

An Improved Compression and Storage Algorithm of the Stiffness Matrix of 3D-FEM based on Rider Binary Classification and Negative Sign CSR

Pei Dongmei¹, Meng Fanjun¹ and Wang HaiLong²

1 Computer and Information Engineering College, Inner Mongolia Normal University, Hohhot, China

*2 College of Network Technology Inner Mongolia Normal University, Hohhot, China
ciecpdm@imnu.edu.cn*

Abstract

In order to reduce the data storage and improve data compression ratio of the stiffness matrix of 3D finite element, after analyzed the relationship between nonzero submatrix and generalized adjacent nodes of the stiffness matrix, this paper proposes an improved stiffness matrix compression algorithm, which combined negative sign compressed sparse line and a rider to store binary classification method. Then the improved algorithm is applied to the storage of the stiffness matrix of 3D-FEM. Through experimental simulation, the results show that this method saves a lot of storage space to ensure the validity of data for finite element analysis.

Keywords: *FEM, stiffness matrix, compression algorithm, stored space*

1. Introduction

A finite element method (FEM) in structural mechanics displacement method [1, 2] is now applied to many kinds of fields, such as animation, games, virtual surgery, etc. A FEM usually requires the solution of large equations $Mx = b$, and x, b belong to R_n , where M is the overall stiffness matrix, b is a whole array of equivalent loads. Stiffness matrix usually has up to thousands of orders. With the continued refinement of the grid, the number of orders of overall stiffness matrix may be exacerbated, and the data that needs to be stored will be enormous. And the complexity, uncertainty and unreasonable storage structure of the pretreatment object makes large storage space, leading to slowing the slowness of processing results.

The stiffness matrix of 3D-FEM is commonly large, banded sparse matrix, and nonzero elements of the matrix generally focus on the diagonal area. Now there are some ways to save storage space, such as Half-band storage method, Variable bandwidth storage method et al. These methods can reduce the storage space of the stiffness matrix, but some nonzero elements are stored. It will waste some storage space. These methods are affected by the node number, it is necessary to optimize the number of nodes. It will waste a lot of computing time. In order to further save storage space, one way is the improved compression and storage method of stiffness matrix. BAO [3] presents a improved one-dimensional variable bandwidth compression storage method for sheet metal. This method uses a one-dimensional array of line to save nonzero elements of stiffness matrix, and uses two auxiliary arrays to record each column and the position of the starting element of each column in stiffness matrix array. This method avoids the problem of optimizing the node number and saves a lot of storage space, but the auxiliary array and the storage array of the stiffness matrix have the same length, and it still consume some storage space.

In addition, there are other storage methods of stiffness matrix in recent years, such as Yao [4], Liu [5], and the storage method according to the improved storage format, such as Guo [6], Simecek [7], Neelima [8], Chen [9], and the storage methods for stiffness matrix matrix-vector multiplication, such as Li [10], Martone [11].

This paper proposes a negative sign compressed sparse row (CSR) storage method according to the distribution law of nonzero submatrix. This method only stores the index of nonzero elements and subzero submatrix, which can save a conventional auxiliary array. On this basis, a rider to store binary classification algorithm are added to negative sign CSR storage method. According to the experiment of different storage stiffness matrix of the FEM, the effectiveness of algorithm is given by the simulation results.

2. Storage Method of the Stiffness Matrix based on CSR

2.1 Traditional CSR Storage Method of the Stiffness Matrix

FEM stiffness matrix is usually a large zonal sparse matrix. It can optimize the storage of sparse matrix by adopting the storage of the nonzero elements of sparse matrix. A popular technique for the storage of the sparse matrix is CSR method.

This method will store nonzero elements of stiffness matrix M in a contiguous storage space according to the order of rows. The floating-point array A stores each nonzero elements of matrix, integer array J records the column number of corresponding element, and integer array K records the location of the first element of each row in the array A . The number of nonzero elements of sparse matrix is the length of array A , as the same as the length of array J , and the length of integer array K is dimension of the sparse matrix. The method stores only nonzero elements in the stiffness matrix which achieves the compression and storage to some extents.

2.2 CSR storage Method based on the Distribution Law of Nonzero Submatrix

Due to the more elements of array J in the traditional storage method, Wang [12] proposed a CSR compression and storage algorithm based on nonzero submatrix distribution law. The length of integer array J is same as the floating-point array A in traditional compression method. The array J and A occupy one-third of the storage space, and it will reduce the storage efficiency. But the distributions of nonzero elements in FEM stiffness matrix are regular. The nonzero elements distribute on nonzero submatrix which corresponding with generalized adjacent nodes. So it can determine the location of the nonzero elements indirectly, as long as record the location of the nonzero submatrix.

Based on this method, the compression and storage scheme is as follows: First, array A stores each nonzero elements of the matrix by a sequential row, and integer array J records the column number in the corresponding nonzero submatrix. Then, integer array K records the position of each row of the first nonzero submatrix in the generalized matrix.

Compared to the traditional method, this method can reduce the length of J thus reducing the storage space and improving the utilization of space.

2.3 Negative Sign CSR Storage Method

In the method of Wang [12], the integer array K records the position in each row of the first nonzero submatrix in array J . The length of the array K depends on the orders of generalized matrix, and it will take up some storage space. According to the problems mentioned above, this paper proposes a negative sign CSR storage method to store the stiffness matrix.

In the CSR storage method based on the distribution law of nonzero submatrix, generalized matrix sign the position of each row of the first nonzero submatrix in the J with an array of K . But in the method of a negative sign CSR storage method, if a nonzero

submatrix is the first one in generalized matrix, then J stores the opposite number of corresponding column. Rather, if a nonzero submatrix is not the first one in generalized matrix, then J stores the corresponding column.

That is to say, the position of the first nonzero submatrix of the I-th line in stiffness matrix is the position of the i-th negative in the array J. Here, we assume that the position is D_i . In the CSR storage method based on the distribution law of nonzero submatrix, $K[i]$ means the position of the first nonzero submatrix of the I-th line in the array J. So $K[i]$ can be replaced by D_i , thus saving an auxiliary array K.

2.4 Generation of Stiffness Matrix

Here, we assume that the overall number of a 3D eight nodes hexahedral cell is I [8], and the stiffness matrix of the cell is $S[8 \times 3][8 \times 3]$, so any element of the stiffness matrix can be described as:

$$S_{ijkm} = S[(i-1) \times 3 + k][(j-1) \times 3 + m] \quad (1)$$

where i and j (values between 1 and 8) are the local node number; k and m (values between 1 and 3) are the number of physical dimension. These numbers in the overall stiffness matrix array are determined as follows:

First, we calculate that the overall node numbers of the S_{ijkm} is $I = I[i]$, $J = I[j]$, where I and J are the row and column numbers of the nonzero submatrix MIJ which S_{ijkm} resides in the generalized matrix respectively.

Second, we calculate that the number of nonzero submatrix NI which locate at the front of the MIJ in the generalized matrix is $NI = D_i - 1$, and the number of nonzero elements NeI which locate at the front of the MIJ in the generalized matrix is $NeI = (D_i - 1) \times 9 - (I - 1) \times 3$, where D_i is the $K[i]$, it means the location of the first nonzero submatrix in the first line I of J, $(I - 1) \times 3$ is the reduction of the nonzero element caused by the diagonal matrix of the nonzero submatrix.

Third, we calculate that the number of nonzero elements which locate at the front of line k in line I nonzero submatrix is $NeIk = (D_i - 1) \times 3 \times (k - 1)$, where $D_i - 1$ is the location of the first nonzero submatrix in line I+i of the array J.

Forth, we iterate through array J to find the array label T of $J[T] = J$ from D_i to $D_i - 1$. Then, in the line I, the number of nonzero submatrix NJ before the column J is $T - J[I]$.

Fifth, we calculate that the number label of S_{ijkm} in the line of stiffness matrix is $LeJm = NJ \times 3 - Q + m$, where Q is the reduction of the nonzero element caused by the diagonal matrix of the nonzero submatrix, and $k = 1, 2, 3$ when $Q = 0, 1, 3$.

Sixth, we calculate that the number label of S_{ijkm} in the stiffness matrix array A is $LeIjkm = NeI + NeIk + LeJm$.

Then, the position of S_{ijkm} in stiffness matrix array A can be determined, according to the steps and methods of generation of stiffness matrix, the overall stiffness matrix can be obtained. Then, the number of auxiliary arrays can be reduced by using negative sign CSR storage method, and it can increase the compression rate further.

3. Storage Method based on Rider Binary Classification

3.1 Stiffness Matrix Compression and Storage Method based on Rider Binary Classification

In the FEM stiffness matrix, the share of the different elements of the storage capacity is different. For example, the binary of 126 is 01111110B, requiring eight bits to store, and the binary of 3 is 0011B, needing four bits to store. If the storage space allocated is using uniform binary bits, the high four bits storage space of number 3 will be wasted. Then a rider binary classification storage method has been proposed for different stiffness matrix of finite element. This method only distributes n bits rather than a fixed bit length for a number r. It can further improve the compression rate of the elements of stiffness matrix.

We use this method to store the stiffness matrix array A, the steps are as follows:

First, we determine the number of bits of the absolutely largest element and the absolutely smallest element of stiffness matrix array A, and the number of bits expressed in Z_{max} and Z_{min} respectively.

Second, we assume that $Z_e = Z_{max} - Z_{min} + 1$ (Z_{min} , $Z_{min} + 1$, ... Z_{max}). And use Z_e different address blocks to store the elements of the stiffness matrix A. Each address block stores the element with the same number of bits. Then, we determine the binary bits of Z_e , showed by N_b . The reason for determining the binary number of Z_e is to use the binary number which has the N_b bit fixed-length to distinguish Z_e address blocks.

Third, the address of next operand of each address block (CUESOR $_n$) has been recorded.

Fourth, the first element of A stores in the address block Z_1 , the length of Z_1 is n_1 . Then, the value of CUESOR $_1$ can be defined as CUESOR $_1 = CUESOR_1 + n_1$.

Fifth, the second element of A stores in the address block Z_2 , the length of Z_2 is n_2 . The value of CUESOR $_2$ can be defined as CUESOR $_2 = CUESOR_2 + n_2$, and recording the label of binary of the Z_2 is N_{b2} in Z_1 .

By analogy, the n-th element of array A stores in the address block Z_n , the length of Z_n is n_n . The value of CUESOR $_n$ can be defined as CUESOR $_n = CUESOR_n + n_n$, and recording the label of binary of the Z_n is N_{bn} in Z_{n-1} . Then, Use this method to store all elements of array A, and recording the label of address block of the first element is Z_x in array A.

All address block cursor CUESOR is set to correspond to the first address in the address block.

For example, a storage space, which is stores number 2,3,6,7,8,9, 2 and 3 use three binary bits to store, 6 and 7 use four binary bits to store, 8 and 9 use five binary bits to store.

Table 1. The Storage Format of Address Block

Label of address block	Storage format													
Z_3	0	1	0	0	0	0	1	1	0	1	Indicate that the next data store in Z_4			
Z_4	0	1	1	0	0	1	0	1	1	1	1	0	Indicate that the next data store in Z_5	
Z_5	0	1	0	0	0	1	0	0	1	0	0	1	Store the next label Z_x	

As shown in Table 1, Z_n means storage with n-bit binary, black font indicates data, and red font indicates the binary label of address block of the next data. The binary label of Z_3 , Z_4 and Z_5 are 00B, 01B, and 10B respectively.

3.2 The Stiffness Matrix Storage and Reading Algorithm based on Rider Binary Classification and Negative Sign CSR

In order to further improve the compression ratio of the stiffness matrix of 3D-FEM, this paper combine rider binary classification method and negative signs CSR storage algorithm to store the element of 3D FEM stiffness matrix.

The method for storage of element of stiffness matrix is the same as section 3.1.

In the reading process of element of stiffness matrix, we determine that the negative signs CSR storage algorithm is been used to calculate the sequence number of an arbitrary number of element S_{ijkm} of cell matrix in overall stiffness matrix array, then follow these steps to determine the location of S_{ijkm} in stiffness matrix array:

First, we record that the label of address block of the first element is Z_x in array A.

Second, we read the cursor CUESOR $_x$ of address block which corresponding to the Z_x . The first element Z_x is actually anything between Z_1 and Z_n . And the CUESOR $_x$ can be replaced by CUESOR $_x = CUESOR_x + n_x$.

Third, we read the next element of Z_x , and record the next CUESOR $_x$.

Fourth, repeat two steps above until the position of the number L_{eIJkm} has been determined. We assume that this position is d , the block number of this position is Z_d , and the length of stored data in this address block is nd .

At last the corresponding element of S_{ijkm} in original stiffness matrix A is nd binary numbers which start from d position in Z_d .

Through above analysis, we can determine the location of the S_{ijkm} in Z_e data blocks. It can accumulate the stiffness matrix according to the generative rule, and thus can obtain the overall stiffness matrix.

The flowchart of read data algorithm based on rider binary classification and negative sign CSR is shown in Figure 1.

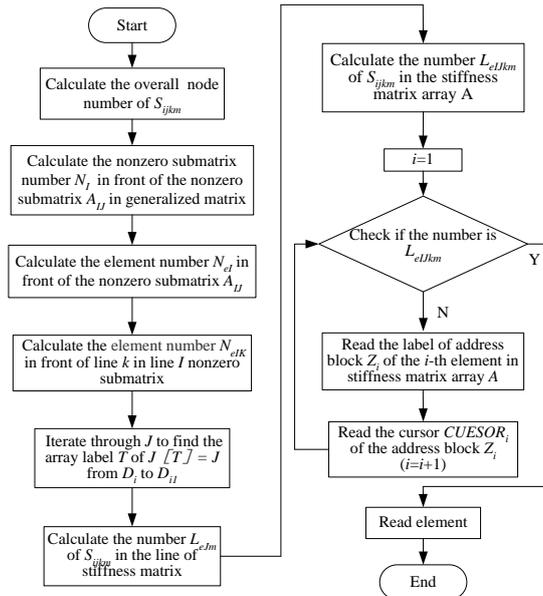


Figure 1. The Flowchart of Read Data Algorithm based on Rider Binary Classification and Negative Sign CSR

4. Algorithm Verification

In order to verify the validity of 3D-FEM stiffness matrix compression and storage algorithm based on rider binary classification and negative sign CSR, this paper use different method to store the element of a stiffness matrix and a beam model. First, the CSR storage method based on the distribution law of nonzero submatrix is used for storage. Second, the negative sign CSR storage method is used for storage. Then, the stiffness matrix storage algorithm based on rider binary classification and negative sign CSR is used for storage. Finally, this paper compares the occupied space of three different algorithms.

4.1 A Stiffness Matrix

Here, we assume that a stiffness matrix is shown in Figure 2. The stored array of the stiffness matrix by using CSR storage method based on the distribution law of nonzero submatrix is shown in Table 2, where the data types of A , J and K are integer. The stored array of the stiffness matrix by using the negative sign CSR storage method is shown in Table 3. The stored array of the stiffness matrix storage algorithm based on rider binary classification and negative sign CSR is shown in Table 4.

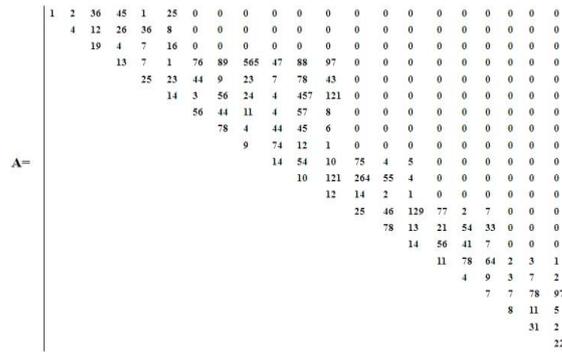


Figure 2. A Stiffness Matrix

Table 2. The Stored Array of the Stiffness Matrix by using CSR Storage Method based on the Distribution Law of Nonzero Submatrix

Array	Storage contents													
A	1	2	36	45	1	25	...	97	8	11	5	31	2	22
J	1	2	1	2	3	1	2	1	2	1	2	1	2	1
K	0	2	5	7	9	11	13							

Table 3. The Stored Array of the Stiffness Matrix by using the Negative Sign CSR Storage Method

Array	Storage contents													
A	1	2	36	45	1	25	...	97	8	11	5	31	2	22
J	-1	2	-1	2	3	-1	2	-1	2	-1	2	-1	2	-1

Table 4. The Stored Array of the Stiffness Matrix Storage Algorithm based on Rider Binary Classification and Negative Sign CSR

Label of address block	Storage format												
Z ₂	0	1	0	0	1	1	...						
Z ₃	0	1	0	0	1	1	0	...					
Z ₄ -Z ₁₀	Storage format are similar to Z ₂ and Z ₃												
Z ₁₁	0	1	0	0	0	1	1	0	1	0	1	Null	

Table 5. The Occupied Space by using Three Different Methods

Stored method	The CSR storage method based on the distribution law of nonzero submatrix	The negative sign CSR storage method	The stiffness matrix storage algorithm based on rider binary classification and negative sign CSR(105*4 means the occupied space of storage the next element)
occupied space (bits)	2016	1904	586+105*4=1006

The occupied space by using three different methods is shown in Table 5. It can be seen from Table 5, the negative sign CSR storage method can save 5.56% occupied space than the CSR storage method based on the distribution law of nonzero submatrix, and the stiffness matrix storage algorithm based on rider binary classification and negative sign

CSR can save 50.1% occupied space than the CSR storage method based on the distribution law of nonzero submatrix.

4.2 Beam Model

The beam model has 807 elements and 2819 nodes, as shown in Figure 3. Here, we assume that the machine word length is 16-bit. Using CSR storage method based on the distribution law of nonzero submatrix, array A occupies $66003 * 16 = 1056048$ bits, array J occupies $7735 * 16 = 123760$ bits, and array K occupies $1204 * 16 = 19264$, then this beam model occupies $1056048 + 123760 + 19264 = 1199072$ bits total.

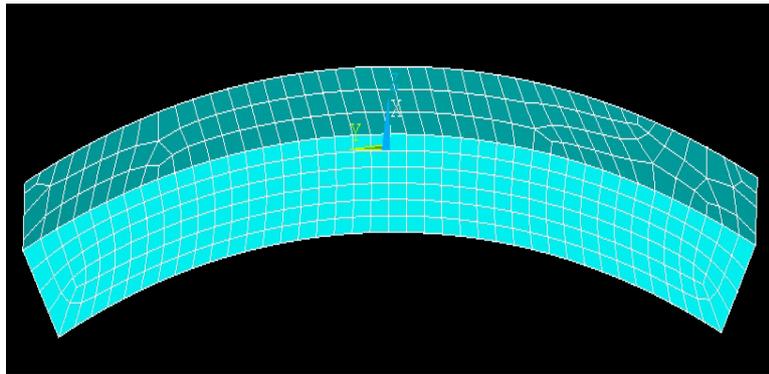


Figure 3. Beam Model

Using negative sign CSR storage method, array A occupies $66003 * 16 = 1056048$ bits. Array J occupies $7735 * 16 = 123760$ bits. Then, this beam model occupies $1056048 + 123760 = 1179808$ bits total.

Similarly, using the stiffness matrix storage algorithm based on rider binary classification and negative sign CSR, the space occupied is $33196 + 322818 + 7735 * 8 = 417894$ bits, and storing binary bits of cursor occupies $66003 * 4 = 264012$ bits, then this beam model occupies $417894 + 264012 = 681904$ bits total.

Occupied storage space of these three methods and the compression ratio are shown in Table 6.

Table 6. Occupied Storage Space of These Three Methods and the Compression Ratio

The machine word length (bits)	The number of elements	The number of nodes	Occupied space (CSR storage method based on the distribution law of nonzero submatrix) (bits)	Occupied space(The negative sign CSR storage method) (bits)	The negative sign CSR storage method achieve the percentage of space savings compared to the CSR storage method based on the distribution law of nonzero submatrix	Occupied space (The stiffness matrix storage algorithm based on rider binary classification and negative sign CSR) (bits)	The stiffness matrix storage algorithm based on rider binary classification and negative sign CSR achieve the percentage of space savings compared to the negative sign CSR method
16	807	2819	1199072	1179808	1.6%	681904	42%
32	807	2819	2398144	2359616	1.6%	681904	71.1%

When the machine word length is 32-bit, the steps for calculating the storage space occupied of the three different methods are the same as 16-bit, and the results of compression ratio are shown in Table 6.

5. Conclusions

This paper proposes an improved stiffness matrix compression algorithm, according to the relationship between nonzero submatrix and generalized adjacent nodes of the stiffness matrix. Compared to CSR storage method based on the distribution law of nonzero submatrix, the negative sign CSR storage method can reduce an auxiliary array, which is used to store the order of generalized stiffness matrix.

Then a rider binary classification storage method has been proposed for different stiffness matrix of 3D-FEM. This method only distributes n bits rather than fixed bit length for a number r of binary length n . It can further improve the compression rate of the elements of the stiffness matrix of 3D-FEM. Finally, combining these two methods, experimental simulation results show that this method saves a lot of storage space to ensure the validity of data for finite element analysis.

ACKNOWLEDGEMENTS

This research was supported by Scientific Research Project of Higher Education of Inner Mongolia Autonomous Region, China (NJZY13052 and NJZY14034).

References

- [1] A. Ghaei and R.M. Movahhedy, "Die design for the radial forging process using 3D FEM", Journal of Materials Processing Technology, vol. 182, no. 3, (2007), pp. 534-539.
- [2] A. B. Behrens, "Finite element analysis of die wear in hot forging processes", CIRP Annals-Manufacturing Technology, vol. 57, no. 1, (2008), pp. 305-308.
- [3] Y. D. Bao, K. G. Du, L. W. Chen, *et al.*, "Solving algorithm for finite element equations in sheet metal forming simulation", Journal of Nanjing University of Aero nautics & Astronautics, vol. 41, no. 5, (2009), pp. 606-610.
- [4] S. Yao and H. Q. Tian, "Compressed storage scheme and assembly procedure of global stiffness matrix in finite element analysis", Zhongnan Daxue Xuebao (Ziran Kexue Ban)/Journal of Central South University (Science and Technology), vol. 37, no. 4, (2006), pp. 826-830.
- [5] C. Liu, J. M. Ye and Y. N. Ma, "Storage and solving of large sparse matrix linear equations", Proceedings - 4th International Conference on Computational and Information Sciences, (2012), pp. 673-677.
- [6] D. H. Guo and W. Gropp, "Applications of the streamed storage format for sparse matrix operations", International Journal of High Performance Computing Applications, vol. 28, no. 1, (2014), pp. 3-12.
- [7] I. Simecek, D. Langr and P. Tvrdik, "Space-efficient sparse matrix storage formats for massively parallel systems", Proceedings of the 14th IEEE International Conference on High Performance Computing and Communications, (2012), pp. 54-60.
- [8] B. Neelima, S. R. Prakash and G. R.Reddy, "New sparse matrix storage format to improve the performance of total SPMV time, Scalable Computing, vol. 13, no. 2, (2012), pp. 159-171.
- [9] L. Chen and G. Y. Wang, "Storage of sparse matrix and linked list of NetCDF data format", World Automation Congress Proceedings, (2012).
- [10] J. J. Li, X. X. Zhang and G. M. Tan, "Study of choosing the optimal storage format of sparse matrix vector multiplication", Jisuanji Yanjiu yu Fazhan/Computer Research and Development, vol. 51, no. 4, (2014), pp. 882-894.
- [11] M. Michele, F. Salvatore, T. Salvatore, P. Marcin and G. Maria, "Utilizing recursive storage in sparse matrix-vector multiplication- preliminary considerations", 25th International Conference on Computers and Their Applications, (2010), pp. 300-305.
- [12] Z. L. Wang, G. Q. Zhao and X. W. Ma, "Compressed storage algorithm of 3D-FEM stiffness matrix", Cailiao Kexue yu Gongyi/Material Science and Technology, vol. 20, no. 2, (2012), pp. 96-100.

Authors



Dongmei Pei, received the BS degrees in computer science from University of Science & Technology Beijing, China, in 1999 and received the MS degrees in computer science from Inner Mongolia Normal University, China, in 2008. Her research interests include Code plagiarism detection, management Information system and compression and storage algorithm.



Meng Fanjun, received the BS and MS degrees in computer science from Inner Mongolia Normal University, China, in 1999 and 2007. Currently, his research interests include scheduling techniques and parallel algorithms for clusters, and also multi-core processors and software techniques for I/O-intensive applications.



Wang Hailong, received the BS in computer science from North Jiaotong University, China, in 1998 and received the MS in computer science from Lanzhou Jiaotong University, China, in 2007. Currently, he is an assistant professor in Computer & Information Engineering College at Inner Mongolia Normal University, China. His research interests include embedded system and multi-core processors and also fault tolerance and real-time database.

