Reasons for Scatter in Crash Simulation Results

Authors:

Clemens-August Thole, Liquan Mei
Fraunhofer Institute for Algorithms and Scientific Computing,
Sankt Augustin, Germany

Correspondence:

Clemens-August Thole
Fraunhofer Institute for Algorithms and Scientific Computing
Schloß Birlinghoven
D-53754 Sankt Augustin
Germany

Tel: +49-(0)2241-14-2739
Fax: +49-(0)2241-14-2181
e-mail: Clemens-August.Thole@scai.fhg.de

Keywords:
Stability analysis, nonlinear structural mechanics,
finite element modeling, crashworthiness analysis

ABSTRACT

In crash simulation, small changes of the model or boundary conditions may result in substantial changes of the simulation results. For a BMW car model, the node positions of the crashed model show differences of up to 14 cm between several executions on a parallel machine for the same input deck. For the Dodge Neon testcase, 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. In the case of the BMW car model, the scattering of the simulation results is a direct consequence of buckling of the longitudinal rail. A slight redesign of this part causes stable results for parallel machines.

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. A 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.

Due to the nature of crash simulation many parts might show instable behaviour. Usually, only a small subset has a real impact on those values, which measure the crash behaviour (like intrusion). Measuring the scatter of simulation results for these characteristic values is a first step. In order to improve the design, it is necessary to trace this scatter back to its origin in space and time.

DIFFCRASH 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.

INTRODUCTION – Stability Observations

Nowadays the car manufacturing industry relies heavily on simulation results. By simulation the number of real prototypes is reduced, the insight into the features of the actual design is increased and the turn-around time between model changes is much shorter than in the case of real tests. Numerical crash simulation is the most computer-time consuming simulation task in car design. Therefore it is obvious, that crash simulation codes were among the first industrial simulation codes, which were ported onto parallel distributed memory architectures during the EUROPORT project[1].

[1] The EUROPORT Project was funded by the European Commission as part of the Esprit programme.[4] (1994-96)

Using the mpp-versions of industrial crash simulation codes the engineers made a surprising discovery for certain models: The result of numerical simulation changed from one parallel execution to the next by more than 10 cm for the node positions, although the input decks and the simulation parameters were identical. Figure 1 shows for model provided by BMW consisting of about 60.000 shell elements the maximal and average differences between several simulation runs. This observation has stopped car manufacturing companies from using mpp-system for crash simulation for several years.

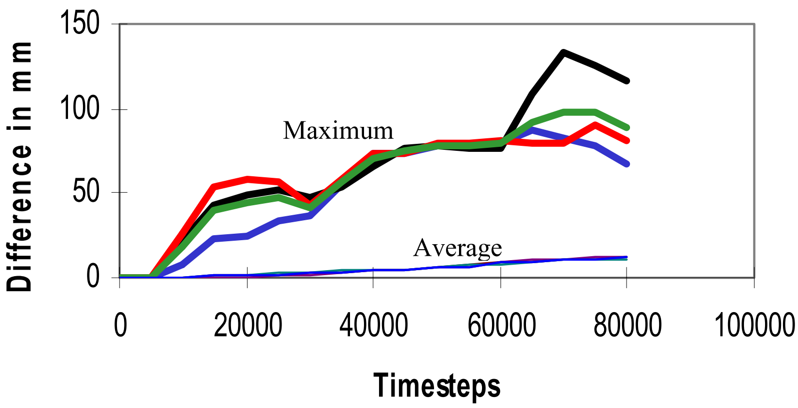

Figure 1 Difference between several simulation runs of a BMW model after a 40% offset crash using PAM-CRASH on a 32 node IBM SP2 using the same input deck

As part of the PROMENVIR project[2] a stochastic analysis tool was developed (now named ST-ORM), which automatically changes certain parameters of input deck, performs simulations, extracts a set of parameters of the results and analyses the dependency between input parameters and result parameters. Using PROMENVIR it was possible to show that small changes in the input deck may result in substantial changes of the simulation results and no correlation between changes and results may be present.

As part of the AUTOBENCH Project[3] and the AUTO-OPT Project[4] the reasons for the scatter of the results were investigated in detail. It turned out that numerical properties of the simulation codes as well as certain features of the car design may be responsible for the "butterfly effects". Typical sources of instabilities are buckling and contact of different parts under an angle of 90° and deficiencies of the contact search algorithms. After a short introduction of the analysis tool DIFF-CRASH, which was developed for this stability analysis, some of the results are discussed in more detail.

[2] The PROMENVIR Project [3] was funded by the European Commission as part of the Esprit programme (1996-97).
[3] The AUTOBENCH Project [5] was funded by the German Minister for Education and Research (BMB+F). (1998-01)
[4] The AUTO-OPT Project is funded by the German Minister for Education and Research (BMB+F). (2002-05)

DIFF-CRASH Overview

For a detailed investigation of the stability of simulation results, DIFF-CRASH performs statistical operations on the binary output files of LS-DYNA3D and PAM-CRASH and generates additional values per point and time step. These results are added as scalar or vector functions to the binary output files and may be visualized using the code specific tools LS-POST or PAMVIEW as well as the GNS-Animator visualization tool.

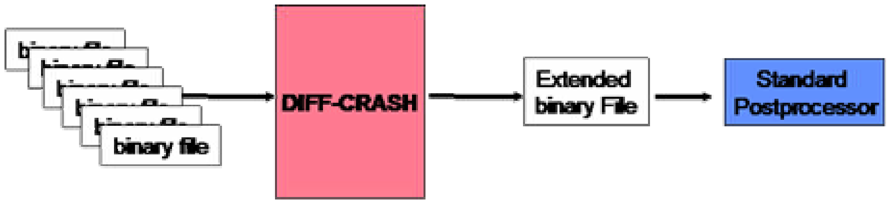

Figure 2: DIFF-CRASH processing path

DIFF-CRASH supports several analysis functions, which are detailed in [1] and [6]. He we use the following functionals:

- PD3MX - for each time step and node, PD3MX measures the maximal difference of the node positions between several runs.
- PD3IJ – for each time step and node, PD3IJ provides the indices of the two most distant simulation runs.
- SIM – for each time step and node, SIM provides a measure for the similarity of the cloud of node positions to a reference cloud of a specific node and time step.
- SIMCluster – for each time step and node, SIMCluster provides the index of a cluster of nodes with similar clouds of node positions.

PD3MX is used to measure scatter and to find areas, which need to be investigated. The SIM and SIMCluster functionals are used to trace the scatter from these areas back to its origin.

Testcases

In this paper results from two different models are reported. The first model was

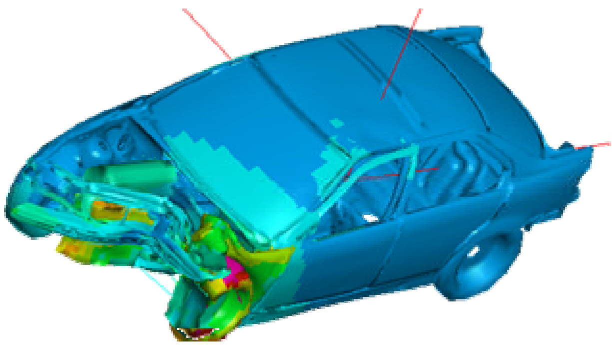

Figure 3: PD3MX as color coding on the BMW test case showing differences of more than 10 cm between several runs.

provided by BMW. It contains about 60.000 elements, is crashed at 30 mph against a fixed wall. Figure 3 shows the functional PD3MX as color coding on the body at 80 ms. 16 runs were performed using mpp-PAM-CRASH 96 on an IBM SP-2 and the same input deck without any variation.

The second model is the publically available Dodge Neon model with 286023 elements, 40 % offset crash at 30 mph. 12 simulations have been performed with a variation of the position of the barrier by about 20 mm.

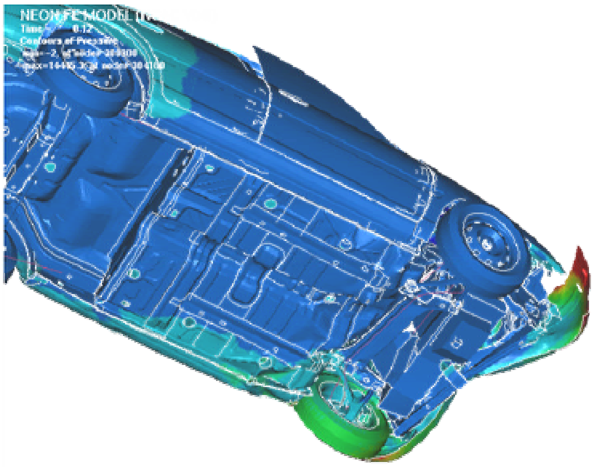

Figure 4: PD3MX as color coding on the Neon testcase showing differences of up to 10 cm in node positions between different simulation runs.

Reason for instable simulation results

Instable simulation results in this context means, that a small or tiny variation in the model or its simulation causes substantial differences in the results. During the investigations it turned out, that parallel execution generates small differences between the simulation runs (if the pipe or stabilization options are not used). Parallel execution implies, that each processor is responsible for a certain part of the model. At interfaces between several processors, the sequence for the summation of the simulation results is not fixed. Different sequences may lead to small differences for example of force calculations at interface nodes. However, it turned out, that other effects than numerical round-off errors magnify these differences by several orders of magnitude.

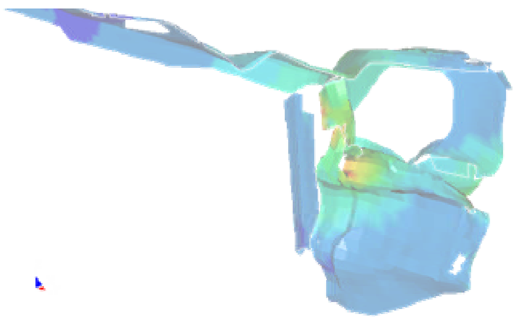

Figure 5: PD3MX as color coding on a part of the Neon model at 21 ms shows differences of up to 3 cm.

One of these effects is contact search. Figure 5 shows scatter of node positions of 3 cm at a certain part just behind the frontal bumper of the Neon model. The DIFF-CRASH functional PD3IJ allows to identify the most extrem runs in this area.

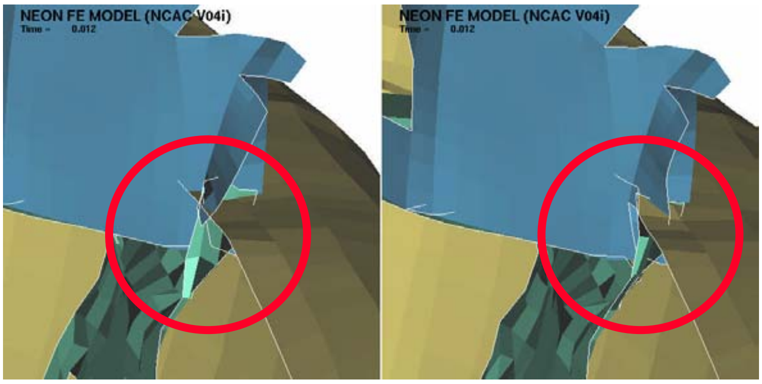

Figure 6: Behaviour of Neon part 377 in two different simulation runs at 12 ms.

Figure 6 compares these two extreme simulation runs. It is obvious, that the contact algorithms has failed in this complex situation. The penetration of the bumper is tried to be resolved in two geometrically different ways. Similar problems have been observed also using PAM-CRASH for the BMW model with contact type 36. In both cases, the failures of the contact algorithms have no essential impact on the scatter of the intrusion.

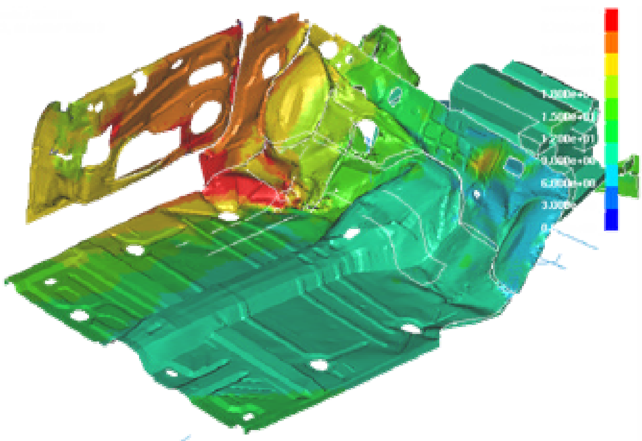

Figure 7: PD3MX as color coding on the footarea of the Neon Model showing scatter of simulation results of more than 3 cm at 114 ms.

Figure 7 shows a scatter of the intrusion of more than 3 cm at 114 ms. The cluster functional at 114 ms (Figure 8) shows the area of similar behaviour of the scatter of node positions. Therefore this scatter is not a result of the variation of the position of the barrier. This change introduces small differences, which trigger a major instability close to the rear part of the longitudinal rail.

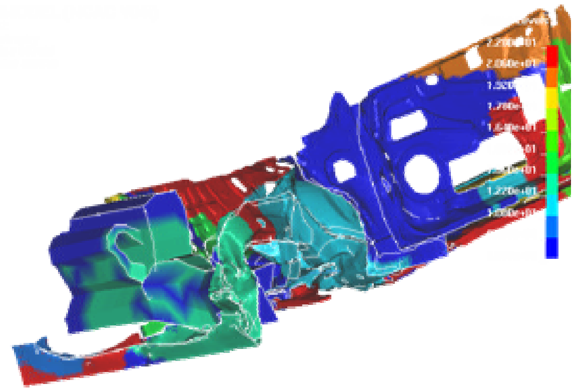

Figure 8: The SIMCluster functional shows areas of similar scatter on the body of the Neon model. All nodes of such an area share the same source of instability.

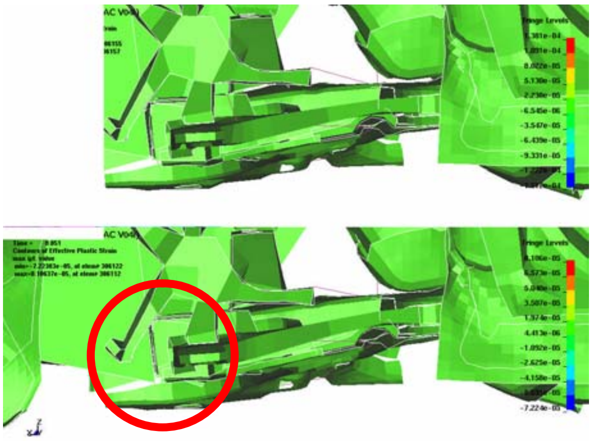

Figure 9: Comparison of the two most extreme runs at time 51 ms of the Neon model.

Figure 9 shows a comparison of the two most extreme runs identified by PD3IJ. In the red circled area the engine block hits the axle with a slightly different position in height. The shape of the engine block in this area causes the axle to slip up the engine block in a different way. This has substantial impact on the way, how the axle impacts the drivers foot area.

Figure 10 shows SIMCluster for the BMW testcase at 80 ms. The cluster of nodes in the drivers foot area is strongly related to an area an the longitudinal rail at 28 ms (Figure 11). Using PD3IJ it is possible to identify two modes of the crash behaviour of the longitudinal rail (Figure 12). Different changes in the area identified in Figure 11 result in different models (F, H, and 1003/1004 models), each of which shows much more stable behaviour than the original model on parallel machines. For a substantial stabilisation, however, a redesign of the longitudinal rail is required.

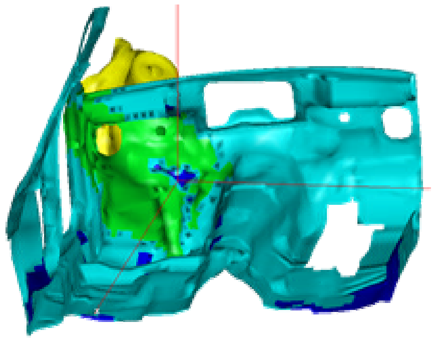

Figure 10: SIMCluster function for the BMW model at 80 ms identifies two clusters, each of which has the same reason for instability.

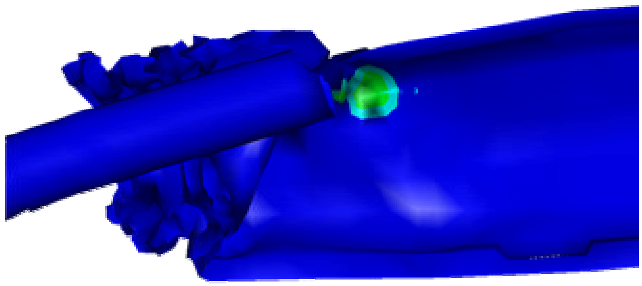

Figure 11: SIMCluster functional at 28 ms on a specific part of the longitudinal rail.

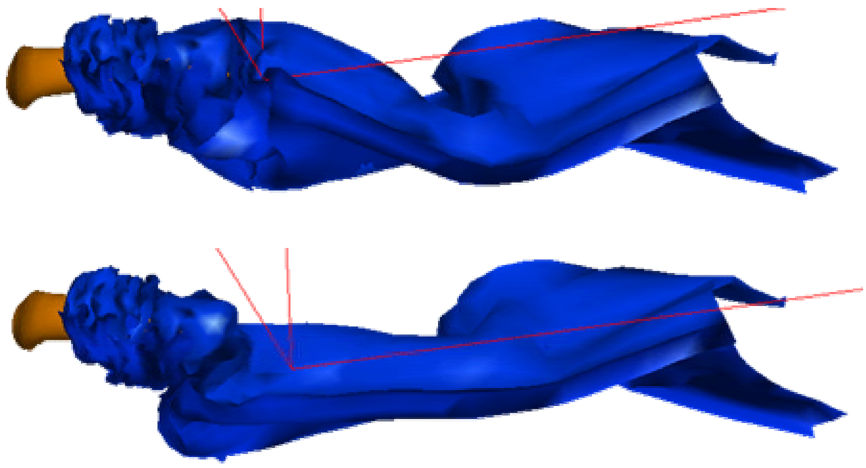

Figure 12: Two different modes of the crash behaviour of the longitudinal rail.

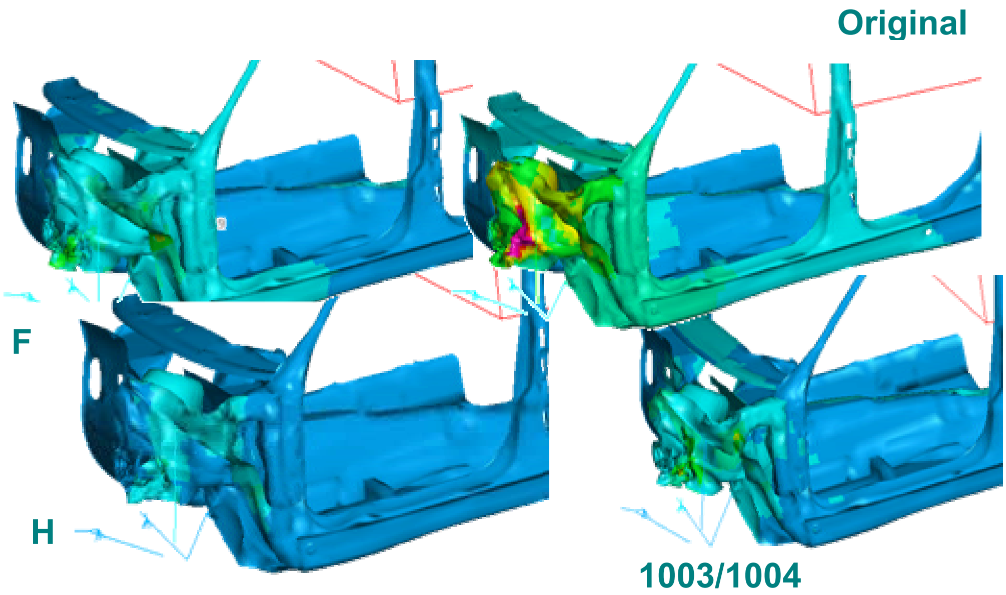

Figure 13: Scatter of several variants of the original model showing a substantial improved behaviour.

Summary and Conclusions

Using DIFF-CRASH, contact search problems, buckling and <Hebel> could be identified as numerical properties or physical features of the concrete model, which magnify noise in the simulation results. For the BMW model, it was possible to achieve a more stable design by small changes of the model.

Although parallel computing was used to generate variants for the BMW testcase, this approach is usually not appropriate. In order to investigate the stability of a model, features like the barrier position, material thickness of several parts or even node positions should be randomly disturbed within the bounds of the typical variations of these parameters in the production process or the concrete tests. Only, if the simulation results are stable under these conditions, the impact of design changes can be predicted by simulation without a complete stochastic analysis.

References

1. BENDISCH, J.; VON TROTHA, H.: Investigation on car stability in crash simulations. In ATTCE Proceedings Volume 1 Safety, SAE International (2001)

2. JAIN, A.K.; DUBES, R.C.: Algorithms for Clustering Data, Prentice Hall (1988)

3. MARCZYK, J.: Principles of Simulation-Based Computer-Aided Engineering. FIM Publications, Barcelona (1999)

4. MIERENDORFF, H.; STUEBEN, K.; THOMAS, O.: EUROPORT-1 Porting industrial codes to parallel architectures. In: Hertzberger, B.; Serazzi, G.

(eds.); High Performance Computing and Networking, Lecture Notes in Computer Science, Number 919, Springer, Berlin, Heidelberg (1995)

5. THOLE, C., KOLIBAL, S.; WOLF, K.: AUTOBENCH - Virtual prototypes for automotive industry. In Deville, M.; Owen, R. (eds): 16th IMACS World Congress 2000 Proceedings, IMACS, Rutgers University, New Brunswick (2000)

6. THOLE, C.A.; MEI, L.: Clustering Algorithms for Parallel Car-Crash Simulation Analysis. Proceedings of High Performance Computing Conference 2003, Hanoi, submitted.