第38卷第6期 计算力学学报 Vol. 38, No. 6
2021年12月 Chinese Journal of Computational Mechanics December 2021

DOI:10.7511/jslx20210220001

基于 PETSc 的非线性逆向运动并行计算方法研究

范志瑞[1], 阎 军*[1], 牛 斌[2], 隋倩倩[1],
许 琦[1], 蒋存存[1], 赵国忠[1]

(1.大连理工大学 工业装备结构分析国家重点实验室,工程力学系,大连 116024;
2.大连理工大学 机械工程学院,大连 116024)

摘 要:在考虑几何非线性的有限元分析中,初始构型和变形构型是严格区分的,并且变形后的构型对结构性能和功能的实现往往具有重要意义。传统的非线性有限元分析主要面向变形前的初始构型为导向的设计问题,而对于变形后构型为导向的设计问题则有较大局限性。针对此问题,引入非线性逆向运动分析方法,为了保证大变形非线性分析迭代的收敛性和计算效率,基于 PETSc 函数库建立了并行分析框架,并对并行框架中的模块划分、数据并行存储以及通信锁死和负载平衡等进行了详细阐述。在算例部分,首先通过正向运动和逆向运动分析结果对比,阐述了两种分析方法的不同以及逆向运动分析方法对变形前构型求解的准确性;其次,采用不同 MPI 进程数对并行分析程序的效率进行了测试。结果表明,合理地选择 MPI 进程数目可显著提高非线性逆向运动分析的效率。

关键词:逆向运动分析;非线性分析;有限元方法;并行计算;PETSc

中图分类号:O242.21 **文献标志码**:A **文章编号**:1007-4708(2021)06-0695-09

1 引 言

装备结构变形后整体或局部的几何形状往往对其结构性能具有重要的影响,尤其对于考虑几何大变形效应的结构设计问题。在此类问题中,与传统基于小变形假设的结构设计问题不同,变形前后的构型存在较大的差别[1,2],这使得装备结构面临设计的构型和实际在位运行的构型不一致的问题。需要说明的是,一般对于承载结构,可近似认为初始构型面向设计和制造,而变形后的构型则对应结构实际在位运行情况,即在考虑实际载荷(如环境载荷、安装载荷和工作载荷等)作用后,结构会变形为何种构型。在实际工程中,变形后的构型往往融入了设计者对特殊性能/功能的需求,如翼面负载和流体阻力作用后机翼表面仍要求保持良好的空气动力学性能[3-5],展开后卫星太阳能帆板能够确保充足的太阳能吸收[6,7],太空天线的展开形状能够使得信号接受质量最佳[8,9],以及变形后的软体机械手对抓取物体表面的贴合能力[10,11]等。因此,变形后构型的精确分析和控制对结构性能及功能的实现具有重要意义。

当采用传统有限元方法处理几何大变形问题时,通常以变形前的构型为参考,求解变形后的构型[1,12],此类分析方法即为正向运动分析方法。由于是以变形前的初始构型为结构分析的出发点,该方法很难适用于上述变形后构型的设计问题。而对于变形后构型的设计问题,更合理的方式是以变形后的构型为参考,逆向求解变形前的构型,即采用逆向运动分析方法。Schield [13]利用坐标变换的方法以及超弹性方程的对偶性建立了最早的逆向运动分析理论。Chadwick[14]将 Eshelby 能量-动量张量[15]引入了逆向运动分析理论,建立了逆向运动分析的虚功平衡方程。然而,Eshelby 能量-动量张量仍然是以初始构型为参考而定义的,其求解过程较为复杂。为此,Govindjee 等[16,17]采用了更为便捷的 Cauchy 应力来对逆向运动问题进行求解。Wallin 等[18]对逆向运动问题进行了系统推导,并建立了该类分析问题的有限元求解方法。

在考虑几何大变形/有限应变的有限元分析中,材料的刚度张量、形函数、内力载荷向量以及切

收稿日期:2021-02-20;**修改稿收到日期**:2021-06-10.
基金项目:国家自然科学基金(U1906233;11732004;51975087);国家重点研发计划(2017YFC0307203);山东省重点研发计划(2019JZZY010801);中央高校基本科研业务费专项资金(DUT20ZD213;DUT20LAB308)资助项目.
作者简介:阎 军*(1978-),男,博士,教授(E-mail:yanjun@dlut.edu.cn).

线刚度矩阵等均与结构位移场相关[1]，因此往往需要采用非线性分析的迭代法求解，并且上述物理量在各个迭代步中不断更新。当结构变形较大时，为了保证分析迭代的收敛性，需要大量的载荷增量步，从而使得非线性迭代的计算成本急剧增加。虽然修正的 Newton-Raphson 迭代法[19,20]可以采用固定的切线刚度矩阵对非线性问题进行求解，但却使得求解过程丧失了二阶收敛速度，有时甚至导致迭代过程无法收敛。

近年来，随着计算技术的不断发展，并行计算在结构有限元分析中发挥着越来越重要的作用[21,22]。根据内存架构，并行计算可分为共享式内存[23]、分布式内存[24,25]和混合式内存[26]三种。根据并行粒度，并行计算通常包含线程并行和进程并行两种。一般而言，共享式内存和线程模式适合单机并行，具有较快的内存访问速度，并且计算资源开销较小。而采用分布式内存和进程模式的并行方案，则在并行的可扩展性及大规模集群计算方面更具优势。MPI(Massage Passing Interface)[24,25]是基于分布式内存和进程模式开发的标准并行协议，在并行计算中得到了广泛应用。但是，其涉及复杂的消息通信过程以及并行协同机制，在一定程度上限制了其在数值计算方面的应用。为此，Argonne 国家实验室开发了可移植可扩展科学计算库 PETSc(Portable, Extensible Toolkit for Scientific Computation)[27]，其对消息通信过程进行了封装，并且提供了高效的线性方程组求解算法。值得一提的是，目前学者已经基于 PETSc 解决了涉及十亿级[28]，甚至百亿级[29]变量的结构优化设计问题。

2 逆向运动分析理论概述

2.1 结构变形描述

以下对逆向分析的理论进行概述，关于正向运动的非线性分析理论可参考文献[1]。在结构逆向运动分析中，变形描述与正向运动分析相反。如图1所示，在 t 时刻，结构体变形后的构型 Ω 已知，并且该构型由某一未知的初始构型 Ω_0 通过外载荷作用得到。变形前后构型的边界分别记作 $\partial\Omega_0$ 和 $\partial\Omega$。为研究基于逆向运动的结构分析方法，在此定义由变形后的构型指向初始构型的映射函数 $\boldsymbol{\phi}$，即

$$\boldsymbol{X}=\boldsymbol{\phi}(\boldsymbol{x}) \tag{1}$$

式中 $\boldsymbol{X}\in\Omega_0$ 为初始构型内部某一物质点的位置，$\boldsymbol{x}\in\Omega$ 为物质点在变形后构型的位置。

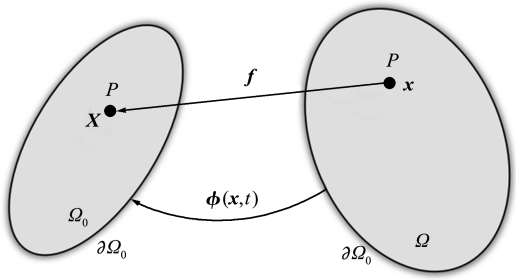


图1 逆向运动分析的结构变形描述

Fig. 1 Deformation description of inverse motion analysis

与正向运动分析类似，逆向运动分析的变形梯度张量可定义为

$$\boldsymbol{f}=\nabla\boldsymbol{\phi} \tag{2}$$

式中 ∇ 为对当前物质点位置 $\boldsymbol{x}$ 的梯度。假设初始构型与变形后构型之间满足微分同胚关系，即两者之间的映射函数连续、唯一且可微，则逆向运动与正向运动的变形梯度张量的关系为

$$\boldsymbol{f}=\boldsymbol{F}^{-1} \tag{3}$$

式中 $\boldsymbol{F}$ 为正向运动的变形梯度。为了保证映射 $\boldsymbol{\phi}$ 的唯一性，要求 $J=\det(\boldsymbol{f})>0$。此外，变形前后的应变能密度 ω_0 和 ω 之间满足

$$\omega=J\omega_0 \tag{4}$$

2.2 虚功方程及其有限元离散化

由于假设了变形前后构型之间满足微分同胚，故正向运动的平衡方程同样适用于逆向运动分析过程。当忽略结构的体力且以变形后构型为参考时，虚功方程可表述为

$$\delta\mathcal{L}(\boldsymbol{\phi},\delta\boldsymbol{\phi})=\int_{\Omega}\delta\boldsymbol{D}:\boldsymbol{\sigma}\,\mathrm{d}v-\int_{\partial\Omega}\delta\boldsymbol{\phi}\cdot\bar{\boldsymbol{t}}\,\mathrm{d}s \tag{5}$$

式中 $\forall\delta\boldsymbol{\phi}\in v$ 为结构的虚位移，$v=\{\delta\boldsymbol{\phi}|\delta\boldsymbol{\phi}=\boldsymbol{0}\ \text{on}\ \partial\Omega_{\phi}\}$ 为在位移边界 $\partial\Omega_{\phi}$ 上满足均匀 Dirichlet 边界条件 $\delta\boldsymbol{\phi}=\boldsymbol{0}$ 的所有可能虚位移的集合。$\bar{\boldsymbol{t}}$ 为施加在边界 $\partial\Omega$ 的外载荷张量，并且上述两种边界之间满足 $\partial\Omega_{\bar{t}}\cup\partial\Omega_{\phi}=\partial\Omega$ 以及 $\partial\Omega_{\bar{t}}\cap\partial\Omega_{\phi}=\varnothing$。关于式(5)具体的推导过程可参考文献[18]的工作。需要注意的是，虽然式(5)与正向运动相同，但是其中的物理量均是以变形后构型为参考的，即 Cauchy 应力张量 $\boldsymbol{\sigma}$[16]及其功共轭张量 $\boldsymbol{D}$ 可表示为

$$\boldsymbol{\sigma}=\omega\boldsymbol{I}-\boldsymbol{f}^{\mathrm{T}}\partial\omega/\partial\boldsymbol{f} \tag{6}$$

$$\delta\boldsymbol{D}=\frac{1}{2}[\nabla\delta\boldsymbol{\phi}+(\nabla\delta\boldsymbol{\phi})^{\mathrm{T}}] \tag{7}$$

式中 $\boldsymbol{I}$ 为二阶单位张量。

为了能够采用有限元方法对式(5)进行求解，对其进行 Taylor 展开，可得

$$0=\delta\mathcal{L}\approx\delta\mathcal{L}(\boldsymbol{\phi},\delta\boldsymbol{\phi})+\mathrm{d}\delta\mathcal{L}(\boldsymbol{\phi},\delta\boldsymbol{\phi}) \tag{8}$$

采用 Galerkin 方法对上式进行离散化，则有

$$\delta\mathcal{L}(\boldsymbol{\phi},\delta\boldsymbol{\phi})=\delta\boldsymbol{a}^{\mathrm{T}}\left(\int_{\Omega}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{T}\mathrm{d}v-\int_{\partial\Omega}\boldsymbol{N}^{\mathrm{T}}\bar{\boldsymbol{t}}\mathrm{d}s\right) \quad (9)$$

$$\mathrm{d}\delta\mathcal{L}(\boldsymbol{u},\delta\boldsymbol{u})=\delta\boldsymbol{a}^{\mathrm{T}}\boldsymbol{K}\mathrm{d}\boldsymbol{a} \quad (10)$$

式中 $\delta\boldsymbol{a}$ 和 $\boldsymbol{a}$ 为 $\delta\boldsymbol{\phi}$ 和 $\boldsymbol{\phi}$ 的节点值，$\boldsymbol{N}$ 为形函数矩阵，$\boldsymbol{B}$ 为位移-应变矩阵，$\boldsymbol{T}$ 为 Cauchy 应力的 Voigt 格式，$\bar{\boldsymbol{t}}$ 为外载荷向量，$\boldsymbol{K}$ 为切线刚度矩阵，可表示为

$$\boldsymbol{K}=\int_{\Omega}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{H}\mathrm{d}v \quad (11)$$

式中 $\boldsymbol{H}$ 为变形梯度的向量格式 $\boldsymbol{f}$ 与节点位移之间的变换矩阵，即 $\mathrm{d}\boldsymbol{f}=\boldsymbol{H}\mathrm{d}\boldsymbol{a}$，$\boldsymbol{C}$ 为逆向运动的刚度张量 $\mathbb{C}$ 的 Voigt 格式，且 $\mathbb{C}$ 定义为

$$\mathbb{C}=\partial\boldsymbol{\sigma}/\partial\boldsymbol{f} \quad (12)$$

根据 1st Piola-Kirchhoff 应力张量 $\boldsymbol{P}$ 之间的关系 $\boldsymbol{\sigma}=J\boldsymbol{P}\boldsymbol{F}^{\mathrm{T}}$，联合式(12)可求得 $\mathbb{C}$ 的表达式为

$$\mathbb{C}=\boldsymbol{\sigma}\otimes\boldsymbol{f}^{-\mathrm{T}}-\boldsymbol{f}^{-1}\overline{\otimes}\boldsymbol{\sigma}-\boldsymbol{\sigma}\underline{\otimes}\boldsymbol{f}^{-1}-\mathbb{d}:(\boldsymbol{I}\underline{\otimes}\boldsymbol{f}^{-1}) \quad (13)$$

式中 $\mathbb{d}$ 为正向运动中 Updated-Lagrange 迭代法[1]采用的刚度张量，即

$$\mathbb{d}=J\boldsymbol{F}\overline{\otimes}\boldsymbol{F}:\frac{4\partial\omega_0}{\partial\boldsymbol{C}\otimes\partial\boldsymbol{C}}:\boldsymbol{F}\overline{\otimes}\boldsymbol{F} \quad (14)$$

对于式(13,14)的非标准张量记号 $\overline{\otimes}$ 和 $\underline{\otimes}$ 的定义，可用二阶张量 $\boldsymbol{S}$ 和 $\boldsymbol{Q}$ 运算规则来表示，即

$$\boldsymbol{S}\overline{\otimes}\boldsymbol{Q}=\mathrm{S}_{ik}\mathrm{Q}_{jl},\ \boldsymbol{S}\underline{\otimes}\boldsymbol{Q}=\mathrm{S}_{il}\mathrm{Q}_{jk}$$

根据式(9)，定义 Newton-Raphson 迭代过程的不平衡载荷向量为

$$\boldsymbol{R}=\boldsymbol{F}_{\mathrm{int}}-\boldsymbol{F}_{\mathrm{ext}} \quad (15)$$

式中 $\boldsymbol{F}_{\mathrm{int}}$ 和 $\boldsymbol{F}_{\mathrm{ext}}$ 分别为内外载荷向量，即

$$\boldsymbol{F}_{\mathrm{int}}=\int_{\Omega}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{T}\mathrm{d}v,\ \boldsymbol{F}_{\mathrm{ext}}=\int_{\partial\Omega_{\bar{t}}}\boldsymbol{N}^{\mathrm{T}}\bar{\boldsymbol{t}}\mathrm{d}s \quad (16,17)$$

将式(10)和由式(9)推导出的式(15)代入式(8)可得，逆向运动的位移迭代可表示为

$$\boldsymbol{K}\mathrm{d}\boldsymbol{a}_{(i)}=-\boldsymbol{R},\ \boldsymbol{a}_{(i+1)}=\boldsymbol{a}_{(i)}+\mathrm{d}\boldsymbol{a}_{(i)} \quad (18,19)$$

式中 下标 i 表示第 i 次迭代。

分析式(13)可知，$\mathbb{C}$ 一般为非对称张量，因此切线刚度阵 $\boldsymbol{K}$ 也相应具有非对称性。矩阵的非对称性会使得数值求解过程无法采用对角存储方法，同时增加了预处理算子求解的困难。较之对称矩阵，非对称矩阵也会使得数值迭代过程的稳定性变差。因此，无论变量存储或是求解过程都使得逆向运动问题的求解难度显著增加。

2.3 Neo-Hookean 材料分析

本文采用 Neo-Hookean 超弹性材料模型[30,31]，该模型对于大应变情况具有很好的适用性。在逆向分析中，Neo-Hookean材料模型的应变能密度 ω 的数学表达式[18]为

$$\omega=\frac{JK}{2}\left[\frac{1}{2}(J^{-2}-1)+\ln J\right]+\frac{JG}{2}\left[J^{\frac{2}{3}}\mathrm{tr}(\boldsymbol{c}^{-1})-3\right] \quad (20)$$

式中 K 和 G 分别为材料的体模量和剪切模量，$\boldsymbol{c}=\boldsymbol{f}^{\mathrm{T}}\boldsymbol{f}$ 为逆向运动的右 Cauchy-Green 应变张量，tr()表示张量的迹。

将式(7,20)代入式(12)，可得 Cauchy 应力的表达式为

$$\boldsymbol{\sigma}=\frac{K}{2}\left(\frac{1}{J}-J\right)\boldsymbol{I}+GJ^{\frac{5}{3}}\left[\boldsymbol{c}^{-1}-\frac{1}{3}\mathrm{tr}(\boldsymbol{c}^{-1})\boldsymbol{I}\right] \quad (21)$$

联合式(3,4,14,20)可得 Updated-Lagrange 迭代法的刚度张量为

$$\mathbb{d}=a_1\boldsymbol{I}\otimes\boldsymbol{I}+a_2(\boldsymbol{I}\otimes\boldsymbol{c}^{-1}+\boldsymbol{c}^{-1}\otimes\boldsymbol{I})+a_3(\boldsymbol{I}\overline{\otimes}\boldsymbol{I}+\boldsymbol{I}\underline{\otimes}\boldsymbol{I}) \quad (22)$$

式中 $a_1=KJ^{-2}+\frac{2G}{9}J^{\frac{2}{3}}\mathrm{tr}(\boldsymbol{c}^{-1})$

$$a_2=-\frac{2G}{3}J^{\frac{2}{3}}$$

$$a_3=-\frac{K}{2}(J^{-2}-1)+\frac{G}{3}J^{-\frac{2}{3}}\mathrm{tr}(\boldsymbol{c}^{-1})$$

3 基于 PETSc 的逆向运动并行分析框架

为了提高结构非线性分析的效率，本研究基于 PETSc(Portable，Extensible Toolkit for Scientific Computation) 并行函数库[27]建立了逆向运动的并行分析程序框架。如图 2 所示，PETSc 是以 Blas/Lapack[32,33]为底层内核建立的并行计算库。同时，信息传递采用标准 MPI 协议[24]，保证了库函数良好的跨平台性以及对大规模集群计算的支持。在基础函数库的基础上，PETSc 采用面向对象思想建立了向量(Vector)、矩阵(Matrix)、索引集 IS(Index Set)、映射(Mapping)、KSP(Krylov Subspace Package)线性求解器及预处理算子 PC (Preconditioner) 等数据结构。通过对数据结构和方法(数学运算及消息传递等)的封装，为用户提供了简便易行的并行解决方案。用户无需关心具体数据结构及进程之间的消息传递模式，而将更多的精力放在顶层并行框架设计层面。因此，在保证程序稳定性的同时，为开发人员提供了灵活的程序设计空间，提高了开发的效率。

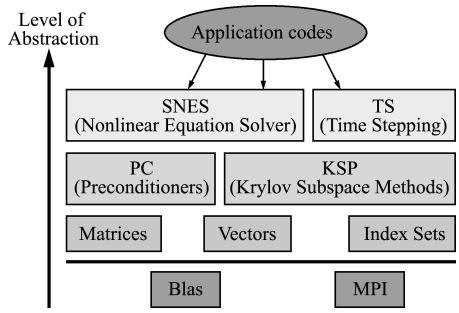


图 2 PETSc 自下而上的数据结构[27]

Fig. 2 Complete framework of PETSc library[27]

3.1 数据并行存储与交互

在 MPI 并行计算中，数据会分割成许多片段分布式存储在不同进程的内存空间中。而各个进程的内存是独立的，不能相互直接访问。进程之间的数据交换须在遵循标准 MPI 协议的情况下，采用进程通信的方式进行。

在数据交互前，需要对每个进程上涉及的单元计算任务进程合理划分。划分的规则为

$$n_i = n_b + n_{z,i},\ n_b = \mathrm{div}(n_e/m) \tag{23,24}$$

$$n_{z,i} = \begin{cases} 1 & (\mathrm{mod}(n_e/m) \geqslant i) \\ 0 & (\mathrm{mod}(n_e/m) < i) \end{cases} \tag{25}$$

式中 n_i 为当前进程 i 的单元计算量，n_b 和 $n_{z,i}$ 分别为各个进程基本计算量以及当前进程额外所需的计算量，n_e 为单元总数，m 为 MPI 进程数，div() 和 mod() 分别为整数除法的求商和取余运算。位移向量、内外力载荷向量以及标识边界条件的向量在各个进程上的分布也可以类似采用上述划分规则。对于切线刚度矩阵，可遵循上述规则按行划分为 m 个子矩阵，分别存储于相应进程的内存中。

对于单元节点编号和节点坐标，以及应力和应变场的存储需匹配单元计算的划分方式。以节点坐标为例，最终的单元坐标信息在各个进程上的划分和存储方式如下。

$$\boldsymbol{p} = [\boldsymbol{p}^{(1)}, \boldsymbol{p}^{(2)}, \cdots, \boldsymbol{p}^{(i)}, \cdots, \boldsymbol{p}^{(m)}] \tag{26}$$

$$\boldsymbol{p}^{(i)} = [\boldsymbol{p}_1^{(i)}, \boldsymbol{p}_2^{(i)}, \cdots, \boldsymbol{p}_j^{(i)}, \cdots, \boldsymbol{p}_{n_i}^{(i)}] \tag{27}$$

$$\boldsymbol{p}_j^{(i)} = [\boldsymbol{p}_{j,1}^{(i)}, \boldsymbol{p}_{j,2}^{(i)}, \cdots, \boldsymbol{p}_{j,k}^{(i)}, \cdots, \boldsymbol{p}_{j,n_p}^{(i)}] \tag{28}$$

式中 $\boldsymbol{p}$ 为最终的各个进程上单元节点坐标信息所构成的向量，$\boldsymbol{p}^{(i)}$ 为第 i 个进程上单元的坐标信息，$\boldsymbol{p}_j^{(i)}$ 为第 i 个进程上第 j 个单元的坐标信息，$\boldsymbol{p}_{j,k}^{(i)}$ 为第 i 个进程上第 j 个单元中第 k 个节点的坐标信息，n_p 为单元节点的数目。

此外，最终节点坐标向量的生成需要借助进程通信。如图 3 所示，首先从本地并行读取已生成的节点坐标二进制信息，形成初始节点坐标向量。其次，根据单元的节点编号信息生成索引集(IS)，用于确定当前单元节点坐标信息在初始节点坐标向量的位置。最后，根据索引集生成分散器(Scatter)，完成进程之间坐标信息的通信和传递。

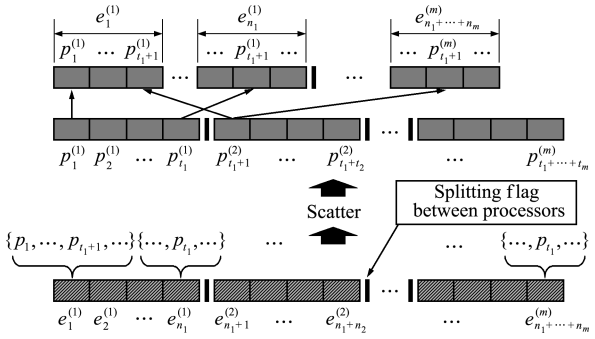


图 3 基于 PETSc 库的并行框架的数据并行存储与交互

Fig. 3 Parallel data storage and interaction in the parallel framework based on PETSc library

完成节点信息交互后，不同进程会同时拥有共有节点信息的副本。因此，虽然基于 MPI 的并行计算能够提高计算效率，但却增加了额外的内存开销。

3.2 通信锁死与负载平衡

在并行计算中，需要注意进程之间的负载平衡问题，即要保证每个进程的计算任务量大致相等。如果某些进程过早地进入下一个计算环节，很容易造成进程通信阻塞，进而引起程序卡死现象。以图 4 所示的计算过程为例，进程 1 和进程 2 均参与了单元分析及整体刚度矩阵组集过程。由于两个进程的任务量分配不均匀，进程 1 过早地进入了后续计算处理环节(如边界条件处理)。如果后续计算环节为全局操作，进程 1 需要等待进程 2 的通信响应。然而，进程 2 仍然处于之前的计算环节，且该环节中计算所得的单元刚度矩阵需要通过进程通信将其组装至全局刚度矩阵。因此，进程 2 也需要进程 1 的通信响应后才能完成相应计算。由于进程 1 和进程 2 都处于进程通信的等待状态，此时会造成进程通信锁死，从而使得计算无法继续。

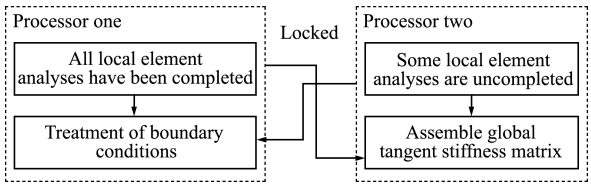


图 4 由于负载不平衡而引起的进程通信锁死

Fig. 4 Locked communication between processors caused by unbalanced tasks

针对上述问题，可选的解决方案之一是在计算任务量较少的进程上再执行一次空的单元分析，即

用各项均为零的单元刚度矩阵执行一次全局刚度矩阵的组集任务。具体地，如果 $\mathrm{mod}(n_e/n_p)\neq 0$，且 $n_z=0$ 时，需要在该进程引入空的单元分析。当然，在实际计算中应尽量保证每个进程的计算量一致，其有助于提高分析的效率，因为即使空的单元分析仍然需要花费一定的进程通信成本。

3.3　基于 PETSc 的逆向运动并行分析框架

图 5 给出了基于 PETSc 并行库所建立的非线性逆向运动并行分析程序框架。为了提高结构分析的效率，主体程序采用以 C++为基础编程语言的 PETSc 函数库编写。同时，基于 Python 语言对 ABAQUS 商业有限元软件[34]进行二次开发，搭建了完善的前后处理模块。并行程序的分析流程如下。

(1) 基于 ABAQUS 软件生成有限元计算所需的模型信息，并以二进制方式写入本地磁盘，然后采用标准 MPI 文件接口读入主分析程序。

(2) 确定各个进程上单元分析的计算量，并进一步生成用于 MPI 进程通信的索引集和分散器。

(3) 根据读入的有限元模型信息，调用结构分析模块对当前 Newton-Raphson 载荷增量步进行迭代求解。此过程需调用单元分析和材料分析模块。

(4)判断是否满足当前迭代步的收敛条件。如果不满足，则继续执行步骤(3)；否则开始新的载荷增量步。

(5) 当所有载荷增量步迭代均收敛后，将计算结果以二进制方式并行写入本地磁盘。基于 ABAQUS 软件的 Python 接口，创建 ODB(Output DatBase)文件库，对计算结果进行可视化显示。

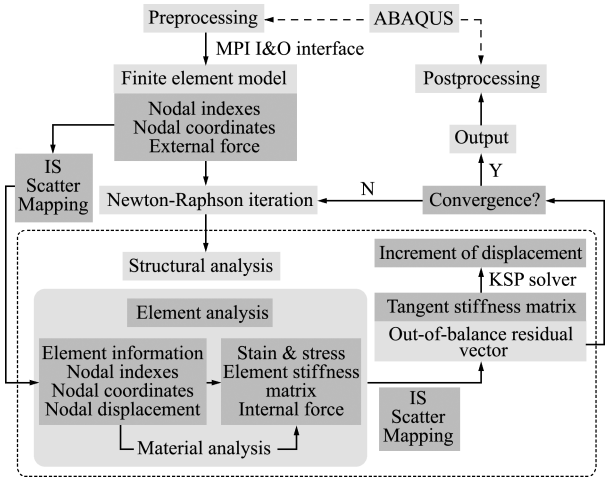


图 5　基于 PETSc 的非线性逆向运动并行分析程序框架

Fig. 5　Software framework of nonlinear inverse motion analysis based on PETSc

4　数值算例

算例 1　逆向运动分析的验证

如图 6 所示长方形板结构，其长度 $L=200$ mm，宽度 $H=50.0$ mm，厚度 $t=1.0$ mm。板的左边界固定，右端施加竖直向上和大小为 8 N/mm 的分布载荷 F。体模量 $K=1.45$ GPa，剪切模量 $G=0.81$ GPa。在有限元分析中，结构离散化为 10000 个平面单元，载荷增量步的数目设置为 200，MPI 进程数为 4。计算测试平台为 HP-Z840 工作站，操作系统为 Ubuntu 18.04 LTS，CPU 型号为 Intel® Xeon(R) CPU E5-2697 v4@2.30 GHz×72，内存大小为 256 GB，内存条型号为 32 GB DDR4-2133 ECC(LR) RAM，硬盘型号为 HP Z Turbo Drive 256 GB SSD。基于正向运动和逆向运动分析方法分别对上述结构进行分析。图 6 给出了构型分别作为正向运动的初始构型以及逆向运动的变形后构型。两种方法的分析结果如图 7 所示。

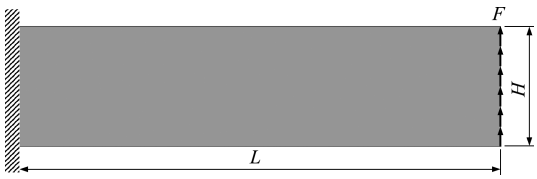
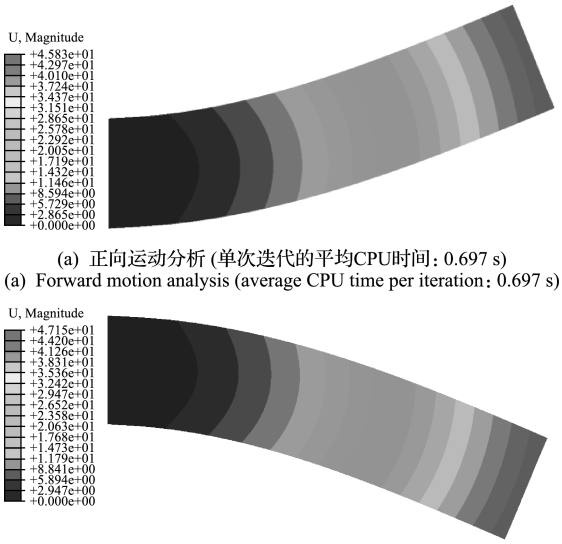


图 6　逆向运动分析方法的验证算例结构

Fig. 6　Structure used for the validation of the inverse motion analysis



(a) 正向运动分析 (单次迭代的平均CPU时间：0.697 s)
(a) Forward motion analysis (average CPU time per iteration：0.697 s)

(b) 逆向运动分析 (单次迭代的平均CPU时间：0.954 s)
(b) Inverse motion analysis (average CPU time per iteration：0.954 s)

图 7　基于正向运动和逆向运动方法计算所得图 6 结构位移场

Fig. 7　Displacement field of the structure in Fig.6 based on forward motion and inverse motion method

正向运动分析的结果如图 7(a)所示。由于正向运动是用于求解外力驱动下结构变形后的状态，

因此结构变形与载荷施加的方向一致。但是，逆向运动则与此相反，其用于求解变形前的状态。因此，如图 7(b)所示，结构变形的方向与载荷施加的方向相反。如之前所述，由于在逆向运动分析中结构的切线刚度矩阵为非对称矩阵，并且其本构分析也更为复杂。因此，逆向运动分析需要耗费更多的 CPU 计算时间。在此算例中，对于单次 Newton-Raphson 迭代，正向运动分析所花费的平均 CPU 时间为 0.697 s，而逆向运动分析则需要 0.954 s。相比于正向运动分析，逆向运动分析的 CPU 时间增加了 36.881%。此外，如果假设初始构型未知，并且需要通过正向运动分析方法使得结构变形后的构型恰好为图 6 所示构型，此分析问题对于正向运动分析方法而言是十分困难的。因为在正向运动分析中，结构信息均是以初始构型为参考的。初始构型的未知使得结构信息的定义和求解丧失了先决条件。

为了验证逆向运动分析的精度，以图 7(b)所示逆向运动分析求解所得初始构型作为分析的起点，采用正向运动分析方法求解该构型所对应的变形后构型。最终，求得的变形后构型与图 6 构型的偏差如图 8 所示。可知偏差的最大值为 3.934×10^{-2} mm。与最大位移 47.150 mm 相比，该偏差可忽略不计。因此，本研究发展的逆向运动并行分析框架和程序可以实现对变形前构型的精确求解。

图 8 逆向运动分析所得初始构型与指定初始构型间的偏差

Fig. 8 Deviation between the undeformed configuration obtained by the inverse motion analysis and the specified undeformed configuration

算例 2 并行计算效率分析

对逆向运动分析的并行效率进行研究。采用的结构如图 9 所示，方形板的长度 $L=400$ mm，宽度 $H=250$ mm，厚度 $t=1$ mm。结构左边界固定，右边界中点处施加竖直向上和大小为 5 N 的集中载荷。材料属性取值与算例 1 相同。采用逆向运动方法对结构进行有限元分析。分析中，结构离散为 $n_e=100000$ 个单元，外载荷均匀地划分为 4 个载荷增量步。选用不同的 MPI 进程数对并行效率进行测试，同时为了保证每个进程的负载平衡，MPI 进程数分别设置为 $m=\{1,2,4,5,8,10,16,20,32\}$。与算例 1 相同，仍然采用单次 Newton-Raphson 迭代的 CPU 时间来衡量并行效率。

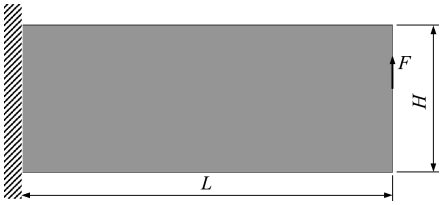


图 9 非线性并行程序效率测试结构

Fig. 9 Structure to test the efficiency of parallel nonlinear analysis program

为了更好地讨论并行的效率，引入并行加速比 η 的定义

$$\eta=T_0/[T(n)] \tag{29}$$

式中 T_0 为串行计算所需的平均 CPU 时间，$T(n)$ 为当 MPI 进程数为 n 时计算所需的平均 CPU 时间。

平均 CPU 时间随 MPI 进程数的变化如图 10 所示。可以看出，当 MPI 进程数小于 16 时，随着 MPI 进程数的增加，平均 CPU 时间快速减少。当 MPI 进程数为 1 时，平均 CPU 时间为 15.327 s；而当进程数为 16 时，平均 CPU 时间缩短至 4.903 s，其仅为单一 MPI 进程计算所用时间的 1/3 左右，且此时的并行加速比为 $\eta=3.13$。因此，采用并行策略可使得有限元分析的效率大幅度提升。其主要原因是随着 MPI 进程数的增加，每个进程上参与的有限元计算量逐渐减少，并且任务量减少的幅度小于进程通信成本增加的幅度。然而，当 MPI 进程数大于 16 时，虽然每个进程上计算量继续减少，但是进程间通信成本却在更大程度增加。因此，平均 CPU 时间逐渐上升，相应地并行加速比 η 的取值逐渐下降。综上，在针对逆向运动的实际并行分析中，可以通过试算选择合理 MPI 进程的数目，以提高有限元计算的效率。

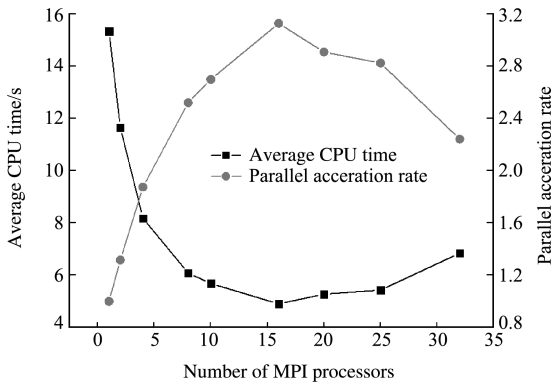


图 10 单元数目 $n_e=100000$ 时，并行效率随 MPI 进程数的变化

Fig. 10 Variation of parallel efficiency with different number of MPI processors when the number of elements is $n_e=100000$

为了验证所建立并行分析程序的可扩展性，将图 9 所示结构的网格进一步加密为 1600000 个单元。并且，分别采用 $m=\{1,5,10,16,20,32\}$ 个 MPI 进程数对结构进行逆向运动分析。并行的效率随 MPI 进程数的变化如图 11 所示。可以看出，随着进程数目的增加，平均 CPU 时间逐渐减少。相比于单进程的 1180.581 s，当 MPI 进程数为 32 时，平均 CPU 时间缩减至 395.644 s，同时并行的加速比为 $\eta=2.98$，表明建立的并行分析程序具有良好的扩展性。当结构的单元数目较多时，仍然保持较高的并行效率。

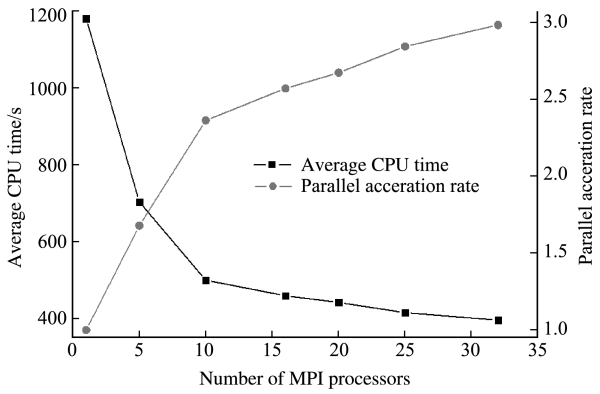


图 11　单元数目 $n_e=1600000$ 时，并行效率随 MPI 进程数的变化

Fig. 11　Variation of parallel efficiency with different number of MPI processors when the number of elements is $n_e=1600000$

5　结　论

本研究针对考虑大变形/有限应变情况下的变形后构型控制问题，引入了逆向运动分析方法，同时为了提高非线性有限元分析过程的效率，基于 PETSc 库建立了并行计算框架。对并行框架的模块划分、数据并行存储与交互以及通信锁死与负载平衡等关键问题作了详解阐述。通过算例分析，得到结论如下。

（1）逆向运动分析方法可以实现对具有大变形特征的结构变形前构型的精确求解，其误差与结构的最大位移相比很小。

（2）由于刚度张量的求解复杂性，以及切线刚度矩阵的非对称性，逆向运动分析相比于正向运动分析需要花费更多的计算成本。

（3）采用并行分析框架可大幅度提高逆向运动分析的效率。在算例 2 中，与单 MPI 进程相比，采用并行框架可节约 2/3 的计算时间，证明了本研究建立并行框架的高效性。

（4）合理选择 MPI 进程数目会使得并行效率大幅度提升。相反，选择不恰当的 MPI 进程数目，则会使得进程通信成本增加，计算效率反而下降。

参考文献（References）：

[1] Krenk S. *Non-linear Modeling and Analysis of Solids and Structures*[M]. Cambridge University Press, 2009.

[2] Bonet J, Gil A J, Wood R D. *Nonlinear Solid Mechanics for Finite Element Analysis: Statics*[M]. Cambridge University Press, 2016.

[3] Obayashi S, Takanashi S. Genetic optimization of target pressure distributions for inverse design methods [J]. AIAA *Journal*, 1996, **34**(5): 881-886.

[4] Liu G L. A new generation of inverse shape design problem in aerodynamics and aerothermoelasticity: Concepts, theory and methods[J]. *Aircraft Engineering and Aerospace Technology*, 2000, **72**(4): 334-344.

[5] 向锦武，张雪娇，赵仕伟，等. 大展弦比复合材料机翼研究进展[J]. 哈尔滨工业大学学报，2017，**49**(10)：1-14.（XIANG Jin-wu, ZHANG Xue-jiao, ZHAO Shi-wei, et al. Recent advance in high-aspect-ratio composite wing [J]. *Journal of Harbin Institute of Technology*, 2017, **49**(10): 1-14. (in Chinese)）

[6] Li Y Y, Wang Z L, Wang C, et al. Planar rigid-flexible coupling spacecraft modeling and control considering solar array deployment and joint clearance[J]. *Acta Astronautica*, 2018, **142**: 138-151.

[7] Li H Q, Liu X F, Guo S J, et al. Deployment dynamics and control of large-scale flexible solar array system with deployable mast[J]. *Advances in Space Research*, 2016, **58**(7): 1288-1302.

[8] 谭述君，侯　健，吴志刚，等. 索网天线的参变量变分及非线性有限元方法[J]. 力学学报，2014，**46**(5)：770-775.（(TAN Shu-jun, HOU Jian, WU Zhi-gang, et al. The parametric varational principle and nonlinear finite element method for analysis of astromesh antenna structures[J]. *Chinese Journal of Theoretical and Applied Mechanics*, 2014, **46**(5): 770-775. (in Chinese)）

[9] 陈传志，董家宇，陈金宝，等. 空间大型星载抛物面天线研究进展[J]. 航空学报，2021，**42**(1)：523833.（CHEN Chuan-zhi, DONG Jia-yu, CHEN Jin-bao, et al. Large spaceborne parabolic antenna: Research progress[J]. *Acta Aeronautica et Astronautica Sinica*, 2021, **42**(1): 523833. (in Chinese)）

[10] Deimel R, Brock O. A novel type of compliant and underactuated robotic hand for dexterous grasping[J].

International Journal of Robotics Research, 2016,**35**(1-3):161-185.

[11] Homberg B S, Katzschmann R K, Dogar M R, et al. Robust proprioceptive grasping with a soft robot hand[J]. *Autonomous Robots*, 2019, **43**(3):681-696.

[12] Ottosen N S, Ristinmaa M. *The Mechanics of Constitutive Modeling*[M]. Elsevier, 2005.

[13] Schield R T. Inverse deformation results in finite elasticity[J]. *Zeitschrift Für Angewandte Mathematik Und Physik ZAMP*, 1967, **18**(4):490-500.

[14] Chadwick P. Applications of an energy-momentum tensor in non-linear elastostatics[J]. *Journal of Elasticity*, 1975, **5**(3-4):249-258.

[15] Eshelby J D. The elastic energy-momentum tensor[J]. *Journal of Elasticity*, 1975, **5**(3-4):321-335.

[16] Govindjee S, Mihalic P A. Computational methods for inverse finite elastostatics[J]. *Computer Methods in Applied Mechanics and Engineering*, 1996, **136**(1-2):47-57.

[17] Govindjee S, Mihalic P A. Computational methods for inverse deformations in quasi-incompressible finite elasticity[J]. *International Journal for Numerical Methods in Engineering*, 1998, **43**(5):821-838.

[18] Wallin M, Ristinmaa M. Topology optimization utilizing inverse motion based form finding[J]. *Computer Methods in Applied Mechanics and Engineering*, 2015, **289**:316-331.

[19] Crisfield M A. A faster modified Newton-raphson iteration[J]. *Computer Methods in Applied Mechanics and Engineering*, 1979, **20**(3):267-278.

[20] Ben-Israel A. A modified Newton-raphson method for the solution of systems of equations[J]. *Israel Journal of Mathematics*, 1965, **3**(2):94-98.

[21] 陈宏博,钱雪忠,甘 霖,等.基于国产众核架构CESM中有限差分计算优化[J].计算机应用研究,2021,**38**(2):501-505.(CHEN Hong-bo, QIAN Xue-zhong, GAN Lin, et al. Parallel optimization of finite difference algorithm in CESM based on many-core processor [J]. *Application Research of Computers*, 2021, **38**(2):501-505.(in Chinese))

[22] 徐传福,车永刚,李大力,等.天河超级计算机上超大规模高精度计算流体力学并行计算研究进展[J].计算机工程与科学,2020,**42**(10):1815-1826.(XU Chuan-fu, CHE Yong-gang, LI Da-li, et al. Research progresses of large-scale parallel computing for high-order CFD on the Tianhe supercomputer[J]. *Computer Engineering & Science*, 2020, **42**(10):1815-1826.(in Chinese))

[23] Dagum L, Menon R. OpenMP: an industry standard API for shared-memory programming[J]. *IEEE Computational Science and Engineering*, 1998, **5**(1):46-55.

[24] Gabriel E, Fagg G E, Bosilca G, et al. Open MPI: Goals, concept, and design of a next generation MPI implementation[J]. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2004, **3241**:97-104.

[25] Gropp W. MPICH2: A new start for MPI implementations[A]. 9th European PVM/MPI Users' Group Meeting [C]. Linz, Austria, 2002.

[26] Yang C T, Huang C L, Lin C F. Hybrid CUDA, OpenMP, and MPI parallel programming on multicore GPU clusters [J]. *Computer Physics Communications*, 2011, **182**(1):266-269.

[27] Abhyankar S, Brown J, Constantinescu E M, et al. PETSc/TS: A Modern Scalable ODE/DAE Solver Library[J]. *arXiv*, 2018.

[28] Aage N, Andreassen E, Lazarov B S, et al. Giga-voxel computational morphogenesis for structural design [J]. *Nature*, 2017, **550**(7674):84-86.

[29] Baandrup M, Sigmund O, Polk H, et al. Closing the gap towards super-long suspension bridges using computational morphogenesis[J]. *Nature Communications*, 2020, **11**:1-7.

[30] Kim B, Lee S B, Lee J, et al. A comparison among Neo-Hookean model, Mooney-Rivlin model, and Ogden model for chloroprene rubber[J]. *International Journal of Precision Engineering and Manufacturing*, 2012, **13**(5):759-764.

[31] Wallin M, Ivarsson N, Tortorelli D. Stiffness optimization of non-linear elastic structures[J]. *Computer Methods in Applied Mechanics and Engineering*, 2018, **330**:292-307.

[32] Dongarra J J, du Croz J, Hammarling S, et al. An extended set of frotran basic linear algebra subprograms [J]. *ACM Transactions on Mathematical Software (TOMS)*, 1988, **14**(1):1-17.

[33] Dongarra J J, du Croz J, Hammarling S, et al. A set of level 3 basic linear algebra subprograms[J]. *ACM Transactions on Mathematical Software (TOMS)*, 1990, **16**(1):1-17.

[34] Smith M. *ABAQUS/Standard User's Manual, Version* 6.9[M]. Providence, RI: Dassault Systèmes Simulia Corp, 2009.

Study on parallel computation of nonlinear inverse motion method based on PETSc library

FAN Zhi-rui[1]，　YAN Jun*[1]，　NIU Bin[2]，　SUI Qian-qian[1]，
XU Qi[1]，　JIANG Cun-cun[1]，　ZHAO Guo-zhong[1]

(1. State Key Laboratory of Structural Analysis for Industrial Equipment，Department of Engineering Mechanics，Dalian University of Technology，Dalian 116024，China；
2. School of Mechanical Engineering，Dalian University of Technology，Dalian 116024，China)

Abstract：In the finite element analysis considering geometric nonlinearity，the initial configuration and deformed configuration are strictly distinguished. And the deformed configuration is of great significance to the realization of structural performances and functions. The traditional nonlinear finite element analysis is mainly employed to the design problem driven by the initial configuration，which is difficult for problems driven by the deformed configuration. Thus，nonlinear inverse motion analysis method is introduced in this study. To improve the efficiency of nonlinear inverse motion analysis，a parallel-analysis framework based on PETSc library is proposed. The corresponding sub-modulus of the framework，parallel data storage and interaction，as well the lock of communication and the remedy using load balance are presented in detail. In the numerical examples，through the comparison of the results of the forward and the inverse motion analysis，differences between the two analysis approaches and also the accuracy of form finding by inverse motion analysis are revealed. After that，the efficiency of the parallel-analysis is investigated by using different numbers of MPI processes. Numerical results show that a reasonable selection of the number of MPI processes can significantly improve the efficiency of the inverse motion analysis.

Key words：inverse motion analysis；nonlinear analysis；finite element method；parallel computation；PETSc

引用本文/Cite this paper：
范志瑞，阎　军，牛　斌，等. 基于 PETSc 的非线性逆向运动并行计算方法研究[J]. 计算力学学报，2021，**38**(6)：695-703.
FAN Zhi-rui，YAN Jun，NIU Bin，et al. Study on parallel computation of nonlinear inverse motion method based on PETSc library[J]. *Chinese Journal of Computational Mechanics*，2021，**38**(6)：695-703.