Scatter Analysis of Crash Simulation Results Enabled by Data Compression

Clemens-August Thole
Rodrigo Iza-Teran
Rudolph Lorentz
Helmut Schwamborn
Fraunhofer Institute for Algorithms and Scientific Computing
Schloß Birlinghoven, D-53754 Sankt Augustin, Germany
Clemens-August.Thole@scai.fraunhofer.de

Abstract

In crash simulation, small changes of the model or boundary conditions may result in substantial changes of the simulation results. For the Neon test case [3], small variations of the barrier position result in substantial scatter of the intrusion. Detailed investigations of several models have shown that in some cases numerical effects might be responsible for the scatter in the results. In most cases, however, the instable behaviour of the simulation results is caused by bifurcations. These bifurcations result from numerical algorithms or are a feature of the car design. In the Neon model the scatter is a result of the interaction between the axle and the engine block.

DIFF-CRASH[1] is a tool, which allows one to measure scatter and to trace this scatter back to its origin. It allows the engineer, to understand the mechanisms of propagation and amplification of scatter during the crash itself as a basis for the improvement of the stability of the car design. For this analysis, DIFF-CRASH uses the complete result files of several simulation runs with a fine time resolution of the states.

Storing these result files requires a substantial amount of disk space. FEMzip[2] allows the reduction of this disk space by a factor of between 5 and 10. One can then store not only key results but the complete result files from optimisation experiments which can also be used for stability analysis. In this paper we discuss the accuracy required by DIFF-CRASH for a precise analysis and its implication on the data compression performance of FEMzip.

Compression of Simulation Data

The increased computing power using PC-Clusters has substantially increased the simulation capabilities of design departments. As a result, much finer models are used for simulation as well as stochastic validation of crash simulation results is investigated. Stochastic validation requires several 10s of simulation runs of similar models. A single result file now consists of several GBytes. The size of all results computed in a crash department of an automotive company may sum up to several 100 TBytes per year.

Effective data compression would allow storing these simulation results and enhancing the handling of these large sets of result files. Comparing the results of stochastic validation not only for some key values but also on the basis of geometry will allow one to obtain additional insight into the nature of the stochastic behaviour of the current model [1].

[1] DIFF-CRASH is a registered trademark of Fraunhofer Gesellschaft, Munich.
[2] FEMzip is a registered trademark of Fraunhofer Gesellschaft, Munich.

In [2] data compression for LS-DYNA[3] simulation results using FEMzip was already introduced. In this paper a basic overview of the functionality and background is provided, in order to introduce the detailed analysis on the relation between required accuracy and compression factors.

Initial results using WINZIP™ [4] When applied to crash simulation results standard compression methods are not very effective. The WINZIP™ compression tool was tested on the results of LS-DYNA calculation for the DaimlerChrysler Neon model [3]. 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%. 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 constructing a dictionary and storing references to the dictionary instead of the original values. LS-DYNA, 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 dictionary-based compression methods must fail.

State of the art of compression methods for simulation results. The field of compression of floating point data is, as of the moment, underdeveloped. To our knowledge, there is no commercially available software for this general purpose. If we restrict our consideration to the compression of formatted floating point data, then we know of two such software packages: FEMzip for LSDyna/Pamcrash data and GRIBzip for the compression of meteorological data in the GRIB format.
On the other hand, the subject itself has been tackled in several papers. Engelson, Fritzson and Fritzson [4], Isenburg, Lindstrom and Snoeyink [5], and Ratanaworabhan, Ke and Burtscher [6] losslessly compress floating point data by treating the IEEE representation of floating point data as integers. Cabrerra compresses 3D meteorological data by first applying the Karhunen-Loeve transform and then compressing the 2D-slices with JPEG2000. Finally, Steffen, Wang and Brummer [7,8] have a series of papers on the lossless compression of GRIB data utilising lossy JPEG compression and a difference file.

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 [5]. FEMzip implements the three basic steps:

- Quantization
- Approximation
- Coding of the residual

[3] LS-DYNA is a registered trademark of Livermore Software Technology Corporation.
[4] WINZIP™ is a registered trademark on the WinZip Computing, Inc.
[5] The general approach as well as the detailed algorithms are subject of the national patent 103 31 431of Fraunhofer Gesellschaft, Munich. Acceptance of the patent on an international level is pending.

Quantization. Quantization means the representation of the data elements as multiples of a basic increment. Quantization therefore reduces the actual precision of the results. For the quantization, FEMzip requires the basic increment for each of the grid functions contained in the simulation results. Therefore, the simulation results are compressed with some loss in precision. The quantization is provided by the user of FEMzip either as absolute values or in relation to the current size of the grid functions.

Approximation of simulation data. FEMzip combines various methods to approximate the grid functions. The first principle is that the difference between simulation results at adjacent time steps is dominated by rigid body modes. Furthermore, the current velocities can be used to approximate the values at the new time steps. In addition, FEMzip implements a hierarchical interpolation based on AMG coarsening [[9]].

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. The method of "suppression of leading zeros" with adaptive word length, detection of sequences of identical values and special treatment of outliers is used to compress this difference.

Results of the Compression

Table 1 shows the maximal and minimal values of typical components from the NEON d3plot result files. Node position, velocity and acceleration are provided per node, sigma and effective plastic strain for each of the 2908 bricks and axial force for each of the 63 beams. Due to the strong variation in the span of the different components, the basic increment must be provided separately for each basic component.

	Minimal value	Maximal value
node positions	-13597	19553
velocity	-189685	218134
acceleration	-5,3082E+10	5,2123E+10
sigma	-148,07	125,32
effective plastic strain	-0,1342	1,0000
axial force	-64012	32522

Table 1: Minimal and maximal values of representative components in the NEON d3plot result files

Table 2 summarizes the size of the different contributions to the d3plot output files for each component. Here it turns out, that the overall size is dominated by those components, which are computed for each node. However, the specific output file contains only element results for solids and not for shells. This relation changes, if shell element results are included. Currently for the analysis using DIFF-CRASH only node positions are used and therefore this test case is representative.

	Size (Kbyte)
node positions	140777
velocity	140777
acceleration	140777
sigma	2863
effective plastic strain	477
axial force	10

Table 2: Size of the respective components in the d3plot output files (sum of all states)

Figure 1 shows the relative size of the compressed d3plot component in relation to the original contribution for different accuracy. The x-axis provides the relative size of the basic increment for each component in relation to the maximal absolute value of this component. For coordinates, an extended precision is required and, in addition, the basic increment is multiplied by 0.01. Table 3 lists the concrete size of the increment for the different components and relative size of 0.0001. For the coordinates, this implies an accuracy of the results of 0.02 mm.

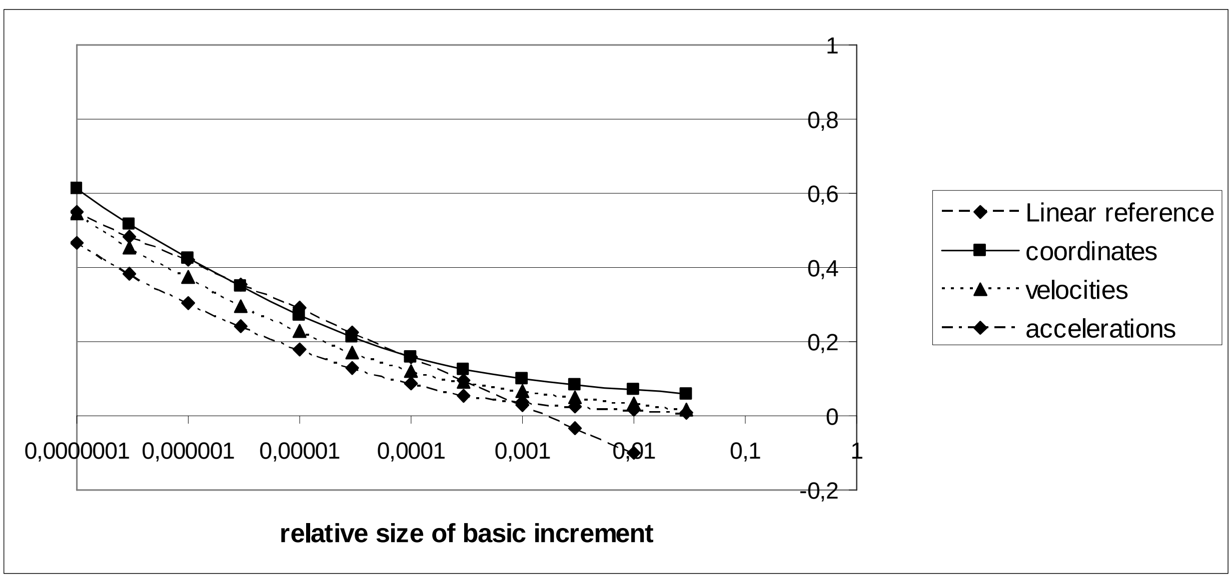

Figure 1: Relative size of the compressed d3plot component in relation to the original data size

	Size of basic increment
node positions	0,0195
velocity	21,8100
acceleration	5308000,0000
sigma	0,0148
effective plast	0,0001
axial force	6,4010

Table 3: Size of basic increment (precision) for the relative accuracy of 0.0001

Figure 1 shows that due to the approximation, the data volume can be reduced by 50% even for unrealistically small basic increments. Quantization improves the size reduction almost linearly in relation to the logarithm of the required precision. Each order of magnitude less precision reduces the file size by an additional 13% of the original data. This linear behaviour ends, when the size of the components reaches about 10% of its original size resulting in a compression factor of 10. Even larger compression factors can be achieved at a lower reduction rate, if the required precision is further reduced.

Impact of Compression on Stability Analysis

Stable crash behaviour of a car model is a design target for the following reasons:

- Simulation results might be misleading, when the impact of changes of the model or model parameters is investigated.
- The numerical model is always only an idealized representation of the real car design. Stable crash behaviour simplifies the prediction of the crash behaviour of the real car from simulation results for the idealized model.
- Smaller bounds for the scattering of the characteristic crash values will improve the possibilities of the engineer to find the best compromise for the car design with respect to the targets of the different load cases.

In [1] a detailed stability analysis of the crash behaviour of the Neon Model is provided. Using statistical methods comparing all geometry positions from different runs, the interaction between the engine block and the axle could be identified as a major reason for the instable behaviour of the footwell intrusion.

The actual scatter of the node positions at state 14 in the area of axle and engine block is about 6 mm. Therefore a precision of 0.1 mm is more than sufficient to perform the required statistical analysis. The relative precision of 0.0003 is sufficient to guarantee these results, leading to a factor of more than 9.26 in the reduction of data space requirements for the node components.

Summary

The example has shown that effective compression can be achieved for crash simulation results. Even if the data results are post processed by statistical methods for scatter analysis, a factor of about 10 can be achieved in the reduction of the node components in the output files. This allows the storage of all simulation results from a stochastic analysis using DIFF-CRASH exploiting geometry information instead of only extracting and analysing only key results.

References

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

[2] Thole, C. A.: Compression of LS-DYNA™ simulation results using FEMzip. In Proceedings of the 3rd LS-DYNA Anwenderforum, Bamberg 2004. http://www.dynamore.de/download/af04/papers/E-III-4.pdf.

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

[4] Engelson, V., Fritzon, D. and Fritzon, P.: Lossless compression of high-volume numerical data from simulations, Data Compression Conference, 2000, 574 – 586.

[5] Isenburg, M., Lindstrom, P. and Snoeyink, J.: Compression of floating point geometry, CAD2004, 495 – 502.

[6] Ratanaworabhan, P., Ke, J. and Burtscher, M.: Fast lossless compression of scientific floating point data, 2006 Data Compression Conference.
[7] Steffen, C. E., Wang, N.: Weather data Compression, Proceedings of the 19th International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, American Meteorology Association, 2003.
[8] Wang, N. and Brummer, R.: Experiment of a wavelet-based data compression technique with precision control, Proceedings of the 19th International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, American Meteorology Association, 2003.
[9] 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.