文章编号：0253-374X(2015)01-0126-07
DOI：10.11908/j.issn.0253-374x.2015.01.019

大规模向量式有限元行为数据无损压缩模型

杜庆峰[1]，周雪非[1]，谢 涛[2]，周晓玮[1]
(1. 同济大学 软件学院，上海 201804；2. 圣迭戈州立大学 计算机科学学院，圣迭戈 92182)

摘要：针对向量式有限元产生的海量的高精度行为数据在基于 web 的体系架构的数据传输和文件读取的瓶颈，依据向量式有限元行为数据的内部联系和冗余特点，建立了行为数据的数学模型，并对冗余数据进行删减和合并，提出一种有效的无损压缩模型. 最后，给出基于模型的压缩算法. 经过验证，数据的规模显著减小，极大提升了数据处理的效率. 解压时则完全还原原始数据，完全不影响行为数据的精度.

关键词：向量式有限元；行为数据；无损压缩
中图分类号：TN919 **文献标志码**：A

Massive Vector Form Intrinsic Finite Element Behavior Data Lossless Compression Model

DU Qingfeng[1], *ZHOU Xuefei*[1], *XIE Tao*[2], *ZHOU Xiaowei*[1]
(1. College of Software Engineering, Tongji University, Shanghai 201804, China; 2. Department of Computer Science, San Diego State University, San Diego 92182, USA)

Abstract: The behavior data of vector form intrinsic finite element computing has a large size as well as high accuracy, which become the bottleneck of data transmission and read that based on web. In order to resolve the problem. According to the characteristics of redundancy and the internal relations of behavior data, a mathematic model for the behavior data is proposed and according to the model, a lossless compression model is put forward by deleting and merging the redundancy data. In the end, an efficient compression algorithm is given based on the model. Experiment indicates that the algorithm greatly compresses size of behavior data and accelerates the load speed of data. The decompressed file can be restored to the original data and the accuracy of behavior data doesn't be affected at all.

Key words: vector form intrinsic finite element; behavior data; lossless compression

向量式有限元技术解决了传统结构力学理论无法解决的结构体断裂、大变形等非线性问题，在行为分析领域有着良好的应用前景。行为分析分为前处理、中处理、后处理 3 个步骤。后处理生成的行为数据往往是海量的、非结构化的，成为应用于基于 web 的体系结构的云端上传及在线处理的瓶颈，传统的压缩算法并不适用于向量式有限元行为数据。本文从分析向量式有限元行为数据关系出发，通过对行为数据进行数学建模分析，找出数据间冗余的特性，提出一种针对向量式有限元行为数据的无损压缩模型。

1 研究背景及问题的提出

结构力学理论在进行行为分析的主要目的是：针对某个结构的每个构件将会发生的最大应力和几何变化，评估结构的安全性，并进一步决定可能需要进行的补强策略.

2002 年美国普渡大学的丁承先教授等提出了向量式有限元[1](vector form intrinsic finite element, VFIFE)理论，是结构行为分析的一个创新概念，解决了大型结构在受到外力之后产生的空间运动如大变位、大变形以及材料变化、破坏、断裂和崩塌等力学行为的行为分析问题[2-5]. 在大变形、断裂、碰撞等工程问题的研究中具有很好的应用前景.

前期建立模型之后，加入材料、受力、初速度等接口，再通过向量式有限元方法进行计算，得到向量式有限元后处理行为数据. 通过对数据进行读取并进行可视化处理可得到动态展示模型在不同的操作情况下产生的行为的三维动态展示.

向量式有限元后处理行为数据往往是海量的(一般大于 100 M 或大于 100 万行)，导致了 VFIFE 方法应用于基于 web 的体系架构的数据传输和文件

收稿日期：2013-10-23
基金项目：国家自然科学基金(41171303)
第一作者：杜庆峰(1968—)，男，教授，博士生导师，工学博士，主要研究方向为软件测试、计算机相关学科交叉领域. E-mail：du_cloud@tongji.edu.cn
通讯作者：周雪非(1990—)，男，硕士生，主要研究方向为计算机相关学科交叉领域. E-mail：snowelf@126.com

读取的效率低下,并耗费大量宝贵存储资源,成为后处理的一大瓶颈. 因此,对后处理行为数据进行压缩,使上百兆的文件能够在几秒之内完成上传和读取. 对于提升 VFIFE 方法在基于 web 的体系架构的数据传输和文件读取的效率具有重要意义.

2　相关工作

前人在数据压缩领域有许多研究成果,例如:对数据进行熵编码[6-7]. 以算术编码[8]为代表,它是一种常用的熵编码算法. 由于算术编码效率较低, 近年来出现了很多提高算术编码效率和压缩性能的变种, 如双层算术编码[9]等.

近年来的研究热点是[10]将三维模型视为点云[11],其压缩过程分为 3 个步骤:量化,几何编码和熵编码. 几何编码对数据进行量化、预测,使之更好地为熵编码所用. 近年的研究成果,例如:2004 年 Waschbüsch 等[12]提出的多分辨率压缩算法以及 2005 年 Chen 等[13]提出的利用 k-d 树对顶点数组分组和搜索并生成校正值的最小生成树等,对向量式有限元数据并不适用,因为它们不是针对向量式有限元行为分析数据的,并且其算法的实现往往比较复杂.

向量式有限元行为数据具有数据量巨大、精度高的特点,是后处理行为分析的直接依据,其精度直接关系到后处理过程能否真实准确地再现三维立体模型受到外力之后发生的大变位、大变形和碰撞、断裂等行为,因此本文主要考虑无损压缩. 首先,分析海量行为数据的特点,通过对数据进行建模,给出相关定义并对数据间关系进行分析,进而提出了针对行为数据的几何数据集和拓扑数据集的压缩模型,最后,对压缩模型予以算法实现并进行了算法验证.

经验证,算法极大地减少了数据的规模,提升了数据的读取和处理效率,解决了 VFIFE 方法在基于 web 的体系架构的数据传输和文件读取效率的瓶颈.

3　压缩及解压缩模型思想

3.1　向量式有限元后处理行为数据特点分析及无损压缩相关定义

3.1.1　行为数据类型及特点

行为数据是一类非常复杂的非结构化的数据,一般涉及到不同类型的数据文件,这些数据文件内部的结构没有特定的类型,仅仅用科学计数法进行表示,且数据量十分庞大,数据内部存在大量的冗余.

根据向量式有限元方法,前处理建立的模型经过程式计算后将会生成 2 类行为分析数据,一类是动态质点的几何位置数据,另一类是质点拓扑结构数据. 两者皆带有时间特性,质点的几何位置数据反映了质点的位置坐标变化过程,质点拓扑结构数据反映了不同时刻质点的连接关系. 2 类数据缺一不可. 下面在纷繁复杂的数据中找到它们的共性和规律.

(1) 数据表现形式. 无论是质点位置数据还是质点拓扑结构数据都是用 24 位科学计数法表示的. 例如:－1.9721501277542906e－004,其中左数第 1 位为符号位,底数精度为 16 位,指数有正负之分,其精度为 3 位.

(2) 数据内部关系和冗余特性. 质点数据和拓扑数据彼此相互依存,质点数据文件共分 5 列,从左到右依次代表时刻、节点编号、节点 x,y,z 坐标值,例如:

1.9950000000000002e－003	4.0000000000000000e＋000	2.1818200799999996e＋000	2.5656499999999999e－001	－2.3272699999999999e－001
1.9950000000000002e－003	5.0000000000000000e＋000	－2.1818167080893844e＋000	－2.8658814451418940e－001	－8.1818618919650532e－002
1.9950000000000002e－003	6.0000000000000000e＋000	2.1818195802275571e＋000	－2.8658814441576036e－001	－8.1818619531342618e－002

质点的位置坐标的变化反映了模型位移、变形、断裂的过程,因为质点的运动轨迹是连续的,故而质点在相邻的时刻的位置坐标是差别不大甚至是相同的,另外对于对称的模型,大量的对称点的坐标只是符号不同,而数值则相同. 质点运动的连续性和质点的对称性使得相同时刻及不同时刻质点的坐标存在着大量相同数据,因而质点数据存在着大量的冗余.

拓扑数据描述连接节点的杆件,共分 4 列,从左到右依次代表时间、杆件编号、杆件两端的节点编号. 例如:

1.5900000000000001e－002	5.8000000000000000e＋001	4.0000000000000000e＋000	1.7000000000000000e＋001
1.5900000000000001e－002	5.7000000000000000e＋001	4.0000000000000000e＋000	3.0000000000000000e＋001

拓扑数据反映了不同时刻质点的连接. 因此,2 类数据缺一不可. 另外,不同时刻的质点连接情况当且仅当模型发生断裂才会改变. 因此,拓扑数据中也存在大量的冗余数据.

3.1.2　行为数据无损压缩相关定义

定义 1　向量式有限元行为分析几何数据集

(以下简称几何数据集)假定存在集合 F(其中 $\forall f \in F$ 代表模型不同的时刻)和集合 ID(其中 $\forall \mathrm{id} \in \mathrm{ID}$ 代表模型质点的编号),以及集合 L_x, L_y, L_z(其中 $\forall l_x \in L_x, \forall l_y \in L_y, \forall l_z \in L_z$ 分别代表模型质点的 x, y, z 坐标),且满足以下性质:

(1) 对 $\forall f_i \in F$, f_i 与全体 $\mathrm{id} \in \mathrm{ID}$ 对应,且 $f_{i+1} = f_i + t$(其中 t 为常数且 $i > 1$). 即集合 F 与集合 ID 是 $1-n$ 关系(其中 i 为整数且 $i \in (1, m)$, m 为 F 集合中元素的个数)且 F 集合内元素 f_i 呈线性增长规律.

(2) 对 $\forall \mathrm{id}_j \in \mathrm{ID}$, $\exists$ 向量(l_{xk}, l_{yk}, l_{zk})与 id_j 与对应(其中 $l_{xk} \in L_x, l_{yk} \in L_y, l_{zk} \in L_z$),且 $\mathrm{id}_{j+1} = \mathrm{id}_j + 1$. 即集合 ID 与集合 L_x, L_y, L_z 是 $1-n$ 关系(其中 $j \in (1, n)$ 且 $k = j + n \times i$, n 为 ID 集合中元素的个数, $k \in (1, m \times n)$).

则几何数据集 GC(geometry data collection)定义为 $\mathrm{GC} = \{F, \mathrm{ID}, L_x, L_y, L_z\}$.

定义 2 向量式有限元行为分析拓扑数据集(以下简称拓扑数据集)假定存在集合 F(其中 $\forall f \in F$ 代表模型不同的时刻)和集合 ID(其中 $\forall \mathrm{id} \in \mathrm{ID}$ 代表模型杆件的编号),以及集合 $\mathrm{ID}_1, \mathrm{ID}_2$(其中 $\forall \mathrm{id}_1 \in \mathrm{ID}_1, \forall \mathrm{id}_2 \in \mathrm{ID}_2$, $\mathrm{id}_1, \mathrm{id}_2$ 对应杆件的 2 个端点的编号). 且满足以下性质:

(1) 对 $\forall f_i \in F$, f_i 与全体 $\mathrm{id} \in \mathrm{ID}$ 对应,且 $f_{i+1} = f_i + t$(其中 t 为常数). 即集合 F 与集合 ID 是 $1-n$关系(其中 $i \in (1, m)$, m 为 F 集合中元素的个数)且 F 集合内元素 f_i 呈线性增长规律.

(2) 对 $\forall \mathrm{id}_j \in \mathrm{ID}$, 存在点对($\mathrm{id}_{1k}$, id_{2k})与 id_j 对应(其中 $\mathrm{id}_{1k} \in \mathrm{ID}_1, \mathrm{id}_{2k} \in \mathrm{ID}_2$),且 $\mathrm{id}_{j+1} = \mathrm{id}_j + 1$. 即集合 ID 与集合 $\mathrm{ID}_1, \mathrm{ID}_2$ 是 $1-n$ 关系. 且 $\mathrm{id}_{1k}, \mathrm{id}_{2k} \in$ GC. ID(其中 GC. ID 是几何数据集中的质点的 ID 集合),则拓扑数据集(topology data collection)定义为 $\mathrm{TC} = \{\mathrm{F}, \mathrm{ID}, \mathrm{ID}_1, \mathrm{ID}_2\}$.

定义 3 几何数据集向量匹配度(以下简称匹配度)假定存在几何数据集 $\mathrm{GC} = \{F, \mathrm{ID}, L_x, L_y, L_z\}$,若存在向量($x_i$, y_i, z_i)和(x_j, y_j, z_j)(其中 $x_i, x_j \in L_x, y_i, y_j \in L_y, z_i, z_j \in L_z$). 对其右对齐从左至右进行比较并计算匹配位数,设若存在向量(M_x, M_y, M_z),其中 M_x, M_y, M_z 分别是两向量中 x, y, z 分量的匹配位数,则称向量(M_x, M_y, M_z)为几何数据集向量匹配度. 例:

3.4995000000000005e−002	1.0000000000000000e+000	−2.1816789375000001e+000	−2.5656499999999999e−001	−2.3272699999999999e−001
3.5995000000000006e−002	1.0000000000000000e+000	−2.1816708999999999e+000	−2.5656499999999999e−001	−2.3272699999999999e−001

上述数据是由几何数据集中抽取出来的,它的匹配度是(8,24,24).

定义 4 几何数据集帧内冗余(以下简称帧内冗余)根据定义 1,假定存在几何数据集 $\mathrm{GC} = \{F, \mathrm{ID}, L_x, L_y, L_z\}$,若存在向量($x_i$, y_i, z_i)和(x_j, y_j, z_j)(其中 $x_i, x_j \in L_x, y_i, y_j \in L_y$ 且 $z_i, z_j \in L_z$,且 $\mathrm{id}_i \neq \mathrm{id}_j, f_i = f_j, j > i$). 且二者符号位不同,数值部分相同,则称向量($x_j$, y_j, z_j)为相对向量(x_i, y_i, z_i)的几何数据集帧内冗余,两向量对应的质点为互对称质点. 例如:

0.0000000000000000e+000	1.0000000000000000e+000	−2.1818179999999998e+000	−2.5656499999999999e−001	−2.3272699999999999e−001
0.0000000000000000e+000	2.0000000000000000e+000	2.1818179999999998e+000	−2.5656499999999999e−001	−2.3272699999999999e−001

其中,两向量的数值部分是完全相同的,仅符号不同,则其中任意一向量为另一向量的帧内冗余,两向量对应的质点为互对称质点.

定义 5 几何数据集帧间冗余(以下简称帧间冗余)根据定义 1,假定存在几何数据集 $\mathrm{GC} = \{F, \mathrm{ID}, L_x, L_y, L_z\}$,若存在向量($x_i$, y_i, z_i)和(x_j, y_j, z_j)(其中 $x_i, x_j \in L_x, y_i, y_j \in L_y, z_i, z_j \in L_z$,且 $\mathrm{id}_i = \mathrm{id}_j, f_i \neq f_j$). 若其匹配度为($M_x$, M_y, M_z),则称匹配度(M_x, M_y, M_z)为向量(x_i, y_i, z_i)及向量(x_j, y_j, z_j)之间的几何数据集帧间冗余. 例如:

3.3995000000000004e−002	1.0000000000000000e+000	−2.1816869749999999e+000	−2.5656499999999999e−001	−2.3272699999999999e−001
3.4995000000000005e−002	1.0000000000000000e+000	−2.1816789375000001e+000	−2.5656499999999999e−001	−2.3272699999999999e−001

其中,两向量之间的帧间冗余为(7,24,24).

定义 6 拓扑冗余. 根据定义 2,假定存在拓扑数据集 $\mathrm{TC} = \{F, \mathrm{ID}, \mathrm{ID}_1, \mathrm{ID}_2\}$,若存在点对($\mathrm{id}_{1i}$, id_{2i})和(id_{1j}, id_{2j})且它们对应的杆件编号分别为 $\mathrm{id}_p, \mathrm{id}_q$(其中 $\mathrm{id}_{1i}, \mathrm{id}_{1j} \in \mathrm{ID}_1, \mathrm{id}_{2i}, \mathrm{id}_{2j} \in \mathrm{ID}_2$,且 $\mathrm{id}_p = \mathrm{id}_q$),且 $\mathrm{id}_{1i} = \mathrm{id}_{1j}, \mathrm{id}_{2i} = \mathrm{id}_{2j}$,则称点对($\mathrm{id}_{1i}$, id_{2i})是点对(id_{1j}, id_{2j})的拓扑冗余. 例如:

0.0000000000000000e+000	8.6000000000000000e+001	1.1000000000000000e+001	2.4000000000000000e+001
7.9000000000000008e−003	8.6000000000000000e+001	1.1000000000000000e+001	2.4000000000000000e+001

定义 7 行为数据矩阵. 根据定义 1 和定义 2 以及行为数据文件中对几何数据集和拓扑数据集的组织形式,对几何数据集和拓扑数据集用矩阵形式进行组织和抽象后,对行为数据矩阵定义如下:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2N} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} & \cdots & a_{iN} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{M1} & a_{M2} & \cdots & a_{Mj} & \cdots & a_{MN} \end{bmatrix}$$

如上式，设若存在维数为 $M\times N$ 的矩阵，若输入数据为几何数据集，M 是几何数据集中 F 集合的大小，N 是几何数据集中 ID 集合的大小. 若 $a_{ij}=\{f_j,\ \mathrm{id}_i,\ l_{x(j-1)\times M+i},\ l_{y(j-1)\times M+i},\ l_{z(j-1)\times M+i}\}$，则称行为数据矩阵为几何数据集对应的行为数据矩阵，以下简称几何数据矩阵. 若输入数据为拓扑数据集，M 则代表拓扑数据集中 F 集合的大小，N 是拓扑数据集中 ID 集合的大小，$a_{ij}=\{f_j,\ \mathrm{id}_i,\ \mathrm{id}_{1(j-1)\times M+i},\ \mathrm{id}_{2(j-1)\times M+i}\}$. 则称行为数据矩阵为拓扑数据集对应的行为数据矩阵. 以下简称拓扑数据矩阵. 对几何数据矩阵元素 a_{ij}，若存在其帧间冗余，则它的帧间冗余可以在 $\{a_{ij-k1}\cdots a_{ij-1}\}\cup\{a_{ij+1}\cdots a_{ij+k2}\}$ 中找到，其中 $0<k_1<j$，$0<k_2<N-j$. 若 a_{ij} 存在帧内冗余，则其帧内冗余可在 $\{a_{1j}\cdots a_{i-1j}\}\cup\{a_{i+1j}\cdots a_{Mj}\}$ 找到.

对几何数据矩阵中任意列向量 $\boldsymbol{a}_j=[a_{1j}\cdots a_{2j}\cdots a_{Mj}]^{\mathrm{T}}$，若 $\exists a_{ij}$ 且 $l_{x(j-1)\times M+i}=l_{y(j-1)\times M+i}=l_{z(j-1)\times M+i}=0$，且 a_{ij} 之后的所有同属于 $\boldsymbol{a}_j$ 向量的元素 a_{i+1j} 直至 a_{Mj} 的分量 $l_{x(j-1)\times M+i+1}=l_{y(j-1)\times M+i+1}=l_{z(j-1)\times M+i+1}=0$ 直至 $l_{x(j-1)\times M+M}=l_{y(j-1)\times M+M}=l_{z(j-1)\times M+M}=0$，则称这些向量对应的点为无效质点.

3.2　压缩及解压缩模型

3.2.1　压缩模型思想

由定义3及定义4所述，几何数据集的冗余分为2类，即帧内冗余和帧间冗余. 下面分别基于2类冗余的特性进行研究并归纳其共性，给出几何数据集压缩模型的思想[14].

3.2.1.1　帧内冗余压缩子模型

该模型简称为帧内压缩模型. 根据定义1、定义4、定义7所述，帧内压缩模型完整步骤如下.

步骤1：读入数据文件，根据定义7将数据文件抽象为行为数据矩阵，其中，$a_{ij}=\{f_j,\ \mathrm{id}_i,\ l_{x(j-1)\times M+i},\ l_{y(j-1)\times M+i},\ l_{z(j-1)\times M+i}\}$，每次解析一个行为数据矩阵的列向量，即 $[\boldsymbol{a}_1,\cdots,\boldsymbol{a}_j,\cdots,\boldsymbol{a}_n]$

根据定义4及定义7有：对行为数据矩阵元素 a_{ij}，由于帧内冗余可能不会连续出现，采用 hash 散列和拉链法对数据矩阵预处理，对对称点进行聚类. 当互对称质点个数占有效点的个数的比例大于0.5时执行步骤2，当互对称质点个数占有效点的个数的比例小于0.5时帧内压缩模型步骤结束，转而利用帧间冗余压缩模型进行压缩.

步骤2：若数据矩阵中存在元素 a_{ij} 和 a_{kj} 使得 a_{kj} 是 a_{ij} 的帧内冗余，则删除 a_{kj} 的 $l_{x(j-1)\times M+k}$，$l_{y(j-1)\times M+k}$，$l_{z(j-1)\times M+k}$ 分量. 根据 a_{kj} 的坐标分量（$l_{x(j-1)\times M+k}$，$l_{y(j-1)\times M+k}$，$l_{z(j-1)\times M+k}$）相对于 a_{ij} 的坐标分量（$l_{x(j-1)\times M+i}$，$l_{y(j-1)\times M+i}$，$l_{z(j-1)\times M+i}$）同异用符号 s 编号并将 s 写入压缩文件，并将 id_k 以整形记录下来. 根据定义2及定义5，选择互对称质点个数占有效点的个数的比例以0.5为界可以降低 hash 散列带来的空间浪费，并可充分利用帧间冗余以提升压缩率. 每次解析一列向量后继续步骤1继续进行判断，若遇到无效质点则转去执行步骤3.

步骤3：根据定义7，若遇到 $l_{x(j-1)\times M+i}=l_{y(j-1)\times M+i}=l_{z(j-1)\times M+i}=0$ 且直至 $l_{x(j-1)\times M+M}=l_{y(j-1)\times M+M}=l_{z(j-1)\times M+M}=0$（$M$ 为 ID 集合中元素的个数），则认为 $a_{ij}, a_{i+1j},\cdots,a_{Mj}$ 为无效质点，将其删除并记录无效质点的个数，将其写入压缩文件以备解压之用.

步骤4：将压缩之后的行为数据矩阵写入压缩文件.

3.2.1.2　帧间冗余压缩子模型

该模型简称帧间压缩模型. 模型的变形、位移和断裂归根到底是质点坐标的变化. 而质点的运动轨迹随着时间的变化是近乎连续的.

根据对帧内压缩模型的定义以及定义1、定义3、定义5及定义7，帧间压缩模型执行步骤如下.

步骤1：根据对帧内压缩模型的定义以及定义1、定义5及定义7，当几何数据集中互对称质点个数占有效点的个数的比例小于设定的阈值时，证明模型在荷载作用后坐标位置发生变化，模型不再对称，对剩下的数据采用帧间压缩模型进行处理：将数据矩阵分为两部分，帧内压缩模型处理部分和帧间压缩模型处理部分，将帧间压缩模型处理部分转置. 因此，转置前后的行为数据矩阵分别为

$$\left[\begin{array}{cccc:ccc} a_{11} & a_{12} & \cdots & a_{1j-1} & a_{1j} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2j-1} & a_{2j} & \cdots & a_{2N} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ a_{M-11} & a_{M-12} & \cdots & a_{M-1j-1} & a_{M-1j} & \cdots & a_{M-1N} \\ a_{M1} & a_{M2} & \cdots & a_{Mj-1} & a_{Mj} & \cdots & a_{MN} \end{array}\right]$$

如上式，第 $j-1$ 列帧内冗余仍然满足阈值 X（X 为互对称质点个数占有效点的个数的比例，本文设为0.5）的限定，但处理到 j 时不满足 X 限定，停止对帧内冗余进行操作，对右半部分矩阵进行转置，

如下：

$$
\left[\begin{array}{cccc|ccccc}
a_{11} & a_{12} & \cdots & a_{1j-1} & a_{1j} & a_{2j} & \cdots & a_{M-1j} & a_{Mj} \\
a_{21} & a_{22} & \cdots & a_{2j-1} & a_{1j+1} & a_{2j+1} & \cdots & a_{M-1j+1} & a_{Mj+1} \\
\vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
a_{M-11} & a_{M-12} & \cdots & a_{M-1j-1} & \vdots & \vdots & \cdots & \vdots & \vdots \\
a_{M1} & a_{M2} & \cdots & a_{Mj-1} & a_{1N-2} & a_{2N-2} & \cdots & a_{M-1N-2} & a_{MN-2} \\
 & & & & a_{1N-1} & a_{2N-1} & \cdots & a_{M-1N-1} & a_{MN-1} \\
 & & & & a_{1N} & a_{2N} & \cdots & a_{M-1N} & a_{MN}
\end{array}\right]
$$

如上式，转置之后左侧为$(M\times(j-1))$矩阵，左侧数据矩阵的压缩已经由帧内压缩模型完成，右侧为$((N-j+1)\times M)$矩阵，写成向量形式则为$[a_1\ a_2\ \ \cdots\ \ a_{j-1}\mid a'_1\ \ a_M{}']$，对右侧继续进行解析，一次解析一个列向量.

步骤 2：根据定义 1、定义 7，对于右侧部分列向量中元素 $a_{ij}=\{f_j,\ \mathrm{id}_i,\ l_{x(j-1)\times M+i},\ l_{y(j-1)\times M+i},\ l_{z(j-1)\times M+i}\}$，$f_i$ 与 id_j 皆为线性增长规律. 首先将时刻信息 f_i 和 id_j 去除，则行为数据矩阵元素成为 $a_{ij}=\{l_{x(j-1)\times M+i},l_{y(j-1)\times M+i},l_{z(j-1)\times M+i}\}$. 因为时刻信息是线性增长的，因此其时刻信息 f_i 可以根据采样步长得到. 根据定义 7 对数据矩阵进行分析后得到其增长规律为：

$$f_i=T_0+k\Delta t \tag{1}$$

式中：k 为整数；Δt 为步长且为整数；f_i 为时刻信息. 其中 $T_0=f_2$，$k=f_3-f_2$. 对时刻信息 f_i，只要得到 T_0 和 k 即可，任意时刻 f_i 可由式(1)迭代得到，极大减少了存储空间的浪费.

随后每次解析一对 a_{ij}（$a_{ij}=\{l_{x(j-1)\times M+i},\ l_{y(j-1)\times M+i},\ l_{z(j-1)\times M+i}\}$且 j 为帧间冗余压缩起始点）和 a_{ij+1}($a_{ij}=\{l_{xj\times M+i},\ l_{yj\times M+i},\ l_{zj\times M+i}\}$)，根据定义 3 及定义 7，将 $l_{x(j-1)\times M+i}$ 与 $l_{xj\times M+i}$，$l_{y(j-1)\times M+i}$ 与 $l_{yj\times M+i}$，$l_{z(j-1)\times M+i}$与 $l_{zj\times M+i}$进行匹配并获得其匹配度($M_x,\ M_y,\ M_z$)，并剔除 a_{ij+1} 中匹配度部分，将剩余部分写入压缩文件. 根据定义 7，若遇 $l_{x(j-1)\times M+i}=l_{y(j-1)\times M+i}=l_{z(j-1)\times M+i}=0$ 直至某 k 时有 $l_{x(k-1)\times M+i}=l_{y(k-1)\times M+i}=l_{z(k-1)\times M+i}$ 不全为零，则认为 a_{ij} 直至 a_{ik-1}为无效质点，将其删除.

步骤 3：将压缩后的数据矩阵写入压缩文件中.

步骤 4：最后将几何数据集中 F 集合中元素的数量和 ID 集合中元素的数量以及 T_0 和 k、帧内压缩模型和帧间压缩模型的分裂点 $j-1$ 写入压缩文件中以备解压缩之用.

根据帧内压缩模型和帧间压缩模型的执行步骤，则几何数据集压缩模型的执行步骤为：

步骤 1：将几何数据集的数据文件抽象为行为数据矩阵，用帧内压缩模型解析其列向量.

步骤 2：若解析某列向量时其互对称质点个数占有效点的个数的比例不满足预设的值，则将行为数据矩阵分为两部分并将右侧矩阵转置. 否则继续执行步骤 1. 直至出现互对称质点个数占有效点的个数的比例不满足预设的值的情况，转去执行步骤 3.

步骤 3：用帧间压缩模型对右侧行为数据矩阵的列向量解析，直至算法结束.

3.2.1.3　拓扑冗余压缩模型

根据定义 2 及定义 6，拓扑数据集存在着大量的冗余. 对拓扑数据集，由于杆件自身的数量在不同时刻 f_i 是不会发生变化的，仅在模型在不同操作情况下发生断裂时杆件的编号 id_i 才会改变，并且，每个杆件的连接状态只可能发生一次变化，因此，当行为数据矩阵为拓扑数据矩阵时，首先将整个拓补数据集对应的数据矩阵进行转置，只需存储初始时刻和断裂时刻对应的杆件信息即可，进而实现压缩.

拓扑冗余压缩模型(以下简称拓扑压缩模型)逻辑为：

步骤 1：读入数据文件，根据定义 3 及定义 7 将数据文件抽象为拓扑数据矩阵，首先将矩阵转置，则矩阵中 $a_{ij}=\{f_j,\ \mathrm{id}_i,\ \mathrm{id}_{1(j-1)\times M+i},\ \mathrm{id}_{2(j-1)\times M+i}\}$，行为数据矩阵的列向量为$[\boldsymbol{a}_1\ \ \cdots\ \ \boldsymbol{a}_j\ \ \cdots\ \ \boldsymbol{a}_n]$.

步骤 2：每次读入一个列向量进行解析，根据定义 7，每次解析一个列向量，即顺序为 $\boldsymbol{a}_1,\boldsymbol{a}_2,\boldsymbol{a}_3,\cdots,\boldsymbol{a}_N$.

根据定义 6，若对于 $\forall a_{ij}$(其中 $a_{ij}=\{f_i,\ \mathrm{id}_j,\ \mathrm{id}_{1(j-1)M+i},\ \mathrm{id}_{2(j-1)M+i}\}$)，若 $\exists a_{ik}$(其中 $a_{ik}=\{f_i,\ \mathrm{id}_k,\ \mathrm{id}_{1(k-1)M+i},\ \mathrm{id}_{2(k-1)M+i}\}$ 且 $k>j$，$\mathrm{id}_{1(j-1)M+i}=\mathrm{id}_{1(k-1)M+i}$，$\mathrm{id}_{2(j-1)M+i}=\mathrm{id}_{2(k-1)}$，将其删除. 根据定义 2 和定义 6，若 $\exists a_{ip}$(其中 $a_{ip}=\{f_i,\ \mathrm{id}_p,\ \mathrm{id}_{1(p-1)M+i},\ \mathrm{id}_{2(p-1)M+i}\}$ 且 $p>j$) 使 $\mathrm{id}_{1(j-1)M+i}$ $\mathrm{id}_{1(p-1)M+i}$，$\mathrm{id}_{2(j-1)M+i}$ $\mathrm{id}_{2(p-1)}$情况成立 1 条或 2 条同时成立则证明模型在 f_p 时刻发生了断裂，将其保留. 由于每个杆件的连接状态只可能发生一次变化，所以，最好情况下，模型不发生连接状态变化，数据矩阵退化为列

向量,即:$[a_{11}\quad a_{12}\quad a_{13}\quad \cdots\quad a_{1N-1}\ a_{1N}]^{\mathrm{T}}$

根据式(1),若 $f_j\in\{f_i, f_{i+1}, \cdots, f_p\}$,则 f_j 可由式(1)推导而得. 根据定义 2,对 id 信息,有 $\mathrm{id}_{j+1}=\mathrm{id}_j+1$,则只需要知道 ID 集合的大小并依次编号即可.

步骤 3:最后将拓扑数据集中 F 集合中元素的数量和 ID 集合中元素的数量以及 T_0 和 k 写入压缩文件中以备解压缩之用.

3.2.2　解压缩模型思想

根据定义 1、定义 3、定义 4、定义 5 以及定义 7 以及对帧内压缩模型和帧间压缩模型的定义,对几何数据集的解压缩分为帧内冗余的解压和帧间冗余的解压.

(1) 根据定义 1、定义 4、定义 7 和帧内压缩模型的定义,帧内冗余子解压模型(以下简称帧内解压模型)定义为:

步骤 1:根据定义 7,读取压缩文件,考虑压缩文件对应的几何数据矩阵,每次读取列向量 a_i.

步骤 2:根据帧内压缩模型的定义及定义 7,若 a_i 存在元素 a_{ij} 和 a_{kj},其中 a_{ij} 是未被压缩的原始数据. a_{kj} 是已被压缩过的 a_{ij} 的帧内冗余,则它的分量只包括以整形存储的 id 和 a_{kj} 相对于 a_{ij} 的对称特点(用符号 s 表示). 由定义 1,读入 T_0 和 k,并将所有元素时刻信息 f_i 还原,再加上 a_{kj} 相对于 a_{ij} 的对称特点和整形存储的 id 可将 a_{kj} 还原为原始数据. 对于采取对称点特性进行压缩的部分,无需转置. 直至帧内冗余的压缩后的向量处理完毕.

步骤 3:将解压缩后的几何数据矩阵写入数据文件. 继续按列向量 $\boldsymbol{a}_i$ 以解压被帧间压缩模型压缩的数据矩阵.

(2) 根据定义 1、定义 4、定义 7 以及帧间压缩压缩模型的定义,帧间冗余子解压模型(以下简称帧间解压模型)的执行步骤为:

步骤 1:根据定义 1、定义 4 及帧内压缩模型的定义,对于利用帧间冗余压缩的部分,每次读入列向量只需要比较相邻时刻元素 a_{ij} 和 a_{ij+1} 之间数据长度的差异即可,基于数据长度的差异,将 a_{ij+1} 被删除的匹配度部分复原.

步骤 2: 根据对帧间压缩模型的定义,对用帧间压缩模型进行处理的矩阵部分转置,得到原始数据矩阵.

步骤 3:根据定义 1,读入 T_0 和 k,并将所有元素时刻信息还原,然后将几何数据矩阵写入解压文件或为后处理工具读入.

则根据帧内解压模型和帧间解压模型的定义和定义 1、定义 3、定义 4、定义 5 及定义 7 可得几何数据解压模型:

步骤 1:读入几何数据集压缩数据文件,将其抽象为行为数据矩阵,获得原始解压时帧内压缩模型和帧间压缩模型对几何数据矩阵的分界,用帧内解压模型对其列向量进行解析.

步骤 2:若解析某列向量时其帧内压缩模型和帧间压缩算法的模型分界不为零,则将行为数据矩阵分为两部分并将右侧矩阵转置,执行步骤 3. 否则继续执行步骤 1 直至帧内压缩部分解压完毕.

步骤 3:用帧间解压模型进行解析右侧行为数据矩阵的列向量. 直至算法结束.

(3) 根据定义 2、定义 6、定义 7 以及拓扑压缩模型对拓扑数据的解压缩模型(以下简称拓扑解压模型):

步骤 1:读取压缩文件,根据定义 7,对拓扑数据矩阵,每次读取向量 $\boldsymbol{a}_i$,若 $\boldsymbol{a}_i$ 存在元素 a_{i1} 和 a_{ik}(其中 $a_{ij}=\{f_j,\ \mathrm{id}_i,\ \mathrm{id}_{1(j-1)\times M+i},\ \mathrm{id}_{2(j-1)\times M+i}\}$, $a_{ik}=\{f_k,\ \mathrm{id}_i,\ \mathrm{id}_{1(k-1)\times M+i},\ \mathrm{id}_{2(k-1)\times M+i}\}$)且点对($\mathrm{id}_{1(j-1)\times M+i}$, $\mathrm{id}_{2(j-1)\times M+i}$)和($\mathrm{id}_{1(k-1)\times M+i}$, $\mathrm{id}_{2(k-1)\times M+i}$)不等,则证明模型在 k 时刻发生了断裂,模型在 1 与 k 之间的时刻中杆件的顶点直接复制 1 时刻顶点连接信息即可,k 时刻之后的顶点连接信息直接复制 k 时刻杆件的顶点即可. 若杆件在模型发生状态变化过程中始终没有发生断裂,则仅复制 1 时刻的杆件的连接情况即可.

步骤 2:读入 T_0 和 k,并将所有元素时刻信息还原.

步骤 3:将拓扑数据矩阵转置并写入解压文件或为后处理工具读入.

4　压缩与解压缩模型算法实现

4.1　压缩模型算法实现

压缩算法的输入为同济大学土木工程学院提供的向量式有限元程式计算行为数据,共分为两个文件,一个是节点数据(node. txt),另一个是拓扑数据(element. txt),算法读入数据后即按照压缩模型遍历数据,直至算法结束. 输出为 2 种数据的压缩文件(comp_node. txt,comp_element. txt).

4.2　解压缩模型算法实现

以压缩算法的输出作为解压缩算法的输入,可得解压缩算法的输入,读取两类数据的压缩文件,并

按解压缩模型,即可还原原始文件,丝毫不影响数据精度(node.txt.element.txt).

4.3 模型算法复杂度分析

(1) 由定义1、定义3、定义4、定义5及定义7以及几何数据压缩模型以及对几何数据压缩算法的实现,设若行为数据矩阵维数为$M\times N$,即M为几何数据集中ID集合的元素数量,N为几何数据集中F集合的元素数量,则对几何数据集压缩算法的时间复杂度分析有如下结论:

根据对帧内压缩算法的实现,帧内压缩算法中对原始行为数据矩阵左侧维数为$M\times(j-1)$的矩阵进行操作.每次读入1个列向量,因此,算法时间复杂度为$T(s)=o(s)$(其中$s=M\times(j-1)$).

另外,步骤2中对原始行为数据右侧维数为$(N-j+1)\times M$的矩阵进行操作,因此,算法时间复杂度为$T(r)=o(r)$(其中$r=((N-j+1)\times M)$).

所以几何数据集压缩算法的时间复杂度为:$T(n)=o(n)$(其中$n=M\times N$).

(2)由定义2、定义6、定义7以及对拓扑数据压缩模型的定义及对拓扑数据压缩算法的实现,对拓扑数据压缩算法存在如下结论:

每次读入1个转置后的拓扑数据矩阵的列向量a_i解析,直至算法结束,因而拓扑数据压缩算法的时间复杂度为$T(n)=o(n)$(其中$n=M\times N$).

5 算法验证

5.1 压缩算法原型的实现

为验证压缩算法的效果编写压缩算法的原型实现.采用C#语言,在Visual Studio2010环境下进行开发,试验环境为Corei3处理器,2.30GHz,4.00GB内存,win7系统.

对2组非线性向量式有限元数据进行压缩验证,数据来源为同济大学土木工程学院经计算导出的桥梁倒塌模型和塔楼倒塌模型.此算法验证的目的在于获得压缩算法的实际压缩率、压缩时间和解压缩时间.

5.2 算法压缩率和无损度分析的验证

表1和表2是对2组数据分别进行进行压缩和解压缩之后得到的结果.

由表1和表2可得:无论是几何数据集还是拓扑数据集均能取得较好的压缩率,特别是拓扑数据集的压缩.

最后通过程序Test_Difference.exe验证得到结论:两者数据完全相同,实现了真正的无损压缩.

表1 桥梁模型压缩及解压验证

Tab.1 Validation of the bridge collapse model compression

数据集	压缩前数据量/MB	压缩后数据量/MB	解压后数据量/MB	压缩率/%
几何	77.9	14.3	77.9	80.36
拓扑	25.3	0.031 8	25.3	99.87

表2 楼房模型压缩及解压验证

Tab.2 Validation of the tower collapse model compression

数据集	压缩前数据量/MB	压缩后数据量/MB	解压后数据量/MB	压缩率/%
几何	5.34	1.31	5.34	75.47
拓扑	9.68	0.094 5	9.68	99.02

5.3 解压数据与原始数据的3D效果对比验证

对解压缩后的数据导入后处理软件中进行测试之后得到解压缩后的数据和原始数据对于三维模型动态演示的效果进行比较,解压缩数据能够良好地实现数据可视化,丝毫不影响动态展示结果.

6 结论和展望

从试验结果分析可以看出:本文模型实现了非线性向量式有限元后处理数据的无损压缩,并且有良好的压缩效果,有较好的应用前景.尤其对于海量的非线性向量式有限元行为数据的压缩,例如大于100 M以上的行为数据的压缩,有着更好的效果.

通过算法的验证可以得到以下结论:

(1) 算法较好地利用了数据间的关系和冗余,证明该方法是可行的.算法很好地解决了向量式有限元方法应用于基于web的体系架构的数据传输和文件读取的效率的瓶颈问题.

(2) 解压缩简单,且解压缩速度很快,尤其适于海量行为数据的压缩和解压.

(3) 行为数据文件在经过压缩之后,读取的速度明显快于未压缩文件的读取.

(4) 解压缩数据和原始数据是完全一样的,达到了本文实现无损压缩的目的.

参考文献:

[1] 卢哲刚,姚谏.向量式有限元——一种新型的数值方法[J].空间结构,2012,18(1):85.
LU Zhegang, YAO Jian. Vector form intrinsic finite element—A new numerical method [J]. Spatial Structures, 2012, 18(1): 85.

(下转第145页)