MACHINE DESIGN

Today is: June 30, 2003

http://www.machinedesign.com

A few tips for Solving the Unsolvable

Engineering math software and FEA make a potent team for solving tough problems and a way to clearly document techniques for the benefit of others.

Ted Diehl
Engineering Consultant
Du Pont Co.
Wilmington, Del.
(**www.dupont.com**)

Half the engineering battle in solving problems is coming up with the right tools for the job at hand. For instance, what would you need to model a paper jam in a copier or the shock in a dropped cell phone?

One approach uses two complementary analysis tools. One tool is engineering calculation software for applying mathematics, such as Mathcad from **Mathsoft Inc.**, Cambridge, Mass. (**www.mathsoft.com**). The program lets engineers work naturally using applied math and engineering equations, mixed units, symbolic equations, numerical methods, graphing, text, and pictures, in an easily readable format. The second program is finite-element-analysis software, such as Abaqus from **Abaqus Inc.**, Pawtucket, R.I. (**www.abaqus.com**), for general-purpose and nonlinear simulations.

To show how these work together, let's examine a general approach to two different analyses. The first examines paper transport in a copier while the other involves the problem of drop shock in a cellular phone. The idea is not to present detailed solutions but rather a method for attacking problems that can be applied to other engineering problems.

While working on the iDEN line of cell phones at Motorola, the author dropped a steel ball 0.5 m onto the phone cover, collected data, and then simulated the event in FEA software. Combining applied mathematics software such as Mathcad, and nonlinear FEA software such as Abaqus, lets users incorporate nonlinear closed-form analysis, real-world FEA simulations, and complex experimentally measured material behavior into a cohesive engineering analysis.

A paper model

Anyone using copiers has experienced paper jams and distorted images. Understanding these problems comes down to analyzing the soft rubber rollers that push and pull paper. The problem includes three categories of nonlinear behavior for structural mechanics: materials, kinematics (large deformations), and contact (changing boundary conditions).

By observation, foam-rubber rollers are repeatedly exposed to relatively large deformations and strains as they turn. Deformations are greater than can be described by Hooke's Law. These large, nonlinear, elastic deformations in the rollers call for a material law based on the theory of hyperelasticity. So the first hurdle to clear is in expanding your understanding of materials from simple Hooke's law,

$$\sigma_1 = E\,\varepsilon_1,$$

to an Ogden-Hill hyperelastic strain-energy density function defined by:

$$W = \sum_{n=1}^{N} \frac{2\mu_n}{\alpha_n^2}\left[\left(\lambda_1^{\alpha_n} + \lambda_2^{\alpha_n} + \lambda_3^{\alpha_n} - 3\right) + \frac{1}{\beta_n}\left(J^{(-\alpha_n \beta_n)} - 1\right)\right]$$

where W = strain energy density (psi); μ_n, α_n, and β_n are material constants for an n-term representation; λ and J are related to strain (ε) through additional equations, and W is related to stress (σ) through various derivatives.

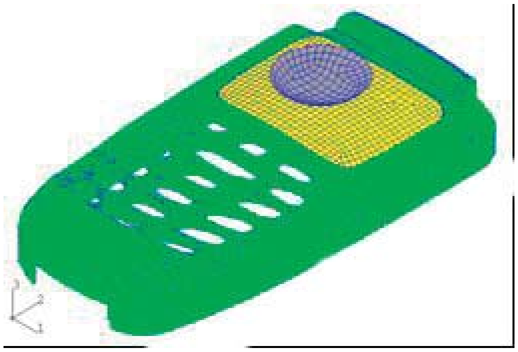
This is the recommended material law and equation for nonlinear FEA codes when modeling foam-rubber materials. Despite documentation in FEA manuals, it is difficult to use this complicated equation, especially when ones previous experience is limited to Hooke's law. But don't be put off by the expression's complexity. It can be tamed.

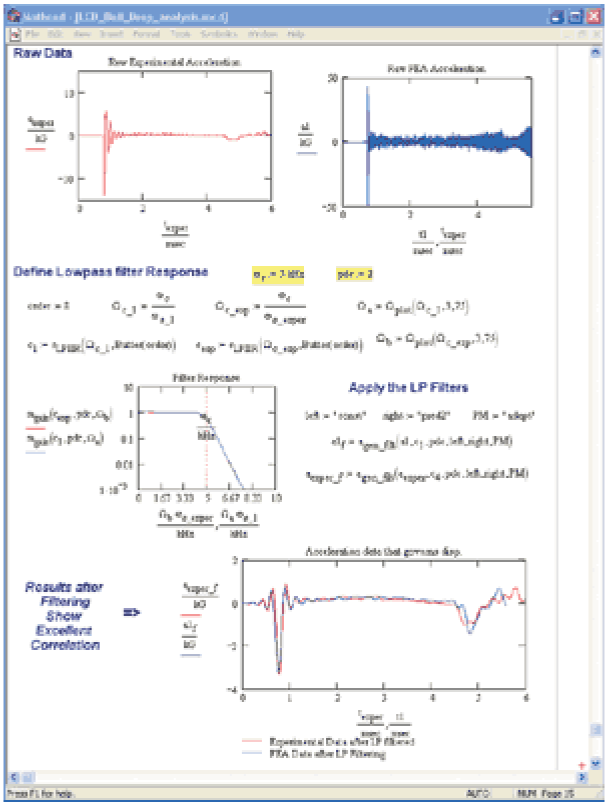
Several experiments with the material provide enough data for the FEA program to provide a third-order curve fit. The FEA program then generates the constants for the equation. Because the engineering math program easily handles symbology, numerical algorithms, graphs, a range of units, and text, users can study such complex equations in more detail and see how different equation parameters affect material response (behavior).

After deriving a workable set of material laws (for paper and rubber) we develop a viable approach to model the entire problem using FEA. While nonlinear closed-form analysis is not likely to solve the entire system-level problem of paper moving in a copier, it does help gain ability and confidence. Studying closed-form nonlinear analytical solutions for a nonlinear beam theory called the Elastica helps understand how to model the nonlinear kinematics of paper transported by the foam rubber wheels. (Elastica is the large-deformation, small-strain beam theory described by Timoshenko and is covered in several texts. Bending a sheet of paper, for instance, produces a large deformation with small strains.) In this case, engineers could type the complex beam equations into Mathcad (just as they appear in the textbook) and ask for numerical solutions. These results can be compared to output from nonlinear Abaqus beam and shell models, and to physical experiments.

Shock and fall

The other problem, understanding what happens to a cell phone when it falls to a hard floor, is somewhat more difficult. Typical cell-phone housings are often crowded with chips, circuit boards, connectors, glass, foam-rubber snubbers, and





Raw data from an experiment and a simulation show no correlation. But they appear similar in a graph (at right) after removing high-frequency noise. A critical step in processing the FEA data was to reinterpret it with a constant time increment scale and remove the high-frequency noise.

metal shields. The questions to answer include: What happens during impact? How much does the circuit board deflect? And how badly are critical components shocked? Answers come from a combination of experiments, closed-form analysis, and nonlinear FEA simulation.

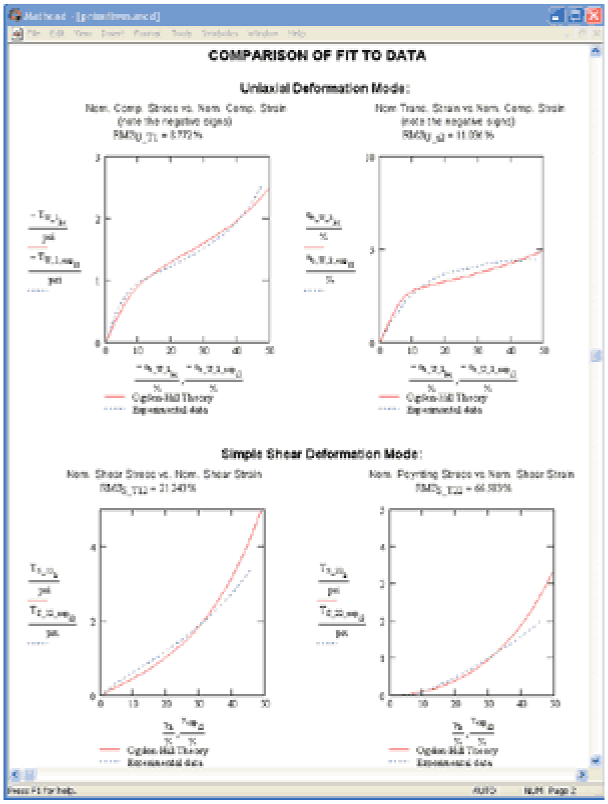
Geometry, materials, contact, and loading are complex, so it is important to ensure models and information processed from them are accurate. A simplified cell-phone impact test helps develop basic-modeling approaches along with data-collection and processing methods.

To get a handle on the problem, a steel ball was dropped onto an instrumented phone cover, and the acceleration signal was compared to that from a simulated test. FEA models for this type of simulation use explicit dynamics technology, the same analysis method used to simulate crash tests. The first problem here: data not properly collected and manipulated. It is easily corrupted and distorted by aliasing. A simple example of aliasing is the apparent backward motion of wagon wheels in old Western movies. Frames in the film were taken at rates too slow to capture the wheel spoke's accurate locations. So at certain playback speeds, wheels appear to turn backwards, which obviously is not happening. A similar defect often occurs when collecting transient data in simulated drop tests if the data is collected from the FEA model at too slow a sampling rate. Portions of high-frequency noise in the FEA solutions may be recorded as low-frequency data, and then misinterpreted. Data that is improperly collected cannot, in general, be fixed. If however, the data is collected properly and aliasing is avoided, then filtering can remove noise.

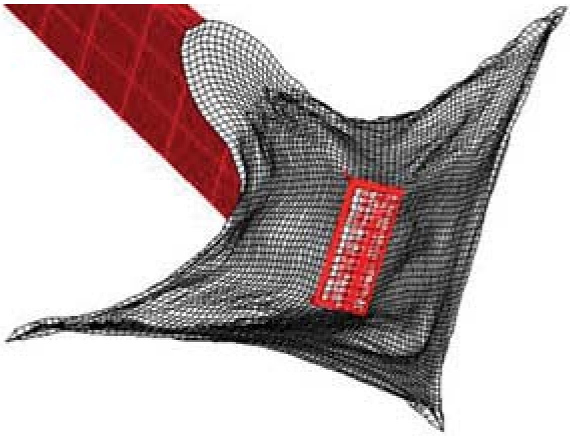
In the physical experiment, raw data looks completely different from that collected during an FEA simulation of the same event. FEA results are extremely noisy. One way to remove the noise, assuming all data was collected without aliasing, is to use smoothing -- a low-pass filter -- that passes low frequencies and rejects high frequencies that make up noise. The problem with explicit dynamic FEA codes, such as Abaqus/Explicit, is that they use an adaptive time-increment algorithm which gives data a varying time increment throughout the solution. This violates the requirement of digital signal-processing (DSP) programs that include filtering and other Fourier analysis. Data must have a constant time increment. The math program provided a solution. Its spline function allowed reinterpolating the data to a constant-time increment. Then it was ready for low-pass DSP filtering. When handled properly, the two signals (from experiment and FEA) correlate quite well all across the scale.

If the same type of processing is tried in a spreadsheet, several problems arise. First, spreadsheets can handle relatively small data sets and would not be able to open typical FEA data files that contain 100,000 lines of data and more. Second, the spreadsheet does not have sufficient capability to perform the manipulations required. Lastly, most of the calculations done in a spreadsheet would be indecipherable to those not intimately involved.

Solving this shock problem called for learning about DSP, data sampling, aliasing, dealing with noisy data, details of Fourier analysis, and more. The details are beyond the scope of this text. But in a nutshell, the effort required several textbooks, a couple short DSP classes, and discussions with other engineers who commonly use DSP technology (acoustic engineers in this case). In particular, Mathcad's WYSIWYG format and general capability helped provide a solid



Mathcad was used to document and study fundamental issues with hyperelastic material models. The document compares closed-form analytical results to experimental results. Experimental results (below) were used to develop an understanding of how to use this hyperelastic approach inside Abaqus FEA.



Products made of Kevlar, such as military and police vests, can already stop high-speed projectiles. Mathcad and FEA programs can provide a way to model and test improved designs of the material that will have a good correlation to real-world performance.

understanding of DSP and a way around FEA's idiosyncrasies.

When a project finishes, the details are stored in Mathcad (or engineering calculation and documentation software) for use by others. It's readily understandable, even to those unfamiliar with the software, and can be modified for new tasks by those with a license to it. Most importantly, the engineering math program lets engineers easily create a toolbox of functions that develops over the length of a career to allow processing of closed-form equations, FEA, and experimental data in ways that neither approach could do alone.

Where math software and FEA work well

Although the article describes two solutions that capitalize on the strengths of mathematics software and FEA, there are other complex nonlinear problems that also benefit from similar teamwork. For instance, when dealing with:

- Advanced material laws. Vendors know the color and price per pound of a material, but they often don't supply appropriate data for advanced materials. FEA outputs are only as good as the material model, so new materials often lead to questions such as, what are the proper constitutive relationships, especially if nonlinear material behavior is to be analyzed? What physical measurements must be made to obtain sufficient data to back-fit material parameters? The math program lets engineers transition from the simplicity of Hooke's law to the more complex hyperelasticity equations.
- When using linear or nonlinear FEA, how do you check the accuracy of results? Closed-form or numerical simulations can assist. How much of a problem can be solved with closed-form analytical analysis? How much requires a numerical approach with nonlinear FEA? What FEA modeling tricks might one need?
- Handling highly transient, noisy data prompts questions such as, how does one properly collect and process such data? For many engineers, the more appropriate question is: "What is DSP?"