

结构非匹配网格区域分裂的并行数值计算研究

张晓虎, 孙秦

(西北工业大学 航空学院, 陕西 西安 710072)

摘要:针对非匹配多域结构有限元问题的并行求解,在L-FETI算法基础上,通过引入框架节点力及修改框架位移协调条件和载荷平衡条件,推导出了非匹配有限元撕裂合并并行算法。基于RBF插值技术,在非匹配界面上引入局部坐标系,使相邻子域间内力和位移的数据传递具有统一的矩阵格式,更易于编程。以四区域板弯有限元模型为例,分别构建匹配网格与非匹配网格两种分区模型,数值结果显示分区框架上相同节点的挠度位移具有很好的吻合度,表明提出的方法在非匹配多子域的并行求解中合理有效。

关键词:并行求解;非匹配网格;有限元撕裂合并;区域分裂;径向基插值

中图分类号:V214.1

文献标志码:A

文章编号:1000-2758(2019)04-0650-06

飞机主要采用由骨架和蒙皮组成的薄壁结构,结构复杂、尺度较大,特别是大型运输机等研制需求,使航空领域面临着日益复杂化、精细化、超大规模化等严峻挑战。计算机技术的不断革新、集群技术的快速发展,使得快速并行计算、精简存储等高效数值算法备受瞩目。有限元撕裂合并(finite element tearing and interconnecting, FETI)^[1]正是一种并行求解大规模结构的数值计算技术,因具有良好的稳定性、收敛性等特点得到不断发展,先后出现了A-FETI^[2]、FETI-DP^[3]、L-FETI^[4]等形式。

然而,实际应用中,对不同子域并行网格划分时通常会采用不同的网格生成标准或工具,且部分独立子域也会有网格加密的需要,往往就导致部分相邻子域公共界面上的网格不匹配。实现非匹配界面“粘接”的方法有很多^[5],如多点约束(multi-point constraint, MPC)、耦合有限单元(coupling finite elements, CFEs)^[6]等,插值法就是一类重要且实用的数据传递技术。一些常用插值方法有最邻近插值(nearest neighbour interpolation)^[7]、投影法(projection method)^[8]及无限平板样条、薄板样条、非均匀有理B样条^[9]等样条插值法。文献[10]对一些插值方法的精度和效率进行了研究,其中,径向基函数(radial basis function, RBF)插值方法得到普

遍关注^[11-14]。Smith等^[15]对几种常用的径向基函数进行了比较,Beckert和Wendland^[16]指出具有紧支撑的径向基函数具有更优异的特性。

为实现非匹配多域有限元问题的快速并行求解,本文将L-FETI方法中相邻子域间界面节点力细划为子域边界节点力和分区框架节点力;通过建立非匹配分区框架上的载荷平衡条件和位移协调条件,推导了非匹配多域问题中框架方程的组织形式;同时,基于RBF插值技术,在局部坐标系下实现了非匹配界面上数据的有效传递。最后,通过简单数值算例验证了本文算法的有效性。

1 非匹配多域 FETI 算法

L-FETI方法在子域界面引入三重变量:界面节点位移、界面节点力、分区框架上耦合节点位移,并基于3个子域条件和2个框架条件,得到一组近似解耦的框架方程,使得各个子域的计算相对于经典FETI算法更加独立。然而,由框架载荷平衡条件(1)式和位移协调条件(2)式可看出,L-FETI方法仅适用于匹配网格多域有限元模型的并行求解

$$\sum_{i=1}^{N_s} \mathbf{L}^{(i)}(-\boldsymbol{\lambda}_b^{(i)}) = \mathbf{0} \quad (1)$$

$$\mathbf{u}_b^{(i)} = \mathbf{B}^{(i)\text{T}} \mathbf{u}^{(i)} = \mathbf{L}^{(i)\text{T}} \mathbf{u}_f \quad (i = 1, 2, \dots, N_s) \quad (2)$$

式中, N_s 为子域总个数, $\mathbf{u}_b^{(i)}$ 和 $\boldsymbol{\lambda}_b^{(i)}$ 分别为子域 Ω_i 边界上节点位移和节点力, $\mathbf{u}^{(i)}$ 为子域 Ω_i 中全部节点位移, \mathbf{u}_f 为总框架上节点位移。 $\mathbf{B}^{(i)}$ 和 $\mathbf{L}^{(i)}$ 为由 0, 1 构成的装配算子, 表示待提取自由度必然是被提取自由度的子集, 若待提取自由度不在被提取自由度中, 则该方法将束手无策。 针对非匹配多域有限元问题的并行求解, 本文将介绍一种改进的 FETI 方法。

将模型划分成 N_s 个子域, 并在模型切割处引入虚拟框架 Γ 。 设子域 Ω_i 和 Ω_j 相邻, 记两子域之间的局部框架为 Γ_{ij} , 与子域 Ω_i 相邻的局部框架界面为 Γ_i 。 一般来说, Γ_{ij} 上节点分布与相邻子域边界 Γ_i 上节点分布无关, 但为了提高存储效率及对计算精度的考量, 当 Γ_{ij} 为非匹配界面时, 可令 Γ_{ij} 上节点分布与细网格边界节点一致。 在 Γ_{ij} 处引入四重变量: 子域边界节点位移 $\mathbf{u}_b^{(i)}$ 和节点力 $\boldsymbol{\lambda}_b^{(i)}$, 虚拟框架节点位移 $\bar{\mathbf{u}}_f^{(i)}$ 和节点力 $\bar{\boldsymbol{\lambda}}_f^{(i)}$, 如图 1 所示。

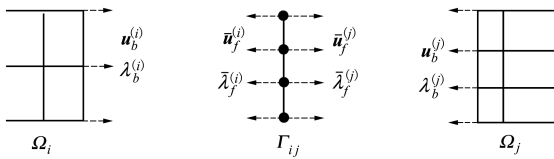


图 1 非匹配界面示意图

对于非匹配多域模型, 将框架载荷平衡条件(1)式和位移协调条件(2)式做如下修正

$$\sum_{i=1}^{N_s} [\mathbf{L}^{(i)} (-\bar{\boldsymbol{\lambda}}_f^{(i)})] = \mathbf{0} \quad (3)$$

$$\begin{cases} \mathbf{u}_b^{(i)} = \mathbf{B}^{(i)\text{T}} \mathbf{u}^{(i)}, \bar{\mathbf{u}}_f^{(i)} = \mathbf{L}^{(i)\text{T}} \bar{\mathbf{u}}_f \\ \mathbf{u}_b^{(i)} = \mathbf{G}^{(i)} \bar{\mathbf{u}}_f^{(i)} \quad (i = 1, 2, \dots, N_s) \end{cases} \quad (4)$$

式中, $\bar{\mathbf{u}}_f$ 为总框架 Γ 上节点位移, $\bar{\mathbf{u}}_f^{(i)}$ 为与子域 Ω_i 相关的局部框架 Γ_i 上节点位移, $\mathbf{G}^{(i)}$ 为局部框架 Γ_i 节点到子域 Ω_i 边界节点的位移插值矩阵, 其余与前文描述一致。 特别地, 当框架界面节点与子域边界节点匹配时 $\mathbf{G}^{(i)}$ 为单位矩阵 $\mathbf{I}^{(i)}$ 。

首先, 由子域有限元方程解得各子域位移解, 并联合子域外载条件代入(4)式, 可得框架位移协调方程

$$\mathbf{B}^{(i)\text{T}} \mathbf{F}^{(i)} \mathbf{B}^{(i)} \boldsymbol{\lambda}_b^{(i)} + \mathbf{B}^{(i)\text{T}} \mathbf{R}^{(i)} \boldsymbol{\alpha}^{(i)} - \mathbf{G}^{(i)} \mathbf{L}^{(i)\text{T}} \bar{\mathbf{u}}_f = -\mathbf{B}^{(i)\text{T}} \mathbf{F}^{(i)} \mathbf{f}^{(i)} \quad (i = 1, 2, \dots, N_s) \quad (5)$$

式中, $\mathbf{R}^{(i)}$ 为漂浮子域刚体模态阵, 可由漂浮子域节点坐标计算得到; $\boldsymbol{\alpha}^{(i)}$ 为漂浮子域刚体位移; $\mathbf{F}_{(s)}$ 为子域柔度阵, 对于约束子域来说, 其为子域刚度矩阵的逆矩阵, 对于漂浮子域来说, 其为子域广义柔度阵, 可根据文献[17] 计算得到。

其次, 根据力的传递关系, 原框架平衡方程(3)式可改写为

$$\sum_{i=1}^{N_s} [-\mathbf{L}^{(i)} \mathbf{G}^{(i)\text{T}} \boldsymbol{\lambda}_b^{(i)}] = \mathbf{0} \quad (6)$$

最后, 将(5)式、(6)式和漂浮子域自平衡条件按子域顺序装配在一起, 可得非匹配多域模型中框架方程组的矩阵表达式

$$\begin{bmatrix} \mathbf{F}_b & \mathbf{R}_b & -\mathbf{M}^{\text{T}} \\ \mathbf{R}_b^{\text{T}} & \mathbf{0} & \mathbf{0} \\ -\mathbf{M} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\lambda}_b \\ \boldsymbol{\alpha} \\ \bar{\mathbf{u}}_f \end{Bmatrix} = \begin{Bmatrix} \mathbf{g}_\lambda \\ \mathbf{g}_\alpha \\ \mathbf{0} \end{Bmatrix} \quad (7)$$

式中

$$\mathbf{F}_b = \begin{bmatrix} \mathbf{B}^{(1)\text{T}} \mathbf{F}^{(1)} \mathbf{B}^{(1)} & & \\ & \ddots & \\ & & \mathbf{B}^{(N_s)\text{T}} \mathbf{F}^{(N_s)} \mathbf{B}^{(N_s)} \end{bmatrix}$$

$$\boldsymbol{\lambda}_b = \begin{bmatrix} \boldsymbol{\lambda}_b^{(1)} \\ \vdots \\ \boldsymbol{\lambda}_b^{(N_s)} \end{bmatrix}$$

$$\mathbf{R}_b = \begin{bmatrix} \mathbf{B}^{(1)\text{T}} \mathbf{R}^{(1)} & & \\ & \vdots & \\ & & \dots & \\ & & & \mathbf{B}^{(N_f)\text{T}} \mathbf{R}^{(N_f)} \end{bmatrix}$$

$$\mathbf{g}_\alpha = \begin{bmatrix} -\mathbf{R}^{(1)\text{T}} \mathbf{f}^{(1)} \\ \vdots \\ -\mathbf{R}^{(N_f)\text{T}} \mathbf{f}^{(N_f)} \end{bmatrix}$$

$$\mathbf{g}_\lambda = \begin{bmatrix} -\mathbf{B}^{(1)\text{T}} \mathbf{F}^{(1)} \mathbf{f}^{(1)} \\ \vdots \\ -\mathbf{B}^{(N_s)\text{T}} \mathbf{F}^{(N_s)} \mathbf{f}^{(N_s)} \end{bmatrix}$$

$$\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\alpha}^{(1)} \\ \vdots \\ \boldsymbol{\alpha}^{(N_f)} \end{bmatrix}, \quad \mathbf{M}^{\text{T}} = \begin{bmatrix} \mathbf{G}^{(1)} \mathbf{L}^{(1)\text{T}} \\ \vdots \\ \mathbf{G}^{(N_s)} \mathbf{L}^{(N_s)\text{T}} \end{bmatrix}$$

本文算法可以方便地嵌入第三方软件中, 充分利用现有有限元分析软件中成熟的单元库来获得各分割子域的刚度矩阵, 进而通过框架上有序节点的位移插值来组装框架方程组。

对于(7)式所示框架方程组, 利用预处理共轭投影梯度(PCPG)等方法进行求解, 可得到各子域

能量守恒原理,可得局部框架 Γ_i 与子域 Ω_i 边界之间节点力的传递公式为

$$\bar{\lambda}_f^{(i)} = \mathbf{G}^{(i)T} \lambda_b^{(i)} \quad (13)$$

3 数值算例

以板弯模型为例,长 200 mm,宽 100 mm,厚 2 mm,材料弹性模量 $E = 70 \text{ GPa}$,泊松比 $\mu = 0.33$ 。平板一端固支,对边等距节点处各施加集中载荷 $f = 10 \text{ N}$,有限元网格采用四边形壳单元,并被分割成 4 个独立子域,如图 2 所示。各子域网格粒度及框架信息如图 3 所示,框架 1 和 3 为匹配界面,框架 2 和 4 为非匹配界面。

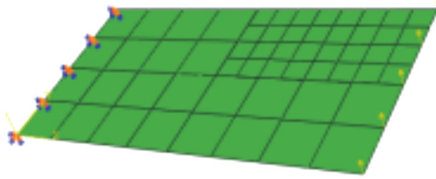


图 2 平板的位移约束和加载

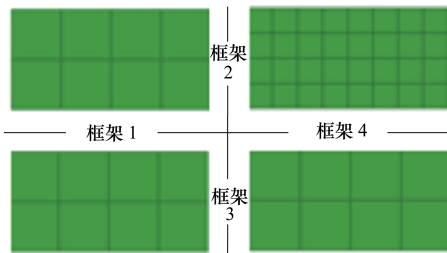


图 3 模型分区示意图

为验证计算精度,本文另外分别构建了相应粗粒度和细粒度的 2 种匹配网格模型,将 3 种模型中各自框架上节点集取并集,并对合并后节点进行统一编号,如图 4 所示,其中含有‘○’,‘●’和‘□’分别表示全粗网格、全细网格和非匹配网格中框架节点。从三者的数值结果中提取出各自框架上节点的数值解进行对比,各模型框架节点 y 向位移如图 5 所示。表 1 为非匹配框架界面 2 和 4 上各节点 y 向数值解,表 2 为匹配框架界面 1 和 3 上各节点 y 向数值解。

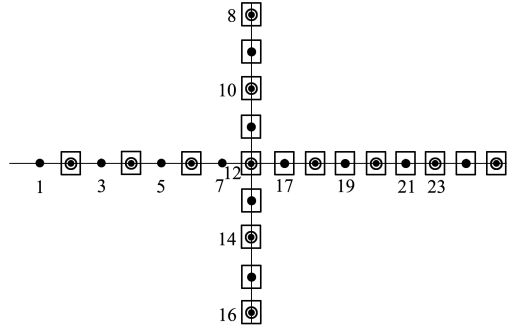


图 4 粗、细和非匹配网格中框架节点集一致编号

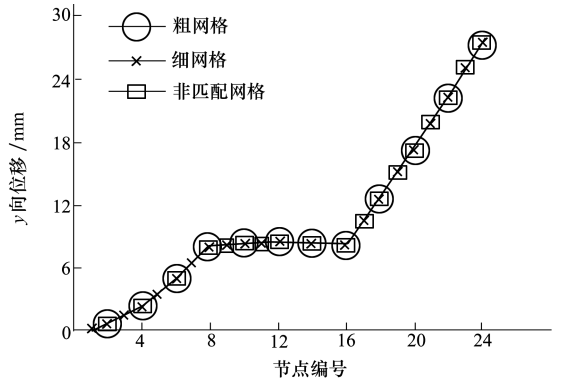


图 5 各粒度网格模型中框架节点 y 向位移

表 1 非匹配框架界面 2 和 4 上各节点 y 向位移值

节点编号	v_f/mm	v_n/mm	$(v_n - v_f)/v_f$
8	8.111 637	8.042 956	-0.008 467
9	8.275 226	8.189 760	-0.010 328
10	8.387 692	8.349 014	-0.004 611
11	8.453 278	8.415 192	-0.004 505
12	8.474 808	8.493 460	0.002 201
17	10.450 394	10.529 226	0.007 544
18	12.568 672	12.706 334	0.010 953
19	14.810 221	15.014 563	0.013 797
20	17.155 376	17.425 888	0.015 768
21	19.584 348	19.922 116	0.017 247
22	22.077 298	22.481 122	0.018 291
23	24.614 392	25.086 747	0.019 190
24	27.176 465	27.713 325	0.019 755

表 2 匹配框架界面 1 和 3 上各节点 y 向位移值

节点编号	v_c/mm	v_n/mm	$(v_n - v_c)/v_c$
2	0.646 444	0.646 355	-0.000 137
4	2.387 474	2.387 176	-0.000 125
6	5.045 638	5.045 121	-0.000 102
12	8.492 431	8.493 460	0.000 121
14	8.402 884	8.456 898	0.006 428
16	8.121 090	8.193 944	0.008 971

由图 5 可见,粗网格、细网格和非匹配网格模型中框架各节点 y 向位移解基本一致,表明本文提出的在虚拟分区框架上引入独立节点位移和节点力的策略是完全可行的。

在表 1~2 中, v_f, v_c 和 v_n 分别表示细粒度、粗粒度和非匹配网格模型中相应节点 y 向位移。观察发现,在非匹配框架界面 2 上,从多子域交点(12 号节点)到靠近模型自由边界节点(8 号节点)误差越来越大,在非匹配框架界面 4 上也一样,因此,分割边界上靠近模型自由边界处的节点误差是重要关注点。

由表 1~2 可以看出,在整个分区框架上最大误差在 24 号节点处,为 1.975%。利用商业分析软件

Abaqus 的绑定功能,采用同样的壳单元对此 3 种粒度有限元模型分别进行数值仿真求解,经相同的分析和对比方法得出框架节点 y 向位移的最大误差也在 24 号节点,为 3.951%,表明本文提出的算法有效。

4 结 论

以上计算分析表明了本文提出方法的可行性,现将主要工作归结如下,

1) 在 L-FETI 区域分裂算法的基础上,引入独立的框架节点力,使子域边界和分区框架上各自的节点力和节点位移充分局部化,修改了框架位移协调条件和载荷平衡条件,推导出了非匹配多域问题中有限元撕裂合并的并行求解算法。

2) 以 RBF 插值技术为基础,在非匹配界面(线或面)上引入局部坐标系,使任意切割方向而界面形式(线或面)相同的分区边界上的传递矩阵具有统一格式,避免了 RBF 插值中为防止插值矩阵奇异而带来的坐标选取难题,便于编程处理。

3) 在四节点机群环境中完成四区域非匹配板弯模型的并行求解,数值结果表明本文提出的非匹配多域并行处理技术具有良好的精度。

参考文献:

- [1] FARHAT C, ROUX F X. A Method of Finite Element Tearing and Interconnecting and Its Parallel Solution Algorithm[J]. International Journal for Numerical Methods in Engineering, 1991, 32(6): 1205-1227
- [2] PARK K C, JUSTINO M R, FELIPPA C A. An Algebraically Partitioned FETI Method for Parallel Structural Analysis: Algorithm Description[J]. International Journal for Numerical Methods in Engineering, 2015, 40(15): 2717-2737
- [3] FARHAT C, LESOINNE M, LETALLEC P. FETI-DP: a Dual-Primal Unified FETI Method-Part I: a Faster Alternative to the Two-Level FETI Method[J]. International Journal for Numerical Methods in Engineering, 2001, 50(7): 1523-1544
- [4] 李斌, 杨智春. 一种充分局部化的子结构并行分析方法[J]. 强度与环境, 2007, 34(4): 1-7
LI Bin, YANG Zhichun. A Localized FETI Method for Structural Parallel Analysis[J]. Structure & Environment Engineering, 2007, 34(4): 1-7 (in Chinese)
- [5] HAIKAL G, HJELMSTAD K D. An Enriched Discontinuous Galerkin Formulation for The Coupling of Non-Conforming Meshes [J]. Finite Elements in Analysis & Design, 2010, 46(6): 496-503
- [6] JR L A G B, MANZOLI O L, PRAZERES P G C, et al. A Coupling Technique for Non-Matching Finite Element Meshes[J]. Computer Methods in Applied Mechanics & Engineering, 2015, 290: 19-44
- [7] THÉVENAZ P, BLU T, UNSER M. Interpolation Revisited[J]. IEEE Trans on Medical Imaging, 2000, 19(7): 739-758
- [8] CEBRAL J, LOHNER R. Conservative Load Projection and Tracking for Fluid-Structure Problems[J]. AIAA Journal, 1997, 35(4): 687-692
- [9] HOSTERS N, HELMIG J, STAVREV A, et al. Fluid-Structure Interaction with NURBS-Based Coupling[J]. Computer Methods in Applied Mechanics & Engineering, 2018, 332: 520-539

- [10] BOER A D, ZUIJLEN A H V, BIJL H. Review of Coupling Methods for Non-Matching Meshes[J]. *Computer Methods in Applied Mechanics & Engineering*, 2007, 196(8): 1515-1525
- [11] 苏波, 石启印, 钱若军. 径向基函数应用于流固耦合分析初探[J]. *工程力学*, 2013, 30(1): 59-63
SU Bo, SHI Qiyin, QIAN Ruojun. Preliminary Study on the Use of Radial Basis Function in Fluid-Structure Interaction Analysis [J]. *Engineering Mechanics*, 2013, 30(1): 59-63 (in Chinese)
- [12] BOER A D, ZUIJLEN A H V, BIJL H. Radial Basis Functions for Interface Interpolation and Mesh Deformation[J]. *Lecture Notes in Electrical Engineering*, 2009, 71: 143-178
- [13] COSTIN W J, ALLEN C B. Numerical Study of Radial Basis Function Interpolation for Data Transfer Across Discontinuous Mesh Interfaces [J]. *International Journal for Numerical Methods in Fluids*, 2013, 72(10): 1076-1095
- [14] WANG L. Radial Basis Functions Methods for Boundary Value Problems: Performance Comparison[J]. *Engineering Analysis with Boundary Elements*, 2017, 84: 191-205
- [15] SMITH M J, CESNIK C E S, HODGES D H. Evaluation of Some Data Transfer Algorithms for Noncontiguous Meshes[J]. *Journal of Aerospace Engineering*, 2000, 13(2): 52-58
- [16] BECKERT A, WENDLAND H. Multivariate Interpolation for Fluid-Structure-Interaction Problems Using Radial Basis Functions [J]. *Aerospace Science and Technology*, 2001, 5(2): 125-134
- [17] FELIPPA C A, PARK K C, FIHO M R J. The Construction of Free-Free Flexibility Matrices as Generalized Stiffness Inverses [J]. *Computers and Structures*, 1998, 68(4): 411-418

Study on Parallel Numerical Calculation of Structural Domain Decomposition with Non-Matching Meshes

ZHANG Xiaohu, SUN Qin

(School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: For the parallel solution of structural finite element problem with non-matching multi-domains, a non-matched finite element tearing and interconnection (FETI) parallel algorithm is proposed based on L-FETI method by introducing the frame node force and modifying the frame displacement compatibility condition and the load balance condition. Based on the radial basis functions (RBF) interpolation technique, the data transfer of internal force and displacement between the neighboring subdomains has a unified matrix format and is easy to program by introducing a local coordinate system into the non-matching interfaces. Taking the finite element model for plate bending problem with four subdomains as an example, two kinds of multi-domains models, matching and non-matching meshes, are constructed respectively. The numerical results show that the out-of-plane deflection of the same nodes on the domain decomposition frame is in a good agreement, which shows that the present method is reasonable and effective for solving the parallel non-matching multi-domains model.

Keywords: parallel solution; non-matching meshes; FETI algorithm; domain decomposition; RBF interpolation