三维结构分析并行自适应有限元软件 PHG-Solid

成 杰 张林波

(中国科学院数学与系统科学研究院科学与工程计算国家重点实验室 北京 100190)

摘 要 介绍了所研制的一个开源三维结构分析并行自适应有限元软件 PHG-Solid。它是以并行自适应有限元软件 平台 PHG 为基础开发的,支持在纯三维结构上进行并行自适应有限元分析。与现有的商业和开源结构分析有限元 软件相比,PHG-Solid 的特点和优势在于:1)支持完全自动化且高度并行的自适应有限元计算;2)能稳健高效地求解 大规模问题,具有很好的计算规模可扩展性;3)易于扩展,用户可根据需要添加相应的计算模块。通过几个大型数值 算例来展示该软件的计算能力和并行可扩展性,其中的最大计算规模超过了 5 亿自由度,最大并行规模达到了 1024 个 MPI 进程。

关键词 三维结构分析,消息传递,并行计算,自适应有限元 中图法分类号 TP301 文献标识码 A

PHG-Solid: A Parallel Adaptive FEM Software for 3D Structural Analysis

CHENG Jie ZHANG Lin-bo

(State Key Laboratory of Scientific and Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China)

Abstract We presented our newly developed open source parallel adaptive FEM software for 3D structural analysis. It is based on the 3D parallel adaptive finite element toolbox PHG. It features parallel adaptive finite element analysis for pure 3D structures. It has several advantages over other structural analysis software, Firstly, it supports fully automatic and highly parallel adaptive FEM computations. Secondly, it is robust, efficient and highly scalable in solving large scale problems. Thirdly, the software is extensible, and users can conveniently add their own modules if needed. Several large scale numerical experiments demonstrate that the largest problem size of our software exceeds 500 million degrees of freedom and the largest number of MPI process reaches 1024.

Keywords 3D structural analysis, MPI, Parallel computing, Adaptive finite element method

1 引言

计算机辅助工程(CAE)在当今航空、航天、国防及制造等 领域发挥着重要的作用。使用有限元方法进行结构分析是 CAE 的重要组成部分,也是有限元方法最早的应用领域。经 过数十年的发展,目前国内外已有大量成熟的用于结构分析 的有限元软件。著名的商业软件包括 ANSYS、Abacus 等,它 们的特点是具有强大的前后处理能力、友好的用户界面以及 大量的技术支持。此外还有大量的开源软件,如 Tahoe^[8]、 OOFEM^[7,10]等,它们基于面向对象的机制进行开发,用户可 以根据需要添加自己的模块,主要面向研究人员。目前,工业 设计、国防建设等领域对结构分析软件的计算规模及模拟精 度提出了越来越高的要求。但是,支持大规模并行并且具备 自适应计算功能的结构分析软件尚不多见,无法满足实际应 用的需求。为了方便将大规模并行计算机和先进的自适应技 术应用于复杂结构分析,我们基于三维并行自适应有限元软 件平台 PHG^[1]开发了一个新的并行自适应结构分析软件 PHG-Solid。

与其它结构分析软件相比,PHG-Solid 有若干显著的特点。首先,它支持完全自动化的并行自适应有限元计算。自适应计算过程中,初始网格被读进程序并被分发到各个进程后,进行一次有限元计算,得到数值解和后验误差估计子,根据后验误差估计子对网格进行局部加密,并对网格进行进程间重分布以保持负载平衡,接着再进行一次有限元计算,如此循环,直到达到给定的停止条件后退出。

其次,它能稳健高效地求解大规模问题。结构分析里最 主要的计算量在于线性方程组的求解。一个好的可扩展的线 性方程组求解算法是高效求解大规模问题的关键。我们可借 助于 PHG 丰富的线性解法器接口来灵活地选择线性解法 器。通过大量的数值试验表明,使用以 BoomerAMG^[11]预条 件的共轭梯度法求解线性方程组,对弹性力学问题具有很高 的计算效率和可扩展性。本文给出的计算规模 5 亿自由度、

到稿日期:2011-07-22 返修日期:2011-09-22 本文受国家 973 项目(2011CB309703),国家自然科学基金项目(60873177)和创新群体基金项 目(11021101)资助。

成 杰(1984-),男,博士生,主要研究方向为并行计算、有限元方法,E-mail; chengjie@lsec. cc. ac. cn; 张林波(1962-),男,博士,研究员,主要 研究方向为并行计算、自适应有限元方法。 并行规模达 1024 个 MPI 进程的数值实验很好地展示了这一 点。

第三,软件易于扩充。PHG-Solid 以 PHG 为基础,使用 C++和面向对象的机制进行开发。用户可根据自身的需要 添加相应的计算模块。PHG 向程序开发者隐藏了并行细节, 因此,开发者只需专注于结构分析的计算模型,而不需花大量 精力去处理消息传递的细节。

目前,PHG-Solid 支持对纯三维结构进行静力学分析、瞬态分析、大变形分析和模态分析。

本文首先介绍软件平台 PHG,然后介绍 PHG-Solid 的框架设计,最后给出若干算例的测试结果。

2 三维并行自适应有限元软件平台 PHG

PHG(Parallel Hierarchical Grid)^[1]是一个为三维自适应 有限元并行程序开发而设计的程序开发平台。它采用 C 语 言开发,基于 MPI 消息传递通信实现并行。本文介绍的结构 分析软件便是基于 PHG 研制的。本节首先介绍自适应有限 元方法,然后介绍 PHG 的功能。

2.1 自适应有限元方法

有限元方法^[4,5] 是离散偏微分方程的常用方法。它的特 点是把计算区域剖分成一个一个的"单元",通过遍历每一个 单元,结合方程的变分形式计算单元刚度矩阵,并将其组装到 总刚度矩阵,形成线性方程组。通过求解线性方程组,得到分 布于网格上的数值解。它最大的优点是能处理复杂的计算区 域并具有较完善的数学理论。

在有限元计算中,网格的质量与网格单元的分布是影响 计算结果精度和计算效率的关键因素。自适应有限元方法最 早由 I. Babuska 和 W. Rheinboldt 于 1978 年提出^[2],它通过 估计有限元计算所得到的近似解在各个单元上的误差分布, 来自动对有限元网格进行调整和优化,从而达到改善近似解 的精度的目的。近年研究表明,对于许多解具有局部奇性的 问题,基于从数学上严格推导的后验误差估计式并采用适当 的自适应策略,自适应有限元方法可以产生拟最优的有限元 网格,大幅提升有限元方法的计算效率。自适应有限元方法 的典型计算流程如图 1 所示。



图 1 自适应有限元计算流程

2.2 PHG 的功能及其主要数据结构

PHG 主要功能在于提供了许多方便用户编写并行自适 应有限元程序的数据结构和函数接口。通过这些数据结构和 函数接口,用户可以很方便地对网格进行并行剖分、动态负载 平衡和局部自适应加密,同时方便地管理自由度数据、计算刚 度矩阵、计算后验误差估计子、完成线性方程组的求解、对计 算结果进行可视化输出,等等。重要的是,这些函数接口向用 户隐藏了并行细节,方便了并行有限元程序的编写。

PHG 的主要数据结构包括:

•GRID 对象:它以分布式二叉树的形式存储由三维四 面体单元构成的网格。二叉树中的一个结点代表一个四面体 单元,所有叶子结点构成当前的网格剖分。四面体单元可进 行二分加密,加密后的新单元作为父单元的左右孩子结点。 在网格负载不平衡的情况下,单元可在进程间迁徙,以达到负 载平衡。

• DOF 对象:它存储了位于本地网格上的自由度数据。 DOF 对象包含了一个叫做 DOF_TYPE 的成员,DOF_TYPE 对象定义了基函数的类型。PHG 定义了许多函数接口来对 自由度数据进行操作。特别地,自由度数据在网格加细和放 粗的时候自动完成插值,在单元于进程间迁徙的时候自动随 之迁徙。

• MAT/VEC 对象:它们分别是分布式存储的矩阵和向 量。矩阵以行压缩的形式存储,每个进程只存储与本地自由 度对应的行和向量元素。

• SOLVER 对象:它包含了 MAT 对象和 VEC 对象作为 它的成员。它的 OEM_SOLVER 成员指向外部解法器。通 过 SOLVER 对象,可以调用许多外部解法器对线性方程组进 行求解。

• MAP 对象:它描述了向量和矩阵在各个进程上的分布 情况,同时管理了自由度数据和向量之间的映射关系。

• QUAD 对象:它辅助单元刚度矩阵和右端项计算过程 中的数值积分运算。

3 PHG-Solid 的框架设计

结构分析软件 PHG-Solid 构筑于 PHG 之上,采用了面 向对象的机制进行设计,并使用 C++语言进行开发。由于 PHG 向用户隐藏了并行方面的细节,因此只需从结构分析有 限元计算的特点出发,像设计串行软件般设计并行结构分析 软件。目前,PHG-Solid 只支持纯三维结构的分析计算,不支 持如板、壳等二维结构以及梁、柱等一维结构的计算,这给软 件的设计带来了一定的简化。

本节将结合结构分析有限元计算的特点来介绍 PHG-Solid 的设计。

3.1 结构分析有限元计算的特点

结构分析有限元计算与通常的有限元计算一样,遵循了 "网格剖分-遍历网格计算和组装矩阵-求解线性方程组"的基本模式,同时它具有自己的特点。

· 刚度矩阵的计算

结构分析刚度矩阵计算的特点是需计算应变矩阵 B 和 弹性矩阵 D。应变矩阵 B 描述了应变与位移的关系,弹性矩阵 D 描述了应力和应变的关系。刚度矩阵为 $\oint_{\alpha} B^{T} DBd\Omega$ 。

・非线性

结构分析的非线性来自两个方面:几何非线性与材料非 线性。

几何非线性意味着应变与位移呈非线性关系。它在线性 的无穷小应变-位移关系的基础上加上非线性修正项。在结 构发生大变形的时候,需把几何非线性考虑进来。描述大变 形常用的有两种体系,一种是始终在初始构型上进行计算的 "完全拉格朗日描述",另一种是始终在更新后的网格上进行 计算的"更新的拉格朗日描述"。

材料非线性指的是应力与应变呈非线性关系。一些材料 会呈现非常复杂的本构关系,例如弹塑性材料应力与应变的 关系与载荷加载的历史有关,加载载荷与释放载荷时的应力-应变关系不一致。这些材料带有一个或若干个状态变量去刻 画它的当前状态。

分析类型多样性

结构分析的分析类型是多样的,它可以是静力学分析、静 力学逐步增量载荷分析、动力学分析、模态分析等。除模态分 析外,其它的分析类型都可以统一在含时的发展型微分方程 框架下,静力学分析可作为发展型微分方程的特例。

3.2 PHG-Solid 软件架构设计

3.2.1 设计分析

从上述结构分析有限元计算的特点出发,基于面向对象 的思想对软件进行了设计。

首先,针对分析类型的多样性,设计了 Integrator 对象, 它代表了一种时间离散格式。注意到,结构分析问题中,进行 有限元空间离散后,得到形如 Mü+Cu+Ku=F 的常微分方 程组,这里 M 是质量矩阵,C 是阻尼矩阵,K 是刚度矩阵,F 是载荷向量,u 是位移向量。不同的时间离散格式的区别仅 仅在于矩阵 M、C、K 的系数不同。特别地,对静力学分析, M、C 的系数为 0,K 的系数为 1,对于模态分析,K 和 M 的系 数均为 1,其余为 0,并且需调用特征值解法器求解。

此外,把逐步增量载荷的静力学分析统一在发展型微分 方程框架下。我们抽象出 Schedule 对象,其用于描述静力学 问题中载荷的增量加载,以及动力学问题中载荷及位移边界 随时间的变化。

针对材料的多样性,设计了 Material 对象,其代表一种材料类型,如弹性材料、弹塑性材料等。它记录了材料的参数,以及材料在分析过程中的状态变量。作为计算刚度矩阵中的一步,它将计算弹性矩阵 D。

有限元计算的核心步骤是完成单元矩阵和向量的计算, 为此设计了 Element 对象。Element 对象将计算应变矩阵 B, 并结合 Material 对象中的弹性矩阵 D,最终完成刚度矩阵 K 的计算。此外,Element 对象还将结合基函数的类型计算质 量矩阵 M,结合 Material 对象计算阻尼矩阵 C,以及结合载荷 对象计算载荷向量 F。

对自由度数据进行管理是有限元计算的重要环节,为此 设计了 Fields 对象。它负责存储、更新以及输出有限元计算 的结果。

对于边界条件的处理,分别设计了位移边界对象 Boundary和载荷对象 Force。需要注意的是,我们只支持给定在面上 的位移边界和载荷,以及体载荷。换言之,我们不支持给定在 网格顶点上的位移边界和点载荷。

经由有限元离散后,需要求解线性、非线性方程组或特征 值问题。我们设计了解法器对象 ModelSolver。它指挥 Element 对象计算相应的矩阵和向量,调用 PHG 中的解法器求 解线性方程或特征值问题,并给出非线性迭代的终止条件。

最后,为自适应计算设计了 ErrorEstimator 对象。它在 Element 对象和 Fields 对象的协助下计算后验误差估计子, 并将其作为网格加密的依据。此外,它给出了自适应迭代的 终止条件。

3.2.2 C++实现

上述的若干"对象",用C++语言中的"类"对其进行实现。其中的 Material 对象、Intgrator 对象、Element 对象、 ModelSolver 对象、ErrorEstimator 对象将被设计成抽象类, 不同的子类实现代表了同一对象的不同实现。用户可根据自己的计算需求增添相应的子类实现,从而对软件进行扩充。

对上述对象进行组合、整理,得到了图2所示的设计框

架。



图 2 PHG-Solid 软件架构

我们设计了 Model 类。Model 类包含了整个问题的描述,它控制有限元计算的整个流程。同时,它是其所包含的各个子部件之间交换信息的渠道。

GRID 结构体来自 PHG,如前所述,它记录了网格单元的 信息。

把 Integrator 对象和 Schedule 对象整合到一起,设计了 TimeManager 类。TimeManager 类控制了时间步的推进。它 包含的 Integrator 对象代表了分析的类型,也代表了时间离 散的格式。Integrator 类是抽象类,现有的子类实现包括 Static 类、HHTalpha 类和 Modal 类,分别对应静力学分析、一 种隐格式的动力学分析以及模态分析。TimeManager 类还存 储了一系列 Schedule 对象,它们描述了位移边界或载荷随时 间发展的变化。

Fields 类封装了 PHG 中的 DOF 对象,用于存储自由度 数据。它还包含了 Boundary 对象,用于存储位移边界条件。 Fields 类负责为 DOF 对象设置初始条件、边界条件,并在每 一个计算步结束后更新自由度的值。这些操作是经由 Integrator 对象的辅助,在"预估-矫正"的框架下完成的。

Element 类实现了上述设计框架中的 Element 对象,它 是一个抽象类,现已实现的子类包括 SmallStrain 类和 Total-Lagrange 类,分别对应于无穷小应变单元以及用完全拉格朗 日描述的大变形单元。为保持整体设计的简洁性,Element 类把 Material 对象以及 Force 对象包含进来。

ModelSolver 类实现了解法器对象。它是个抽象类。现 有的 ModelSolver 子类实现包括 LSolver 类,对应线性解法器;NLSolver 类,对应牛顿法非线性解法器;EigenSolver 类, 对应特征值解法器。

ErrorEstimator 类实现了后验误差估计子对象。在每一个时间步完成一次有限元计算后,可通过 ErrorEstimator 类计算后验误差估计子,并据此标记网格,对网格进行局部加密。它也是个抽象类。目前实现的 ErrorEstimator 子类包括 UniformRefine 类和 ResidualEstimator 类,分别对应于一致加密及残量型后验误差估计^[3]。

3.3 程序执行流程

ł

PHG-Solid 的输入文件采用 XML 格式。程序首先从输入文件读取参数,初始化各个对象,然后调用 Model 类中的 Solve 函数完成所有计算。Model::Solve 函数描述如下。void Model::Solve(void)

InitialCondition(); / * 设置初始条件 * /

While(fTime->Step()){ /*时间步推进*/

While(TRUE){ / * 自适应迭代 * /

• 280 •

```
InitStep(); / * 设定边界条件、初始解 * /
SolveStep(); / * 完成线性/非线性/特征值问题求解 * /
CloseStep(); / * 输出结果 * /
If(StopAdaptiveLoop()) / * 判断自适应迭代终止条件 * /
break;
}
```

4 算例与测试

}

本节介绍几个采用 PHG-Solid 软件计算的大型算例。所 有算例均是在"科学与工程计算国家重点实验室"的浪潮天梭 TS10000 集群^[6]上完成的,该集群共包含 282 个计算结点,每 个结点含两颗 Intel X5550 4 核 CPU 和 24GB 内存,互联网络 为 DDR Infiniband。

4.1 Pantheon 神庙

此算例来自日本 ADVENTURE 项目^[9]。图 3 展示了 Pantheon 神庙的初始网格剖分,它包含了 1,329,027 个三维 四面体单元。它的底部边界固定,在自身重力载荷作用下发 生变形。



图 3 Pantheon 神庙初始网格

分别使用一次元、二次元和三次元对它进行了无穷小应 力-应变分析。线性解法器方面,对此类问题采用共轭梯度法 (CG)结合代数多重网格(AMG)预条件子进行求解,具体是 通过 PHG 的线性解法器接口调用开源线性解法器软件包 HYPRE^[12]中的 PCG 解法器和预条件子 BoomerAMC^[11]实 现的。在 BoomerAMG 预条件子中,粗化方法选用"falgout", 并且对应于线性方程组 3 乘 3 的子块结构,将 HYPRE_ BoomerAMGSetNumFunctions 的参数设置为 3。该算例中,通 过对网格进行逐步一致加密来测试线性解法器的可扩展性。

为考察数值解的收敛情况,比较了 Pantheon 神庙重心下 降的位移的计算结果。把采用 3 次元离散,自由度数达 210, 318,717 的结果作为参考解,观察重心位移的误差随网格加 密下降的情况。

表1、表2和表3分别展示了使用一次元、二次元和三次 元进行有限元离散的计算时间和计算结果。所有计算均使用 了128个计算节点、1024个 MPI 进程。

表1 线性元计算结果

自由度数	组装时间	迭代步数	迭代时间	重心位移	误差
819,471	0.0167s	79	34.0265s	6.361141e-02	2.07%
2,848,167	0.0614s	80	44.2940s	6.412133e-02	1.28%
8,583,672	0.1942s	90	57.6482s	6.434951e-02	0.93%
25,805,955	0.6039s	104	86.1496s	6.455344e-02	0.62%
70,694,202	1.7696s	115	133.7010s	6.469702e-02	0.40%
197,594,571	4.8968s	123	323, 1896s	6.479252e-02	0.25%
512.358,978	13. 2497s	133	757, 7582s	6.486665e - 02	0.14%

表 2 二次元计算结果

自由度数	组装时间	迭代步数	迭代时间	重心位移	误差
6,006,855	0.2029s	143	74.5276s	6.477767e-02	0.27%
20,325,024	0,7333s	123	105, 9414s	6.486437e-02	0.14%
63,666,225	2,3311s	124	170.6217s	6.490992e-02	0.07%
190,053,570	7.3645s	135	357.2976s	6.493471e-02	0.03%

表 3 二次元计算结界

自由度数	组装时间	迭代步数	迭代时间	重心位移	误差
19,549,236	2.0697s	203	150, 2033s	6.490157e-02	0.08%
66,646,398	7.3681s	150	280.9809s	6.493719e-02	0.03%
210,318,717	23, 2472s	158	732, 4989s	6,495463e-02	

可以看到,主要的计算时间花在线性方程组的求解上。 对于弹性力学方程而言,随着有限元网格的加密,离散所得之 刚度矩阵条件数为 $O(h^{-2})$,h是单元的尺寸。线性方程组迭 代算法的迭代步数随着条件数的增大而增加。因此,随着网 格的不断加密,h变小,自由度数增多,不仅单步迭代的计算 量增大,而且迭代步数也会增加。好的预条件子能使预条件 后的矩阵条件数与h无关,从而迭代法的迭代步数也与h无 关。由表1、表2和表3的结果可以看到,方程迭代步数随着 网格的加密能大致保持不变,这说明了代数多重网格预条件 子的有效性。此时,求解线性方程组的计算量与问题的规模 呈线性增长的关系,将其称之为可扩展的线性方程组求解算 法。

同时,可以看到,使用高次元能在更少的自由度下得到更 好的精度。

4.2 悬臂梁

如图 4 所示,悬臂梁左端固定,上部施加均匀向下的面载 荷。此问题的解在固定端具有奇性,以此算例展示自适应有 限元方法的网格最优性。



图 4 悬臂梁

自适应有限元方法首先在给定网格上求解,然后计算误 差指示子,以此作为网格加密的依据,遵循的原则是误差在各 单元间均匀分布,误差大的地方网格更密,从而以最少的自由 度数目达到最大的精度。

根据有限元分析的数学理论^[5],当解足够光滑时,椭圆型 问题的有限元方法在使用分片 P 次基函数时,其 H₁ 误差的 范数下降阶为 P。对于解带有奇性的问题,若使用一致加密 的网格,此下降阶是无法达到的。而自适应有限元方法能够 产生拟最优的网格,在该网格上此下降阶仍然能保持。



图 5 三次元后验误差收敛过程

(下转第 309 页)

- [14] Hellemans A, Optical R A M, IEEE-Spectrum, 2010
- [15] Helios Project[OL]. http://www.helios-project.eu/
- [16] Platon Project[OL]. http://www.ict-platon.eu/
- [17] SOFI Project[OL]. http://www.sofi-project.eu/
- [18] Kurian G, Miller J E, et al. ATAC: A 1000-Core Cache-Coherent Processor with On-Chip Optical Network [C] // Proceedings of Parallel Architectures and Compilation Techniques (PACT). September 2010
- [19] Psota J, Eastep J, et al. ATAC: On-Chip Optical Networks for Multicore Processors [R]. Technical Report, 2010. http:// groups.csail.mit.edu/carbon/docs/Psota_ATAC_BARC_1-07.pdf
- [20] Psota J, Miller J, et al. ATAC: Improving Performance and Programmability with On-Chip Optical Networks[C]// Proceedings of International Symposium on Circuits and Systems(ISCAS). Paris, 2010; 3325-3328
- [21] Biberman A, Droz N S, et al. Network-on-chip Architecture Using 3D Integration[C] // Invited Paper, In Proceeding of SPIE 2011, 2011

(上接第281页)

图 5 比较了使用三次元时一致加密和自适应加密下的后 验误差估计。后验误差估计子使用的是残量型的后验误差估 计^[3]。可以看到,为达到相同的后验误差,自适应网格所需自 由度更少。

图 6 和图 7 分别是悬臂梁问题的初始网格和自适应迭代 得到的自适应网格。自适应网格中,单元在带有奇性的固定 端的分布更加密集。

图 6 悬臂梁初始网格

图 7 悬臂梁自适应网格

4.3 神光-III 装置靶球结构

本节介绍一个模态分析算例,所计算的装置如图 8 所示, 其底座是固定的。该算例中采用二次元离散,并对网格进行 逐步一致加密,调用 ARPACK 对代数特征值问题进行求解。



图 8 光学固定装置

表 4 段态分析计算绢	爭果
-------------	----

自由度数	组装时间	计算时间	频率1	频率 2	频率 2
686,679	0.2847s	11.2629s	2.244210	2.308494	3.172591
1,361,109	0.6375s	23.2910s	2.208258	2,279572	3, 138892
3,404,370	1.7627s	69.8647s	2.168395	2.242561	3.092803
8,672,937	4.9665s	225. 3922s	2.147446	2,220872	3.070956

- [22] Xue J,Garg A, et al. An Intra-Chip Free-Space Optical Interconnect[J]. Proceedings of The 37th Annual International Symposium on Computer Architecture, 2010, 38(3)
- [23] Kash J A. Internal Optical Interconnects in Next Generation High-Performance Servers[C]// Avionics Fiber-Optics and Photonics, 2005. IEEE Conference. Yorktown Heights, NY, USA, 2005:23-30
- [24] STMicroelectronics, Inc. The Silicon Photonics Technology[R]. Technical Report. 2010
- [25] Chan J, Hendry G, et al. Phoenixsim: A Simulator for Physicallayer Analysis of Chip-Scale Photonic Interconnection Networks
 [C]// Proceedings of the Conference on Design, Automation and Test in Europe European Design and Automation Association.
 3001 Leuven, Belgium, Belgium, 2010; 691-696
- [26] Hendry G, Chan J. PhoenixSim` 1. 0 User Manual: Photonic and Electronic Network Integration and eXecution Simulator, Lightwave Research Laboratory[OL]. http://lightwave. ee. columbia. edu/downloads/phoenixsim/PhoenixSim. pdf, 2010

计算中使用了 8 个结点、64 个 MPI 进程。表 4 列出了计 算时间以及前 3 个最小特征频率。

结束语本文介绍了三维结构分析并行自适应有限元软件 PHG-Solid。它是一个支持大规模并行、并行自适应,并且 完全可扩展的结构分析软件,可用于大型复杂结构的有限元 分析。

参考文献

- Zhang Lin-bo. A parallel algorithm for adaptive local refinement of tetrahedral meshes using bisection[J]. Numer. Math. Theor. Mech. Appl., 2009, 2(1):65-89
- [2] Babuska I, Rheinboldt W. Error estimates for adaptive finite element computation[J]. SIAM J, 1978, 15:736-754
- [3] Verfurth R. A review of a posteriori error estimation techniques for elasticity problems [J]. Comput. Methods Appl. Mech. Engrg., 1999, 176(1-4): 419-440
- [4] Zienkiewicz O C, Taylor R L. The finite element method for solid and structural mechanics(6th edition)[M]. Butterworth-Heinemann,Oxford, 2005
- [5] Ciarlet P G. The finite element method for elliptic problems[M]. Amsterdam: North-Holland, 1978
- [6] Inspur Ts10000 cluster [EB/OL]. http://lsec. cc. ac. cn/chinese/lsec/docQuickstart. txt
- [7] Patzák B. OOFEM project home page [EB/OL]. http://www. oofem. org,2000
- [8] Tahoe discussion forums[EB/OL]. https://www.paklein.com/ tahoe/forums/
- [9] ADVENTRUE PROJECT[EB/OL]. http://adventure.sys.t. u-tokyo.ac.jp/
- [10] Patzák B, Bittnar Z. Design of object oriented finite element code
 [J]. Advances in Engineering Software, 2001, 32(10/11):759-767
- [11] Henson V E, Yang U M, BoomerAMG: a parallel algebraic multigrid solver and preconditioner[J]. Applied Numerical Mathematics, 2002, 41:155-177
- [12] HYPRE: High performance pre-conditioners [EB/OL]. http:// www.llnl.gov/CASC/hypre