

Integrating Theory, Experiments, and FEA to Solve Challenging Nonlinear Mechanics Problems

Ted Diehl

DuPont Engineering Technology, Chestnut Run, Bldg 722, Wilmington, DE 19880-0722
Ted.Diehl@USA.DuPont.com

Bodie Technology, Inc., 103 Mougins Circle, Kennett Square, PA 19348
Ted.Diehl@BodieTech.com

Abstract: *Solutions to challenging structural mechanics problems often rely on approaches that properly integrate theoretical analysis, experiments, and finite element simulation, along with judgment and practical approximations. Combining these elements is important not only for the original methods development of a given problem class, but also for repeated applications of such methods. This paper describes how multiple software solutions, namely Abaqus, Mathcad®, and Kornucopia™ have been blended together to create robust approaches to address several challenging problems in nonlinear mechanics. The examples highlighted will include the evaluation of performance characteristics for a nonwoven envelope under inflation loads to burst, “methods development” for simulating peeling mechanics and failure, and the efficient analysis of tedious shock acceleration post-processing calculations for models with a large number of output sensors. Techniques discussed will include creation of an average nonlinear material description from measured data containing large variations, the salvaging and improving of “non-ideal” experimental data, and additional post-FE analysis that can significantly improve analysis efficiency and/or results interpretation.*

Keywords: *Nonlinear FEA, Averaging Multiple Data Curves, Applied DSP (Digital Signal Processing), Filtering, Smoothing, Healing, Extrapolation, Failure, Nonwoven Fabric, Energy Methods, Implicit Finite Element Analysis, Explicit Dynamics, Abaqus, Mathcad, Kornucopia.*

1. Introduction

In many cases, finite element analysis (FEA) alone is not sufficient to create the needed methodology for addressing challenging problems in industry and research. Solutions in these cases require the analyst to augment their FEA with one or more of the following: theoretical mechanics analyses, experimental data, and/or additional pre/post processing calculations. These additional requirements, beyond just FEA, usually become significant stumbling blocks, preventing or delaying the analyst from doing a proper and accurate engineering job.

Consider the following issues. Raw experimental data is often “messy” and non-ideal, requiring data manipulation of potentially multiple large data files, various data “clean-up” steps that often

require judgment, additional processing techniques, and ultimately interpretation of how to use this data in conjunction with a given FE analysis. Pre/post processing calculations (applied to experimental data or FE results) may require theoretical development, solving of non-trivial equations, and numerical manipulation of digital data such as trimming, interpolation, extrapolation, re-mapping, smoothing or other DSP (Digital Signal Processing) filtering, differentiation, integration, etc.

While these types of tasks should be, in principle, familiar to the FE analyst, they are NOT tasks that they commonly perform, nor are they tasks that any commercially available FE software alone is capable of addressing. FE users attempting to perform such task may use a variety of software tools, but the most likely tool that will be employed is the ubiquitous spreadsheet. While a spreadsheet is a useful calculation tool and generally easy to use, it has significant shortcomings relative to addressing all the needs for combining experimental data, theoretical calculations, and performing pre/post processing calculations.

To be successful, the FE analyst needs numerical tools with the flexibility to easily address these types of calculations and data manipulations. A new software tool called Kornucopia™ (www.BodieTech.com), which runs in Mathcad® (www.ptc.com), provides unique functionality that fills this important need. This paper presents examples and discussion that demonstrate how the addition of Mathcad® and Kornucopia™ to the overall FEA process enables the analyst to readily integrate theory, experiments, and FEA to solve challenging nonlinear mechanics problems. Gains in overall analysis accuracy, work-flow efficiency, and improved documentation of the analysis process and results are also presented.

2. Creating an average curve from multiple curves

Most analyses, FEA or otherwise, require material properties to perform calculations. These properties are usually derived from experiments, which typically involve testing multiple samples to obtain an average representation and its variation. Consider the case of obtaining the elastic modulus (initial tangent modulus) for a material. Multiple samples are tested, an elastic modulus for each sample is computed from the test data, and then the mean of these moduli is numerically calculated to represent the average modulus for the material. A variation might also be calculated, but generally only the average value is repeatedly published and used in practice.

What about obtaining an average nonlinear stress-strain curve for the material? Experience shows that, in general, the process just described is NOT followed. Instead, a representative curve is often visually selected amongst the samples measured, and then this curve is reported and becomes the de facto average curve for the material. For the elastic modulus calculation described earlier, would it be acceptable to plot (or list) all the elastic moduli measured and then visually select the modulus value that looks to be in the “middle of the pack?” Not likely. So why is this approach acceptable for nonlinear curves of data? Answer – producing an average curve via calculation has traditionally been too difficult in most cases (given the tools being used).

Figure 1a presents three stress-strain curves, from which, an average curve is desired.¹ How should the average stress-strain curve for this data be computed? Attempting to supply this data to the function `AVERAGE(dataRange)` in a spreadsheet will not work. Most FE analysts (and other engineers, experimentalists, and statisticians) will struggle to quickly obtain an answer to this request because calculating a plausible average for such data can be highly problematic.

Using Kornucopia™, a general-purpose solution is now available. The entire image of Figure 1 (all the graphs and additional comments) is a single screen capture from a Kornucopia™ worksheet that demonstrates using four different averaging approaches to create an average representation to the data in Figure 1a. It is seen that the user easily computed average curve representations while having the flexibility of selecting from several viable approaches. This was all accomplished by utilizing a single Kornucopia™ function `averageXY_k(data, method, Adv)`, where `method` was selected as `"x"` or `"arc length"`, and where the `Adv` argument was taken as default `("")` in some cases and was modified to `"norm:yes"` in other cases. A description of how this one function computes an average curve is described below.

Regardless of the `method` selected, the first step is to map all the data curves to a common incrementation relative to some independent variable, and then average the data at common values of that independent variable. This is all done automatically inside the `averageXY_k` function, no user intervention or additional coding is required. Figure 1c shows a result, "X ave", computed this way, where the independent variable was taken as strain (X-axis). This result looks plausible, but can only be computed up to a strain of approximately 0.4-m/m because Sample 1 has no data beyond this value. Further evaluation of the "X ave" result reveals that the last value of Sample 1 (perhaps a failure value, or perhaps the value at which the test was simply stopped) is being averaged with a value that is clearly on a different portion of the curve for samples 2 and 3 (not near their end values). Is that calculation appropriate? Engineering judgment and perhaps other factors are needed to answer that question. If you think not, then the other average result presented in Figure 1c, "Normalized X ave", may be deemed more appropriate. In this calculation, the X-based average is created by first normalizing the strain data for all three samples to a common range before averaging. This allows for the first values of all the data curves to be averaged together, the final values of all the data curves to be averaged together, and then all the data in between to be averaged accordingly. This "Normalized X ave" result is also plausible.

Figure 1d presents a different averaging scheme in which a temporary independent vector is automatically created based on computing the arc length of the XY curve for each sample. Two variations of this approach are presented, one without normalization (denoted in the figure as "Arc length ave") and one with normalization (where the arc length is the normalized entity, not the strain axis). In these arc length techniques, the strain values for each data sample are averaged at common XY arc length locations (similar calculation is done for stresses) and then the resulting average stress and strain curves are plotted together (the temporary arc length is discarded automatically). For these particular data sets, the arc length approaches might be considered more plausible than the "X-based" methods of Figure 1c, again depending on engineering judgment.

¹ It is noted that some of the non-ideal aspects of these particular data sets have been enhanced for illustrative purposes. However, the author has experienced actual cases with data containing similar severity of issues.

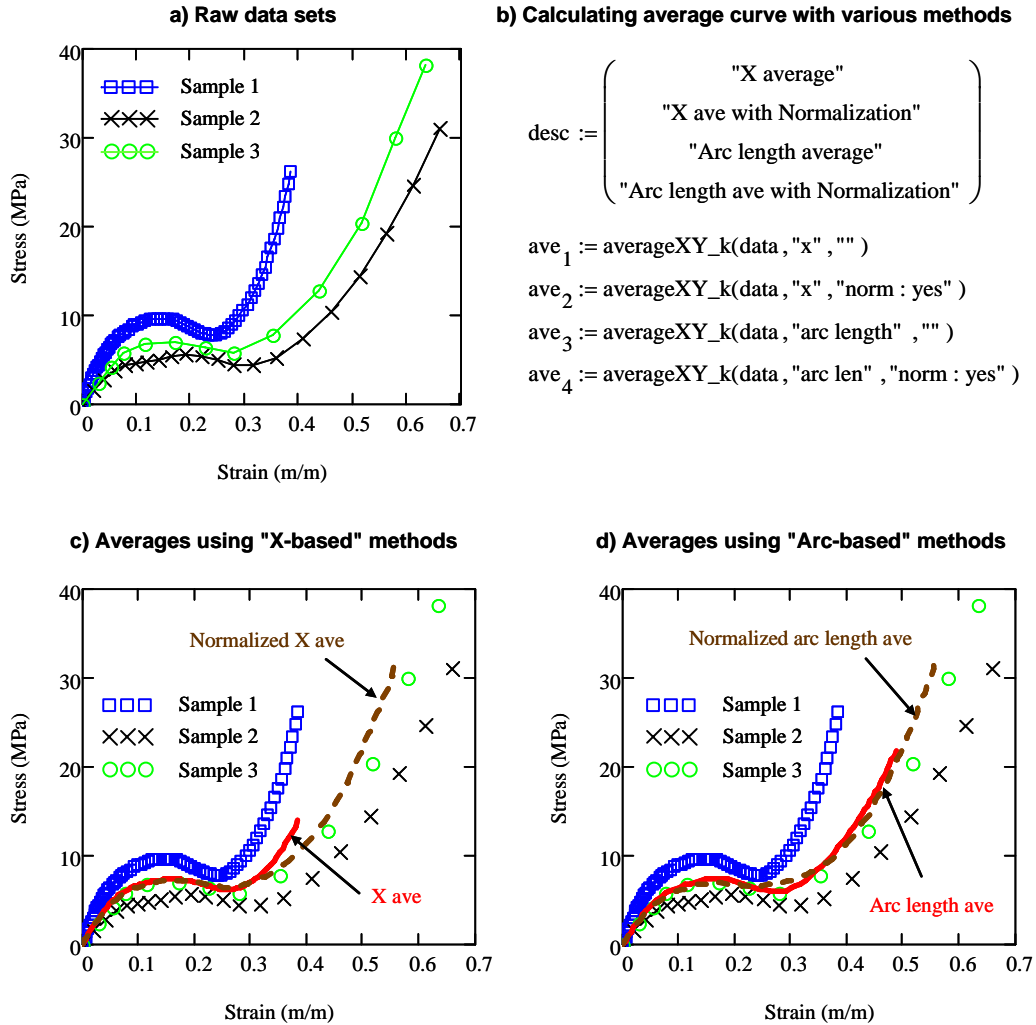


Figure 1: Using Kornucopia™ to easily compute an average curve based on the selection of different engineering assumptions.

The calculations utilized to obtain these average curves are relatively simple in concept, but very tedious and difficult to actually perform with most software tools in use by FE analysts. However, with Kornucopia™, the calculation is easily performed and automatically documented in a clear fashion that can be readily used for a report or presentation.

It is also useful to point out that the use of such averaging methods for curves is applicable to many other aspects of advanced FE analysis and data analysis in general. The parametric-based arc length averaging method works well for a variety of other uses including the creation of an average representation of data depicting the outline of any general 2-D shape (circles, distorted ellipsoids, outline of a fiber tow reinforcement in a set of samples, etc.).

3. Application demonstrations

This section presents brief summaries of four different nonlinear mechanics problems where FEA simulations were significantly improved by the inclusion of theoretical considerations, experimental data, and pre/post processing of FE data. A list of additional publications where the author has used these types of “inclusive” analyses is found in the reference section.

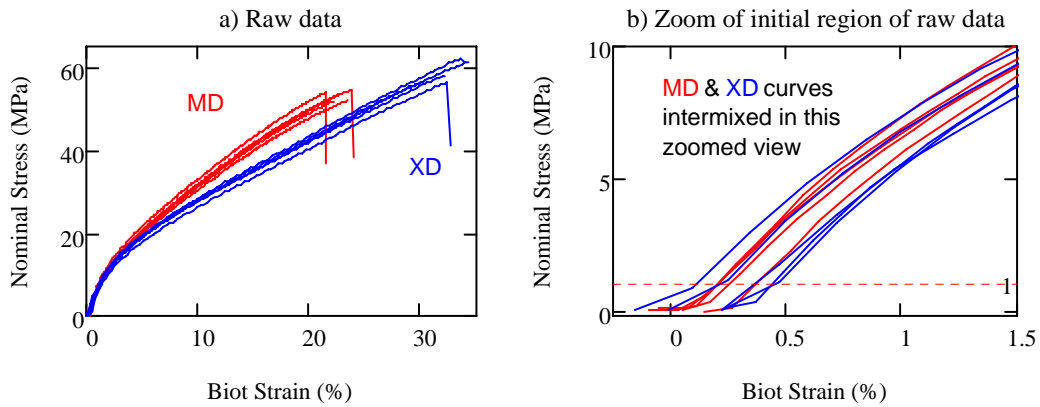
3.1 Analyzing the inflation of a nonwoven envelope

Figures 2 - 4 present the analysis method and results for the inflation to failure of a nonwoven polymeric envelope (see Diehl, Design News, 2004 for additional details). The problem has many nonlinear aspects: material plasticity, failure, large strains, and complex deformations including wrinkling and buckling. In addition, the nonwoven material shows modest variation in material properties at different locations across the sheet. The results presented in the three figures are single screen captures of the Kornucopia™ worksheet that created the material law from raw experimental data and also postprocessed the FEA simulations for comparison to various experiments.

Figure 2 documents the creation of a material law representation derived from measured experimental stress-strain data. Figures 2a and b show some interesting aspects of the raw data that needed to be addressed. First, the material exhibited mild anisotropic behavior when uniaxial tests were performed in two orthogonal directions (MD and XD). Based on pragmatic engineering judgment, the differences were not considered large enough to attempt an orthotropic plasticity analysis. So all the MD and XD data was to be averaged using the arc length method without normalization (it was deemed that failure in MD and XD was too separate to use normalization).

However, before averaging could be performed, the raw data needed to be cleaned-up. Figure 2b shows a zoomed view of the beginning of the measured stress/strain curves – they are not all aligned to begin at strain = 0 and they all show some undesirable curvature at the beginning which is intentionally created by the testing protocol. To avoid accidental over-stressing of the test samples during initial fixturing, the protocol requires the sheet samples to be mounted with a small amount of slack, hence creating the artifacts in Figure 2b. The Kornucopia™ computation syntax² displayed in the middle of Figure 2 “heals” all 20 data curves by first trimming away the initial parts of each curve for stress below a minimum threshold (1.0-MPa) and also trimming away the

² The three lines of syntax used to compute “healed_i” is actually a simple example of using the Mathcad®’s programming language. The programming capability in Mathcad® is quite powerful, including the ability to easily make function subprograms and libraries of reference functions to use as needed.



```

healed_i := [
  trimmed ← trim_k [ data_i, (data_i)^y, 1, "max", "" ]
  tweaked ← tweakXY_k ("start", trimmed, y, 0.0, 5%, "" )
  shiftData_k [ tweaked, "start", (0 "" ), "" ]
]

```

Select averaging method

X
 Arc length

Normalize during averaging?

No
 Yes

```
AVE := averageXY_k (healed, method, adv)
```

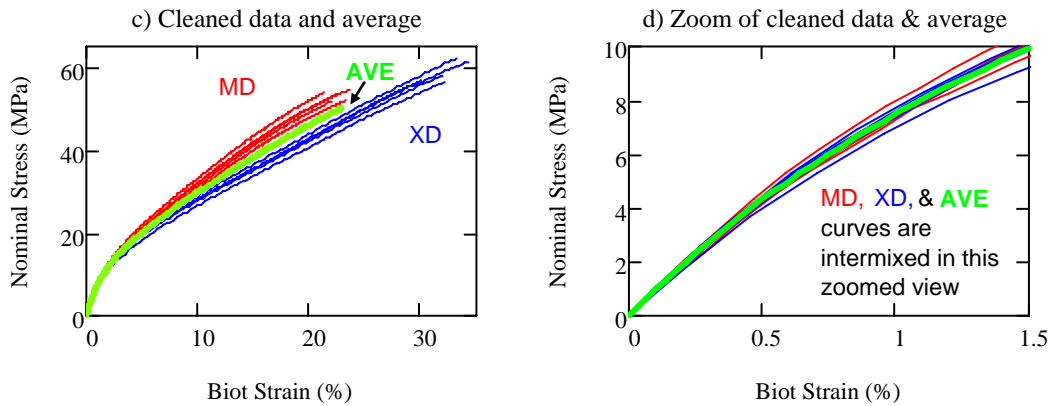
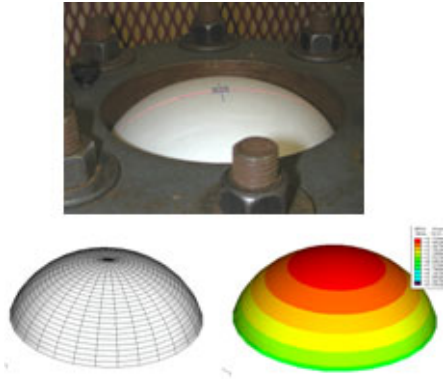
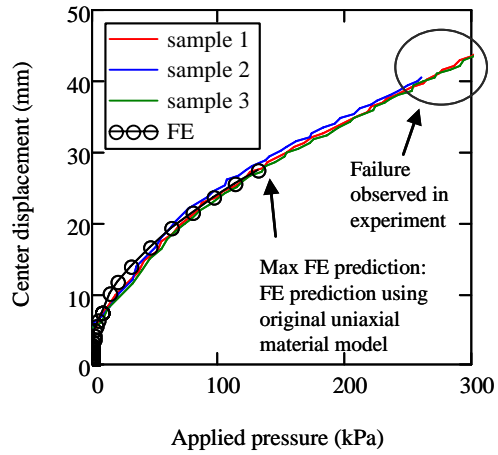


Figure 2: Healing and averaging stress-strain data from 20 uniaxial test samples of a nonwoven fabric.

a) Bubble experiment and FE model

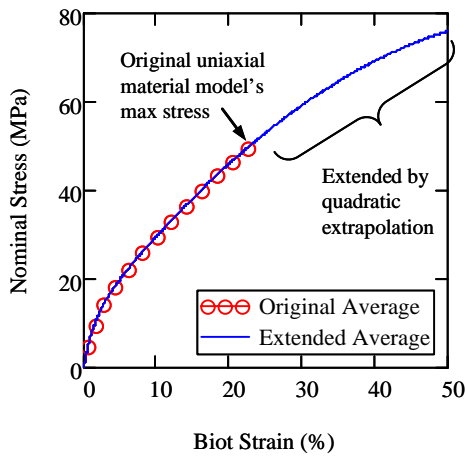


b) Center deflection vs. applied bubble pressure



c) Extending of original uniaxial material law

ref := 30% extendTo := 50% Adv := "excess : 3"
 extended := tweakXY_k("end" , AVE, x, extendTo, ref, Adv)



d) Improved results with extended material model

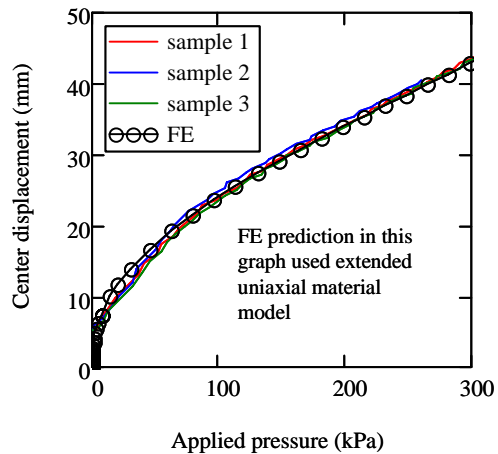


Figure 3: Assessments of biaxial loading and failure of a nonwoven fabric, including material enhancement via extrapolation of uniaxial material law using Kornucopia™.

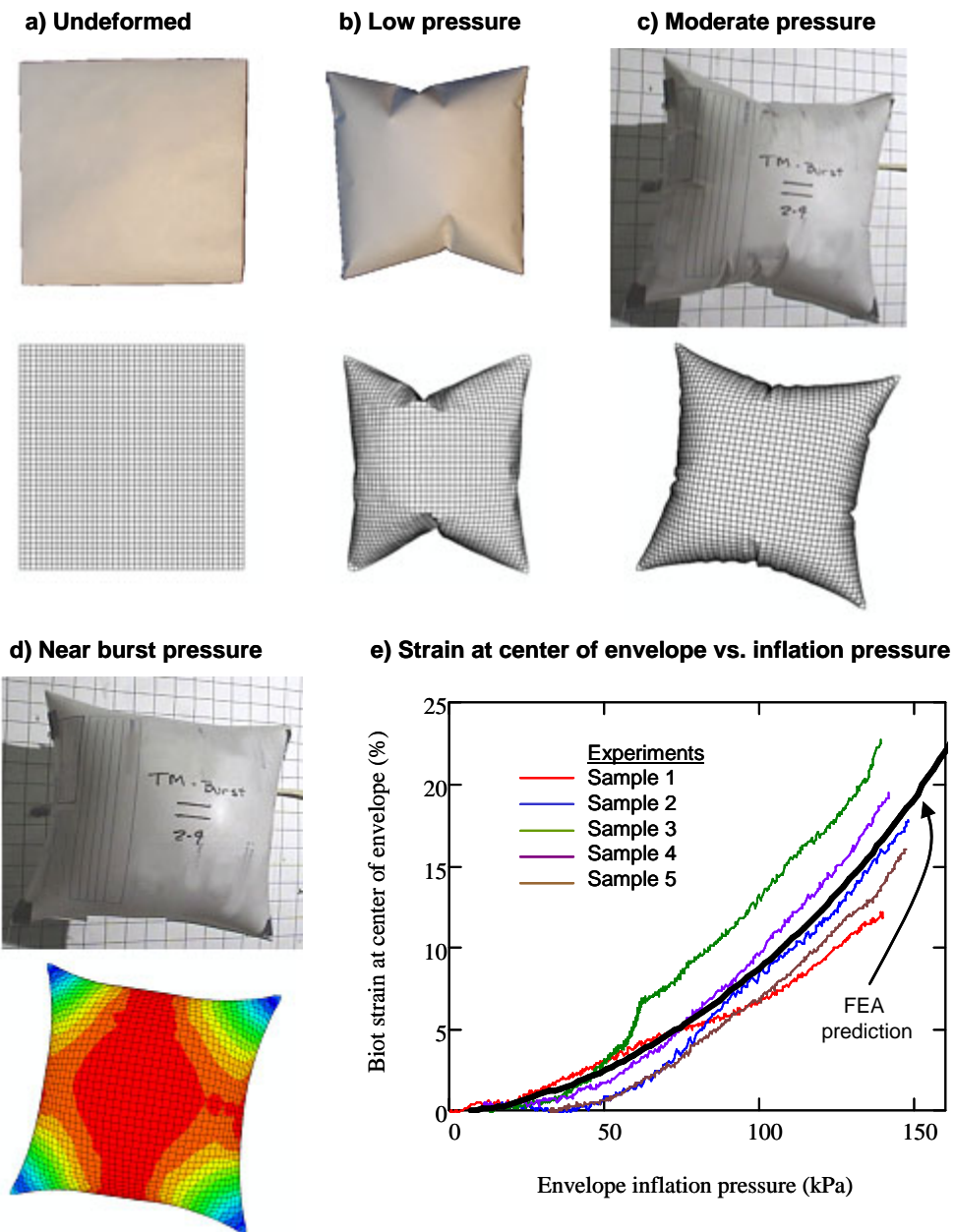


Figure 4: Product-level assessment of FE model accuracy on predicting bag inflation performance.

“failure drop-offs” after the max stress has been reached (Figure 2a). This trimmed data is then automatically extrapolated to a value of stress = 0 by the `tweakXY_k` function (using the first 5% of the trimmed data to create a quadratic extrapolation function for each curve). Finally the strain values are all shifted and reset to start at 0. This set of operations is automatically looped through the index counter `i` for all 20 samples. The healed experimental data, being stored in a nested array (an array of 20 elements where each array element holds the array of stress vs. strain for that sample), is then fed into the `averageXY_k` function which readily computes an average of the data curves using the options selected (results shown in Figure 2c and d). To complete the FEA material law definition, this data is then converted to true stress and logarithmic strain, from which stress vs. plastic strain is computed (This last step, converting to true values and obtaining the plastic table, was also done in this Kornucopia™ worksheet, but is not shown for brevity).

Figure 3 presents evaluation of a bubble test, designed to assess the biaxial characteristics of the material. The bubble test inflates a flat circular sheet, clamped around its perimeter, until it bursts. A measurement of the strain at the center of the bubble and the center vertical displacement are recorded (for brevity, only the displacement results are depicted in the figure). Figure 3b shows that the FE model matches the experimental results well, but that the FE model significantly under-predicts failure. This is caused by surface non-uniformity in the material, which in a long slender uniaxial tension test specimen causes early failure (like the preverbal weakest link of a chain). However, in a biaxial loaded state of a larger sample, load paths around weak spots are more likely, allowing the structure to survive larger loads. As shown in Figure 3c, addressing this issue is easily accommodated in the FE model by modifying the uniaxial-based stress/strain material average curve using the Kornucopia™ `tweakXY_k` function. This time the first parameter is set to extrapolate at the end of the data (previous demonstration of this function extrapolated at the start of the data). The extrapolation shown is longer than the reference section of the data being used to create the extrapolation, so an error (warning) is initially issued. This was overridden by an advanced option stating that extrapolation could be up to 3 times the length of the reference section of the data used. As seen in Figure 3d, the bubble simulation, using the extended data, now correlates well with the measurement over the entire range of interest.

Figure 4 presents the final results of envelop inflation, both simulation and experiments. Figure 4d shows how strain at the center of the envelope increases with inflation pressure. The figure presents five experimental test samples along with the FE model prediction (using the material law developed and described in Figures 2 and 3). Excellent correlation is observed.

Relative to clearly and efficiently documenting analysis work, all this data, including the images of deformed shapes, resides inside the Kornucopia™ document within Mathcad®. The Abaqus/Explicit FE deformed shape and deformed strain contour images were simply pasted into Kornucopia™ worksheet from Abaqus/Viewer.

3.2 T-bracket failure analysis

Figure 5 presents Abaqus/Standard (implicit) and Abaqus/Explicit (explicit) simulation results for the failure of a metal T-bracket. The goal of the analysis is “methods development” for the use of cohesive elements for this type of event (computed with both implicit and explicit FEA techniques).

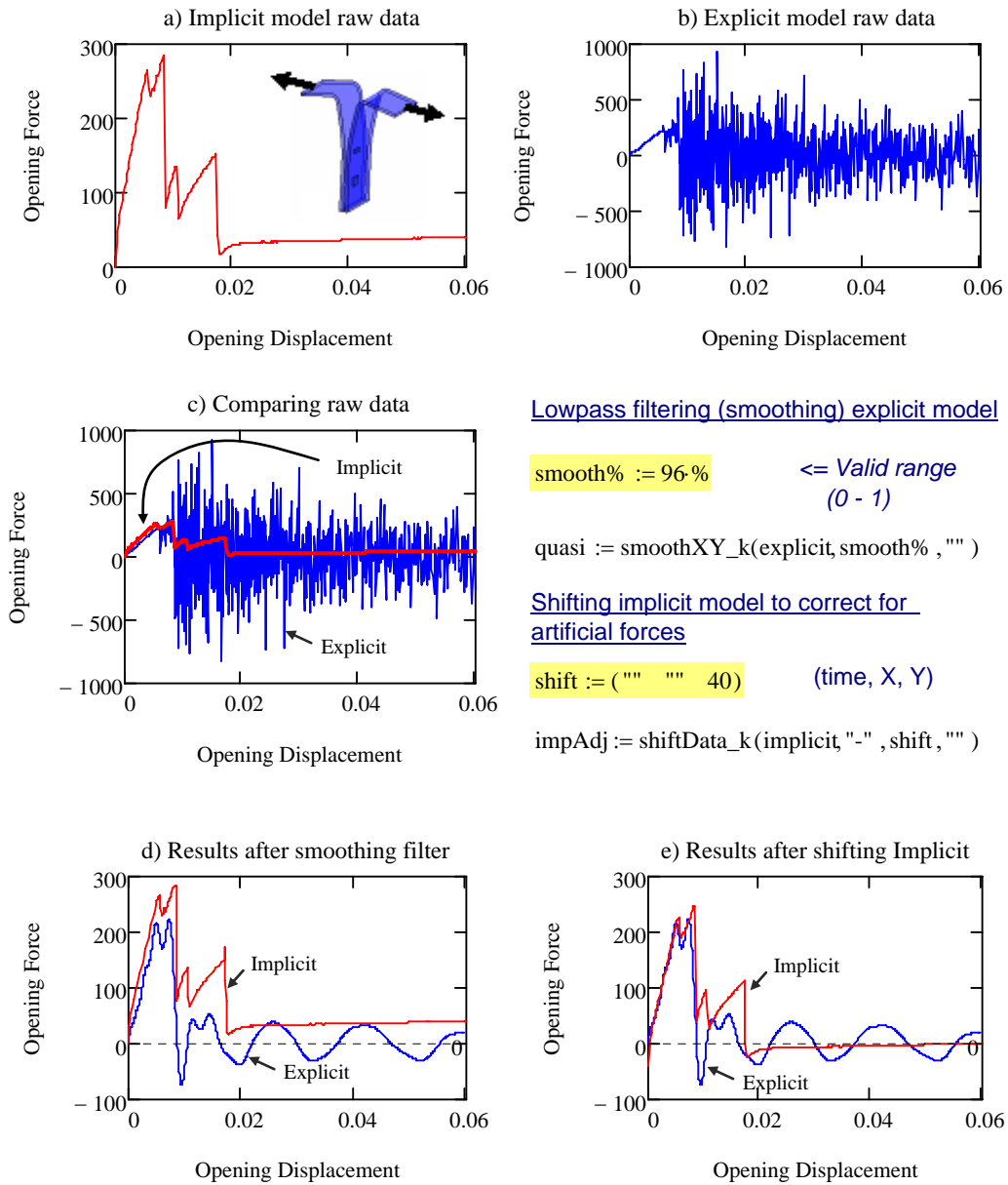


Figure 5: Using Kornucopia™ to significantly improve interpretation of both an implicit static and explicit dynamics quasi-static simulation of a T-bracket failure.

Displacement and reaction forces at the loading points of the structure are exported from the Abaqus odb files from Abaqus/Viewer via the Report XY feature. For the explicit model, the Abaqus run-time antialiasing filter was used to obtain a reasonable size output (~ 700 time points) representation of the event while avoiding aliasing. (See Abaqus, 2007, Diehl, 2000, and Diehl Bodie Technology, 2007 for information about avoiding aliasing and properly working with noisy data). This data is then read into Mathcad®, plotted, and further processed with the Kornucopia™ worksheet depicted in Figure 5.

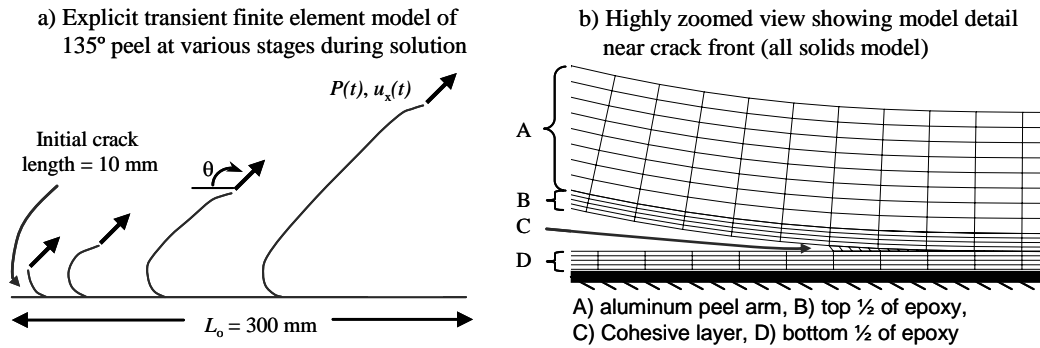
The first issue to be addressed is smoothing the explicit dynamics results. Since the model is intended to be quasi-static, the user is unlikely to have any knowledge about natural frequencies of the structure. Hence, the analyst will find using classical DSP lowpass filtering methods uncomfortable to use because they will require information that is not readily available. Kornucopia™ provides an easy-to-use DSP-based smoothing filter, `smoothXY_k`, that can be controlled with single smoothing parameter that ranges from 0 (no smoothing) to less than 1 (remove all frequency content). Comparing filtered explicit results with the non-noisy implicit results, the user quickly gains experience on filtering the explicit model results. It is important to note that attempting to smooth this type of data with a non-DSP “running average” method, like the Abaqus/Viewer operator `smooth()` or similar functions in Mathcad® (or elsewhere) may work too, but often these methods will not yield as good a result as a DSP-based filter (such as `smoothXY_k`) due to `smoothXY_k`'s flat passband characteristics (Diehl, BodieTech, 2007).

Once the noise is removed from the explicit results, a direct comparison to the implicit result can be obtained. This comparison, Figure 5d, shows similar results, but clearly demonstrates that the implicit model predicts higher peak force. Further inspection indicates that the implicit model predicted a nearly constant, non-zero load well after the two parts had fully failed and completely separated – a physically impossible result! This issue with the implicit result was not noticed by the FE analyst initially working this model (they did not have the explicit model for comparison initially). The problem with the implicit model is excessively high artificial forces being created by numerical penalty parameters set by the user to enable this very challenging failure problem to compute in an implicit solution. Plotting of model energies from the implicit model would have alerted the analyst to the problem, but this was apparently not done (which is commonly the case for many analysts). A simple shifting of the implicit result and comparison to the explicit result shows that this artificial force was corrupting the analysis from the very beginning (Figure 5e).

This example is an interesting case of the implicit model helping to guide understanding of how to smooth explicit results, while at the same time the explicit results pointing out a significant distortion in the implicit model that had previously gone unnoticed.

3.3 Peeling of an epoxy-bonded Aluminum strip

Figure 6 presents details from one of the key findings related to using a novel penalty framework for enhancing the use of the Cohesive-Zone Method (CZM) in finite element models simulating crack initiation and crack propagation of surface-bonded structures (Diehl, IJAA 2008). In this work, Diehl studied a variety of modeling parameters related to CZM use, ultimately running and post-processing 246 different Abaqus/Explicit models for the study.



Adjusting the Peel force by removing the force associated with the kinetic energy

$$P_{KE} = \frac{d}{du_x} ALLKE = \frac{d}{dt} ALLKE \cdot \frac{1}{\left(\frac{d}{dt} u_x\right)} = \frac{ALLKE'}{u_x'}$$

Number of FE models to analyze

numOfCases = 246

i := 1..numOfCases

Computing numerical derivatives

$$ALLKE'_i := \text{derivative1}_k(t_i, ALLKE_i, "")$$

$$u_x'_i := \text{derivative1}_k(t_i, u_{x_i}, "")$$

$$P_{KE}_i := \frac{ALLKE'_i}{u_x'_i}$$

Calculation of quasi-static peel force without kinetic energy contributions

$$\Rightarrow P_{qs}_i := P_{total}_i - P_{KE}_i$$

Specify a case to plot

num := 63

cases_num = "case_22g_135deg_1p32G"

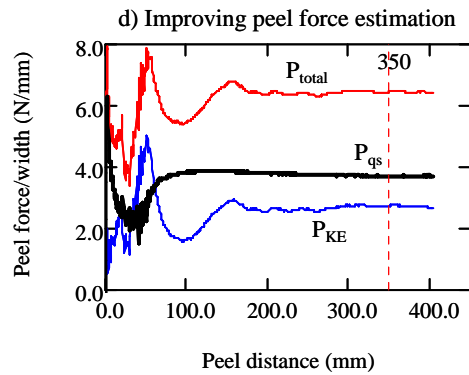
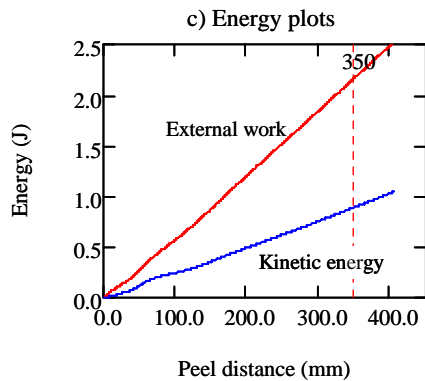


Figure 6: Significantly enhancing analysis accuracy and solution speed by adjusting the calculated peel load using derivatives of computed energy terms.

The results displayed in Figure 6 show how peel force estimation from an explicit dynamics model, intended to represent quasi-static peeling, can be significantly improved by compensating for forces caused by the continual increase of kinetic energy due to an ever-increasing peel arm. Based on energy methods principles, the derivation of a “kinetic energy force,” P_{KE} , is defined and then computed numerically for all 246 explicit dynamics models. Figures 6c and d show specific result details for one particular case, namely a 135° peeling with a critical energy release rate, G_c , of 1.32 kJ/m^2 . In this particular case, the correction in peel force when the model is in steady-state ($\sim 350 \text{ mm}$ peel distance) is significant, $P_{qs}/P_{total} = 0.58$. Allowing the explicit models to run with this amount of kinetic energy, without the correction, would generally create an unacceptable distortion in the reported peel force. But with the correction, computational speed-ups of 2 times to 4 times are achieved with significant improvement in accuracy. For a study using 246 runs, where individual simulations range between 20 minutes and 2 hours, such savings are significant.

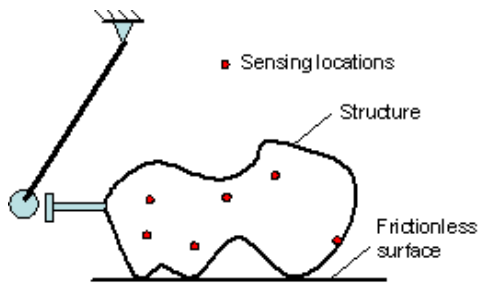
The results presented in Figure 6 were created by a single Kornucopia™ worksheet that imported all 246 abaqus.rpt files (not shown in figure), all of which were automatically generated from Abaqus result output databases by a custom Python script. While it is true that such energy derivative calculations could have been done in Abaqus/Viewer, driven by a Python script to loop through all the models, such a method would not create as clear a set of documentation for the methodology and its implementation. Within the Mathcad® software that Kornucopia™ runs in, this type of “numerically operable” derivation and documentation is natural and easy to exploit.

Additionally, developing and implementing the methodology within Kornucopia™ has the benefit of keeping the fundamental principles and equations in view of the engineer. In this particular case, that work-flow led to the realization that such derivatives of FEA-computed energies could be used for other design guidance purposes. For peeling systems that have multiple layers, such as multi-layered polymeric films, computing derivatives of internal strain energy quantities (grouped by the different material layers) relative to peel distance provides a direct measure of how much each of the layers contributes to the total quasi-static peel force of the system. This calculation demonstrates that peel force is controlled by much more than how strong the interface is (characterized by G_c). This realization ultimately turned the CZM FE model into a powerful and insightful applied research and design tool to drive new experimental testing procedures and new product development plans.

3.4 Shock analysis of a structure with many output sensors

Figures 7 and 8 present efficient postprocessing of an impact analysis involving the evaluation of multiple output sensors from several transient explicit models. The models are run in Abaqus/Explicit using the run-time antialiasing filter feature for output requests, allowing for optimum-sized abaqus.rpt files to be generated without alias distortions. Each model output file contained transient data columns of time and 30 sensors (velocity output at various locations in various directions). The requirement of the analysis was to lowpass filter the sensor data using customer-specified filter parameters and then to take these filtered signals and compute average accelerations for the various sensors, using a customer-specified method.

For presentation brevity, only three models are evaluated and only three sensors within each model are selected from the 30 sensors computed. The actual customer DOE (Design of Experiments) study required evaluation of all 30 sensors from over 50 FE simulations.



```
names = (
  "model1.rpt"
  "model2.rpt"
  "model3.rpt"
)
```

```
i := 1..rows(names)
```

```
data = (
  {2511,31}
  {2502,31}
  {2498,31}
)
```

Select sensors of interest

```
sensors := reorder_k[sensors, "keep", "cols", (1 5 13 6), "" ]
```

```
sensors = ("Time" "TopRear, X" "Front, X" "TopRear, Y" )
```

```
data_i := reorder_k[data_i, "keep", "cols", (1 5 13 6), "" ]
```

Perform DSP

```
fs_i := samplingFreq_k[(data_i)^(1)·sec, "" ]
```

```
fs = (
  10.043
  10.006
  9.956
)·kHz
```

Define cutoff frequency

fc := 50·Hz

```
C_i := cIIR_k("lp", "butter", fc/fs_i, "" )
```

```
dataf_i := filterXY_k(data_i, C_i, "start assumpt : zero" )
```

select a sensor to view

s := 4

sensors^(s) = ("TopRear, Y")

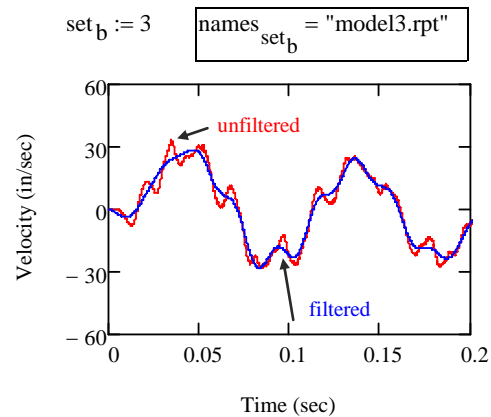
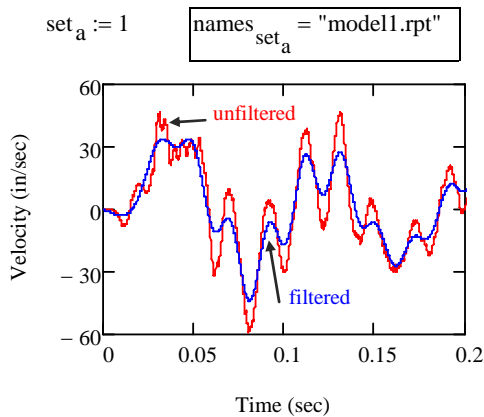


Figure 7: Efficiently computing and clearly documenting an impact analysis with many output sensors, part 1 – sensor selection and Butterworth lowpass filter application to computed FE results.

Specify a rough maximum time trigger `tmax := 0.1` counter to loop through sensors
 Specify minimum velocity trigger `trigger := 5` `j := 2..cols(dataf_1)`

Trim out the data section in each curve to be analyzed

$$\text{analyze}_{i,j} := \left[\begin{array}{l} \text{rough} \leftarrow \text{trim_k} \left[\text{dataf}_i, (\text{dataf}_i)^{\langle 1 \rangle}, "", \text{tmax}, "" \right] \\ \text{trim_k}(\text{rough}, \text{rough}^{\langle j \rangle}, \text{trigger}, "max", j) \end{array} \right]$$

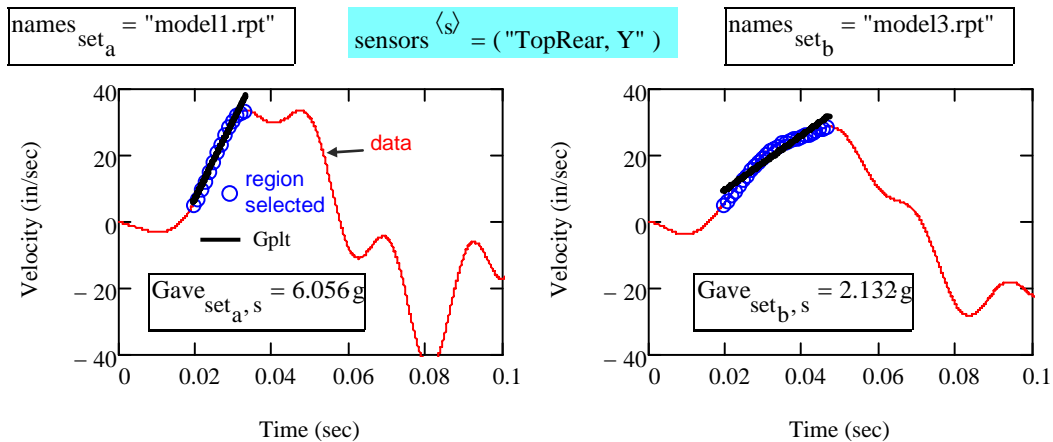
Assigning time and sensor velocity variables $t_{i,j} := (\text{analyze}_{i,j})^{\langle 1 \rangle}$ $V_{i,j} := (\text{analyze}_{i,j})^{\langle j \rangle}$

Computing mean acceleration values

$$\text{Gave}_{i,j} := \text{slope} \left(t_{i,j} \cdot \frac{\text{in}}{\text{sec}}, V_{i,j} \cdot \frac{\text{in}}{\text{sec}} \right)$$

$$\text{Gplot}(t, V) := \left[\begin{array}{l} M \leftarrow \text{slope}(t, V) \\ B \leftarrow \text{intercept}(t, V) \\ M \cdot t + B \end{array} \right]$$

$$\text{Gplt}_{i,j} := \text{Gplot}(t_{i,j}, V_{i,j})$$



Acceleration Summary (units = g)			
	TopRear, X	Front, X	TopRear, Y
model1.rpt	8.67	24.69	6.06
model2.rpt	1.00	20.81	2.15
model3.rpt	0.99	20.78	2.13

Figure 8: Efficiently computing and clearly documenting an impact analysis with many output sensors, part 2 – average local acceleration calculation and summary.

The portion of the Kornucopia™ worksheet displayed in Figure 7 clearly and naturally documents the post-processing analysis, including filter parameter specification, application, and influence on the simulation data. Figure 8 documents how the customer-specified method to compute average accelerations from filtered velocity data was accomplished in an efficient manner. The data was first trimmed to keep data before 0.1 seconds, from which a low-level velocity trigger combined with a max value velocity trigger was used to automatically select the initial “velocity rise.” An average acceleration was then calculated based on the best-fit slope for this region. Both the details of the computation and the resulting values are clearly presented. Attempting this type of analysis with a spreadsheet, especially for 50+ models and 30 sensors, would be extremely difficult and even more difficult to present calculation details.

A summary table of accelerations for the sensors of interest across all models computed is displayed at the bottom of Figure 8. A quick inspection of this summary table indicates that the results for model1 are significantly different than the other two models. If only the summary table was reported, without displaying the calculations, this large difference could create significant concern amongst the engineers and managers working on this design study. Fortunately the documentation provided in Figure 7 and 8 easily explains the cause. Figure 8 show that the region (blue circles) used to compute the average acceleration (straight black line) is noticeably different due to the fact that the max velocity in model1 occurred at the first “hump” (~0.03 sec) while the max for model3 occurred at the second “hump” (~0.05 sec). Further evaluation of the influence of lowpass filtering from Figure 7 shows that before filtering, the maximum velocity for both models occurred at a similar time of ~0.03 sec, but that after filtering the location of the maximum velocity was noticeably altered for model3. This occurred because the frequency content defining the hump in model3 is slightly different as compared to model1 and just happened to be attenuated enough by the filter so that the second hump became the maximum. This type of subtlety is very common in this kind of analysis and presenting summary tables of results only, without calculation details to provide perspective, can be extremely misleading. If calculation documentation naturally occurs by the act of making the calculation, then these types of confusion are noticeably reduced.

4. Conclusions

Solutions to challenging structural mechanics problems often rely on approaches that properly integrate theoretical analysis, experiments, and finite element simulations, along with judgment and practical approximations. These additional requirements, beyond just FEA, often become significant stumbling blocks, preventing or delaying the analyst from doing a proper and accurate engineering job. The examples presented in this paper have shown that by using a combination of Abaqus Unified FEA (www.simulia.com), Mathcad® (www.ptc.com), and Kornucopia™ (www.BodieTech.com), many of these stumbling blocks are avoided and significant improvements in analysis accuracy and results communications are achieved. Additionally, techniques have been demonstrated for the creation of an average nonlinear material description from measured data containing large variations.

5. References

1. Abaqus, *Abaqus Version 6.7 Documentation*, SIMULIA, Dassault Systemes, 2007.
2. Carbajal, L., Diehl, T., Lamontia, M. A., Deakyne, C., "Modeling Techniques for Simulating Lightweight Woven Fabric Barriers Protecting Body Simulants Subjected to a Non-Penetrating Impact Threat," ABAQUS User's Conference, Boston, MA, May, 2004.
3. Diehl, T., "Simulating the Transport of Very Flexible Sheets: Beam-Type Solutions," Abaqus Users Conference, Aachen, Germany, 1993.
4. Diehl, T., "Methods of Improving ABAQUS/Standard Predictions for Problems Involving Sliding Contact," ABAQUS User's Conference Proceedings, Paris, France, June, 1995.
5. Diehl, T., Carroll, D., Nagaraj, B., "Ball Drop Simulation on Two-Way Radio Lens Using ABAQUS/Explicit," ABAQUS User's Conference, Chester, England, May, 1999.
6. Diehl, T., Carroll, D., Nagaraj, B., "Applications of DSP to Explicit Dynamic FEA Simulations of Elastically-Dominated Impact Problems", *Journal of Shock and Vibration*, Vol. 7, Number 3, 2000.
7. Diehl, T., "Modeling the "Shrink-Wrap Effect" in Polymers and Elastomers, Including the Influence of Very Large Elastic and Inelastic Strains," ABAQUS User's Conference Proceedings, Newport, RI, May, 2002.
8. Diehl, T., Dixon, R. D., Lamontia, M. A., Hanks, J. A., "The Development and Use of a Robust Modeling Approach for Predicting Structural Performance of Woven Fabrics Using ABAQUS," ABAQUS User's Conference Proceedings, Munich, Germany, May, 2003.
9. Diehl, T., "A Few Tips for Solving the Unsolvable," *Machine Design*, V. 75, June 5, 2003.
10. Diehl, T., Sloan, J., Dickerson, D., Lamontia, M. A., Deakyne, C., "An Up-Close Model Characterizing a Highly Nonlinear, Fluffy-cored, Skinned, Composite Material Used in a Local Buckling Analysis," ABAQUS User's Conference, Boston, MA, May, 2004.
11. Diehl, T., "How To Deal with a Nasty, Nonlinear Problem," *Design News*, Oct. 25, 2004.
12. Diehl, T., *Improving Abaqus/Explicit Analysis via Filtering and DSP*, short course, Bodie Technology, Inc., www.BodieTech.com, 2007.
13. Diehl, T., "On the Measurement and Modeling of Polymeric Materials that Exhibit Unstable Localized Necking," Abaqus Users Conference, Paris, France, 2007, pp. 188 – 201.
14. Diehl, T., "On Using a Penalty-Based Cohesive-Zone Finite Element Approach, Part I: Elastic Solution Benchmark," *International Journal of Adhesion and Adhesives*, Vol 28, No. 4-5, June 2008, pp. 237-255.
15. Diehl, T., "On Using a Penalty-Based Cohesive-Zone Finite Element Approach, Part II: Inelastic Peeling of an Epoxy-Bonded Aluminum Strip," *International Journal of Adhesion and Adhesives*, Vol 28, No. 4-5, June 2008, pp. 256-265.
16. Hanks, J. A., Scott, B. R., Lamontia, M. A., Diehl, T., Dixon, R., Hamlet, S., "The Testing, Modeling, and Analysis of Residential Wall Panels Designed to Resist Tornado Borne Debris," *Proceedings of the 20th International Symposium on Ballistics*, Technomic Publishing, Orlando, Sept. 2002.