Compression of LS-DYNA™[1] Simulation Results using FEMZIP©[2]

Author:

Clemens-August Thole, Fraunhofer Institute for Algorithms and Scientific Computing

Correspondence:

Fraunhofer Institute for Algorithms and Scientific Computing
Clemens-August Thole
Schloß Birlinghoven
Sankt Augustin, Germany
thole@scai.fraunhofer.de
Tel.: +49 (2241) 14 2739, Fax: +49 (2241) 14 2181

Keywords:

Crash simulation, compression, approximation

[1] LS-DYNA3D is a registered trademark of LSTC, Livermore.
[2] FEMZIP is a registered trademark of Fraunhofer Gesellschaft, München.

Abstract:

The standard usage of simulation as part of the automotive design process has increased the demand for archiving simulation results. Intensive collaboration during the development process requires the exchange of simulation results. Compression of simulation results reduces the size of archives and the time for data transfers. Like compression of video streams and pictures, the effective compression of simulation results requires specific tools which exploit the specific data structures of LS-DYNA3D simulation results and allow for a reduced precision of the results. FEMZIP is especially designed for the compression of crash simulation results and achieves a reduction by factor of 7 for reasonable precision requirements.

Introduction

The usage of computing clusters for simulation tasks in automotive design has substantially increased the available computing power for design departments. As a result, the simulation models include more details which has resulted in models with more than 1 million nodes, and several companies use stochastic validation for the design process. A single result file now consists of several GB and the size of all results computed in a crash department of an automotive company may sum up to 100 TB per year.

Effective data compression would not only reduce the requirements regarding the size of archive storing simulation results but also support the collaboration of distinct development teams by reducing transfer times for simulation results.

Initial results using WINZIP™ [3]

Standard compression methods are not very effective when applied to crash simulation results. The WINZIP™ compression tool was tested on the results of LS-DYNA3D calculation for the DaimlerChrysler Neon model [1]. This model consists of 286023 nodes, 269329 shells, 2908 bricks and 63 beams. The d3plot files for 40 states require 494 Mbytes of disk storage. WINZIP™ reduces this size to 392 Mbytes – a gain of 20%. The WINZIP™ is a general-purpose compression tool and therefore does not exploit the internal data structures of the result files. It exploits the repeated occurrence of the same word by determining appropriate alphabets and storing references to the alphabet instead of the original values. Sequences of the same word are replaced by using appropriate replication counts. LS-DYNA3D, however, stores the simulation results as floating point numbers in binary formats. It is not very likely that the same number is computed at different geometric locations, and therefore alphabet-based compression methods must fail.

Compression algorithms for video streams, pictures and audio streams

Compression methods have been applied to video streams, pictures and audio streams. In the following a short and very basic outline of the compression strategies is provided. (see [2] for more details.) In a first step, the content of the streams is reduced to the necessary minimum. Frequencies, for example, which are out of the range of the human ear, are eliminated. In a second step, the remaining information is approximated using a reduced data base. Examples are the capture of picture motion: A vector stores the object's movement for a set of pixels between two adjacent frames. Based on these vectors, an approximation

[3] WINZIP™ is a registered trademark on the WinZip Computing, Inc.

for the new frame can be computed from the old one. A further common method for the approximation step is the transformation into another basic representation, like the cosines transformations. In this case, a few low-frequency coefficients include the important information, while the coefficients of the higher frequencies just represent noise. In order to guaranty a certain precision of the uncompressed data, the difference between the desired result and the approximation is computed. This difference is small, has little variance, and values like 0 and 1 are often repeated several times. This difference can therefore be compressed using standard methods at a very high compression rate.

Compression methods for meteorological simulation results

So far, specific optimized compression methods have not been used for numerical simulation results. An exception is numerical weather prediction and climate simulation. In order to reduce the size of output files, the data fields are stored in the GRIB format [3], which is a standard of the World Meteorological Organization (WMO). The numbers are stored in the following format:

$$(y - R) * 2^E = X + eps \qquad (1)$$

Here y is the original number, R is a reference vector, E a binary scale factor and X an integer number with a given maximal number of bits. eps is a value less than 1. Stored are the common reference values R and E for a whole field and the value of X for each number. A typical length of X is 16 bits which results in a reduction of about 4 compared to the original 64 bit data length. These data fields have been furthermore compressed by Steffen and Wang using Wavelet approximations and standard gzip-type approximation for the difference between original and approximation. An additional compression factor between 3 and 8 is reported[4].

FEMZIP compression algorithms for LS-DYNA simulation results

The algorithms developed by Steffen and Wang show, that compression methods can be successfully applied also to simulation results. They cannot be used directly for crash simulation because Wavelet transformations cannot be applied to grid functions on unstructured shell element grids. The Fraunhofer FEMZIP tool therefore combines a number of new algorithms to compress simulation results. The general approach as well as the detailed algorithms are subject of a pending patent of Fraunhofer. FEMZIP implements the three basic steps:

- Quantization
- Approximation
- Coding of the residual

Quantization

For the quantization FEMZIP requires the basic precision for each of the grid functions contained in the simulation results. Due to approximation errors on one hand and the stochastic nature of the simulation results on the other hand[5], the precision of the simulation results is limited and far less than the 32 bit floating point numbers in the result files suggest. In addition to the results, the engineer requires only a certain precision for the decision on a good design. Therefore, it is acceptable to compress the simulation results with some loss in precision.

The quantization is either provided by the user of FEMZIP as absolute values or in relation to the actual size of the grid functions. If no values are given, standard values for relative accuracy compared to the maximal value of the grid function are used.

In some cases it would be desirable to transform the values of the grid function before quantization, in order to allow for good approximation in certain areas of the value distribution and less precision in other areas. The effective plastic strain would be a candidate, because values close to zero require rather small intervals. In certain areas plastic strain might take rather huge values which do not need to be stores with the same tiny precision as required for values close to zero.

Approximation of simulation results

FEMZIP combines various methods to approximate the grid function. The first principle is that the difference between simulation results at different time steps mostly differ by rigid body modes. Furthermore, the actual velocities are used to approximate the values at the new time steps. In addition FEMZIP implements a hierarchical approximation based on AMG coarsening [6].

Arithmetic encoding of the residual

As a result of the quantization and the approximation the difference between the results of the quantization and the approximation is of type integer and small. Arithmetic encoding with adaptive word length, detection of sequences of identical values and special treatment of outliers is used to compress this difference.

Results

defines basic increments for several grid functions of the simulation results for a fine, medium and coarse precision. Between 9 and 19 bits are required to represent the results.

	Original	fine		medium		coarse	
		Increment	Bits	Increment	Bits	Increment	Bits
geometry	5000	0,01	18,93	0,1	15,61	1	12,29
velocity	300000	5	15,87	50	12,55	500	9,23
Acce-leration	50000000000	100000	18,93	1000000	15,61	10000000	12,29

Table 1: Increments for quantisation and related size of the required word length for the representation of the maximal values of the grid function for a fine, medium and coarse approximation

shows, in addition, the size of the compressed data sets for each of the grid functions and the related average number of bits per data item. As long as the size of the compressed data sets is not tiny, the gain in reduction by approximation and coding is more or less independent of the required accuracy and varies between 9 and 12 Bits. This implies that the reduction of precision requirements may lead to super linear improvement of the reduction of the size of the compressed data sets.

	original	fine Increment	Bits	medium Increment	Bits	coarse Increment	Bits
geometry	5000	0,01	18,93	0,1	15,61	1	12,29
velocity	300000	5	15,87	50	12,55	500	9,23
accelleration	50000000000	100000	18,93	1000000	15,61	10000000	12,29
	Size	Size of compressed data sets					
geometry	36038898	68689□	6,10	3909238	3,47	2628123	2,33
velocity	36038898	6903009	6,13	3701604	3,29	1990007	1,77
accelleration	36038898	9593770	8,52	5345861	4,75	2410367	2,14
		Δ Bits		Δ Bits		Δ Bits	
geometry			12,83		12,14		9,95
velocity			9,74		9,26		7,46
accelleration			10,41		10,86		10,15

Table 2: Compression of several grid functions depending on several precisions

Figure 1 compares the original size of the simulation results, the size of the dataset compressed by WINZIP™ and the size of the compressed data set using fine, medium and coarse precision. In this case, the compression factor varies between 5 and 10.

Summary

The example has shown that effective compression can be achieved for crash simulation results. There is a price to pay. Compression requires computing time. Currently 110 sec are required on a Pentium IV processor (2 Ghz) for data compression of the complete data sets and 70 sec for the uncompression. In future versions of this compression tool, a substantial reduction of the time for data compression is expected.

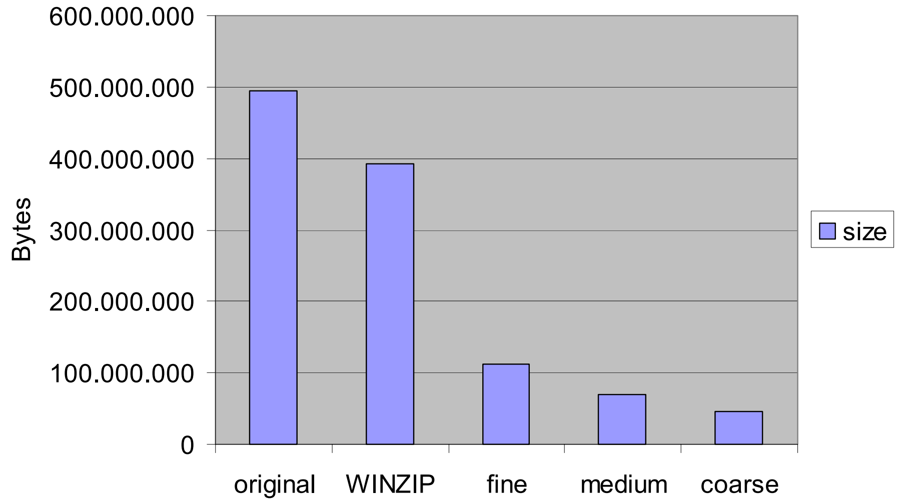

Figure 1: Required size of the compressed data sets using fine, medium and coarse precision

[1] George Washington University, Virginia, USA. National Crash Analysis Center. Finite element model archive. Dodge Neon. http://www.ncac.gwu.edu/vml/models.html

[2] T.Sikora, "MPEG Digital Video Coding Standards", In Digital Electronics Consumer Handbook, McGraw Hill Company, Ed. R.Jurgens, 1997.

[3] National Weather Service 1997: *GRIB1* Edition 1. http://weather.unisys.com/wxp/Appendices /Formats/GRIB.html

[4] E. Steffen, N. Wang: Weather Data Compression. In 19th Int. Conf. on Interactive Information Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, Long Beach, CA, Amer. Meteor. Soc., CD-ROM, 4.9.

[5] C.A. Thole, M. Liquan: Reason for Scatter in Simulation Results. In Proceedings of the 4th European LS-DYNA user conferende, Ulm 2003. http://www.dynamore.de/download/eu03/papers/B-III/LS-DYNA_ULM_B-III-11.pdf.

[6] Ruge, J.W. and Stüben, K.: Algebraic multigrid. In McCormick, S.F., editor, *SIAM Frontiers on Applied Mathematics, Volume 3, Multigrid Methods*, pages 73--130. SIAM 1987.