

大规模向量式有限元行为数据压缩模型及算法

杜庆峰¹, 周晓玮¹, 谢 涛², 周雪非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, 美国 圣迭戈 92182)

摘要: 针对向量式有限元程式计算所产生的结果数据体积庞大, 上传云端和读取处理的速度瓶颈, 必须对其进行有效的压缩. 通过对向量式有限元行为数据进行分析, 依据数据本身的冗余度高、无效信息多、数据格式较长等特点, 进而提出一种适用的压缩算法, 对行为数据进行时间帧合并、删减冗余数据、舍去过长精度、线元素增量表示等压缩方法. 经过验证, 本算法压缩后的数据体积明显减小, 而程序读取数据的速度则有显著提高, 大幅改善了向量式有限元行为分析软件的使用效率.

关键词: 向量式有限元; 数据压缩算法; 行为数据

中图分类号: TN919

文献标志码: A

Massive Vector Form Intrinsic Finite Element Behavior Data Compression Model and Algorithm

DU Qingfeng¹, ZHOU Xiaowei¹, XIE Tao², ZHOU Xuefei¹

(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 result data of the mid-processing computation of vector form intrinsic finite element analysis has a large size. That is a considerable bottleneck in the speed of the Cloud uploads and program load, which should be highly compressed. The behavior data of the vector form intrinsic finite element is studied. After analyzing the characteristics of high redundancy, invalid data and numerical notation format of the data, a practical compression algorithm is proposed to merge the time information, delete redundant data, round the over precision and use the line element incremental expression. The numerical experiments indicate that, this algorithm highly compresses the behavior data size and accelerate the data load speed. It greatly improves the usage efficiency of the vector form intrinsic finite element behavior analysis software.

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

1 背景及研究内容

1.1 背景及问题的提出

在工程技术和科学技术领域, 有限元分析是一种十分有效的数值分析方法. 相比于传统的有限元分析方法, 普渡大学丁承先教授等基于向量力学与数值计算, 提出了向量式有限元分析法(vector form intrinsic finite element, VFIFE)^[1-4]. 它可以有效处理连续体几何变形、非线性与不连续的材料本构关系等各种复杂情形, 因此在大变形、断裂、碰撞等工程问题的研究中具有很好的应用前景.

模型通过 VFIFE 程式计算(下文简称“中处理”)后, 得到的结果数据称为向量式有限元行为分析数据(简称行为数据, behavior data), 行为数据被分别写入多个文本文件中, 如 Node.txt 和 Element.txt, 计算机在后期图形化处理(简称后处理)过程中读取这些文件, 绘制出其中的点、线等行为数据, 用于动态展示并分析模型的形变、断裂等三维效果形态.

由于原始的工程模型十分庞大, 可达数百万个结构单元, 所以在经过中处理之后, 行为数据的规模也通常十分庞大, 文件体积可达几十兆字节. 在后处理时, 庞大的数据量导致文件读取与处理缓慢, 计算非常费时. 同时, 为了加强向量式有限元行为分析软件的实用性, 行为数据的上传至云端也成了迫切的实际需求. 但数百兆的数据无法在短时间内完成上传, 不利于之后数据文件在网络中的传输, 所以庞大的数据规模成为了后处理的一大瓶颈. 为了使计算

收稿日期: 2013-09-24

基金项目: 国际自然科学基金(41171303)

第一作者: 杜庆峰(1968—), 男, 工学博士, 教授, 博士生导师, 主要研究方向为软件测试. E-mail: du_cloud@tongji.edu.cn

通讯作者: 周晓玮(1989—), 男, 硕士生, 主要研究方向为计算机交叉领域. E-mail: waynechou1204@gmail.com

结果的储存更精简、读取更快速、传输更高效,就有必要对行为数据进行压缩。

1.2 相关工作及本文的研究内容

目前,同济大学与浙江大学在向量式有限元计算中处理和后处理渲染已有了一定成果.于丹的并行计算追赶算法^[5]可以对模型的程式化计算进行加速,并且通过计算机生成行为数据.后处理的数据读取与行为分析已经完成,通过调用 OpenGL,整个有限元分析过程可以以三维动画的形式展示在用户面前。

模型压缩分为两大类,分别是几何模型压缩和拓扑模型压缩.在几何模型数据压缩方面,龙海军等^[6]的斜率比较旋转门压缩算法通过比较历史增量数据,对相邻两个数据点做斜率比较,舍去小差值数据,最后进行霍夫曼编码压缩,但压缩有损率与阈值相关,不够稳定;李帅等^[7]的基于无拓扑矢量曲线的快速压缩算法通过比较曲线各特征点之间的斜率差,对特征点进行阈值取舍,不过此方法适用于曲线,对于直线较多的模型效果不佳;伏鹏^[8]的三角网格压缩算法几何压缩部分将数据点的坐标进行重排,并记录对应变换数列,排序后连接成曲线,再舍去相近数据点,达到压缩的目的,但此方法复杂,且需要额外记录变换数列,并不经济。

在拓扑压缩方面,刘迎等^[9]提出的基于上下文算术编码的非三角网格拓扑压缩参考了流行的边界分割三角拓扑压缩算法和三角形网络(KB)面片压缩算法,用“跳”代替分割三角区域,将18种拓扑关系简化为13种,最后再进行霍夫曼编码压缩,但该方法仍存在较多拓扑关系分类,需要分情况讨论,算法复杂。

以上各方法均是针对三角形网络结构的几何模型压缩,目前并没有针对 VFIFE 行为数据压缩处理相关的研究.基于以上分析,结合数据自身特点,进而提出了一种 VFIFE 行为数据压缩模型,解决了由于大型结构模型计算生成的行为数据过于庞大,导致后处理费时、占用存储空间过多等问题。

首先分析行为数据的特点,然后给出相关定义,提出了行为数据质点元素压缩模型和行为数据线元素压缩模型,利用降低数值精度、替换数据表示格

式、去除无效数据、合并冗余信息等方法,在保证数据不丢失的情况下,将行为数据进行大幅压缩.最后对压缩模型予以算法实现并进行验证。

2 向量式有限元行为数据压缩模型

2.1 行为数据及其特点

在对 VFIFE 模型的研究过程中可以发现,行为数据自身拥有一些特点,针对这些特点可以有效地进行数据的整理和压缩.目前中处理生成两个文本格式的行为数据,一个是 Node.txt(含时间信息的几何模型数据),另一个是 Element.txt(含时间信息的拓扑模型数据).这些数据文件有4个主要特点:规模大、使用科学计数法格式表示、存在无效信息和存在冗余数据. Element.txt 中还含有 Node.txt 的相关信息,这两个文件存在一定的关联。

(1) 数据规模大的特点

行为数据文件的体积与模型的质点、线元素数量,以及模型演示时间帧数量成正比,可庞大数十兆字节.例如桥梁倒塌模型的 Node.txt 文件,大小为 77.9 MB。

(2) 数值表示格式的特点

在 Node.txt 和 Element.txt 文件中,储存的所有数据表示格式均为 24 位科学计数法,例如, $-2.181\ 817\ 999\ 999\ 8e+000$. 其中第1位是数值的正负符号,第2至第19位为尾数部分,第20位是指数符号 e,第21位为指数正负符号,第22位至第24位为指数部分。

(3) 存在大量无效信息的特点

Node.txt 文件描述的是行为数据质点元素的信息.在有限元模型分析初期,部分质点元素还未出现.基于特点(2),此时这些质点的位置坐标均为零,并不会显示到屏幕上,存在空间冗余现象,如图1所示;在计算后期,这些行为数据质点元素的位置坐标发生改变,成为非零数值,将根据它们的坐标显示到屏幕上.因而 Node.txt 数据也呈现先稀疏、后稠密的特点。

```
0.0000000000000000e+000 1.3500000000000000e+002 0.0000000000000000e+000 0.0000000000000000e+000 0.0000000000000000e+000
0.0000000000000000e+000 1.3600000000000000e+002 0.0000000000000000e+000 0.0000000000000000e+000 0.0000000000000000e+000
0.0000000000000000e+000 1.3700000000000000e+002 0.0000000000000000e+000 0.0000000000000000e+000 0.0000000000000000e+000
```

图1 行为数据质点元素文件中的默认质点元素

Fig.1 Default node element in behavior data node element file

(4) 存在冗余数据的特点

在 Node.txt 和 Element.txt 文件中,第一列时间信息数据重复度十分高.而在 Element.txt 文件中,则由于数据组成相对稳定,所以在不同的时间帧

下,多数行为数据线元素重复出现,存在大量时间冗余.如图 2 中,第一列中时间 0 在每行都会重复;图 3 中显示,在 0 时刻和 0.000995 时刻,第 215、216、217 号线元素对应的数据在两个时刻都对应相同.

```
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
0.0000000000000000e+000 3.0000000000000000e+000 -2.1818179999999998e+000 2.5656499999999999e-001 -2.3272699999999999e-001
```

图 2 行为数据质点元素文件中的冗余时间

Fig.2 Redundant time in behavior data node element file

```
0.0000000000000000e+000 2.1700000000000000e+002 8.0000000000000000e+000 5.2000000000000000e+001
0.0000000000000000e+000 2.1600000000000000e+002 6.0000000000000000e+000 5.0000000000000000e+001
0.0000000000000000e+000 2.1500000000000000e+002 5.0000000000000000e+000 4.9000000000000000e+001

9.9500000000000001e-004 2.1700000000000000e+002 8.0000000000000000e+000 5.2000000000000000e+001
9.9500000000000001e-004 2.1600000000000000e+002 6.0000000000000000e+000 5.0000000000000000e+001
9.9500000000000001e-004 2.1500000000000000e+002 5.0000000000000000e+000 4.9000000000000000e+001
```

图 3 行为数据线元素文件中的冗余时间

Fig.3 Redundant information in behavior data node element file

2.2 相关定义

定义 1 行为数据时间集

假设 t 代表了整个向量式有限元行为模型中的某一时刻,模型中存在的 t 的集合称为时间集 T ,则可以表示为 $T = \{t_1, t_2, \dots, t_m\}$. 即行为数据时间集 (behavior data time set, BDTS).

定义 2 行为数据质点元素

行为数据质点元素 (behavior data node element, BDNE) 是向量式有限元模型的基本组成单元,代表了向量式有限元行为模型中的一个节点,是几何模型在时间跨度上的数据.假设存在一个行为数据质点元素 (BDNE) 为 P ,它是由时间、标号 (ID)、坐标等信息组成,则它的构成^[10]可以表示为 $P(T, A, C_x, C_y, C_z)$,令 $A = \{a_1, a_2, \dots, a_i\}$; $C = \{c_1, c_2, \dots, c_i\}$. BDNE 的 ID 集 A 的元素 a 表示了此质点的 ID 编号;坐标集 C 的元素 c 则表示质点在向量式有限元行为分析模型中所处的三维空间坐标, C_x, C_y 与 C_z 分别代表了质点所处的 x, y, z 轴坐标位置.

定义 3 行为数据非零质点

基于定义 2,假设在压缩模型中存在一个 BDNE,其 x, y, z 坐标值不全为零,则称之为行为数据非零质点 (behavior data non-zero node, BDNN). 例如: $P(0, 1, 2, 3, 4)$.

定义 4 行为数据默认质点

基于定义 2 的基础,在模型中存在某一个 BDNE,其 x, y, z 坐标值全部为零,则称之为行为

数据默认质点 (behavior data default node, BDDN). 例如: $P(0, 2, 0, 0, 0)$.

定义 5 行为数据线元素

行为数据线元素 (behavior data line element, BDLE) 是向量式有限元行为模型中的另一种元素,代表了模型中的梁结构,是拓扑模型在时间跨度上的数据集合.假设存在一行行为数据线元素 (BDLE) 为 E ,由时间、标号 (ID)、相关行为数据质点的标号 (ID) 等组成.它的构成可以表示为 $E(T, B, A_1, A_2)$,令其中 $B = \{b_1, b_2, \dots, b_r\}$; $A_1 \subseteq A$; $A_2 \subseteq A$. BDLE 的线 ID 集 B 中的元素 b 代表了此线元素的 ID 编号; A_1, A_2 分别代表了组成线元素的 2 个行为数据质点元素的 ID,它们都是 BDNN.

2.3 行为数据的数学模型

2.3.1 行为数据质点元素的数学模型

根据对 BDNE 的定义,行为数据质点元素文件中的每一行记录代表了一个 BDNE.质点元素文件格式可以用数学模型表示为 $P = \{p_1, p_2, \dots, p_n\}$.其中, BDNE 集 P 的元素 p 代表了向量式有限元模型中的一个 BDNE,如图 1 所示.

假设:整个模型过程共由 m 个时间帧组成,模型中一共出现过 i 个不同 BDNE,质点数据文件的第 n 行数据的长度为 l_n 个字符,则文件的数据规模为

$$V_p = \sum_{n=1}^m l_n \quad (1)$$

由第 2.1 节中的特点(2)可知, P 中的每个向量均由 24 个字符组成的科学计数法表示,每个向量数据之前用一个空格符号隔开,所以每一行数据都是

定长,长度为 $l_n = 5 \times 24 + 5 = 125$ 个字符。

将长度 l_n 代入式(1)得到质点数据文件的规模为 $V_p = 125mi$ 个字符。

2.3.2 行为数据线元素的数学模型

根据对 BDLE 的定义,行为数据线元素文件中的每一行记录代表了一个 BDLE. 可以用数学模型表示为 $E = \{e_1, e_2, \dots, e_w\}$. 其中 e_w 为模型中存在的 w 个 BDLE.

其中, BDLE 集合 E 的元素 e 代表了向量式有限元模型中的一个 BDLE, 如图 3 所示。

假设: 整个模型过程共由 m 个时间帧组成, 一共出现过 r 个不同的 BDLE, t 时刻存在 j_t 个 BDLE, 由 2 个 BDNN 组成, 其中第 n 个 BDLE 数据的长度为 l_n 个字符, 则数据规模为

$$V_l = \sum_{t=1}^m \left(\sum_{n=1}^{j_t} l_n \right), 0 < j_t \leq r \quad (2)$$

E 中的每个向量同样由 24 个字符组成的科学计数法表示, 每个向量数据之前用一个空格符号隔开, 长度为 $l_n = 4 \times 24 + 4 = 100$ 个字符. 将长度 l_n 代入式(2)得到线元素数据文件的规模为 $V_l = 100 \sum_{t=1}^m j_t$, $0 < j_t \leq r$, 平均数据规模为 $\bar{V}_l = 50rm$ 个字符。

第 2.3 节即为向量式有限元行为数据文件的数学模型分析. 接下来将根据数据的特点和模型的实际应用, 对这些数据进行压缩, 以降低存储成本, 提高读取和传输效率。

2.4 压缩模型的思想

通过对行为数据特点的分析, 可以对其进行简单的逻辑压缩, 达到省略无效、重复数据的目的, 这是由数据自身特点所决定的压缩表示^[11]. 基于以上的相关定义以及数学模型的分析, 可以提出以下压缩思想。

2.4.1 针对行为数据质点元素文件的压缩

根据定义 2, 在每个 p 元素中, 时间向量 t 记录了所在的时间帧信息, 然而由特点(4)可知, 在同一个时间帧中, 所有的时间向量都相同, 存在数据冗余。

定义 6 行为数据质点帧

假设在某一时刻 t 上, 存在一个或多个 BDNN, 则称这一时刻 t 以及这些 BDNN 为一个行为数据质点元素帧 (behavior data node frame, BDNF)。

在某个 BDNF 中, 将这些同一时刻上的 t 向量提取出后, BDNF 可以用模型表示为 $F(T', P' +)$,

符号 $+$ 表示存在一个或多个该元素, 令其中 $F = \{f_1, f_2, \dots, f_m\}$; $P' = \{p'_1, p'_2, \dots, p'_i\}$; $T' \subseteq T$.

逻辑压缩后的 BDNF 集 F 中的元素 f 表示一个 BDNF. 其中 P' 的元素 p' 表示压缩后的 BDNE. P' 可以用模型表示为 $P'(A', C'_x, C'_y, C'_z)$, 其中 p' 均为 BDNN, 可表示为 $(C'_x, C'_y, C'_z) \neq (0, 0, 0)$, $A' \subseteq A$. 任意一个 f 包含一个时间元素 t' 和至少一个 p' .

假设: 整个模型过程共由 m 个时间帧组成, u 时刻有 k_u 个 BDNN, 时间向量 t' 的长度在 u 时刻为 q_u 个字符, 其中第 n 个向量 p' 的长度为 l_{nu} ; 模型中一共出现过 i 个不同的 BDNE, 则可表示为以下模型:

$$V'_p = \sum_{u=1}^m \left(q_u + \sum_{n=1}^{k_u} l_{nu} \right), 1 < k_u \leq i \quad (3)$$

由实际模型需求可知, 原数据中的数值精度过多, 并不需要 24 个精度位. 各向量值的精度基本为 6 位有效位, 浮点数精度 (float) 可以达到 6 至 7 位有效位, 满足精度需求. 根据对原数据的分析, 使用 float 精度后, 数值长度大约为 10 个字符, 包括整数位、符号位和小数位. 向量 t 的平均长度 q_u 为 10 个字符, 向量 p' 每行数据 l_{nu} 的长度平均为 $4 \times 10 + 3 = 43$ 个字符, 代入式(3), 则数据规模为 $V'_p = (10 + 43k_u)m$, 其中, k_u 为 u 时刻的 BDNN 个数. 平均数据规模为 $\bar{V}'_p = (5 + 43i/2)m$ 个字符, 与原数据规模 $125mi$ 个字符相比, 平均逻辑压缩率为 $(V_p - \bar{V}'_p)/V_p = (207i - 10)/250i$. 可见总质点数 i 越大, 质点位置信息数据的平均逻辑压缩率越高. 压缩后的数据形式如图 4 所示。

2.4.2 针对行为数据线元素文件的压缩

定义 7 行为数据线元素帧

假设在某一时刻 t 上, 存在一个或多个行为数据线元素帧 (behavior data line frame, BDLF), 则称这一时刻 t 以及这些 BDLE 为一个行为数据线元素帧 (BDLF)。

定义 8 行为数据线元素增量

基于定义 7, 假设存在两个相邻时刻 t 与 $t+1$ 上的 BDLF, 其中 $t+1$ 时刻上的 BDLF 中所含的 BDLE 与 t 时刻上的 BDLF 中所含的 BDLE 的不同部分, 包括新增、减少和数值不同三种类型情况, 则称 $t+1$ 时刻上这些不同的 BDLE 部分为行为数据线元素增量 (Behavior Data Line Increment, BDLI). 在每个 e 元素中, 同一个时刻中所有的时间向量 t 都相同, 将这些同一时刻中的向量 t 提取出后, 重新整理 BDLE. 通过观察 BDLE 文件进一步发

现,由于 BDLE 对应的 BDNE 信息只当模型中发生结构变形、断裂时候才发生变化,出现 BDLE 新增、消失或对应的 BDNE 信息变化等情况,所以可以将 BDLE 按基于时间帧的增量法表示,这样既可以消除冗余信息,又可保证数据的完整性,也便于数据的恢复。

在第一个 BDLF 中,默认认为所有出现的 BDLE 均为新增类型变量. 压缩整理后的数据构成表示为 $G(T', D+)$, 令其中 $G = \{g_1, g_2, \dots, g_m\}$, $D = \{d_1, d_2, \dots, d_n\}$; $T' \subseteq T$. 逻辑压缩后的 BDLF 集 G 中的元素 g 表示一个 BDLF. BDLI 集合 D 的元素 d 表示在某一时间帧中,与上一时间帧相比新增、消失或发生变化的 BDLE. 它的数据模型可以表示为 $D(H, E'+)$. 符号 $+$ 表示存在一个或多个该元素。

BDLI 类型集合 H 的元素 h 表示增量类型,即新增、消失和发生变化的线元素变化类型,分别用“+1”、“-1”和“+0”表示. 数据模型可以表示为 $H = \{“+0”, “+1”, “-1”\}$, $E' = \{e'_1, e'_2, \dots, e'_w\}$. 其中 E' 的元素 e' 表示压缩后的对应增量类型的 BDLE. E' 的数据模型可以表示为 $E'(B', A'_1, A'_2)$. 其中 e' 为增量整理后的 BDLE,其包含的向量含义为 $B' \subseteq B, A'_1 \subseteq A, A'_2 \subseteq A$. 向量的含义与原始线元素数据中的向量含义相同,只是去除了重复的时间向量以及对相应的冗余数据进行了增量整理。

假设:整个行为数据模型由 m 个时间帧组成;其中共出现过 r 个不同的 BDLE, t 时刻存在 s_t 个 BDLI,时间向量 t' 的长度为 q_t 个字符; t 时刻第 n 个 BDLI 数据 e'_n 的长度为 l_n 个字符,则数据规模为

$$V'_l = \sum_{t=1}^m (q_t + \sum_{n=1}^{s_t} l_n), 0 \leq s_t \leq r \quad (4)$$

与质点元素文件一样,在线元素数据文件中,基本只有时间和 ID 数据,不需要很高的精度,10 个字符足以表示当下的数据范围. 通过对精度的压缩,采用 Float 或整数(Int)类型,将数值的字符平均长度控制在 3 至 10 个字符以内,则 l_n 的平均长度为 $3 \times 10 + 2 = 32$ 个字符, q_t 为 10 个字符. 代入式(4),得到线元素文件经过逻辑压缩后的数据规模为 $V'_l = \sum_{t=1}^m (10 + \sum_{n=1}^{s_t} 32)$, $0 \leq s_t \leq r$, 平均数据规模为 $\bar{V}'_l = (16r + 10)m$ 个字符,与原数据规模平均值 $50rm$ 相比,平均逻辑压缩率为 $(\bar{V}_l - \bar{V}'_l) / \bar{V}_l = (34r - 10) / 50r$. 可见线元素总数 r 越大,线元素数据的平均逻辑压缩率越高. 压缩后的数据形式如图 5 所示。

```
0 51
2 0 0 0.3
3 0 0 0.6
4 0 0 0.9
5 0 0 1.2
6 0 0 1.5
7 0 0 1.8
8 0 0 2.1
9 0 0 2.4
10 0 0 2.7
```

图 4 压缩后的 BDNE 文件格式

Fig.4 Compressed BDNE file format

```
0
+1
96 26 52
95 52 39
94 13 26
93 13 39
92 25 51
91 51 38
90 12 25
89 12 38
```

图 5 压缩后的 BDLE 文件格式

Fig.5 Compressed BDLE file format

3 压缩模型的算法

3.1 压缩算法的实现

算法根据时间帧为单位,压缩每一帧上的模型数据,只需要一次遍历即可完成压缩. 此外,该算法便于即时传输和即时读取数据,符合未来项目的线上云端发展需要. 而且由于本压缩算法使用了符合后处理程序的格式,使得在后处理中并不需要解压缩,接收到的数据可以直接应用于后处理。

为了实现行为数据压缩算法并对其进行验证,可以编写一个压缩算法原型. 对 BDNE 文件压缩,输入原始 BDNE 文件 Node. txt,生成 PointComp. txt 压缩文件. 对 BDLE 文件压缩,输入原始 BDLE 文件 Element. txt,生成 LineComp. txt 压缩文件. 由于行为数据的压缩并没有改变数据的编码,而是改进了数据的结构与表示方式,更利于程序的直接读取。

为了验证压缩算法的正确性,不出现数据丢失等情况,压缩程序提供解压缩与验证功能. 输入文件 * Comp. txt,输出解压后文件 * Decomp. txt. 数据转换功能 Transfer 重现由科学计数法转为浮点数表示所产生的舍入误差和去除无效数据. 输入文件 Node. txt 与 Element. txt,输出转换后的文件 PointTransfer. txt 和 LineTransfer. txt. 数据比较功能 Compare 对比解压后文件 * Decomp. txt 与转换后的文件 * Transfer. txt 内容是否一致. 一致则可认为数据完整。

具体算法实现代码略。

3.2 算法复杂度及压缩有损率分析

3.2.1 压缩算法时间复杂度分析

BDNE 压缩算法是线性的,主循环次数取决于原始 BDNE 数据文件的行数,所以该算法的时间复杂度为 $O(n)$ 。

在 BDLE 压缩算法中,一共使用了 4 种函数,而各函数的时间复杂度均与时间帧所包含的 BDLE 个数有关,假设平均每个帧含有 r 个 BDLE,一共有 t 个时刻帧,则原始 BDLE 文件的总数据规模为 $n = tr$,所以它的复杂度为 $O(nr)$ 。

3.2.2 压缩有损率分析

有限元分析软件在数据精度、网格精度与计算速度、数据大小之间需要取得平衡^[12]。行为数据压缩算法属于有损压缩,VFIFE 后处理程序对行为数据进行模型观测,产生的精度截断误差基于中处理的结果,并不会造成误差的积累。

国家标准(GB)对于不同工程领域有着不同的精度要求和测量标准,如针对建筑物的观测精度^[13],误差应小于允许变形值的 5%~10%。假设建筑本身设计允许倾斜度为 0.4%,则观测误差需控制在 0.02%~0.04%。本文压缩算法采用的单精度浮点数完全可以满足此数量级的观测精度要求。

根据数据特点(2)和(3)可知,由于数据文件本身的存储格式为 24 位科学计数法,精度远高于实际需求的 0.02%,造成不必要的空间浪费。而且文件中存在大量无效信息,对于实际应用无益。所以在本算

法中,将对数据精度和无效信息进行缩短和去除,形成舍入误差和无效信息丢失等数据损失,无法在解压缩时对其恢复。但这些损失并不会对实际应用产生影响,所以可以认为此算法是合理的。

此外,拓扑模型使用的 BNLE 压缩算法中处理的字段,除了时间信息,均为整数,改变格式并不影响数值,实际效果等同于无损压缩。而后处理程序设定的时间帧间隔为 0.01 s,即使缩短计算间隔,压缩后的时间帧信息也可满足最小 10^{-7} s 数量级。

4 算法验证

基于压缩算法的原型,采用的验证环境为 AMD Athlon X2 250 处理器、2GB 内存、32 位 Win 7 操作系统。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建筑物倒塌模型。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,以及压缩有损率。

用此压缩算法对原始数据文件 Node 和 Element 各 2 个测试数据进行压缩。测试数据见表 1—2。其中, O_S 为原始文件大小, C_S 为压缩后的文件大小, D_S 为解压后的文件大小, T_{CR} 表示理论压缩率, $P_{CR} = (O_S - C_S) / O_S$ 表示文件的实际压缩率, L_R 为文件压缩有损率。 T_C 为压缩时间, T_D 为解压缩时间。 P_N 为质点文件 Node 中的 BDNE 总数, N_L 为线元素文件 Element 中的 BDLE 总数。

表 1 桥梁倒塌模型 BDNE 文件压缩验证

Tab.1 Validation of the BDNE file compression of the bridge collapse model

文件名	O_S /MB	P_N	$T_{CR}/\%$	C_S /MB	$P_{CR}/\%$	D_S /MB	$L_R/\%$	T_C /ms	T_D /ms
Node. txt	77.9	536	82.8	9.84	87.4	11.5	85.2	4 218	310

表 2 桥梁倒塌模型 BDLE 文件压缩验证

Tab.2 Validation of the BDLE file compression of the bridge collapse model

文件名	O_S /MB	N_L	$T_{CR}/\%$	C_S /MB	$P_{CR}/\%$	D_S /MB	$L_R/\%$	T_C /ms	T_D /ms
Element. txt	25.3	217	67.9	0.011 9	99.95	4.3	83	2 013	324

4.1 桥梁倒塌模型压缩验证

对于后处理程序读取文件的速度比较,测试数据见表 3。 T_{OL} 为读取原始数据文件的时间, T_{CL} 为读取压缩后文件的时间, $R_{LTC} = (T_{OL} - T_{CL}) / T_{OL}$ 为读取时间压缩率。

4.2 压缩算法之间的比较验证

由于 BDNE 和 BDLE 压缩算法是针对 VFIFE 行为数据格式的特定压缩算法,所以在此使用 Huffman、LZW、LZ77 等几种通用压缩算法与本文

表 3 桥梁倒塌模型文件读取速度验证

Tab.3 Validation of the loading time compression of the bridge collapse model

文件名	T_{OL} /ms	T_{CL} /ms	$R_{LTC}/\%$
Node. txt	11 291	1 144	89.9
Element. txt	9 250	44	99.5

的压缩算法作比较。表 4 以桥梁倒塌模型作为测试数据,比较各种压缩算法的压缩后数据大小及压缩率。

表 4 各压缩算法间的压缩率比较

Tab.4 Comparison of compression rate between different algorithms

文件名	文件原始大小/KB	Huffman 算法/KB	LZW 算法/KB	LZ77 算法/KB	BDNE 算法/KB	BDLE 算法/KB
BridgeNode. txt	79 839	28 732	11 849	13 410	10 076	
BridgeElement. txt	25 960	7 953	2 098	2 169		11. 9
平均压缩率/%		65. 33	86. 82	85. 27	87. 38	99. 95

由表 4 可见, BDNE 和 BDLE 压缩算法即使与物理编码压缩算法比较, 压缩率仍有一定的优势. 特定压缩算法对于行为数据有着更好的适用性, 因此也就拥有更高的压缩率.

5 结论与展望

实验结果证明, 本算法在对向量式有限元行为数据的压缩上有着显著的效果, 压缩后的文件大小都达到了预期的效果, 有很大的应用价值.

可以得出以下结论:

(1) BDNE 文件的压缩率和理论值十分接近, 实际压缩率略优于理论压缩率.

(2) BDLE 文件的实际压缩率与理论值相比有较大的提高, 这是由于原始数据文件中不同时刻之间, 数据重复度非常高, 在使用增量表示法后可以发生大幅度的精简. 因此, 数据压缩率仍比较依赖于原始文件中的数据冗余度等特征.

(3) 解压缩后虽然有损率较高, 但均属于精度舍入误差和无效信息的不可恢复, 如实际效果图所示, 并不影响实际的模型展示效果.

对于向量式有限元行为数据压缩的应用前景, 可以做出以下展望:

(1) 目前实验的文件最大为 77. 9MB, 但在实际应用中, 需要压缩的行为数据模型可能有更大的体积, 所以可以考虑在并行压缩方面进行深入的研究, 加快压缩速度.

(2) 本算法只考虑了质点元素数据和线元素数据的压缩, 尚未涉及其他类型的元素数据, 今后可以针对新的模型元素进行算法的改进和研究.

(3) 今后可以结合云端存储, 对数据文件的压缩、读取等实现在线管理, 更具实用性.

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.

[2] Ting E C, Shih C, Wang Y K. Fundamentals of a vector form intrinsic finite element: part I. basic procedure and a plane frame element[J]. Journal of Mechanics, 2004, 20(2): 113.
[3] Ting E C, Shih C, Wang Y K. Fundamentals of a vector form intrinsic finite element: part II. plane solid elements [J]. Journal of Mechanics, 2004, 20(2): 123.
[4] Shih C, Wang Y K, Ting E C. Fundamentals of a vector form intrinsic finite element: part III. Convected material frame and examples[J]. Journal of Mechanics, 2004, 20(2): 133.
[5] 于丹. 一种改进的基于线性有限元并行计算的追赶算法[J]. 科技传播, 2013(1): 112.
YU Dan. An improved thomas algorithm for linear finite element parallel computing [J]. Public Communication of Science & Technology, 2013(1): 112.
[6] 龙海军, 姚光霖. 核电站 DCS 系统历史数据压缩算法的研究[J]. 科技资讯, 2013(8): 29.
LONG Haijun, YAO Guanglin. Study of compressing algorithm on historical data in NPP DCS[J]. Science and Technology Information, 2013(8): 29.
[7] 李帅, 方源敏, 喜文飞. 基于无拓扑矢量曲线的快速压缩算法[J]. 科学技术与工程, 2011, 11(18): 4324.
LI Shuai, FANG Yuanmin, XI Wenfei. Rapid compression algorithm based on the no topological vector curve[J]. Science Technology and Engineering, 2011, 11(18): 4324.
[8] 伏鹏. 三角网格压缩算法 [D]: 杭州: 浙江大学, 2009.
FU Peng. Compression algorithms for triangle meshes [D]: Hangzhou: Zhejiang University, 2009.
[9] 刘迎, 韩忠明, 陈谊, 等. 基于上下文算术编码的非三角网格拓扑压缩[J]. 计算机工程与应用, 2010, 46(22): 178.
LIU Ying, HAN Zhongming, CHEN Yi, et al. Connectivity compression for non-triangular meshes by context-based arithmetic coding [J]. Computer Engineering and Applications, 2010, 46(22): 178.
[10] 古天龙. 软件开发的形式化方法[M]. 北京: 高等教育出版社, 2005.
GU Tianlong. Formal methods of software development[M]. Beijing: China Higher Education Press, 2005.
[11] 吴家安. 数据压缩技术及应用[M] 上海: 科学出版社, 2009.
WU Jiaan. Data compression techniques and applications [M]. Shanghai: Science Press, 2009.
[12] 李海峰, 吴冀川, 刘建波, 等. 有限元网格剖分与网格质量判定指标[J]. 中国机械工程, 2012, 23(3): 368.
LI Haifeng, WU Jichuan, LIU Jianbo, et al. Finite element mesh generation and decision criteria of mesh quality[J]. China Mechanical Engineering, 2012, 23(3): 368.
[13] 陈正阳, 曹传芬, 郑军. 工程建筑物变形观测精度和观测方法[J]. 中外建筑, 2003(6): 93.
CHEN Zhengyang, CAO Chuanfen, ZHEN Jun. Deformation observation precision and method of buildings [J]. Chinese and Overseas Architecture, 2003(6): 93.