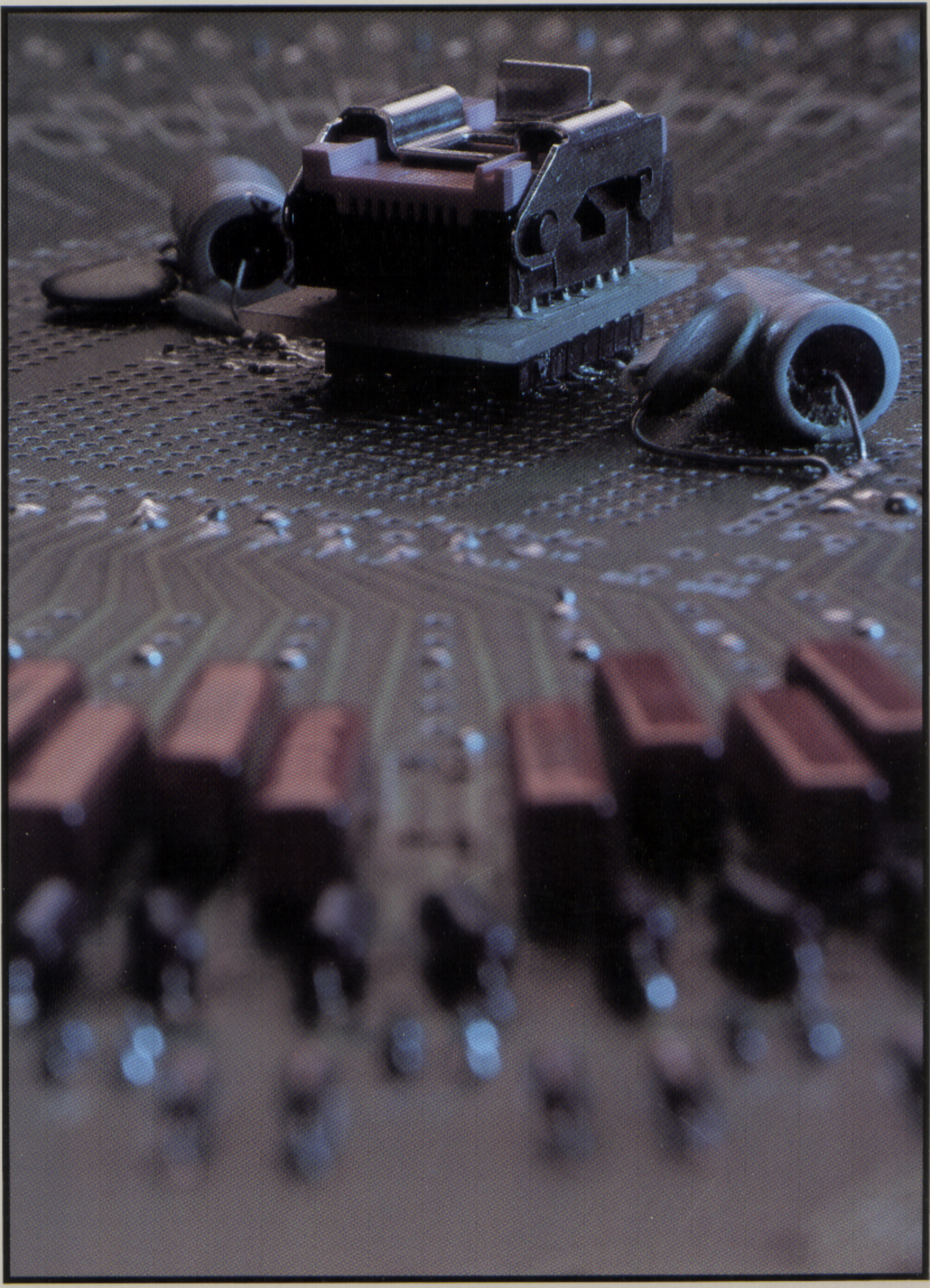
A CRAY RESEARCH, INC. PUBLICATION

CRAY CHANNELS

Winter 1987

FEATURE ARTICLES:

DEPARTMENTS:

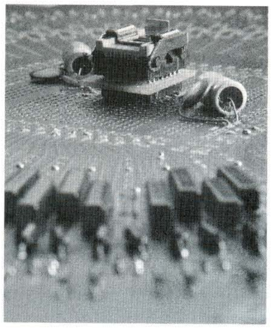


IN THIS ISSUE

Cray Research products — high-speed general-purpose scientific computer systems — are put to the test every day in industrial laboratories around the world. But perhaps in no industry are the words "general purpose" put to the test more than in the auto industry. Automakers use Cray systems for diverse applications that range from modeling external aerodynamics, fuel flow, and passenger compartment ventilation to conducting crashworthiness and other structural engineering analyses.

Feature articles in this issue of CRAY CHANNELS demonstrate the broad range of supercomputer applications in the automotive industry and offer valuable insights for a variety of engineering environments. In this issue we also include a report explaining the latest features of the Cray Fortran compiler CFT. Our regular departments describe new versions of engineering analysis software for Cray systems and engineering applications being developed on Cray systems at U.S. national laboratories.

Using Cray computer systems, computational researchers are redefining the state-of-the-art in industrial design and engineering. Automakers on three continents are using Cray computer systems to create innovative, economical, and attractive new vehicle designs. By shortening design turnaround times and reducing the need for costly trial-and-error prototyping, Cray systems will continue to provide a decisive advantage to industrial designers and engineers.



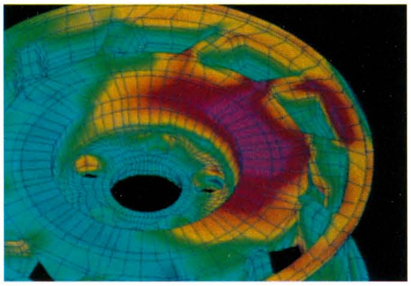
On the cover is a test board used to test 16-gate gate arrays used in CRAY X-MP and CRAY-2 computer systems. Testing on the boards is the final processing of the gate arrays prior to actual use, and is performed after they are subjected to a burn-in cycle that simulates accelerated aging. The boards test all electrically functional parts of the gate arrays. Approximately 135,000 gate arrays are tested by Cray Research each week.

CRAY CHANNELS
A Cray Research, Inc. publication

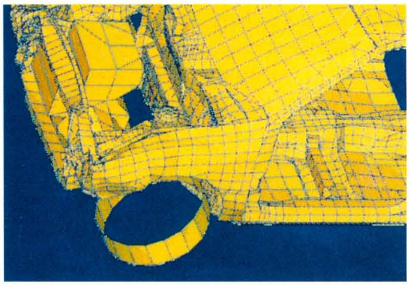
Winter 1987

FEATURES

DEPARTMENTS



2



8

CRAY CHANNELS is a quarterly publication of Cray Research, Inc. intended for users of Cray computer systems and others interested in the company and its products. Please mail subscription requests, feature story ideas, and news items to CRAY CHANNELS at Cray Research, Inc., 608 Second Avenue South, Minneapolis, MN 55402.

Volume 8, Number 4
Winter 1987

Editorial Staff
Tina Bonetti
Ken Jopp
Carol Smith
John Swenson

Design
Eric Hanson
Cynthia Rykken

CAE AND SUPERCOMPUTER APPLICATIONS AT NISSAN

Mizuho Fukuda, Nissan Motor Co., Ltd., Kanagawa, Japan

The availability of supercomputers, computer graphics, and improved software has spurred the growth of computer-aided engineering (CAE) in the automotive industry. Large-scale structural analyses and simulation are being used increasingly to evaluate performance and reliability during the initial stages of vehicle design. This capability is becoming crucial as automakers encounter growing international competition, new laws covering exhaust, safety, and noise, and public demands for higher performance and greater fuel efficiency. To expand CAE capacity at Nissan, a CRAY X-MP/11 supercomputer was recently incorporated into the company's CAE system. This article describes Nissan's CAE system and discusses some of the application areas in which the system is used today.

Early CAE at Nissan

Development of a full-fledged analytical system at Nissan began in 1972 with the introduction of the MIT/STRUDL program. Initially, structural analysis using the finite element method (FEM) was applied to car body strength and rigidity problems. However, increased computer processing capacity and advances in software allowed analyses to grow in scale and complexity. Analytical models grew from two to three dimensions, and analysis advanced from linear to nonlinear, from individual parts to components and vehicle systems, from static to dynamic analysis, and from shell parts to solid parts. In addition, a CAE system was constructed for performance simulation.[1]

Nissan initially used analytical techniques for troubleshooting; that is, for finding countermeasures to defects or in response to market complaints. Now, however, analytical techniques increasingly are used for planning, where the methods are showing their real effectiveness. Use of computer time for such analysis is increasing at an annual rate of 30-50 percent at Nissan. The introduction of a supercomputer in 1985 extends this trend to areas where computer processing speed had been an obstacle to developing such methods. These areas include external aerodynamics, combustion, vehicle system vibration and noise, impact strength, manufacturing process simulation, and human engineering.

CAE system configuration

Along with the CRAY X-MP/11 system, hardware used for technical computation at Nissan includes seven large-scale and six medium-sized computers. Connected to these are approximately 1000 terminals, including graphics and character terminals. The large-scale computers are for applications systems such as CAE, computer-aided design (CAD), development management, and product specifications data management. The medium-sized computers are used for laboratory automation. They act as host computers for some 90 minicomputers, and are distributed among test courses and other locations.

The Cray supercomputer is used exclusively for batch processing in structural analysis and large-scale simulation, and can take jobs from any terminal. A graphics system connected to CAD computers, using CAD data and high-precision graphics terminals, performs the pre- and post-processing for structural analysis performed by the Cray system. These terminals are shared with the CAD system.

Medium-scale computers included in the analytical system are used for various performance simulations required mainly for time sharing system processing, besides serving as front-end computers for the Cray system. Experimental data is collected and processed by the laboratory automation computers and then sent to the analytical computers.

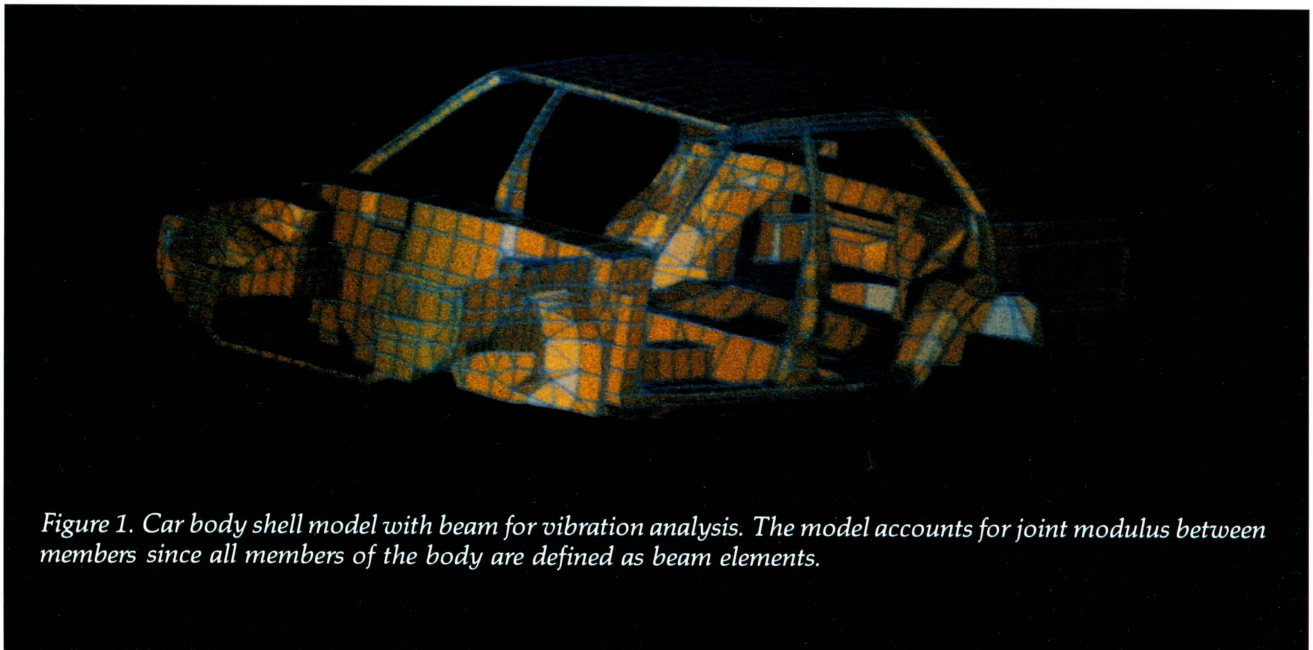


Figure 1. Car body shell model with beam for vibration analysis. The model accounts for joint modulus between members since all members of the body are defined as beam elements.

Role of the supercomputer

Nissan needed a supercomputer for two reasons. One reason was the need to process large volumes of analytical data produced on the CAE system. Centralization of the CAE system during initial design stages, rather than simply troubleshooting, had increased the amount of analytical processing that was required. The second reason was the need to handle large-scale simulations that evolve from research and development efforts.[2] Large-scale simulations previously were compromised by insufficient computer power, despite improved techniques for analyzing aerodynamics, combustion, and other factors.

For these reasons Nissan decided to obtain a supercomputer with high-speed vector processing ability and superior price/performance. The present processing capacity, based on an overall use of programs such as MSC/NASTRAN, MARC, and large-scale Fortran programs on the supercomputer, is about ten times greater than would be provided by conventional large-scale computers such as the IBM 3081 computer.

Even programs requiring mostly scalar calculations are being run more often on the Cray system as a way of increasing processing speed. Considering the particular nature of the Cray supercomputer, however, its use is mainly concentrated on large-scale linear and nonlinear structural analysis and on aerodynamic and other large-scale simulation. The remainder of this article describes typical applications for which the Cray system is used.

Vehicle system vibration and noise analysis

Vibration and noise considerations have become increasingly important factors in quality improvement. At the same time new analytical techniques and greater computer capacity have made possible large-scale and highly precise analysis not only of individual parts but also of component and vehicle systems. As a result, attention is focusing on factors such as booming noise, road noise, and differential noise, which until recently were exceedingly difficult to analyze. The supercomputer has made possible the use of highly detailed large-scale models for such analysis and has improved computational precision.

Interior sound field analysis

Booming noise occurs when vibrations from the engine and other sources are transmitted to the car interior, causing skeletal and panel vibrations that create sympathetic air vibrations in the passenger compartment.

The system used at Nissan to study vehicle sound fields consists of two subsystems, one for uncoupled analysis, which considers the sound field in relation to vibration input from the vehicle's structural system, and the other for coupled analysis, which considers the influence of the sound field on the structural system vibrations. MSC/NASTRAN and CAEDS, which are widely-used FEM analysis programs and are thus easy for design analysts to become familiar with, are used in this system in conjunction with internally created programs.

Precise analysis of the interior sound field depends in many respects on the precision of the vehicle body model. The strength, rigidity, and vibration characteristics of the body are normally calculated using a beam model, shell model with beam, large-scale detailed shell model, and so on. The present analytical system uses a shell model with beam, with a total of approximately 1200 nodal points, based on considerations such as calculating time, man-hours for model construction, and degree of precision.

Since all members of the body are defined as beam elements, the model accounts for joint modulus between members. The regions where joint modulus is used are selected on the basis of the strain energy of the various joints. Joint modulus values are derived from calculations using a detailed model of the joint regions. These results are incorporated into the body model as spring elements, then the vibration characteristics of the body are calculated. Comparison with experimentally derived values shows the simulation results to be within a ten percent error margin in the low frequency range, an acceptable level of accuracy.

Figure 1 shows a body model, while Figures 2 and 3 give typical calculations of body deformation mode and vehicle compartment acoustic mode, considering the primary resonant frequency of flexural vibration of the body. Results of the interior sound field analysis are used to research ways of structuring the body to reduce booming noise. Since many variables are involved in this study, it requires a highly versatile and sophisticated system.

Calculation of body vibration accounts for a large percentage of the overall analysis time for booming noise, but the Cray supercomputer has cut this time to only one-eighth of that previously needed.

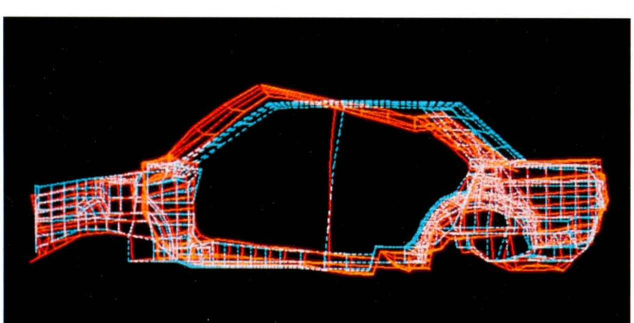


Figure 2. Body vibration mode at its primary resonant frequency.

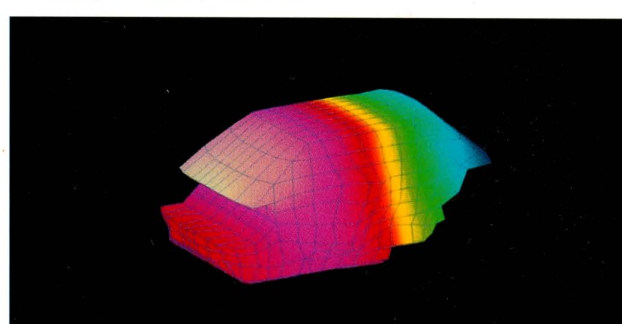


Figure 3. Acoustic mode of the vehicle using the body's primary resonant frequency of flexural vibration.

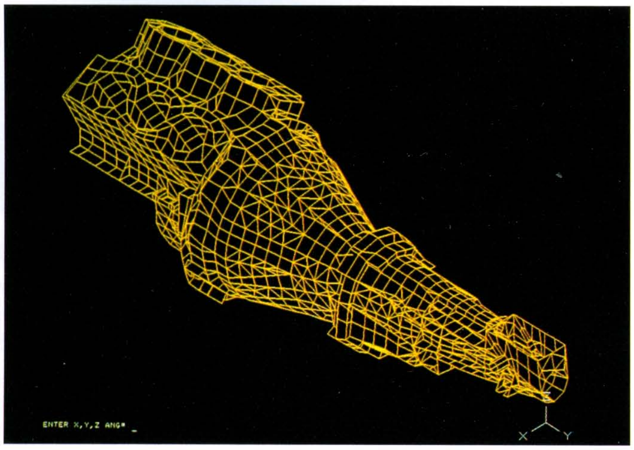


Figure 4. Analytical model of the engine of a front-engine, rear-drive vehicle. The model contains 2800 nodal points.

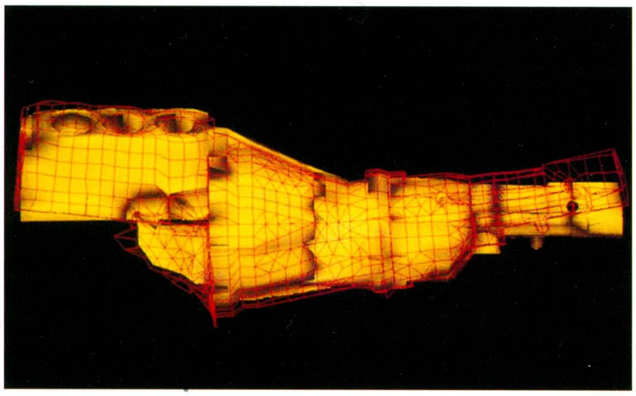


Figure 5. Vibration mode of the engine at the primary resonant frequency of flexural vibration.

Engine vibration analysis

Engine vibration characteristics have a major effect on noise inside a vehicle. The conventional approach to this analysis has been to conduct vibration analysis of each of the component parts, such as the cylinder block and transmission case, using FEM. More recently engine analyses using the building block approach (BBA) have also been attempted.

An analytical model of the engine of a front-engine, rear-drive vehicle is shown in Figure 4. Figure 5 gives the results of the vibration mode calculation at the primary resonant frequency of flexural vibration. This kind of analysis requires construction of a large-scale model for each component part. One also must consider countermeasures such as additional ribs. The number of nodal points used in the example was 2800, and 3800 elements were used. Analysis of this scale and complexity has been made more feasible through the use of BBA, super-element and other techniques, and the supercomputer.

Tire vibration analysis

The tires of an automobile are always in contact with the road and transmit the power of the vehicle to the road. Therefore, the tires directly affect general vehicle performance. The tires also are an important key to the study of vibration and noise performance, including ride comfort and road noise.

Tires are formed of layers of rubber and fiber-reinforced rubber. Considering their structure and use conditions, structural mechanical analyses can be highly complex. Until recently, numerical analysis of tires involved extremely simplified structural models, and the typical approach had been to build models from experimental results. With the rapid advances in computer performance, however, it has become possible to create more faithful models of tire structure using FEM. The supercomputer has made such analyses much more practical.

A tire model is shown in Figure 6. Figure 7 shows the calculated results for an unrestrained tire vibrated vertically at one point on the tread surface. Results show the vertical response of the opposite side of the tread surface. In this example, the model treated the rubber material as a solid element and the belt and carcass as shell elements. Also, because the calculations were made using a one-fourth scale model that takes into account symmetry, the number of nodal points is approximately 1800 and the number of elements is approximately 2000.

Aluminum wheel analysis

Vehicle wheels, aluminum wheels in particular, must fully satisfy styling needs while meeting strength and weight requirements. They also must meet vibration requirements to minimize road noise.

Aluminum wheels, because of their thickness, must be modeled using solid elements. Their shape is complex, with sudden changes in thickness and many windows and ribs. In addition, a far greater number of elements are required to analytically predict the stress concentration regions than in the case of steel wheels, making the analysis large-scale.

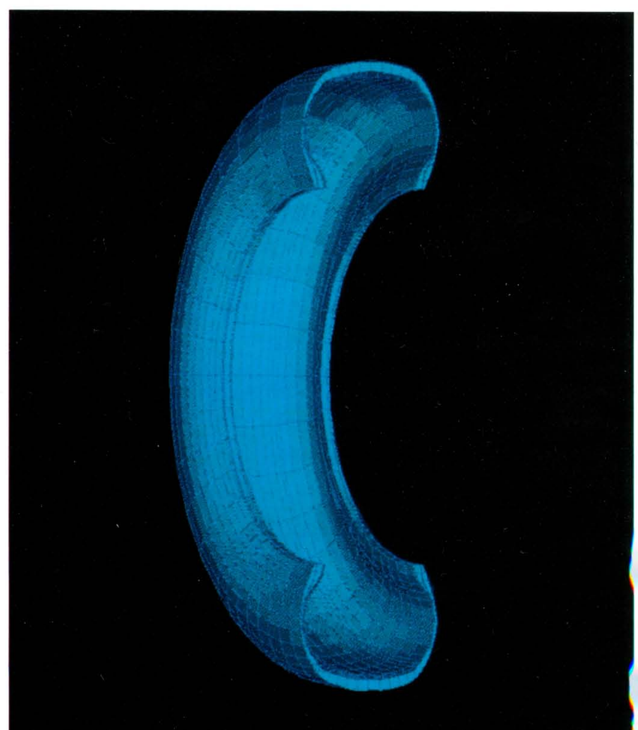


Figure 6. Analytical model of a tire. The model treats the rubber as a solid element and the belt and carcass as shell elements.

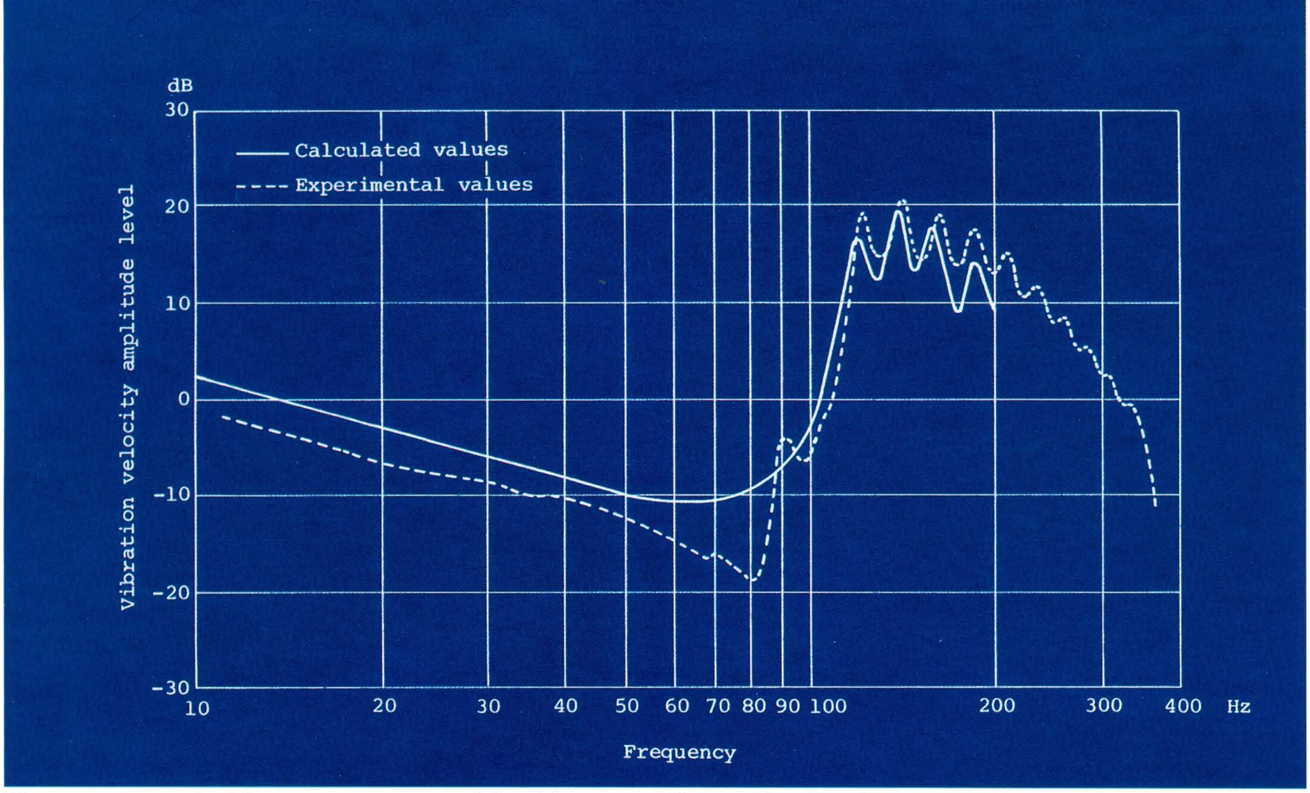


Figure 7. The vertical frequency response of the side opposite the tread surface for an unrestrained tire vibrated vertically at one point on the tread surface.

An example of a three-dimensional solid model to study impact strength is shown in Figure 8. In accordance with SAE standard J175, stress distribution is shown when a load is applied at a 30° incline to the perpendicular. Figure 9 shows the primary resonant mode using the same model. In this example there is no symmetry in the shape, so the analysis was conducted using a 1:1 model with approximately 5000 nodal points and approximately 3200 elements. For eigenvalue calculation, about 20 minutes of CPU time were required on the Cray supercomputer. Introduction of this computer system has made it possible for the first time to conduct practical studies of strength and vibration of aluminum wheels.

Performance simulation

Large-scale simulations, such as aerodynamic, crash, and combustion analyses, involve difficulties not only in terms of analytical techniques, but also in terms of the calculation time they require. Until recently such simulations could barely be carried out with adequate precision. This is the area in which the supercomputer is being counted on to show its real effectiveness. Whereas in the past most of this type of analysis at Nissan was forced to make use of simplified methods, introduction of the supercomputer has allowed the models to become three-dimensional and more detailed.

Flow analysis

A vehicle's external aerodynamic characteristics greatly influence fuel consumption, maneuverability, and performance at high speeds. Normally these aerodynamic characteristics are investigated using the Navier-Stokes equation, which is the basic equation of fluid motion. Even with today's computer capability, however, it is difficult to achieve a complete solution of this equation, and so two other methods are used as approximate solutions.[3,4]

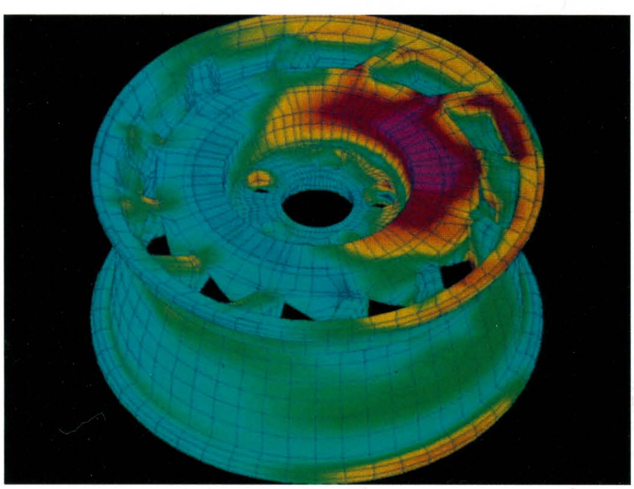


Figure 8. Stress distribution of an aluminum wheel. The load is applied at 30° to the perpendicular.

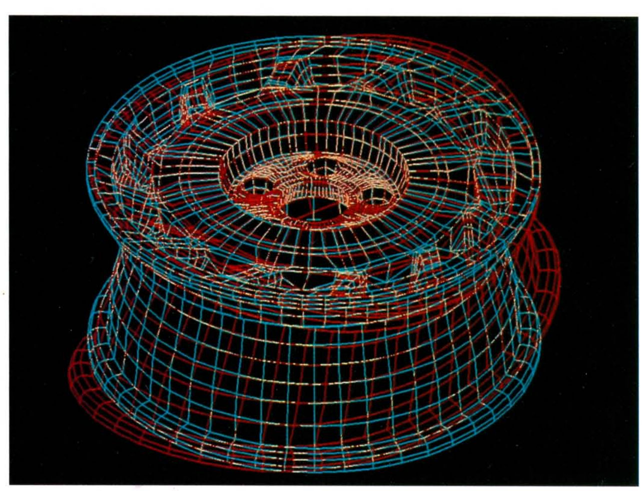


Figure 9. Vibration mode of an aluminum wheel at primary resonant frequency.

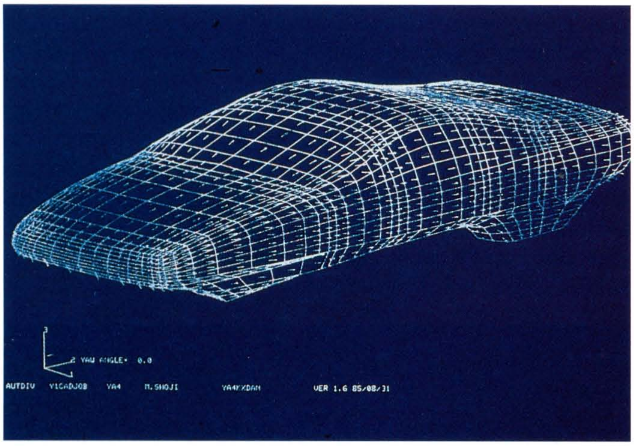


Figure 10. Flow velocity distribution determined by aerodynamic simulation using the panel method.

One of these approaches involves creating a model from body shape data at the styling stage via linkage with the styling CAD system, and performing calculations using the panel method. Flow velocity distribution determined by this method is shown in Figure 10. These results agree closely with experimental results. The panel method, using a CAD system or other means, requires a faithful rendition of the body surface only for a panel segmentation. Since the calculations assume a flow without a vortex, however, this method cannot deal with separation at the rear end. A practical level of precision can be achieved nonetheless by postulating a wake extending from the body along the rear window.

Simulation by the difference method is another alternative to solving the Navier-Stokes equation, and is one that can account for vortex flow. Simulation using the difference method has until lately been virtually impossible because of the large amount of calculation time required. Here again the Cray supercomputer has resulted in great progress in both technology and practicality. Results of a two-dimensional simulation of flow around the body at a velocity of 100 kilometers per hour are shown in Figure 11. Whereas a three-second simulation using the IBM 3081 computer required 10 hours of CPU time, with the Cray system this computation is possible in only 20 minutes.

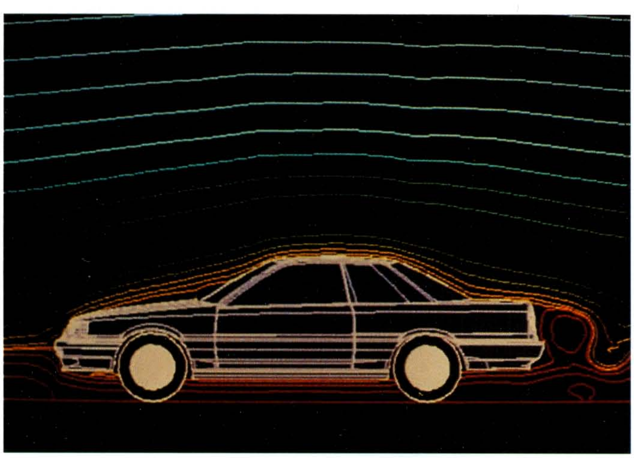


Figure 11. Two-dimensional simulation of flow around the body at 100 kilometers per hour using the difference method.

As with the aerodynamic characteristics of the body, flow analysis through the vehicle compartment has also become far more practical. The simulation results shown in Figure 12 are for the cooling of the passenger compartment of a vehicle left in the sun. The phenomenon simulated here is the blowing of cold air at 5°C through a dashboard duct, at an initial interior temperature of 50°C. Twenty CPU hours were required on the IBM 3081 computer to model this three-second phenomenon, but using the Cray system has reduced the time to 40 CPU minutes.

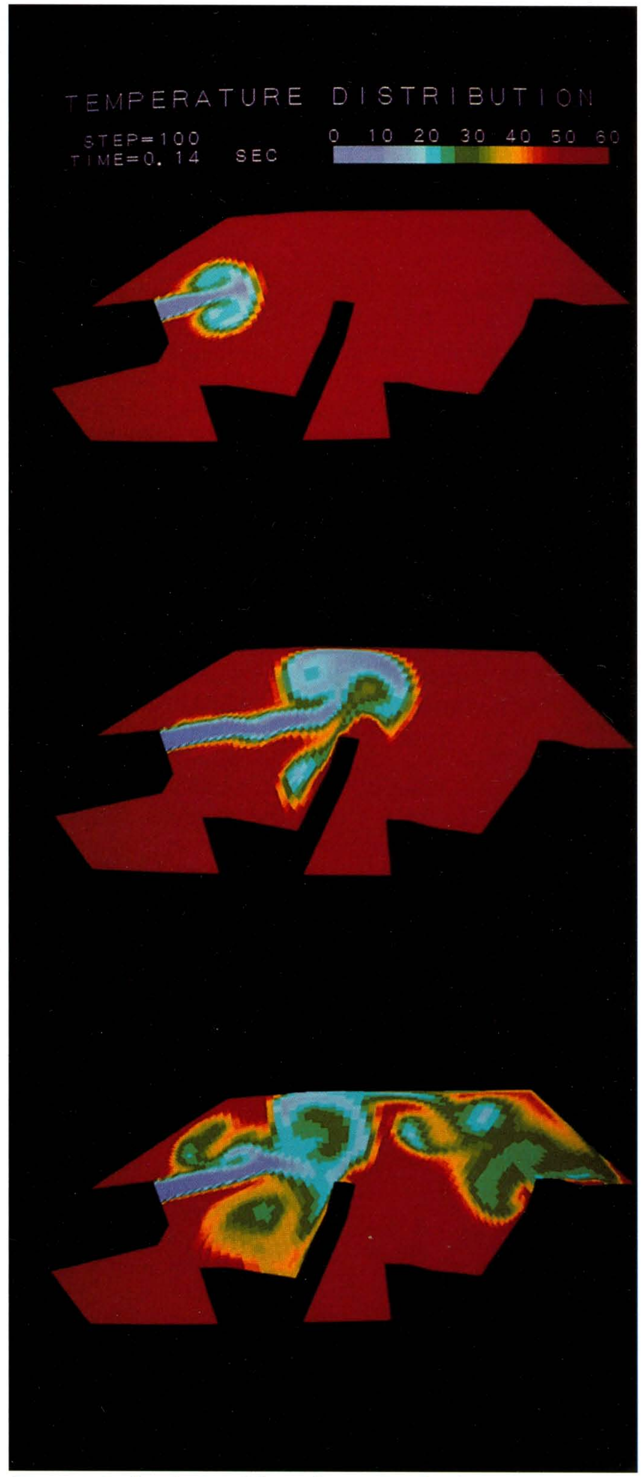


Figure 12. Analysis of flow through vehicle compartment. Simulated air at 5°C is shown flowing through a dashboard duct; initial compartment temperature is 50°C.

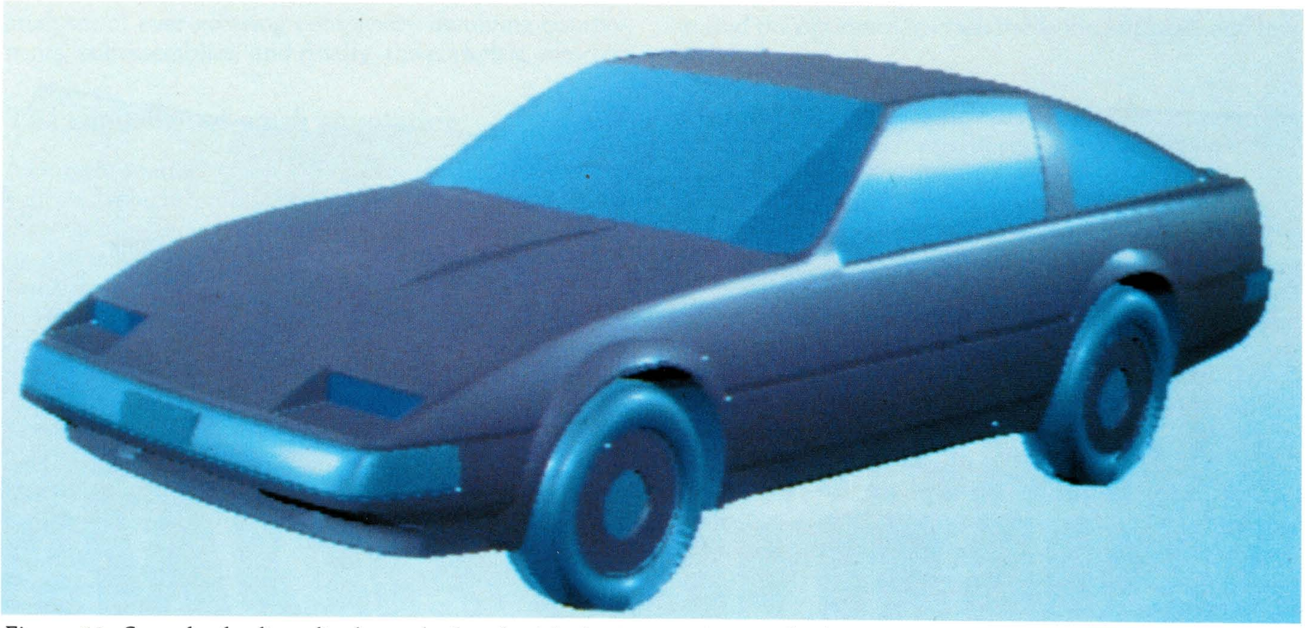


Figure 13. Sample shading display calculated with the ray-tracing method.

Image processing

Use of CAD in the styling stages has progressed along with the development of CAD/CAM techniques. At Nissan a styling CAD system (DIMS) was developed as an aid to the creative activities of designers.[5]

DIMS is aimed at converting the styling process from the conventional method centered on clay models to a data model approach centered on data. Since use of a data model makes aerodynamic simulation possible at the styling stage, it is useful also in that it brings together the consideration of style and aerodynamic performance.

Once the vehicle surface shape has been defined by means of some 200-300 adjustable curved surfaces derived from 15,000 to 20,000 flat elements, DIMS uses the ray-tracing method to generate images of the vehicle, such as the shaded image shown in Figure 13. As part of the CAD system, the DIMS system is run on CAD computers. For shading, which requires a long processing time, the Cray supercomputer is being used to improve processing efficiency.

Future directions

Supercomputers are making possible high-speed processing of important application programs. Their use can accordingly be expected to raise the level of the CAE system as a whole, improving analytical techniques and making analysis tasks more efficient.

Integrating systems to predict and evaluate overall vehicle performance in an integrated fashion and with a high degree of precision and sophistication will require not only further development of individual analytical techniques, but also a systematic integration of all the related programs, centering on a comprehensive engineering database. The use of artificial intelligence techniques to accumulate and make analytical knowhow readily available is also indispensible.

Among hardware supporting the CAE system, workstations for graphics processing and interactive computing will likely become more prevalent, and should take over a portion of the processing load of the system. The supercomputer will no doubt be made even faster, contributing its might all the more to areas requiring high speed large-scale processing. □

Acknowledgment

This article is adapted from the author's presentation at the International Conference on Supercomputer Applications in the Automotive Industry, October 1986, Zurich, Switzerland.

About the author

Mizuho Fukuda is a CAE manager in the product development systems department at Nissan Motor Co., Ltd. Fukuda received his B.S. degree in 1968 in mechanical engineering and his M.S. degree in 1970 in mechanical engineering, both from the University of Tokyo. He joined Nissan Motor Co., Ltd., in 1970 and has since worked on structural analysis and vehicle performance simulation in the product development systems department.

References

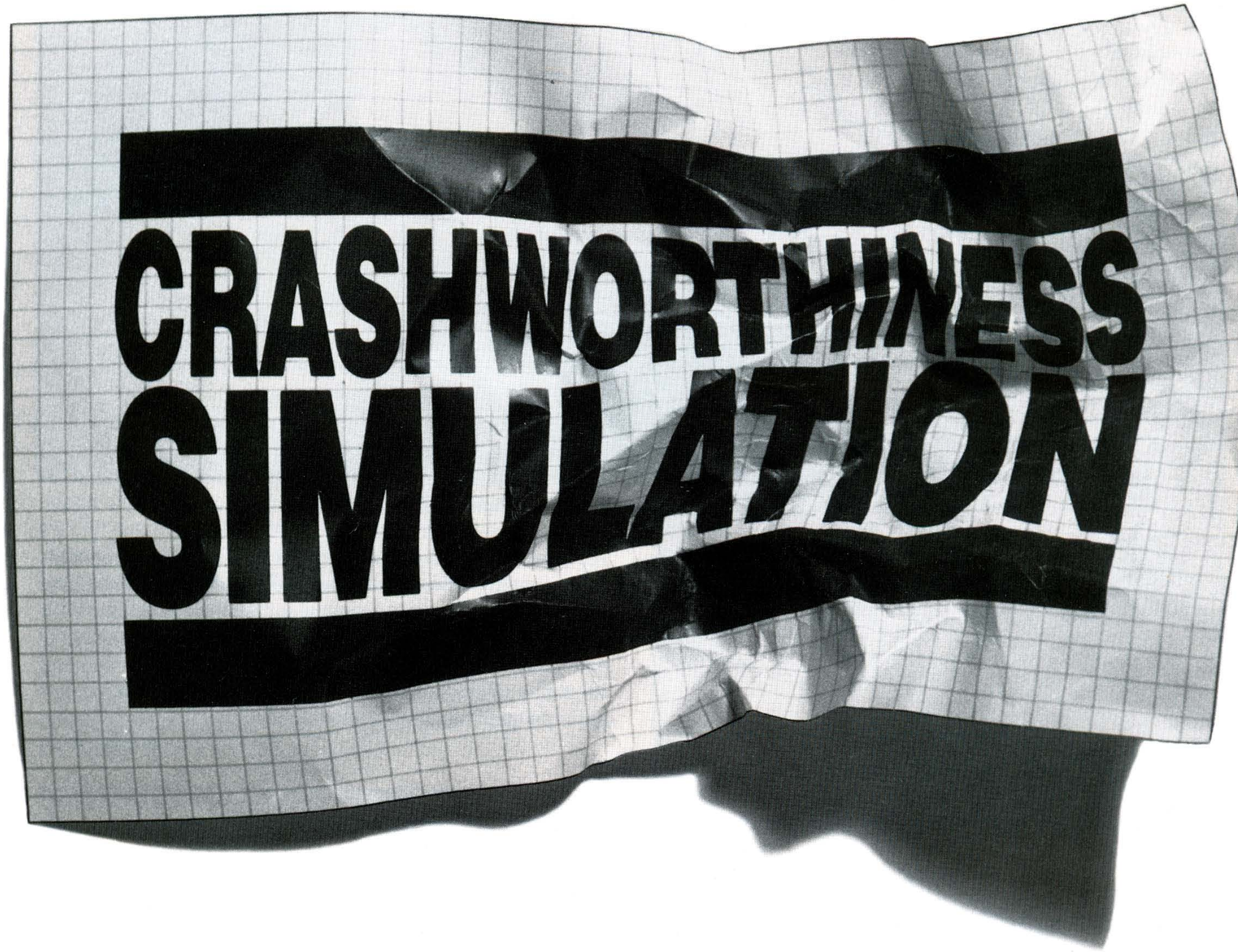
1. Fukuda, M., Y. Hori, "CAE for Design of High-Performance Cars," (in Japanese), *JSAE Symposium*, 1986.

2. Fukuda, M., "Use of a Supercomputer in Structural Analysis," (in Japanese), *Information Processing*, Vol. 22, No. 12, 1981.

3. Himeno, Y., M. Takagi, "Numerical Analysis of Car Body Aerodynamic Characteristics by the Panel Method," (in Japanese), *JSAE Proceedings* 842, 1984.

4. *Nikkei Computer Graphics*, (in Japanese), special inaugural issue, 1986.

5. Mori, K., et al., "Color Shading Technology for Design CAD Systems," SAE paper (to be published), 1986.



an emerging tool for vehicle design optimization

Ernst J. Nalepa and Hung Le-The, Adam Opel Aktiengesellschaft, Rüsselsheim, West Germany

The development of mass-produced automobiles is a continually growing and changing process temporarily influenced by increasing customer demands, technological progress in fields such as electronics and control techniques, changes in traffic policy and the general sociological environment, and the increasing automation of manufacturing.

The marketplace demands continuing product improvement and a proliferation of car model lines — demands which can only be met through the judicious application of computers. With the introduction of digital mainframes, the computer-aided design analysis of automotive vehicles has gained considerable importance. Rather than attempting to predict or evaluate the causes of structural failures, emphasis has turned to optimization in the initial design stages. In other words, the major task of design analysis is to arrive at the best possible combination of vehicle design parameters that satisfy the engineering targets — before committing designs to costly prototypes.

An acceptably accurate prediction of structural behavior is very difficult to obtain, requiring at a minimum a sufficiently detailed simulation model (analysis model), and a precise simulation of the physical phenomena. In addition, there is also a constantly increasing variety of analyses of ever-growing complexity involving components, subassemblies, and finally, the complete vehicle.

The challenge of crash simulation

Among the many types of analysis necessary in the design process, vehicle crashworthiness simulation has become increasingly important in recent years. Heightened public awareness and tighter government guidelines throughout the world have prompted vehicle manufacturers to look much more carefully at the crash behavior of their designs. To date, most crash behavior information has been obtained experimentally at the prototype stage of the design process. Vehicle prototypes are crashed into barriers at various speeds, and then the vehicles are examined to determine the patterns of deformation and stress. This is much too late in the design process, however, and allows only limited optimization of the design to reach engineering targets.

In 1983, several West German automobile manufacturers undertook a joint crash analysis effort in an attempt to address this issue. The first positive results of that effort became available in late 1985. At about the same time, Opel purchased and installed a CRAY-1/S computer system and began its own crash analysis program.

Vehicle structures must be dimensioned to withstand impacts at low speeds with limited permanent deformation. The design must therefore be optimized for impact attenuation and its ability to absorb kinetic energy. At the same time, however, design engineers are also seeking to minimize weight, increase driving comfort, and improve the overall performance of vehicles — design criteria that oppose crash performance needs. The stiffness of the siderails and chassis, for example, is directly related to driving comfort; components must have well-defined structural stiffness to ensure ride comfort and fatigue strength. However, stiff components in the chassis-system and the siderails absorb very little kinetic energy, thus offering very poor crash behavior. As a result, the design engineer must evaluate many different design alternatives to find the optimum in crash behavior, stiffness, and low weight.

The crash phenomenon is both spatial and temporal. The structural analyst is forced to consider not only nonlinear effects with large strains and large displacements but also surface contact with contact forces in time domain. For example, the optimum crash behavior would be much easier to obtain if the velocity of the center of gravity decreased in a linear fashion from beginning of impact to rebound. Vehicle crashes, however, are highly nonlinear (when one calculates the varying crash behavior of, for example, engine contact with the firewall, engine barrier, and wheel-wheelhouse contact as seen in Figure 1), making it impractical to attempt to achieve an energy absorption that is constant over time.

Dynamic simulation of crash events by the finite element method can apply an implicit or an explicit solution algorithm, or a combination of both types. The essential difference between the two types of algorithms is the solution strategy used to integrate the nonlinear equations of motion (for example, the differential equations).

Opel has implemented PAM-CRASH in its computer facility. PAM-CRASH uses the explicit method, which writes the equations of motion in a form that yields nodal accelerations, from which the central difference formulas are applied to obtain nodal velocities and displacements. This approach uses an efficient lumped mass matrix formulation. The solution advances through the impact duration using a small timestep based on the wave propagation time across the smallest mesh element. This ensures that the physical phenomena, including stress wave effects, are followed completely and that the solution is stable.

The path- and time-dependent nature of the crash problem makes its solution very time-consuming. For an analytical simulation of a frontal impact, the use of a supercomputer is a prerequisite. Only with such computational power can crash optimization of the structure be achieved within a reasonable timeframe.

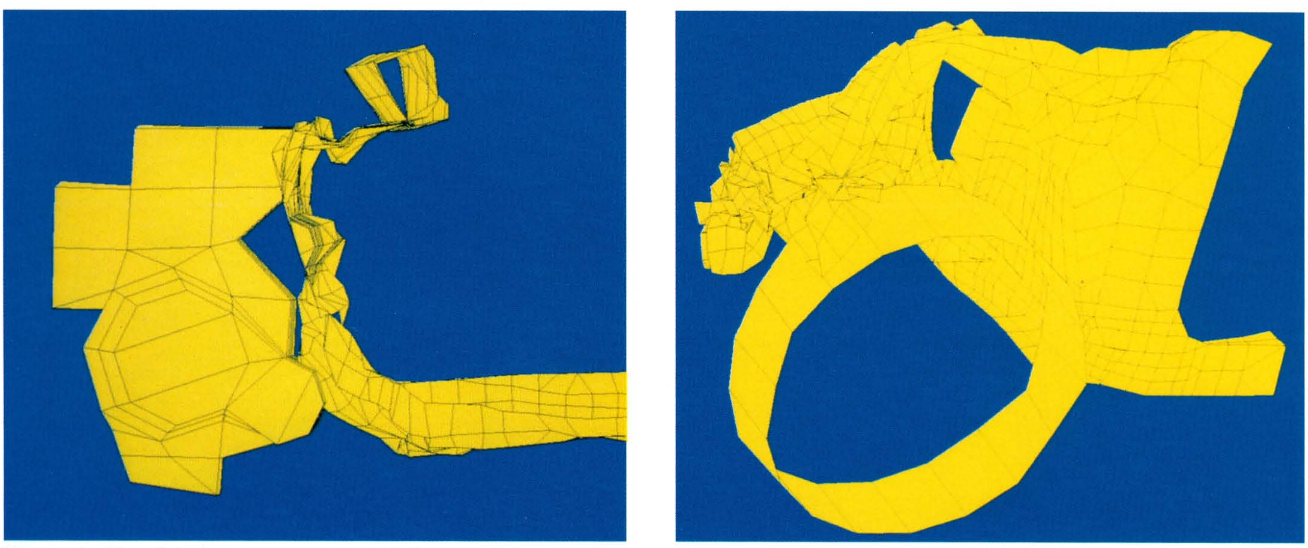


Figure 1. Graphic representations of the complex nonlinearity of the crash problem are shown above. Deformation of the firewall is shown as the engine crashes into it during impact (left). Contact of the wheel with the wheelhouse is shown during frontal impact; the complex steering column was modeled using beam elements (right).

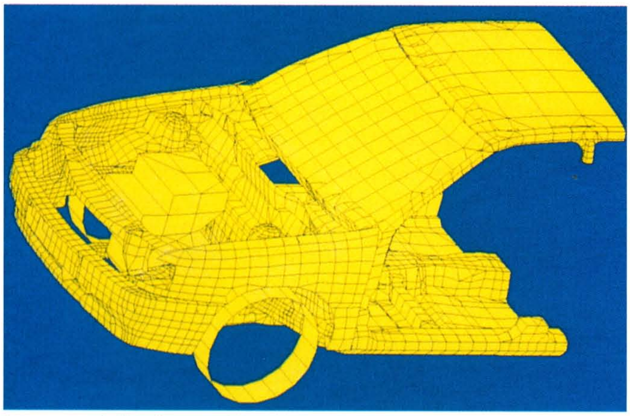


Figure 2. The finite element model for an analytical simulation of a frontal impact.

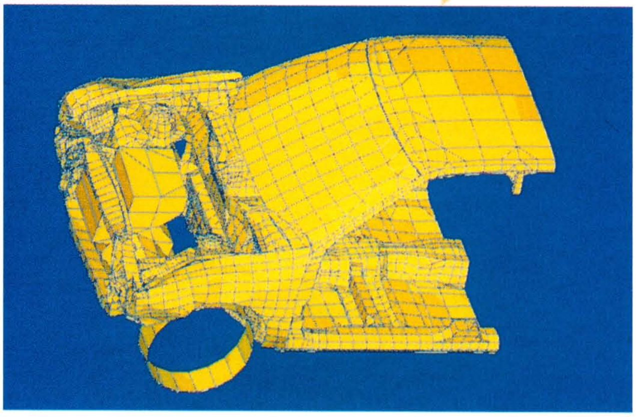


Figure 3. Deformation of the vehicle structure 30 milliseconds after impact.

Crash simulation of a new vehicle design

Opel is currently developing a vehicle for the European market. Crashworthiness guidelines in West Germany and other countries are very similar, as are the market demands for performance and fuel economy. As outlined at the beginning of this article, the design of any new vehicle must find the optimum in crash behavior and low weight. However, the time period to bring this vehicle to market is very short, and the costs for building prototypes are prohibitively high. We therefore set about the task of simulating the vehicle's characteristics very early in the design process.

In our study of a 50 kilometer-per-hour frontal impact, a large finite element model (Figure 2) was necessary to obtain results useful for further vehicle development. The model consisted of 7911 shell elements and 7233 nodes.

The objective to achieve a good correlation between analysis and experiment necessitated an accurate description of the dynamic parameters such as the location of the center of gravity, masses, and mass inertias.

Figure 3 shows deformation of the vehicle 30 milliseconds after impact. Crash simulation can provide the design engineer with important data on the deformation behavior of various vehicle components and their interaction during a crash. For example, it can be observed by simulation that the siderails greatly influence vehicle crash behavior, because more than 40 percent of the kinetic energy of a frontal impact will be absorbed by the siderails. Additionally, the crash simulation can yield useful information on global variables such as forces, velocities, and deformations, further enabling the development engineer to evaluate the crash behavior of different structural modifications at an early stage.

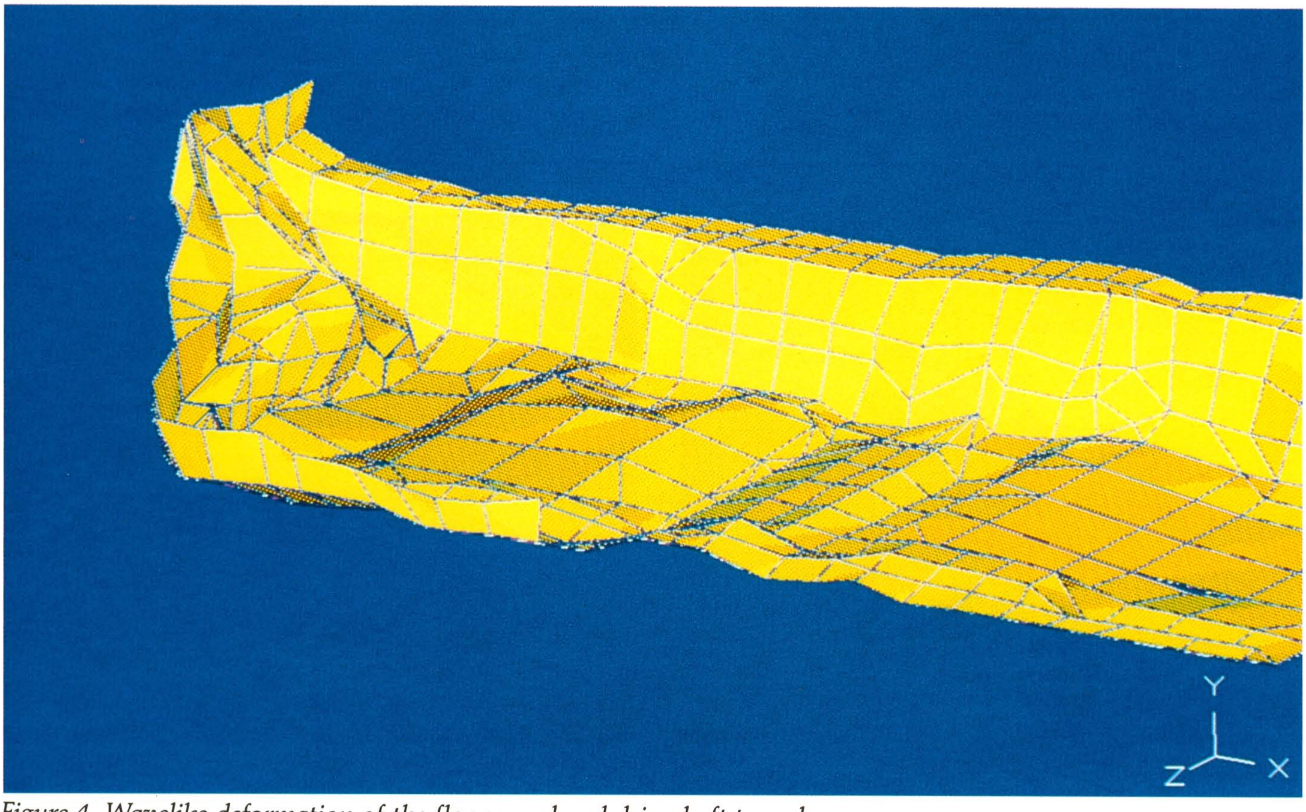


Figure 4. Wavelike deformation of the floor panel and driveshaft tunnel.

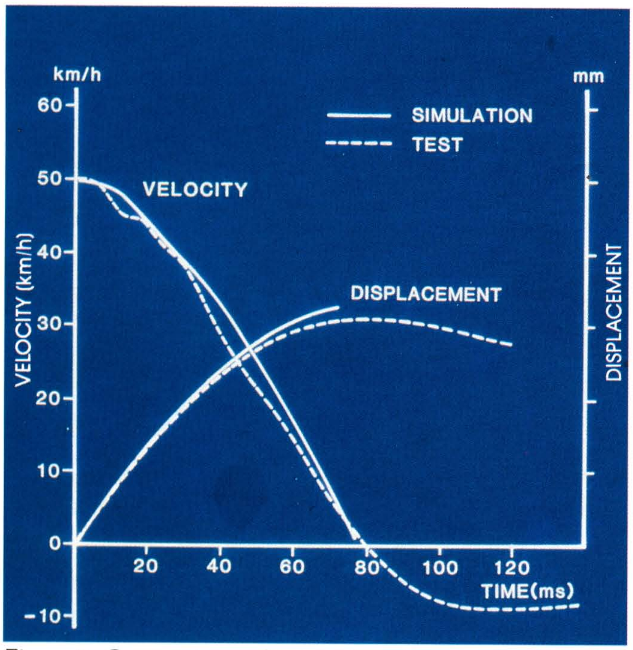


Figure 5. Comparison of test and analysis results for deformation and velocity over time for the 50 kilometer-per-hour frontal impact.

The investigation of different vehicle variants yields almost immediate answers simply by modifying the existing analysis models. For example, different engine-transmission combinations can be examined with hardly any extra effort except for the need to realistically simulate the engine contact with the remaining structure, particularly with the firewall. The calculated deformation of the floor panel front resulting from the transmission crashing into it also provides an early insight into the energy conversion of the passenger compartment (Figure 4).

The behavior of the steering system under crash conditions is of utmost importance in terms of safety. For the simulation to yield good results, the simulation model must include the tire model with an accurate description of the contact with the inner wheelhouse, and the models of the steering knuckle and the steering linkage. In addition, in order to avoid impact of the driver's head with the steering column, the column's tip must stay within a given rectangular volume during the crash. As a result, the displacement of the top of the steering column becomes an important design parameter.

Putting it all together

Figure 5 shows a comparison of experiment versus analysis of deformation propagation, velocity absorption, and the acceleration response at any instant over the crash duration. Figure 6 shows the deformation of the finite element model and the actual test vehicle. Both examples indicate the excellent agreement between test and analysis, made possible by the large model used in the analysis.

Also, it must be mentioned that for accuracy reasons (round-off error accumulation) only long word machines such as Cray computer systems are suitable for running crash simulation programs. Large memory (1 million words minimum) and high storage capacity are necessary as well.

Computers will not replace the creativity of engineers. They will, however, support a systematic and extensive integration of up-to-date knowledge into a vehicle layout. In the future, supercomputers will place Opel in a position to perform such activities in a more complete and timely manner. Above all, they will move the forefront of computerized analysis one step further. □

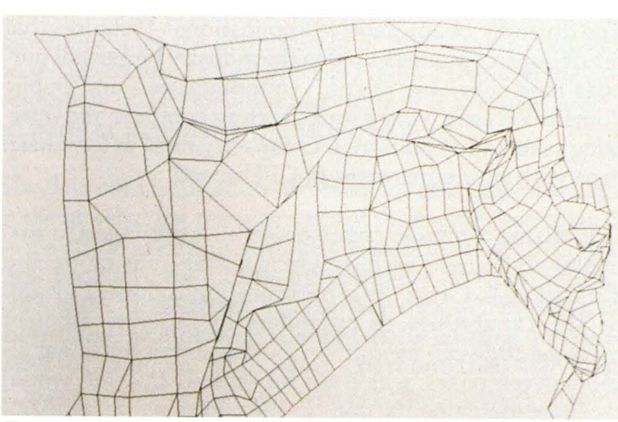
Acknowledgments

This article is adapted from the authors' presentation at the International Conference on Supercomputer Applications in the Automotive Industry, October 1986, Zurich, Switzerland. The authors would like to acknowledge Engineering Systems International, developers of PAM-CRASH, for their assistance in the implementation of the Adam Opel crash analysis program.

About the authors

Dr. Ernst J. Nalepa is assistant director for static and physical analysis in the Technical Development Center at Adam Opel AG. Prior to joining Opel five years ago, he served in the department of technical engineering at Man Company in West Germany. He received a Ph.D. in mechanical engineering from the University of Berlin in 1980.

Hung Le-The has worked in the static and physical analysis department at Adam Opel AG since 1983, specializing in crash analysis. He received a degree in mechanical engineering from the Technical University of Berlin in 1983 and is currently completing doctoral work at the University of Berlin.



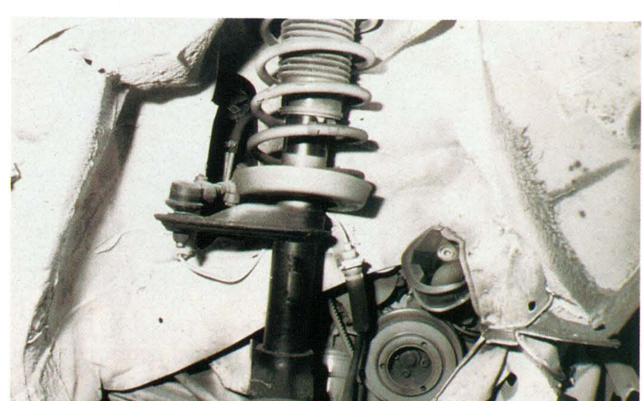


Figure 6. Deformation behavior of the right siderail: the finite element analysis (left) was confirmed by later testing (right).

ENGINE COMBUSTION MODELING: prospects and challenges

Keith Meintjes, General Motors Research Laboratories, Warren, Michigan

Engine combustion modeling requires large computer codes and supercomputers to calculate the flows, turbulence, and chemical processes occurring inside the cylinders of reciprocating internal combustion engines. While this is an extremely complex task, it is encouraging to reflect on the advances of the past few years. A number of challenges remain, but the models are progressing to the point where they can have a major impact on combustion system design.

The primary attribute of these models is their ability to resolve the physical processes in time and in three-dimensional space. The models are based on fundamental laws of physics, but do contain empirical submodels for certain phenomena. A research goal is to incorporate in the submodels as much physical realism as possible. The models can then be used to study the interplay between various physical processes and gain understanding that can be used to improve engine performance. The models also can be used to interpolate and explain experimental data and, with some caution, to make predictions in areas where experimental data do not exist.

Multidimensional combustion models were first used at the General Motors Research Laboratories (GMR) in the late 1970s. GMR applications have included conventional homogeneous-charge, spark-ignited (SI) engines, diesel engines, and direct-injection, stratified-charge (DISC) engines. Diesel and DISC engines include the complications of liquid fuel sprays and nonhomogeneous combustion. Multidimensional models are required if we are to gain a deeper understanding of such nonhomogeneous processes and then contribute to the development of these engines. Indeed, the combustion systems of direct-injection engines require significant improvements if they are to meet future emissions regulations and become feasible alternatives to the SI engine.

The potential for using multidimensional models to improve conventional homogeneous-charge engines is less obvious because the modern passenger car engine is already highly optimized. Improvements continue, but in small incremental steps. Big gains are not possible; predicting a "small" gain (say one percent in fuel economy) demands a highly accurate and complete model. Also, zero-dimensional engine simulations can model most aspects of homogeneous engine combustion and treat the engine as a system, including flows through the intake and exhaust systems. Yet engine simulations neglect many details of engine fluid mechanics and depend on empiricisms, inhibiting their use in the absence of experimental data.

The greatest impact of multidimensional models may be to reduce the design cycle time, rather than to advance the state of the art. Compared to experimentation, modeling is extremely time- and cost-effective and the models can be used to refine designs and screen different hardware proposals to minimize cut-and-try development efforts.

What we might call first-generation three-dimensional models are already available. We have enough, albeit limited, experience with these models to lay out the specifications for a second generation of more general and flexible models. Some of the specifications will be discussed in the remainder of this article.

In addition to actually running the code, the task of using a large computer model involves two steps, preparation and interpretation. A particular application may take a few hours to run, but preparation (usually grid generation) and interpretation (understanding three-dimensional transient data) can take months. This imbalance would seem to imply that, at least for the moment, the capabilities of supercomputers have outstripped our ability to use them effectively. Certainly, this imbalance must be overcome before multidimensional combustion models develop from their current state to viable, routinely used design aids.

Resolving this imbalance requires sophisticated graphics and the ability to transfer large amounts of data at high speeds over a distributed network of mainframes, workstations, and computer terminals.

Supercomputer needs

In December 1985, GMR's CRAY-1 S/2300 computer system was replaced by a CRAY X-MP/24 computer system with a 128-million-word SSD solid-state storage device. Engine combustion modeling is a significant part of the CRAY X-MP system's load. The decision to acquire a supercomputer at GMR was made only after extensive evaluations, including benchmark testing.[1] The obvious justification for such a machine is faster turnaround; in fact, the ability to approach previously intractable problems is more compelling.

The diversity of automotive applications[2,3] requires that future supercomputers used in the industry be general purpose machines able to run vendor-supplied and public domain software, with a stable Fortran compiler and operating system. In addition, I/O capacity must at least keep pace with future increases in CPU speed.

Optimizing computer code to take full advantage of a particular computer's architecture is a time-consuming and difficult task. Often, little optimization is done. Certainly, the capabilities of compilers to generate efficient machine code can be improved. Identifying the linear algebra in our codes, which is a substantial amount, would facilitate matching the code to the computer because optimized linear algebra routines are often available.

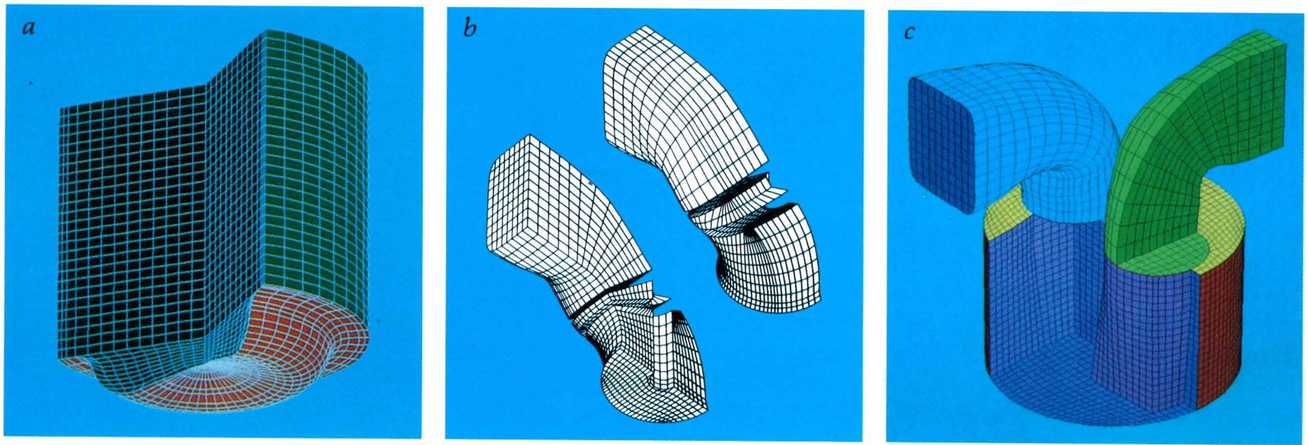


Figure 1. (a) A computational mesh for the KIVA code. This is the geometry used by Diwakar to calculate the scavenging efficiency of a large two-stroke diesel engine. (b) Preliminary mesh for an intake port. (c) Combined cylinder and port mesh created to help define code requirements and mesh generation methodology.

Preparation—grid generation

Grid, or mesh, generation is essentially the creation of a computer database that defines the subdivision of a three-dimensional object by small elements of volume, or computational cells. For example, Figure 1a shows a diesel engine mesh for the KIVA code.[4] Figure 1b illustrates a preliminary mesh for an intake port. It consists of three unconnected (as yet) blocks. For a combined cylinder and port calculation, we might propose a mesh like that in Figure 1c.

These grids illustrate the complexity of the engine geometry and lead to specifications of the capabilities of the computer code. These specifications include the requirements that each cell is a hexahedron and that the mesh should not be required to be orthogonal or simply connected (each cell need not have six neighbors). It may even be worthwhile to design a code in which the basic element is a tetrahedron. (These general meshes imply the need for indirect addressing and gather/scatter operations on a vector computer.)

The grid of Figure 1a was obtained using the optional mesh generator in KIVA; the other grid figures required specially-written software. This software tends to be highly interactive, and makes extensive use of linear interpolation and Laplacian smoothing to generate the interior mesh after the surface nodes are specified.

Graphics is an essential component of mesh generation, especially since the definition of a "good" mesh seems to be so subjective. The graphics in Figure 1 required the development of a hidden-line algorithm able to render selected parts of irregularly connected objects.

The data communications problem for grid generation lies in the fact that the relevant parts of the engine design, such as the piston and head, are likely to exist in some CAD system, probably on a different computer or workstation. Obviously, an electronic link between the CAD system and the mesh generation system is needed. Re-creating the database, for example, by building a part and then digitizing it, is not acceptable. In fact, we need the capability to send a refined design back to the CAD system.

Interpretation—understanding the results

Supercomputers enable us to generate enormous amounts of data in a relatively short time. (A single run might involve 100,000 mesh points, 1000 time steps, and a dozen variables of interest.) The challenge is more than to translate such immense amounts of data into something we can comprehend; it is then to synthesize our understanding into meaningful and succinct design guidelines. For example, about five years ago, Alex Alkidas and I combined his experiments and my calculations in a study of the flows in diesel engine prechambers.[5] We obtained "excellent qualitative" agreement between the experiments and computations, and were able to explain various experimental results of other authors whose experiments we had modeled, but not duplicated. Yet I was unable to answer the essential question: "Should we change the injection angle?" Until we can answer such questions, the real power of modeling will not have been realized.

Graphics is the key to understanding the results of three-dimensional transient calculations. This is demonstrated in the innovative display of Figure 2 generated by Rolf Reitz

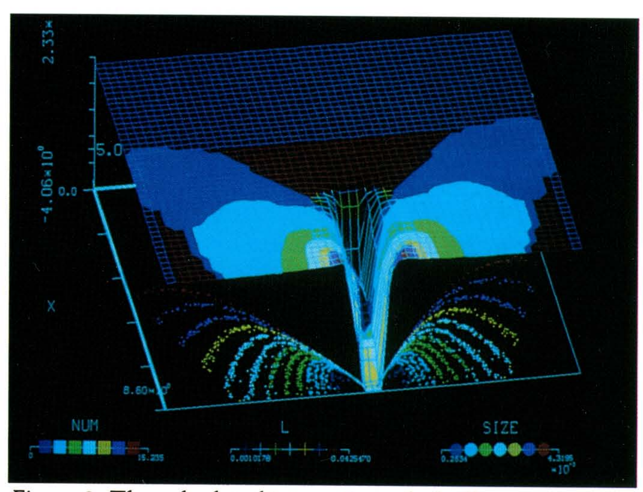


Figure 2. The calculated interaction of a hollow-cone spray with an oncoming air stream. The colored dots show drop positions and size. Surface height shows the axial gas velocity component, surface color shows drop number density, and grid color shows the turbulence length scale.

at GMR. The figure shows the results of a spray computation, modeling the injection of an axisymmetric hollow-cone spray into an oncoming airstream. The picture summarizes the following data: liquid drop size and position, axial gas velocity, turbulence length scale, and droplet number density.

In considering the graphics problem, one inescapably concludes that color animation is required. Real-time animation is not presently possible, but the technology exists to make animated sequences in a relatively short time. A programmable video animation controller can record on video tape single-frame copies of a graphics terminal display. The speed of the process depends on the complexity of the images and the data rate between the computer and the terminal. It is now possible to generate one minute of animation in one hour of real time and to use animation routinely in research.

Multidimensional combustion models

Engine combustion models solve a coupled system of partial differential equations for the conservation of mass (continuity), momentum (compressible Navier Stokes), energy, and chemical species. Additional equations account for turbulence, sprays, and combustion. The models are reasonably able to predict turbulent nonreacting flows. At least for cold flows, two-equation (k-ε) models of turbulence appear to be adequate.[6]

Studies have shown that the flow field in an engine cylinder is sensitive to the details of the flow through the intake valve; measurements of the intake valve flows have been made to provide boundary conditions for in-cylinder calculations. Note that coupling the port and cylinder calculations, as suggested by Figure 1c, is the only way to evaluate designs for which no hardware exists. Future codes must be able to handle moving valves and the high (sonic) velocities that the inflow can attain.

Spray models continue to be refined. The spray model in KIVA was formulated by Dukowicz[7] and extended by O'Rourke and Bracco.[8] A recent improvement by Reitz and Diwakar[9] involves the addition of drop breakup. Without drop breakup the initial drop size (a boundary condition) required to compute the correct spray penetration is unrealistically small. With drop breakup the correct penetration is obtained for a range of conditions, and the sensitivity of the results to the initial drop size is greatly reduced. Figure 3 shows the qualitative agreement obtained between model and experiment for a pulsed hollow-cone spray. The greatest need in the spray model is for the inclusion of wall-wetting, which occurs when liquid fuel impinges on the surfaces of the combustion chamber.

Despite various efforts, little progress has been made towards a comprehensive and predictive treatment of the relevant chemistry. Manifestations of the chemical processes in engines include

- □ Heat release
- □ Turbulent flame propagation
- □ HC, CO, and NO emissions
- □ Knock, or auto-ignition of the end gas

The simple stoichiometry often assumed for fuel-disappearance,

$$\text{Fuel} + O_2 \longrightarrow CO_2 + H_2O$$

gives an incorrect heat release, except for fuel-lean conditions. To obtain the correct post-flame temperature (and cylinder pressure) one must add CO and H_2.

Inclusion of post-flame kinetics for NO formation (the Zeldovich reactions) and CO_2/CO interconversion (the water gas reactions) adds NO, OH, O, and H to the species list. Models that include a full list of species have been used for some time. They assume that most of the species are in chemical equilibrium, and the equilibrium calculation can be made very efficient.[10] The Zeldovich reactions for NO formation are very temperature sensitive, but if the heat release rate is correctly computed, good predictions of NO emissions can be obtained.

A model for end-gas auto-ignition is available, and has been used with good results. However, the original formulation of the model must be changed so that it properly conserves mass and energy, and a "switch" is needed to turn off the knock reactions in the flame.[11,12]

The turbulent flame typically has been modeled as an enhanced laminar flame, in which the turbulence enters only through the transport coefficients. The fuel-disappearance reaction rate is specified as a nonelementary Arrhenius expression. This is inadequate. The model constants must be adjusted for each case, and additional ad hoc changes must be made to obtain the correct heat-release rate.

Supercomputers have made it possible to create detailed kinetic models of hydrocarbon oxidation. These models are able to predict laminar flame structure and, in some instances, auto-ignition. From this knowledge of the detailed kinetics, it should be possible to derive the global kinetic models to be used in multidimensional codes.

The propagation of a fully developed turbulent flame is certainly governed by the turbulence, and homogeneous-

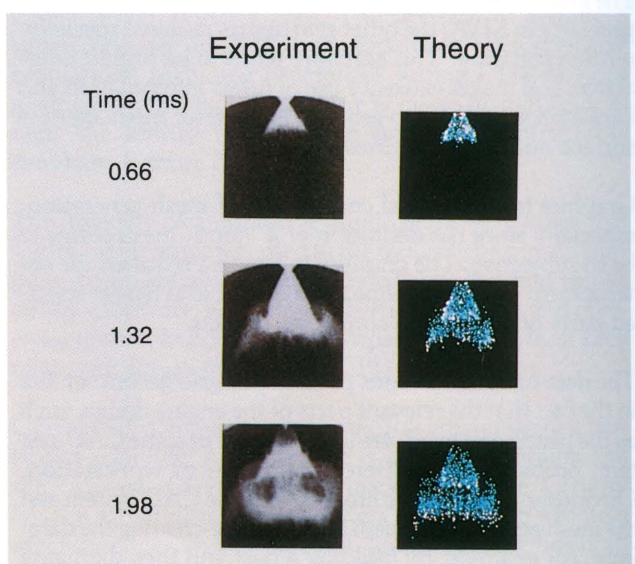


Figure 3. Comparison of experiment and computation for a hollow-cone spray.

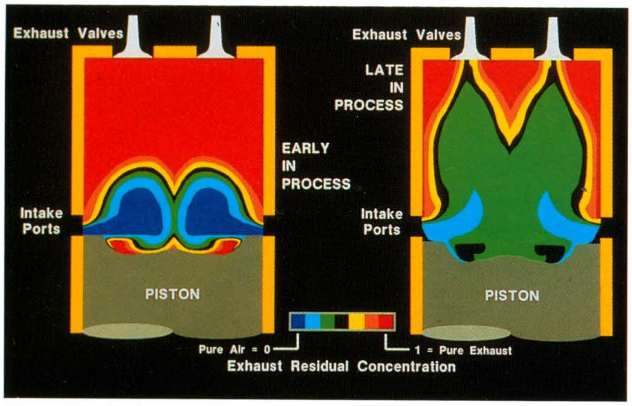


Figure 4. Exhaust gas concentration in a two-stroke diesel engine. The scavenging efficiency is affected by the swirl imparted to the incoming fresh air by vanes in the intake ports.

charge combustion rates have been reproduced by a model which considers both the laminar (chemical) and the turbulent mixing time scales.[13] Turbulence effects on combustion are receiving increased attention, but there is much to be done concerning the chemical processes in engines and how the chemistry is affected by turbulence.

Discussion

Diwakar[14] illustrates the usefulness of multidimensional models, even at their current stage of development. He examined the effects of intake swirl on the scavenging and trapping efficiencies of a large two-stroke diesel engine. Computed results are shown for two different times in Figure 4. (This is a three-dimensional calculation.) Uzkan and Hazelton[15] later expanded on these results, to point out that swirl level also affected the stratification of exhaust gas and fresh charge in the engine at the time of injection.

Studies such as these will never replace experiments, but they are certainly more cost-effective than cut-and-try development. The value of multidimensional models is their ability to screen hardware proposals, suggest experiments, and identify measurement locations.

Two major challenges remain. The first is resolving the imbalance of effort required when using these models. With sophisticated graphics and the ability to transfer large databases within a distributed computing environment, we will be able to construct pre- and postprocessors that will greatly facilitate results, preparation, and interpretation. The second challenge is more fundamental. A predictive model for turbulent combustion is urgently needed, a model that not only includes turbulent flame propagation, but also fits into the framework of a comprehensive treatment of the thermodynamics and chemistry of internal combustion engines. □

Acknowledgments

 This article is adapted from the author's presentation at the International Conference on Supercomputer Applications in the Automotive Industry, October 1986, Zurich, Switzerland. The author thanks Di Diwakar, Myron Ginsberg, Roger Krieger, and Rolf Reitz for useful suggestions concerning this article.

About the author

Keith Meintjes is a staff research engineer in the fluid mechanics department at the General Motors Research Laboratories. He received his B.S. and M.S. degrees in mechanical engineering from the University of the Witwatersrand in Johannesburg, South Africa, and received his Ph. D. degree in 1980 from Princeton University. Meintjes researches numerical simulation of flow and combustion in internal combustion engines, with a particular interest in realistic models of the complex chemical reactions that occur in engines.

References

1. Ginsberg, M., "An Overview of the Computational Environments Offered by Current and Future Supercomputers," SAE Paper 850472, 1985. See [2], pp. 1-25.

2. Ginsberg, M. (Ed.), "Supercomputers in the Automotive Industry," SAE SP - 624, SAE, Warrendale, Pennsylvania, 1985.

3. Marino, C. (Ed.), *Supercomputer Applications in Automotive Research and Engineering Development*, Computational Mechanics Publications, Southampton, Great Britain, 1986.

4. Amsden, A. A., J. D. Ramshaw, P. J. O'Rourke, and J. K. Dukowicz, "KIVA: A Computer Program for Two- and Three-Dimensional Fluid Flows with Chemical Reactions and Fuel Sprays," Los Alamos National Laboratory Report LA-10245-MS, 1985.

5. Meintjes, K. and A. C. Alkidas, "An Experimental and Computational Investigation of the Flow in Diesel Engine Prechambers," SAE Paper 820275, *SAE Transactions*, Vol. 91, pp. 1148-1165.

6. El Tahry, S. H., "A Comparison of Three Turbulence Models in Engine-Like Geometries," *Proceedings of International Symposium on Diagnostics and Modeling of Combustion in Reciprocating Engines*, Japan Society of Mechanical Engineers, Tokyo, 1985.

7. Dukowicz, J. K., "A Particle-Fluid Numerical Method for Fuel Sprays," *Journal of Computational Physics*, Vol. 35, 1980, pp. 229-253.

8. O'Rourke, P. J. and F. V. Bracco, "Modeling of Drop Interactions in Thick Fuel Sprays," *Stratified Charge Automotive Engines*, Conference Publications 1980-9, I. Mech. E., London, 1980.

9. Reitz, R.D. and R. Diwakar, "Effect of Drop Breakup on Fuel Sprays," SAE Paper 860469, 1986.

10 Meintjes, K. and A. P. Morgan, "A Methodology for Solving Chemical Equilibrium Systems," GM Research Publication GMR - 4971, 1985.

11. Natarajan, B. and F. V. Bracco, "On Multidimensional Modeling of Auto-ignition in Spark-ignition Engines," *Combustion and Flame*, Vol. 57, 1984, pp. 179-197.

12 Schapertons, H. and W. Lee, "Multidimensional Modeling of Knocking Combustion in SI Engines," SAE Paper 850502, 1985.

13. Abraham, J., F. V. Bracco, and R. D. Reitz, "Comparisons of Computed and Measured Premixed Charge Engine Combustion," *Combustion and Flame*, Vol. 60, 1985, pp. 309-322.

14. Diwakar, R. "Multidimensional Modeling of the Gas Exchange Processes in a Uniflow-Scavenged Two-Stroke Diesel Engine," *International Symposium on Flows in Internal Combustion Engines - III*, ed. T. Uzkan et al., ASME—FED Vol. 28, 1985, pp. 125-134.

15. Uzkan, T. and J. R. Hazelton, "The Influence of Swirl on the Fresh Charge Stratification in an IC Engine Combustion Chamber," SAE Paper 860466, 1986.

COMPUTATIONAL REQUIREMENTS AND TRENDS IN DESIGN OPTIMIZATION

Garret N. Vanderplaats, University of California, Santa Barbara, California

The computational power available from today's supercomputers is making large-scale engineering analysis commonplace. However, the next step needed to automate design fully — design optimization — is only now beginning to be realized. Design optimization technology has matured to the point that it can be considered as a practical design tool. Because automation of the design process involves the iterative analysis of candidate designs, it is a computationally intensive undertaking that is made possible only through the availability of supercomputers.

In 1960, Schmit introduced the concept of *structural synthesis,* in which finite element structural analysis was combined with mathematical programming methods to solve the general nonlinear design task. He used a simple three-bar planar truss to demonstrate the concepts, and showed that traditional design methods did not yield the optimum design (in this case minimum weight). The basic idea was to use available structural analysis methods to calculate the structural response to the applied loads, given the member sizes and basic structural layout. Numerical optimization was then used repeatedly to call this analysis with different proposed member sizes. In his 1981 paper reviewing this technology, Schmit[1] stated that this first application required nearly one-half hour of CPU time on the IBM 653 computer and noted that, considering this, "only a congenial optimist could have been so enthusiastic about future prospects." Using today's microcomputers, this same problem is solved in under two seconds, and using supercomputers, the computational cost is near zero. This design efficiency has come about both through the increased speed of computers and the maturing of the basic technology itself. In light of this, one might ask why more computational speed is needed. The answer, of course, is that computational requirements have consistently exceeded available hardware and that today we wish to solve design problems not dreamed of in 1960. In our quest to use the computer to make the actual design decisions, we find that, typically, one to two orders of magnitude more computation is required for design than for a single analysis.

To understand the needs here, it is important to distinguish between analysis and design. Here we define analysis as the process of determining the structural, aerodynamic, thermal, or other response of a part or system to its environment. It is assumed that the part or system is defined so that we only analyze it to determine how good it is. Design, on the other hand, is the process of actually defining the system to perform a specified function or functions. Thus, analysis is a necessary subtask of design because analysis is the process of judging the quality of a proposed design. The process of design may be quite intuitive and heuristic. For example, the basic shape of a structure may be defined based on experience or aesthetic requirements. On the other hand, once the basic system is defined, the actual sizing of structural members or aerodynamic shapes can often be treated as a repetitive or algorithmic process. When design can be defined in this way, optimization becomes a viable tool. Also, it might be noted that this is the relatively tedious and uninteresting part of the design task. Thus, if we can provide the tools to relegate this to the computer, it frees us to dwell on the more interesting and creative aspects of design.

Basic numerical optimization concepts

The basic design task, from a numerical optimization viewpoint, is to find the set of variables, $\underline{X}$, that will

$$\text{minimize} \quad F(\underline{X}) \qquad (1)$$

subject to

$$g_j(\underline{X}) \leq 0 \qquad j=1,m \qquad (2)$$

$$X_i^l \leq X_i \leq X_i^u \qquad i=1,n \qquad (3)$$

Here, $F(\underline{X})$ is referred to as the *objective function*, and may be weight, cost, or any other function of $\underline{X}$. Typically, from 10 to 100 design variables are contained in $\underline{X}$, although research is continuing to expand this number. If one wishes to maximize $F(\underline{X})$, this is easily accomplished by simply minimizing the negative of $F(\underline{X})$. The inequality conditions defined by Equation 2 are referred to as *constraints*, and there may be hundreds or even thousands of these. For example, if the stress at some point in a structure is not to exceed a prescribed allowable, this would be written, in normalized form, as:

$$g(\underline{X}) = S_{ij}/S_A - 1 \leq 0 \quad (4)$$

where S_{ij} is the stress at point i under load condition j, and S_A is the allowable stress. Recognizing that several stress points may exist for each of hundreds of finite elements in the structural model, and that numerous load conditions may be considered, it is clear that the total number of such constraints can be quite large. The bounds on the design variables given by Equation 3 are referred to as side constraints because they limit the region of search for the optimum. Reasonable side constraints will improve the efficiency of our search for the optimum. More importantly, these can be used to limit the search to designs that are physically meaningful. For example, the thickness of a structural element should never be allowed to be negative, because such a design would not be meaningful and the analysis of such a structure may not be possible.

In addition to stresses, constraints may be imposed on displacements, frequencies, buckling loads, dynamic, and even aeroelastic response. Thus, it is clear that the problem statement given here is quite general. The optimum design is defined as the set of design variables, $\underline{X}$, that minimizes $F(\underline{X})$ subject to the constraints. Both the objective and the constraint functions can be highly nonlinear implicit functions of $\underline{X}$, leading to a computationally intensive task to search for the optimum.

Searching for the optimum design is usually based on the following iterative equation:

$$\underline{X}^q = \underline{X}^{q-1} + a * \underline{S}^q \quad (5)$$

where q is the iteration number, $\underline{S}$ is a vector search "direction," and a is called the step length. To visualize this process, imagine standing on a hillside, blindfolded. The "design task" is to find the highest (maximum) point on the hill while staying inside a series of fences. Thus, the elevation on the hill is the objective function and the fences define the constraints on the problem. The design variables are the coordinates of our position on the hill. We can imagine taking a small step in the North-South direction and another in the East-West direction. From this, we can sense the steepest ascent direction up the hill. By doing this, we have actually calculated the gradient of the objective function by finite difference steps. Assuming we are not up against a fence, we can call this direction the search direction, $\underline{S}$, and move in this direction. We continue to move in direction $\underline{S}$ until we reach a crown on the hill or encounter a fence. In engineering problems, the latter is the common case. This process of moving in direction $\underline{S}$ is called the one-dimensional search, and the parameter a in Equation 5 is called the step length, physically the number of steps we take in this direction. In practice, we will try several step lengths and interpolate for the maximum objective, subject to the constraints.

Once we have moved as far as possible in this direction, we increase the iteration counter, q, by one and repeat the process. Now, assuming we are against a fence, we calculate the gradient of both the objective function and this constraint. We then mathematically determine a new search direction that will move us up the hill without going outside the fence. This will no longer be the steepest ascent direction since the fence prevents us from moving in that direction.

Note that every time we evaluate the elevation on the hill and the distance from the fences, we are doing what we usually call analysis. If the analysis is expensive, the optimization task will also be expensive, because numerous separate analyses will be needed to reach the optimum.

The method just described for finding the optimum is referred to as a *feasible direction method*. A multitude of other methods are available for solving the general nonlinear optimization problem and each has unique features.

In general applications, we may use this direct approach where finite difference gradients are calculated and the actual functions are evaluated at each step. This is usually the case in such problems as aerodynamic shape optimization. On the other hand, if the design problem is of a linear elastic structure that we model by the finite element method, the state of the art is much more advanced. First, it is possible as part of the analysis to calculate the gradients of the objective and a wide variety of constraints. Secondly, we now have sufficient understanding of this design task to create a high-quality approximation of the original problem. We then optimize, based on this explicit approximation, without the need to perform a detailed finite element analysis. The proposed optimum is then analyzed in detail and the process is repeated until convergence to the optimum is achieved. Using this approach, it is usually possible to design practical structures using as few as five detailed finite element analyses. This design efficiency makes optimization possible for large structures where the finite element model may contain thousands of elements and tens of thousands of displacement degrees of freedom. Thus, while only a few detailed analyses are required, the computational effort still involves hours of supercomputer time.

Structural optimization

Structural optimization techniques have been under development for more than 25 years, and are probably the best understood of the engineering applications of this technology. The most common objective function is the weight of the structure. A wide variety of constraint functions can be considered, including stress, strain, displacement, local and system buckling, frequency response, time-dependent dynamic response, and aeroelastic limits. Almost all modern applications are based on the finite element method of analysis. A unique feature of structural optimization is that, in most cases, gradient information can be calculated with relatively minor modifications to the analysis program. The availability of gradient information (the sensitivity of the structural response such as stress, displacement or frequency, to changes in the design variables) is of major importance because this is used to guide the optimization process. It is noteworthy that, for the first few years of research in this field, gradient information was calculated by finite difference methods. This was because the structural response is an implicit function

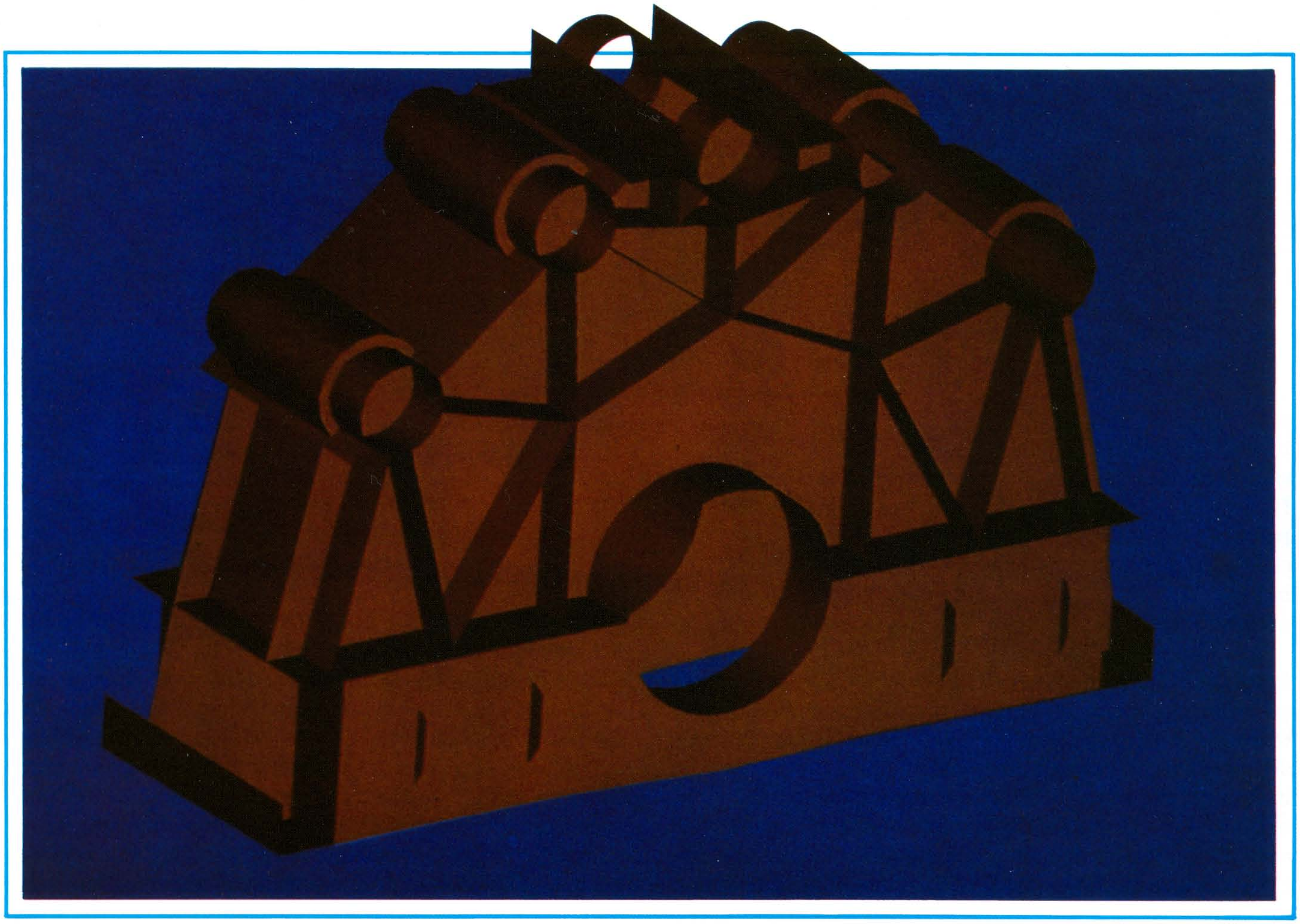


Figure 1. Model of large gear housing for structural optimization study.

of the design variables and it was therefore assumed that gradient information is not easily obtained. However, to understand that it is, in fact, relatively straightforward, consider the standard form of the finite element static analysis:

$$K\underline{u} = \underline{P} \qquad (6)$$

where *K* is the master stiffness matrix and the summation of elemental stiffness matrices. Vector $\underline{u}$ is the displacement vector and $\underline{P}$ is the load vector (multiple vectors $\underline{u}$ and $\underline{P}$ are allowed). Now we need only to implicitly differentiate Equation 6 and rearrange to obtain the equation for the gradients of displacements with respect to some independent design variable, X_i:

$$\frac{\partial \underline{u}}{\partial X_i} = K^{-1}\left[\frac{\partial \underline{P}}{\partial X_i} - \frac{\partial K}{\partial X_i}\underline{u}\right] \qquad (7)$$

Here the derivative of the master stiffness matrix is just the sum of derivatives of element stiffness matrices and is usually very sparse. The element derivatives are usually calculated easily or may even be calculated by finite difference (at this level, finite difference calculations are quite inexpensive; this is often referred to as a *semianalytical* method). Assuming an analysis has already been performed, K^{-1} is available in the form of a decomposed matrix, so solution of Equation 7 involves only some additional forward and backward substitutions. Once the gradient of displacements is available, gradients of stresses are calculated directly using the stress-displacement relationships.

The net effect is that gradient information for many constraints can be evaluated for less than the cost of a single additional analysis. Also, gradients of numerous other, more complex constraints can almost as easily be calculated.[2]

Once gradient formation is calculated, it can be manipulated in various ways so that it can be used in the optimization phase of design. The simplest approach would be to use the gradient information directly as a linear approximation to the general problem. This could be solved, with reasonable limits on the changes in the design variables, using well-established linear programming methods. However, we can actually be more sophisticated in our approximations by noting that stress and displacement responses are approximately proportional to the reciprocal of member sizing variables such as cross-sectional areas and thicknesses. Thus, it is now common to transform the approximation into reciprocal or mixed space to create a very high-quality approximation. The overall motivation here is to create such a good approximation to the original problem that it may be used for optimization, rather than repeatedly calling the large-scale finite element analysis. Once the approximate optimum has been found, the analysis is repeated to create a new approximation, iterating as necessary until the optimum design has been achieved. The effect of this "smart" approach to structural optimization is that if the finite element analysis is called directly from the optimization program, dozens or even hundreds of analyses will be needed. By creating high-quality approximations based on our knowledge of the mathematics, however, we are able to design practical

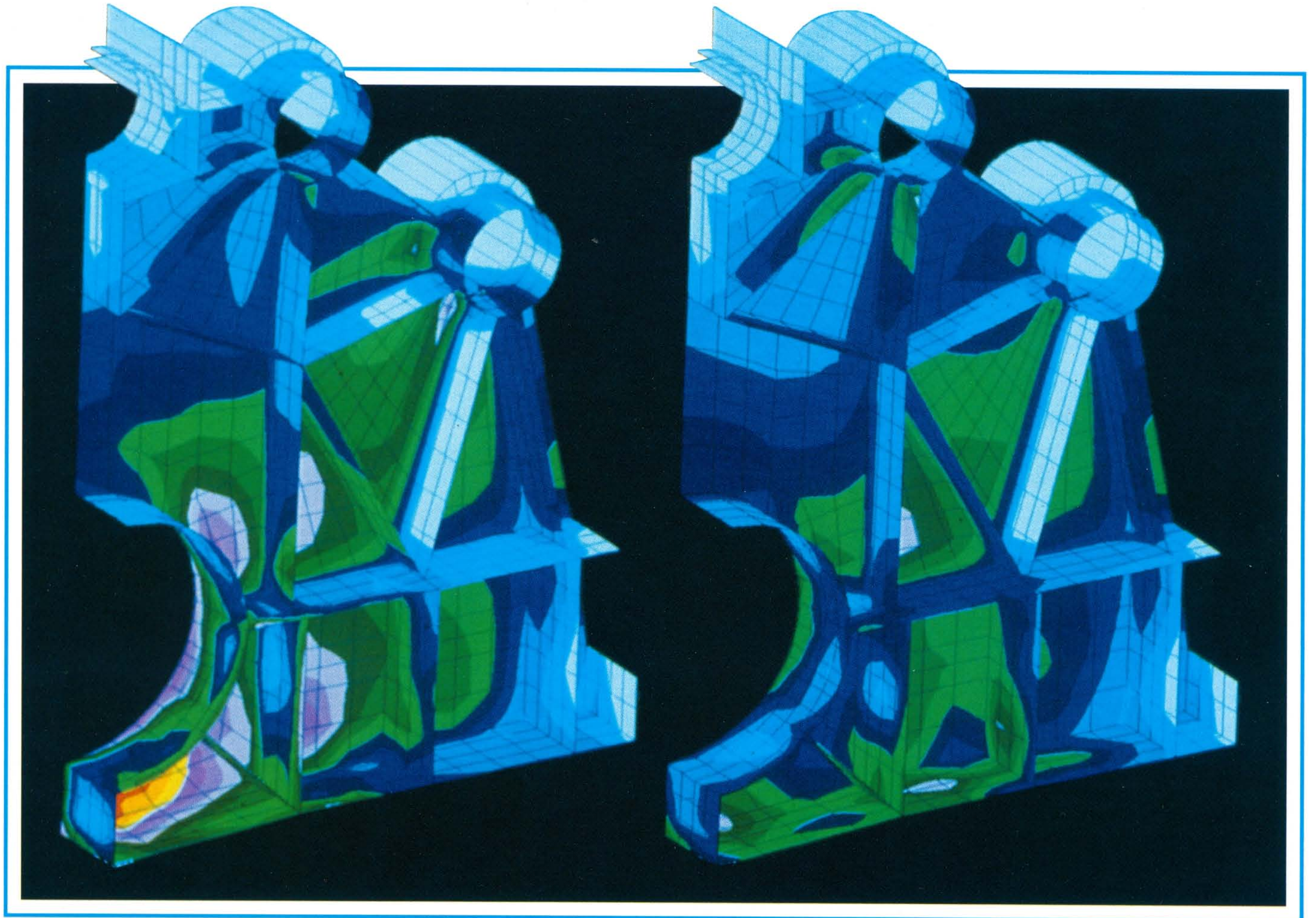


Figure 2. Finite element models of gear housing showing stress distribution for original design (left) and optimal design (right). Color coding runs from red (highest stress) through yellow, violet, and green, to blue (lowest stress).

structures using from five to ten detailed finite element analyses.

To demonstrate the present state of the art in structural optimization, two examples are offered here: a large marine gear housing and an automotive engine connecting rod.

Figure 1 shows a large gear housing for marine applications.[3] The structure has two planes of symmetry, although the applied loads are not symmetric for all cases. An MSC/NASTRAN finite element model of one-fourth of the structure was created. The finite element model consisted of 1623 elements, 7239 displacement degrees of freedom, and 6 load cases. The design program was created by coupling MSC/NASTRAN version 63 with the CONMIN optimization program. Figure 2 shows the stress distribution (with the bottom cover removed) for the dominant load case for the initial and final designs.

Thirty independent sizing variables of plate thicknesses and stiffener cross-sectional areas were considered. There were 5620 nonlinear inequality constraints, including stress limits under each loading condition as well as rather tight deformation constraints. These deformation constraints included bearing out-of-roundness, transverse rotation, and center-to-center displacements. Some of the constraints were initially violated by nearly 100 percent. The initial design was scaled up to provide a near-feasible starting point. (This should not normally be required, because the optimization can be used to obtain a feasible design; however, this was a new design code and a rather large problem, and so the conservative approach was taken.)

The design iteration history is shown in Figure 3. Here one iteration consists of a complete finite element analysis, gradient computations for approximately 150 of the most critical constraints, and creation and solution of the approximate optimization problem. The entire optimization process required six analyses and five sensitivity calculations. Each iteration required approximately 600 seconds on a CRAY-1/S supercomputer and the total design time required just under one hour. This example is not

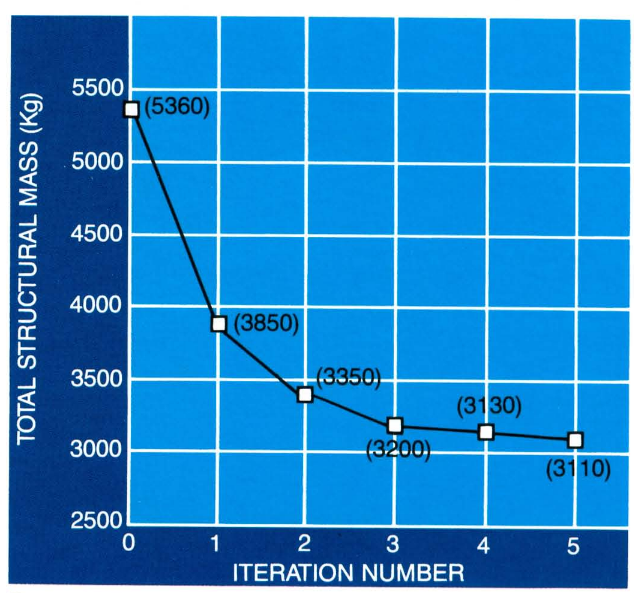


Figure 3. Design iteration history for gear housing optimization study.

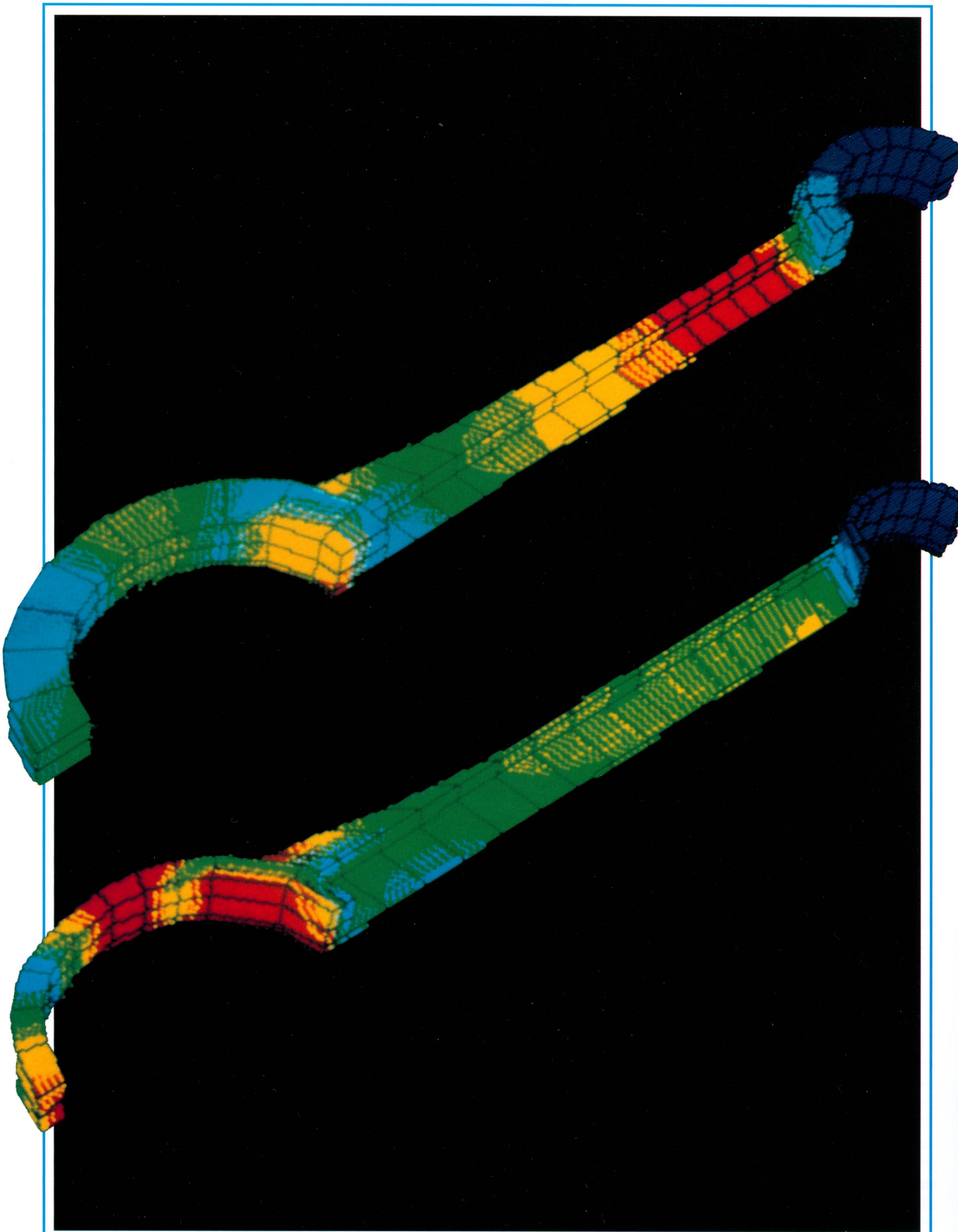


Figure 4. Finite element models of connecting rod showing stress distribution for original design (top) and optimal design (bottom). Color coding runs from red (highest stress) through yellow and green to blue (lowest stress).

considered to be near the largest structural optimization task that can be solved with today's technology, so it is clear that immense computational power will be needed to use structural optimization on a routine basis. The use of optimization techniques is clearly justified, however, particularly when it is noted that traditional design methods had failed to solve this problem.

Figure 4 shows the second example: a connecting rod for an automotive engine, provided courtesy of General Motors Research Laboratories, Warren, Michigan. The finite element model consisted of 105 solid elements, 928 nodal points, and 2126 degrees of freedom (for a one-fourth scale model, because there are two planes of symmetry). Eight design variables were used to define the shape of the structure.[4] This design problem was solved using the MSC/NASTRAN finite element program coupled with the ADS optimization program.

This design task is relatively difficult for two reasons. First, determining the external shape of the structure was required, and the methodology is not as well-developed for this as for the earlier sizing example. Second, sensitivity calculations with respect to shape variables are not presently contained in the finite element program (this is presently being added) and had to be externally calculated using what is called the *material derivative method*. The iteration history is shown in Figure 5. While the design process was allowed to continue for 20 iterations, it is noteworthy that a practical solution was obtained in under 10 iterations.

Remaining challenges

The purpose here has been to identify the power of numerical optimization techniques as a design tool and to demonstrate that, with today's supercomputers, fully automated design can be a reality. However, as our ability to solve larger design problems improves, so does the size and complexity of those problems.

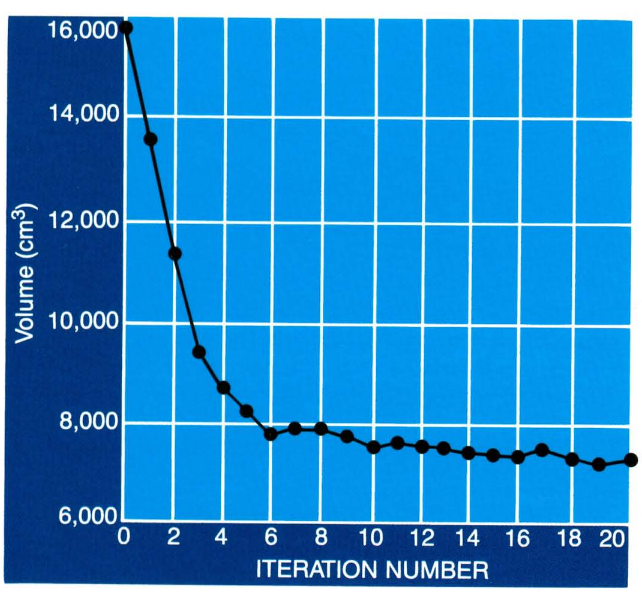


Figure 5. Design iteration history for connecting rod optimization study.

This discussion focused on only one fairly familiar application. Many other problems are amenable to solution by numerical optimization. For individual discipline design, other possibilities include aerodynamics, mechanical components, control augmented structures, combustion process optimization, and electronic cooling devices, to mention only a few. However, this is only the beginning. Few systems are limited to only one discipline. Structures, aerodynamics, controls, and human factors must all work together as a system, and the optimum system is seldom the sum of optimum parts. Therefore, the ultimate goal must be that of complete system synthesis. Research has been underway for many years for system synthesis using optimization.[5] Until now, however, this has been limited to conceptual design or simple problems.

Numerical optimization techniques have been applied to engineering design for some years, but only now are being widely recognized as practical design tools. That recognition stems, in part, from the computational power of today's supercomputers. As this technology becomes more widely used, computational needs will continue to grow. As we look forward to tomorrow's supercomputers, we can only be sure that, as they become available, our needs will continue to exceed their capacity. □

Acknowledgment

This article is adapted from the author's presentation at the International Conference on Supercomputer Applications in the Automotive Industry, October 1986, Zurich, Switzerland.

About the author

Garret N. Vanderplaats is a professor of mechanical engineering at the University of California, Santa Barbara, specializing in design optimization and structural mechanics. He received an M.S. degree in civil engineering from Arizona State University and was awarded a Ph.D. in structural mechanics from Case Western Reserve University. He previously worked at NASA-Ames Research Center as a research scientist and was associate professor at the Naval Postgraduate School in Monterey, California. Vanderplaats also authored a general-purpose optimization program used in the aerospace and automotive industries.

References

1. Schmit, L. A., Jr., "Structural Synthesis—Its Genesis and Development," *AIAA Journal,* Vol. 19, No. 10., 1981.

2. Haftka, R. T. and M. P. Kamat, *Elements of Structural Optimization,* Martinus Nishoff Publishers, Boston, 1985.

3. Vanderplaats, G. N., H. Miura, and M. Chargin, "Large-Scale Structural Synthesis," *Finite Elements in Analysis and Design,* Vol. 1, No. 2, 1985, pp. 117-130.

4. Botkin, M. E., R. J. Yang, and J. A. Bennett, "Shape Optimization of Three-Dimensional Stamped and Solid Automotive Components," General Motors Research Laboratories Report GMR-5168, in *Proceedings of the 1985 GMR Symposium on the Optimum Shape,* 1985.

5. Vanderplaats, G. N., *Numerical Optimization Techniques for Engineering Design - With Applications,* McGraw-Hill, 1984.

CFT

NEW FEATURES AUGMENT CFT POWER

Richard Hendrickson, Cray Research, Inc.

The 1.14 and 1.15 releases of the Cray Fortran compiler CFT contain many new features that make it easier to write programs and to get optimum performance from CRAY X-MP and CRAY-1 computer systems. This article discusses enhancements in three general areas:

- □ Productivity aids. New aids make it easier to write and debug programs.
- □ Vectorization and code generation. CFT vectorizes more types of loops.
- □ Compiler invocation options. Several options are provided to allow "fine-tuning" of the optimizer.

Productivity aids

The most important productivity aid is the CFT 1.15 LOOPMARK feature. It is enabled under COS via

```
CFT, LOOPMARK ...
```

or, under UNICOS via

```
cft -v msgs ...
```

The LOOPMARK facility produces an annotated source listing, brackets DO loops, and marks vectorized DO loops. All of the examples in this article were compiled with the LOOPMARK option.

Coupled with LOOPMARK is an enhanced cross-reference table for DO loops and a set of diagnostic messages for loops that are not vectorized. As Figure 1 shows, these messages usually refer to explicit variable names and source line numbers.

An important new debugging aid is the INDEF invocation option. The INDEF option is enabled under COS by specifying

```
CFT, INDEF, ALLOC = STACK ...
```

or, under UNICOS,

```
cft -e I -a stack
```

When this option is enabled, CFT generates code that fills the stack with out-of-range floating point values. To understand how this aid can be useful, consider the RANDOM function in Figure 2. It uses a variable named SEED to produce a sequence of pseudo-random numbers. Although the CFT manual explicitly says this routine is illegal, the routine uses a reasonably common coding style and works as expected. This is because CFT, like many other compilers, normally allocates static storage for each variable in a routine. However, if the routine is compiled in stack mode either to save memory or to use multitasking on a CRAY X-MP system, SEED will not be assigned

```
26  :--            DO 40 I = 1,N
27  :              A(I) = X * B(I)
28  :--        40  X = C(I) + D(I)
PRNAME TEST NOVECTOR — 'X' REFERENCED AT S.N. 27 BEFORE BEING DEFINED AT S.N. 28
```

Figure 1. LOOPMARK listing for a nonvectorizable loop.

```
      FUNCTION RANDOM (RESET)
      IF (RESET .NE. 0.0) SEED = RESET
      SEED = MOD (SEED **3 , 10.0)
      RANDOM = SEED
      END
```

Figure 2. Usage of an unsaved variable.

static storage. But it will share stack storage with variables in other routines and it will not remember its previous value.

The solution for codes such as RANDOM is to insert a

```
SAVE SEED
```

statement in the function to force static storage. Problems like this can be found by reading the source code and cross-reference listing, but in a large code this is tedious. With the INDEF option specified, however, use of an unsaved variable generally causes a floating point interrupt rather than unexpected results.

Vectorization and code generation

In addition to new productivity aids that make it easier to write and debug codes, Cray Research has ongoing efforts to make codes running on Cray systems execute even faster. The results of some of these efforts take place behind the scenes through improvements in the generated code, rather than through visible features such as enhanced vectorization. A modification to CFT 1.14 recognizes more chaining opportunities on CRAY-1 machines. Modifications in CFT 1.15 generate more efficient code for vector reductions and use registers more efficiently when evaluating scalar temporaries. Consequently, the short loop in Figure 6 runs about 13 percent faster using CFT 1.15 than it did using CFT 1.14.

The most obvious performance enhancement is, of course, enhanced vectorization. Figures 3 and 4 show simple examples of some of the loops CFT 1.14 and CFT 1.15 will now vectorize. (These figures show the general form of vector constructs. Obviously, they are not intended as a style guide for production programs.)

Beginning with the 1.14 release, CFT vectorizes block IF constructs using gather/scatter instructions. These can be either the block IF — THEN — ELSE — ENDIF form, as in Figure 3, or can be formed with an arithmetic or logical IF and GOTOs. It is the form that counts, not the particular syntax. While there can be more than one IF block in a vectorized loop, IFs that are nested will not vectorize. (The IF at line 12 is a special case controlled by the PARTIALIFCON option, which is discussed later in this article.)

Line 17 in Figure 3 is an example of gather and scatter vectorization. Vector loops may contain either irregular subscripts, such as I/17, or array references, such as INDEX(I). The former often occur when doing table-lookup or Monte Carlo operations, the latter when processing arrays with

```
 5  :--------          DO 200 K = 1,N
 6  :                  G(K) = 0
 7  :   :-----         DO 200 J = 1,M
 8  :   :   V--        DO 100 I = 1,L
 9  :   :   V          BIG = MAX (BIG,A(I))
10  :   :   V          ALL = ALL .AND. LOG(I)
11  :   :   V          IF (B(I) .GT. C(I))  THEN
12  :   :   V            IF (B(I) .LT. 0.0) D(I) = 1.0
13  :   :   V            B(I) = 0
14  :   :   V          ELSE
15  :   :   V            D(I) = 2.0
16  :   :   V          ENDIF
17  :   :   V--  100   F(INDEX(I)) = H(I/17)
18  :--:-----    200   G(J) = G(J) + K
```

Figure 3. Examples of constructs that CFT 1.15 will vectorize.

```
     Vc-         DO 20 I = 51,71
     Vc          C(I) = 0.0
     Vc          IF (B(I) .LT. 0) GO TO 30
     Vc          IF (D(I) .GT. 0) GO TO 35
     Vc-    20   A(I) = 2. * A(I+N)
```

Figure 4. A search loop with ambiguous subscripts.

a sorted key array. On machines that do not have gather/scatter hardware instructions, CFT may simulate the hardware operation by compiling an inner scalar loop within the vector loop. However, since the simulation is obviously much slower than the hardware operation, CFT estimates the loop complexity and will compile the entire loop in scalar mode if it appears that the gather/scatter simulation will produce an overall performance degradation.

Lines 9 and 10 of Figure 3 are examples of new vector reductions. In addition to vectorizing sum and product reductions, CFT 1.15 now vectorizes reductions using the MAX or MIN functions and logical reductions using the .AND. or .OR. operators.

Figure 4 is an example of another type of vectorizable IF, a search loop. A search loop is one in which the exits are taken when particular conditions are met. Because of the complexity of the decisions, a search loop that also contains a block IF will not be vectorized.

Figure 4 also illustrates an important compilation technique as well: runtime resolution of ambiguous subscripts. Line 23 has a potentially unvectorizable dependency. If N is between -20 and 0, CFT must execute the loop in scalar mode. If N is positive or less than -20, it can execute the loop in vector mode. Since there is no way for the compiler to know the values of variables, CFT 1.13 would generate guaranteed safe scalar code. But, the CFT 1.14 compiler uses a clever trick. It compiles the loop twice, once in scalar mode and once in vector mode and generates an IF — THEN — ELSE — ENDIF to select the correct version at runtime. This results in guaranteed safe vectorization of ambiguous subscripts, without the need to insert compiler directives. Of course, CFT still issues diagnostic messages describing the ambiguities, and compiler directives can be used to eliminate the small overhead.

Invocation options

The last set of features, new invocation options, allows interaction between the programmer and the CFT optimizer. CFT is basically a two-pass compiler. The first pass reads the source code, checks it for errors, and translates it into an internal format. The second pass breaks the internal format into blocks and compiles optimized, vectorized code for each block. There is no "middle pass" that gathers information about overall usage of variables and data flow. To make up for this missing pass, CFT supports some options asserting that certain data flow patterns will never occur. It also supports options that permit or restrict some aggressive optimizations. The most important of these options under COS are

```
CFT, OPT = BTREG:KILLTEMP:FASTMD:
PARTIALIFCON...
```

or, under UNICOS,

```
cft -o btreg,killtemp,fastmd,partialifcon...
```

Figure 5 is a comparison of execution times for 11 "typical" jobs. These are real jobs, not just kernels; the biggest one is about 16,000 lines of source code. The graph plots the ratio of run times of the jobs compiled with the previous four options selected, to run times obtained with the options off (default). On average there is about a ten percent performance improvement. Unfortunately, one job has slowed down by 13 percent; that is partly why these optimizations are optional.

Of the user controlled optimizations, the most important is

```
OPT = BTREG...
```

or, under UNICOS,

```
-o btreg
```

This option allows CFT to assign variables in a subroutine to T registers rather than to memory. This has an obvious and a subtle advantage. First, T registers are fast. For example, on a CRAY X-MP system a value can be transferred from a T register in one clock period, rather than the 17 clock periods required by a memory reference. Perhaps more importantly, T register transfers do not interfere with vector memory operations. Consider the following loop:

```
      DO 10 J = 1, 64
10    A(J)  = 0
```

The rules of Fortran require that J have a value of 65 after the loop is finished. If J is stored into memory, the hardware will delay the scalar store until the vector store has completed — in this case, a delay of about 64 clock periods. With J in a T register, the scalar store and the vector store can be overlapped.

Because of the way it was implemented, BTREG cannot be enabled by default. Variables assigned to T registers behave much like variables assigned to a stack; that is, their values are lost when a subroutine returns. This means that function RANDOM in Figure 2 would not work as expected if compiled with OPT = BTREG. The value of SEED would be lost after each return. However, if a SAVE SEED statement were added to the function, then SEED would

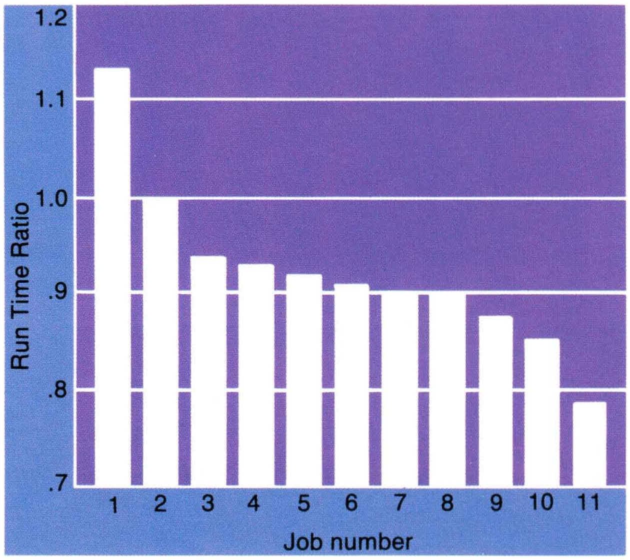


Figure 5. Job execution time comparisons.

```
3        V--          DO 50 I = 1,1000
4        V            T = A(I) + B(I)
5        V--     50   SUM = SUM + T
```

Figure 6. A loop with summations and scalar temporaries.

not be assigned to a T register and the function would work as expected.

Because BTREG is such a significant optimization, all important codes should be tried with OPT = BTREG. If the job fails to run correctly, the problem is invariably due to unsaved variables. The variables should be found and saved. The INDEF option, discussed earlier, can be a help in finding them.

KILLTEMP is another option that affects variable storage. It is an assertion that values computed for scalar temporaries, variables like T in Figure 6, will not be used after the DO loop is finished. In effect, the code that CFT then generates will not change the value of T in memory. This has obvious advantages — fewer scalar stores and no delay while waiting for a vector store to finish. The more important advantage is hidden — it allows CFT to generate better vector code in the DO loop. The transfer of the last vector value (A(1000) + B(1000), in this example) to a scalar register is difficult for CFT to schedule efficiently. The transfer will not chain and often causes CFT to run out of vector registers in a big loop. The BTREG and KILLTEMP options would both be on by default if CFT did data flow analysis between optimization blocks.

FASTMD is an assertion that integer values will be less than 2^{46}. If this is true, then CFT can generate faster code for integer multiplies and divides. This is usually useful only in codes that operate on integer variables that are not used as array subscripts. Subscript operations are so heavily optimized that FASTMD almost never affects them.

The final significant option is PARTIALIFCON. This option controls the interpretation of one-line logical IF statements, such as line 12 in Figure 3. The option allows CFT to "replace" some IF statements with "equivalent" simple assignment statements. A statement like

```
IF(A(I).GE.0.0) B(I) = C(I)
```

becomes

```
B(I) = CVMGP(C(I), B(I), A(I))
```

This new statement is then vectorized using the vector-merge instruction, rather than gather/scatter instructions. On machines without gather/scatter hardware, vector speed is obtained rather than scalar speed. Even on a CRAY X-MP system, a speed advantage is realized if the arrays B or C are used elsewhere in the DO loop. While this option normally speeds up program execution, it can slow it down or even cause it to abort. The problems come from the nature of the vector merge operation. It requires that both the "true" and "false" values be "computed" before the desired ones are selected. In this simple example that is not a problem; the compiler merely generates a load for the arrays B and C. But suppose the example were more complicated — for example, in a situation where

```
IF(A(I).GE.0) B(I) = ASIN(C(I))
```

were evaluated as

```
B(I) = CVMGP(ASIN(C(I)), B(I), A(I))
```

Suppose that whenever A(I) is negative then C(I) is greater than 1.0. The ASIN function will abort when it tries to evaluate an unneeded result. Or, suppose that A(I) is almost never positive. Then many unneeded ASINs are computed and even though it is done at vector speed, performance may suffer.

IF statements are probably the most significant area for interaction between CFT and the programmer. CFT does not have good heuristics to choose the best way to compile them. However, the programmer can use the invocation options and the matching compiler directives to investigate performance trade-offs in particular loops.

Many other invocation options and many compiler directives can affect job performance. In general, these other options have rather specific effects on rather limited code sequences. These should certainly be looked at, but are probably best tried in the main kernels of production codes rather than in casual development. The default values for these options are nearly optimum for most applications.

Closing statements

In addition to the latest CFT releases, users interested in new features, user friendliness, and better performance must not overlook Cray's new Fortran compiler, CFT77. (See the Summer 1986 issue of CRAY CHANNELS.) Like CFT, CFT77 is an ANSI 77 conforming compiler and it supports the same language extensions that CFT supports. It has several new features, including an array processing syntax. It vectorizes the same types of DO loops that CFT vectorizes. It has a "middle pass" which means it performs many more optimizations — both scalar and vector — than CFT, and does them without user intervention.

There have been many changes made to the Cray Fortran compilers in the past two years. Some aid coding and debugging, while others improve performance. The compilers vectorize many more types of loops than they did in the past. Users can significantly decrease a program's execution time by using the latest releases and experimenting with compilation options. □

About the author

Dick Hendrickson is a senior programmer analyst in the compilers department for Cray Research in Mendota Heights, Minnesota. Since joining Cray Research in 1976 he has worked primarily on the Cray Fortran compiler, CFT. His current interests are in evaluating the performance characteristics of the company's Fortran compilers. Hendrickson received a Ph.D. in physics from the University of Minnesota in 1972 and continued there as a research associate until 1976.

CORPORATE REGISTER

VM Station enhanced

Release 4 of the VM Station continues to provide users with easy access to Cray computer systems from mainframes running IBM's Virtual Machine/System Product. Release 4 of the Station adds the support of the new Cray operating system UNICOS and broadens coverage of the expanding range of COS applications. New features added for release 4 of the VM Station include

- ☐ Compatibility with UNICOS release 2.0
- ☐ Minidisk file FETCH and ACQUIRE transfers
- ☐ Station internal multitasking
- ☐ Automatic closure of console and hard copy at specified times
- ☐ Disposed spool file text field parameters to set spool file name/type, distribution code, and VM user identification
- ☐ Dataset transfer notification to local VM users.
- ☐ VMTAPE product support
- ☐ IBM 3480 tape drive support
- ☐ Common station slot support
- ☐ CRDISK and CRGRAPH command enhancements
- ☐ CRSTAT status display enhancements, including COS tape device configuration (TAPE) display, COS tape job status (TJOB) display, and PF key definition support
- ☐ CRSTAT functionality for remote VM users
- ☐ A task-oriented Primer that provides new users with an introduction to the Cray computer system through the VM station

The new Cray operating system interface requires UNICOS release 2.0 running the UNICOS Station Call Processor (USCP).

Cray Fortran compiler enhanced

Release 1.15 of the Cray Fortran compiler CFT includes many new features, including several enhancements to CFT's powerful optimization features. The latest optimization enhancements automatically vectorize additional DO loops and speed up the execution of many vector loops. These enhancements include

- ☐ Generalization of constant increment integers (CIIs). This feature allows vectorization of DO loops containing type REAL values that are incremented once during each pass through the loop. CIIs are referred to in CFT 1.15 documentation as constant increment values (CIVs) to reflect the change.
- ☐ Vector temporaries. CFT 1.15 attempts to move the store of vector temporaries to the end of a DO loop, saving the time that would be taken to store the value inside the loop. The value of the vector temporary A in the following example is stored only after the loop has completed:

```
      DO 1 I=1,N
         A=B(I)*C(I)
         D(I)=A+E(I)
    1 CONTINUE
```

  This feature improves performance for loops containing vector temporaries that have a trip counter greater than 64.
- ☐ Vectorization of search loops. CFT 1.15 generates vector code for search loops. A search loop is one that contains a conditional branch out of the loop or a conditional block of code ending with an unconditional branch out of the loop.

CFT 1.15 also offers new features that make the compiler more versatile and easier to use, including

- ☐ UNICOS compatibility. CFT fully supports Cray Research's new UNICOS operating system. CFT programs will execute under UNICOS on CRAY X-MP and CRAY-1 systems.
- ☐ CPU targeting. CFT 1.15 enhances the CPU targeting feature introduced in CFT 1.14, which allows the CFT user to specify the hardware characteristics of the target mainframe for a CFT program. CFT 1.15 lets the COS CFT user specify characteristics of the target CPU either with the TARGET control statement, which applies to an entire COS job, or in the individual CFT control statement. Several new CPU target characteristics have been added.
- ☐ DO-loop table enhancements. CFT's DO-loop table, a user-selectable list option, has been revised to contain more information about DO loops, including which loops vectorized, which did not vectorize and why, which were replaced by library routine calls, and which were unrolled.

For more information on CFT's new features, see article on page 22.

Cray systems serve a global market

Cray system sales and installations continue to reflect the diversity of the supercomputer market. Recent sales demonstrate active government, commercial, and university sectors, and a growing international demand for Cray systems.

In August Cray Research announced the order of a CRAY-2 computer system by a consortium of French educational and governmental research organizations, the Groupement pour un Centre de Calcul Vectoriel pour la Recherche. The system will be housed in the consortium's computer center at L'Ecole Polytechnique, an engineering school in the Paris area. The system is scheduled for installation during the first quarter of 1987, pending export license approval. The CRAY-2 system will replace a CRAY-1 S/1000 system installed in 1983, and will be used for academic, meteorological, aerospace, mathematical, oceanographic, and nonnuclear military research.

In August Cray Research also announced that ARCO Oil and Gas Company of Dallas, Texas, upgraded its CRAY X-MP/24 computer system to a CRAY X-MP/28 system and added an SSD solid-state storage device. The system upgrade was installed during the third quarter of 1986 at the ARCO Exploration and Technology Company in Plano, Texas.

In August Cray Research also announced that the Max Planck Gesellschaft in West Germany had ordered a CRAY X-MP/24 computer system to be installed during the second quarter of 1987, pending export license approval. The purchased system will be installed at the computer facility of the Institute for Plasma Physics in Garching, near Munich.

The University of California, Berkeley, announced the installation of a CRAY X-MP/14 computer system in October. The system will be used primarily by students and faculty to conduct research in areas including chemistry, physics, and mathematics.

Cray Research announced in October that a CRAY X-MP/24 computer system will be installed during the first quarter of 1987 at the University of West Berlin, pending export license approval. The system was ordered by a West German government-funded research group and replaces a CRAY-1/M system in operation since 1984. The CRAY X-MP system will be used by universities in Berlin and northern Germany for research in areas such as computational chemistry, structural analysis, and weather modeling.

In October Cray Research also announced that the U.S. Army had ordered two Cray computer systems: a CRAY X-MP/48 system with an SSD storage device and a CRAY-2 computer system. Both computer systems are scheduled for installation during the first half of 1987 at the Army Ballistics Research Laboratory in Aberdeen, Maryland.

In November Cray Research announced that CERN, a European particle physics laboratory, had ordered a CRAY X-MP/48 computer system with an SSD storage device. The system is scheduled for installation during the fourth quarter of 1987 at the CERN computing center in Geneva, Switzerland, pending export license approval. The Cray system will be used to analyze data involved in particle physics research.

Cray Research also announced in November an order for a CRAY X-MP/48 computer system from Kernforschungsanlage Julich (KFA), a West German national research laboratory. The purchased system is scheduled for installation during the first quarter of 1987 at KFA's computer facility in Julich, West Germany, pending export license approval. The Cray system will be used for basic scientific research in areas such as solid-state physics, computer science, and high-energy physics.

In December Cray Research announced that Scientific Computer Centers (SCC) had installed a CRAY-1/M computer system at SSC's computer facilities in Houston, Texas. SCC is a service bureau providing supercomputer services to the scientific and engineering world. The Cray system will be used specifically for petroleum, aerospace, biotechnology, advanced graphics, and electronics research.

Cray Research also announced that the U.S. Department of Energy, Richland Operations, ordered a CRAY X-MP/12 computer system to be installed the first quarter of 1987 in Richland, Washington. The supercomputer will be used for reactor safety analysis, geological and groundwater studies, advanced reactor design, and radiography, physics and chemistry studies.

Also announced in December was the order of a CRAY X-MP/18 computer system by the Office National d'Etudes et de Recherches Aerospatiales (ONERA). The system will be installed at ONERA's computer facility in Chatillon, France, in the second quarter of 1987, pending export license approval. ONERA was established to develop, direct, and coordinate aerospace research in cooperation with other French scientific and technical research organizations. The CRAY X-MP/18 system will replace a CRAY-1 S/2000 in operation since 1984.

Also in December, Cray Research announced that Pratt & Whitney Government Products, a division of United Technologies Corporation, had ordered a CRAY X-MP/28 computer system with SSD solid-state storage device. The system will be installed the first quarter of 1987 at Pratt & Whitney's facility in West Palm Beach, Florida. The Supercomputer will be used to support various jet engine design projects including the National Aerospace Plane propulsion system.

Cray Research also announced in December that the French automotive group Peugeot S.A. had ordered a CRAY X-MP/14 computer system. The system will be installed in the first quarter of 1987 at the headquarters of Automobiles Citroen, a subsidiary of Peugeot. The Cray system will be used for vehicle research, development, and design; including structural analysis, combustion, acoustics, aerodynamics, and crash simulation.

APPLICATIONS IN DEPTH

Automotive conference explores supercomputer applications

Design engineers, software developers, and data processing executives learned more about the impact of supercomputers in the automotive industry at a conference held recently in Zurich, Switzerland. The "International Conference on Supercomputer Applications in the Automotive Industry" was held October 7-9, 1986, and attracted more than 125 attendees from the United States, Europe, and Japan.

The opening keynote address was delivered by Dr.-Ing D. Radaj of Daimler-Benz AG. His talk explored the historical development, current status, and trends of computational analysis in the automotive industry. Dr. Radaj's overview spanned more than 75 years of automotive history and included examples of logarithmic slide rules, calculating cylinders, and mechanical desk calculators commonly used in early automotive design. He presented statistics from his own experience at Daimler-Benz reflecting the growing demand for computational analysis in automotive design; demand largely brought on, he noted, by the increasing use of simulation techniques and the expanding range of powerful applications available to the design engineer.

Other presentations focused on structural analysis, simulation of internal and external fluid flow as it relates to aerodynamics and passenger compartment comfort, design optimization, expert systems as an engineering decision-support tool, and full-vehicle simulation. Two applications that received particular attention were crashworthiness simulation and engine combustion modeling. While presenters noted that these applications were primarily experimental a few years ago, both are now routinely implemented in commercial environments. These applications as well as others from the conference are included in the feature section of this issue of CRAY CHANNELS.

Bound copies of the conference proceedings are available from Computational Mechanics Publications, Ashurst Lodge, Ashurst, Southampton, S04 2AA, England, or directly from Cray Research, Inc., 1333 Northland Drive, Mendota Heights, MN 55120, attention: Linda Yetzer.

BEASY boundary element code enhanced

The BEASY engineering analysis package, available on Cray systems, has been developed over the last year to include major enhancements. BEASY uses the boundary element method to solve engineering problems in thermal and stress analysis, field problems, and corrosion protection system simulation. BEASY is available with self-contained pre- and postprocessors, or may be interfaced to popular CAD packages.

The boundary element method (BEM) used in BEASY differs from the widely-used finite element method (FEM) in that only the surface (or boundary) of a body needs to be described by elements. Lines are boundary elements for two-dimensional problems; triangles and quadrilaterals are boundary elements for three-dimensional problems. This method has two main advantages: BEASY models are very quick and easy to generate, and BEASY models are very simple to change because all changes are local. These features, among others, make BEASY an attractive package for component design, particularly for the initial stages of the design process.

Although the equations formed by BEASY refer only to boundary values (such as displacements, temperatures, and stresses) on the surface of the body, results inside the body are obtained very easily and selectively using "internal points." These are specified by coordinates at any desired locations inside the material to be analyzed. Computing the behavior at all internal points from the boundary solution is the last stage in the BEASY analysis. Because the boundary solution can be stored in a file, users can easily go back repeatedly to find results for more and more internal points. The postprocessor includes a variety of options for displaying internal point results.

Recent enhancements to BEASY include

- ☐ Transient thermal analysis capability. BEASY now solves problems involving time-dependent heat transfer governed by the diffusion equation. An efficient time-stepping scheme is used.
- ☐ Nonlinear thermal analysis for steady-state and transient problems. Here, the material properties vary as functions of temperature.
- ☐ Applications of "internal conditions." Users can now specify loads and conditions internal to the problem. For example, internal displacement restraints, sliding interfaces and thermally permeable membranes.

For more information on using BEASY with Cray computer systems, contact

Jon Trevelyan, Computational Mechanics Inc., 400 West Cummings Park, Suite 6200, Woburn, MA 01801; telephone: (617) 933-7374; or Andy Mercy, CM BEASY Limited, 52 Henstead Road, Southampton SO1 2DD, United Kingdom; telephone: (0703) 221397; or the engineering applications group, applications department, Cray Research, Inc., 1333 Northland Drive, Mendota Heights, MN 55120; telephone: (612) 681-3652.

Boeing announces revised EASY5

Version 3.2 of the EASY5 engineering analysis system is now available from Boeing Computer Services. EASY5 is a complete set of model building and dynamic analysis tools that has been optimized for use on Cray computer systems. EASY5 integrates model building, analysis, simulation, and design capabilities in a single engineering environment. The system's interactive execution enables users to watch simulations execute, pause, change inputs, then restart. EASY5 requires no reformatting of results or programming between steps.

EASY5 has an extensive library of components (which users can supplement) that speeds up the translation of a block diagram into an executable model. Mathematical relationships for components also can be expressed directly in the model. Models can be verified by generating a block diagram schematic that describes all connections made, and a list of input data requirements that describes all data necessary for model execution.

EASY5's nonlinear analysis provides complete simulation capability plus a superior steady-state finder. Nonlinear simulation offers eight separate algorithms including BCS-GEAR, Boeing Computer Services' version of Stiff-Gear that handles both continuous and discrete models. Users also have the option of adding their own algorithm.

EASY5 also can generate a linear model at any operating point of the nonlinear model and evaluate the quality of the linearization. This procedure reduces model building time and eliminates discrepancies that might result from independently developed versions.

Version 3.2 of EASY5 introduces techniques to handle control system discontinuities efficiently — discontinuities such as mechanical gear transmission (for example, where a clutch plate stops slipping and becomes fully engaged) or electrical switching circuits.

In addition, a multivariable control system design technique, parameter optimization, has been incorporated. The technique manipulates up to 20 model parameters to maximize a user-defined objective function while satisfying user-defined constraints in the form of desired frequency response characteristics.

Typical EASY5 applications include

- ☐ Passenger vehicle dynamic performance assessment
- ☐ Aircraft and missile simulation and control
- ☐ Power plant modeling and control analysis
- ☐ Chemical process simulation and control

For more information on using EASY5 with Cray computer systems, contact John Corrie, Boeing Computer Services, P.O. Box 24346, Mail Stop 7 W-01, Seattle, WA 98124-0346; telephone: (206) 644-6437.

ABAQUS enhanced for vector computers

ABAQUS is a comprehensive finite element code for nonlinear and linear structural and heat transfer analysis. Over the past year, enhancements have been made to the program for improved performance on vector processing computers. As a result, ABAQUS provides efficient solutions to complicated models, such as those used for automobile crashworthiness studies, metal forming simulations, and nuclear power plant design.

ABAQUS includes independent libraries for element modeling, material specification, and analysis procedures. The element library provides beam, shell, and continuum formulations, including elements capable of representing incompressible behavior. Material behaviors include time-independent and time-dependent metal plasticity models, rubber, concrete, clay, and sand. In addition, users can specify their own material behavior by user subroutines. The analysis procedures in ABAQUS include statics (full Newton iteration and equal arc length methods for unstable response), dynamics (implicit and explicit algorithms for direct integration and modal superposition methods for linear response), eigenvalue extraction, eigenvalue buckling, and heat transfer (transient and steady state). The modular approach incorporated in ABAQUS provides the program with the flexibility demanded of a truly general-purpose program, and simplifies the rigorous quality assurance procedures required of the software.

ABAQUS selects appropriate timestep increments automatically, based on user-specified accuracy requirements. This feature of the program allows users to achieve results for difficult nonlinear analyses efficiently and with a minimum of user interaction. For example, in a nonlinear dynamic analysis involving impact, when the structure's response is dominated by high-frequency response, ABAQUS will typically use small time increments. And when the response is dominated by low-frequency behavior, large time steps are used.

Proven ABAQUS applications include automotive crashworthiness simulations, pipe whip and high-temperature reactor component analyses, metal forming operations, and general elastomer analyses.

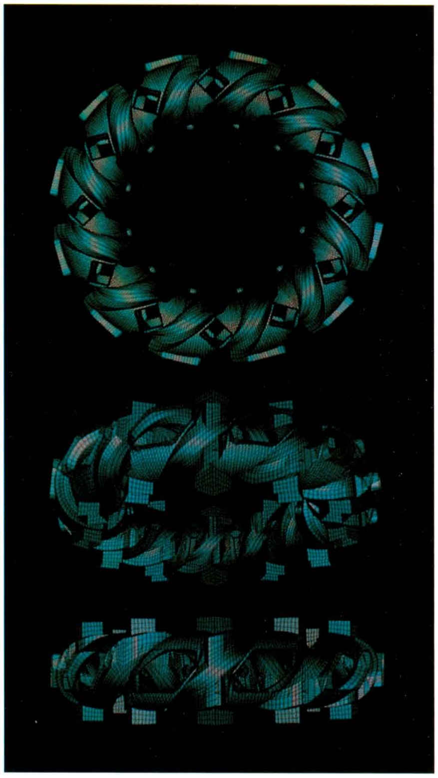
For more information on using ABAQUS with Cray computer systems, contact: E. P. Sorensen; Hibbitt, Karlsson & Sorensen, Inc., 100 Medway Street, Providence, RI 02906; telephone: (401) 861-0820; or the engineering applications group, applications department, Cray Research, Inc., 1333 Northland Drive, Mendota Heights, MN 55120; telephone: (612) 681-3652.

USER NEWS

Fusion experiment takes shape

When it begins operating later this year, the Advanced Toroidal Facility (ATF) will be Oak Ridge National Laboratory's latest nuclear fusion confinement device, extending a series of experiments that began in the 1950s. The ATF will suspend a superhot ionized gas, or plasma, in its magnetic field, allowing lab physicists to develop methods for confining nuclear fusion reactions. Successful confinement of such reactions may yield a clean, inexhaustable source of energy. Fusion energy would require only readily available hydrogen for fuel and would generate inert helium as waste. But designing a structure to withstand the physical stresses involved requires an advanced computational facility. In the case of the ATF, a Cray supercomputer provided some of the needed processing power.

Cray systems are being used by national laboratories and corporations in the United States and abroad to study fusion reactor design. The expense and complexity of fusion confinement make computer modeling critical for defining



Aerial view of the ATF assembly (left) at Oak Ridge, onto which the vacuum vessel will be lowered, and views of a computer-generated model of the vacuum vessel (right) used for structural analysis. The original structural shape of the vessel was generated mathematically and translated into MOVIE.BYU format for the initial conceptual sketches. As the design project matured, the MOVIE.BYU geometry was translated into an MSC/NASTRAN bulk data deck and an analysis of the structure was conducted on a CRAY-1 computer system. The MSC/NASTRAN bulk data deck was translated back into a MOVIE.BYU format, and these images were generated, during an MSC/NASTRAN seminar held in September at Cray Research's Mendota Heights, Minnesota, facility.

safe and useful operating parameters before containment structures are built.

At Oak Ridge, three tokamak configurations have been built for magnetic fusion research. A *tokamak* is a doughnut-shaped device originally developed for fusion research in the Soviet Union. The ATF at Oak Ridge, however, is based on a concept called a *stellarator*, or *torsatron*, which was first developed by Lyman Spitzer at Princeton University. The ATF retains the tokamak's doughnut shape, but its two primary coils are twisted to create a helical, or spiral, magnetic field. The helical field will prevent particles of unlike charge from separating in the plasma, enhancing the plasma's stability.

The intensity of the coils' magnetic field makes devising the confinement structure a major design and engineering challenge. Confinement is absolutely crucial because the plasma will reach temperatures near 100 million°F. Thus, structural analyses performed prior to building such a device are crucial, and supercomputers are a practical necessity for carrying out such analyses.

"We used Cray systems for structural analysis of the ATF primarily because of the size of the problem," said David Williamson, a mechanical engineer at Oak Ridge National Laboratory. "Calculations that took only 30 CPU minutes on a CRAY-1 system would have taken up to 12 CPU hours on mainframes typically used for large engineering problems. This time savings is significant in speeding up the whole design and development process." Structural analyses of the ATF were conducted on a CRAY-1 computer system at Lawrence Livermore National Laboratory in California.

"We performed analyses of the vacuum vessel to be sure it wouldn't buckle under atmospheric pressure," Williamson explained. "We also ran studies of the electromagnetic loads due to the currents running through the magnetic coil. These studies helped us determine stresses on the outer structure, on the coil support structure, and within the coil. These computations are very involved and require fast computers to make them practical to solve, particularly when the model needs to be modified and run several times." The structural analyses are now complete, and the project is proceeding to the construction phase.

Unfortunately, atoms do not come with instruction manuals. Safely tapping the energy source within them will only result from a long process of exploration and learning. Merely designing the tools needed to study fusion reactions is an involved engineering endeavor. But supercomputer modeling provides a way to accelerate the design of devices such as the ATF and thereby significantly advance the state of the art.

The eyes have it

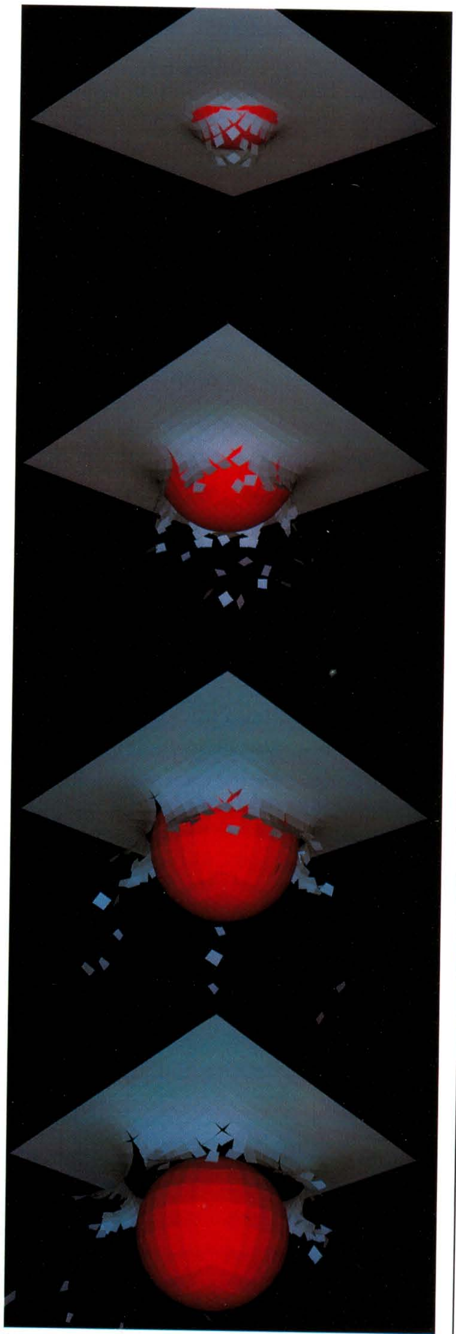
A picture is worth a thousand lines of printed output. This becomes an inescapable conclusion to anyone analyzing the data produced by supercomputers. The value of images for presenting large quantities of scientific data is demonstrated by the following examples from Lawrence Livermore National Laboratory in Livermore, California. Researchers in the laboratory's mechanical engineering division are developing applications for their finite element program, DYNA3D, which was run on a Cray supercomputer to produce the results shown here graphically.

A crowning step

The finite element method has proven its value in diverse applications. But the method's usefulness is limited by its poor ability to model the tearing of materials. If it could be used to model tearing accurately, the finite element method would find many additional uses. This is a goal the researchers at the Livermore lab are pursuing by adapting DYNA3D to model tearing.

The preliminary study shown here demonstrates the code's ability to model a type of tearing called *petaling*, which occurs during ballistic penetration. Petaling produces the crown-shape of torn material that often surrounds a puncture in a metal surface.

"The calculations required to produce these images were fairly complex; each took about 20 to 30 minutes to compute on a CRAY X-MP/48 computer," explains David Benson, co-developer of DYNA3D at the Livermore lab. "We need to do more work on DYNA3D before we can regard it as a reliable tool for penetration analysis. But this demonstration is a promising first step."



Various stages showing the penetration of a metal surface by a projectile. These images demonstrate an approach to modeling the tearing of materials.

FEA as design tool

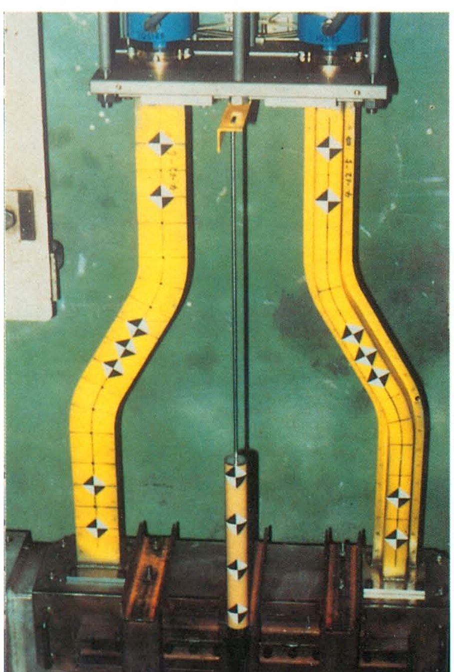
Among recent projects, the Livermore lab researchers have applied DYNA3D to studies of automobile crashworthiness. Suzuki Motor Co., Ltd., provided the researchers with the structural data for a car chassis frame member along with the automaker's experimental results from a 30 kilometer-per-hour impact against a barrier.

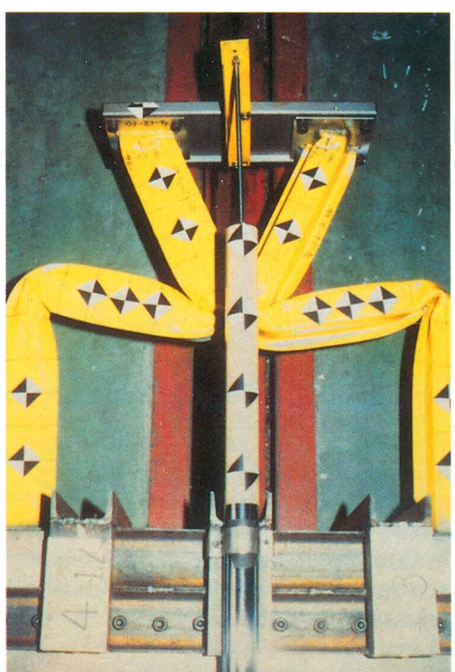
The frame member was tied at each end to a rigid body, one representing the barrier and the other representing the sled which provided the momentum to crush the frame. The constitutive model was the Prandtl-Reuss model with isotropic, linear strain hardening. The analysis shown here used a mesh of 12,000 shell elements, resulting in approximately 72,000 degrees of freedom, with four integration points through the thickness.

An earlier simulation using 1600 shell elements provided similar results. This run required a little over four CPU hours to simulate the entire event on a CRAY X-MP/48 system. The peak deceleration, an important number to chassis designers, which occurs at only five milliseconds after impact, can be calculated on the Cray computer system in less than one-half CPU hour with the 1600-element model.

"The results of the lab analysis matched the peak deceleration almost exactly, although its duration was too short. The discrepancy probably was caused either by an overly simple material model or by the 2000 Hz filter that Suzuki used to smooth their data," explains John Hallquist, lead code developer in the Livermore lab's mechanical engineering division. "Based on the accuracy and cost of our results, finite element analysis in crashworthiness design should not be regarded strictly as a research tool, but as a tool for the designer."

These are two examples among many that demonstrate the time- and labor-saving advantages of finite element modeling over traditional engineering analysis methods. Supercomputers such as Cray systems open the range of finite element applications to the largest and most complex problems now solvable. Improved programming methods also contribute to expanding the range of solvable problems. And throughout, graphic representations allow a quick and complete interpretation of scientific data; data that would otherwise by unmanageable by its sheer volume.





Car chassis frame member before (left) and after (right) 30 kilometer-per-hour impact.

Three frames from a high-speed film of the impact simulation. The buckling of the simulated beam shows good agreement with the buckling of the physical beams.

WORLDWIDE BUSINESS CENTERS

Domestic

Corporate Headquarters
608 Second Avenue South
Minneapolis, MN 55402
(612) 333-5889, telex 4991729

**Government Marketing/
Government Relations Office**
National Place
1331 Pennsylvania Avenue, Suite 1331 N
Washington, DC 20004
(202) 638-6000

Central Region Office
5350 Manhattan Circle
Boulder, CO 80303
(303) 499-3055
Albuquerque
Chicago
Colorado Springs
Detroit
Minneapolis/St. Paul
St. Louis

Eastern Region Office
4041 Powder Mill Road, Suite 600
Calverton, MD 20705
(301) 595-5100

Atlanta	Laurel
Boston	Pittsburgh
Cincinnati	Rochester
Huntsville	Tampa

Petroleum Region Office
5847 San Felipe, Suite 3000
Houston, TX 77057
(713) 975-8998
Dallas
Tulsa

Western Region Office
5776 Stoneridge Mall Road, Suite 350
Pleasanton, CA 94566
(415) 463-2800
Los Angeles
Seattle
Sunnyvale

International

Cray Asia/Pacific Inc.
Suite 1003
Sutherland House
3 Chater Road Central
Hong Kong
(5)-214669, telex 89765

Cray Canada Inc.
4211 Yonge Street, Suite 210
Willowdale, Ontario
Canada M2P 2A9
(416) 229-2729

Cray Research France S.A.
7 Rue de Tilsitt
75017 Paris, France
(01)-766-01-55, telex 660568

Cray Research GmbH
Perhamerstrasse 31
8000 Munich 21, West Germany
(089)-56014-0, telex 5213211

Cray Research Japan, Limited
W 1154, Shin Aoyama Building
1-1 Minami-Aoyama 1-chome
Minato-Ku, Tokyo 107, Japan
(03)-403-3471, telex 28893

Cray Research S.R.L.
Via Vivaio, 15
20122 Milan, Italy
(02)-70-98-27, telex 315403

Cray Research (UK) Ltd.
Cray House
London Road, Bracknell
Berkshire RG12 2SY
United Kingdom
(0344)-485971, telex 848841

CRAY RESEARCH, INC.