Performance evaluation of efficient data compression
JHPCN-DF for large-scale structural analysis

Lijun LIU* and Masao OGINO*
* Information Technology Center, Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan
E-mail: kellyliu@hpc.itc.nagoya-u.ac.jp

Received 24 February 2016

Abstract
Numerical simulations such as finite element analyses have increased in size thanks to the explosive development of computer technology. However, large amount of data produced by large scale analysis leads to I/O bottlenecks in simulation, data processing and visualization at the same time. To alleviate this problem, we propose to employ JHPCN-DF technique to large scale finite element analysis for providing a powerful way of data compression. To reduce the quantity of transmission data which is used for further analyses and visualizations, we investigate the possibility of post calculations on local computers and evaluate the precision of post-run results which are calculated by JHPCN-DF encoded data. Over our numerical results, JHPCN-DF performs significantly better than the generic zlib compressor and obtained compression rate of 0.83 with a user defined allowed error $\varepsilon$ of $10^{-2}$ and 0.68 with $\varepsilon$ of $10^{-4}$. Our results also indicated that the post-run data calculated by reconstructed data with $\varepsilon$ lower than $10^{-4}$ achieves almost same level of precision with the one calculated by original data.

Key words: Data compression, JHPCN-DF, Large-scale structural analysis, Finite element method, Visualization

1. Introduction

The understanding of global scale phenomena is strongly needed for quick and appropriate responses to global environmental problems. Thanks to the explosive progress of computer technology, large scale finite element analysis to simulate and predict the behavior of natural things or artifacts with up to 1.8 trillion degree of freedom (DOF) are able to be performed (Ichimura et al., 2015). A large scale simulation run on a cluster of hundreds to thousands of supercomputer nodes results vast amounts of single- or double-precision floating-point numbers that represent one or more variables of simulation state per node/element (Schwan, 2003). Data explosion requires high capacity storage and causes a significant I/O bottleneck in simulation, data processing and visualization that stall the application. However, since the performance of data storage devices lagging greatly behind CPU speed, this problem is expected to become even more serious over the coming years. As most subsequent analysis and visualization are needed to be done on other computers rather than on where simulations are carried out, data transmission also becomes an increasingly difficult task.

Data compression strategies have potential to combat this problem. It can reduce the number of bytes that need to be transferred and stored. Data compression algorithms are classified in two ways i.e., lossless and lossy data compression algorithm. Lossless compression allows the utilization of data compression calculations to pack the content data furthermore permits the original data to be perfectly reconstructed from the compressed data. By contrast, lossy compression permits reconstruction only of an approximation of the original data. There are several lossless compression algorithms exist for floating-point data compression, whereas most of the compression schemes are very modest (Engelson et al., 2000, Isenburg et al., 2005, Ratanaworabhan et al., 2006). For analyzing and visualizing simulation results, to alleviate the problem of file transmission, higher compression rate is desirable if required precision can be obtained. Therefore, a compression method providing high compression rate and adapting to user’s
requirement for precision is preferable. Compression method Jointed Hierarchical Precision Compression Number-Data Format (JHPCN-DF) (Hagita et al., 2014) is exactly the one meets all the conditions mentioned above.

Except for data compression, reduction of output variables is another way for alleviating the problem caused by data explosion. In large scale finite element analysis, the most difficult work is solving the finite equilibrium equation, which is needed to be done on supercomputer. After solving the finite equilibrium equation, displacement is obtained and other variables can be calculated by obtained displacement much easier. If we obtain displacement only on supercomputer and compress and transfer it to local computers for further calculations, then a significantly decrease in the quantity of transmission data can be achieved. Here, the most important issue is to validate the precision of post-calculated data, however, no research is done about it until now. Therefore, although the way mentioned above is quite effective, it is seldom used.

To address these needs, in this paper, we evaluate the compression effectiveness of JHPCN-DF through implementing it to a parallel finite element code ADVENTURE_Solid (Mukaddes et al., 2014, Yoshimura et al., 2002), and investigate the precision of post-run data calculated by JHPCN-DF encoded displacement. As a result, JHPCN-DF is able to produce substantial increase in compression rate and provide required precision for visualizations and post analyses. We now describe JHPCN-DF technique in detail.

2. JHPCN-DF technique

JHPCN-DF is a data compression technique for floating-point dataset, which is useful for a wide range of applications in data science and simulation (Hagita et al., 2015). The concept of JHPCN-DF is using different numbers of bits in the IEEE 754 format to represent data for different purposes as shown in Fig. 1.

For the case of IEEE 754 single-precision floating point data, the stored value of the original one-variable requires zero padding and a 32-bit integer to record the separated bits necessary to reconstruct higher precision data and the original data (lossless). Zero padding of single-precision floating-point data is presented in Fig. 2. The lower bits of the fraction of a reconstruct data $y$ are filled with zeros by bitwise right shift operation then left shifts for the raw data $x$.

Since the lower bits seem to present random noise-like behavior which negatively affects data compression, they are considered to be unnecessary for visualizations and analyses. If a combination of neglecting the lower bits and the data compression such as the Huffman coding, a higher compression rate using only the higher bits can be obtained. The required bits can be determined by a user defined allowed error according to direct numerical computations and
their corresponding resultant visualizations and analyses. The allowed error $\varepsilon$ restrains the difference between the original value and reconstructed value, and we have

$$\max \left\{ \frac{|x_j - y_j|}{x_j}, j = 1, 2, \ldots, n \right\} \leq \varepsilon,$$

where $x_j$ is the original value of $j$-th component and $y_j$ is its reconstructed value.

3. Numerical examples and results

First, we investigate the compression effectiveness of JHPCN-DF through a Pantheon model (Ogino et al., 2008). Second, we evaluate the effect of JHPCN-DF on the precision of stress through a spherical cavity model (Timoshenko and Goodier, 1970). For data compression in JHPCN-DF technique, the generic zlib library is employed after data encoding. Our experiments were run on supercomputer Fujitsu PRIMEHPC FX100 in Nagoya University, and all computations were performed in double precision.

3.1 A Pantheon model

The boundary value problems of elasticity of a Pantheon model are solved by ADVENTURE_Solid. For boundary conditions, the displacements on the bottom plane in the direction of body weight are completely constrained, and the body’s own weight is applied as body force. Two finite element meshes with different DOF are established for structural analysis. One having 862,216 quadratic tetrahedral elements and 1,294,787 nodes with about 4 million DOF is called 4M DOF mesh, and the other having 6,897,728 quadratic tetrahedral elements and 9,778,949 nodes with about 29 million DOF is called 29M DOF mesh. The structural analysis is based on hierarchical domain decomposition method (Yagawa and Shioya, 1994). Such that, both models are decomposed to 16 parts and each part is decomposed into 128 subdomains for parallel computations. Output datasets of simulation results are 3D double-precision (8-bytes) nodal displacement and 1D double-precision nodal equivalent stress. The original output files of ADVENTURE_Solid are in binary format and the original data is decomposed into subdomains hierarchy. To eliminate the effect of the number of subdomains and make numerical experiments easy, functions to output the whole domain data in VTU format, which is used by the Visual Toolkit for unstructured grid, are implemented in ADVENTURE_Solid.

3.2 Compression rate of JHPCN-DF

Several different datasets included in output files are used to evaluate the compression effectiveness. The compression effectiveness is measured by computing the compression rate (CR), which is defined as follows:

$$CR = \frac{\text{uncompressed size} - \text{compressed size}}{\text{uncompressed size}}.$$

To understand the general level of compression rate on scientific data, firstly, we investigated the compression rate of the zlib compressor through the 4M DOF mesh and summarized the results in Table 1.

<table>
<thead>
<tr>
<th>Data set name</th>
<th>C data type</th>
<th>Raw data size [KB]</th>
<th>Compressed size [KB]</th>
<th>CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>4M3D displacement</td>
<td>double</td>
<td>31,075</td>
<td>29,515</td>
<td>0.050</td>
</tr>
<tr>
<td>4M1D nodal equivalent stress</td>
<td>double</td>
<td>10,358</td>
<td>9,758</td>
<td>0.058</td>
</tr>
<tr>
<td>4M3D coordinate</td>
<td>double</td>
<td>31,075</td>
<td>18,280</td>
<td>0.557</td>
</tr>
<tr>
<td>4M connectivity</td>
<td>int</td>
<td>34,489</td>
<td>15,295</td>
<td>0.557</td>
</tr>
<tr>
<td>4M offsets</td>
<td>int</td>
<td>3,449</td>
<td>1,128</td>
<td>0.673</td>
</tr>
<tr>
<td>4M element types</td>
<td>unsigned char</td>
<td>862</td>
<td>0.9</td>
<td>0.999</td>
</tr>
</tbody>
</table>

In Table 1, 4M3D displacement and 4M1D nodal equivalent stress are simulation results, 4M3D coordinate, 4M connectivity, 4M offsets and 4M element types represent mesh information. C data type means the type of a variable in
the C programming. As can be seen in Table 1, the key datasets of 4M3D displacement and 4M1D nodal equivalent stress cannot be compressed by more than 5% although a certain compression rate of 41% are achieved for the dataset 4M3D coordinate. These results indicate that we urgently need to employ a more effective compression method for compressing the floating-point data of numerical simulation results.

Next, we show the compression rates of JHPCN-DF with different levels of allowed error for the datasets 4M3D displacement, 4M1D nodal equivalent stress, 29M3D displacement and 29M1D nodal equivalent stress in Fig. 3. As can be seen in Fig. 3, JHPCN-DF exceeds the performance of zlib compressor on all of our datasets. With a given allowed error, JHPCN-DF provides stable compression performance on different datasets with different dimensions, DOF, and magnitudes. In the example of the data set 29M1D nodal equivalent stress, we obtained a compression rate of 0.83 with \( \epsilon \) of \( 10^{-2} \) and 0.61 with \( \epsilon \) of \( 10^{-5} \).

![Fig. 3 Compression rates for different data sets in cases of the 4M DOF mesh and the 29M DOF mesh.](image)

Table 2. Output file sizes without JHPCN-DF and with JHPCN-DF for different DOF problems.

<table>
<thead>
<tr>
<th>Compressor</th>
<th>Allowed error ( \epsilon )</th>
<th>File size [MB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>(uncompressed)</td>
<td>-</td>
<td>106</td>
</tr>
<tr>
<td>zlib</td>
<td>-</td>
<td>71</td>
</tr>
<tr>
<td>JHPCN-DF</td>
<td>( 10^{-7} )</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>( 10^{-6} )</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>( 10^{-5} )</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>( 10^{-4} )</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>( 10^{-3} )</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>( 10^{-2} )</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>( 10^{-1} )</td>
<td>38</td>
</tr>
</tbody>
</table>

We compare the sizes of output files with JHPCN-DF against the one without JHPCN-DF. Table 2 shows the size of output file consisting of 6 datasets in reference to Table 1. In one output file, only simulation results (displacement and nodal equivalent stress) are compressed using JHPCN-DF technique since the compression rates of them are too bad as shown in Table 1, while other datasets require lossless compression and are compressed by zlib compressor without JHPCN-DF technique. Here, JHPCN-DF with the allowed error of \( 0 \) is equivalent to zlib compression. As can be seen in Table 2, JHPCN-DF has successfully reduced the disk space for storage. Even with a low allowed error of \( 10^{-5} \), 443 MB reduction of storage space is achieved comparing with uncompressed one for 29M DOF problem.

The contents above report the effect of allowed error on compression rate, and a high allowed error value is preferable if a high compression rate is desired. Besides compression rate the precision of reconstructed results and their visualizations is another important aspect that attract user’s attentions. JHPCN-DF provides allowed error \( \epsilon \) to evaluate the maximum relative error between original and reconstruction and to quantify the precision of reconstructed data. For instance, when \( 10^{-2} \) is selected for allowed error, the maximum relative error of reconstructed data relative to original data is \( 10^{-2} \) according to Eq. (1), and then users can judge if the maximum relative error of 1% is appropriate or not.
Despite the quantitative way mentioned above, the images can be seen to have drastically different perceptual quality, and users can also judge the visualization quality just depending on their perception. An example of visualizations of the Pantheon model is shown in Fig. 4, which plots nodal equivalent stress distribution of the original data and encoded data with different allowed error $\varepsilon$. Fig. 4 (a), Fig. 4 (b) and Fig. 4 (c) seem to have the same perceptual quality, in contrast, Fig. 4 (d) shows a similar trend but low visualization quality. With the increasing of allowed error, more bits at the end of the bit string representation are filled with zeros such that the compression rate is increased whereas values of physical quantities have large round-off error. For example, in case of JHPCN-DF($\varepsilon = 10^{-1}$) for single-precision floating-point number, values of 7.9998 and 7.9999 may become to the same value of 7.5, however, 8.0001 may become to 8.0. Therefore, when $\varepsilon$ is $10^{-1}$, Fig. 4 (d) exhibits low visualization quality and undesirable contouring artifacts. The visualization quality with a given allowed error may differ for different examples as distributions of physical values may be different.

Fig. 4 Visualizations of nodal equivalent stress for the 29M DOF pantheon model with different allowed error $\varepsilon$.

3.3 A spherical cavity model

With consideration for reducing the disk space to store the simulation results, we can directly compress the recorded data just like illustrated in section 3.2 or record selected physical quantities and then calculate other quantities using recorded ones. In this section, we evaluate the effect of JHPCN-DF on the precision of such post-calculations. Fig. 5 shows flowcharts to calculate stress from displacement with JHPCN-DF. In this figure, $K$, $u$, $f$, $B$, $D$, and $\sigma$ are, respectively, the stiffness matrix, the displacement vector, the right-hand-side vector, the strain-displacement matrix, the elasticity matrix, and the stress tensor. We evaluate stresses at the Gauss integration points used in the element stiffness integration rule and then extrapolate to the element node points. Fig. 5 (a) shows that stress is calculated by original displacement data and then reconstructed by JHPCN-DF, and Fig. 5 (b) means that stress is calculated by strain which is calculated from JHPCN-DF reconstructed displacement. In here, since JHPCN-DF reconstructed data are just double-precision variables in computer memory, we could use the usual finite element code for post calculations.

Fig. 5 Flowcharts to calculate stress from displacement with JHPCN-DF.
The evaluation mentioned above is carried out using a spherical cavity model in reference to the book by Timoshenko and Goodier (1970). According to this book, theoretical solution of a problem considering the remote stress distribution around a spherical cavity in an infinite elastic medium is given as follows:

\[
\sigma_{zz} = \sigma_0 \left[1 + \frac{4-5\nu}{2(7-5\nu)} \frac{a^2}{r^2} + \frac{9}{2(7-5\nu)} \frac{a^4}{r^4}\right],
\]  

(3)

where the subscript \(zz\) means in the \(z\) direction and on the plane \(z = 0\), \(\sigma_0\) is the applied uniform tension, \(\nu\) is Poisson ratio of material, \(a\) and \(r\) are, respectively, the radius of the cavity and the distance from origin \(O\). Based on this model, we extract half-quarter of the spherical cavity modal from the infinite body shown in Fig. 6. We set \(l_x = l_y = l_z = 50\,a\) and consider the dimensions of our model are large enough to give almost same solution with the theoretical model mentioned above. We use \(\sigma_{zz}\) as the reference solution and compare our simulation results with it. The finite element mesh which consists of 1,815,814 nodes and 1,325,898 quadratic tetrahedral elements is used for structural analysis, and it is decomposed to 96 parts and each part is decomposed to 160 subdomains for parallel computations.

![Fig. 6 Boundary conditions and material properties of a spherical cavity model (Figures are not to scale).](image)

3.4 Precision of post-run results

To evaluate the difference between the reference solution \(\sigma_{zz}\) and the approximate solution \(\hat{\sigma}_{zz}\), the relative error \(e\) is defined as follows:

\[
e = \frac{|\sigma_{zz} - \hat{\sigma}_{zz}|}{\sigma_{zz}}.
\]  

(4)

First, \(\hat{\sigma}_{zz}\) is calculated by original displacement data and then encoded by JHPCN-DF technique as shown in Fig. 5(a). According to Eq. (3), stress is highly localized at the region near the spherical cavity, therefore, we discuss the results at this important region in the following contents. The plot of stress \(\hat{\sigma}_{zz}\) on \(x\)-axis near the spherical cavity is given in Fig. 7. As can be seen in Fig. 7, obtained approximate solutions show good agreement with the reference solution when the allowed error \(\varepsilon\) is not higher than \(10^{-3}\). With \(\varepsilon\) of \(10^{-2}\), the curve becomes slightly non-smooth and looks like steps, however, it still agrees with the reference one to some extent. When \(\varepsilon\) is increased to \(10^{-1}\), the step-like curve becomes obvious and explains the visualization result shown in Fig. 4(d).

Next, \(\hat{\sigma}_{zz}\) is calculated by JHPCN-DF reconstructed displacement as shown in Fig. 5(b). As can be seen in Fig. 8, when \(\varepsilon\) is not higher than \(10^{-3}\), obtained solutions are in accordance with the reference solution. However, with \(\varepsilon\) of \(10^{-5}\), obvious variability is observed. When \(\varepsilon\) is increased to \(10^{-1}\), the range of variability becomes wider.

Then, we calculate relative error according to Eq. (4) using the stress calculated by the two ways mentioned in Fig. 5. Maximum/average/minimum relative error around the spherical cavity against to different allowed error are shown in Fig. 9. As can be seen in Fig. 9, when \(\varepsilon\) is not higher than \(10^{-4}\), good precision is obtained by using both methods and even much lower minimum relative error is achieved compared with original one. When \(\varepsilon\) is \(10^{-3}\), average relative error and maximum relative error are slightly increasing in both cases. Although the errors keep almost same level compared with original ones, in the case of JHPCN-DF for displacement, average error and maximum error become higher than the ones of JHPCN-DF for stress. When \(\varepsilon\) is further increased to \(10^{-2}\), in the case of JHPCN-DF for displacement, great increasing of maximum error and average error are observed compared with the case of JHPCN-DF for stress. Therefore, in the case that stress is calculated by JHPCN-DF reconstructed displacement, the allowed error is suggested to be not higher than \(10^{-3}\), while in the case that stress is calculated by original displacement and then reconstructed by JHPCN-DF, the allowed error is suggested to be not higher than \(10^{-2}\).

We check the effect of reducing storage space when the method shown in Fig. 5(b) is used. Sizes of files including output data of displacement only and files containing displacement and nodal equivalent stress are summarized in Table 3. As shown in Table 3, since the output data is well compressed by using JHPCN-DF technique, even if we output displacement and nodal equivalent stress, in the case of \(\varepsilon = 10^{-5}\), only 1MB storage space is increased compared with
output displacement only. Therefore, although the way using JHPCN-DF encoded result to calculate other physical quantities to reduce quantity of transmission data is possible, for small model with a few output variables, direct compression using JHPCN-DF is more efficient.

![Fig. 7 Stress $\sigma_{zz}$ on x-axis near a spherical cavity with different allowed error $\varepsilon$ which are used for reconstruction of stress.](image1)

![Fig. 8 Stress $\sigma_{zz}$ on x-axis near a spherical cavity with different allowed error $\varepsilon$ which are used for reconstruction of displacement.](image2)

![Fig. 9 Maximum/average/minimum relative error of stress $\sigma_{zz}$ around a spherical cavity with different allowed error $\varepsilon$ which are used for reconstruction of stress or displacement.](image3)

![Table 3. Sizes of output file including geometric information and one (displacement) or two (displacement and nodal equivalent stress) simulation results with different allowed error $\varepsilon$.](table)

<table>
<thead>
<tr>
<th>Compressor</th>
<th>Allowed error $\varepsilon$</th>
<th>File size [MB]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>displacement</td>
<td>displacement and nodal equivalent stress</td>
</tr>
<tr>
<td>(uncompressed)</td>
<td>-</td>
<td>140</td>
</tr>
<tr>
<td>zlib</td>
<td>-</td>
<td>72</td>
</tr>
<tr>
<td>JHPCN-DF</td>
<td>$10^{-7}$</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>$10^{-6}$</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>$10^{-5}$</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>$10^{-4}$</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>$10^{-3}$</td>
<td>43</td>
</tr>
</tbody>
</table>
4. Conclusions

In this paper, an efficient compression method JHPCN-DF for large scale finite element analysis is introduced and the precision of post-run results by applying JHPCN-DF technique in two ways is validated. The main characteristics of our work are (1) effective compression for large scale finite element simulations, (2) user-determined allowed error for different requirements of precision for visualization and data reconstruction.

Our numerical experiments showed that the compression technique JHPCN-DF outperforms on floating-point datasets. For a 1D double-precision dataset nodal equivalent stress, we obtained a compression rate of 0.83 with the allowed error $\epsilon$ of $10^{-2}$ and 0.68 with $\epsilon$ of $10^{-4}$. With $\epsilon$ of $10^{-4}$, JHPCN-DF has successfully reduced the storage of 464 MB comparing with uncompressed one for the 29M DOF problem. In addition, the precision of post-run results is validated. When $\epsilon$ is not higher than $10^{-4}$, stress calculated by original displacement data and then reconstructed by JHPCN-DF and stress calculated by JHPCN-DF reconstructed displacement data show good agreement with the reference stress and the precisions of stresses calculated by the two ways are almost the same. In the case that stress is calculated by JHPCN-DF reconstructed displacement, the allowed error $\epsilon$ is suggested to be not higher than $10^{-3}$, while in the case that stress is calculated by original displacement and then reconstructed by JHPCN-DF, $\epsilon$ is suggested to be not higher than $10^{-2}$.

Acknowledgements

This research was supported by JST CREST project “Development of a Numerical Library based on Hierarchical Domain Decomposition for Post Petascale Simulation”. One of the authors was also supported by JSPS KAKENHI Grand Number 24760062. This research was partially supported by “Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures” and “High Performance Computing Infrastructure” in Japan.

References


