，但是同时，工业软件固有的特点也决定了其研发的长期性和艰巨性，要有理性的看待。

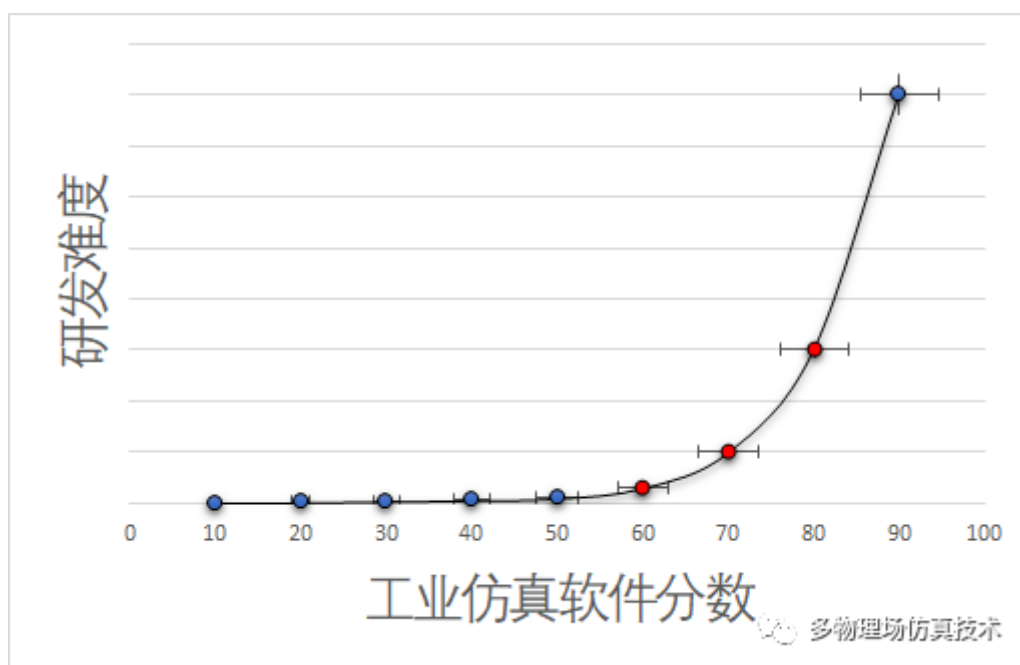
从目前国家政策和趋势来看，未来一二十年，中国的科技一定会迅猛发展，其中也包括工业软件，但同时也要应对外来的恶意制裁和竞争。古人说“好学近乎知，力行近乎仁，知耻近乎勇”，未来的一二十年，可能真的需要拿出“知耻而后勇”的精神，积极面对科技领域的挑战和竞争，而对于工业软件的研发，更要做好长期吃苦的准备。

给工业仿真软件研发打分

原创 邓子平 多物理场仿真技术



和大多数考试一样，拿个及格线可能问题并不大，但是要再往上走困难程度就是成指数级上升了。



60分标准

这一标准主要是能实现基本的仿真功能，走通流程，处理小规模的用例

- 1 基本三维实体建模
- 2 支持面网格，体网格划分
- 3 能求解基本的稳态，瞬态问题
- 4 支持行业文件的导入识别
- 5 支持step, igs等常用标准文件格式
- 6 完整的给出计算流程，并能与NAFEM用例做基准测试
- 7 有限的，基本的几何前处理功能

多物理场仿真技术

- 8 提供完整的GUI，视图显示，操作等功能
- 9 提供基本的后处理功能
- 10 计算行业的标杆经典问题
- 11 在可接受时间范围内给出计算结果

70分标准

这一标准强调功能完善，各种实际实用功能，对准确性，稳定性，可靠性都提出了很高的要求，是真正意义上工业仿真软件的及格线要求。

- 1 支持导入导出*.sat,*.x_t
- 2 台式机百万网格100秒内划分
- 3 支持虚拟拓扑
- 4 支持三角形，四边形，四面体，六面体网格划分
- 5 支持网格布点以及硬点
- 6 支持几何切分
- 7 支持Cell几何结果和网格划分
- 8 基本的几何清理修复
- 9 自适应网格加密稀疏
- 10 支持几何实例引用
- 11 参数驱动网格划分
- 12 混合网格划分
- 13 自适应改进网格质量
- 14 分布式几何网格处理
- 15 支持分布式求解器，大规模并行计算
- 16 支持Windows，Linux
- 17 底层求解器使用Krylov，多重网格，快速多级等加速算法
- 18 单机求解效率接近或使用MKL
- 19 完善的后处理流程和报告
- 20 提供优化设计功能
- 21 求解结果和行业标杆软件或者实验结果接近
- 22 完备的License机制
- 23 提供脚本，二次开发相关功能
- 24 支持主流第三方行业软件接口
- 25 最大程度的支持多线程，多进程，分布式，乃至大型机超算
- 26 支持GPU计算
- 27 提供云端服务，云计算

- 28 多国语言，完善的使用文档和测试报告
- 29 部分支持AI
- 30 积累相当的基准测试和用户测试案例

80分标准

这一标准强调处理实际工程问题的能力，尤其是在工作效率，以及保证准确性前提下的处理大规模用例的性能问题，并为能逐渐成为行业内的标杆软件打下基础。

- 1 稳定准确的处理中等规模的用例
- 2 有能力处理行业内大规模的用例
- 3 有能力对新的设计进行验证和仿真
- 4 较好的平衡仿真准确性，稳定性，性能
- 5 逐渐成为行业内的可靠设计仿真工具
- 6 对于工业实际问题能提供有效解决方案
- 7 可以成为行业内标杆软件替代品
- 8 围绕核心软件提供一系列辅助工具
- 9 支持各种前沿技术
- 10 作为可靠模块，可以成为产品链上的一环

需要说明的是，国内的研发基本停留在60分标准及以下，60到70是个门槛，各项标准可以对号入座。

工业仿真软件的基础技术

原创 邓子平 多物理场仿真技术



按照模块和功能，基础技术可以描述为：

一个核心，两个框架，三大支柱，四层应用

一个核心：求解器

两个框架：软件开发框架，软件架构框架

三大支柱：图形，几何，网格

四层应用：业务算法，优化算法，UI，测试

以上十点每一点都可以展开成一个大项，甚至可以用一门或者多门学科描述，在之前的文章中多有描述和讲解。

纵观当前市场上所有的工业仿真产品，没有一个是单独凭借一己之力完成的，或多或少都在产品生态链上。就像波音飞机，核心部件自己生产，其它大部分的零件在全球采购。

这也是笔者一贯的观点：工业仿真软件的研发是一项系统工程，要达到世界级的水平，在当前环境下，不管是从技术角度和商业角度看，单独完成成本都较高；开发世界级的工业软件要避免闭门造车和埋头走路，而应该融入全球行业生态链，在世界范围内寻求合作。

一篇文章入门多物理场有限元（全篇）

一篇文章入门网格划分

一篇文章入门三维几何内核

从技术角度聊聊“系统工程”

少一点“国产自主”，多一点“合作共赢”：写给国内工业软件发展

工业软件研发有多“难”

原创 邓子平 多物理场仿真技术

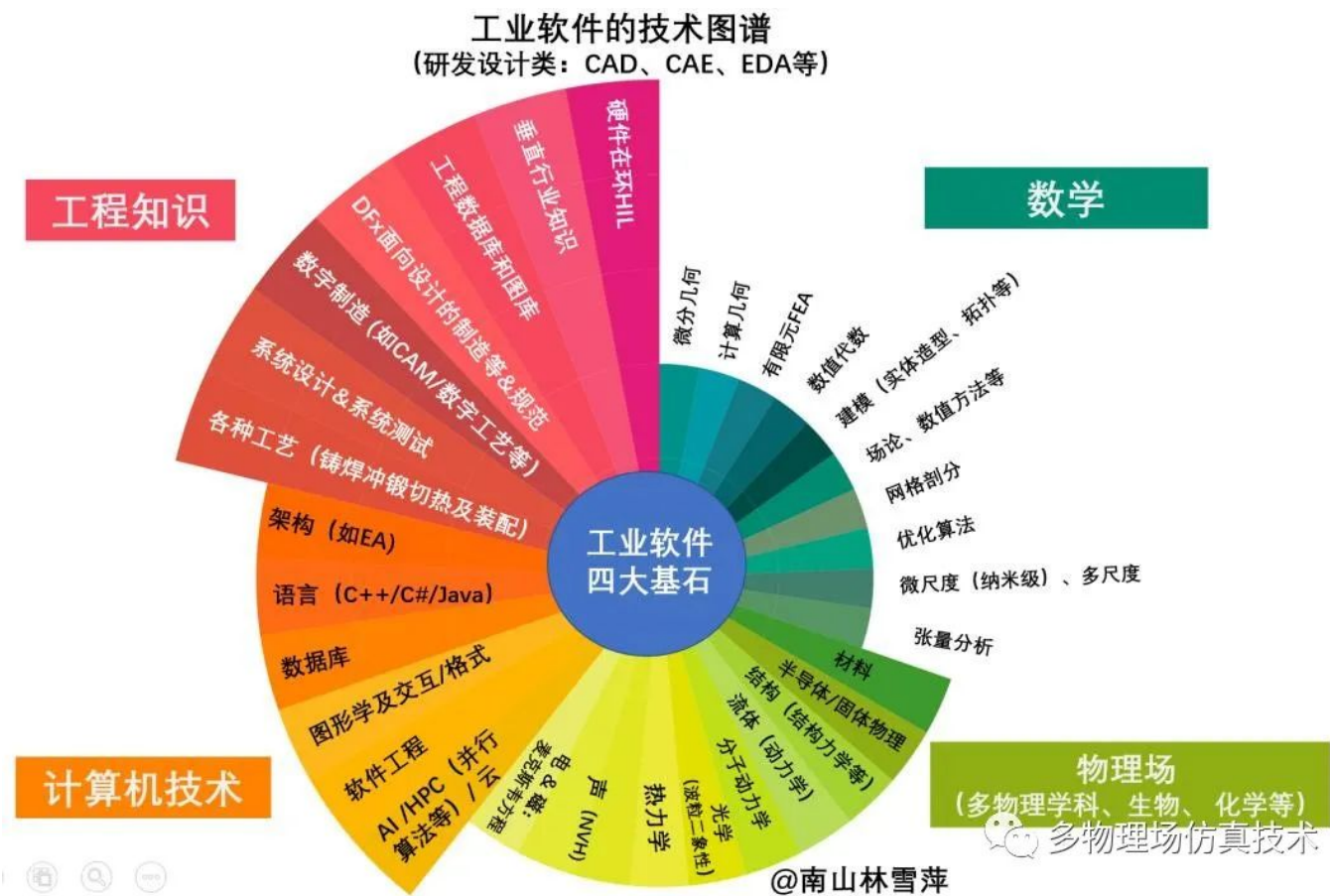
收录于话题
#工业软件杂文系列 17 #求解器开发 22 #CAE 2 #仿真研发工具 21 #产品研发 1



工业软件研发的“难”是共识，但是到底有多难，很少有人讲清楚。

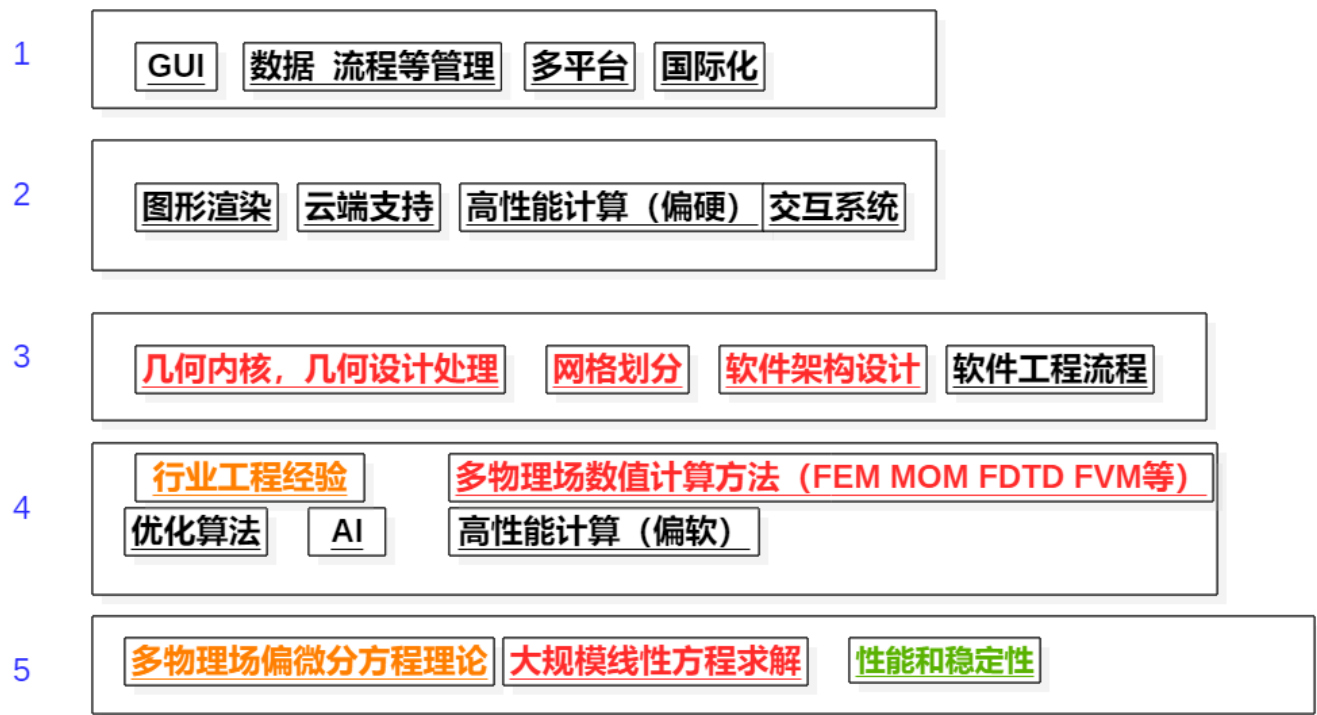
在笔者看来，工业软件研发真正的“难”是：人家研发投入打磨三四十年，而你只想三四年就“赶上”！

之前在网上看到如下图：



图片来自于网络，如有侵权请与作者联系

从科普的角度看，确实能看到不少内容；但是从产品研发的角度讲，没那么多事，落实到一线软件开发还得看这张图：



多物理场仿真技术

本公众号中的内容基本也是围绕这张图展开的，最近的一次是关于网格划分（[点击查看](#)）。后面也将继续围绕这些内容做更深层次的讨论。

内容从上到下分为1-5层，其中1和2属于上层应用，偏向于IT；第三项是工业软件几大上层应用支柱；第四项是工业软件的核心价值所在；第五项属于偏底层基础支柱。

- 1. 分层并不按照某单一原则，而是参考属性特点，难易程度以及软件架构，是一个综合性考量的层次分级。
- 2. 每一项内容并不完全独立，之间有相关性
- 3. 每一项都是产品研发绕不过的内容，早期最好确定研发方向和技术选型
- 4. 在有限资源内，借助该图可以帮助定义产品功能
- 5. 产品研发也可以从这张图出发，在研发迭代中帮忙找出适合团队的研究流程方法和技术路线。

图中每一项都可以作为独立的内容或者学科，但对于研发来讲，主要是出产品而非研究学科，从这个意义上来讲，工业软件开发都基于成熟的技术和工业实践迭代，而非一般意义上的“技术瓶颈”。工业软件的研发并没有想象的那么难，短时间出个原型和入门版基本没什么问题。

工业软件开发有其自身规律，只要有固定投入，产品技术研发完全没有问题，至于产品质量则要依靠长期的打磨，不太可能一蹴而就；真正需要解决的是长远研发的收支平衡，实际工程问题，用户迭代以及底层基础模块的替代。

最近两年，国内工业软件开发风起云涌，有新“李逵”，也有老“李鬼”，写文章的目的也是给国内工业软件开发鼓鼓气，希望更多的人来关注工业软件开发，推动国内工业软件的发展！

如何构建世界级的中国工业仿真软件

通俗有限元方法理论介绍

原创 www.cae-sim.com 多物理场仿真技术

收录于话题

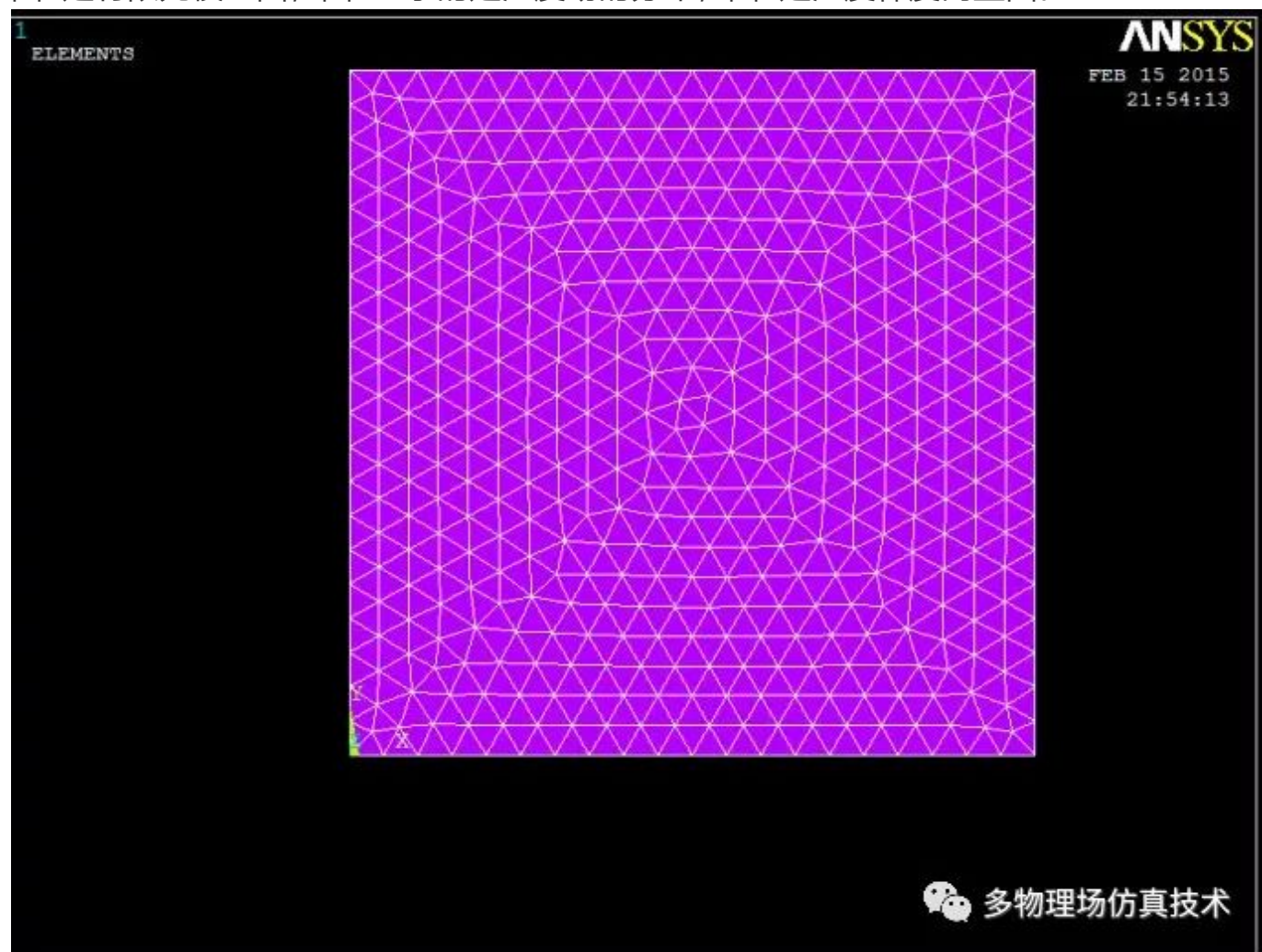
#仿真研发工具

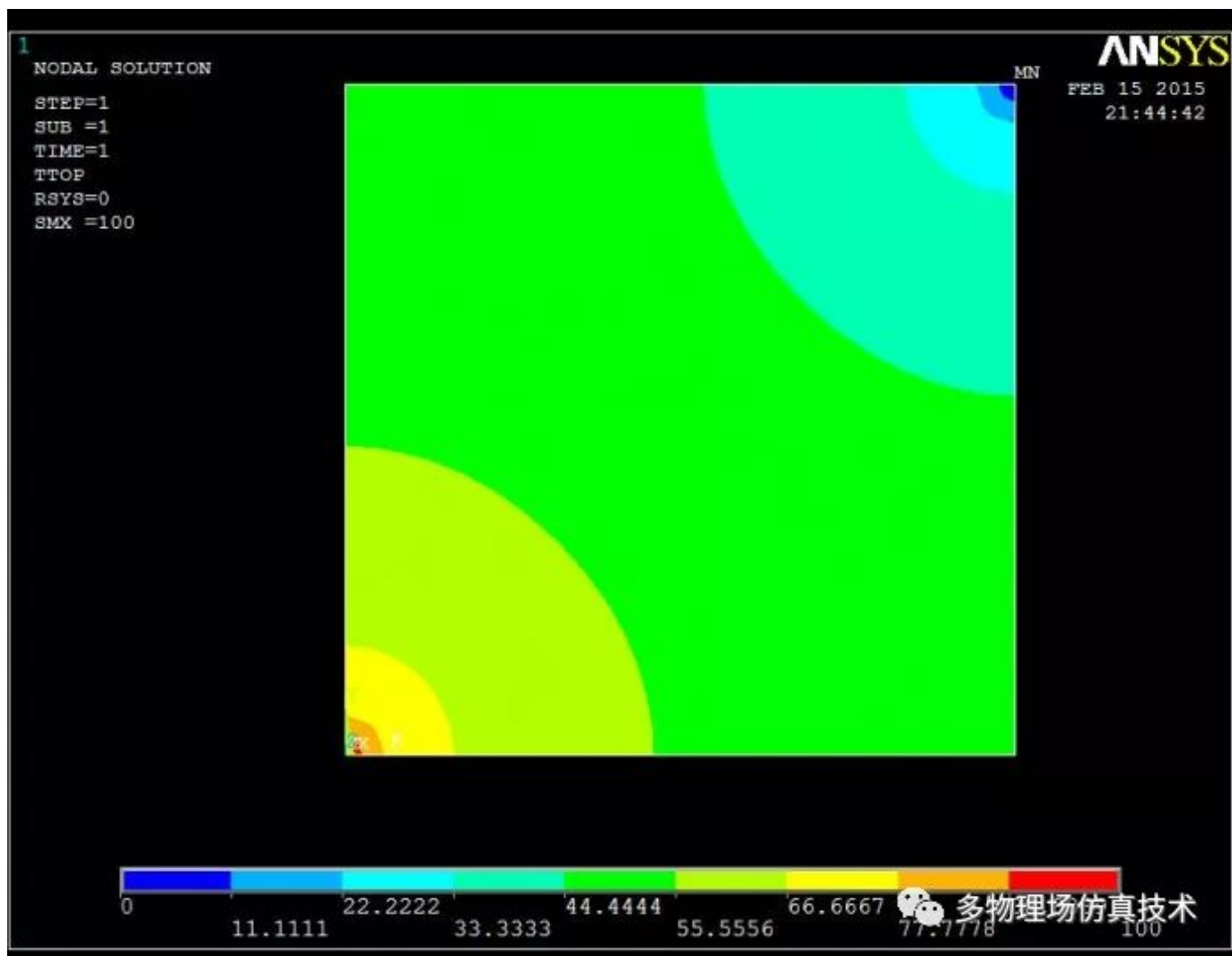
21个

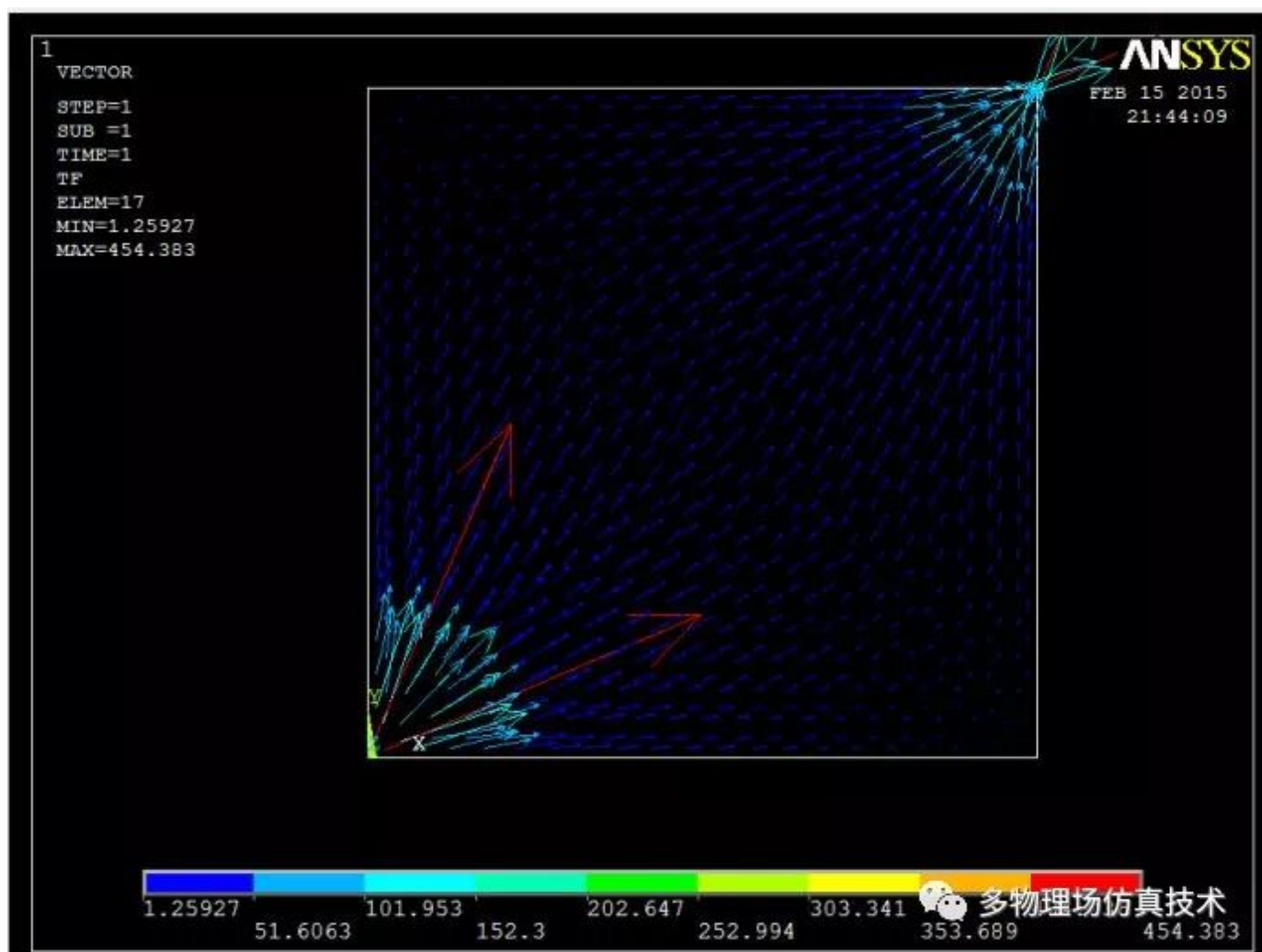
先来看一个平面热传导的例子：

该平板左下方的顶点温度设置100度，右上方顶点设置0度，其它边自由。

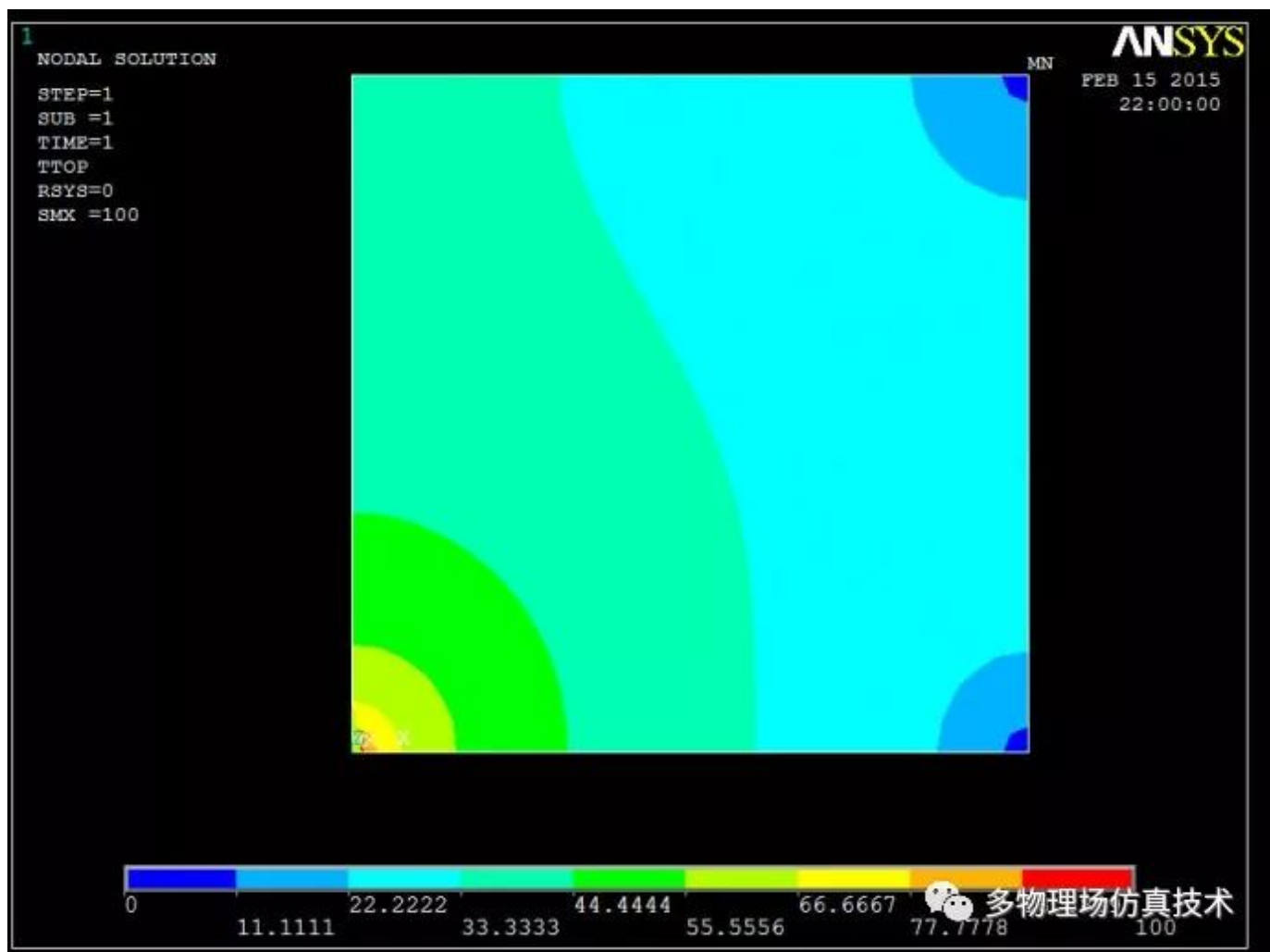
图1是有限元模型图，图2显示的是温度场的分布，图3是温度梯度向量图。

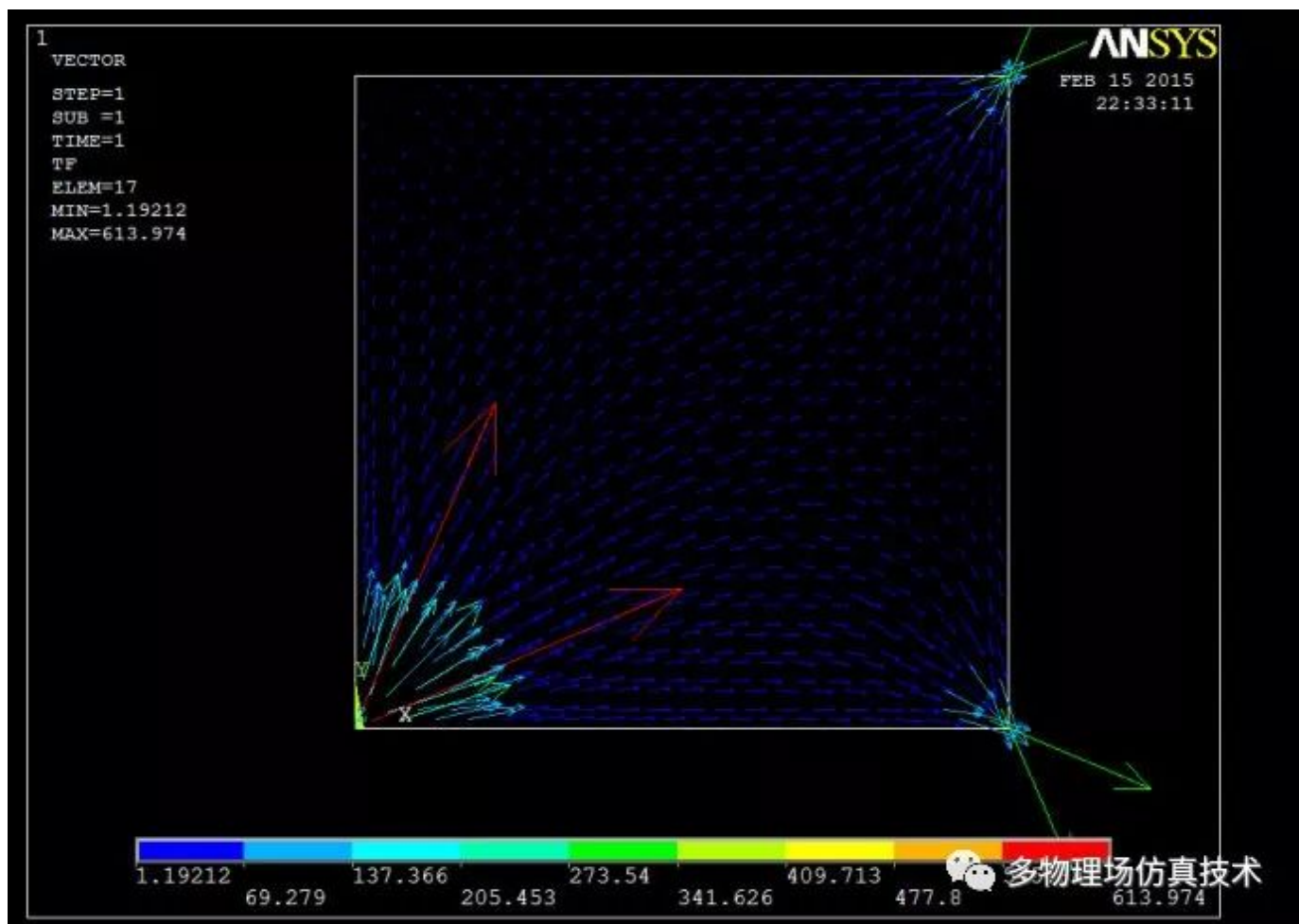






在上面的例子中，右下方的顶点也加上一个温度为0的约束。





问题来了：

1. 为什么温度场会服从这种分布，而不是别的形式？
2. 温度场的分布是否是唯一的？如果是，为什么？
3. 对于这类问题如何求解？

偏微分方程（Partial Differential Equation），简称PDE

偏微分方程最早是科学家从振动，热传递等一系列的物理现象归纳总结出来的，实际上绝大部分物理现象都可以用PDE来描述。最常用的PDE都是二阶的，即场的导数是二阶，高阶的PDE也可以通过变换降成二阶PDE。

首先介绍导数来预热，我们知道把位移对时间求导，就是速度，把速度对时间再求导就是加速度，简写为 $a=s(t)''$;

a 为加速度， S 为位移， t 为时间；

这个表达式实际上有两层含义：

1. 位移是时间的函数。例 $S(t) = t*t*t-5*t+4$, $S(t) = \sin(t)/\cos(2*t)$;
2. 要使加速度不为0， $S(t)$ 一次求导以后必须不为常数，因为常数求导为0。

PDE中要求解的是场函数的表达式，即上面中的 $S(t)$ ，知道了 $S(t)$ 的表达式，给出 t ，就能求出任意时刻的 S 。

下面是一个典型的二阶PDE，其中 u 为场函数

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

方程看起来很简单，其实不管是通解，还是定解，u的解析解表达式都是很复杂的，通常都是高阶函数，正弦，余弦，指数，对数的组合表达式。理论上PDE都可以找到解析解，但现实大部分PDE的解析解都无法求出。找解析解是数学家们的任务。

该PDE说明u是含有x, y的函数，当对x求导时，y看做常量，对y求导时，x看做常量。

假设 $u(x,y)=x*x-y*y+x$ ---(1)

u对x求导一次: $u'/x=2*x+1$;

u对x求导2次: $u''/x=2$;

同理对y求导2次: $u''/y=-2$

说明式1是方程的一个解，但问题是没有给定边界条件，方程的解实际上是有无限多个的，要确定定解，必须要设定边界条件，这也是例子中右下方加上一个新约束，导致温度场发生变化：即使边界一个小的变化也可能导致完全不同的定解，即不同的u表达式。

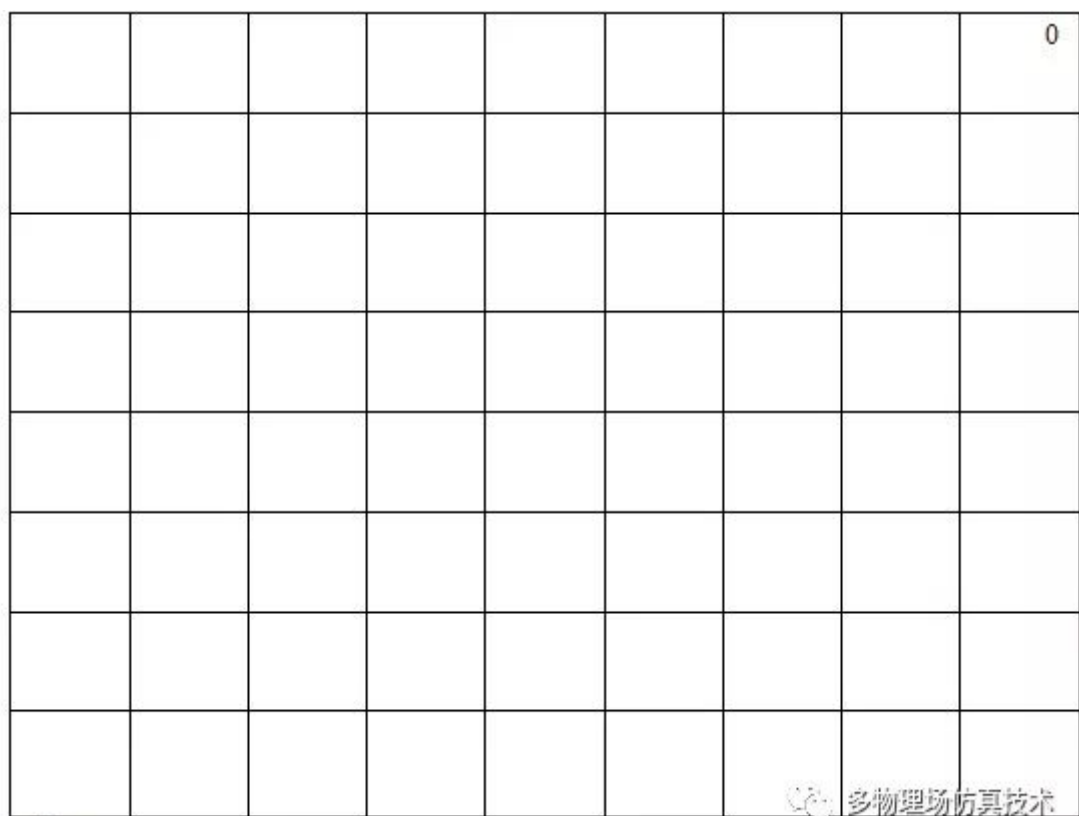
关于PDE 的边界条件和常用PDE参考：三角单元介绍

上述方程是无热源稳态热传导的PDE，场函数为温度，称为拉普拉斯方程（Laplace，名人得记住），拉普拉斯方程式是电磁，热，流体，天文等领域经常会碰到的一类数学问题。详细介绍可参考PDE方面的书。

从上面两个例子也可以看出，即使很简单的PDE也是很难找到精确的解析解。找到解析解固然最好，但是找不到也没关系。就跟圆周率一样，我们没必要知道它精确多少位，取7位数对于绝大部分应用足矣。找不到解析解，就只能依靠数值解了，数学家们首先用到的是 **差分法**。

任然以平面正方形温度场为例，假设左边，上，下边温度为100度，右边为0度。

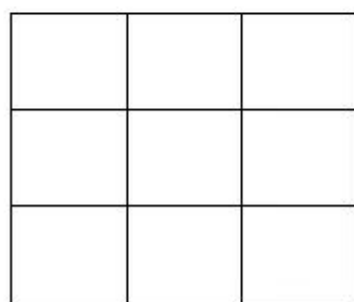
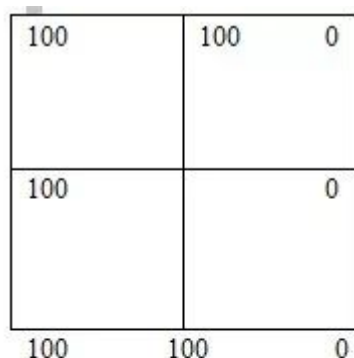
差分法就是在求解区域画上 一定密度的结构化网格，如下图：



100

上图中的任意一个点的温度值都可以用其周围临近的四个点的平均值表示，如果是长方形，表达式会略有不同。将网格简化如下：

中心点的温度为 $(100+100+100+0) / 4 = 75$



若将网格加密一些，如上图，中心四个点为未知量，每个点用周围四个点的和求平均，可得四个方程，联立解线性方程组即可，这是差分法的基本思想。

在差分法的基础上，发展起来的是有限元法。

有限元方法的理论基础是变分原理和加权余量法。通常有限元方法，变分法，加权余量，以及编程实现之间的关系很容易让人迷惑。

关于有限元方法的原理，建议直接记住一点：就是伽辽金法(Galerkin)。虽然还有变分，Razi等，但用Galerkin来理解有限元足矣。Galerkin方法是加权余量法的一种。加权余量法的基本原理是：

假设现在有PDE的近似解，而且这个解也满足边界条件，那么将这个解带入PDE，计算结果将产生误差（因为解场函数是近似的），为了在整个求解区域上保持与精确解相同的特性，就必须想办法将这些误差在近似解内部消除掉。

常用加权余量法有：子域法，最小二乘法，配点法，伽辽金法。使用主要涉及到两个方面：

1. 如何设计近似解。

理论上可以设计出无限多个近似解，指数，幂数，正弦，余弦等。但是教科书上无一例外的都是以多项式和为主。原因只有一个：简单。

2. 如何消除误差。即采用什么方法，强制设置误差为0，以满足方程的解。

伽辽金法要求误差相对于权函数是正交的，即两者的乘积积分为0。通常权函数为假设的解多项式。将求解区域离散后，不同的单元通过节点连接，节点所连接的每一个单元在该点产生的误差与设计的权函数乘积积分之后相加为0（有点拗口）。简单理解就是强制一个节点连接的所有单元在该点产生的误差和为0。由此可见，一个节点就可以产生一个方程（严格来说是一个自由度产生一个方程，因为温度是标量，一个节点只有一个自由度，如果取位移，平面一个节点就是2个自由度，X和Y两个方向），最终的解就是节点个数的多项式。

总之任何形式的PDE，离散成网格之后，都可以通过伽辽金方法得到一组近似的线性方程组 $K \cdot x = F$ 。而且利用伽辽金方法最大的好处就是K是对称矩阵，方便计算求解。这也是采用伽辽金方法建立有限元方程的主要原因。

说到这里，利用有限元求解PDE原理很简单：设计近似解，产线性方程组，求解方程组。产生线性方程组的方法有很多，最常用的就是伽辽金法。当微分方程存在相应的泛函时，伽辽金法和泛函可以推导出相同的结果。

如果说差分法和有限元法有什么区别的话，从网格角度看：差分法是把网格画好然后铺在求解对象上，而有限元法是直接把对象划分网格。因此差分法适合求解区域比较规则的几何，而有限元可以对任意形状几何求解。

除伽辽金法外，流体中常用有限体积法，有限体积法采用加权余量法中的子域法。

现在实际上已经回答了前面提出的三个问题：

1. 为什么温度会服从这种分布，而不是别的形式？ ---服从PDE
2. 温度场的分布是否是唯一的？如果是，为什么？ ---唯一的边界条件对应唯一的分布
3. 对于这类问题如何求解？ ---差分法，有限元法，有限体积法

更多内容可以访问我们的技术博客：

<http://blog.sina.com.cn/multiphysics>

阅读原文

仿真软件网格功能开发

原创 www.cae-sim.com 多物理场仿真技术

网格是多物理场仿真分析中很重要的一个因素，从技术方面考虑：

1. 网格的数量影响到求解的精度和效率
2. 网格的质量影响到求解的精度（网格单元是否畸形，网格密度是否合理）
3. 网格的阶次影响到计算精度
4. 不同分析类型对网格类型要求不同（相同的几何，流体，热，结构所需要的网格不同）
5. 复杂几何网格错误难以检查。

从以下几个方面介绍前处理器中的网格开发

1. 网格生成算法
2. 网格类型与质量检查
3. 网格加密/自适应网格划分
4. 网格显示
5. 网格开源工具
6. 网格商业工具

1. 网格生成

结构化网格：

结构化网格具有统一的拓扑结构，区域可以划分为规则的单元，节点之间有规律的索引。结构化单元只适合于求解模型简单，几何规则的情况。结构化网格算法也比较简单很容易实现。例如对一个六面体实体划分网格，在其中一个面上划分四边形，然后沿垂直该面方向上扫略即可生成规则的六面体网格。

非结构化网格：

大部分工程案例几何都不规则，网格需要使用非结构化网格。

常用的算法有：Delaunay 三角形算法，波前法 (Advancing front method)，映射 (Mapping) 方法，分治法，四叉树/八叉树。具体可参考《Mesh Generation》一书

网格工具与几何接口：

网格划分的对象是几何，因此需要定义好几何和网格工具的接口，即几何数据以何种方式传给网格工具。通常网格划分需要点线和边的数据，对于Nonmanifold几何，还需要提供公共部分的属性，比如相邻的面需要设置成double-side以方便网格工具识别。

网格参数：

网格参数的确定与调整，在开发过程中，网格的默认参数确定是一项比较繁琐的工作，没有哪个网格算法或者工具能够保证完全生成高质量的网格，尤其对于通用CAE软件，开发专业的CAE软件可能还好一点。需要找到高效的调试手段，保证网格生成的稳定和质量

2. 网格类型与网格质量检查

按形状分：

常用面网格：三角形/四边形；

常用体网格：四面体/六面体/金字塔/；

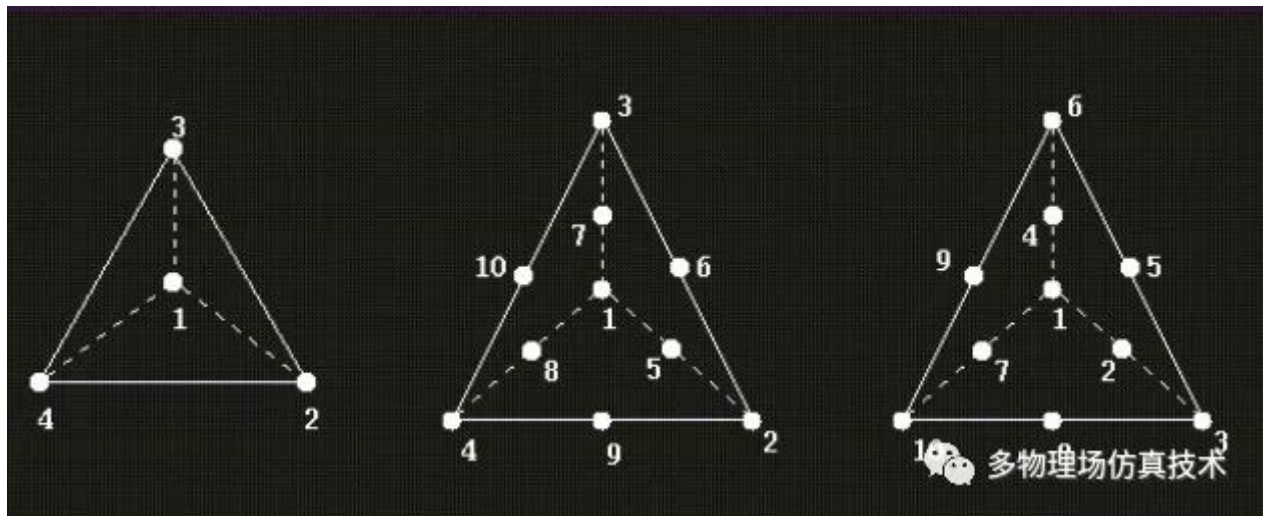
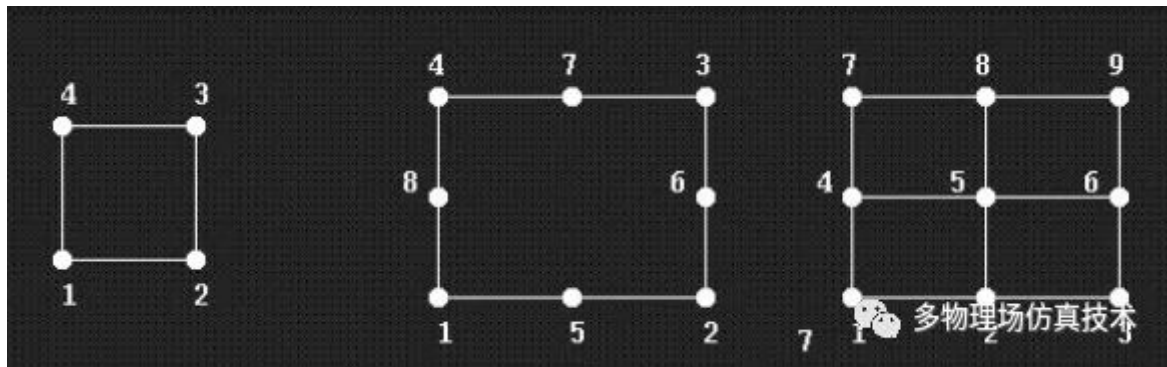
按阶次分：

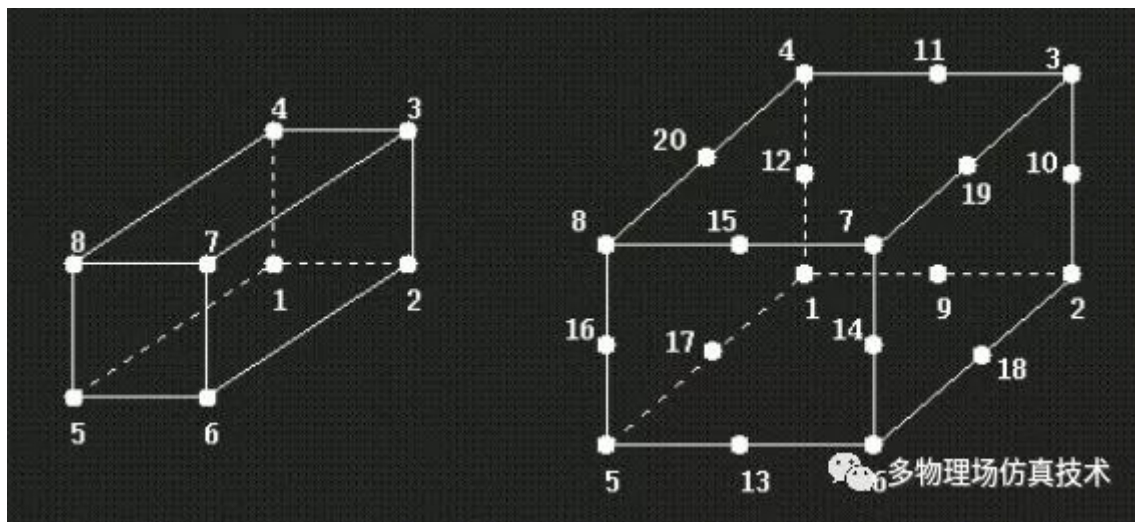
线性：每条边只有两个点；

二阶：每条边上中点有一个点；

多阶：每条边上中点有多个点；

几种常见类型网格：





质量检查：

网格质量直接影响到求解器求解的精度。差的网格甚至有可能导致求解器计算失败。

下面介绍一下常用的网格质量检查指标：

Aspect Ration(纵横比)：最长边与最短边的比值

Jacobian：雅克比值反映了单元偏离理想形状的程度，取值范围从0-1，取值越高网格质量越好。

Skew (扭曲度)： $=90 - \min(A1, B1)$ 其中A1 B1为单元中心线与底边夹角 $A1 + B1 = 180$

Warpage (翘曲角)：

Tetra Collapse (四面体坍塌比) $= \min(H / \sqrt{\text{Area}}) / 1.24$;

H为定点到对面的高度，Area为定点对面的面积，0为完全坍塌，1为正四面体。

由于一般网格数据量很大，通常调试网格质量做法是:生成网格后,转化为常用CAE软件网格文件格式，比如*.bdf，*.inp等，然后导入到HyperMesh中，检查网格质量。

3. 网格加密/自适应网格划分：

参考附录1-5：

FEA精度之网格加密(4)--ICEM网格质量分析；

FEA精度之网格加密(3)--HFSS网格质量分析；

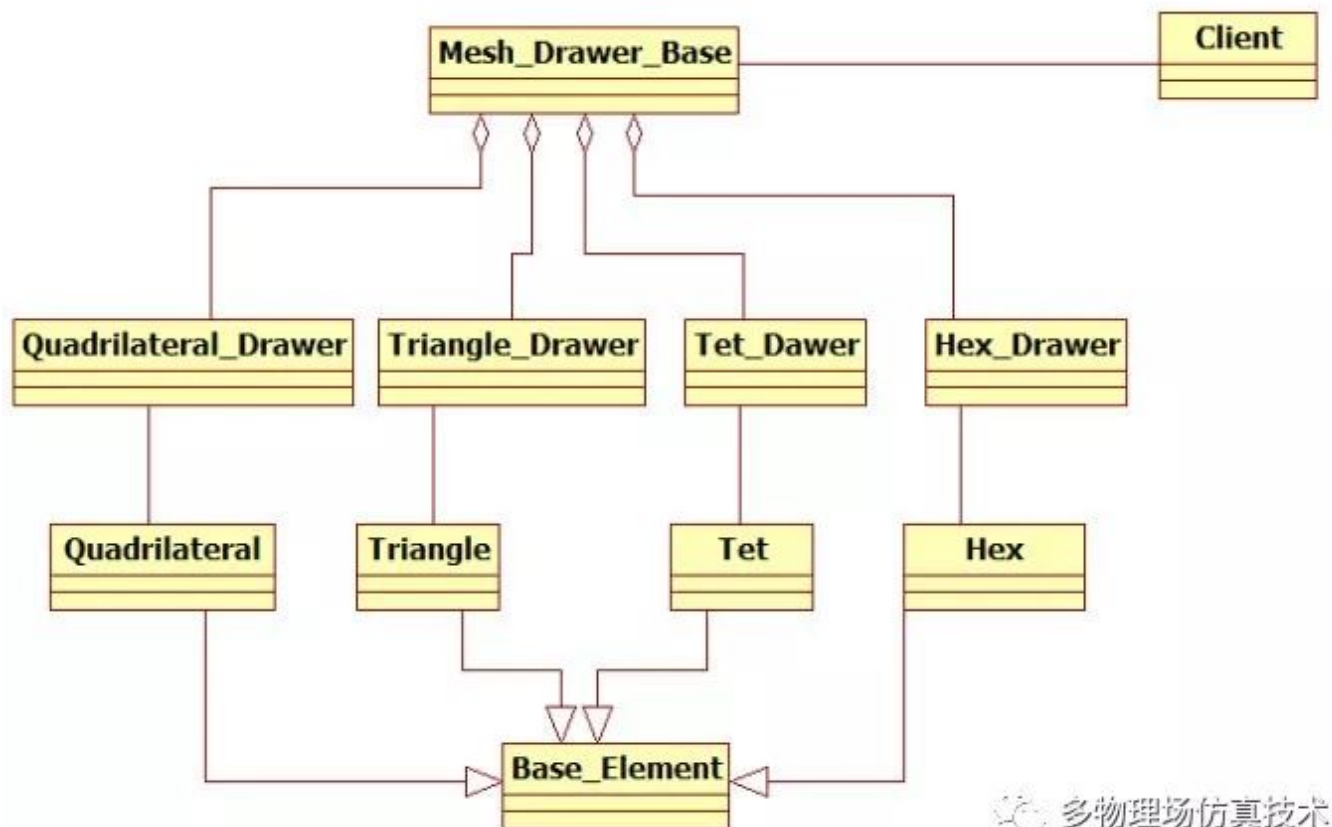
FEA精度之局部网格加密(3)；

FEA精度之局部网格加密 (2) ；

FEA精度之网格加密 (1) ；

4. 网格显示：

在有网格数据后，需要显示网格。网格的显示很容易实现，主要是结构的物理设计要注意一下。网格的数据结构可以使用公共组件的定义。定义如下结构：



多物理场仿真技术

网格数据结构仍然和单元数据结构保持一致，针对不同的单元设计不同的显示网格类，调用方只需调用Mesh_Drawer_Base接口。这样即能与有限元网格数据兼容，也实现了网格数据与显示的解耦，方便了整个结构的模块化。

5. 开源网格工具：

这里讨论的是能用于实际工程的开源工具，做研究和学术的工具不在之列。

开源的划分网格的工具箱很多，信手拈来就可以有一大串，但大部分的开源网格工具不是很难用，就是网格质量差，或是复杂几何划分不出来。以下介绍几个难易和实用程度都不错的开源网格工具：

Gmsh

<http://www.geuz.org/gmsh/>

Gmsh是一个自动的三维有限元网格生成器，并带有CAD生成功能和后处理器。支持三角形，四边形，四面体，六面体，三菱柱，金字塔形等。Gmsh生成的网格质量比起商业网格引擎毫不逊色。自带脚本生成几何的功能也为研发提供了不少便利，省去了利用商业CAD软件建模的开销。Gmsh提供可直接调用的Binary，也提供了源码。核心算法用C++实现。

TetGen

<http://www.wias-berlin.de/software/tetgen/>

TetGen是一个在德国的中国人写的生成四面体网格生成工具。考虑到目前大部分网格都是三角形和四面体，TetGen的从网格生成效率和质量，都相当不错。

NetGen

<http://www.netgen-toolbox.org/>

CGAL

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

CGAL全称是Computational Geometry Algorithms Library。

网格划分只是其中一部分功能。作为一个通用的算法几何库，在网格生成功能方面也有相当的参考和实用价值。

以上几种工具不仅自己用的比较顺手，在国外也被很多科研单位作为研发工具。

6. 商业网格工具：

这里介绍的商业网格工具是指用于研发的网格引擎，而不是划分网格的产品。以下几款都是本人用过的商业网格工具：

[Distene](#)

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

[PTC, Ansys最新划分网格的工具.](#)

特点：API接口调用简单，划分质量和效率都相当高。

缺点：不便宜

[VKI](#)

[www.vki.com\](http://www.vki.com/)

很多商业CAE软件划分网格工具，包括Ansys,Nastran.

特点：接口齐全，与商业CAD和商业CAE软件网格接口齐全。

缺点：不能生成六面体。

[Simmetrix](#)

[www.simmetrix.com\](http://www.simmetrix.com/)

很多中小CAE软件划分网格工具.

特点：模块齐全，几何，网格，仿真都有涉及。

MeshChecker是独立开发的一款网格查看和质量检查工具，运行于windows平台：

1. 支持导入Abaqus inp, Nastran bdf, Ansys cdb, stl, unv等格式文件；
2. 支持大部分网格质量检查标准；
3. 支持三角形，四边形，四面体，六面体，楔形，金字塔形等单元；
4. 各种查询标准功能，高亮各种查询单元，高性能的图形显示；
5. 大模型高效支持（100万节点，i5双核，显卡GT-730 PC机上流利显示和操作）；
6. 支持多视图查看网格单元，并支持单个单元的拾取查看，列表方式显示所有单元网格质量数据；

下载地址：

www.cae-sim.com

[阅读原文](#)

CAE软件研发的一些思考(1)

www.cae-sim.com 多物理场仿真技术

区别于其他行业的软件开发，CAE软件的研发有其特殊性，这里的CAE软件主要指使用有限元/有限差分/有限体积等数值计算方法进行仿真的软件，其中包含了前处理器，后处理器，求解器以及相关模块，求解领域包括但不限于静力，动力，流体，电磁，声，热，光等

中国在数值计算方法很早就有比较突出的成果，以冯康为代表的老一代数学家在有限元理论方面的研究处于世界领先地位，在工程应用方面也取得了相当的成功。只可惜由于种种客观原因的限制（主要是计算机在中国发展的限制以及投入不足），中国的CAE软件没能取得长足的发展。也导致了目前中国CAE软件市场被国外软件一统天下的局面。

长期从事CAE软件的管理研发工作，本文想从几个方面探讨一下CAE软件研发的一些相关话题。

1. 中国有无研发大型通用CAE软件的必要性。

2. 中国自主研发CAE软件的出路是什么

3. CAE软件架构设计

4. CAE软件研发管理

5. CAE软件的核心技术和风险

1. 中国有无研发大型通用CAE软件的必要性。

这是个老生常谈的话题，每逢开各种学术会议，最后总会得出结论：发展国产通用CAE软件是当务之急云云，开完会后不了了之。殊不知通用CAE软件研发涉及到的范围相当广，需要从技术，资金投入，开发周期，人才储备，市场应用，项目支撑等各个方面做好调研。开发一款优秀的通用CAE软件绝不是买个求解器，做一个前处理器，算两个case就能成气候的。Ansys, Nastran, Abaqus 发展了几十年，在研发方面投入巨资，不仅自己做开发，还收购了相当多优秀的CAE公司，将产品应用到无数的实际工程中，一步一个脚印才逐步发展到现在的水平的。

个人以为开发国产通用CAE软件虽然是个美好的愿望，在中国目前的环境下，不切实际（OS研发都不愿意投入，CAE更不用讲了）。退一万步讲，即使能开发出来一个像样的产品出来，结局也能想到：基本上是被国外的产品直接拍死，只能做做科研或者给高校使用。

2. 中国自主研发CAE软件的出路是什么

既然没必要开发大型通用CAE软件，那国产CAE软件研发的出路在哪里呢？

既然通用的道路走不通，那就走专业化的道路。使用CAE软件的目的不外乎两点：优化和验证设计，缩短产品上市周期。在这个过程中，用户最耗时的部分是 几何模型和有限元模型的建立以及模型的优化。如果能针对这个部分做一些专业定制的开发，将会给客户带来很大的价值。试想，对一艘船进行仿真，建立整船的几何到建立有限元模型需要1个月。分析的工况有20个，这些要全部手工建立，仿真结束后要自己生成报告。如果能开

发出一款专业的船舶建模软件，用户通过输入参数就能很快建立几何模型，而且可以从模板库直接取船舶的部件，考虑到船舶分析的一些特殊情况，比如大量使用shell和梁，杆单元，能够帮助用户快速建立好有限元模型，同时把常用的工况也做成模板，求解可以调用公认标准求解器，也可以调用针对船舶开发的专业求解器，仿真结束后把结果输出为用户想要的报告格式。可能3天就能完成整个仿真流程。这样可以大大减少用户的工作量。尤其对于精密制造业比如飞机，发动机，芯片等，效率就是王道。

. CAE软件架构设计

略，以后通过UML图的状态图 用例图 活动图等来描述一般CAE软件的架构和设计

4. CAE软件研发管理

CAE软件各个功能相对独立，因此很容易模块化，但是集成后各个模块之间耦合度很高。曾经调试过一个case，仿真出现错误，从求解器开始debug，经过Mesh，FEM模型，CAD模型，最后找到问题的原因是CAD单位问题，也有过从求解器debug开始到CAD再又回来，最终问题定位到Mesh上。

人员配置：

不同于常规软件开发，CAE软件开发对研发人员背景有较高要求，需要理解CAE里很多概念。

对于一款常规的专业CAE软件，参考开发人员配置：

1> 系统架构师（1名）

系统架构师主要负责 与应用工程师和客户工程师沟通，进行需求分析，技术选型，概要设计，模块设计，数据结构设计，保证产品按预计进度开发，以及软件仿真流程正确运行。这就要求系统架构师不仅需要有丰富的研发，架构设计经验，而且要对CAE行业有较深刻的理解。

2> 前处理开发（2-3名）

前处理开发主要负责 将CAD模型转化为有限元模型，这个过程中主要是对CAD模型（按照各个行业需求，模板等，快速生成CAD模型）和有限元模型（有限元单元自动设置，网格密度优化，边界自动识别设置等）进行编辑，以及可视化。这块的操作需要 应用工程师的大力协助，因为这块是最需要专业知识的地方。也是对图形学要求最多的地方。

3> CAD开发（1名，可选）

CAD通常需要一个CAD内核，主要是建立和编辑几何模型（三维），可以用开源OCC，也可以用商业ACIS/Parasolid，也可以自主开发，看实际需求。

4> GUI/常用功能（若干）

主要是进行各种界面，以及常用功能进行设计和实现，比如工程文件设计，打开，保存。这块实际上和前处理会有很多重叠的地方。

5> 求解器开发人员（1-2名，可选）

不多说，核心开发。一名开发求解器，一名辅助做benchmark，也需要应用工程师的支持

画外音：如果有一个厉害的求解器就够了，如LS-Dyna和powerflow，各种前后处理器会围着团团转

6>后处理开发（1名，可选）

辅助：

1>应用工程师

2>客户工程师

应用工程师主要负责和系统架构师，客户工程师沟通，讲实际工程需求和客户需求转化为实际可开发需求。可以这样说：应用工程师是整个团队的军师。

敏捷开发可以应用到CAE的研发管理中，早期做原型时，可以一个月作为一个Sprint，半年左右完成原型，以后逐步缩减Sprint时间，转为迭代开发。敏捷开发的思路比较清晰，做法也很明确，可以根据实际需要进行运用。

5. CAE软件的核心技术和风险

目前研发CAE软件并没有太多的技术障碍，通常认为的技术障碍，比如CAD内核，网格，甚至求解器都可以购买商业库。个人认为CAE软件的核心技术在于：**给用户提供一个整套问题的解决方案，帮助用户快速解决实际工程问题，尤其在通用CAE软件解决不了或者解决起来很费劲的领域，软件在这个过程甚至可以只起辅助作用。**当然从长远来，所有的基础开发都不可避免。

在国外尤其是美国，CAE软件研发的生态环境比较完善，高校，各种科研机构，公司，技术人员都能从CAE研发中获得稳定长期的利益，包括开源产品，这也是为什么CAE软件产品能在国外蓬勃发展。

开发专业CAE软件的目的是提高分析效率，但是如果提高的效率的程度不能中和产品研发的成本，就会存在商业上的风险，这是做需求分析时必须要考虑到的。

FEA精度之网格加密(3)--HFSS网格质量分析

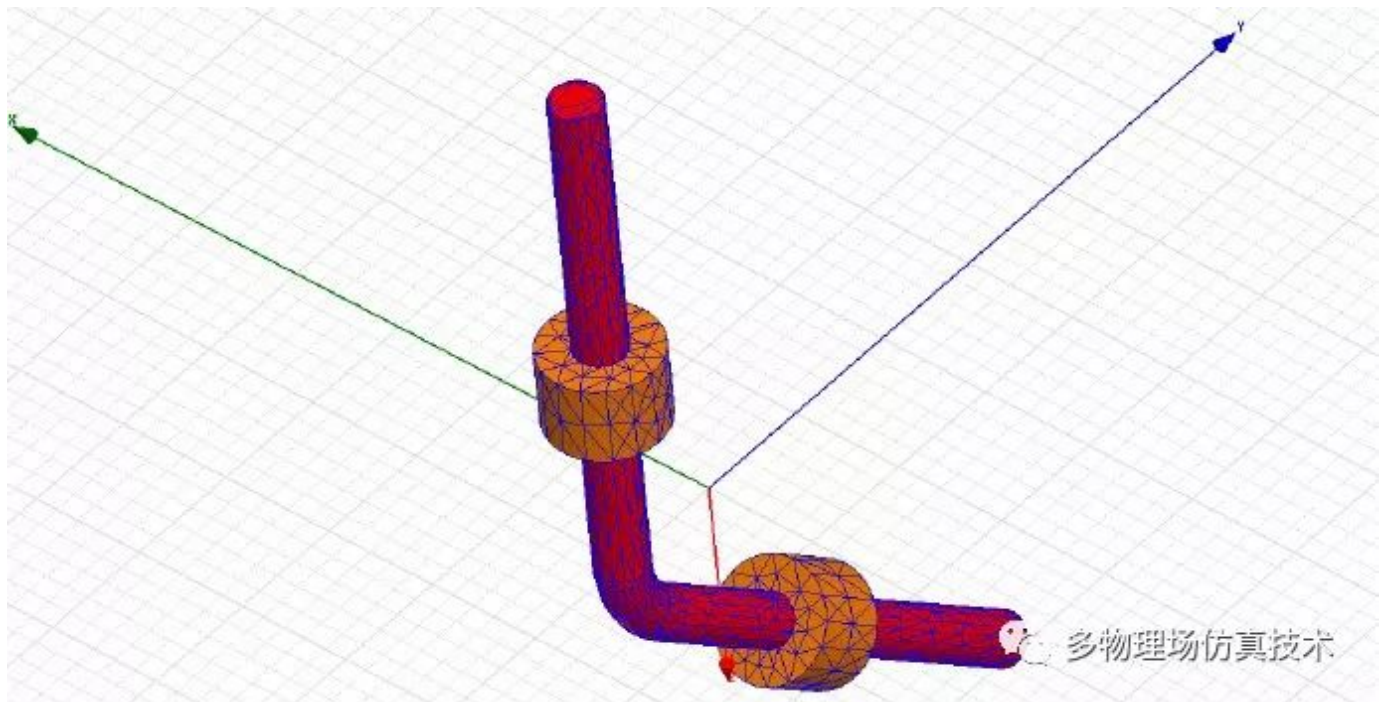
www.cae-sim.com 多物理场仿真技术

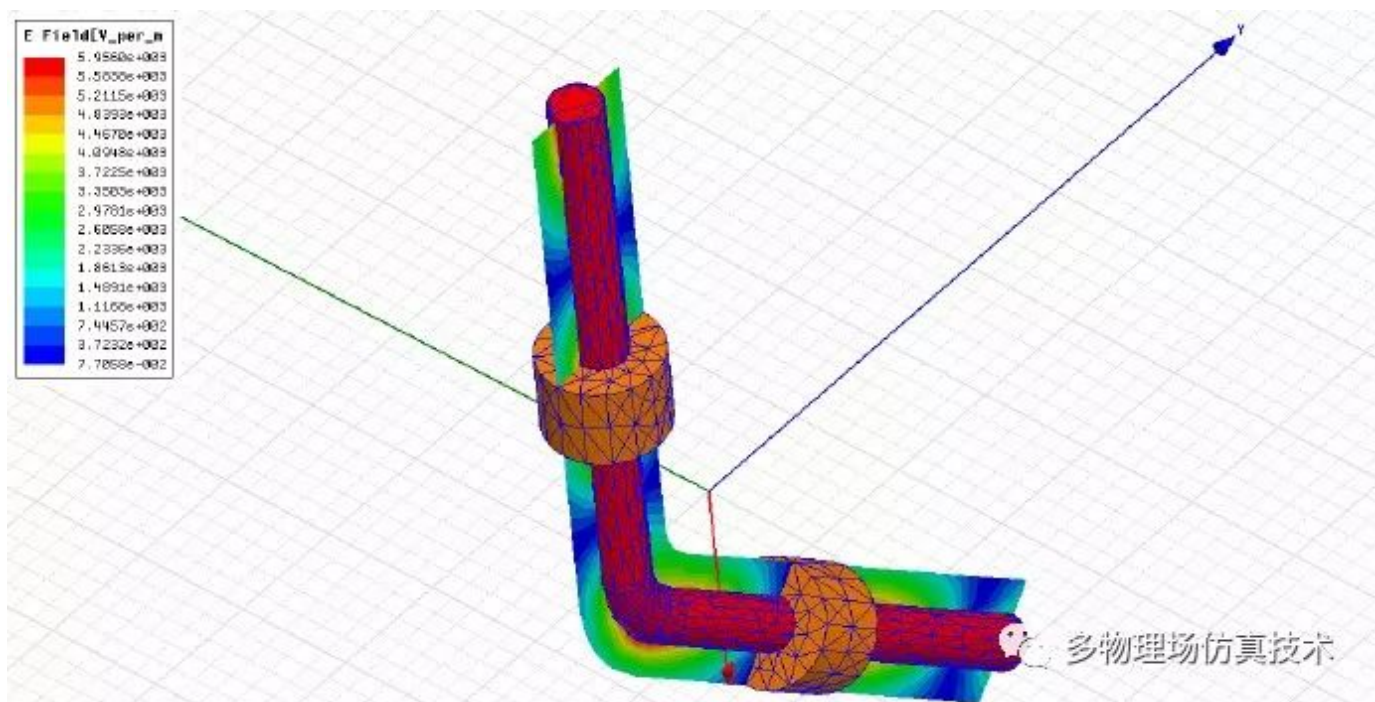
HFSS以其无以伦比的仿真精度和可靠性，快捷的仿真速度和易用的操作界面，成为高频结构设计的首选工具和行业标准。尤其是以自适应网格划分技术可以达到很高的仿真精度。本文就针对HFSS 网格质量进行详细分析。

1. 选取HFSS自带case 仿真
2. 解析HFSS网格文件
3. 将其解析的网格数据转换Abaqus格式
4. 读入Abaqus格式文件，网格质量分析指标：Aspect，最大角度 最小角度, Skew

实现：

1. 导入HFSS例子工程 coaxbend，仿真





2. 仿真结束后，网格文件保存在 result 目录下：

current.pnt

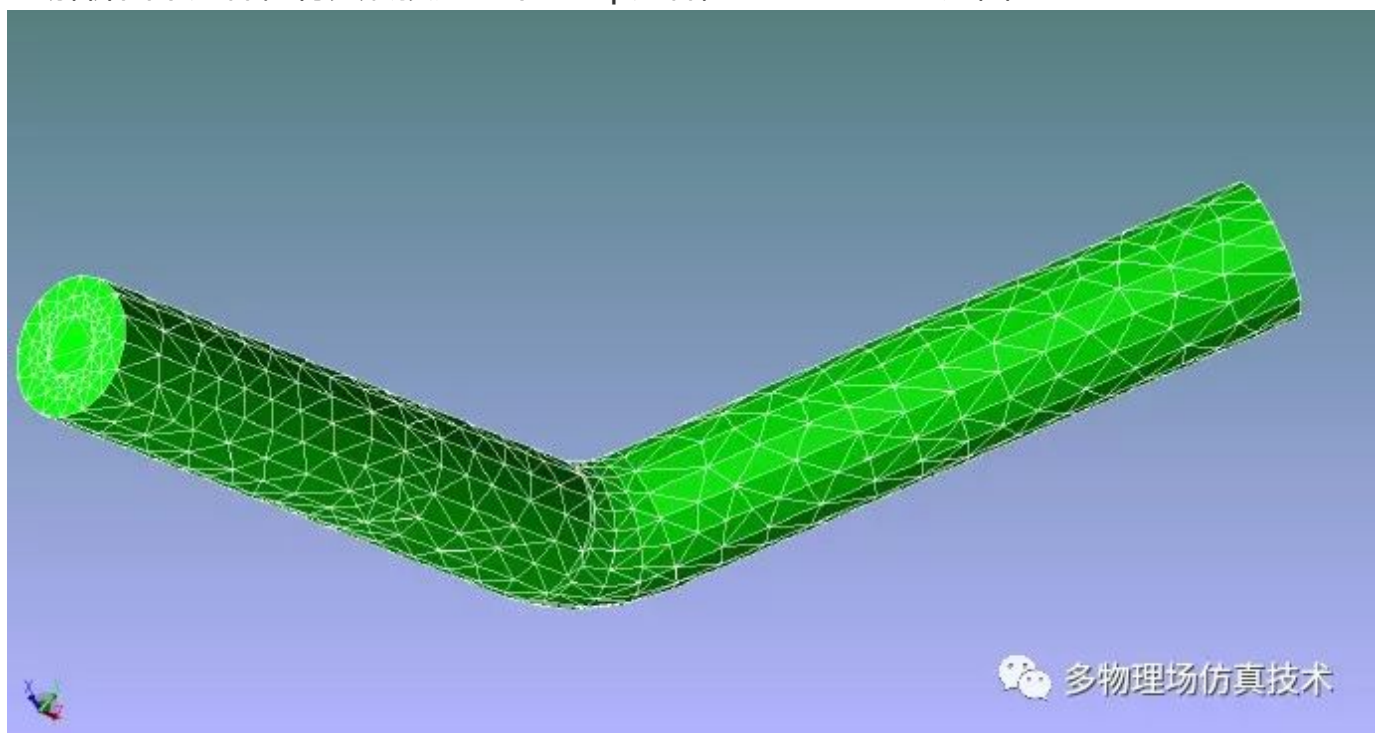
current.lin

current.fac

current.hyd

四个文件分别保存的是点，线，面，体的数据

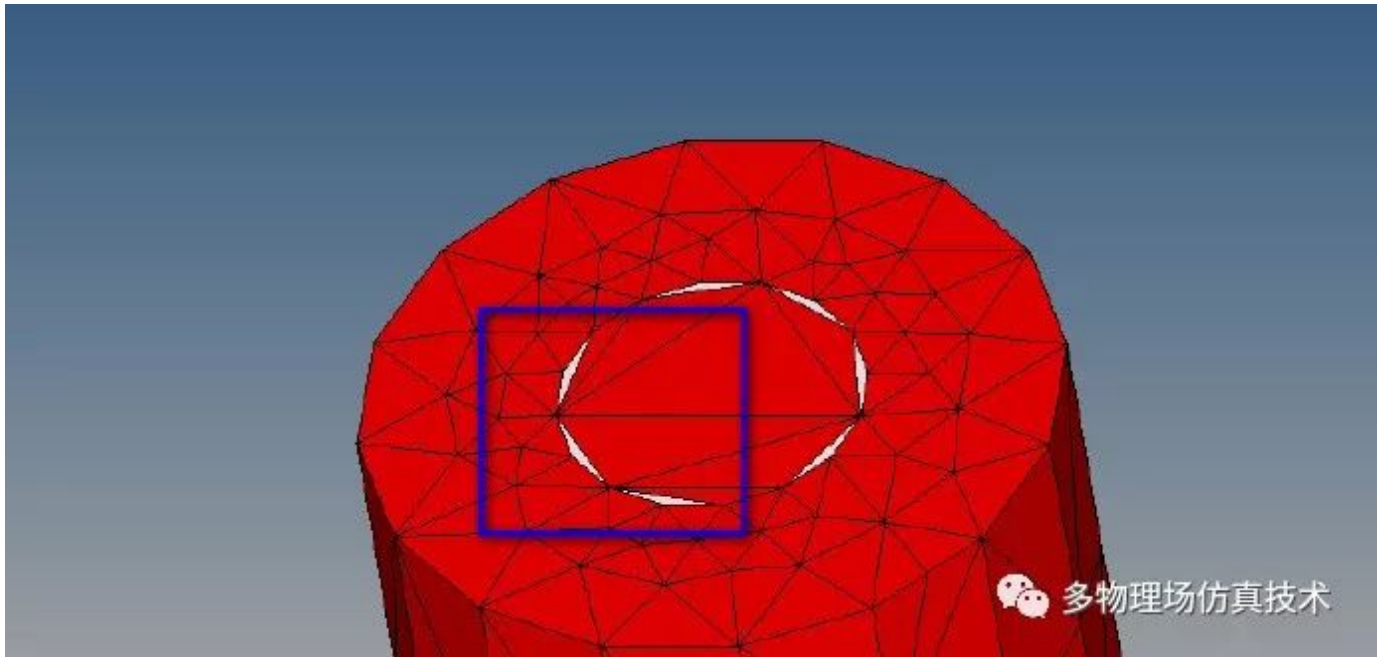
3. 解析四个文件，将其转换成一个 *.inp 文件, FaModel 显示如图



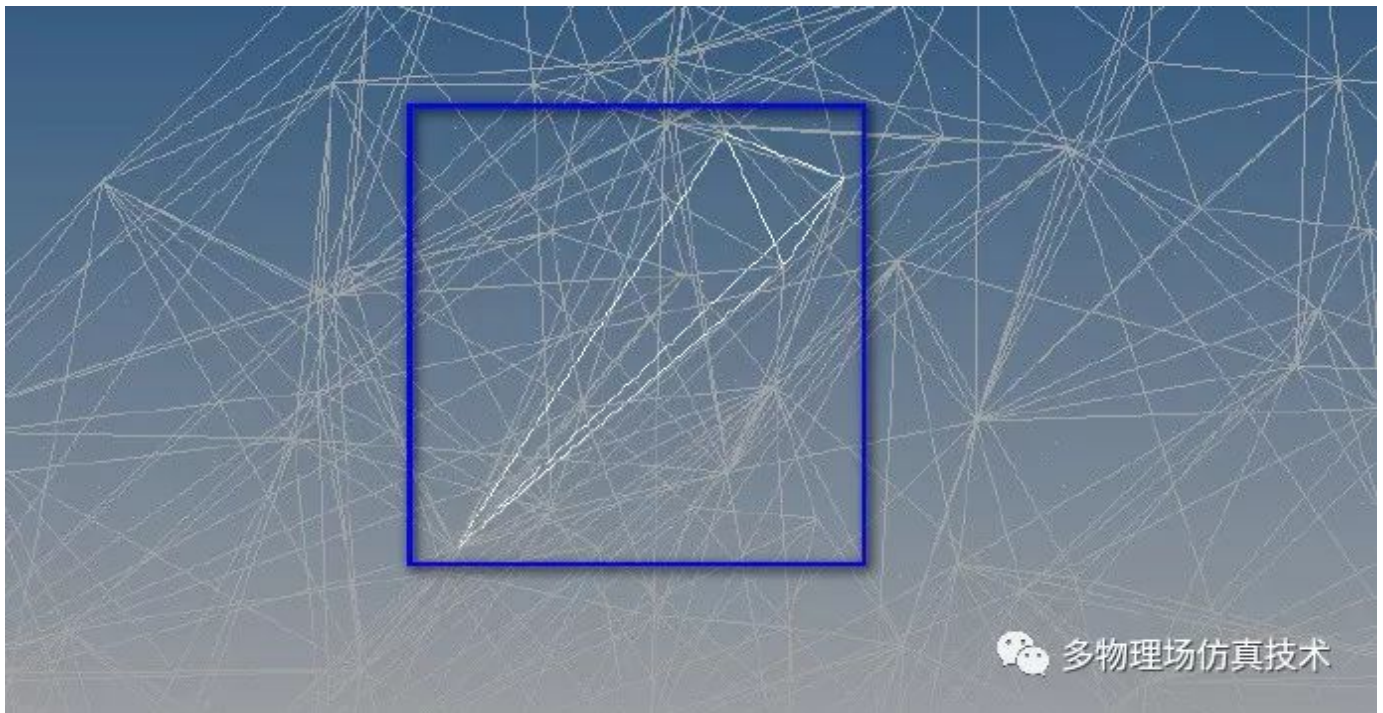
4. 计算四面体单元质量:

设置网格指标标准

- 4.1. Skew > 60, 36%单元达不到
- 4.2. Skew > 70, 12%单元达不到
- 4.3. 最小角度 < 10度, 有 7% 单元 如图



- 4.4. 最大角度 >150度 有 2%的单元 如图



- 4.5. Aspect >5 10%
- 4.6. Aspect >4 21%

一般来说，四面体单元的质量要比面网格差，从以上统计数据可以看出，HFSS生成的网格质量并不算高，还有 7%的单元最小角度小于10度，算是畸形单元。

HFSS中可以设置收敛误差的判断标准，如果该标准设置的不合适的话，从网格质量角度来看，有可能最后会得到不正确的结果。

SimSolid 技术分析

邓子平 多物理场仿真技术



SimSolid 是一款三维结构分析工具，可以处理静力，动力，热等结构问题；不使用传统有限元方法FEM(Finite Element Method)的三角形/四面体/六面体网格，计算时间大大减少，而Altair收购SimSolid更是引起了广泛关注。

公开资料显示，SimSolid摒弃了传统的多项式的基函数构造方法，而使用了几何特征的体积，面，边，点的信息，这是与传统FEM最大区别。当然仅仅只是简单的体，面，边，点的几何信息无法构造出合适的基函数，在原始具有BREP几何基础上，还需要进行更多的数据提取，比如特征识别，参数识别等，让几何信息和附加属性更加丰富，便于用于构造基函数。

要了解SimSolid的底层技术，需要了解一下有限元的理论基础。有限元求解的对象是偏微分方程，理论上是基于瑞兹和伽辽金方法。在这个过程中需要构建一个基函数以及相关的形函数。理论上可以使用各种各样的基函数，而多项式无疑是最简单，最容易实现的。但是这种方法也割裂了与几何的关联，几何中的很多拓扑信息没能有效利用，SimSolid正是从几何入手，强化了对几何数据的利用，降低了对网格的依赖。目前大部分三维几何使用BRep的表达方式，拓扑信息比较丰富，理论上从数学角度找出可靠的基函数不是问题。

既然使用了新的基函数，那么精度就成为一个关心的问题。其实检验这种新函数方法也非常简单，使用不规则的曲线和曲面，检查计算结果，理论上SimSolid的计算结果在这些区域的计算值比精确的FEM要差。因为不规则的圆弧曲面需要更多的离散值来表达，不管是FEM网格还是拟合表达式。当然可以通过加密来解决这个问题，加密会带来性能上的损失，事实上SimSolid也是使用类似COMSOL，HFSS中的自适应网格技术(Adaptive Mesh)来提高求解精度。

精度和速度是仿真软件绕不开的一对矛盾，所以SimSolid的价值还需要更多的实际工程项目来检验。

下面是传统FEM方法和SimSolid的一个简单比较:

	传统FEM	SimSolid
自由度	在节点上	几何上任意
基函数	多项式	体积，面积，点，边等组合
对几何处理	几何清理	对几何进行特征识别，加工
计算资源	高	中，低
计算精度	高	中，高
是否必须加密迭代	否	是
是否需要面片网格	是	否



在本公众号《仿真软件十年回顾和展望》一文中提到了 未来CAD/CAE的无缝结合是仿真软件发展的一种趋势，如果SimSolid的方法能够证明有效解决工程问题，扩展到流体，电磁等其它领域也是早晚的事。

深入解析HFSS自适应网格加密策略

www.cae-sim.com 多物理场仿真技术



网格加密的核心是什么？

如果网格数量偏少，在物理场变化大的地方，网格内部不足以表达这种变化，从而导致计算偏差。类似于用多边形模拟圆形，线段数量越多，越接近圆形。

网格加密通常有以下几种标准：

1. 根据几何特征，比如在曲面，曲线，孔边等加密。这种加密依赖于几何拓扑信息，但并不准确，比如在尖角处网格加密反而得出错误的结果。
2. 根据业务特点，比如流体在边界层，PCB电磁仿真在金属部分需要加密，在初次划分网格的时候就可以进行设置；
3. 根据两次仿真所得物理参数结果，比如在应力/压力/速度/温度变化梯度大的地方；
4. 根据两次仿真业务数据进行加工，比如HFSS根据S参数误差，能量误差。

HFSS网格加密的原理前面做过介绍：

1. 进行第一次计算；
2. 在结果内部搜索误差最大的区域，对网格进行加密（划分更多的网格）；
3. 再进行计算，判断误差是否达到收敛标准；
4. 如果收敛，则结束，否则转到步骤2；

FHSS不同求解设置，所使用的的判断标准不同：

1. 波端口（Wave Port），集总端口（Lump Port）激励使用S参数误差；
2. 电压源，电流源，入射波和磁偏置激励使用能量激励

以集总端口为例，在自适应网格划分过程中，如果两次仿真S参数差 ΔS 小于0.02（默认数值），则认为收敛，且每次网格加密数量为30%（默认值）。

在这个过程中，有几个需要解决的核心问题：

1. 初始网格如何划分；
2. 第一次网格如何加密；

3. 网格加密数量和迭代次数如何设置；

事实上，初始网格划分和第一次网格加密，两次操作基本决定了整体网格质量和仿真计算结果，后续的网格加密是让计算精度更高。

1. 初始网格如何划分

初始网格主要根据物体的材料属性和仿真频率点的波长来确定；

2. 第一次网格加密

第一次网格加密很关键，由于不知道 ΔS 值，只能采用能量计算误差；计算单元电场磁场强度以及单元内部的电场磁场强度差，对这些变量进行简单加权平均，即可得到每个四面体单元的能量误差。对单元单按照能量误差大小进行排序，然后再根据网格加密数量百分比（默认30%）设置，按顺序得到需要加密的单元。加密也有两种方法：一是在原有单元上加密，该方法简单，但是网格质量较差；二是重新划分网格，在原有网格所在区域设置网格参数，该方法实现比较复杂，HFSS使用第一种。

3. 网格加密数量和最大迭代次数如何设置

网格加密百分比默认为30%，这是一个经验值，如无特殊需求，使用该值即可，调大该值可能造成网格加密过多，浪费计算资源，设置过小容易因为网格加密不足，两次迭代差值过小导致伪解。最大迭代次数也可以使用相关默认值。

仿真收敛以后，网格会保存下来，如果需要进行扫频，HFSS会使用已有的网格，在新的频点上不会重新划分网格，这是需要注意的地方。

一篇文章入门三维几何内核

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

在三维建模软件和仿真软件中，我们经常能听到“几何内核”一词。本文就以三维几何为题，从仿真的角度讨论以下几方面内容：

1. 几何内核是什么，有什么作用
2. 几何内核和仿真的关系
3. 现状
4. 作为仿真类公司，有无必要开发几何内核

1. 几何内核是什么，有什么作用

从最简单的三维基本实体“长方体”说起

一个三维长方体，在笛卡尔坐标系中至少有以下种表达方式：

1. 给出长方体的两个顶点坐标即可确定长方体；
2. 给出长方体8个顶点坐标；
3. 给出初始点坐标以及长，宽，高；
4. 如果一条边由2个点确定，给出12条边即可确定；
5. 如果一个面由四个顶点确定，给出6个面即可确定；
6. 给出一个函数表达式，限定任意一点坐标在长方体限定范围内；

1,2表达是同一方法，可以理解为常规表达，3是参数表达，4和5是最广泛使用的边界表达Boundary Representation（简称B-Rep结构），6则是约束表达，通常在建模过程中使用；

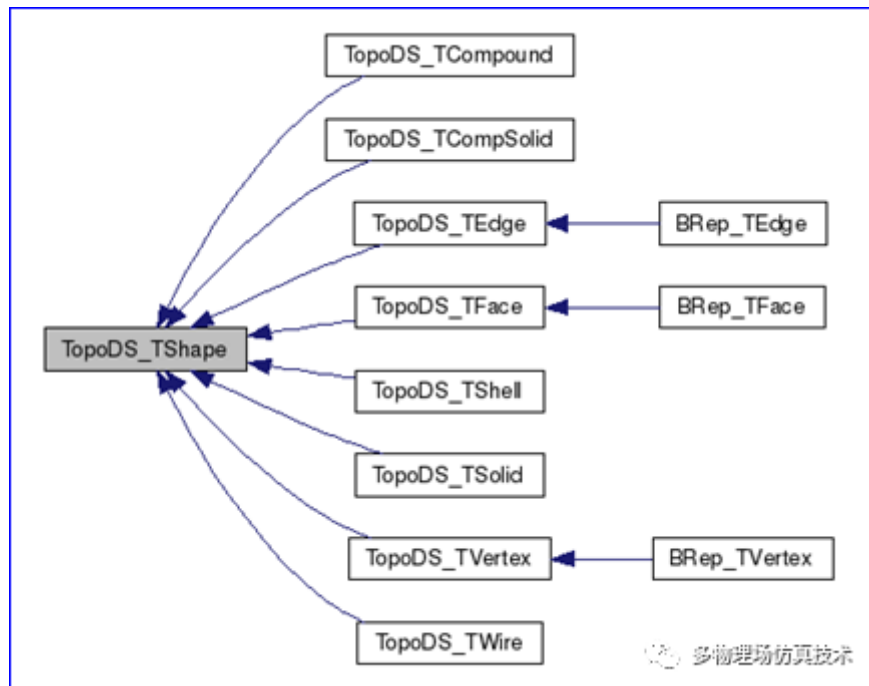
在实际应用中，我们可能还会有如下需求：

1. 编辑修改长方体(修改长宽高，并执行undo和redo操作)
2. 离散面片化长方体用来显示和划分网格
3. 网格化长方体供仿真使用
4. 长方体与别的物体进行布尔（并，减，差）运算；
5. 移动，缩放，旋转长方体
6. 根据业务需求找到理想的长方体长宽高参数，需要多次自动调整参数
7. 复制，阵列，变换该长方体
8. 点/线/面/实体 到长方体的距离/夹角

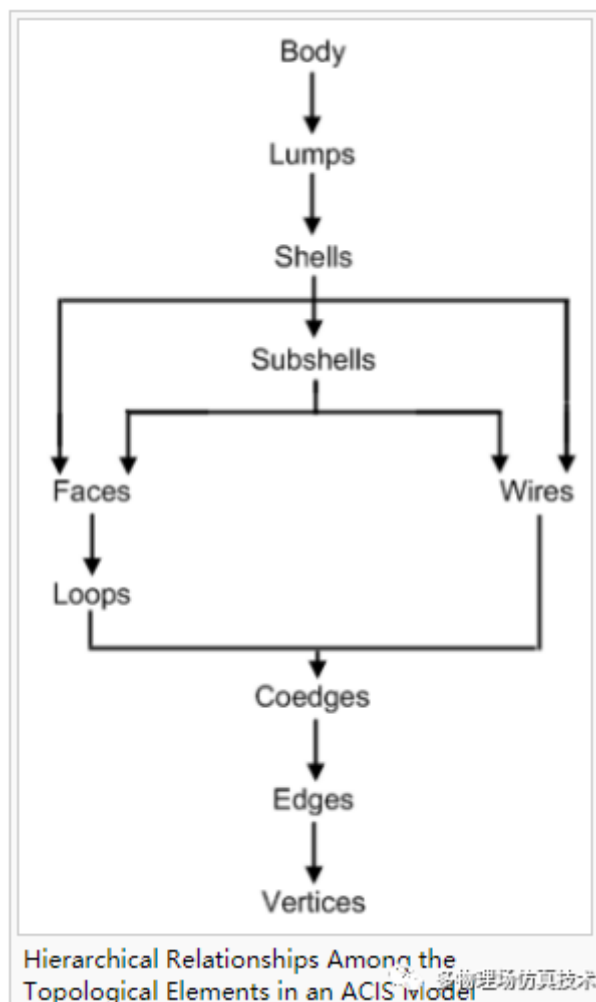
9. 赋值属性(颜色, 材料, 边界条件, 荷载等)到长方体, 或者长方体的面/线/点上
10. 需要对长方体进行切割加工;
11. 将几何导出为中间格式, 供第三方软件使用
12. 计算长方体本身的属性 (质量, 体积, 面积, 重心)

稍微介绍一下B-Rep结构, 主流的几何内核都使用B-Rep结构, 主要原因是这种几何表达信息丰富, 可以满足几乎所有对几何数据的要求, 但缺点也显而易见, 数据过多冗余, 在大规模数据上, 性能是一个很大的问题。B-Rep就像是工具中的瑞士军刀, 什么都能做。B-Rep 边界表示方法, 把所有的三维实体都通过封闭的边界面表示, 比如长方体可以用过六个面表达。

开源Opencascade的拓扑结构表达如下:



ACIS的拓扑几何定义类似:



更多边界表示法（Boundary Representation）B-Rep的详细信息可以[百度谷歌](#)。

由此可见，几何内核的作用在于表达，处理几何对象，对外提供接口。以两个长方体的布尔运算为例，如果只知道顶点坐标信息，布尔运算无法进行，顶点也无法表达出计算结果。只有知道了边/面，位置等足够信息后才能进行，几何内核的实体表达可以提供这些信息。

此外，好的几何内核提供了约束建模，参数建模，驱动建模，事务等各种底层机制，方便了上层CAD应用的开发。

2. 几何内核和仿真的关系

在数值计算仿真中，求解器的输入是离散的网格数据，网格数据来自于几何。好的几何内核可以提供足够的信息给网格划分引擎。

对于二维和三维的网格，网格引擎需通常要将几何离散的三角面片作为输入，或者要拿到面的拓扑几何数据，也就是面，边，顶点或者环和shell信息。在网格生成过程中有几个地方需要注意：

1. 几何数据需要保证正确，如果几何有错误，比如有非常细长的面片（最长比最小大于1000），实体几何之间有干涉，即相互重叠，存在自由面片，几何拓扑不完整不连续等等。这些会导致网格划分出现错误，或者网格划分出来也不能用于仿真计算。

2. 几何属性和网格属性的映射。即把几何实体上的所有属性信息也要赋值到网格的属性上。比如材料信息，边界荷载等相关信息。对于复杂的几何，属性查找会带来性能问题，需要在网格划分过程中做好这件事。

3. 几何网格映射参数化。通常情况下我们需要修改几何，由此带来网格的改动。在参数驱动和自动化过程中，几何会多次改动造成整体几何变动以及网格频繁变化，这就要求几何和网格的接口要足够稳定。

4. 网格参数设置。合理的参数设置可以在保证仿真精度的前提下大幅提高仿真效率，目前大部分网格划分需要人工干预，或者进行自适应网格加密等，网格划分参数智能化是一个很有挑战的技术，也是人工智能，深度学习在仿真领域应用的一个技术点。

3. 现状

目前主流的商业三维几何内核商业有ACIS，Parasolid，这两个三维内核同宗同源，只是两个发展分支，ACIS的Spatial公司被达索收购，而Parasolid经历多次收购现收归西门子。现在的两个商业内核除了加强底层内核功能，ACIS更偏向于CAD/CAE/EDA/CAM等客户应用领域，而Parasolid更偏向于加强西门子自身业务的支撑，在CAD功能上，Parasolid要优于ACIS。

目前市面上的大多数三维CAD软件，和拥有三维几何造型仿真软件几何内核都采用ACIS或者Parasolid。

目前开源的几何内核有法国公司推出的Opencascade，对于小型项目的开发是个不错的选择。

<https://www.opencascade.com/>

国内的中望，广联达公司拥有自主开发的三维几何内核，主要用作支撑其自身业务。

不少开源类库都有其自己的几何定义，但是仅仅用作自己业务，没有通用性，算不上几何内核，比如Gmsh，CGAL等等。

俄罗斯作为一个科技大国，自然也有其开发的几何内核，但是技术生态圈封闭，在和主流CAD软件数据文件交互中问题不少，开发的话只能用其部分功能。

4. 作为仿真类公司，有无必要开发几何内核

从笔者自身开发经验看，如果长期在自己的仿真领域深钻，几何内核开发是绕不开的。笔者用过ACIS/Parasolid/OCC，并有过开发几何内核的经验。但从表面上看，几何内核开发没有太多技术瓶颈，已有的功能需求都有解决方案，但实际问题是几何内核涉及的内容过多，而且作为底层技术支持，对稳定性，可靠性要求非常高，短期内无法开发出像样的产品，所以开发几何内核需要经过实际迭代，长期技术积累，研发前期投入巨大。

前面讲过BRep结构就类似瑞士军刀，似乎什么都能做，但对于很多业务来说，数据冗余度很高。根据自身业务定义几何拓扑结构更高效，比如Solibri公司的产品，通过自定义三维几何结构，将解析显示三维IFC文件做到了全球最快，如果使用ACIS的话，时间是其几十倍。

而类似OCC的开源产品，无论从功能，稳定性，性能上，都难以用于大规模的实际商用，而且一旦出了问题，几乎没有解决方法。

在当前国内大力自主发展技术，知识产权保护趋严的背景下，几何内核作为工业软件的一项基本底层技术，应该被拿到产业链的战略地位上来，否则永远只能在国外产品基础上开发，利润大头送给他人。

(非) 线性方程组求解库大全

原创 www.cae-sim.com 多物理场仿真技术

介绍下线性方程组工具，作为以后介绍HPC基础。

问题：

求解线性方程组 $Ax=b$.

Matlab 中提供了方程的求解方法：

$x=A\backslash b$ ($\backslash$ ----反除符号)

最简单的问题，一群兔子和鸭子，一共有40条腿，20张嘴，问兔子鸭子各多少只，两个变量，两个方程组成一个简单的线性方程组：

$$4x+2y=40$$

$$x+y=20$$

在Matlab中输入：

```
>> A =
```

```
4    2
```

```
1    1
```

```
>> b=[40, 15]'
```

```
b =
```

```
40
```

```
15
```

```
>> x=A\b
```

```
x =
```

```
5
```

```
10
```

当方程求解规模达到1000*1000的时候，Matlab还能应付，速度也不错。当规模达到10w*10w的时候，Matlab会运行相当长时间，然后弹出一个错误。这是因为对于10w*10w的矩阵，用原始的数据结构，性能会下降很快。由有限元计算的矩阵特征可知，矩阵是数据大部分为0的稀疏矩阵，Matlab提供了稀疏矩阵功能，在使用 $x=A\backslash b$ 之前使用如下命令：

```
x=sparse(x);
```

10w*10w的方程组1秒之内求出结果。

对于大型线性方程组的求解，使用Matlab做验证是个不错的选择

回到本文的主题：

1. BLAS(Basic Linear Algebra Subprograms)

blas 是许多数值计算软件库的核心，用Fortran语言写成，库一共有三个level，第一个level包含了vector计算的一些function；第二个level则是矩阵与vector计算；第三个

level是矩阵与矩阵的计算。BLAS接口稍微复杂，一般很少直接使用。

开源

2. Intel MKL

商用是不错的选择。

Intel数学核心函数库（MKL）是一套高度优化、线程安全的数学例程、函数，面向高性能的工程、科学与财务应用。英特尔 MKL 的集群版本包括 ScaLAPACK 与分布式内存快速傅立叶变换，并提供了线性代数 (BLAS、LAPACK 和 Sparse Solver)、快速傅立叶变换、矢量数学 (Vector Math) 与随机号码生成器支持。

主要包括：

- ① LAPACK (线形代数工具linear algebra package)
- ② DFTs (离散傅立叶变换 Discrete Fourier transforms)
- ③ VML (矢量数学库Vector Math Library)
- ④ VSL (矢量统计库Vector Statistical Library)

MKL的主要功能

2.1) BLAS 和 LAPACK

在英特尔处理器中部署经过高度优化的基本线性代数例程BLAS (Basic Linear Algebra Subroutines) 和 线性代数包LAPACK (Linear Algebra Package) 例程，它们提供的性能改善十分显著。

2.2) ScaLAPACK

ScaLAPACK是一个并行计算软件包，适用于分布存储的MIMD并行机。ScaLAPACK提供若干线性代数求解功能，具有高效、可移植、可伸缩、高可靠性的特点，利用它的求解库可以开发出基于线性代数运算的并行应用程序。ScaLAPACK 的英特尔MKL 实施可提供显著的性能改进，远远超出标准 NETLIB 实施所能达到的程度。

2.3) PARDISO稀疏矩阵解算器

利用 PARDISO 直接稀疏矩阵解算器解算大型的稀疏线性方程组，该解算器获得了巴塞尔大学的授权，是一款易于使用、具备线程安全性、高性能的内存高效型软件库。英特尔 MKL 还包含共轭梯度解算器和 FGMRES 迭代稀疏矩阵解算器。

2.4) 快速傅立叶变换 (FFT)

充分利用带有易于使用的新型 C/Fortran 接口的多维 FFT 子程序（从 1 维至 7 维）。英特尔 MKL 支持采用相同 API 的分布式内存集群，支持将工作负载轻松地分布到大量处理器上，从而实现大幅的性能提升。此外，英特尔 MKL 还提供了一系列 C 语言例程（“wrapper”），这些例程可模拟 FFTW 2.x 和 3.0 接口，从而支持当前的 FFTW 用户将英特尔 MKL 集成到现有应用中。

2.5) 矢量数学库 (VML)

矢量数学库 (Vector Math Library) 借助计算密集型核心数学函数（幂函数、三角函数、指数函数、双曲函数、对数函数等）的矢量实施显著提升应用速度。

2.6) 矢量统计库—随机数生成器 (VSL)

利用矢量统计库 (Vector Statistical Library) 随机数生成器加速模拟, 从而实现远远高于标量随机数生成器的系统性能提升

商用

3. Eigen

主页:

http://eigen.tuxfamily.org/index.php?title=Main_Page

Eigen 是一个轻量级的库, 使用只需头文件, 接口比较齐全, 文档也很全面, 一般的矩阵操作运算都能满足。提供了DenseMatrix 和 SparseMatrix, 对于稀疏矩阵的效率也不错。

开源

4. Spoolse

早期很多人使用, 性能不错, 使用简单。维护的不是很好, 文档使用ps格式。

<http://www.netlib.org/linalg/spooles/spooles.2.2.html>

开源

5. Lapack

使用最广泛的线性代数包, 底层调用 BLAS, 能求解 $AX=b$, 矩阵分解、求逆, 求矩阵特征值、奇异值等。文档齐全, 使用灵活, 稳定性好, 用作研究和商业开发都是不错的选择。提供SVN源码下载

<http://www.netlib.org/lapack/>

开源

6. Pardiso

PARDISO是为解决大 稀疏对称和非对称线性系统的方程的软件包, 如下特点: 线程安全, 高性能的, 和内存使用率低, 支持共享内存和内存多处理器。

<http://www.pardiso-project.org/>

商用, 学术用免费

7. Mumps

使用多波前法的稀疏矩阵求解库, 支持并行计算。

<http://mumps.enseeiht.fr/>

开源

8. PETSc

PETSc是一个高大上的科学计算库, 求解线性方程是其中一个功能, 支持内存共享并行计算机, 支持多线程, GPU加速等, 支持求解稀疏矩阵

<http://www.mcs.anl.gov/petsc/documentation/linearsolvertable.html>

开源

9. SuperLU

SuperLU是一个求解大规模, 稀疏, 非对称系统对线性方程组的通用库,c语言编写, 可供Fortran和C调用。最新版本为4.3 (2014/10/1) 支持并行计算和分布式计算。

<http://www.cs.berkeley.edu/~demmel/SuperLU.html>

开源

10. Umfpack

UMFPACK是 用来求解不对称稀疏线性系统软件包, $Ax = b$,使用非对称多波方法, SuiteSparse有此算法

使用比较简单, Windows下需要自己编译。

<http://www.cise.ufl.edu/research/sparse/umfpack/>

开源

11. TAUCS

求解稀疏矩阵线性方程软件包, 最新版本支持多线程

<http://www.tau.ac.il/~stoledo/taucs/>

开源

12. SuiteSparse

SuiteSparse是一组C、Fortran和MATLAB函数集, 用来生成空间稀疏矩阵数据。在 SuiteSparse中几何多种稀疏矩阵的处理方法, 包括矩阵的LU分解, QR分解, Cholesky分解, 提供了解非线性方程组、实现最小二乘法等多种函数代码

<http://faculty.cse.tamu.edu/davis/suitesparse.html>

开源

13. Sparselib++

用C++写的 稀疏矩阵库, 可以和 IML++一起使用做线性方程组的迭代求解。

<http://math.nist.gov/sparselib++/>

开源

14. Trilinos

Trilinos也是个高大上的东西, 求解线性方程只是其中一部分功能

<http://trilinos.org/capability-areas/#ScalableLinearAlgebra>

开源

15. IML++

使用C++模板库 和迭代方法 求解对称, 非对称矩阵库

<http://math.nist.gov/iml++/>

开源

使用推荐:

1. 简单快速上手 Eigen
2. 商业使用 Intel MKL
3. 研究使用 Lapack, PETSc, SuiteSparse

一篇文章入门多物理场有限元（全篇）

原创 邓子平 多物理场仿真技术

收录于话题

#仿真研发工具 21 #求解器开发 22 #应用工程师 4



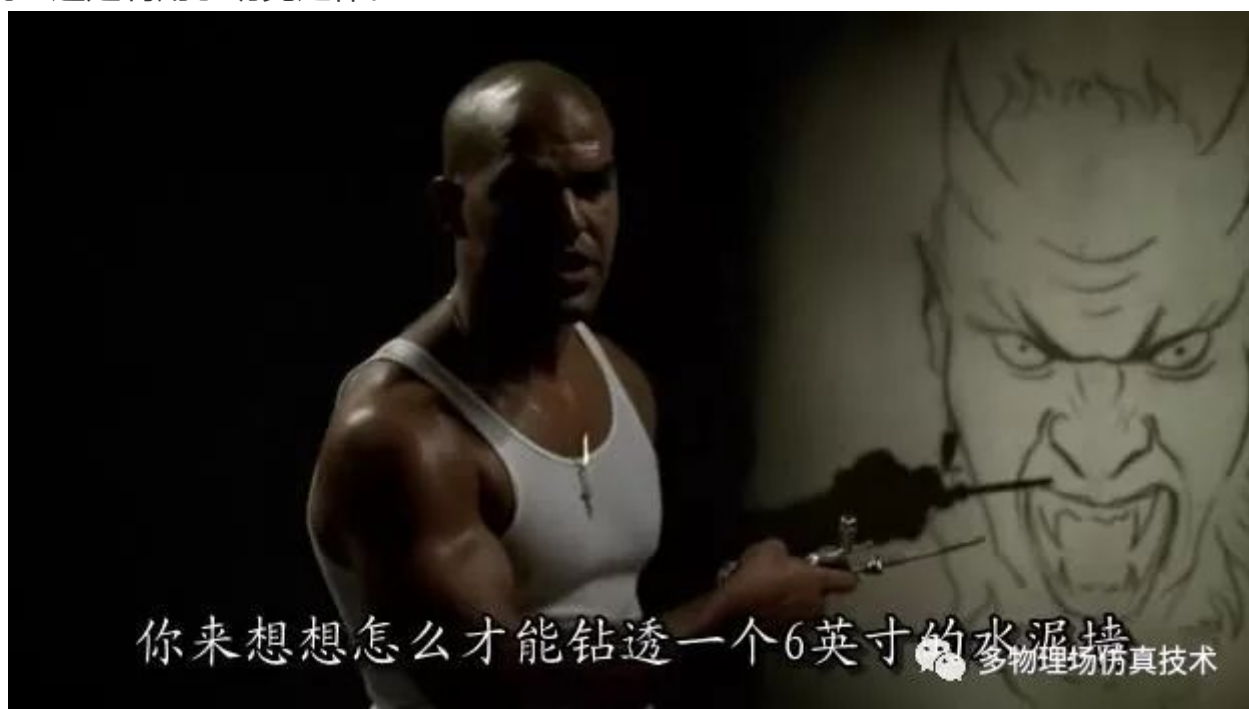
导读

根据部分读者的反馈做了改进，增加了商业对多物理场仿真软件的介绍。本文主要写给从事CAD/CAE/EDA/CFD等相关的软件研发测试人员，也可以作为有限元入门文章，对于从事仿真分析的工程师也可以对软件实现细节有所了解。

-----前言-----

从《越狱》里的胡克定理说起

看过美剧《越狱》的朋友对其中一个桥段应该印象很深刻，男主让狱友将事先画好的一张图铺在墙上，然后用打蛋器在固定点打孔，最后将墙敲穿推倒。狱友问这是什么原理，男主说：这是利用了胡克定律。





高中物理课本对胡克定律的定义是：

弹簧在发生弹性形变时,弹簧的弹力 F 和弹簧的伸长量(或压缩量) x 成正比。

很显然，如果仅仅用胡克定律无法解释几个小孔就能把墙凿穿。事实上男主角应该是事先拿到了墙的结构图，算好了砖块的距离，在砖头的缝隙处打孔。至于说原理的话，一张白纸很难从中间撕开，但如果纸中间有破损，沿着破损处就很容易撕开，非要扯上一个定律原理的话，孔边应力集中原理可以很好地解释这种现象。所以男主所说的胡克定律只不过是搪塞狱友的一个借口~

广义胡克定律

在材料的线弹性范围内，固体的单向拉伸变形与所受的外力成正比。

高中物理课本对胡克定律的定义可视为广义胡克定律的简化。

-----从刚体到非刚体-----

一根柔软的钢尺一端固定，一端悬空，悬空的一端会发生明显变形。但是高中物理只能告诉我们固定的一端受到向上的力以及力矩，无法告诉悬空一端的变形大小，这是因为其研究对象的属于刚体范畴，即研究的对象本身不会发生变形，而现实中绝大部分物体受力时自身会发生变形，工程力学中使用应力，应变，位移来度量对象的受力情况，而材料需要引入弹性模量和泊松比。有限元分析的几何对象是物体本身。

-----从一维到二维-----

弹簧是典型的一维单元，扩展到二维需要引入偏微分方程PDE。拉普拉斯方程和泊松方程是最简单的二维偏微分方程。这两个方程可以描述静电场，导热，二维波动，以及天体运动等。简单讲，PDE (Partial Differential Equation) 是描述这个世界运行规律的一种方式。从量子原子微观运动，到宏观热，声音，电磁，以及天体万有引力规律，都有各自的方程来描述。声场对应赫姆霍兹方程，电磁波对应麦克斯韦方程，流体对应纳维斯托克

方程，热传导对应泊松方程，量子力学对应薛定谔方程等等，而这个方程基本都是PDE。我们所学的各种自然定律规律，比如牛顿三定律，浮力定律等只不过是常微分方程的简化，而常微分方程则是PDE的简化。

有限元方法求解的对象是PDE，有限元方法的本质是将求解对象离散成小的单元，在每个小的单元上用基函数和形函数表征其坐标和物理量，使函数在其自由度所在的网格单元几何上满足偏微分方程的解（之所以是几何不是顶点，是因为自由度不一定在顶点）

-----有限“单元”篇网格-----

网格是个很广泛的称谓，为了避免歧义，FEM中的网格做如下规定：

1、网格的英文名为Mesh，非Grid

Grid 英文翻译也是网格，但是基本意思为格子，通常指结构化网格，即四四方方非常规整的网格。

2、网格是用于有限元计算的输入数据

网格对应于有限元的单元数据；线单元，面单元，实体单元都称之为网格（对，线单元的一条直线也是网格）

3、几何离散成的三角形称之为“面片”，非网格

大部分显示引擎底层都使用三角形来渲染对象，因此2维、3维几何都需要三角化（称之为“面片化”），比如一个长方体共有6个面，每个面需要离散成两个三角形，总共12个三角形。这12个三角形我们称之为12个“面片”，而非网格！一般情况下“面片”质量很差。“面片”有两个功能：一是用来做显示渲染数据；二是可以作为网格划分的输入数据，用来生成面网格。

4. 网格计算泛指“分布式计算”

“网格计算”（Grid Computing）是分布式计算的一种，它研究如何把一个需要非常巨大的计算能力才能解决的问题分成许多小的部分，然后把这些部分分配给许多计算机进行处理。这里的网格翻译也是Grid，和有限元计算无关！

-----单元-----

有限元方法的输入基本对象是“单元”（element），即几何模型被离散之后形成的网格模型，通常根据单元的几何特点可以分为0维（点单元），1维（线单元），2维（面单元）和3维（实体单元）

实体单元（3维单元）很容易理解，因为任何物体都是3维的，非实体单元通常是对模型的抽象和简化。这种抽象和简化基于一定的前提假设，能在不降低计算准确度的前提下大幅降低计算时间。比如常见的梁单元，可以将实体梁简化为线单元，大大减少计算量。

弹簧只能在一个方向上发生变形，是典型的1维单元；同理壳单元（shell）需要XY两个方向来定义，是2维单元；四面体，六面体是3维单元，也称为实体单元；对象可以看做

质点的为0维单元，比如称之为“定楼神球”的调谐质量阻尼器。

单元的阶数：

以三角形为例

一次线性单元为3个顶点

二次单元为6个点，包括3个顶点和3条边的中点

三次单元有9个点，每条边上有2个点

为了方便，统一将线性单元称之为0阶单元，二次单元称之为1阶单元，以此类推。

-----边界条件 (Boundary Condition) -----

有限元求解的对象是偏微分方程 (Partial Differential Equation)，理论上偏微分方程的通解有无数多个，但实际上确定的工况下模型只有唯一一个确定解。而决定最后唯一解的就是边界条件！简单理解就是通解是一个带未知数的函数，而边界条件可以求解出这些未知数！未知数的个数和边界条件能确定的数值个数相同。

边界条件按照数据类型分为三类：

- 1、只有确定数值的，比如确定的电流，压力，温度，位移等数值，称之为第一类边界条件，英文为Dirichlet
- 2、用函数表示的，函数可以用导数，积分表示的任意函数。比如随时间变化的荷载，电压；用导数表示的换热系数，称之为第二类边界条件，英文为Neumann
- 3、第一类和第二类的混合，称之为第三类边界条件，英文为Robin

很多书将三类边界条件用中文表示，但翻译有偏差，比如Dirichlet，翻译有狄里克雷，狄里克莱，狄立克莱，狄力克莱，狄里赫利。

对于工程中使用的英文名称，建议统一用英文或者无歧义的中文（第N类边界条件）表示，不使用英译。

边界条件的设置基于对有限元模型的正确理解。三维电磁分析时，设置的理想吸收边界需要将求解域包围住；施加的Port激励根据类型设置在不同的几何上；结构中的位移约束和荷载不能同时施加在同一几何上；流体中的进口速度压力和出口速度压力边界要与计算使用的CFD模型类型匹配。

在实际使用仿真软件时，用户无需过多关注第几类边界条件，因为在软件中边界条件统一使用离散的方式输入，即用户在设置边界条件时，通常是选中某个几何（比如选中一个面），然后通过对话框输入数据，将属性数值直接关联在该几何上，对于用函数表达的第二类，第三类边界，通常也是采用用户输入数据，采用简单拟合插值的方法得到表达式。同时商业软件中也会根据实际情况，提供多种便利的边界设置方法。比如要在100片相同的叶片上设置相同的荷载，可以先定义一个荷载，然后一起关联到所有几何上。这也是商业软件的价值所在：提高前处理效率！

在开发边界条件处理功能时，本质上是有共性的：即将数值和几何关联（最终是将数值和网格自由度关联），不同的只是数值的单位和类型。所以基类即可以实现大部分功能，节省开发工作！拟合插值功能也可以做成公用模块，插值方法无非线性，二次，对数插值常用的几种。

正确的使用单元和设置边界条件是得出正确计算的重要条件！

-----单元设计（针对软件开发）-----

ANSYS和ABAQUS中都有上百种单元，根据前面介绍，单元分类按照空间维度有0维，1维，2维，3维

按照面向对象设计方法，可以设计出四个接口基类；按照阶数来分，至少有0阶，1阶和2阶，阶数可以作为属性放在实现类中；引入自由度，单元设计会稍微复杂一点，每个节点自由度根据特定单元类型不同而不同，比如有些节点只考虑平动，不考虑转动，有些节点只考虑磁场不考虑电场；再引入多物理场，单元设计会更加复杂，以三维四面体单元为例，对于温度，0阶单元可作为默认单元，对于结构0阶单元误差过大，默认为1阶单元，而对于电磁分析由于自由度在边不在点上，0阶1阶2阶都适用，可根据几何实际情况选择。

早期的有限元分析程序多采用FORTRAN编写，没有采用面向对象思想和方法，面向过程的好处是程序流程会比较清晰，任何时候都能对单元的类型结构状态数据有准确的定位，但是难以维护和扩展。当使用类似面向对象C++语言设计大型有限元程序时候，不能抛弃面向过程的设计方法！过多依赖面向对象，从工程角度讲会造成软件开发的不可控，虽然从程序角度看会带来程序的灵活性，扩展性！

对于大型通用有限元分析程序，合理的设计单元架构是有效建立有限元模型的关键，单元框架需要同时供前处理和求解器使用！

-----误差-----

误差天然存在，误差不是错误，但是会导致错误

以实体建模为例，网格划分精度通常为双精度，也就是double型；而渲染显示（比如OpenGL）精度为单精度为float型。创建一个长方体，长方体的顶点坐标如果用鼠标从屏幕上取渲染数据值，即为float型数据而不是double型，生成的几何数据精度会丢失，直接结果就是生成的长方体会和其它相接触的实体产生干涉，如果不进行修复，划分网格就会在边界处生成非常细长的三角形，最终导致畸形网格，求解器无法进行计算。

有限元方法中有三种误差：

离散误差，累积误差和截断误差。 多物理场仿真技术

对于开发人员而言，需要注意累计误差和截断误差，比如等参单元使用的高斯积分，其积分误差和选择的积分点数和位置有关。在求解器求解线性方程组时，采用的不同的求解方法，其最终产生的误差也不相同。

网格的数目，密度，形状，不同的离散方法，不同的求解方法等最终产生的误差很可能导致完全错误的结果。对于研发而言，每一步产生的误差都需要仔细分析，给出处理逻辑。

-----求解器干了什么-----

有限元模型生成后，通常会导出保存为一个文件，文件中保存了节点，单元，材料，边界，荷载，工况，求解等设置信息，求解器会读入该文件进行计算。ANSYS为cdb，ABAQUS为inp，PATRAN为bdf。

一般商业软件都会把求解器做成单独的可执行程序（*.exe程序），单独启动进程求解，而不是集成在前处理器中。一是单独的进程便于管理和分布式计算；二是解耦前处理器和求解器，方便调试；三是可以方便导入到第三方工具中，缺点是模型文件容易被非法查看，可以通过二进制和加密来解决。

求解器读入有限元模型后，首先会检查模型的有效性，比如网格节点是否正确（重复节点，重复网格，索引错误等），还会检查网格质量，最小角度，边长，Aspect ratio，skew ratio 以及对应的求解配置是否正确，比如边界是否设置，工况是否合理（振动频率阶次过多），材料是否有效（比如泊松比大于0.5）等等。当出现不合理设置时，给出警告信息；出现错时，则返回不计算。模型检查合格之后，根据求解器设置，计算每个单元的刚度矩阵，然后组装刚度矩阵成全局矩阵，最终建立起一个非齐次线性方程组，最后求解该线程方程组。

在整个过程中最耗时的是求解方程组。受限于软硬件，求解大规模线性方程组仍然是世界性技术难题，目前对于上了十亿自由度的方程组，除了硬件GPU，分布式计算，超级计算机硬件加速外，并无快速有效的求解方法。显式计算方法不要求解大规模线性方程组，更适合分布式计算。期待不久的将来量子计算能解决此类问题！

求解器计算完成后，会将计算自由度所在点的原始数值输出到文件。之所以说是原始数值，是因为求解器得到的可能并不是用户最终关心的数据，还需要进行一系列处理，比如有些求解器算出点的结果数据为积分点上的值，而非顶点值；用户通常关心Vonmises应力而非最大最小主应力；电路设计用户只关心SNP参数；天线设计用户更在意电磁场场强分布，等值面通常比单个点值更直观等等，所以需要对原始结果数据进行加工处理，给出对用户最有价值的结果！

-----什么是多物理场-----

工程领域的多物理场通常是指宏观领域的物理场，包括电磁，流体，结构，热，声，光（光本质上属电磁）等等，偏向于工程应用领域。需要和理论物理中的四大物理场（微观领域的强力，弱力，以及电磁力和天体的万有引力）相区别。

-----形函数，基函数，伽辽金-----

伽辽金，俄罗斯人，伽辽金方法一如俄罗斯的武器，便宜，耐用，适应能力强，保养简单。

形函数（shape function）是用一个试探性的函数来表达单元内的偏微分方程，通常取多项式。

基函数（basis function）是用来表达一个单元内的物理量的坐标表达式。

伽辽金方法是加权余量法的一种（加权余量包括 配点法 子域法 矩量法 最小二乘法 伽辽金法），将形函数和基函数设为相同，同时将偏微分方程强形式变成弱形式，降低了求解限制，对于非连续介质，只要在插值点满足要求即可，提高了其适用范围。相对于其他方法，伽辽金方法最终生成的线性方程组的系数矩阵，稀疏对称，更容易求解。**简单讲，伽辽金方法简化了有限元方法使用门槛，搭建起了有限元数值理论到软件实现的桥梁，成为早期商业软件实现的基础。**

-----等参单元/高斯积分-----

等参单元：划分网格后，一般网格并不是标准网格（等边三角形，等边四面体或者立方体），直接使用形函数会造成计算上的困难，所以通常使用一个新标准网格来代替原有网格，并在标准网格和原有网格之间建立映射关系，新的网格称为等参单元。

高斯积分：在使用等参单元后，会有很多定积分需要求解，通常方法是在积分区域内按照一定规则选出一些点，称为积分点，算出被积函数在积分点的值，然后再乘以加权系数作为近似积分值。高斯积分是精度最高的一种方法。这也是为什么很多有限元软件（比如 ABAQUS），最终结果并不在顶点上，而是在积分点上，需要注意转换。

--显示算法/隐式算法/稳态/瞬态/动力--

显示算法/隐式算法在比较结构动力分析方法中比较常见，显示算法中的节点状态主要由前两部决定，不要求解整体刚度矩阵，适合大规模并行计算，ABAQUS Explicit和LS-DYNA主要使用显式方法。

通常说的稳态指没有时间项的计算，物理场处于平衡不变化的状态。

瞬态/动力 中有时间项，即物理量随时间变化而变化，温度热传导，冲击荷载，结构振动，电磁时域都是和时间相关。常用求解方法是将时间离散，求解每一个时间点的状态。时间的离散步长需要根据实际情况计算，时间步长过大，求解不精确，过小，浪费计算资源。

几个多物理场例子

--被风吹塌的桥（流体，结构，振动）--

塔科马海峡吊桥（英语：Tacoma Narrows Bridge）是位于美国华盛顿州塔科马的两条悬索桥，横跨塔科马海峡，第一座桥于1940年建成，但不到五个月便倒塌，公认的直接原因是桥梁在受到强风的吹袭下引起卡门涡街，使桥身摆动，当卡门涡街的振动频率和吊桥自身的固有频率相同时，引起吊桥剧烈共振而崩塌，这次事件成为研究空气动力学卡门涡街引起建筑物共振破坏力的活教材，也被记载为20世纪最严重的工程设计错误之一。而事故的根本原因是节省成本，减少桥面厚度所引起的刚度降低，虽然桥梁整体强度毫无问题，但是刚度降低破坏了整体稳定，无形中增加了风载造成的冲击，为事故埋下了隐患；事实上即使桥梁不被吹垮，长期经受高强度的冲击荷载，其使用寿命也会大大降低！卡门涡街是流体力学中重要的现象，在自然界中常可遇到，在一定条件下的定常流体绕过某些物体时，物体两侧会周期性地脱落出旋转方向相反、排列规则的双列线涡，经过非线性作用后，形成卡门涡街。如水流过桥墩，风吹过高塔、烟囱、电线等都会形成卡门涡街。常用的CFD软件中都会用卡门涡街作为基准测试建立标准算例。

--NVH（声音，振动，声振舒适度）--

NVH是衡量汽车舒适度的一个重要指标，在中国，随着家庭汽车越来越多，NVH（Noise、Vibration、Harshness）在汽车研发中成为热门研究内容，汽车行业的火爆可是说是NVH发展的重要推手。有统计资料显示，整车约有1/3的故障问题是和车辆的NVH问题有关系，而各大公司有近20%的研发费用消耗在解决车辆的NVH问题上。

-----任正非采访（热分析）-----

华为老总任正非在一次接受采访时表示**散热和发热机理可能是电子技术最核心的竞争力**“芯片越来越小，元器件越来越集中，硬件工程、电子工艺最大的问题就是散热。有专家表示，未来50%的能源将消耗在芯片上，散热和发热机理也可能是电子技术最核心的竞争力”，未来电子产品不仅要进行电磁兼容，电源完整，信号完整，机械强度等仿真，散热分析更会成为重中之重。

-----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！

-----设计和优化-----

关于设计和优化，之前谈过很多。在工程项目中，有限元方法仿真只是其中一环，即对模型进行计算得到仿真结果。按照ANSYS的说法，多物理场仿真仅仅停留在第二个层次（第一层次是单物理场仿真）。ANSYS现在关注更高第三层次的设计驱动仿真和第四层次的业务驱动仿真。简单讲设计驱动仿真让仿真来决定设计，而业务驱动仿真则是让仿真决定业务，之前做过详细介绍，可以查看历史文章。

其实按照目前的水平，在世界范围内，能做到第三层次即仿真驱动设计，或者优化设计，已经是把仿真做的相当不错了！

--不同物理场的有限元方法一样吗--

理论基础一样，但实现方法不一样

有限元的核心之一在于对形函数的选择，也就是形函数要能准确表达PDE以及边界条件。以电磁单元为例，早期使用和结构相同的三角面片单元（自由度在定点），但是在计算边界的时候发现电磁场不满足连续条件，后来发明了矢量单元（自由度在边上）来解决这个问题，形函数和结构分析使用的形函数不同。

再比如热分析需要对环境温度进行定义，电磁分析需要设定吸收边界（物体到吸收边界区域也需要划分网格），而结构中则不需要。结构分析中常见的沙漏，自锁，电磁分析中因为自适应网加密数目过少造成的伪解，CFD边界层网格没有加密造成的求解误差过大，都是由各自不同物理场的特点决定的。

--多物理场软件--

COMSOL是名至实归的多物理场软件，在前处理方面提供了多种物理场建模功能，同时可以自定义偏微分方程，求解偏微分方程，这是其它软件无法比拟的。COMSOL作为学习，科研绝对是利器！所以在高校推广的特别好，在研究所研究院也有相当的市场！But，在工程领域，COMSOL拓展的比较艰难。这也是本人经常说到的一个问题，软件只是工具，一旦涉及到工程领域，软件必须要能解决实际问题。从工程角度看，COMSOL就像是一个大杂烩，似乎什么都能做，但什么都做不精通！涉及电磁电子有专业的EDA工具，做结构有老牌的ANSYS，ABAQUS，空气动力飞行器有老牌的NASTRAN，流体Fluent，Star-CD，材料化学有VASP，MS。

综合来看COMSOL和MATLAB属于同一类型的产品，着重在于研究解决某个点上的问题，而非解决综合工程问题

-----ANSYS-----

作为老牌商业仿真工具，ANSYS一直讲多物理场仿真作为支撑和核心技术。尤其以Workbench作为多物理场仿真的平台。每完成一次仿真软件收购之后，就会将其融入到

多物理场平台中。

功能较全能提供多物理场耦合仿真的软件有ADINA, Altair Multiphysics, STAR-CCM+, 其它号称能解决多物理场问题的工具其实都算不上真正意义上的多物理场工具, 比如MSC的SimXpert, 达索的SIMULA等。

后记

在仿真软件 (CAD/CAE/EDA/CFD) 开发工作中, 发现很多软件研发测试人员没有相关背景, 缺乏对有限元基础知识的了解, 所以一直以来想写一篇不需要用积分求导公式的有限元科普文, 希望该篇文章能有所帮助。

大规模线性方程组解法简介

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个



前面介绍过主要的线性方程组求解库，参考附录。求解大规模线性方程组是仿真软件求解器的底层技术，求解器时间基本都消耗在方程组求解上。线性方程组的解法比较成熟，方法也有很多，而且不同的方法对应不同类型方程组，所以在方法选择上实际很讲究。

商业软件通常将方法封装起来，用户包括开发人员都接触不到线性方程组求解方法。商业软件内部一般会根据求解规模，求解类型等选择合适的线性方程组求解方法。

有些商业软件开放了部分接口供用户选择；开源软件比如OpenFOAM，以及使用开源软件的平台Simscale等提供了许多选项供用户选择。

本文简单介绍下线性方程组的常用解法。通常将线性方程组表示为：

$$A \cdot x = b$$

A为已知 $N \times N$ 的矩阵，通常称为刚度矩阵（刚度是力学中的概念，电磁，热等也习惯性这么称呼），b为已知向量，x为待求向量。解线性方程组的操作基本围绕矩阵A展开。

首先介绍一些相关术语：

1. 矩阵条件数

条件数是一个表征矩阵稳定特性的标志，条件数越大，说明矩阵越不稳定，即当矩阵中数据出现微小变化时，x结果变化非常大。Matlab中可直接使用命令`cond(A)`查看矩阵条件数。

2. 满秩矩阵

用初等行变换将矩阵A化为阶梯形矩阵，矩阵中非零行的个数就定义为这个矩阵的秩，记做 $r(A)$

即如果矩阵A为 $N \times N$ ， $r(A) = N$ ，则矩阵A为满秩矩阵

在边界元，矩量法等计算方法中，最终形成的A为满秩矩阵。

3.稀疏矩阵

矩阵中绝大部分元素都是0

4.对称矩阵

矩阵的上三角和下三角关于对角线对称

5.求解线性方程组直接法

先求出矩阵A的逆矩阵，再乘以向量b

6.求解线性方程组迭代法

简单讲，就是给出一个初始解 x' ，带入原方程中，每次评价差距逐步修正，直到最终 $A*x'-b$ 接近0为止。

7. OOC (out of core)

对于大型方程组，内存通常无法装下，在求解过程中需要将部分数据写到硬盘上，再次使用的时候再读回来。

常用直接法：

直接法的本质是要计算出A的逆矩阵，通常在求解小规模和特征值问题是可以考虑使用直接法。

1. 高斯消去法/Doolittle /三角分解法/追赶法

这是最基本的方法，时间复杂度和空间复杂度都是N的三次方，软件一般都不会使用。

2. 矩阵分解相关

2.1.LU分解法

LU分解就是将矩阵分解成单位下三角矩阵L和上三角矩阵U，本质上仍然属于高斯消去法

2.2. Cholesky分解

Cholesky 分解是把一个对称正定的矩阵表示成一个下三角矩阵L和其转置的乘积的分解。如果矩阵是正定的，使用 Cholesky分解会比LU分解更加高效。

2.3. LDLT分解法

Cholesky 分解法的改进

2.4.QR分解

QR分解是把矩阵分解成一个正交矩阵与一个上三角矩阵的积。

2.5. Schur分解

2.6. SVD/GSVD

奇异值分解/一般奇异值分解

在求解大规模线性方程组中，一般不会使用直接法求解，但在使用迭代法过程中需要使用直接法中的方法加工数据。

间接法的本质是迭代，不同方法的区别在于如何选取初始值以及迭代方法。

1. 牛顿迭代，Jacobi(雅克比)迭代，Gauss-Seidel迭代

线性迭代方法，一般情况下收敛太慢，大规模方程组不推荐使用。

2. JOR -- Jacobi Over Relaxation

Jacobi 方法加入松弛因子

3. Lanczos方法

适用于稀疏矩阵特征值问题

4. 共轭梯度方法以及改进方法

共轭梯度法

Conjugate gradient method--CG

双共轭梯度法

Bi-conjugate gradient method--BCG

稳定双共轭梯度法

Bi-conjugate gradient stabilized method--GCGS

预条件共轭梯度法

Preconditioned gradient stabilized method--PCG

相比牛顿和Jacobi，通过优化使其共轭的求解向量和方向，加速了求解性能。

共轭梯度法的思想就是找到N个两两共轭的方向，每次沿着一个方向优化得到该方向上的极小值，后面再沿其它方向求极小值的时候，不会影响前面已经得到的沿哪些方向上的极小值，所以理论上对n个方向都求出极小值就得到了N维问题的极小值。

5. GMRAS广义最小残量

(Generalized Minimal Residual Algorithm)

非对称系统的线性方程组的数值解迭代法，该方法与最小残量的 Krylov 子空间中向量来逼近解，是求解大规模线性方程组的常用算法之一，具有收敛速度快、稳定性好等优点。

改进的GMRAS：

SGMRAS Simpler GMRAS

PGMRAS Preconditioned GMRAS

6. 快速多级 (Fast Multi pole Method)

当矩阵为满秩矩阵时，传统计算方法资源需求和求解规模呈指数级上升，大规模的系统求解需要使用快速多级方法。本技术博客和公众号中有详细介绍。

改进方法：多层快速多级

7. Successive Over Relaxation(SOR)连续松弛

连续过度松弛（SOR）方法是高斯-赛德尔（Gauss-Seidel）方法的一种变体，用于求解线性方程组，从而可以更快地收敛。任何缓慢收敛的迭代过程都可以使用类似的方法。

改进的SOR:

Accelerated Over relaxation (AOR)过松弛

Preconditioned AOR -- PAOR 预条件过松弛

Quasi AOR -- QAOR准过松弛

8. Krylov子空间迭代法

将矩阵A分解成多个子矩阵，并用一系列线性表达式组合表示。将整个系统降维，利于并行计算，是大规模线性方程组有效的一种解法。其中涉及到了Lanczos方法和Arnoldi方法。

按照矩阵A的特点，我们可以做如下分类：

1. 是否是满秩矩阵；
2. 是否是稀疏矩阵；
3. 是否是对称矩阵；
4. 是否是病态矩阵；
5. 是否是正定矩阵
6. 矩阵规模（矩阵中N的大小）

在选择求解方法的时候，需要考虑到方程组矩阵以上特点。根据作者的经验，方程组的求解性能高度依赖矩阵特征，矩阵规模和硬件。

(非) 线性方程组求解库大全

快速多级算法简介

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

一篇文章入门求解器模型降阶方法

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个



名词介绍

POD-- Proper orthogonal decomposition 本征正交分解

PCA-- Principal Component Analysis 主成分分析

Krylov subspace --Krylove 子空间

CGM -- Conjugate Gradient Method 共轭梯度

PCGM --Precondition CGM 预条件共轭梯度

AWE --Asymptotic Waveform Evaluation 渐进波形估计

MINRES --minimal residual 最小余量

GMRES --Generalized Minimum Residual 广义最小余量

QMR -- Quasi Minimal Residual 准最小余量

IDR -- Induced Dimension Reduction

BiCGSTAB -- Biconjugate Gradient Stabilized 稳定双共轭梯度

PRIMA --Passive Reduced-order interconnect macromodeling algorithm

SPRIM -- Structure-preserving reduced-order interconnect macromodeling

在之前文章中提到了工程领域的模型数据和支付宝数据最大的区别就是：前者模型是一个整体，不可分割，这就导致模型计算难以分解，模型越大，计算时间越长。在硬件多核，网络分布式计算，GPU计算，软件OpenMP，OpenMPI等并行计算技术越来越成熟之后，这种矛盾就越来越突出。

模型降阶，简单讲就是在保证设定精度的前提下，简化模型，减少计算量。这种简单的思想很早就出现在工程领域的文献中，但其作为具有理论以及的较为系统的数学方法也就是近一二十年的事。

在多物理场仿真领域，最后都需要求解大规模线性方程组，自由度少则几十万，多则上千万，利用常规求解方法效率低下，而模型降阶的目的正是降低此类问题的计算复杂性，例如，在大规模动力系统和控制系统的仿真中，通过减少模型的关联状态空间尺寸或自由度，可以计算出与原始模型的近似值。再比如，在进行精确模型分析之前，用户只想对模型部分进行建模仿真，或只想了解大概数值范围和趋势，都可以通过模型降阶方法来减少模型计算时间。

模型降解的通常要求是：

1. 计算结果与原始模型相比，误差小或者在设定介绍范围内；
2. 保留原始模型的特点和特征；
3. 降阶后的模型计算要保证稳定性和有效性。

可以看出，广义上的模型降阶泛指简化减少模型计算。本文要讨论重点是求解大规模线性方程组的降阶方法。

在模型降阶中，最基础和最常用的就是Krylov子空间方法。Krylov全名Aleksey Nikolaevich Krylov，1863年出生于俄国，俄罗斯海军工程师，一生著有300多篇论文和书籍，涵盖了流体力学，造船，天文，测量，数学等领域。1931年Krylov提出了Krylov子空间和方法的论文，Krylov方法通常采用所构造的标准列正交向量基底对系统进行模型降阶，使得降阶系统的传递函数对于原始线性系统的传递函数，在其指定的区域内有很好的近似。Kylov子空间方法在数学理论上相当完善，其主要优点是算法稳定，能保持原始传递函数的特点，有时还能保持系统的基本特性。

对于线性方程组 $A \cdot x = B$

其中A为 $N \times N$ 的矩阵，N为大数

常规方法为 $x = A^{-1} \cdot B$

很显然，直接求逆矩阵比较困难，尤其A是稀疏矩阵时

我们将 $A^{-1} \cdot B = m_1 \cdot B + m_2 \cdot A \cdot B + m_3 \cdot A \cdot A \cdot B$

其中 m_1, m_2, m_3 为未知的标量系数，可以看出A，B都是已知量，只要求解 m_1, m_2, m_3 就可以了

Krylov已经证明，当m个数=N时，其结果为精确解。

实际工程中，可以根据具体情况选择m的数量。

一些常用的Krylov子空间方法包括：

Arnoldi降阶算法

共轭梯度系列相关算法

最小余量系列相关算法

C正交Arnoldi算法

块Arnoldi算法

对称和非对称Lanczos降阶算法

多重Krylove子空间降阶

在算法工具方面，Matlab 仍然是首选。推荐两个基于Matlab的MOR算法工具：

MORLAB

<https://www.mpi-magdeburg.mpg.de/projects/morlab>

RBmatalb

<https://www.morepas.org/software/>

一篇文章入门大规模线性方程组求解

一篇文章入门时域有限差分方法 (FDTD)

原创 www.cae-sim.com 多物理场仿真技术



时域有限差分 (Finite Difference Time Domain简称FDTD) 由美籍华人Yee于1966年提出，是求解Maxwell方程的一种方法，其核心思想是将求解空间离散成笛卡尔坐标上的长方体网状结构，其中每个点赋值电场和磁场，随着时间变化每个点以蛙跳形式交替更新电场和磁场，从本质上讲，FDTD是对电磁场问题的最原始最完善的模拟，具有非常广泛的适用性。

不同于有限元方法，该方法只需要对求解空间划分结构化网格，不需要对对象划分网格，不需要求解线性方程组，不需要复杂导出方程，适合计算程序实现，非常容易应用并行计算。但由于FDTD计算量巨大，导致其方法提出后应用发展缓慢，最近一二十年，由于计算机硬件快速发展，尤其是多核多线程，GPU计算，分布式计算应用，使得FDTD可用于计算量大的实际工程。Navida 的CUDA计算案例里甚至有专门针对FDTD的例子。

之前简单介绍了计算电磁学，这里首先介绍一下数学基础理论知识。

基础介绍

在积分微分公式中我们经常看到倒三角符号 ∇ ，叫做哈密顿算子，这是一个常用算子，读作del，表示在空间xyz各方向上的全微分，另一个是 $\nabla \cdot \nabla = \nabla^2 = \Delta$ ， Δ 称为拉普拉斯算子。

梯度: ∇u

散度: $\nabla \cdot \mathbf{A}$

旋度: $\nabla \times \mathbf{A}$

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

电磁场本构方程为麦克斯韦方程

名称	微分形式	积分形式
高斯定律	$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0}$	$\oiint_S \mathbf{D} \cdot d\mathbf{s} = \frac{Q}{\epsilon_0}$
高斯磁定律	$\nabla \cdot \mathbf{B} = 0$	$\oiint_S \mathbf{B} \cdot d\mathbf{s} = 0$
法拉第感应定律	$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$	$\oint_L \mathbf{E} \cdot d\mathbf{l} = -\frac{d\Phi_B}{dt}$
麦克斯韦-安培定律	$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \mu_0 \epsilon_0 \frac{\partial \mathbf{D}}{\partial t}$	$\oint_L \mathbf{H} \cdot d\mathbf{l} = \mu_0 I + \mu_0 \epsilon_0 \frac{d\Phi_E}{dt}$

多物理场仿真技术

以上四个方程构成了麦克斯韦方程组。

其中 $\mathbf{E}$ 为电场强度， $\mathbf{H}$ 为磁感应强度， $\mathbf{B}$ 为磁通量密度， $\mathbf{J}$ 为电流密度。

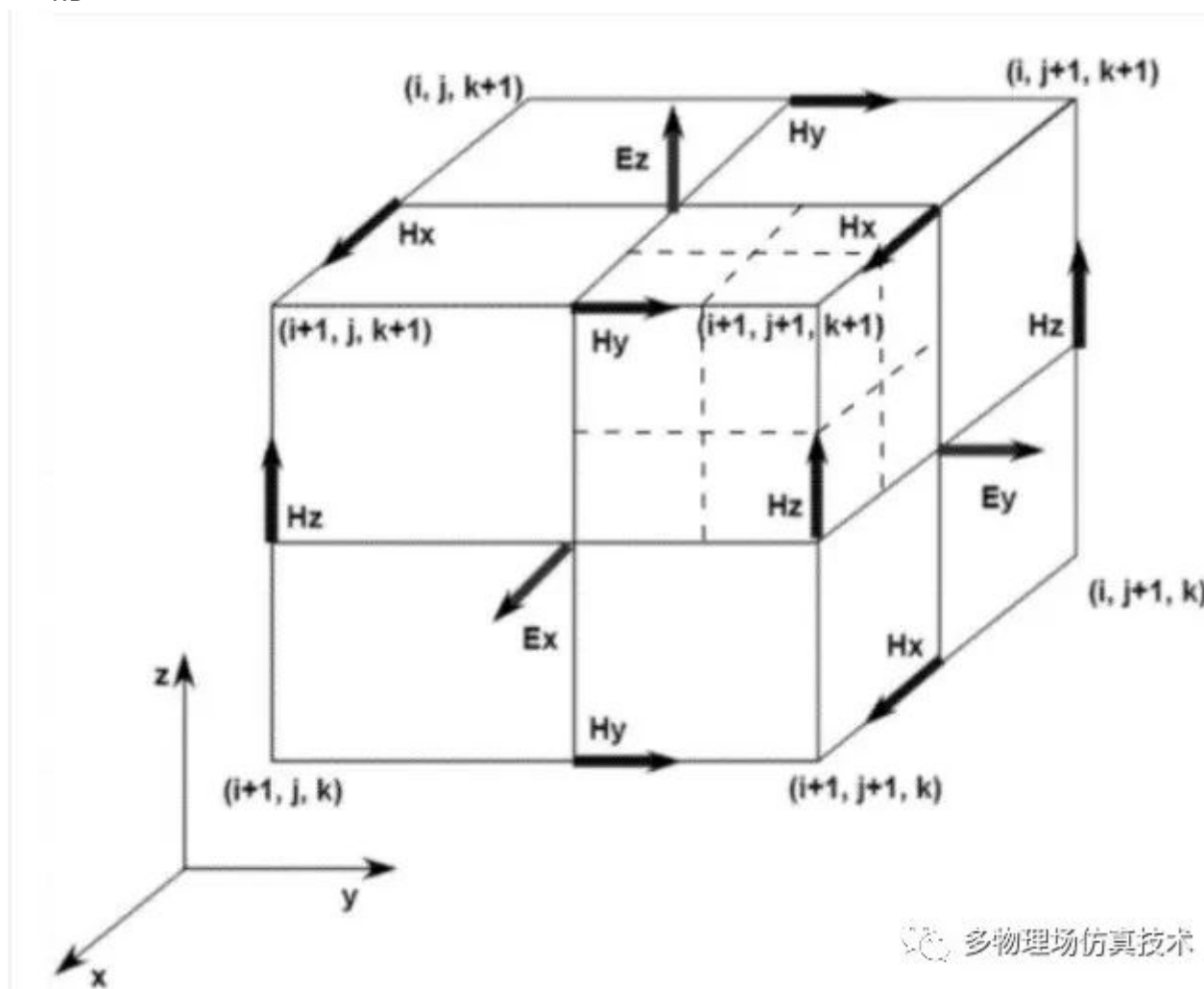
通常说的磁生电，电生磁，麦克斯韦方程准确描述为：

- 1.电场的散度跟这点的电荷密度成正比；
- 2.感生电场的旋度等于磁感应强度的变化率；
- 3.磁场的散度处处为0；
- 4.感生磁场的旋度等于电流密度和电场强度变化率之和

FDTD的Cell（元胞）结构

FDTD的核心思想是将求解空间离散成笛卡尔坐标上的长方体网状结构，下图（图3）是网状结构中的一个单元，一个非常典型的三维Yee元胞结构，又叫Yee网格。

FDTD的Cell



1.靠近远点的坐标为 i, j, k ，对角坐标则为 $i+1, j+1, k+1$ 。

2.在这个长方体上，我们将其分为八个等分长方体，长方体的每条边的中点存放磁场分量，面上的中点存放电场分量。当到下一个长方体时，则反过来：边的中点存放电场分量，面的中点存放磁场分量，以此类推。由上图可以看到：每一个电场矢量都被四个磁场矢量环绕；每个磁场矢量被四个电场所环绕，类似磁场的旋度。

3.FDTD算法在离散的时间瞬间取样和计算场值，电场和磁场取样计算并不在同一时刻。对时间步长 t ，电场取样时间为 $0,1t,2t,3t$ ，磁场取样时刻为 $0.5t,1.5t,2.5t,3.5t$ ，两者之间相差半个时间步长。

4.对于求解区域内不同介质的对象（介电常数，磁导率，电导率等），也分布在FDTD的整个网格节点上。

从以上描述可知，FDTD是一种在时间上迭代的差分方法，在给定的时间上，更新网格节点上电场，磁场，不需要对对象划分网格，也不需要求解线性方程组。

几个核心问题：

1. 如何确定网格大小；
2. 如何确定迭代时间步长；
3. 边界条件如何设置；
4. 如何设置激励

1. 如何确定网格大小；

FDTD的网格，我们通常用Grid，表示是对空间的离散，而非对物体本身的离散，对空间划分网格比划分物体几何要简单，这是差分法和有限元法最显著的一个区别。FDTD差分格式所能模拟的最小尺寸为一个网格，对于小于一个网格的尺寸，需要近似成一个网格，但是存在的问题是在细小几何，比如圆孔，曲面曲线，缝隙等结构物体时，计算误差会比较大，如果全体加密网格，网格单元数和计算量也会快速增加。通常的办法是在这些细小几何的地方，采用局部加密的方式，如图1所示。

通常情况下，FDTD离散网格的大小和波长有关，应满足条件网格尺寸小于计算区域内最小波长的0.1倍。网格加密尺寸需要更加几何和波长动态确定。

2. 如何确定迭代时间步长；

利用波矢量和频率的色散关系，可以求出空间和时间间隔之间应该满足的关系，该关系称为Courant稳定性条件。对于一维，时间间隔必须小于等于波以光速通过一个Yee网格所需的时间，二维和三维，时间间隔必须小于等于波以光速通过一个Yee网格对角线的 $1/3$ (三维)或者 $1/2$ (二维)所需要的时间。而对于数值色散，时间间隔要取得更小。

3. 边界条件如何设置

许多电磁场边值问题都包含了无限大的空间范围，例如天线设计，电磁散射，电磁传播等。求解的一大困难是无法存储无限空间的网格数据，需要采取一定措施将无限大空间问题转成有限空间问题，通常的做法是设置人工的边界，将无限大区域截断，构造出一个有限的空间区域，这个区域包含了要求解的对象，激励源等，同时需要在人工边界上接入一定的边界条件。使得电磁波在这些边界上与无限大空间一样无条件地向外传播，这些边界

条件称为吸收边界条件。解决问题的关键是获得合适的吸收边界条件，其中早期比较典型的是**Mur吸收边界条件**。

完全匹配层，通过在截断边界处附加匹配的有耗材料来实现。在实际三维电磁场中，电磁波会以不同角度入射到截断边界上，要使所有入射波都能够无反射地进入到有耗媒质中比较困难。完全匹配层吸收边界通过场分裂法和单轴媒质两种方法实现。

4. 激励源

在电磁仿真软件中，需要设置激励源，也就是通常所说的类似Port端口的概念。FDTD常用的激励源有：时谐场源；脉冲源，包括常见的高斯脉冲，升余弦脉冲，截断三正/余弦脉冲，调制高斯脉冲，双指数脉冲。

利用FDTD计算还会涉及到近远场转化，色散模型，并行计算，S参数计算等一系列问题，后面介绍天线，矩量法等再介绍相关话题。

FDTD软件

由于FDTD原理相对简单，开源和商用的软件都比较多，开源的FDTD++，Meep，OpenEMS，商用的GEMS，xFDTD，FDTDSolution，国内的有EastFDTD。

最后用一个简单的C++ 程序演示了二维FDTD求解流程，包含网格生成，电磁场更新，设置吸收边界，激励等基本元素。

需要可到 [多物理场仿真技术 技术博客](#) 下载。

文章已于修改

一篇文章了解云端仿真

cae-sim.com 多物理场仿真技术



亚特兰大2020年12月24日，霍尼韦尔（纽交所代码：HON）近日宣布以13亿美元的现金交易收购New Mountain Capital私募股权公司持有的Sparta Systems公司（以下简称Sparta）。Sparta是生命科学领域领先的企业质量管理软件（QMS）（包括下一代SaaS平台）供应商。通过此次收购，霍尼韦尔进一步巩固了在工业自动化、数字化转型解决方案和企业绩效管理软件方面的市场领先地位。

Sparta公司总部位于美国新泽西州汉密尔顿，全球员工约250名。Sparta为超过400家客户提供服务，其中包括42家全球50强制药公司和33家全球50强医疗器械公司

Sparta人工智能化的软件即服务（SaaS）质量管理软件（QMS）将与行业领先的企业绩效解决方案Honeywell Forge以及Experion 过程知识系统（PKS）相结合，从而为美国、欧洲和亚洲的生命科学领域的客户创造更多价值。

这是近年来一起较大的关于Saas收购事件。

2019年10月23日，PTC宣布以约4.7亿美元的价格收购SaaS CAD厂商Onshape。Onshape成立于2012年，总部位于马萨诸塞州剑桥市，是业内第一个软件即服务（SaaS）产品开发平台的创造者，该平台将CAD与数据管理工具结合在一起。Onshape 2015年3月11日开放公测，同时宣布获得了由North Bridge 和NEA 领投的6400 万美元的种子轮融资。2016年1月Onshape正式进行商业发布。目前，Onshape已经在全球超过150个国家应用了Onshape CAD，访问者遍及全球各地。在发展过程中Onshape获得过超过1.69亿美元的投资，拥有5000多名全球用户。

可见服务上云，软件上云不仅在IT领域比较活跃，在工业软件领域也是热点。

2020年1月31日，德国云端仿真公司SimScale宣布完成了2700万欧元（约2亿人民币）的C轮融资，此轮融资由总部位于纽约的Insight Partners领投。此次融资事件表明了资本对云端仿真前景的乐观态度。

SimScale是一家成立于2012年的SaaS (Service-as-a-software) 公司，主打多物理场仿真，可以求解结构，热，流体，增材等仿真项目，提供基于web的云端仿真平台，其前处理，后处理都在web的浏览器上进行，求解调用云端的开源求解器。用户可以将所有数据保存在云服务器上，只要能访问网络，就可以访问自己在SimScale上的项目。

相比于大厂的仿真软件，SimScale提供了灵活的多的解决方案，如果用户只是需要跑一些快速的仿真，比如耗时1分钟的结构分析，或者1000次重复的热分析，使用Simscale的成本要低得多，几乎可以忽略不计。SimScale在仿真方面给仿真工程师提供给了更多的选择，打破了只能高价购买license的传统。

总体来看，云端仿真的优点：

1. 增加用户协作；
2. 避免软件安装，更新成本；
3. 集中管理数据安全；
4. 任何地点可以办公；
5. 降低软件使用成本；
6. 便于分布式存储备份和版本管理

也是因为类似SimScale公司的出现，一些大厂在软件使用方面也采用了云端license的策略，最典型的就Autodesk的Fusion360。Fusion360将Autodesk建模仿真产品融合在一个云端平台，按照使用付费，可以说是云端化比较成功的应用，但是Autodesk本身的仿真研发能力较弱，在壁垒很高的仿真领域对其它大厂难以构成威胁。

这种基于web浏览器的前后处理云端厂商，除了www.simscale.com，还有
www.onscale.com
www.designworld.com
www.caeplex.com
www.ceetron.com
以及国内的www.simright.com

除了基于浏览器的web仿真厂商，另一种SaaS云端仿真更容易理解，即使用HPC降低成本。很多仿真大厂，亚马逊，微软，华为，阿里云，TotalCAE，UberCloud等等，将桌面仿真软件部署在云端，用户提交工程作业，由HPC计算，使用按照节点使用时间收费。

对于厂商而言，这种方式可以最大程度利用IT硬件，积累用户资源，加速产品迭代，降低了用户使用软件门槛，利于软件的推广和普及，对于用户而言，既降低了软件使用成本，又提高了仿真效率。

Web云端仿真对传统大厂有冲击吗？

我觉得影响有，但是非常非常有限，主要原因有几点：

1. SimScale计算引擎采用了开源软件，这就约束了其求解范围，只能解决仿真领域的一些基本问题，动不了大厂的根基；
2. 在前处理领域，尤其是大模型，桌面版的就已经非常耗时，在web上性能就更差，所以SimScale的对象还是中小企业，学校，科研机构等；
3. 安全问题。对于使用仿真工具的企业而言，数据就是核心资产，数据丢失对公司而言就是灾难。

就在2020年2月，中国香港上市的SaaS服务厂商微盟发生运营人员“删库跑路”事件，导致其几百万用户无法访问数据。受删库事件影响，微盟股价24日出现下跌，市值一日之内蒸发约12.53亿港元。微盟事件再次给SaaS厂商的安全问题敲响了警钟。

对于仿真软件，网格，求解器，后处理，HPC等底层技术没有太多改变，只是将PC端上层的诸如UI以及对底层调用等换成了B/S模式，传统软件稍作改动也能达到相同的效果，比如BIM软件bentley，在整体协作建模功能上已经相当成熟

长远来看，云端仿真应该会成为一种趋势。即使在公有云无法普及的场景，私有云也会快速发展，在企业内部搭建云平台，使用云端软件，会成为许多使用仿真软件公司的刚需。

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



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；

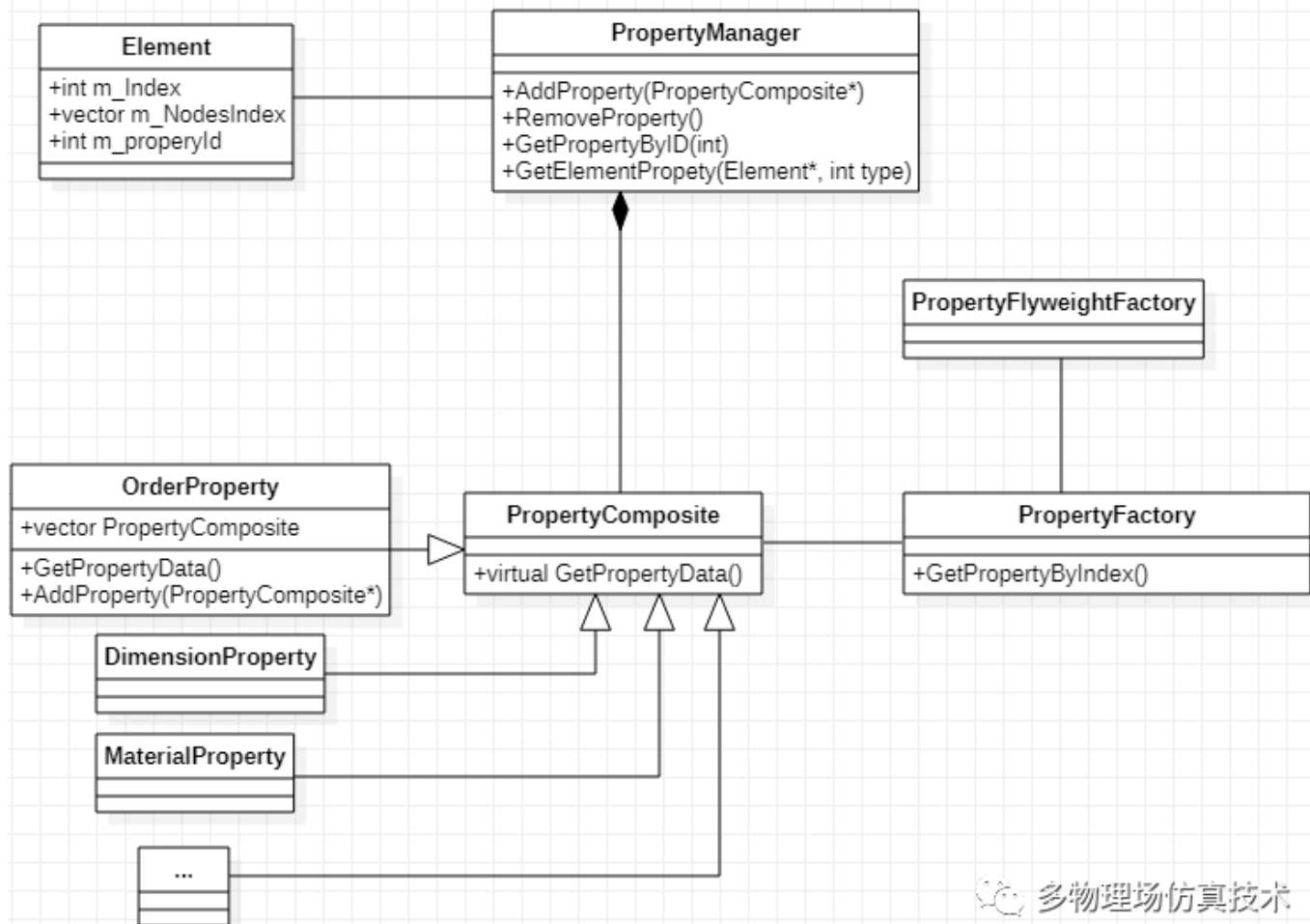
属性模式需要使用以下基本设计模式：

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。

一篇文章入门无网格方法(1)

原创 邓子平 多物理场仿真技术



我们知道在有限元，边界元，矩量法等数值计算方法中，都是以网格为基础的。关于网格划分的介绍，参考一篇文章入门网格划分。

考虑到有限差分，FDTD（时域有限差分）等方法也有网格，且这种网格和有限元方法网格有本质的区别。为了严谨和统一，笔者将以下内容作为区分是否有网格的标准：

即网格的拓扑结构是否在计算仿真中得到了使用。

拓扑结构即连接网格的方式，举几个例子：

1. 三角形，三个点，连接网格的方式为点1，点2，点3，这种拓扑结构在有限元仿真中作为形函数使用，因此是有网格结构。
2. 二维的FDTD 元胞网格，每个节点被其它四个节点包围，虽然在显示中每个网格为四边形，但是并没有连接点1,2,3,4的拓扑信息，因此可以认为FDTD是一种无网格方法。
3. 使用扫掠生成的六面体三维梁网格单元。六面体一般直接使用四边形扫掠生成，因此比较规则，虽然网格看起来和三维FDTD一模一样，但是因为每个六面体会有基函数和形函数表达，因此是有网格方法，但是在具体网格类型上属于结构化网格，即非常规则整齐的网格。
4. 一维梁单元，一维的梁单元在几何表达上只有一条直线，两个点，但在计算中需要连接点的直线拓扑关系，因此也是有网格方法。
5. 格子玻尔兹曼方法，该方法主要计算分子之间的关系，不存在分子之间的拓扑关系，因此是无网格方法。
6. 边界元中，需要使用二维的网格计算三维内容，使用一维网格计算二维内容，也是一种有网格方法。
7. 有限体积法中会使用变形的拓扑网格结构，因此是一种有网格方法。

无网格方法的核心在于不使用网格拓扑构造的形函数，多数无网格方法构造的试函数是非线性的逼近或者拟合，不具有插值特性，或者使用粒子系统，因此当施加荷载或者添加边界条件时，需要用新的方法，比如拉格朗日乘子，直接配点，以及罚函数等。

按照计算原理，笔者把无网格方法分为三类：

多物理场仿真技术

第一类，构建新的形函数：

- 1.光滑粒子法(Smooth Particle Hydrodynamics Method, 也简写成SPH)
- 2.移动最小二乘法(Moving Least-Squares Approximation, 简称MLS)
- 3.重构核粒子法(Reproducing Kernel Particle Method, 简称RKPM)
- 4.单位分解(The Partition of unity method)
- 5.径向基函数法(Radial Basis Functions ,简称RBF)
- 6.点插值(Point Interpolation Method, 简称PIM)
- 7.移动Kriging插值法(Moving Kriging Interpolation method)

第二类：使用粒子系统

- 1.格子玻尔兹曼(Lattice Boltzman method, LBM)
- 2.离散元(Discrete Element Method, DEM)

第三类：使用Grid

- 1.有限差分/时域有限差分(FDM/FDTD)

无网格方法优点：不需要划分网格，不存在划分网格失败的情况，只需要节点信息，在计算过程中很容易进行加密和稀疏；而且一般无网格方法具有高阶连续性，且形式灵活，对于高梯度，大变形，奇异性等问题有较好的求解结果。对于粒子系统，可以很好的模拟不连续系统模型。

无网格方法有其固有优点，但缺点也是显而易见的，主要体现在以下两点：

1. 很多无网格方法并没有完善的数学理论基础，比如在收敛性，稳定性以及误差估计方面，还在探索阶段。这就导致工程应用上如果出现问题，很难解决。
- 2.无网格方法在计算效率上没有优势，有限元方法中的形函数取用统一的多项式函数，方便简单，而大部分无网格方法需要针对点取形函数，形函数有可能不同而且复杂，在计算对应矩阵的时候费时费力；再比如三维FDTD，需要在XYZ三个方向内嵌循环，计算量巨大。

无网格方法的其它相关缺点也直接间接和这两项相关。

一篇文章入门计算机辅助优化(Computer Aided Optimization)

原创 邓子平 多物理场仿真技术

收录于话题

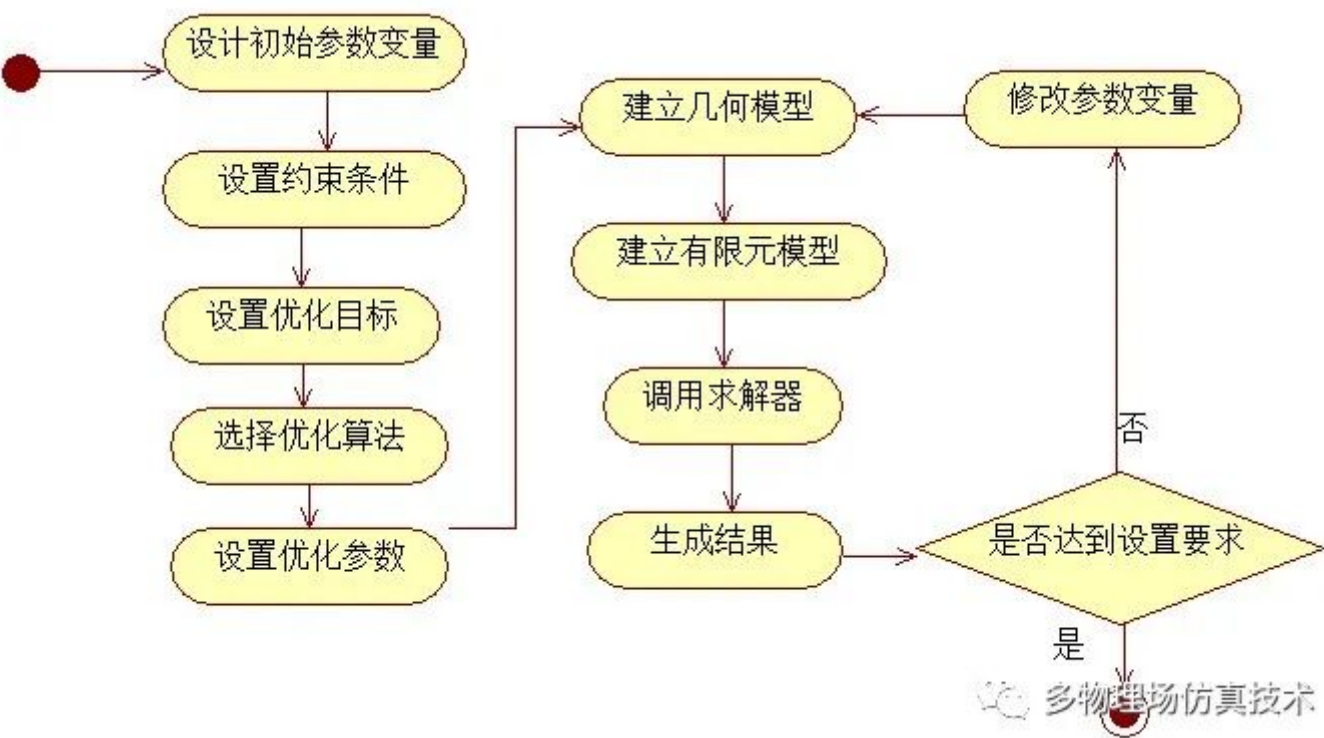
#大咖话题 13 #仿真研发工具 21 #求解器开发 22



Computer Aided Optimization(计算机辅助优化)是计算机辅助工程领域一个重要技术分支，在设计阶段是一个非常有用的方法：设计变量过多时，不可能一个个去试验，借助优化算法，可以大大减少仿真次数。尤其是发展起来的智能算法，如遗传，神经网络，蚁群，粒子等算法在解决多目标优化领域有很多成功应用，非常适合仿真方面的优化。

优化功能是很多CAE软件产品中的一个模块，比如Altair OptiStruct，Ansys DesignExplorer，HFSS Optimetrics，Abaqus ATOM 以及独立的优化软件Isight，Tosca等，Matlab等科学计算软件中也提供了专门的优化工具箱，开源的不计其数了如Tao，Petsc，Alglib，dlib等。

工程上的优化流程可以通俗的用下图来表示：



上图只是做优化一个非常简单的流程，在实际应用中会有很多的工程性问题，展开的话会有很多的子图。其中**修改参数变量**一项 是通过**优化算法**来实现的，也是本文介绍主要介绍的内容。优化算法和CAD， Mesh一样属于基础研究领域。

1. 优化算法

简单讲优化算法分两种，一种是对目标函数进行计算，对象函数表达式已知；另一种是试探法，通过各种搜索算法确定输入参数对结果的影响，来查找优化解。通常第一种算法函数的导数，梯度等可以计算出，迭代速度较快，而第二种方法适合对象函数未知，只知道优化的目标值的情况，但是收敛速度较慢，尤其是在输入参数过多，求解器一次计算时间很长的情况下，实际应用中很难得到有效解。这也是开发求解器的原因之一，知道了求解器的具体实现，优化算法能更有针对性，效率更高。

单纯的凭借算法很难解决实际工程问题，实际应用中往往利用工程和算法，经验DOE等特点选择合适的优化流程，并采用多种优化算法结合，利用等来减少迭代次数。

本文只介绍常用的搜索算法：

局部优化算法：

1. 模式搜索法，又叫Hooke-Jeeves方法
2. Rosenbrock法
3. 单纯形搜索法
4. Powell法

全局优化算法：

5. 遗传算法
6. 模拟退火算法
7. 神经网络算法
8. 蚁群算法
9. 粒子群算法

2. 优化算法与仿真软件集成

以调用求解器 Nastran为例，设计实现优化流程如下：



因为是调用第三方求解器，实现中要注意几个问题：

1. 模板文件通常在仿真软件中生成，然后通过一定方法将参数识别提取出来，然后再生成模板文件。
2. 第三方求解器要支持命令行调用（大部分软件都支持）。
3. 进程同步，确保每一步调用的前一步都已经完成。
4. 大部分软件仿真结果都以二进制格式给出，需要解析结果文件。
5. 使用文本文件交换数据。

3. 例子演示

举最简单的例子，一根长为4m的绳子，求围绕的长方形面积最大，长和宽都为1时，面积最大。

该优化问题可以用下面表达式描述

$$A(\max) = x * (2 - x) \quad 0$$

文件列表：

1. InputParameter.txt
2. Template.txt
3. Input.txt
4. CalculaFence.exe
5. OptController.exe
6. ResultExtract.exe
7. Output.txt

1. 为参数文件，长和宽；
2. 模板文件；
3. 输入文件，由文件1和2中的数据生成；
4. 计算程序，读入输入文件中的数据，进行计算 即长*宽，完成后将计算结果写入结果文件7 Output.txt；

5. 控制整个计算流程，调用每一步的exe和优化算法，保证进程的同步；
 6. 结果解析器，读入结果文件，提取优化算法需要的参数；
- 对于这类简单问题，各种优化算法都能得到 $1e-5$ 精度以内的解。

本人用过Tosca 和 ISight来做研发，用来找参数的最好设置（给客户把最优数值找好，不让用户输入该参数），说实话功能很强大，当然也不是像软件所吹嘘的什么优化问题都能解决，当遇到超大规模或者设计变量过多的情况时，这些软件也无能为力。

通常优化算法参数很多，但不外乎 计算范围，收敛准则，初始值设置等
要用好优化算法，个人认为做到以下几点：

1. 了解每种算法的优缺点，每种算法都是其使用的范围，而且同一种算法，相同case，优化参数设置不同，结果也会不同。
2. 经验相当重要。在做大Case计算前，可建立小的case进行快速计算。比如可以减少网格数目加快计算，虽然可能计算不准确，但与正确结果的趋势是一样的。
3. 通过适当的人工计算，优化输入参数的数目和范围。减少不必要的计算。
4. 使用源码调用的方式减少文件读写的开销。
5. 使用DOE方法

关于优化算法的入手，个人觉得学习Matlab是个不错的选择，入门简单，可以很快掌握常用优化算法的思想。

4. AI和优化算法

名词解释：

AI - Artificial Intelligence 人工智能

CAE - Computer Aided Engineering 计算辅助工程

Tensor - 张量

GPU - Graphic Processing Unit 图形处理单元

CPU - Central Processing Unit 中央处理器

4.1. AI简介

1.1. AI概念

AI (Artificial Intelligence) 人工智能，它是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。人工智能是计算机科学的一个分支，它企图了解智能的实质，并生产出一种新的能以人类智能相似的方式作出反应的智能机器，该领域的研究包括机器人、语言识别、图像识别、自然语言处理和专家系统等。**简单讲人工智能就是用计算机模拟人类的行为。**

人工智能发展水平：第一个阶段是计算智能，能存会算，比如我们现在使用的个人计算机；第二个阶段是认知智能，能说会听、能看会认；第三个阶段也是目前的最高阶段，是感知智能，它要求机器或系统能理解和思考。第三阶段在我们这代基本没有出现的可能性，即使出现了也不是什么坏事，毕竟人类也不傻，氢弹出现这么多年，按钮还是被牢牢控制在理智的手指下。我们主要讨论的是第二阶段。

1.2. AI的三大要素：

算法，硬件，数据。

算法是核心，硬件和数据是基础

算法：

下图引用自

<http://www.cnblogs.com/doit8791/p/6058999.html>

机器学习算法一览

think big data

回归

普通最小二乘回归 (OLSR)
线性回归
对数几率回归
逐步回归
多元自适应回归样条法(MARS)
局部加权平滑估计(LOESS)
jackknife回归

正则化

岭回归
最小绝对收缩选择算子(LASSO)
弹性网络
最小角回归

基于实例的方法

也叫 case-based, memory-based

k最近邻(kNN)
学习矢量化(LVQ)
自组织映射(SOM)
局部加权学习(LWL)

降维方法

主成分分析(PCA)
主成分回归(PCR)
偏最小二乘回归
Sammon Mapping
多维尺度(MDS)
投影寻踪
判别分析(LDA,MDA,QDA,FDA)

深度学习

深度玻尔兹曼机(DBM)
深度信念网络(DBN)
卷积神经网络(CNN)
堆栈式自动编码器

贝叶斯

朴素贝叶斯
高斯朴素贝叶斯
多项式朴素贝叶斯
平均单核分类器 (AODE)
贝叶斯信念网络(BBN)
贝叶斯网络(BN)
隐马尔科夫模型
条件随机场(CRFs)

决策树

分类回归树(CART)
迭代二叉树3代 (ID3)
C4.5 和C5.0(同一方法的两个版本)
卡方自动交互检测(CHAD)
单层决策树
M5 模型
随机森林
条件决策树

聚类方法

单链聚类
k 均值算法
k 中值算法
期望最大值算法(EM)
层次聚类
模糊聚类
DBSCAN 算法
OPTICS 算法
非负矩阵分解算法
隐狄利克雷分布 (LDA)

神经网络

自组织映射
感知机
反向传播算法
霍普菲尔神经网络
径向基函数网络(RBFN)
反传算法
自编码器(Autoencoders)
霍普菲尔神经网络



硬件：

机器算法需要进行大量的训练和计算，传统的CPU无法满足需求，因此发展出了GPU计算，将大量的简单计算放到GPU上；谷歌甚至研发了张量计算机（张量，学力学的是不是听着好亲切？）专门用作机器学习。有人说为什么不发展更大更多更快的CPU而要转向GPU和张量计算呢？性价比！性价比！性价比！任何工程学问题都讲究性价比。

数据：

巧媳妇难为无米之炊，数据就是人工智能的“米”，人工智能的计算依赖于大量和高质量的数据输入。比如要让计算机识别阿拉伯数字，必须输入大量的数字样本进行训练，数字相对简单，如果是其他复杂的物体，比如汽车，要达到理想的识别效果，需要的数据用海量形容一点不为过。这里提一下，阿尔法狗的能力也是来自于海量的数据训练，即对大量棋局都进行过训练识别，简单说阿尔法狗下过海量盘棋。早期的电脑棋局只是预先设计好程序，按照固定的模式应对棋局走法，这与阿尔法狗有着本质的区别。

4.2. CAE中的优化算法

人工智能算法的基础是数值计算和优化算法，包括了以梯度计算为主的传统算法和诸如神经网络，蚁群，模拟退火等智能算法，以及AI机器学习算法。用过CAE优化软件的朋友对前两种算法应该不会陌生。

CAE中传统优化算法：

优化算法在CAE仿真软件中广泛使用，是设计阶段的得力助手。但现阶段CAE的优化软件也有很多不足：

1. 工程上CAE仿真时间都很长，一般规模的用例短则几小时，长则几天，迭代次数难以满足优化要求。
2. 优化变量过多，多目标优化情况下，计算能力捉襟见肘；
3. 每种优化算法尤其特定适用条件和限制，很难找到一种通用的算法解决所有优化问题；
4. CAE优化通常有很强的业务逻辑，选择合适的优化算法需要工程经验积累；

5. 理想情况下优化流程是一个自动化的过程，但由于算法本身限制和实际问题的不确定性导致在优化过程中人工干预过多，使得优化结果具有一定的主观性；

4.3. AI与CAE优化算法；

AI目前还难于应用在CAE优化领域，原因在于：

1. 从计算量上看，AI需要大量计算，而CAE本身计算量也大，用在一起计算效率低下；
2. AI多应用于图形语音识别，模式识别，预测，决策等，在日常领域有着广泛的应用基础，而CAE迭代周期长，短时间难以出有效成果；
3. AI通常需要海量训练样本，如何有效的抽取高质量CAE训练样本是个难题；
4. 从非技术角度看，目前国内CAE应用普遍停留在第一层次，能在第二层次计算一下多物理场就很不错了。（参加了2017年上海站的Ansys新品发布会，会上提出了Ansys的四个仿真层次：1. 单物理场仿真；2. 多物理场仿真；3. 设计驱动仿真，即在设计阶段引入仿真，指导设计，而不是在设计完成后仿真验证；4. 业务驱动仿真，仿真与业务紧密联系，前面举过风力发电机的例子。发电机宕机-> 实时仿真->找到问题 -> 修复 ->再仿真 ->重启运行。在这个阶段，对仿真的效率准确性都提出了非常高的要求，也是Ansys公司现在大力推广和发展的阶段）

AI在CAE领域的应用前景是非常广阔的：

1. 拿飞机设计来讲，飞机从原始设计，仿真，验证到生产试飞通常耗费几年时间甚至更长。但是如果有足够多的原始飞机设计资料经过AI训练，一种新设计的飞机，几乎不用或者少量仿真实验，就可以得到飞机设计的整个性能参数，设计成本和时间可以大大降低；
2. 针对已有仿真结果而且拓扑基本相同的零件，不用仿真计算，就可以得到新设计零件的仿真结果，而且还可以有针对性的提出优化方案；
3. 老工程师凭借经验可以发现产品设计的不足，而AI不仅可以部分替代老工程师，而且分析的更精确，更加量化；
4. CAE建模仿真中的重复性劳动，比如划分网格等工作可以交给AI独立完成，而且可以在短时间内获得比人工更好的网格质量。

虽然现阶段的AI在CAE领域少有应用，但是随着AI的兴起和热门，以及软硬件的不断发展，AI应用到CAE领域只是时间问题。

作者从事工业仿真软件研发十余年，深知优化对于CAE应用的重要性，因此在优化算法方面积累了相当经验，作为工程经验的总结，开发了CAE优化软件FasOpt，后续会做介绍，在以后的开发中，也将逐步融入AI算法，尝试AI在CAE优化里的应用。

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

深入剖析三维几何内核(2)--功能介绍

原创 邓子平 多物理场仿真技术



深入剖析三维几何内核(1)--基础

深入剖析三维几何内核(2)--功能

深入剖析三维几何内核(3)--算法

我们将几何内核的功能进行分类，看看几何内核做了哪些事情，或者什么情况下上层应用会用到几何内核。

1. 二维草图

“草图”是一个专业术语。我们设计一个对象或者绘制一个对象时，开始很难用三维进行建模，通常的做法是先用二维表示，工程上的图纸一般也都是用二维表示，比如我们常见的俯视图，侧视图，截面图等等。AutoCAD之类的绘图设计软件最初都是二维的。

有人可能觉得二维比较简单，没什么难度，实则不然。单个的二维对象确实比较简单，点线面数据通过基本设计即可实现。

考虑如下设计：

两个互相咬合的齿轮，已经分别设计出齿轮的半径，进行了咬合。现在需要修改其中一个齿轮的半径，如果只是简单的分别绘制出两个齿轮，其中一个改动后，另外一个也需要手动修改以保证齿轮咬合。一般的做法是将两个齿轮关联，一个改动后，另外一个自动修改以保证咬合。如果一个齿轮同时和多个齿轮相连，则涉及到多个同时关联。

再考虑一种情况，在平面墙上绘制门和窗户，当我们修改墙高度和宽度的时候，门和窗户要随动，比如我们加了约束（窗户不能靠房顶，门底和地面平齐等等），随动要满足这种约束关系。

这就引申出几何内核里的一项基本功能：约束设计求解器。

约束设计求解器在于给定各种约束情况下，求解类似刚体运动自由度方程，约束设计求解器常用于二维草图绘制，零件在装配体中定位，以及各种几何外轮廓的绘制。

2. 三维实体建模，属性计算

三维实体建模是几何内核的基本功能。基于几何特征，几何内核需要提供完整的实体表达。最常见的长方体，根据长宽高等参数建立起对应的数据结构。以BREP为例，一个长方体至少包含如下信息：

1个体，6个面，6个环，12条边，12条有向边，8个顶点，8个顶点的坐标

这也是为什么专业软件的几何文件很小，而三维几何模型文件大的原因。

除了长方体，圆柱，圆台，圆环，球，楔形等各种基础实体模型外，各种常用的建模方法也是必须的，常见的拉伸，扫掠，放样，阵列，平移，缩放，矩阵变换，布尔运算，拆分。

创建的对象实体要提供边长，面积，体积，包围盒等基本信息。

3. 历史系统

历史系统主要用来跟踪记录几何对象的变化信息。这些信息在参数建模和约束建模中要经常使用，是undo/redo机制的基础，同时也是几何模型质量的保证。

看几个例子：

一步步设计好模型后，保存工程文件，关闭程序退出。再次打开工程文件，需要undo进行编辑

一个复杂的模型，执行到某一步出现错误，整体模型坏掉，需要恢复到上一步模型完好的状态

一个模型，需要进行一些修改才能拿到某些信息，但是并不希望对模型本身进行修改

以上需求都可以通过历史系统来完成

4. 属性系统

几何对象一般不会单独存在，几何对象上会有很多属性，比如颜色，显示方式，材料，业务类型，参数。对于数值仿真，还会有各种边界，荷载等信息。附加在几何上的属性系统也会涉及到增加，删除，修改，以及随几何变动而变动，比如两个不同属性的几何进行并运算，并之后新几何上的属性系统要有明确的设定。

5. 参数建模和约束建模

参数建模和约束建模都很好理解。底层需要有参数系统，基本功能包含参数表达，驱动更新，参数表达式计算，几何对象参数管理，以及约束求解器等等。这些功能是CAD软件几何设计(Design)的基础，而非Drawing

几何拓扑，属性系统，历史系统，参数建模和约束建模构成了几何内核底层管理的基础。

6. 几何离散化

几何内核要提供几何离散化功能，主要有两个目的：1. 生成基本的三角面片信息供渲染使用；2. 提供更多参数控制的面片的生成，以满足上层业务不同的需求。

7. 位置运算

位置运算提供各种几何之间的位置信息计算，常见的线线，点线，线面，点面，面面，面体，线体等各个维度几何的距离，相交，交点等信息。

通常位置运算需要大量使用计算几何或数值计算方法，这也是几何内核底层的基本功能之一

8. 布尔运算

布尔运算涉及到两个对象之间的合并，减，公共基本操作。在基本操作上衍生出切分，投影，缝补等操作。布尔运算主要涉及到线线，面面，面线的求交和裁剪运算，后续详细介绍。

布尔运算的准确性，稳定性和性能是衡量一个几何内核质量的重要标准

9. 特征识别和抑制

识别模型中的倒角，突出，凹陷，以及常用基本几何实体，并对其进行编辑操作。

10. 外部数据交换

在实际工程中，通常使用的软件和数据文件格式有多种，为了保证整体数据格式的统一，几何内核需要提供与外部的数据的接口，即能导入和导出第三方的文件格式。这一块和几何内核没有直接关系，技术上也没有太多难度，主要问题是不同数据格式的接口兼容性问题，比如公差，数据表达转换。

11. 与网格接口

这个在 深入理解数值计算网格(7)--几何与网格 已经做过介绍，不多讲。

12. 几何清理修复

除非建立封闭的生态系统，否则这是几何内核工程应用的一个基础模块。在建模工程中，因为种种原因导致模型出现问题，比如各种拓扑和几何不满足设定条件或者出现错误。这种错误会导致模型本身的分析，比如面积，体积计算出错，还会导致仿真的网格划分出现错误。

除了提供模型的拓扑几何检查外，还要提供基本的清理，修复以及必要的编辑功能。

很多时候CAD设计的模型即使没有错误，也无法用来直接仿真，比如CAD设计中的小边，小孔，倒角，印刻的字体，需要在划分网格前对其进行删除清理。

除了模型几何对象本身的错误外，实际应用中还要进行诸如几何对象之间的干涉检测，自由对象，重复对象的检测操作。这些功能可以在业务层实现，也可以在几何内核中完成。

13. 图形显示接口

几何内核本身是没有图形显示系统的。换言之，几何内核创建的数据无法直观的表现出来。为了将几何对象显示，需要和外部的图形显示引擎合作。通常的做法是将几何对象离散化面片，然后送给图形显示引擎显示。离散化的面片和几何对象建立映射关系，以方便后续的操作。

14. 提供仿真所需的几何功能

比如提供抽中面，供有限元分析中的壳单元使用；三维实体简化成一维单元，供有限元分析中的杆，梁单元使用；处理非二维流体，以解决复合材料，流体，碰撞，接触分析中的共形网格；提供高次曲面以满足透平机械设计分析需求等等

15.脚本系统

提供脚本系统，便利上层应用的开发和调试

16.提供必要的调试工具

使用几何内核出错后，往往难以检查和调试，提供必要的工具帮助发现和解决错误

17.支持高性能计算

提供工具方便上层应用使用多线程，多进程以及分布式计算

CAE软件研发的一些思考(3)--CAE软件架构设计

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



这里的CAE是广义的，包含了传统意义上的 CAD/CFD/EDA/CAE/CAPP等
本文的设计主要覆盖软件工程流程中的[概要设计](#)和[详细设计](#)。因为客观原因，CAE软件与软件标准开发流程有所不同，设计文档更倾向基于可快速实现的[原型开发](#)。
设计中并不强调某一模块的功能，也不追求某一方面的性能效率，而是基于软件工程，着重搭建一个通用性，可靠性，稳定性，扩展性，可测试性，维护性优秀的CAE软件平台。
基于此平台，既可以开发大型通用有限元软件，也可以快速开发出行业CAE软件产品。

第一部分：模块划分

根据功能分以下几大[模块](#)：

1. 输入输出 (Input/Output或 I/O)
2. 几何
3. 有限元模型
4. 后处理器
5. 求解器
6. 图形

-
7. 公共模块
 8. 高性能计算 (HPC)
 9. 参数优化设计

用例图 (略)

第二部分：详细设计

1. 输入输出模块(Input/Output)

1.1.文件的读入与写出。文件类型包括：

- 1>. 软件自定义的工程文件
- 2>. 标准的CAD文件 (DWG/Step/IGS/SAT/X_T/STL/Model/等)

3>. 标准有限元模型文件（ bdf/cdb/inp/k/NEU/ModelFlow/I-deas/Ansa/Mar/Admin等）

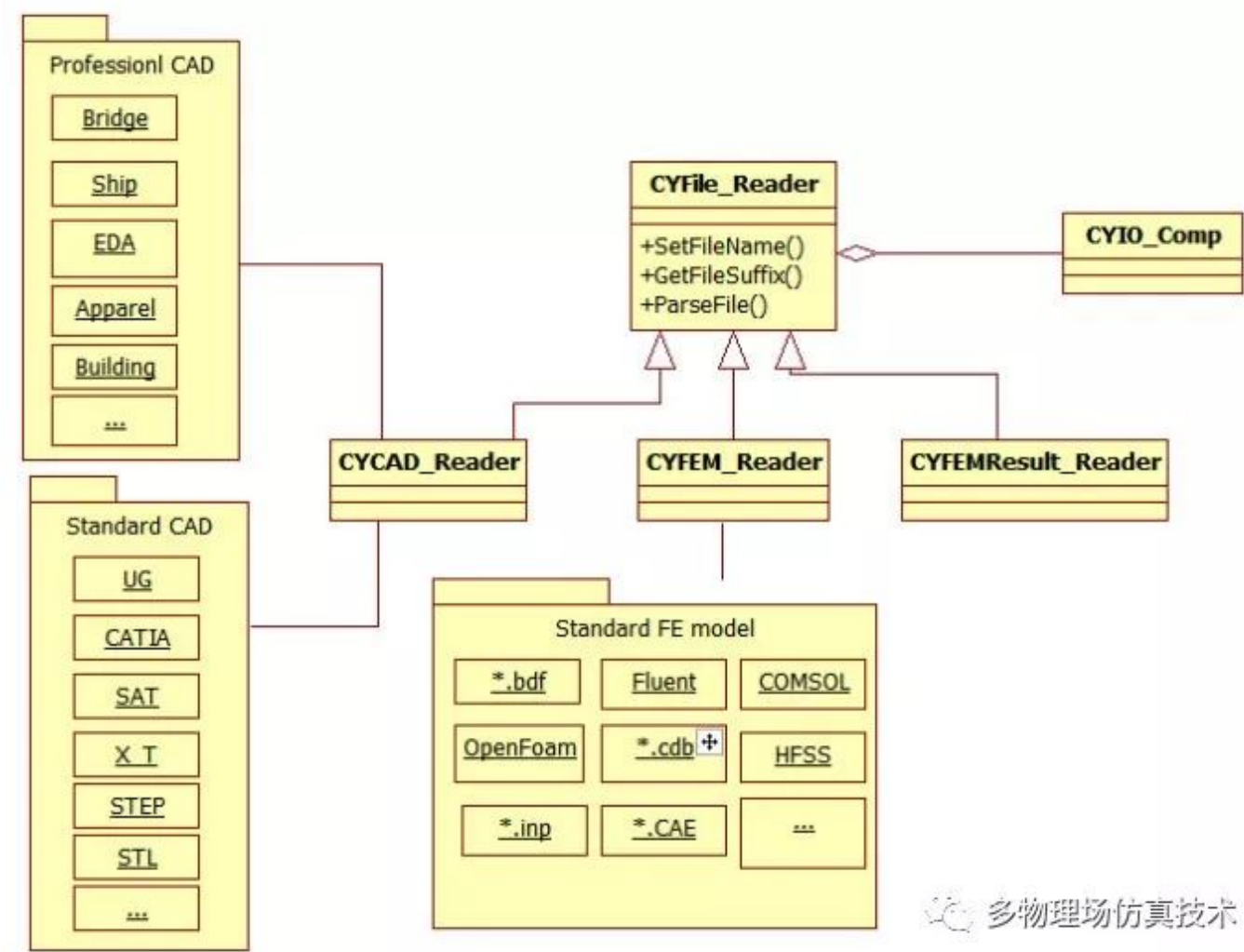
4>. 标准的行业CAD文件（制衣CAD格式，PCB板CAD格式等）

需要说明的是第四种文件，该文件的格式数据通常包含了几何描述信息和属性，几何描述信息是指无几何的拓扑信息（比如一个长方体，几何描述信息是用初始点坐标和长，宽，高四个参数来表示，而没有实际的点线面体拓扑信息），读入该类文件后，需要用CAD内核重新建立模型。

类图

1.2. 外部接口

通常CAE软件能够供其他软件调用，或者作为其中一个模块，也能够实现参数化命令行调用。



2. 几何模块 (Geometry)

这里的几何指广义上的CAD，包含了几何的创建，编辑，管理等。一般的商业CAE软件提供了CAD功能，都比较简单。对于复杂的实际工程，很少在CAE软件中建模，而是在专业CAD软件（Catia/UG/ProE/Inventor等）或者行业CAD软件中建立几何模型，然后导入到CAE软件中。所以CAD不是CAE软件的重点，但是好的CAD模块能提高CAE仿真的效率，避免外部CAD数据与CAE接口之间的损耗（最典型的的就是CAD数据修补）。CAD建模本身也是一项技术含量很高的工作（参数建模，约束建模）而且

Catia/UG/ProE/Inventor这些产品在CAD方面已经做的非常好了。CAD与解方程组数值算法一样，也是基础性学科领域。

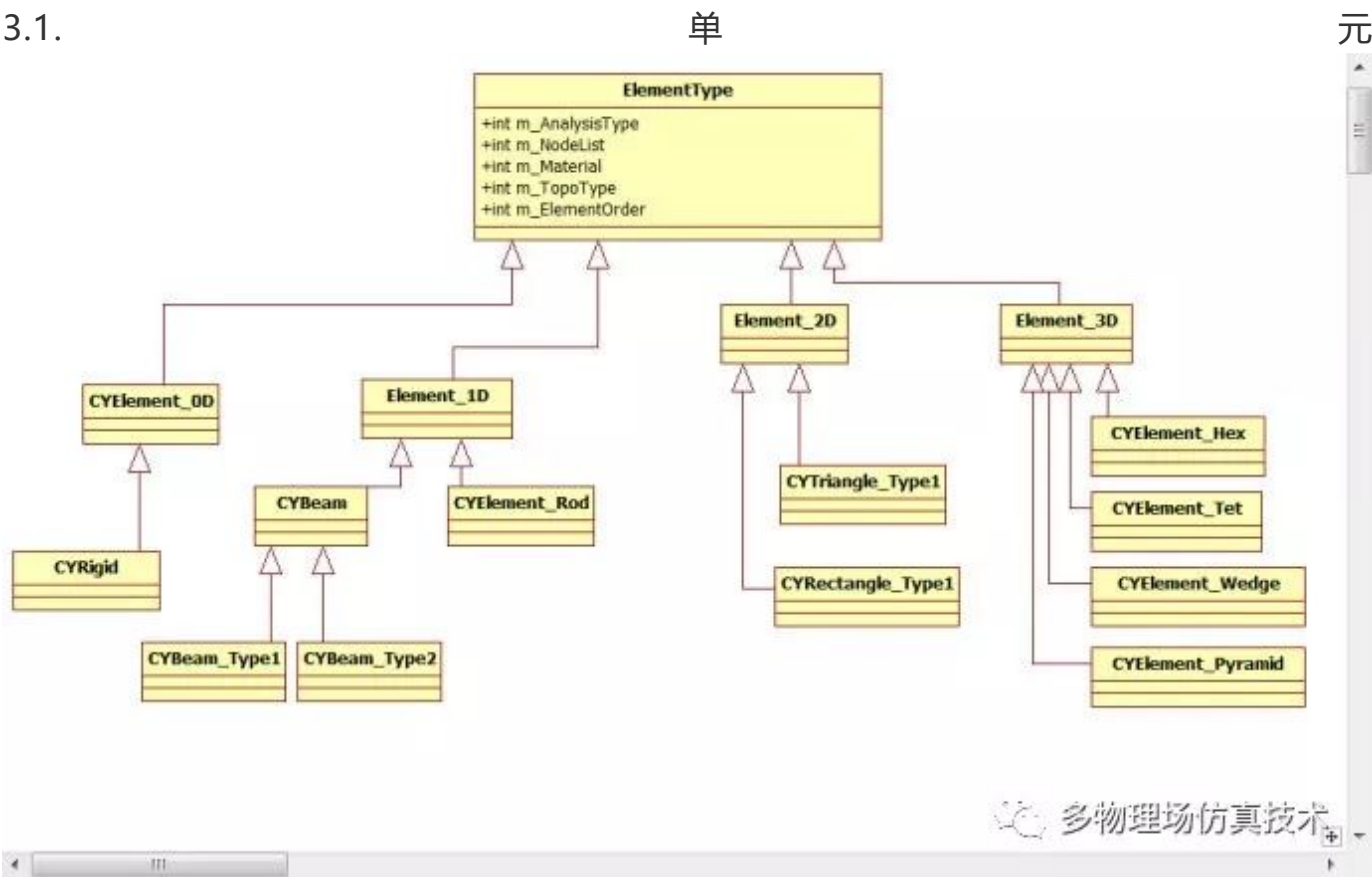
这里不涉及通用的CAD建模，重点在建立CAD模板。所谓的模板是指可反复使用，用户输入参数或者导入模板参数文件，即可建立所需的CAD模型，不需要用户手工去生成。

（比如用户定义了一架飞机模型参数，参数可以保存为文件，生成实体几何后，修改机翼长度的参数，可再次生成模型而不用做其他修改）。这也是专业CAD软件的优势所在。

CAD建模，首先需要内核，目前商业的3D CAD内核有Parasolid和ACIS，Granite，开源的OCC。国内的CAD软件长期没有自主核心技术，在CAD内核方面也鲜有成熟产品。

3. 有限元模型(FE Model)

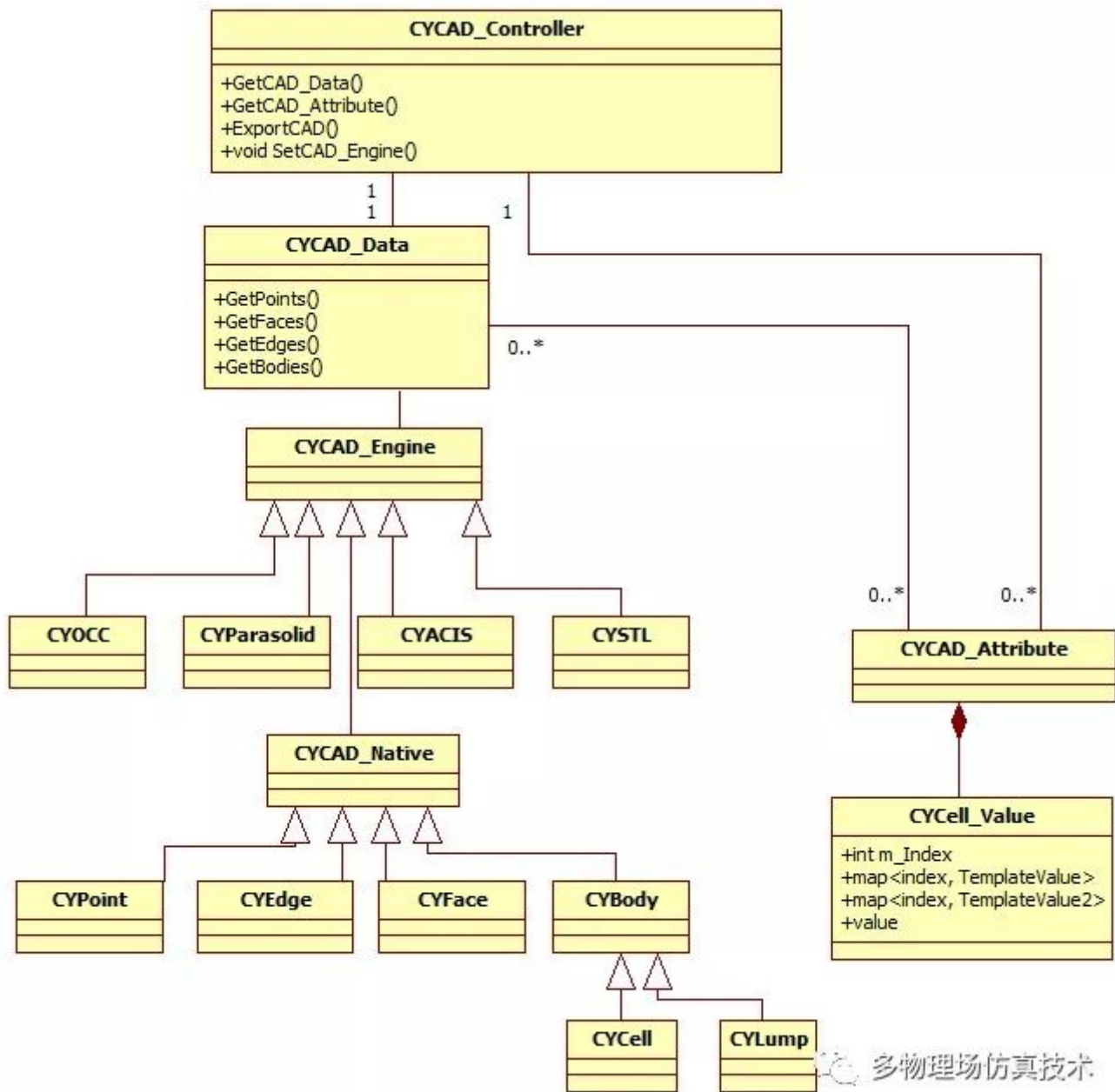
3.1.



面向对象方法很适合构造有限元中的单元，ElementType为所有单元的基类。主要方法有：点的集合，材料，分析类型，阶次。根据单元的空间维数将单元划分为 0/1/2/3D单元。如需扩充新单元类型，在相应的单元类型上派生即可。

3.2. CAD和CAD属性（边界条件）

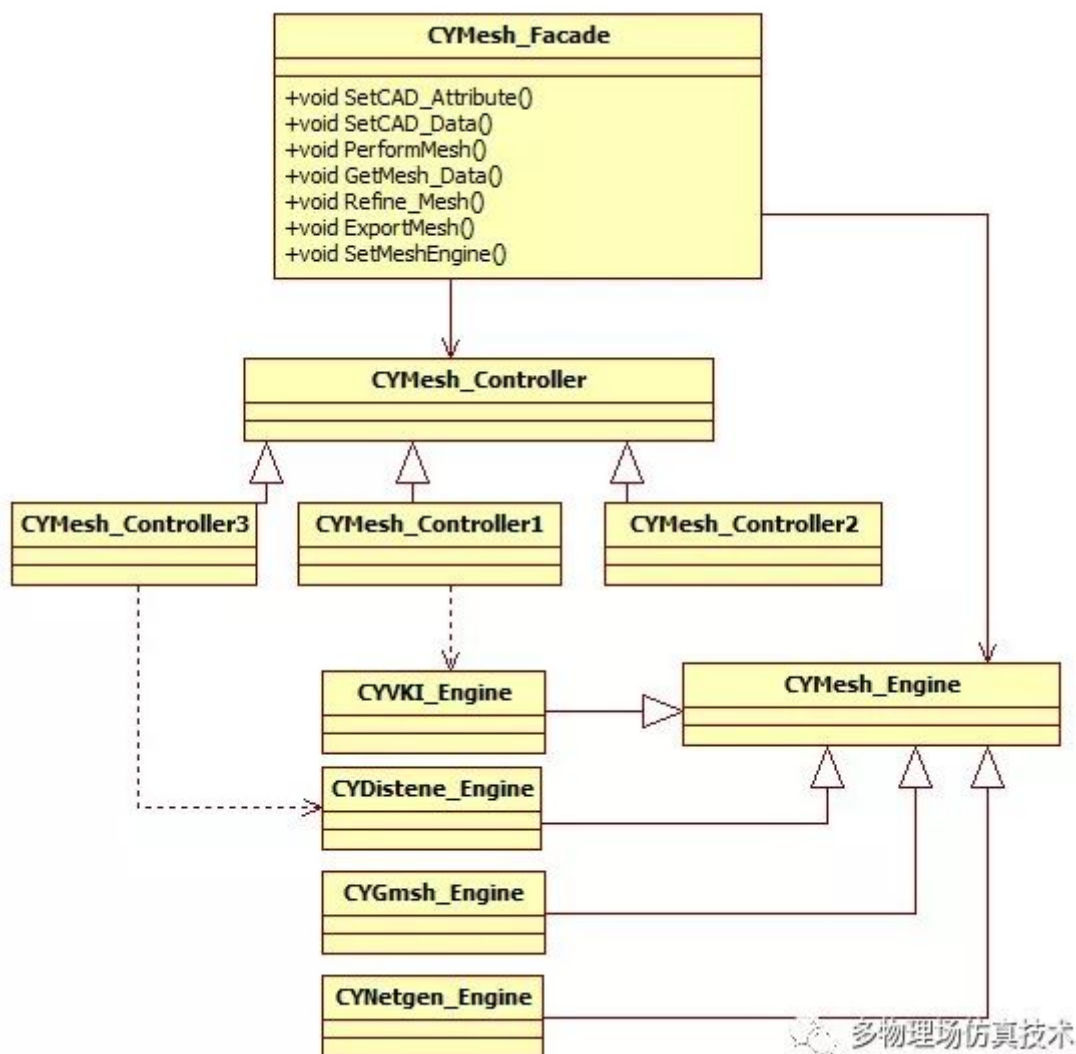
这里边界条件泛指任何在几何（点，线，面，体）上的属性（位移/荷载/温度/吸收边界/约束。。。)



多物理场仿真技术

CYCAD_Controller 负责获得几何数据和附加在几何上的属性（边界条件）。CYCAD_Data 存储几何数据，CYCAD_Attribute 存储与几何对应的属性。CYCAD_Engine是CAD内核的接口，通过接口也可以调用自己开发CAD内核。基于该结构，实现了CAD与MESH的解耦，对于已经有的CAD/Mesh接口，能实现最小限度的修改。

3.3. 网格划分



从名称可以看出采用了Factory和Facade模式，CYMesh_Facade负责与外部的接口，主要是设置1. CAD数据，2. CAD属性（材料，荷载，温度，边界等），3.网格划分参数和策略。Controller主要负责Mesh流程，CYMesh_Engine 是执行Mesh引擎的接口，可以根据实际需要选用网格引擎，以上选取了Gmsh，NetGen，Distene，VKI等，在此结构上，开发人员可以方便的对网格划分流程和网格引擎进行扩展。

3.4.网格属性

网格划分完成后，通常做法是把网格数据保存为规定的文件格式以供求解器调用。也有直接在内存中供求解器调用的，虽然提高了效率，但是不利于调试。对于自主开发的求解器，可以把网格导出为Ansys/Nastran等求解器的格式，以方便做Benchmark。

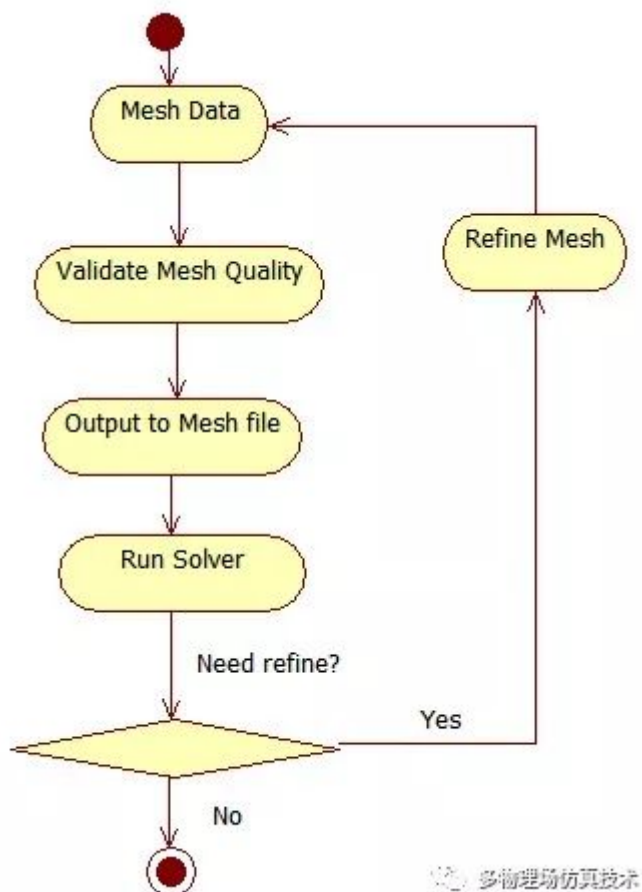
网格的数据包含了如下信息：

1. 节点信息：索引和坐标（Vertex）
2. 边信息：索引和点索引（Edge）
3. 面片信息：索引和边索引（Face）
4. 单元信息：索引和节点索引（Element）

通常以上信息只需得到 点和单元的信息，其它都可以求出。

5. 属性标签：加在几何上的属性会分配在相应的网格单元或者节点上。（边界条件/荷载/材料 等）

网格活动图：



说明：

1. Mesh Data为 网格数据
2. Validate Mesh quality 验证网格的质量，网格质量指单元的形狀的好坏，有通用的公式计算（比如最大角度，长宽比，Aspect ratio/Skew 等）。网格质量的好坏直接影响到求解精度
3. Output to Mesh file导出为网格文件
4. Run Solver 调用求解器
5. Need refine:根据求解器反馈的结果和预定义的收敛准则，判断是否需要加密网格
6. Refine Mesh: 加密网格

4. 后处理

后处理是对仿真结果的处理，包括可视化，归纳分析，导出报告，与实验数据对比等。技术上来讲没有太多瓶颈，主要是数据的组织，再就是大数据显示的效率。

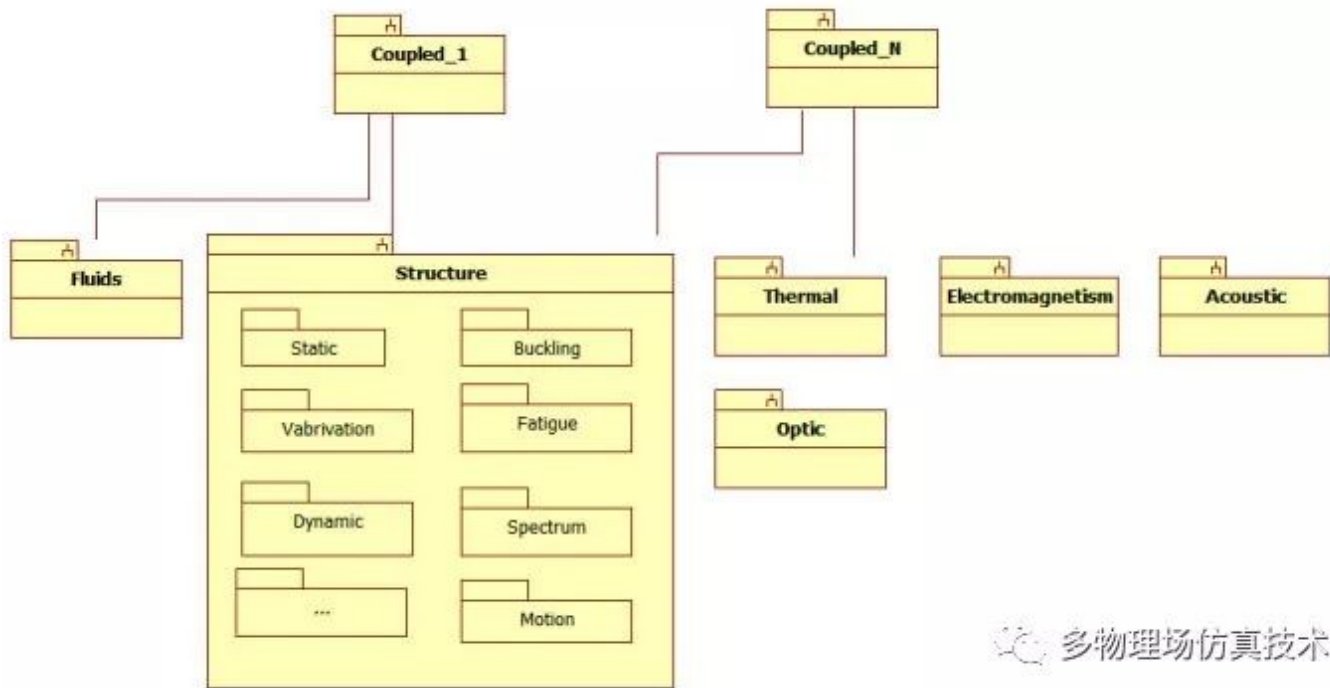
- 3.1. 可视化主要包括等值线，云图，XYplot，动画等，
- 3.2. 归纳分析，是指对仿真的原始数据进行加工，得到所需要的参数，比如VonMises应力，S参数等。
- 3.3. 与实验数据对比，这个也是以后CAE仿真的一个发展方向：即如何将仿真得出的数据与实验数据，保证仿真数据具有实际参考价值。

后处理主要用到**图形显示模块**。

目前市场上比较通用的商业后处理器：Enight, HyperView, Femap.

5. 求解器

求解器核心是求解偏微分方程(PDE)，有限差分/有限元/有限体积等数值方法二三十年前就已经很成熟，功能上没有太多难点。开发求解器主要在稳定性，扩展性，效率以及定制化等方面还有很多改进余地。开发求解器面临的主要问题是技术问题，就跟键盘布局一样，大家用习惯了，再出来新的除非特别优秀或者有足够多的亮点，否则用户很难认可。拿求解器Radioss来说，Altair声称兼容Nastran，但是行业认可度并不高。另一个例子是Comsol，Comsol在求解器，前后处理上并无太多亮点，但打出了多物理场的旗号，并且在多物理场建模方面也确实提供了一些便利。



求解器涉及的领域有结构，电磁，热，声，光，流体等领域，可参考附件。

6. 图形

该模块包括以下几部分：

1. 几何模型，有限元模型，结果的可视化
2. 几何有限元模型的图形化编辑
- 3 GUI设计

1. 几何模型，有限元模型，结果的可视化

可视化开发工具首选OpenGL，使用OpenGL需要较深的图形学功底。目前绝大部分可视化工具都是基于OpenGL。也有些公司在OpenGL之上做了封装，提供了更高层次的调用接口。

开源可视化工具VTK，ParaView。

也有提供商业可视化工具的（VKI/HOOPS/Gravies/），价格不便宜。

2. 几何/有限元模型的图形化编辑

涉及到编辑几何，编辑网格。允许用户通过鼠标，输入等方式编辑几何和网格，支持更复杂的功能（比如装配体的网格组合）

CAE软件通常采用卡片式编辑，即在一个ListView中按Index显示所有内容，提供编辑，删除，添加等，在新的窗口中实现功能。

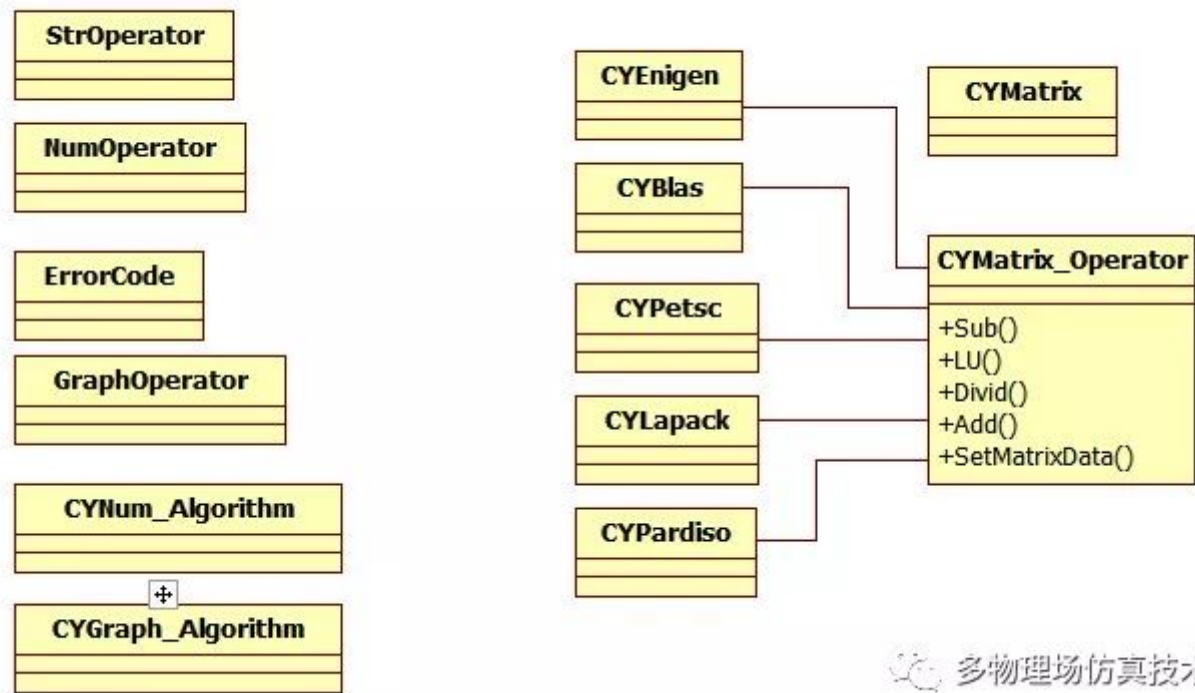
3. GUI

图形化用户接口开发工具有MFC，QT，WPF等。最近十年QT的发展已有超过MFC的势头。作为GUI开发工具，QT具有使用简单，不依赖平台，开发资源丰富等特点，可作为开发的首选。

因为GUI涉及到软件的每个部分，同样，为了建立兼容的CAE开发平台，在开发之初最好能确定GUI工具，以后就不再改动。

7. 公共模块

公共模块没有太多核心技术，实现也没有难点，但在CAE开发中相当重要，尤其到开发的中后期，公共模块的重要性更加突出。一个好的公共模块能大大提高开发测试效率。



公共库主要包括 字符操作，数值计算，基本数据结构，错误返回代码定义，图形算法，常用算法，矩阵类等。

CYMatrix_Operator是定义的一个中间矩阵类，主要是统一矩阵操作的接口。考虑到求解器用的矩阵库各不相同，通过此接口可以统一，但性能上会有所降低。

8. HPC(High performance computation)

参考 FEM之求解器加速(1)---HPC简介

9. 参数优化设计

参考 FEM之优化算法(1)

总结：

1. 本文只是一个初步的架构设计，其中有很多细节很难用文档来描述。

2. 设计中参考了如下软件：

ANSA/Catia/SolidWork/Patran/Nastran/Ansys/Abaqus/HyperWork/HFSS/FEMAP/LMS/ADMIN/Comsol/Cubit/SimLab

写文本的目的只想说明，不管是开发大型通用CAE软件，还是行业的CAE软件，在技术上都是可行的。

从技术角度聊聊“系统工程”

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



2017年，法国导演吕克贝松拍了一部科幻电影《千星之城》，电影一开始介绍了千星之城的由来：最开始是一个环绕在地球轨道的国际空间站，然后各个大国的空间站陆续对接上来，后来全球所有国家包括第三世界国家都发射了空间站，对接到国际空间站，再后来外星人的星际飞船也对接到国际空间站，对接的外星飞船越来越多，越来越大，以至于整个空间站已经开始对地球引力产生影响，威胁到了地球安全。最后整个空间站不得不移动到别的星系。

从“系统工程”角度看，不同空间站对接需要明确软硬件，通讯，补给等接口，对于地球上的空间站，可以事先确定好；对于外星飞船，需要在不知道接口的情况下完成上述功能，此外，对于所有的空间站要考虑到安全管理，权限设置，资源分配等事项。所以对于管理者，即需要对尽可能了解每个空间站的细节，又需要从整体上完成管理职能，而这正是“系统工程”核心思想的体现。

在工程和软件研发领域，有一个耳熟能详的故事：瑞士瓦萨号战舰沉没事件。

“瓦萨”是一艘古战船的名字，它是瑞典国王古斯塔夫二世于1625年开始建造的，1628年首航几分钟便沉没，直到1961年才重见天日。

战舰在建造过程中，因为国王的干预，在最初设计上做了各种改动，其中加重了战舰的装饰功能（增加了大量无用的装饰品和雕塑，当然让它成为了少有的艺术品，这是后话），同时加大了火炮数量，徒增加了船体偏心重量，另一方面，需求改动过大，单层改成双层，造成了船体重心上升，而初期又没有进行足够的验算测试，最终酿成了悲剧。

系统工程是一个用于实现产品的跨学科方法。它将每个产品作为一个整体来理解，更好地构建产品规划、开发、制造和维护过程。系统工程一定是来源于实践的提升，即需要对每个细节了解，又要从整体上对产品进行把握。

开发工业仿真软件，除了按照常用软件研发模式外，需要用“系统工程”的思想来指导开发。工业仿真软件基础模块众多包括几何内核，图形引擎，用户交互，外部文件接口，业务文件接口，单元管理，网格划分，仿真模型生成管理，求解器，高性能计算，后处理，对比基准测试，实物测试验证等等。除了技术之外，作为技术工业标准，还需要试验数据，很多软件公司没有试验能力，需要借助第三方资源或者由客户来帮助。早年很多中国客户，不仅高额购买外商软件，客观上也在帮助其迭代产品。

从“系统工程”角度看，就不难理解为什么主流的商业仿真软件都在大公司手里，而不是小公司。一方面，仿真软件的“系统”性说明了小公司除非在某个点上有很大的突破，按照传统技术小公司很难靠几个人开发出优秀的产品；另一方面，仿真软件的“系统”复杂性就决定了只能有大公司来主导，小公司一旦在技术成熟，再往上走不管是在技术，市场，销售等方面都很难再有大的突破，这也是仿真软件行业“大鱼吃小鱼”，不断并购的根源所在。

一篇文章入门前处理器开发

原创 邓子平 多物理场仿真技术

收录于话题

#仿真研发工具

21个



本文介绍前处理器开发的相关内容，主要还是围绕如何开发高质量的商业CAE软件。

1.首先介绍一下Cubit

Cubit是由Sandia国家实验室开发的一款前后处理软件。早期Cubit是免费使用的，推荐用作OpenFOAM求解器的前处理，现在已经改名，成为商业版本。官网是<https://cubit.sandia.gov/>

主要是从技术角度对软件进行解析。

以下是Cubit的一些功能介绍：

1. I/O 支持以下格式:

ACIS sat, sab: 几何模型输入/输出

Pro/E: 几何模型输入/输出

中间格式: STEP, IGES

曲面格式: STL, AVS面

Exodus II : 本地网格格式

Abaqus, Patran : 网格及边界条件输出.

NATRANA边界条件, IGES通用格式, DYNA3D格式, 流体: 只有网格格式输出.

支持Patran,Abaqus,Pro/E,IDEAS,Matlab,ANSYS等工程软件

2. CAD/FEM

几何模型树工具包含实体模型中所有的层次结构。它包括组件、边界条件、组和几何实体。在树中选择一个实体,将在图形窗口中选择相同的实体。

诊断几何模型及非特征化几何模型。

诊断及非特征化几何工具，包含诊断工具及分析和修复工具。

诊断网格化及检查网格质量

诊断网格质量工具可以用来确保模型的可被划分网格，在网格生成后，可以执行单元质量检查，如果当前网格的质量太差，提供网格改进方案。

3. 网格功能：

三角形表面网格（波前算法）

三角形表面网格（Delaunay 算法）

四边形面网格

四边形面网格（映射算法）

四边形面网格（子映射算法）

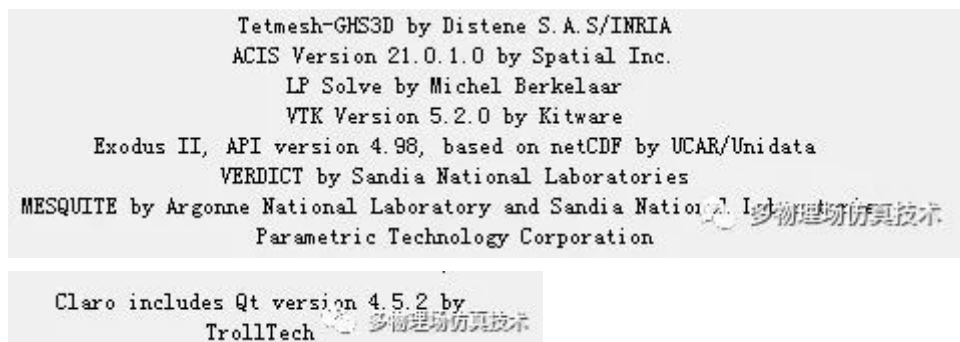
四面体网格

六面体网格（扫描）

六面体网格（映射）

还有其他一些CAD建模，优化，参数化建模，接口等功能，基本上覆盖了CAE前处理器所有设计模块。

看一下Cubit的一些技术实现：



1. TetMesh-GHS3D

该组件是由Distene开发的商业网格生成器，提供了丰富的接口，可快速生成高质量的Tri/Quad/Tet/Hex 等网格。是为数不多的能自动生产Hex网格的引擎。试用过该工具，参数简单，生成速度非常快，而且网格质量也很好。

2. ACIS

使用最广泛的商业3D内核引擎之一，可以用于实体建模和编辑。

3. VTK

著名的开源图形显示包，用于很多商业/开源图形显示程序。

4. ExodusII

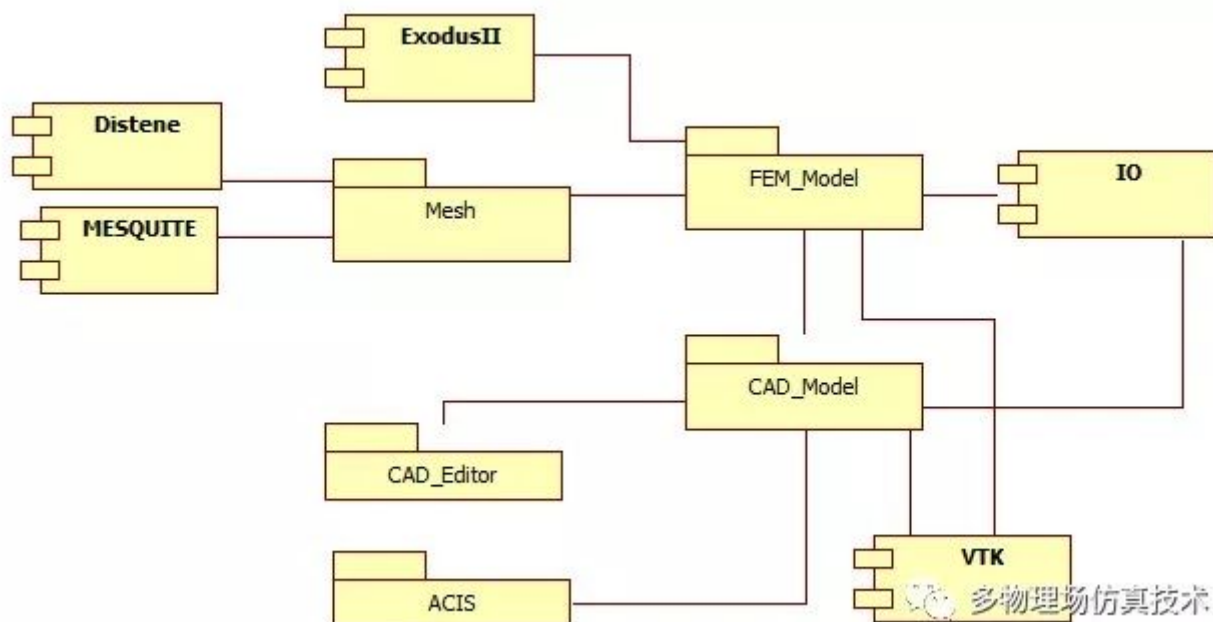
用来处理有限元数据的一个模块，可以用来做前后处理。

5. MESQUITE

改进网格质量工具。

6. QT

使用最广泛的GUI开发工具，提供有免费和商业两个版本，免费版本对于通常开发已足够。



可以看出Cubit中使用了众多的第三方工具，其主要亮点是向导式交互 几何修复，参数化建模，网格快速生成，网格质量诊断，较全的外部接口等。相比于ANSA，Hypermesh，SimLab等算是一个轻量级的前处理器。从技术角度看，开发这样一个前处理器不存在任何瓶颈。

2.几何修复

我们很少会在仿真软件里建立几何模型，大部分情况下是在专业CAD软件中建立几何模型，然后导入到仿真软件中。这种几何和仿真的分离是目前仿真软件中第一道瓶颈，也催生了仿真前处理中的一个专项：几何修复。在设计和加工阶段，很多细节是需要注意的，比如倒角，圆角以及一些细小的特征，但是在仿真中并不需要，而且这些特征会导致网格划分失败。再比如设计中两个相连的线由于没有使用捕捉并不完全重合，导致自由边的存在，此外几何设计最后导出文件，会丢失某些参数信息，从而导致修复困难等等。

几何修复涵盖内容比较广泛，目前市面上有比较专业的几何修复工具。针对特定行业，特定领域以及特定特征的几何修复是前处理器开发的重点。

3. 边界荷载

前处理器重要的功能是建立离散模型，离散模型中最关键的信息就是边界条件和荷载，直接决定了模型的特性和仿真结果。这里介绍如下信息：

1. 稳态边界/荷载与几何的关系
2. 稳态边界荷载与网格的关系
3. 边界/荷载生成自动化
4. 边界/荷载数据结构定义

边界荷载介绍

求解定解的PDE（偏微分方程）需要知道边界条件，对应于CAE软件中就是边界条件和荷载条件，简称边界荷载

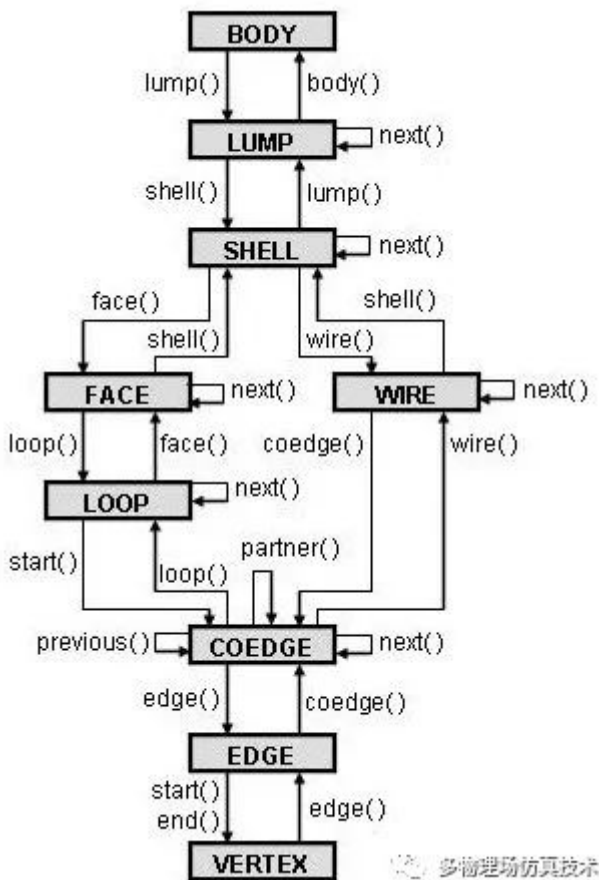
1. 稳态边界/荷载与几何的关系

稳态通常是指PDE中不存在与时间有关的项。

一个好的前处理器能 让用户方便的将边界荷载加载到任意的几何上（用户能激活不同的选择模式，比如Edge Level激活，用户的选择操作只能选中边，然后将边界荷载与选中的几何关联起来）。一个定义良好的几何通常包含几何信息与拓扑信息，几何信息用来构建几何，拓扑信息用来构建几何信息之间的关系，参见（图1）。一般化，几何信息包含 点，线，面，体，加载到这些几何上的边界荷载最终会以属性形式加载到网格单元的 顶点(Vertex)和面片 (triangle/quadrangle) 上。

以ACIS为例，介绍边界荷载如何加载到几何上，其它几何内核与之类似。

ACIS有一套严格的自顶向下的几何/拓扑层次结构。给定一个对象（Entity），可以获得与其相连接的对象，不管是向上还是向下。所以加载到几何上的边界荷载也可以很容易得到相连接几何上的边界荷载信息。



多物理场仿真技术

以下是一段将荷载加载到ACIS面上的代码：

```
void Attach_Geo_Loads()
{
    //给定实体
    BODY* ReulstBody;
    //存储所有的面
    ENTITY_LIST AllFaces;

    //得到所有的面
    api_get_faces(ReulstBody; , AllFaces);

    FACE* curFace = (FACE*)AllFaces[0];
    //选中第一个面，该操作可以通过GUI生成，在知道几何准确信息的情况下，可以代码直接生成（自动化）

    api_get_entity_id((ENTITY*)curFace, tagID); //得到面唯一的ID

    map_Geo2Loads[tagID] = loadIns; //将面的ID和荷载关联起来
}
```

其他边界荷载，加载到体，边，点上 的操作类似。

在几何上加边界荷载的实质就是：将边界荷载信息和指定的几何进行关联，建立映射关系。

建立模型时候，经常会碰到多种材料，接触，装配的情况，称为NonManifold（非流）几何。Parasolid/ACIS/OCC 都提供了处理 NonManifold 几何的功能，Parasolid/ACIS还提供了对NonManifold几何三角化网格的功能。

2. 稳态边界荷载与网格的关系

边界荷载加载到几何上后，进行剖分网格。加载在几何上的 边界荷载信息会附属到对应的网格单元上。例如对上例中的实体进行六面体网格划分，其中第一个面上的边界荷载信息会随之附属到该面上的四边形，即每个四边形都有该荷载的信息。根据不同的计算方法，边界荷载信息即可以放到每个四边形上，也可以放到四边形的点上，还可以放到四边形的中点上。

一个好的前处理器能支持对 网格上的边界荷载信息的编辑。例如用户想改变某些网格上的荷载，要支持对网格单元的选择，赋值，删除等操作。

3. 边界/荷载生成自动化

边界荷载生成自动化的意义在于 对大模型可以快速建立有限元模型。试想一个大模型，验算之后发现几何或者网格不合格，修改模型和网格过程中，边界荷载信息丢失，不得不对边界荷载信息也进行修改，大部分工作是重复性的手工劳动。参考上述代码不难发现，荷载是与几何的ID建立了映射关系，并不是与几何实体。当几何实体发生改变时，仍然可以用该几何的ID来标示。也就是说Tag与几何始终保持着关联关系。也就达到了一次建立多个有限元模型的目的。前提是几何只能调整参数，而不能改变拓扑结构。

4. 边界/荷载数据结构定义

边界荷载种类繁多，在软件实现中，如何正确的定义与组织数据结构很关键。这里的边界荷载仍然是一个广义的概念，不仅仅是力学中的边界荷载。

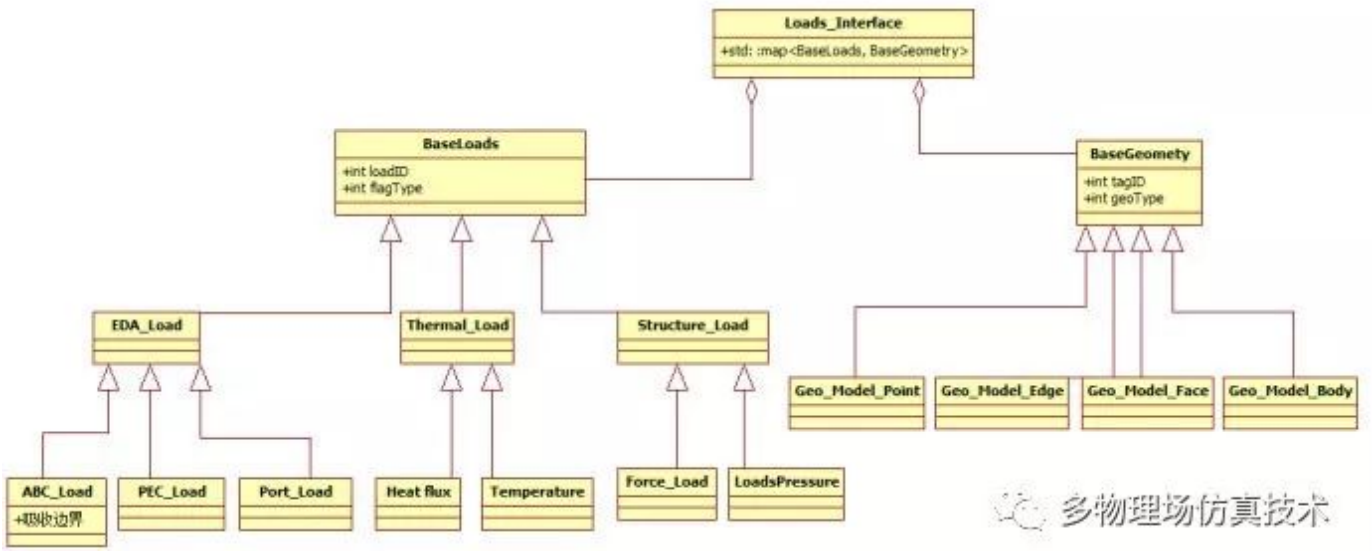


图2。

图2展现了荷载边界与几何建立映射 的 类图。因为篇幅原因只画出了部分边界载荷类型。

其中EDA主要针对电磁计算。边界荷载通常包括 ABC（Absorb boundary condition）PEC（Perfect electric conductor），Port（激励）等 Thermal 包括 temperature，convection，flux，heat transfer 等 structure包括 force，pressure等。

图2的定义满足了绝大部分荷载数据，以及对几何模型的映射，也能满足对有限元模型的映射。

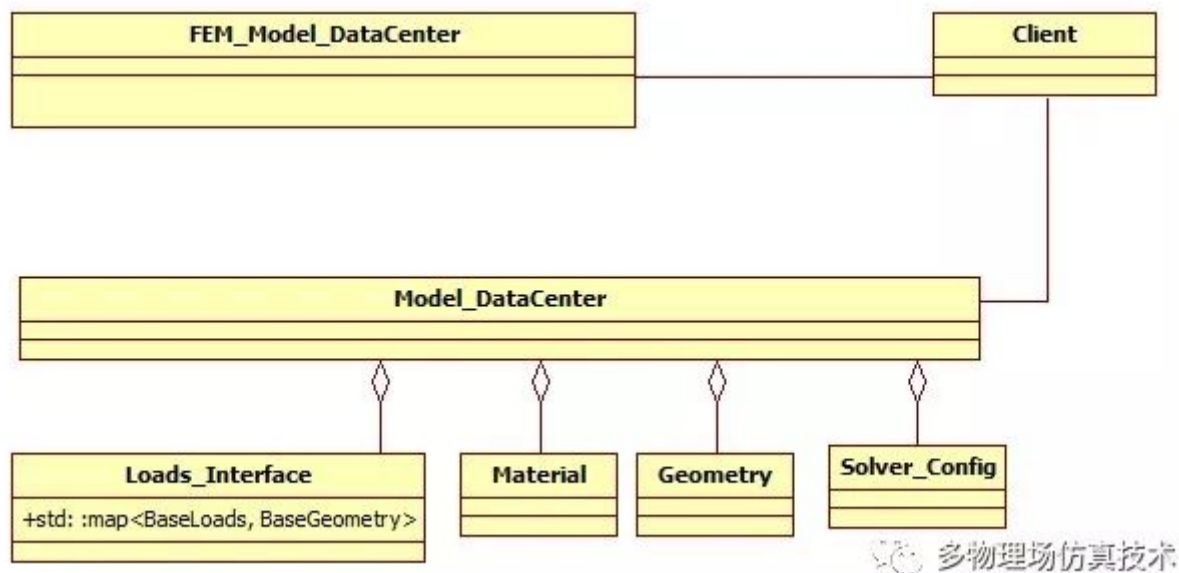


图3

图3列出了边界荷载数据结构在模型数据结构中的位置。

其中 FEM_Model_DataCenter 和 Model_DataCenter 是将几何数据和有限元模型数据通过Facade模式用warpper进行包装，供外部客户端调用数据，例如图形调用数据用来显示，求解器可以获取有限元模型数据生成特定格式的求解器输入文件等。

4. 网格划分

由于网格划分在前处理中的核心地位，专门一篇文章介绍介绍网格开发。

参见

一篇文章入门网格划分

5. 性能问题

性能是所有工业软件绕不开的一个话题，前处理器也会有性能方面的问题，后面会专门一篇文章介绍性能相关内容。

开发一个ABAQUS多少钱？

邓子平 多物理场仿真技术



“我们只需要做CATIA百分之五十的功能，需要多少钱？”

“投资一千万，两年可以做个ANSYS吗？”

“我们有母公司背书，五年可以投入1个亿，开发一个COMSOL要多久？”

“MATLAB禁用了，不是有开源的吗，我们投500万把开源的改进一下替换它？”

“ACIS不让用了，用OCC换一下不就可以了吗？”

“工业软件会成为下一个热点，国内有推荐的股票吗？”

“开发一个ABAQUS多少钱？”

简单说一下，从1987年ABAQUS具有前后处理器算起，研发加上维护，不考虑通货膨胀，费用平均每年在3-5千万美元左右。

少一点“国产自主”，多一点“合作共赢”:写给国内工业软件发展

原创 www.cae-sim.com 多物理场仿真技术

收录于话题

#大咖话题

13个



当前“国产自主”成为技术领域的一个热词，但是“国产自主”并不是新热点，距今较近的九十年代一系列事件让国家层面感受到了技术受制于人的切肤之痛，从而有了2000年之后的技术大投入，促进了基建，航天航空，电子通讯等领域的快速发展。

了解工业软件现状的人都知道，国内几乎所有主流软件都是欧美公司的产品。造成这种局面的原因之前也做过很多分析，不再累述。肯定会有人问当前不是提倡国产自主吗，为什么还要少一点呢？其实国产自主本身并没有问题，但是过度强调“国产自主”很容易出现诸如技术封闭，产品低端，依赖补贴等一系列问题。仔细分析当前国内热点报道的技术断供事件，本质上很大程度已经是商业问题，而非简单的技术封锁。在国际经济技术依赖加强，共赢合作更紧密的趋势下，一味强调“自主”只会固步自封，将自己绝缘于世界主流技术发展之外，之前由于政策上一味强调“自主”，至少公开报道中出现了“汉芯”“红芯”“木兰”等一系列“技术造假”事件，在国内外科技界都造成了非常恶劣的影响，至于学术界的造假，就更不用提了；同时我们也多次看到了媒体上提到“国产自主可控”的工业软件产品，但最后都销声匿迹，在市场上激不起一点浪花。

市面上的工业软件产品历史少则十几年，多则三四十年，经历了长期的技术积累和市场打磨，同时也建立了相对稳定的生态环境。国内在缺乏人才，市场，底层核心技术的情况下，工业软件产品想要达到国外同类产品水平，一味“自主”只会付出更多代价；换言之，只凭“自主”很难达到国外同类产品水平，对外合作必不可少。

共赢合作的本质是取长补短，其核心是：底层技术点上突破，上层技术寻求合作，应用领域自主掌握，建立更加柔弹性的产业生态链。简单的以软件研发GUI组件为例，目前世界上通用的GUI组件为芬兰公司开发的QT，QT经过十几年发展，技术上已经非常成熟，成功的应用于各个行业各个平台开发，有免费版本，开源版本以及商业版本，是非常理想的开发工具。除非特殊需求，完全可以使用商业版本，没有必要自己重新再做一套GUI系

统；再以CAD内核为例，ACIS，PS都被美国公司垄断，在寻求这方面合作的同时也可以寻求其它欧洲公司比如Matra Datavision（OCC公司）商业合作；再比如CFD开源工具OpenFOAM，其历史悠久，广泛应用于研究领域，完全可以作为学习参考的工具，避免重复造轮子。

冰冻三尺非一日之寒，工业软件发展历史悠久，技术生态壁垒都较高，“国产自主”是一个长期的战略指导，在当前立足于现实，踏踏实实做好技术的同时，加强对外合作沟通必不可缺。在当前国情下，少一点“国产自主”，多一点“合作共赢”！

一篇文章了解计算机辅助工程CAE

原创 邓子平 多物理场仿真技术



之前就有过想写一篇关于CAE的科普文，缘起是看到很多以CAE为标题的文章，内容围绕某一门学科，某一个行业，或者某种仿真软件介绍，有一定的局限性。CAE涉及的领域很广，只深入某个方面讲难以有全局观，泛泛而谈又没有重点。好在写过一系列关于CAE仿真的技术和行业文章，结合笔者十多年的工业软件研发经历，希望本文给读者带来对CAE领域不同的认知和理解。

计算机辅助工程

Computer Aided Engineering，简称CAE，广义上讲，凡涉及到利用计算机解决工程问题，都可以纳入CAE的范畴。在实际应用中CAD，CAM，CFD，EDA，TCAD，CAPP，计算机辅助优化，高性能计算等所定义的范畴和CAE交叉或者紧密相连。

Wiki里对CAE解释道：

CAE在广义上用来描述计算机技术在工程中的使用，而不仅仅是工程分析。该词由SDRC的创始人1970年代后期提出，但是今天通过术语CAx和PLM（Product Lifecycle Management）更好地了解了此定义。

随着行业细分领域的发展，比如EDA，CFD发展出独立的分支，CAE的概念也在逐步弱化和抽象，现在人们通常所说的CAE主要指利用计算机仿真分析来帮助或者解决实际工程应用中的查验，设计，验证，优化等问题。在谈到CAE时，有可能是指CAD，EDA，CFD，TCAD或者结构动力分析，热分析，电磁分析，耦合分析等等其中一种或多种。用某些行业领域或者学科来等价CAE是不太准确的。狭义上的CAE，更多是指利用CAE软件进行仿真优化分析。

讨论以下几个方面：

1.CAE的地位和作用

2.CAE软件的发展

3.CAE软件的选择和使用

4.CAE国内发展和前景

5.CAE软件的核心技术和未来发展趋势

多物理场仿真技术

6. 国内CAE研发的一些想法

1. CAE的地位和作用

CAE在制造业里或者科技领域的地位可以说是举足轻重，更有四两拨千斤的作用。举三个大家都相对熟悉的例子：汽车，航空发动机，芯片。

1. 汽车

汽车在设计中有两个很重要的设计点：NVH和风阻。NVH是噪声、振动与声振粗糙度（Noise、Vibration、Harshness）的英文缩写，这是衡量汽车制造质量的一个综合性问题，能够直观反映出乘坐汽车时的舒适度，汽车公司约有20%的研发费用在解决NVH问题上。NVH问题在汽车的任何部位都可能产生，是一个综合性的问题，所以在设计之初就要考虑到。对于一款没有原型的新车，是无法得到NVH的相关数据的，只能依靠数值计算模拟，也就是利用CAE软件建模，然后仿真分析，模拟可能出现的NVH问题，再生产出原型，加以验证，修改计算机模型，再仿真，如此反复迭代，最终达到一个理想的效果。但事实上因为工艺，零件质量，加工质量，设计缺陷等客观原因，NVH问题仍然无法完全避免。

汽车高速行驶中，风产生的阻力会快速上升，增加油耗，同时对车身稳定性产生影响。在车身设计中，利用目前的商业CFD软件，能很好的模拟风阻对汽车的影响，优化车身外形，从而减少实际风洞试验的次数，降低研发成本。

2. 商业航空发动机

之所以加上商业，是因为可以通过不计成本的暴力试验，让发动机达到一个可用的水平，就类似米格25发动机，超高速飞行太多，发动机就报废了。商用最重要的还是要讲究可靠性，稳定性以及性价比。

航空发动机研制涉及气动力学、传热学、材料力学、理论力学、流体力学、断裂力学、弹性力学，机械动力学等诸多学科，是牛顿力学时代所有力学的集大成，是所有科技成果的结晶。航空发动机是气动、燃烧、传热、控制、机械传动、结构、强度、材料等多种学科或专业综合优化的结果。航空发动机包括难度极大的多个部件，各个部件在高温、高压、高转速的复杂环境下工作且相互影响很大，加之高性能、长寿命、高可靠、轻重量、隐身、经济性、安全性等要求和日益苛刻的环保性约束，已经成为一个逼近极限的综合性产品。

所以在设计发动机时，不仅需要使用CAE分析软件进行不同类型的仿真分析，还要考虑到各种苛刻的材料，空间，性能，工艺等各种约束条件。抛开制造不谈，发动机设计本身就是一个包含多学科，多优化，多尺度，多物理场的工作。**可以毫不夸张的说，离开CAE，商业航空发动机研发根本无从下手。**

3. 芯片

芯片的典型研发流程是1.架构设计2.验证 3.流片 4. 批量生产，在第1,2环节，主要依赖EDA软件和工具套件，EDA是电子设计自动化(Electronics Design Automation)的缩写，也隶属于CAE的范畴，是针对电子产品的自动设计，这里的自动是指使用计算机技术和软件算法工具，给出预定义条件下的最优核心设计。比如大规模的PCB布线，高频电磁信号仿真等工作，必须依赖相应的软件工具，无法人工进行。第3个流片环节是耗时耗钱的一项工作。流片就是设计完后试生产，由芯片代工厂小批量生产一些，供测试用。它看起来是芯片制造，但实际属于芯片设计行业，由于其行业特点，每次流片成本相当高。根据美国一家非营利性多项目晶圆服务组织CMP的服务报价，以业内裸芯（die）面积最小的处理器高通骁龙855为例（尺寸为8.48毫米×8.64毫米，面积为73.27平方毫米），用28纳米制程流片一次的标准价格近400万元人民币，芯片设计企业可以拿到约25个裸芯，平均每个16万元！并且流片并不是一次性的事，流片失败，需要修改后再次流片；流片成功，可能需要继续修改优化，二次改进后再次流片。每一次都需要至少几百万元。

以国内手机厂商小米为例，公开报道其自研芯片澎湃S2到2019年为止，其流片失败5次，而每次流片成本在3千万人民币左右。流片失败的原因很多，除去工艺和流程上的因素，最重要的是原始的设计，如果使用EDA软件哪怕能减少一次流片次数，也能节约相当的成本，用好EDA软件能发现绝大部分设计方面的问题。

从上面几个例子我们可以看出，利用CAE工具，可以在产品设计过程中简化问题，减少实际产品的迭代次数，优化整个设计流程，及早发现问题，减少试验次数，降低设计成本，完成某些实际试验测试无法进行的工作。好的CAE工具不仅提供了软件工具，更重要的是融合了对设计至关重要的行业标准和累计的经验，提高研发设计效率和成功率。

2.CAE软件的发展

很多人容易将CAD等同于AutoCAD，将CAE和ANSYS等同，这也是CAE软件领域的马太效应。关于CAE软件的发展史，是一部大鱼吃小鱼，小鱼吃虾米的软件并购史，网上有详细介绍，笔者也写了 工业仿真软件十年回顾和展望 一文，回顾了从2008年开始的行业内并购，从技术角度分析和展望了CAE软件的发展。目前主流的CAE软件都被ANSYS，达索，西门子几家大的公司把持，在细分领域也基本是几家龙头公司垄断，常年少有变化。尤其是CAE软件排名第一个ANSYS公司，早年通过原始技术积累以及后来的不断并购收购，将市面上的优秀CAE软件尽数囊括其中，形成了垄断效应。

3.CAE软件的选择和使用

目前市面上CAE软件种类繁多，用户操作很类似。基本流程就是建立或者导入几何模型，模型清理修复，设置材料，荷载，激励，边界等，然后划分网格，求解器计算，查看结果。所以掌握精通了一种CAE软件，其它软件的操作基本也能无师自通；用不同的软件进

行计算，更能体会到软件之间的设计差异和指导思想，方便做结果对比，也有助于对问题进行更深入的思考和理解。

掌握一门CAE软件操作并不困难，困难的是在深入理解数学物理模型的基础上，建立起更接近真实世界的仿真模型，比如清理对仿真无用的几何，设置更符合实际工况的边界和荷载，理解网格单元所代表的具体含义，选择合适的网格单元，区分不同网格比如四边形，六面体的优劣，根据求解类型选择合适的求解模式和方程组解法以及参数等等。每个行业都有相应的设计规范，掌握设计规范，将其与仿真结合可以充分发挥软件的验证作用，所有这些知识的获得需要常年的学习积累以及大量实际项目的历练，不太可能靠软件的学习和操作短期内完成。此外市面上有很多关于CAE软件的书籍，购买时最好选择那些不止教如何操作，更是花费较大篇幅介绍理论的书籍，做到知其然知其所以然。

商业软件为了给客户提供便利，隐藏了大量技术细节。对于资深工程师，在熟练使用软件的基础上，要尽可能的弄懂软件底层的知识，并学会使用软件提供的二次开发功能，建立工作模板，积累行业知识，提高工作效率，这就需要工程师不仅要熟悉行业知识和掌握软件操作，更需要掌握1, 2种编程语言。比如优化设计，可以自己写一些第三方优化程序，或者脚本对接辅助优化软件。

4. CAE国内发展和前景

最近十几年，国内软件业方兴未艾，尤其是在风口的互联网企业发展迅猛，以移动支付，自媒体，移动视频，社交网络，人工智能应用，5G等实际落地应用的推动，使得中国在互联网应用领域走在了世界前列。但是很遗憾，在CAE软件领域，国内市场被欧美企业产品垄断，国产几乎是一片空白，市场份额连5%都不到。关于国内CAE现状之前也有过详尽分析。

中兴，华为的断供事件，让国人第一次看到了国内高科技行业的表面繁荣，而实际大部分行业的底层核心技术仍然牢牢的掌握在欧美日本企业手中。各种技术断供，包括CAE软件断供也让国内认识到要摆脱受制于人，需要有自己的CAE软件。这对国内的CAE软件发展是个有利消息。关于断供，详见一文：

工业软件“断供”是一把“达摩克利斯剑”吗

5. CAE软件的核心技术和未来发展趋势

数学物理模型通常有两种求解方法，解析解和数值解。解析解精准而且高效，但现实中很多问题找不到解析解。CAE软件主要还是使用数值解，也就是以离散为主的数值计算方法。主流的数值方法包括有限元，有限差分，有限体积，离散元，边界元，矩量等方法。CAE软件的核心部分是求解器，求解器的功能就是使用数值计算方法求解未知自由度，有些需要求解线性方程组，有些不需要，比如显式动力，无网格方法等。主流CAE软件每年都会有更新，但是新功能并不太多，主要在外围进行改进。稳定性，可靠性是主流CAE软件的一个重要特点。在可以预见的未来，CAE软件的改进也是循序渐进小跬步似的前进。

更新的领域在无网格，CAD/CAE无缝集成，部分人工智能，并行计算，软件系统融合。CAE软件另两个核心部件CAD内核和网格，基本不会有太大变化。此外公司并购也是常态，有新技术新产品的小公司一般都会被大企业并购垄断。附录一文也多有描述。

6. 国内CAE研发的一些想法

国内在CAE研发领域比较薄弱，从几个方面说说笔者的想法

1. 人才问题。CAE软件是个跨学科的大集成，知识领域涉及到计算机图形学，几何学，数值计算，网格划分，软件架构等等，随便一个都是研发大项，因此CAE软件研发对软件架构师要求较高，国内在这方面基本很少能碰到独当一面的人才。人才缺乏是国内CAE发展的一个大瓶颈。
2. 利润问题，工业软件投入少的根本原因还是利润低下，投入产出比低，如果全民都有版权意识，法律法规和政策方面给予支持，保证CAE软件的利润和可持续性，CAE软件自然也就发展起来。
3. 加强国际合作。这也是笔者经常讲的，虽然有贸易战，科技战，但是既然坚持改革开放，那也就还是要加强对外的技术合作。凡事亲力亲为，讲国产自主成本太高，对于很多非尖端科技领域的问题，合作能解决的就没必要再造轮子。毕竟只有保证了经济利益，其它的才能往前发展。
4. 国家要重视基础学科，底层学科的建设和发展。以FEM开源软件为例，大部分都是老外写的。包括炙手可热的人工智能开源工具tensorflow, caffe, pytorch等也都是国外开发，国内主要是上层应用。很多开源工具都是作者博士读书时的作品。国内也需要加强这方面的引导，鼓励支持原创。
5. 对于国内的CAE研发公司，生态环境差是客观现实，但是也有很多好的软件都在小国，比如希腊的ansa，南非的feko，瑞典的comsol，比利时的SAMCEF等等，例子一挙一大堆。所以困难固然有，但是只要专注坚持，办法总是有的。

如果您有任何想法和建议，欢迎留言

文章已于修改

哈工大被禁用MATLAB传递出什么信号

原创 www.cae-sim.com 多物理场仿真技术



这是2020年六月份的时候写的文章，当时哈工大被禁用matlab，成为国内科技领域的热点新闻。半年之后来看，其实正如笔者之前所讲影响非常有限，一朵小浪花都算不上。反而是此类事件几乎在频频向我们敲响警钟：基础学科和数值计算软件领域，国内必须要发展起来，不要等最后被放大招的时候才去采取措施！

MATLAB历史悠久，在2002年笔者读大学时已经是数据建模的利器。MATLAB名字其含义Matrix+Laboratory头三个字母，其最初起源于美国大学教授对矩阵计算程序，经过几十年发展，已经成为涵盖数据处理，科学计算，优化算法，行业工具箱，符号处理，图形图像，AI等跨各个行业领域的计算工具。而MATLAB的Mathwork公司凭借其在行业内的深耕和积累，在2019年的利润达到10亿美金，成为仿真数值计算行业不可小觑的力量。

笔者认为此次哈工大禁用MATLAB对中国的科研技术发展利大于弊：

1. 长期以来，国内的科研底层基础研究薄弱，从大学到科研机构以及公司几乎都停留在应用层面，非常依赖各种已有的各种软件工具，而这些软件工具大部分都为国外公司开发，此次禁用也向国内机构敲响警钟：一旦这些软件工具被禁用，很多上层应用都无法进行；
2. 对国内软件公司而言，立足自身条件，进行技术积累，开发具有自主知识产权的产品，对技术进行战略投入，以后会带来丰硕回报；
3. 对于国内用户而言，尤其是高端资深工程师，要逐步放弃依赖商业软件的习惯，进行底层技术的学习和积累，学会用开源软件或者自己开发相应模块功能；
4. MATLAB属于“平民化”的工具性产品，并无太多的技术壁垒，开源软件Octave也具有MATLAB的部分功能。禁用MATLAB带来的影响皮毛都算不算，但是如果工业软件CAD尤其是CAE/EDA/CFD 等仿真软件禁用的话，对国内很多公司和科研机构无异于釜底抽

薪，会产生极大影响。这也促使这些公司机构早做准备，此次禁用事件不仅仅只是敲响警钟，而是真正意义上的严重警告！

5. 国内的开源科学计算软件几乎没有，希望此次事件能促使高校从应用型向研发型转变，促进国内开源软件的发展。

仿真软件开发工具介绍大全（1/29/2021更新）

原创 邓子平 多物理场仿真技术



笔者对仿真软件开发工具做过一系列详细介绍，本文再对其做一个简要总结，并且加入一些其它常用开发工具介绍，需要查看详细内容的可以查阅历史文章。

1. CAD开发工具：

ACIS 商用

Parasolid 商用

Opencascade (OCC)

参考平台：FreeCAD, Salome

2. 网格开发工具：

Simmetrix 商用

MeshGem 商用

VKI 商用

TetGen

NetGen

Gmsh

CGAL

MESQUITE

3. 求解器开发工具：

SuperLu

MKL

Spoolse

Pardiso

Mumps

Eigen

4. GUI开发工具：

多物理场仿真技术

QT

目前最流行的跨平台GUI开发工具

5. 显示引擎:

HOOPS 商用

VTK

OGRE

OSG

6. HPC:

OpenMPI

CUDA

多线程, 多集群

7. 优化算法:

ISight 商用

Dakota

用于仿真的参数优化设计

8. 第三方开发工具:

Boost

成熟的第三方C++库

9.持续集成系统:

Jenkins

用于保证软件质量和持续集成开发

10. AI:

Caffe

TensorFlow

11. 验证调试

MATLAB 商用

用于验证求解原型开发, 适合算法验证, 数据调试等

12. 图形识别

OpenCV

实现图像处理和计算机视觉方面的很多通用算法

多物理场仿真技术

以下是部分详细介绍：

[仿真软件开发工具介绍\(10\)--cfmesh](#)

[仿真软件开发工具介绍\(9\)--PETSc/TAO和Dakota](#)

[仿真软件开发工具介绍（8）--VKI](#)

[仿真软件开发工具介绍（7）--Parasolid/OpenCascade](#)

[仿真软件开发工具介绍（6）--MeshGems/MESQUITE](#)

[仿真软件开发工具介绍\(5\)---Simmetrix](#)

[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](#)

[仿真研发系列工具相关文章](#)

如何构建世界级的中国工业仿真软件

原创 邓子平 多物理场仿真技术

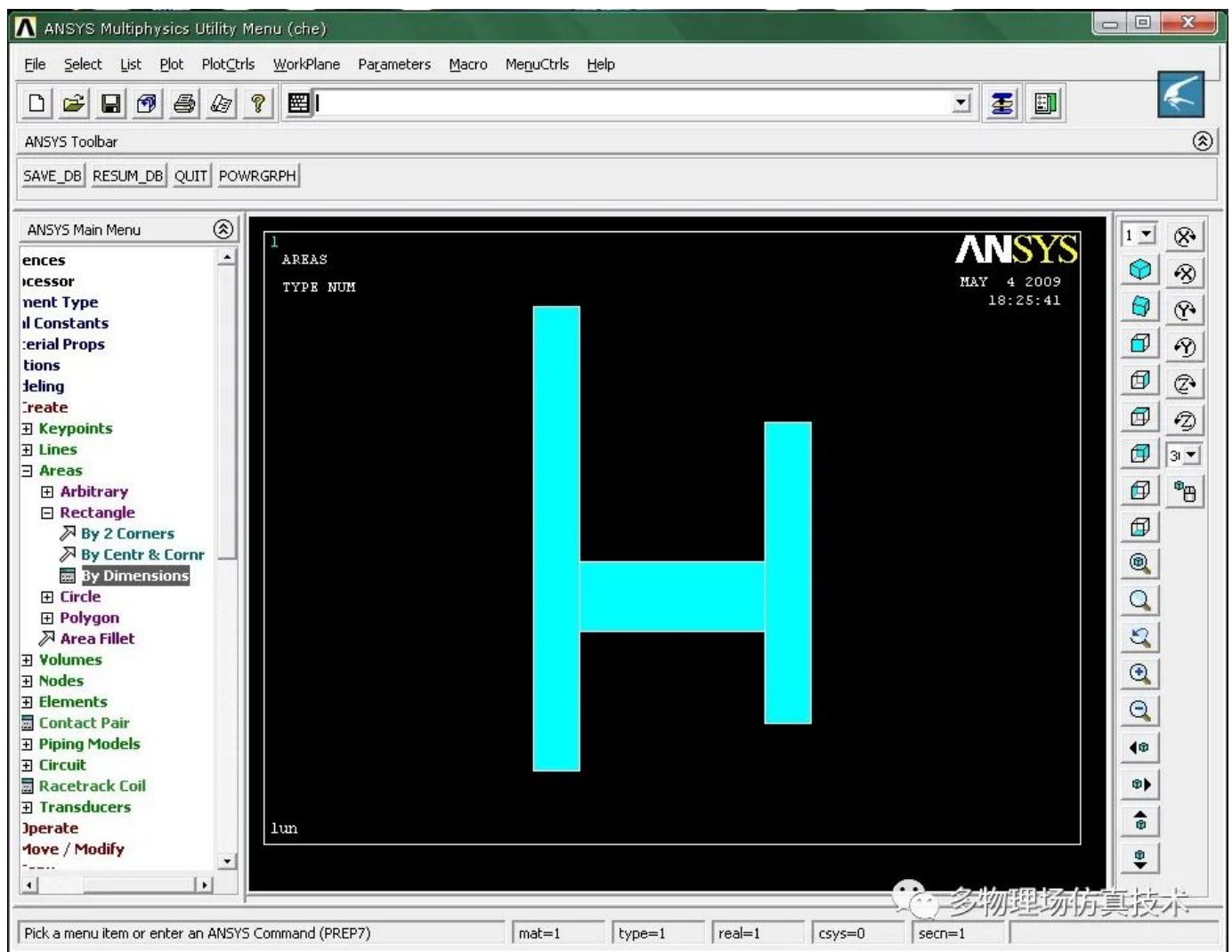


看到标题，很多人第一反应就是：谁又开始吹牛画饼了？

如果20年前有人这样写标题，笔者也会有相同的反应。

开始讲故事！！

2000年左右的时候，一套正版ANSYS的售价在国内近百万人民币。当时全国的房子均价在2000左右，一线城市在4000左右。也就是说一套ANSYS可以在上海北京买两套100平方米的房子。这个时候，很多国外产品经历了高速发展，有些刚刚进入中国，比如NASTRAN/PATRAN, ABAQUS, ADINA, LSDYNA. 有些还没出现，比如COMSOL



2010年左右时候，国内一家国企以500万人民币购买了多套CATIA软件。也就在同一年，国内某军工企业发生了SolidWorks泄密事件，当年在科技圈里闹得沸沸扬扬，具体可以百度。现在CATIA和SolidWorks都属达索旗下三维CAD产品。

2018年，西门子以45亿美金(310亿人民币)天价收购EDA公司Mentor Graphics.

2020年，ANSYS以7.7亿美金(50亿人民币)收购LS-DYNA公司。

点击链接查看最近十年公司收购：
仿真软件十年回顾和展望(整理版)

早年的工业软件外企在国内没有分公司，销售只能靠国内公司代理，这些代理公司利用国内外市场的巨大差价赚取着高额利润，在国内行业内形成了代理公司都无研发，挤破脑袋抢独代的奇葩场景；而虚假研发，空手套白狼的销售方式更是成为行业内公开的秘密，不表。

---笔者简介---

十五年工业仿真软件研发，管理，销售经历。

软件使用上：世界主流

CAD,CAE,EDA,CFD,TCAD 工业软件基本都用过,不少软件做过实际项目。

HYPERWORK,NASTRAN,ANSYS,ABAQUS,HFSS,SOLIDWORK,SPACECLAIM,AUTOCAD,INVENTOR,CST,SIMLEWARE,FLUENT,MATERIALSTUDIO,COMSOL,ADS,LSDYNA,ADAMS,ENSIGHT,ISIGHT.

开源仿真工具方面：主流则基本都摸过一遍，有些应用到实际研发项目。

OPENFOAM,CODEASTER,CALCULIX,OOFEM,IMPACT,DAKOTA,MEEP,FDTD++,FEMBio，工具性的开源软件太多，不介绍。

研发方面：涉及三维几何，图形显示，网格划分，数据交换，性能提升，多物理场求解器。基于研发经历，写了工业软件研发入门系列文章，[点击链接查看](#)。

写给仿真软件研发的“一篇文章入门”系列（终）

---给工业仿真软件打分---

和大多数考试一样，拿个及格线问题并不大，但是要再往上走困难程度就是成指数级上升了。[点击链接查看](#)

[给工业仿真软件研发打分](#)

---工业仿真软件的基础技术---

[点击链接查看](#)

[工业仿真软件的基础技术](#)

---我所理解的工业软件---

[点击链接查看](#)

[我所理解的“工业软件”](#)

---工业仿真软件需要多少投入---

[点击链接查看](#)

[开发一个ABAQUS多少钱？](#)

[开发一个ABAQUS这么多钱](#)

---工业软件人才之痛---

[痛吗？痛！缺人吗？缺！没人才吗？有！](#)

多物理场仿真技术

很多行业大厂里主力开发，乃至架构设计主导都是华人！
任正非有一句话：“钱给多了，不是人才也变人才。”
人才之痛本质上还是个商业问题，不多讲述。

---工业和软件工程---

软件架构

---工业仿真软件的未来：站在更高的起点---

仿真软件十年回顾和展望(整理版) 一文中下半篇对未来的技术发展有所展望。

---商业模式---

既然是世界级的产品，那理所当然的要卖到其它国家，靠实力占领外国市场。膜拜，跟随，模仿，竞争，超越，再熟悉不过。

南非的FEKO，英国的STARCCM，SIMPLEWARE，奥地利的AVL，希腊的ANSA，比利时的LMS，Optimus，OOFELIE，德国的CST，SIMSCALE，法国的达索，CODEASTER，加拿大的MAPLE，Presagis，MapleSim，Infolytica，韩国的MIDAS，瑞典海克斯康（收购MSC），COMSOL，还有数不清的中小厂。
印度虽然没有叫的响的产品，却是各个大厂的最大的外包地。俄罗斯和日本基本独门独户，自娱自乐。

故事讲完了，欢迎转载。

文章已于修改

一些常用软件的几何内核

邓子平 多物理场仿真技术



介绍一些常用软件的几何内核，有兴趣的可以补充

Autodesk® Inventor (ACIS)

Autodesk® Inventor Fusion (ACIS)

Autodesk® Revit (ACIS)

Solid Edge (Parasolid)

SolidWorks (Parasolid)

Unigraphics (Parasolid)

SpaceClaim (ACIS)

UGNX (Parasolid)

Solid Works (Parasolid)

CoCreate (ACIS)

HFSS (ACIS)

CST (ACIS)

FEKO (Parasolid)

CATIA (CGM)

ADINA (Parasolid)

LightTool (ACIS)

Gambit (ACIS)

InterSolid (ACIS)

COMSOL (Parasolid)

Inspire (Parasolid)

Femap (Parasolid)

CD-Adapco (Parasolid)

MSC (Parasolid)

Fluent (Parasolid)

Simmetrix (Parasolid)

ANSYS (Parasolid)

Bentley (Parasolid)
ITI (Parasolid)
IronCAD (Parasolid)
SRAC (Parasolid)

文章已于修改

工业仿真软件研发底层技术介绍系列

原创 邓子平 多物理场仿真技术



之前讲了 工业软件研发有多“难”，可能有些朋友没有get 到重点！
其实工业软件研发有其自身特点和规律：一是长期技术积累非常重要，所以需要持续投入；二是开发内容涉及面比较广，对核心研发人才要求比较高；三是用户迭代和解决实际工程问题是重要的一环，不可或缺。

只要能客观认识到这些特点和规律，用巧力且持之以恒，工业软件研发没有想象的那么困难。

工业设计软件研发是个系统工程，需要整合很多资源，公众号从2014年到现在几乎所有的原创文章都围绕这个在讲。

- 1

1.GUI2.数据 流程等管理3.多平台4.国际化
- 2

5.图形渲染6.云端支持7.高性能计算（偏硬）8.交互系统
- 3

9.几何内核，几何设计处理10.网格划分11.软件架构设计12.软件 engineering 流程
- 4

13.行业工程经验14.多物理场数值计算方法（FEM MOM FDTD FVM等）15.优化算法16.AI17.高性能计算（偏软）
- 5

18.多物理场偏微分方程理论19.大规模线性方程求解20.性能和稳定性

多物理场仿真技术

按照如上条目，1,2主要是偏上层应用的；3,4,5偏底层，也是工业仿真软件的核心内容；按照3,4,5详细分类了之前的技术文章，[后续也还将围绕该图更详尽的剖析工业仿真设计软件的一些底层技术。](#)

9.几何内核，几何设计处理

一些常用软件的几何内核

三维几何内核性能数据比较

几何相关文章

一篇文章入门三维几何内核

10. 网格划分

说说Mesh Fusion功能

深入理解数值计算网格(全篇)

一篇文章入门网格划分

11. 软件架构设计

CAE软件研发的一些思考(3)--CAE软件架构设计

CAE软件研发的一些思考(5)--系统的开发求解器

从技术角度聊聊“系统工程”

从“系统工程”角度看开发语言的选择

14. 多物理场数值计算方法

一篇文章入门多物理场有限元（全篇）

一篇文章入门边界元方法

一篇文章入门时域有限差分方法（FDTD）

CFD流体求解器算法--LBM 介绍

一篇文章入门高频电路三维有限元

15.优化算法

一篇文章入门计算机辅助优化(Computer Aided Optimization)

FEM之优化算法(5)---浅谈AI在CAE领域的应用

FEM之优化算法(4)---Isight在仿真软件开发中的应用

17. 高性能计算（偏软）

一篇文章入门HPC(高性能计算)

HPC高性能计算知识: 计算架构演进

18. 多物理场偏微分方程理论

FEM之理论通俗有限元（2） ---偏微分方程

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

FEM之理论通俗有限元(2-2)---带时间项的偏微分方程有限元解

19.大规模线性方程组求解

大规模线性方程组解法简介

（非）线性方程组求解库大全

一篇文章入门大规模线性方程组求解

20.性能和稳定性

一篇文章入门仿真软件性能优化

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

收录于话题

#工业软件杂文系列

17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。再发一次，老朋友可忽略。

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。

随笔：系统的开发工业仿真软件

原创 邓子平 多物理场仿真技术



大厂做过研发的都知道，大型软件在整个软件开发生命周期中，写代码的比例大概只占到30%-40%。有人可能会问，软件开发不就是写代码么？剩下的时间是做什么的？

这就引出本文要讨论的一个问题：如何系统的开发工业仿真软件？

再讲另外一个关于研发的真实情况，早年风靡全球的游戏魔兽争霸核心开发人员是个位数。没错，上市一年就卖了近50万份拷贝的游戏就几个人开发。

联系国内现在工业软件的风口，可能有人摩拳擦掌了：当年ANSYS，ABAQUS不也就两三个人起家的么？

问题是：时代已经变了！

之前就讲过：工业软件起始于技术，成长于用户迭代，成功于商业模式。技术-》迭代-》模式，到如今，工业软件已经融入到高端制造业中，成为其中的一部分，成为工业创新的强大推动力。

系统的开发工业仿真软件，简单讲就是需要把制造业的规范和流程搬到工业软件研发流程中，需要遵循软件研发流程和市场规律，用制度保证软件的功能和质量，而不是简单的依靠少数高产开发人员或者某一项算法和工具。根据实际情况，合理进行需求分析，文档设计，架构设计，开发人员配置，研发流程管理，任务分解，开发测试，持续集成，用户迭代等等。

千里之行始于足下，并不是说人数少不能开发工业软件，而是说研发之初就遵循工业软件的研发规律来做，才能少走弯路。

最后啰嗦一句，已经不止一次看到有人把“工业”和“软件”拆开，讨论工业软件姓“工”还是姓“软”！！还有把工业软件说成工业领域的皇冠，这种八股文式的讨论不仅毫无意义，而且容易把市场带歪。如果要讨论工业软件可以多讲讲高性能的线性方程组解法，几何内核

的底层算法，软件架构框架，高性能计算，高性能渲染，任何一个话题都是工业软件的核心。