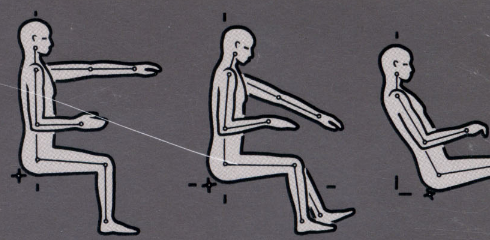
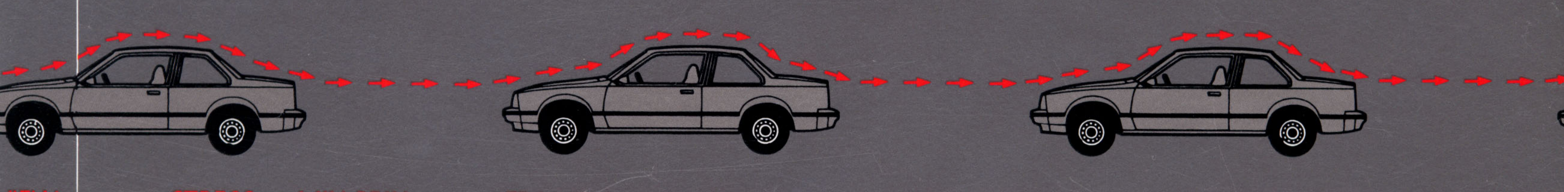
CRAY
RESEARCH, INC.
608 Second Avenue South
Minneapolis, MN 55402
612/333-5889
Telex: 6879144





CRAY CHANNELS

WINTER 1989 · A CRAY RESEARCH, INC., PUBLICATION

Supercomputers in automotive development

Announcing automatic parallel processing and larger memory CRAY-2 model

CRAY CHANNELS

In this issue

It doesn't take a computer expert to notice that today's automobiles look remarkably different from those designed just a few years ago. Not only do they appear more sleek and aerodynamic, but many are safer, more fuel-efficient, and easier to drive. Much of this progress has been achieved through supercomputer-aided design.

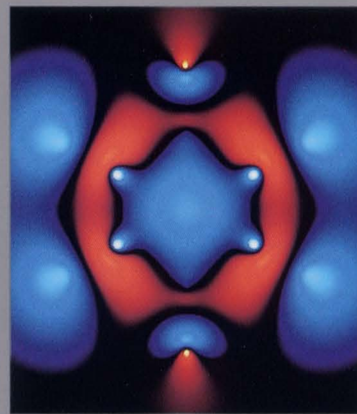
Nor does it take a computer expert to recognize that supercomputers are becoming easier to use. Parallel processing, a fruitful but elusive method for improving problem turnaround time, has taken a dramatic step forward with Cray Research's new "autotasking" capability. This issue of CRAY CHANNELS describes how autotasking improves program performance without extra effort on the part of users.

Cray Research gained its first automotive customer about five years ago. Since then, Cray systems have become important design tools for automotive companies around the world, including BMW, Chrysler, Daimler-Benz, Opel, Fiat, Ford, General Motors, Honda, Michelin, Nissan, Peugeot, Toyota, and Volkswagen. Each of these companies is finding new ways to use supercomputers to improve vehicle design. For example, less than three years ago, crashworthiness simulation of frontal impact was just beginning to emerge as a practical design tool. Today, researchers at Ford are taking crash simulation beyond studies of front-end impact, into side-impact simulation. This issue of CRAY CHANNELS includes stories from Ford as well as from Michelin, BMW, Peugeot, and Renault.

This issue of CRAY CHANNELS also introduces the new CRAY-2/4-512 system. With 512 million words of memory, the new system has the largest directly addressable memory of any commercially available computer system. In addition, our regular departments feature announcements of VAX/VMS Station release 4.01 and Cray Link Software, as well as summaries of two fluid dynamics codes. Our User News department showcases three unique applications of Cray systems: modeling snow formation, transportation systems, and bearing wear.

From modeling tires and trucks to automobiles and their electronic components, Cray systems offer a cost-effective way to solve a diversity of problems. With Cray Research's new autotasking capability, users also can improve computer performance without added effort — another example of Cray Research's commitment to provide users with the best possible tools to accomplish their goals.

Features



4

Departments

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 feature story ideas, news items, and "Gallery" submissions to CRAY CHANNELS at Cray Research, Inc., 608 Second Avenue South, Minneapolis, MN 55402.

Volume 10, Number 4

Editorial staff
Tina Bonetti
Ken Jopp
Elizabeth Knoll
John Swenson, editor

Design and production
Barbara Cahlander
Eric Hanson
James Morgan
Cynthia Rykken

On the cover is a model of a Chrysler LeBaron. The model was generated on a Cray system for use in design studies. Automakers around the world are using Cray systems to study and improve vehicle aesthetics, crashworthiness, performance, fuel-efficiency, and other important characteristics. (LeBaron data courtesy Chrysler Motors.)

Autotasking

Automatic parallel processing on Cray systems

Parallel processing long has been recognized as a potentially powerful tool for faster computing. But the relatively high level of programming expertise required to exploit it has limited its use. Parallel processing tools until now have required that users either restructure their programs or add to them a series of directives after having performed a detailed data analysis. But with release 3.0 of the Cray Fortran compiler CFT77, Cray Research provides a compiling system that partitions programs among multiple processors automatically, a capability we call *autotasking*.

Autotasking brings the benefits of parallel processing to the broad community of Cray system users. It enables users to run programs in parallel regardless of their programming expertise. Autotasking also can support programmers who want to exploit potentially higher levels of parallelism. On a four-processor system, autotasking can result in performance improvements of up to 3.9 times, and on an eight-processor system, in improvements of up to 7.8 times, that of single-processor execution.

"Autotasking enables researchers to concentrate on their science," explained Robert Ewald, executive vice president, software, for Cray Research. "It frees them from having to rewrite or specially structure programs to take advantage of parallel processing. This new capability can help researchers use their computing resources more efficiently."

Autotasking can be used on multiprocessor CRAY Y-MP, CRAY X-MP EA, and CRAY X-MP computer systems that run under the UNICOS operating system. Autotasking support will be added for CRAY-2 systems in early 1989 under release 5.0 of the UNICOS operating system. Support for Cray systems running the COS operating system also will be available in 1989.

The compiling system

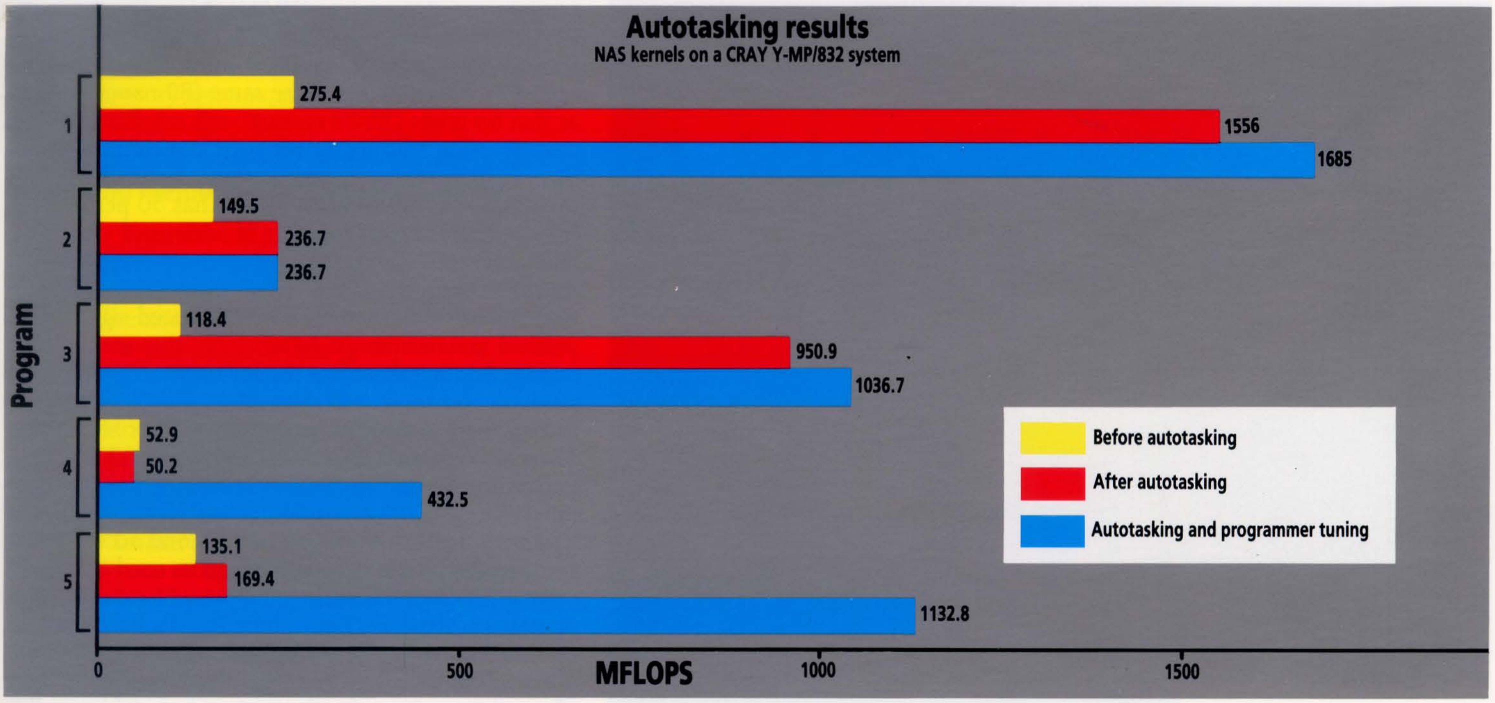
The compiling system that makes autotasking possible consists of three phases: dependence analysis, translation, and code generation. The dependence analyzer looks for parallelism within program units. It recognizes iterations of DO loops that are independent and inserts directives that express the parallelism. The autotasking directives tell the translator how and where to restructure the code for parallel execution. The translator then produces Fortran source code containing calls to machine-dependent library routines and compiler-intrinsic functions to control parallel execution. The code generator then produces optimal executable machine code for execution on multiple CPUs.

Programmers can invoke the three compiling system phases together or pass directives to them individually. For example, a programmer knowing that a certain large program segment accounts for only 1 or 2 percent of a code's total run time may choose to disable the extensive dependence analysis performed in this region because no payoff would result (and the programmer would save compilation time). Or a programmer may know that a DO loop that contains an external call may be safe to execute in parallel, and wish to alert the compiling system to that fact (the dependence analyzer does not look beyond subroutine boundaries).

All three components can be invoked with the **cf77** command, which functions similarly to the **cc** command found on most UNIX systems. Alternatively, other simple commands may be used to invoke the phases individually. Directives may be sent to any of the phases to guide the kinds of analysis and transformation performed and to augment the overall parallel processing performance of the code.

The third generation

Autotasking represents the third generation of parallel processing software offered by Cray Research. The first generation enabled users to partition programs among multiple processors, but required



that they insert into their programs explicit calls to a special library of synchronization routines. This form of parallel processing, called *macrotasking*, works best when the amount of work to be partitioned over multiple processors is relatively large. If it is not, then synchronization time may become noticeable. Macrotasking offers significant performance improvements for some applications, but because it requires users to modify their programs it has not been widely used.

Microtasking evolved from macrotasking by addressing its weaknesses and building on its strengths. This method of parallel processing requires only that users insert compiler directives into their programs as comment statements. These comments indicate where parallel sections of code can be found, and are translated by a preprocessor into parallel Fortran code. The synchronization cost with this form of parallelism exploitation is extremely low. This feature is more user friendly than macrotasking and is used most often to exploit parallelism at the loop level.

Autotasking retains many of the advantages of microtasking, including self-scheduling to ensure balanced loads, very low synchronization overhead, use of idle CPU cycles, excellent performance in both batch and dedicated environments, and the ability to run source code in parallel without modification. However, it also relieves users of any need to indicate where in their codes parallelism exists.

Autotasking can be completely automatic; the compiling system can perform the analysis and generate parallel code. However, users also can use autotasking to support potentially higher levels of parallel processing. Autotasking works on DO-loop boundaries, but it easily expands to "parallel regions" and to subroutine boundaries.

Autotasking performance results from a benchmark from NASA's Numerical Aerodynamic Simulation (NAS) project. Results for each problem were obtained before autotasking, after autotasking, and after autotasking and some tuning by a Cray benchmark analyst. In cases where the analyst intervention made a significant difference, the analyst used information about parallelism inhibitors generated by the dependence analyzer, in addition to past experience and knowledge about the problem domains. These results probably are typical of the results that one is likely to obtain from autotasking and autotasking augmented by a knowledgeable programmer. Sometimes the automatic system can find little or no parallelism; other times it detects and exploits parallelism very effectively. However, an experienced programmer often can improve performance by, for example, applying knowledge about tripcounts for important DO loops, looking beyond subroutine boundaries, and rewriting algorithms.

Other CFT77 3.0 features

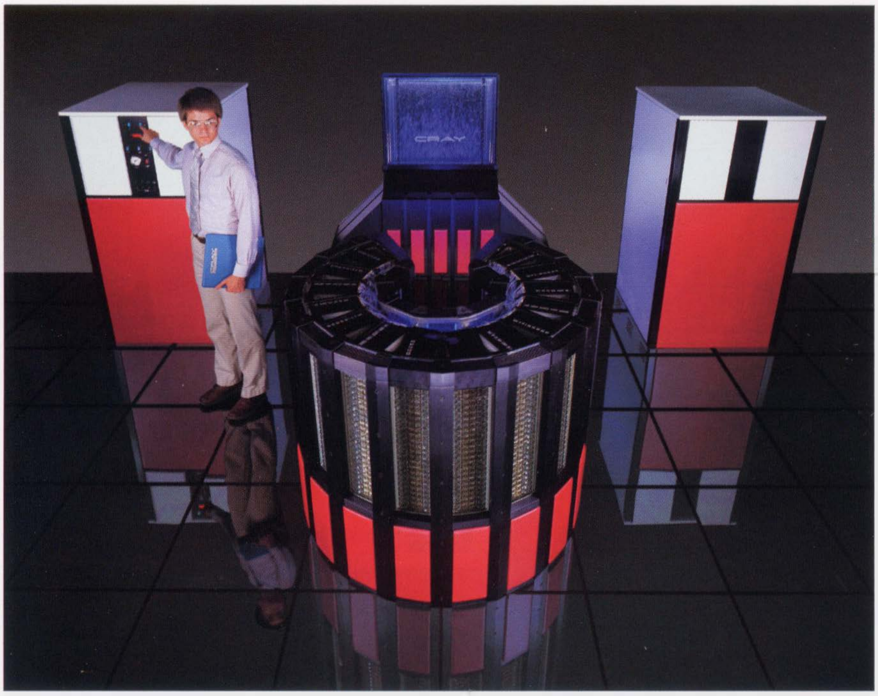
Release 3.0 of the CFT77 compiler runs on all Cray computer systems and under all Cray operating systems. A CFT77 program that can be compiled and run on one Cray system can be compiled and run on all Cray systems. In addition to autotasking support, release 3.0 of the CFT77 compiler provides the following new features:

- ☐ Debugging support for optimized code
- ☐ In-line subroutine expansion to improve performance
- ☐ New intrinsic functions that provide additional capabilities for bit manipulation
- ☐ A new command-line keyword that simplifies support of extended memory addressing (EMA)
- ☐ Additional cross-compile support for CRAY Y-MP and CRAY X-MP EA systems
- ☐ Character and numeric data equivalencing to aid in the migration of programs written for other systems
- ☐ Enhanced RANSET and RANGET intrinsic functions to improve usability and compatibility with the Cray Fortran compiler CFT

Release 3.0 of the CFT77 compiler also includes the following new features to improve performance on CRAY-2 systems:

- ☐ A new scheduling algorithm
- ☐ A new compiler directive, REGFILE, that forces specified common blocks to local memory and improves compatibility with other systems

Release 3.0 of the CFT77 compiler represents a milestone in supercomputer software development. By relieving users of the need to analyze programs for parallelism, autotasking makes more accessible the performance advantages of parallel processing. With release 3.0 of the CFT77 compiler, Cray Research demonstrates its commitment to providing users with the best possible tools to realize the greatest possible performance gains from their Cray systems.

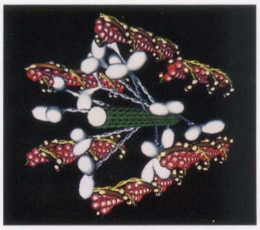


Introducing the CRAY-2/4-512 computer system

Expanding the boundaries of science

Larger memory and better performance make the new CRAY-2/4-512 computer system an exciting addition to the CRAY-2 series of computer systems. With the largest directly addressable memory of any commercially available computer system, the new CRAY-2 system helps researchers tackle problems that were once too large or complex to even dream of attempting. In addition, the CRAY-2/4-512 system features significant performance improvements over the original CRAY-2 computer system introduced in 1985.

One frame from an animated sequence of muscle contraction. This animation, which is used for teaching and research, was developed by University of Minnesota researchers Max Schaible, Richard Ludescher, and Joel Neisen.



Performance enhancements

The CRAY-2/4-512 computer system is ideal for applications with large memory requirements, such as molecular structure analysis, computational fluid dynamics, semiconductor design, and finite element analysis. Researchers can jump memory hurdles by applying the new CRAY-2 system's four processors and more than 512 million 64-bit words (approximately 4.3 billion bytes) of directly addressable common memory. This common memory is twice as large as the common memory of the original CRAY-2/4-256 system and is approximately the same as the capacity of four DD-49 disk drives. Common memory in the CRAY-2/4-512 system consists of 1-Mbit dynamic random access memory (DRAM) chips. The original CRAY-2 model uses 256-Kbit DRAMs. Despite this increase in capacity, the rate at which the CPU may access each chip remains the same (80 nanoseconds) as that for other DRAM systems with smaller memories. By using 1-Mbit DRAMs with four times as many bits per chip as those used in predecessor CRAY-2 models, the CRAY-2/4-512 system has 50 percent fewer memory parts, resulting in additional system reliability.

Since the first CRAY-2 system was shipped, Cray Research continually has introduced significant product enhancements. Advances in chip technology and availability, as well as the redesign of certain modules and functional units of the CRAY-2/4-512 system, have resulted in performance improvements of up to 30 percent over the original CRAY-2 system, depending on the application being run. For example, the dynamic memory components now used in CRAY-2 systems have an 80-nanosecond access time, compared with 120-nanosecond parts used in the original CRAY-2 system.

Instruction buffer capacity has been increased from 16 to 32 words, reducing the frequency of out-of-buffer jumps and thus reducing the number of memory accesses for some programs. Additionally, the new CRAY-2/4-512 design increases the number of pseudobanks from two to four, yeilding an increase in performance for most codes.

Other design changes that yield performance improvements include a reduction in the time required to issue successive local memory references. Also, over 175,000 discrete transistor parts have been eliminated, along with the 525,000 connections to those parts — changes that provide added system reliability.

The entire CRAY-2 line now comprises five models; two models offer very large dynamic memory and three models offer faster static memory.

Minnesota Supercomputer Center receives first system

The first CRAY-2/4-512 system was installed in December 1988 at the Minnesota Supercomputer Center (MSC), which is an affiliate of the University of Minnesota (U of M). The center has provided supercomputer services to the university, as well as other academic and commerical customers since 1982. "This acquisition is an important technological advancement for our program," said John Sell, president of MSC. "Our users are accustomed to pushing the limits of the most advanced computational technology available and the 512-Mword CRAY-2 system provides them with the capability to tackle previously intractable problems."

According to Sell, the large memory of the CRAY-2/4-512 system offers a welcome alternative to virtual memory that must be used on some systems as a substitute for real memory. Computer users are able to shed or avoid writing cumbersome code for an out-of-core solution process that substitutes mass storage for real memory.

For many researchers, the new CRAY-2 system will expand the scale of problems as well as the accuracy of solutions. Sell explained that the larger memory and higher performance may dramatically

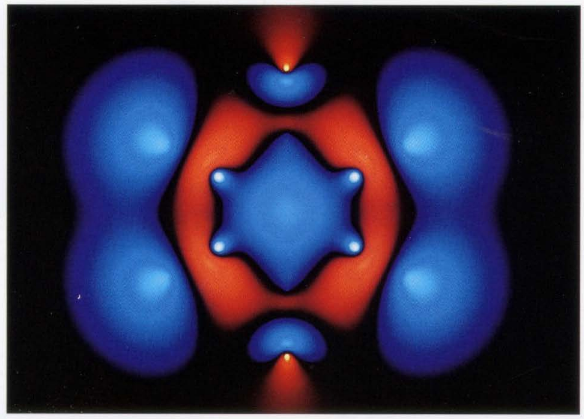


Figure 1. A molecular orbital of parabenzoquinone modeled by Martin Feyereisen and Jan Almlöf of the University of Minnesota Supercomputer Institute.

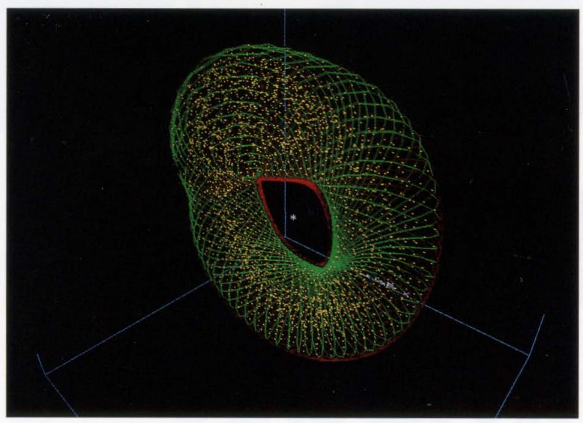


Figure 2. Charles Peskin and David McQueen of New York University's Courant Institute use the Minnesota Supercomputer Center's CRAY-2 system to model blood flow in a torus. One frame from an animated sequence is shown. This work is one step toward a full three-dimensional model of the human heart.

change the scope of research and pave the way for new solution methods. In many cases, two-dimensional models, which can be the largest models feasible on smaller-memory systems, are replaced by three-dimensional models on the CRAY-2/4-512 system.

Pioneering memory-intensive applications

"The new 512-million-word CRAY-2 system will open frontiers by solving problems we previously only could have dreamed of," said Donald Truhlar, director of the University of Minnesota Supercomputer Institute and chemistry department faculty member. "The new CRAY-2 system allows researchers to extend their programs with the same techniques, using main memory, that have been developed for our previous CRAY-2 system. This will allow us to accommodate larger problems without spending unnecessary effort reworking algorithms."

Truhlar's research group has been using a CRAY-2 system to model chemical reactions at the molecular level by applying quantum mechanics. His colleague, Jan Almlöf of the U of M's chemistry department, is using a CRAY-2 system to explore the electronics structures of large metal and carbon clusters. These studies can lead to a deeper understanding of chemical processes in the atmosphere and energy utilization technologies, and can provide a basis for engineering new molecular forms and substances (Figure 1). Researchers in chemical engineering and materials science are able to couple physical experiments with CRAY-2 computer simulations to model phenomena such as the dynamics of polymer chemistry, which may be difficult or impossible to study in the laboratory.

Researchers in the biomedical field use the CRAY-2 system's large memory to explore the patterns of life. Ernest Retzel of the U of M's microbiology department uses several hundred million words of memory on the CRAY-2 system to predict the shape of viral genomes and to compare these predicted structures among related species that have different biological functions. George Wilcox of the U of M's pharmacology department has simulated a neural network on a CRAY-2 system. The network can perform image classification and enhancement. This is accomplished by repeated exposure of the network to varying input images of tissue slices together with output images that the experimenters associate with each input. Gradually, the network "learns" to associate the proper result with a given input. Wilcox said that memory size is crucial to this application, with some simulations requiring from 100 to 400 million words. "Neural network approaches are well-suited to this task, yet the available neural computers and software are not nearly capable of handling the scale of problems we see in practical situations," he said. "The CRAY-2 system is the only computer that provides the resources to deal with these sorts of problems."

Charles Peskin and David McQueen of New York University's Courant Institute use the MSC CRAY-2 system to model the human heart in three dimensions. This staggeringly complex endeavor integrates chemical kinetics, nonlinear mechanics, and anatomical structure (Figure 2). The additional memory of MSC's new CRAY-2 system will enable more refined and complex three-dimensional studies of artificial valves, pathological valve defects, blood oxygenation kinetics, and normal resting-heart and stressed-heart physiology.

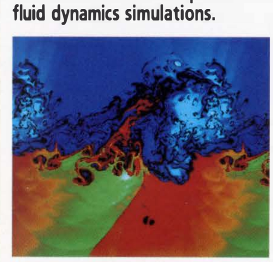
Also using the CRAY-2 system at MSC are the Minneapolis Federal Reserve Bank and the U of M Department of Economics. These two organizations jointly operate the Institute for Empirical Macroeconomic Studies, which is funded by the National Science Foundation. Researchers model the economy by sifting through observed data to distinguish patterns. The researchers attempt to relate the patterns to institutional and individual behavior, with the goal of better understanding the effects of macroeconomic policy.

Interactivity, connectivity, and compatibility

The CRAY-2 series of computer systems is complemented by the UNICOS operating system, which is based on the AT&T UNIX System V operating system. UNICOS is an interactive operating system that supports two automatic vectorizing Fortran compilers: CFT2, which is based on the Cray Research Fortran compiler (CFT); and CFT77, which represents the leading edge in compiler development. UNICOS also supports Cray Research's C, Pascal, and Ada compilers, libraries, tools, and utilities, and a growing range of scientific and engineering applications on CRAY-2 computer systems.

All of these improvements, including the largest memory configuration commercially available, and the interactivity, connectivity, and compatibility of the UNICOS operating system, add up to an advanced computer system that will help researchers expand the boundaries of science.

Interaction of gases at supersonic speeds, modeled by Paul Woodward and Jeff Pedelty at the University of Minnesota Supercomputer Institute. The CRAY-2 system is used extensively by industrial and academic researchers for such computational fluid dynamics simulations.



A numerical simulation of the proposed European side-impact test procedure

John Stanger, Peter Ng, and Tim Hatch
Ford Motor Company, Essex, England

Paul Du Bois, consultant, Offenbach, West Germany

In most countries, about 30 percent of car accident fatalities can be attributed to side impacts. To address this problem, many governments conduct research to develop appropriate test procedures on which to base side-impact legislation. The United States recently has published a "Notice of Proposed Rulemaking" on the subject, and a European proposal is being drafted by the EEC's European Regulation Global Approach (ERGA) group.

Motor vehicle manufacturers also have been active in this area, conducting test programs and evaluating and developing computer codes that can be used to improve the side-impact performance of their products[1] The evolution of supercomputers, such as Cray systems, has made these computationally intensive analyses feasible for the first time. This paper describes Ford Motor Company's application of the finite element code RADIOSS to side impacts as part of an overall design methodology for future Ford products.

Over the past few years several automotive companies have achieved considerable success in using finite element methods for the simulation of frontal impacts[2] Used in combination with traditional experimental methods, these analyses can provide engineers with important information early in the design process. It is now possible to compare many alternative designs and to optimize a structure for weight, stiffness, and strength before a full prototype vehicle is crash tested.

Finite element codes — explicit and implicit

The crash phenomenon is extremely complex and involves highly nonlinear effects. An accurate computer simulation must not only take account of large strains and displacements, but also rapidly changing surface contacts. The finite element codes capable of handling these analyses basically fall into two categories: explicit and implicit solvers. The use of an implicit time-integration operator means that the nonlinear dynamic equations are solved at each time increment by iteration. This process is expensive, and the equations can be so nonlinear that a solution is very difficult to obtain. The principal advantage of the implicit method is that it is unconditionally stable; this means that no mathematical limit restricts the size of the time increment that can be used.

The use of an explicit central-difference operator means that the solution is only conditionally stable and therefore requires the use of a very small time increment. However, because it is explicit, no system of equations is solved. Thus, the method is completely reliable and the cost per increment is small. In simple terms, the method consists of defining the acceleration at each degree of freedom from the dynamic equilibrium equation for that degree of freedom, then integrating forward to obtain the next displacement increment. A significant advantage of this method for crash analysis is that the cost rises linearly with problem size. With the implicit method, the cost rises more rapidly because of the need to solve nonlinear equations.

When studying crash behavior, an analyst typically is concerned only with the first 50ms of an impact. Even though this may represent 50,000 time increments for an explicit operator, it is usually the most cost-effective solution. The explicit finite element code RADIOSS-CRASH therefore was chosen for the side-impact study.

The RADIOSS-CRASH code

Developed by Mecalog SA, Paris, France, RADIOSS-CRASH is an explicit, Lagrangian finite element code capable of handling large displacements, large rotations, nonlinear material models, and a variety of complex contact conditions. It has been optimized for Cray supercomputers and has the advantage of a minimal core memory requirement. Thus it allows

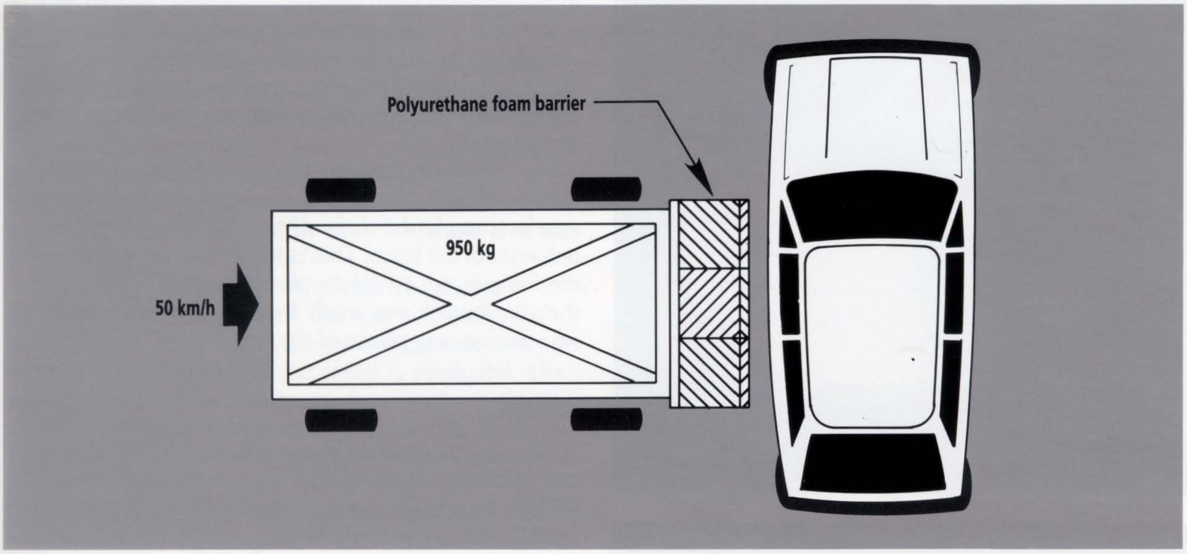


Figure 1. Representation of the proposed EEVC side-impact test procedure.

The CRAY X-MP system at Ford Motor Company in Detroit can be accessed directly from the United Kingdom.

other applications to run in parallel with the crash analysis. It may be the fastest code of its type available on the market today.

Modeling for side impact

The characteristics of a side-impact analysis are quite different from those of a typical frontal-impact analysis. Key events in a frontal impact, such as accordion folding of the front rail, crumpling of the wheel house, bending collapse of the aft rail, and contact of the engine with the dash panel, require the use of certain modeling techniques to simulate the events correctly. For example, very high curvatures arise in the folding and crumpling phase, requiring a very fine mesh to avoid an overstiff response.

Major events for a side-impact analysis include deformation of the foam or honeycomb barrier, contact between the barrier and outer panels of the car body, contact between outer and inner panels of the body, bending collapse of the B pillar, door, and rear quarter, and twist of the A pillar.

Consequently, very different issues have to be considered to model side impact effectively. For example, a coarser mesh can be used because all of the main structural events are global bending collapses. Also, all major force transmissions are through contacts between deformable bodies with very different stiffnesses.

In addition, the global movement of the vehicle is an important feature of side impact and seems to be a problem of relative inertias more than one of wheel friction upon the floor. Previous analyses based upon spring-mass models have indeed never emphasized the importance of this friction.[3] Of primary interest is the realistic mass distribution over the car model, particularly with respect to heavy components such as the engine.

To assess the feasibility of using RADIOSS-CRASH for side-impact simulations, it was decided to address a 50 km/h side impact between a small three-door hatchback car and the proposed European Experimental Vehicle Committee (EEVC) test barrier[4] (Figure 1). The accuracy and stability of the numerical results, the cost and duration of the analysis, and the availability and usefulness of the results were some of the key issues addressed.

The vehicle model

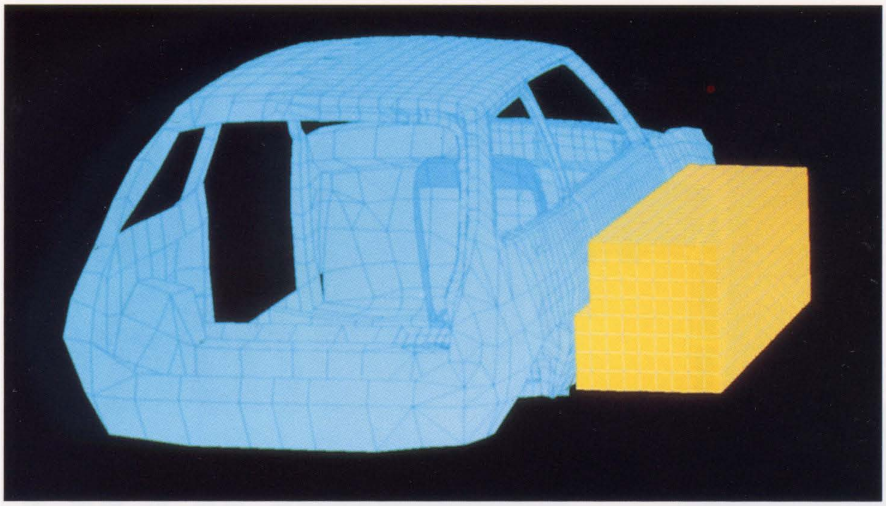
A complete and accurate description of stiffness and mass distribution for the target vehicle was essential for the analysis. The construction of a mesh totaling some 7000 elements was a major undertaking, but some time was saved by adapting a model that originally had been built for full-body linear static analysis.

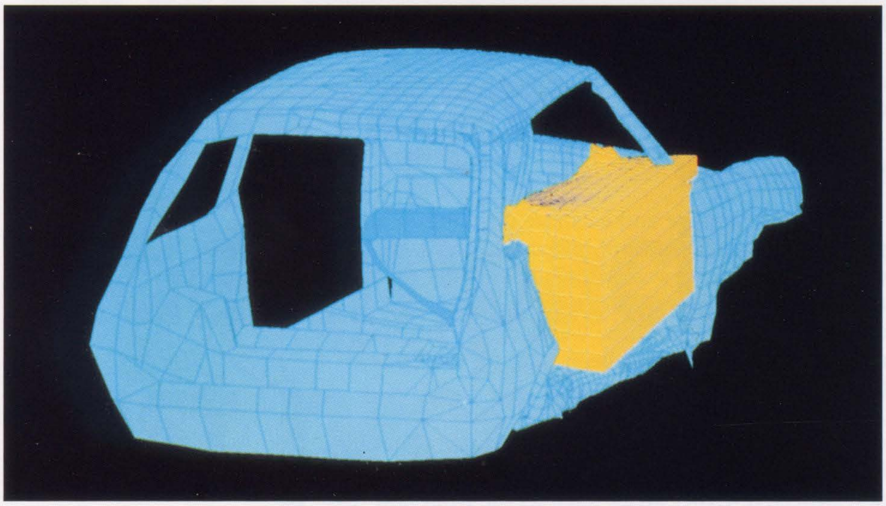
Areas such as the B pillar, rear quarter, and floor pan on the impacted side were meshed in fine detail to represent the significant local deformation expected. Other areas, such as the front and rear ends, where an accurate stiffness is less important to the analysis of side impact, were meshed more coarsely. A correct mass distribution was obtained by using added masses and truss elements to represent components such as the engine, transmission, and rear axle. A rudimentary seat was included in the full model, but this was not intended to be entirely representative; it was included to enhance the visual presentation of the results.

The EEVC barrier model

The moving barrier is supported by a rigid nondeformable cart, with a total mass of 950 ± 20 kg. The barrier is made of six composite foam blocks that can be deformed during contact with the side of the target vehicle. The stiffness of the barrier is designed to be approximately the same as the front of an average European passenger car.

The foam barrier was modeled using brick elements and six elasto-plastic material definitions to simulate the six blocks. The force-deflection and hardening characteristics together with the moduli were obtained by running an actual barrier into a rigid wall. In this compression test, the barrier thickness decreased by 60 percent. Six load cells measured force-displacement characteristics for each block and for the barrier as a whole. Material properties for the finite element model were determined to obtain the correct force-





displacement characteristics. Modeling the cart was unnecessary because it is rigid and no deviation from a straight path has been observed experimentally. The rear of the barrier therefore was declared a rigid body and given an added mass to ensure the correct mass distribution.

When defining the material characteristics, the one-dimensional load curve was assumed to represent accurately the three-dimensional compressive behavior of the foam. This is not strictly correct because the foam is not an elasto-plastic material. However, the assumption was made because the chief concern was the loading phase of the impact.

Results

The analysis was run until 100 ms of simulation had been obtained. This represented approximately 10.5 hours of CPU time on the CRAY X-MP system at Ford Motor Company in Detroit, which can be accessed directly from the United Kingdom. Consequently, the analysis could be run overnight.

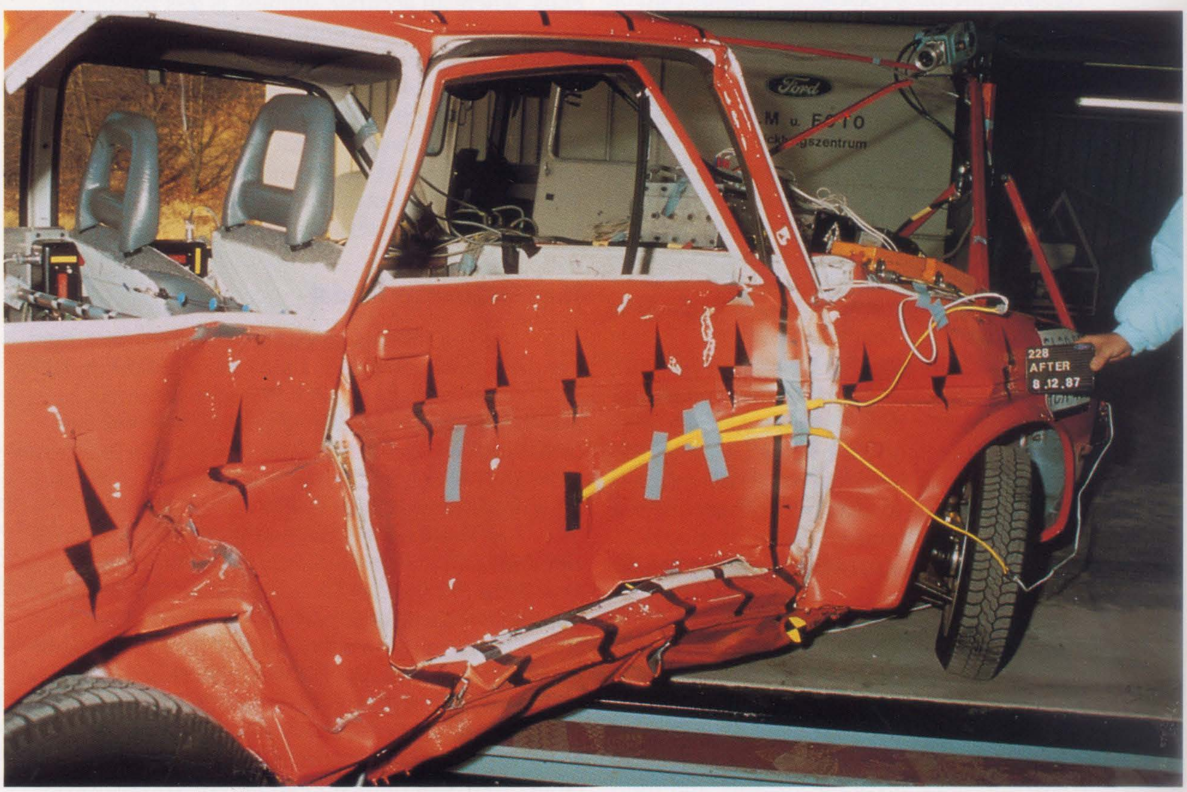
Figures 2 and 3 show the side structure of the vehicle at 0 ms and at 100 ms. Note the severe bending of the B pillar and failures in the upper waist and lower joint areas. The door is shown pushed over the rocker panel with significant intrusion into the passenger compartment. Major deformation also occurred in the floor, rocker panel, and rear seat pan areas.

A full vehicle crash test was conducted to validate the RADIOSS simulation. Every attempt was made to ensure that the circumstances of the test were identical to the computer analysis. Instrumentation was installed to measure displacements and accelerations at certain key positions along the side of the vehicle. This equipment did not allow room for passenger dummies to be included in the test, so 75 kg

Figure 2 (top). Full vehicle model with barrier before analysis.

Figure 3 (middle). Full vehicle model with barrier 100 ms after impact.

Figure 4 (bottom). Vehicle after impact, used for comparison with model.



of ballast were added to the front and rear seat areas to ensure the correct mass distribution.

A simple visual comparison of the simulation model and the test indicates that the model correctly predicted all the major failures in the side structure. Figure 4 clearly shows the failures in the upper, waist, and lower B-pillar joints and waist rail areas. Furthermore, the buckling of the floor pan and rear seat pan, which had been predicted by the simulation, also was observed in the test vehicle.

A close correlation of barrier velocity analysis against the physical test was achieved. The vehicle center-of-gravity velocity comparison is not quite as good, but the correct trend is predicted. The predicted velocity at 100 ms is very similar to that observed in the test, indicating that ignoring friction between the road wheels and the floor is a reasonable assumption.

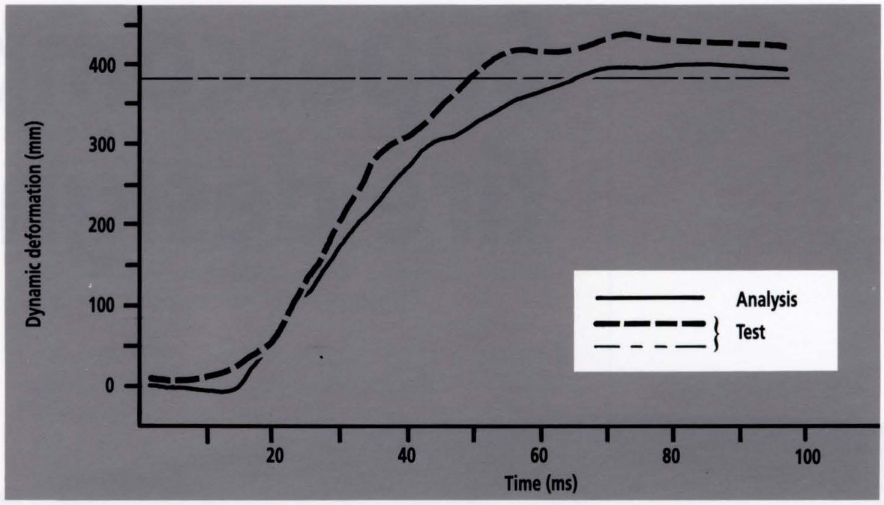
When the test results were analyzed, many were found to be suspect. The spring potentiometers appeared to be unable to track the deformations correctly when very high accelerations were experienced during the first 40 ms of impact. Some results, however, are reasonably accurate, such as the displacement of the door inner panel (Figure 5). Figure 6 shows the displacement of the waist rail in front of the rear occupant. The comparison is reasonably close, although the test instrumentation failed after 60 ms.

The correlation exercise highlighted some important issues concerning the way in which the analysis was compared with the physical test. A comparison of the displacements at single points on the side structure is insufficient, because of the significant local effects. The deformation values must be averaged over a sufficiently large area to reflect the overall value for that region. Furthermore, the modeling of the window-winding mechanisms in the door is important because they affect the transfer of load through the inner and outer panels of the door and consequently influence the load imparted to the passenger.

Conclusions

Our results indicate that the RADIOSS-CRASH finite element code may be able to simulate the proposed European side-impact test procedure.[4] However, further correlation tests are required to validate the techniques more fully. The CPU resource requirement for the analyses is within the capability of the current generation of supercomputers, which provide a practical turnaround time of 24 hours per run. This duration is likely to be reduced as faster machines become available and modeling techniques become more efficient.

The principal advantage of a fully validated numerical model is the ability to further the understanding of the crash phenomenon in a time frame and at a cost that would be impossible to achieve with traditional testing methods. In addition, the effects of design changes could be studied without the repeatability problems of physical tests. It should be noted that the simulation provides many results that are impossible to obtain with conventional crash-test instrumentation. Used in conjunction with concept models, these techniques represent a very effective method of improving the side-impact crashworthiness of vehicles.



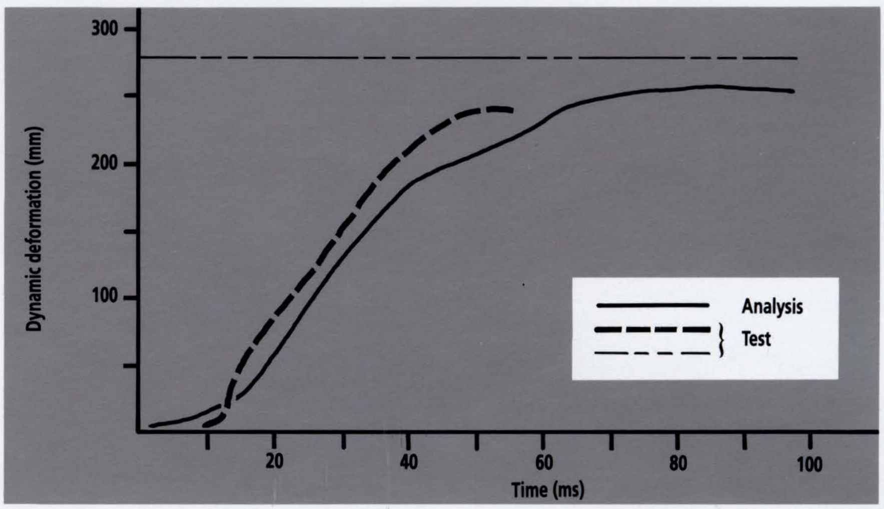


Figure 5 (top). Displacement of a point on the door inner panel. The horizontal line represents the static deformation measured after the test and the curved broken line represents dynamic measurements made during the test. The dynamic deflection is greater than the static deflection because of the elastic energy stored in the vehicle at the 100 ms point.

Figure 6 (bottom). Displacement of a point on the rear quarter panel. The curved broken line ends abruptly because the test instrumentation failed after 60 ms.

Acknowledgment

RADIOSS is a trademark of Mecalog SA, Paris, France.

About the authors

John Stanger, Peter Ng, and Tim Hatch work at Ford Motor Company's Research and Engineering Center, Dunton, United Kingdom. John Stanger is a senior CAE analyst in body engineering. Peter Ng is a senior research engineer working in body research. Tim Hatch is a CAE analyst in body engineering.

Paul Du Bois is an independent consultant in numerical simulation with an emphasis on explicit methods. He previously worked on finite element analyses as an engineer at ESI in West Germany.

References

1. Suthurst, G., P. Ng, and M. Sadeghi, "Inclusion of Crashworthiness in Concept Design," 10th ESV Conference, Oxford, England, 1985.
2. Du Bois, P. A. and J. F. Chedmail, "Automotive Crashworthiness Performance on Supercomputers," SAE International Congress and Exposition, Detroit, Michigan, U.S.A., 1987.
3. Hobbs, C. A. et al, "Development of the European Side Impact Test Procedures and Related Vehicle Improvements," The Eleventh International Technical Conference on Experimental Safety Issues, Washington, U.S.A., 1987.
4. Proposed European Side Impact Test Procedure, Economic Commission for Europe Inland Transport Committee, TRANS/SC1/WP29/GRCS/R.58, 1984.

Supercomputing and tire design at Michelin

A. Guillet
Manufacture Francaise des Pneumatiques Michelin
Clermont-Ferrand, France

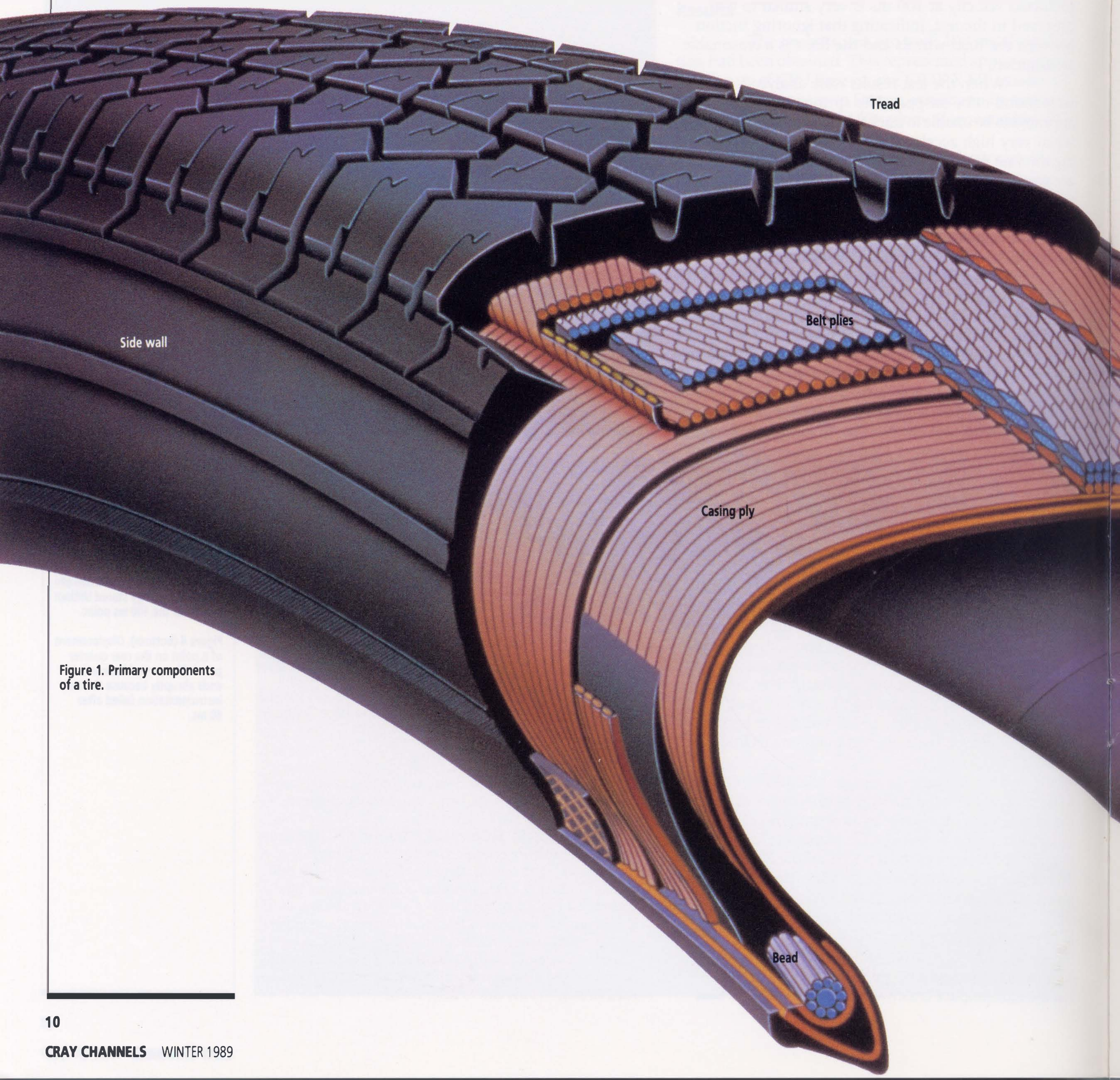


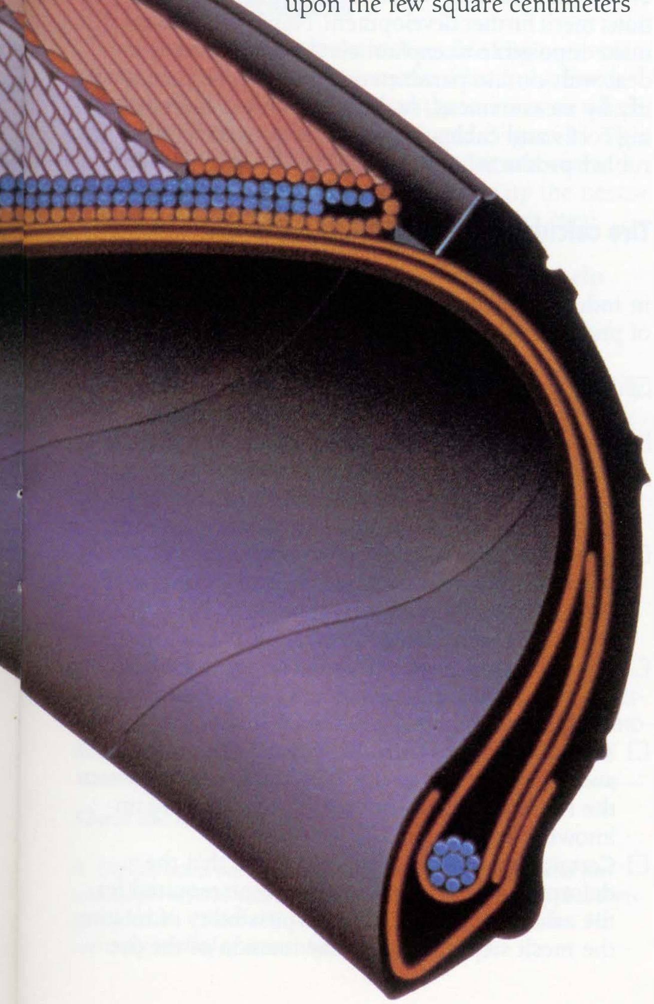
Figure 1. Primary components of a tire.

Although at first glance a tire may seem to be very simple, it is in fact a complex mechanical structure. Its design involves many aspects of engineering and materials science. This complexity makes necessary the use of highly sophisticated design and production techniques. The development of high-speed data processing tools make it possible and profitable to use a physical/mathematical approach to tire design based on two- and three-dimensional models. The CRAY X-MP/14se system at Michelin's research center solves the thousands of nonlinear equations associated with these models much faster and enables designers to analyze more complex models than would be possible with other computers.

The tire: a high technology product

Although all tires have essentially the same purpose, their applications are extremely varied. A tire might be fitted to a bicycle, a car, or a 200-ton truck. The tire provides contact between the vehicle and the ground under widely differing conditions, often with conflicting roles to play to ensure safety, comfort, and good performance. A tire must carry loads, transmit torques and forces, roll, guide, absorb shocks, and perform other functions — and endure.

The load for a car tire is on average 50 times its own weight, a ratio that increases to 270 times in the case of an aircraft tire. Safe and efficient operation of the vehicle — effective braking, accelerating, and cornering — depend upon the few square centimeters of tire that contact the ground. And this surface is changing constantly; at 130 kilometers per hour, a given tread block plays its part for only 2/1000 of a second during each revolution. When the driver brakes hard, the surface temperature can rise to over 200°C. In spite of these conditions, and in the face of curb scrubbing, skids, potholes, and other sources of wear, the tire has to fulfill its role faithfully, forgotten in the background, without compromising safety.



A tire comprises various elements, each of which has a precise function (Figure 1):

- ☐ The casing ply (for a passenger tire) is a composite product made of textile cords that might be made of rayon, polyester, or nylon, encased in rubber. Its job is to transmit loads while contributing to the rigidity of the crown area and providing side-wall flexibility.
- ☐ The belt plies are made of steel or aramid cords in rubber. They determine the shape of the crown and play a major role in tire behavior.
- ☐ The tread contributes to road holding and grip and must last many thousands of kilometers.
- ☐ The side walls provide flexibility and comfort and their rubber must withstand scrapes against curbs and the adverse effects of ozone.
- ☐ The beads, which comprise bead wires and various filler and reinforcing rubbers, are used to fix the tire to the wheel or rim and render the whole assembly airtight.

The role of the tire designer

The structure of a new tire is defined during its design. The designers establish the geometry and materials that best satisfy the desired performance criteria at an acceptable cost. This process is extremely complicated. It is in fact relatively easy to give an optimum answer to each parameter taken separately — a Formula One or rally driver, for example, can have tires suited exactly to such surface conditions as a dry track, rain, snow, or other conditions. Optimum performance for snow will depend on transversal sipes; for a wet road surface, on deep tread blocks; and for dry surfaces, on the smooth treads of "slick" tires.

Ordinary car tires, on the other hand, have to reconcile a number of different requirements: traction on wet and dry surfaces, precise handling, endurance, hard wear, comfort, safety, and other considerations. If traction is improved, rolling resistance and wear rate may increase. If the pressure is increased, stiffness is increased to the detriment of comfort. Thus, each range of tires is a subtle balance between conflicting requirements. The tire designer has to obtain optimum answers for the different design parameters, such as the type of architecture, materials, geometry, tread design, sipes, and so on, to obtain the best compromise possible.

How the designer proceeds

In the past, trials using road simulators, test tracks, or the open road were the only methods available to the designer to test the combinations of parameters selected. The data collected, together with

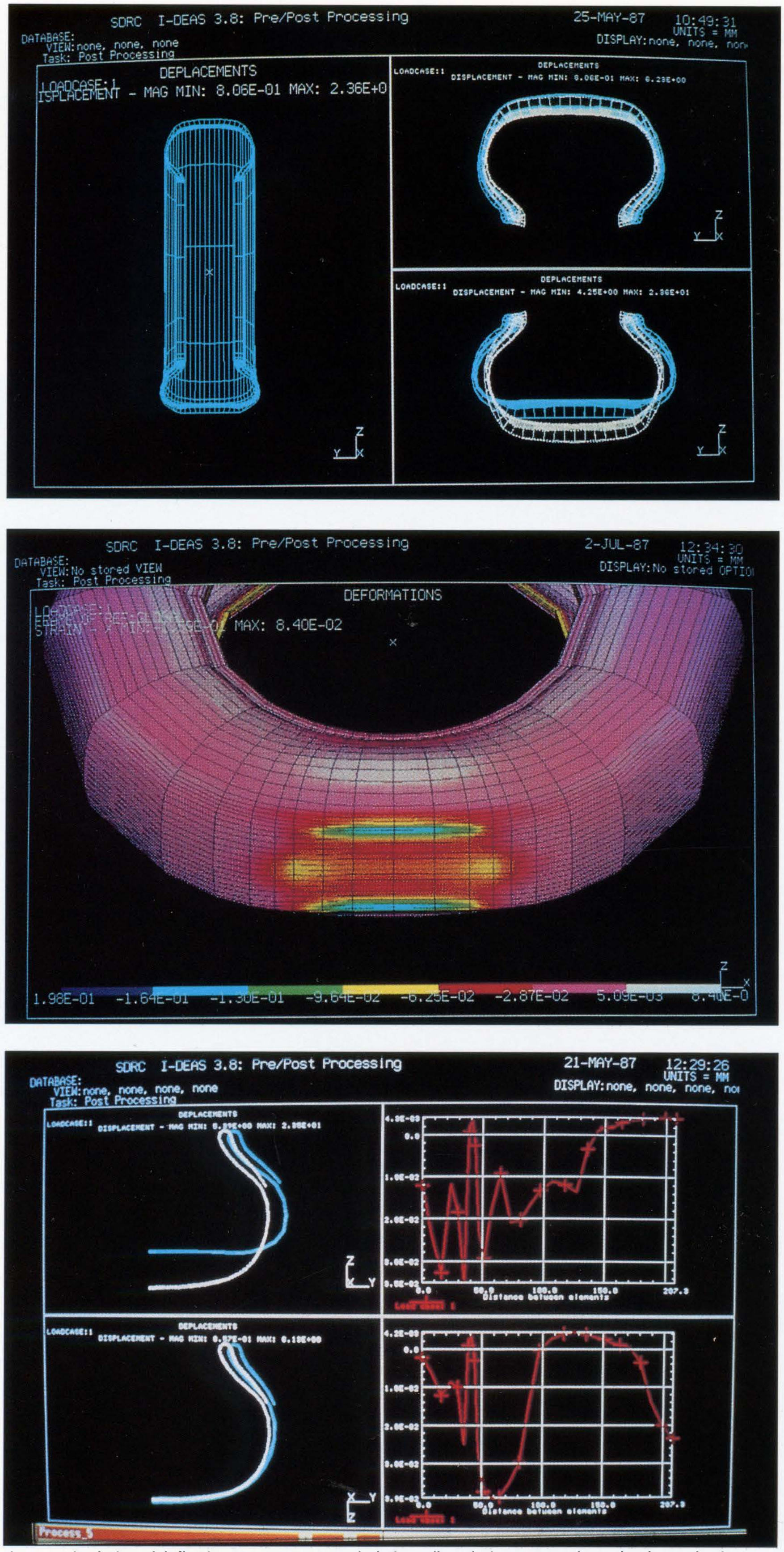


Figure 2. Simulation of deflection. Supercomputer calculations allow designers to evaluate the shape of a tire under a load and the positions occupied by the various components (top) as well as the ground contact pressures (middle) and the state of tension in the reinforcements with the tire inflated and under a load (bottom). This last consideration applies particularly to the tension cycles imposed on the cables during rotation.

physical measurements made during behavior trials of the new product under various conditions, allowed designers to choose the best solutions and obtain the required compromise. However, this long and expensive procedure is not the answer to present-day design requirements. Market demands now must be met very rapidly, and a rising number of increasingly difficult problems have to be addressed due to product diversity and increasing customer demands. By improving the designers' speed and efficiency, computers and numerical modeling not only help meet market demands, but also help restrain the growth of prototype development costs.

The most general numerical models attempt to show the response of a tire to steering and camber angles and to obstacle envelopment. These models are based upon functional theories of tire behavior and contain only a few parameters. They typically include considerations such as vertical rigidity, transversal flexibility of the crown, and other general considerations. These parameters can be measured on real tires and are simple enough to be combined with vehicle models to make possible behavioral studies of complete systems.

The sudden increase in computational power provided by supercomputers enables designers to build and execute models that can predict the response to given load conditions. These models, which are based on the definition of the tire's geometry and materials, use the finite element method. They enable designers to predict the effects of a change in geometry or material and help them to decide which combinations merit further development. Numerical models also make it possible to explain certain phenomena and deal with certain parameters that are difficult to quantify by measurement, such as the tensions in reinforcing cords and cables or stresses within the various rubber products.

Tire calculations using finite elements

The finite element method is used widely in industry, but when it is applied to tires a number of problems arise:

- ☐ The theory of linear elasticity cannot be used because a tire undergoes considerable deformation.
- ☐ The materials used in a tire behave very nonlinearly through a range of temperatures and rotational speeds. This nonlinearity must be taken into account in the calculations and integrated into the model.
- ☐ Numerical instabilities can result because the composite materials of the tire have ingredients that differ drastically in rigidity (by a factor of 10,000 in the case of rubber and steel).
- ☐ Incompressible elements must be used because the volume of a rubber object does not change when the object's shape is deformed.
- ☐ The model must take into account the contact areas between the ground and the tire and between the rim and the tire, but these quantities are unknown initially.
- ☐ Certain types of calculations require that the deformed structure rotates. The time required for the calculations precludes the possibility of rotating the mesh step by step, so the rotation of the tire

and the contact area kinematics have to be taken into account during the mechanical formulation of the problem.

☐ Forces caused by internal air pressure are applied to the deformed structure and thus have to follow the structure (remain perpendicular) as it moves.

We therefore have developed a method of calculation using finite elements that takes into account these characteristics and enables us to simulate the response of a tire to static and dynamic loads. These simulations lead to the creation of three-dimensional models with between 10,000 and 60,000 degrees of freedom; the simulations require the solution of nonlinear systems with as many equations. The CRAY X-MP/14se system requires from a few minutes to a few hours to solve these problems.

The use of such models requires the preparation of an enormous file of data and the analysis of numerous results, including displacement, strains, stresses, and pressure fields. To assist in these operations, preprocessor software is available at a network of workstations that provide semi-automatic mesh and input data generation from an existing CAD-based definition of the tire. Similarly, postprocessor programs are available for visualization of the state of stress of the product, the state of tension in the cords of the carcass ply, the shape of the surface contact area between tire and ground, the distribution of contact pressures, and other responses of the tire.

Results and conclusions

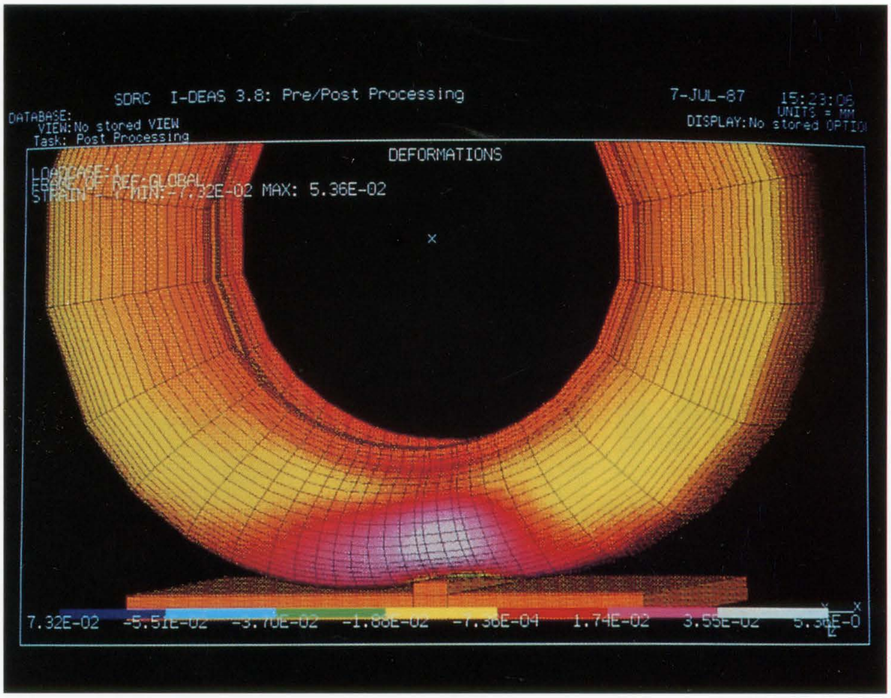
Structural calculations enable engineers to predict the shape of the tire under a load, the forces induced in the contact area, and the stresses to which the various components are subjected (Figure 2). This capability enables designers to consider and evaluate new combinations without having to set up the necessary tooling for their production, sometimes even before looking into their feasibility.

Computer simulation also provides help in analyzing existing tires. Measuring the stresses or distortions inside a tire is very difficult, and despite the progress made in this field, a large number of measurements are still impossible to carry out directly. Many of these measurements can be calculated using models, which are thus used in addition to other test procedures for physical analysis of tires (Figures 3 and 4). The availability of a supercomputer means that even more sophisticated models can be considered, which will better represent viscoelasticity, friction, transient dynamics, and other physical phenomena.

Structure calculations remain, however, only one step in the tire design process. They cannot give a direct appreciation of performance, which often is subjective according to the driver. Forecasting tire performance is a complicated process that must take into account the entire vehicle as well as the driver.

About the author

A. Guillet has a background in applied mathematics and has worked for several years on numerical software and applications at Michelin's Centre d'Etudes et de Recherche de Ladoux. He currently is chief of a scientific computing group.



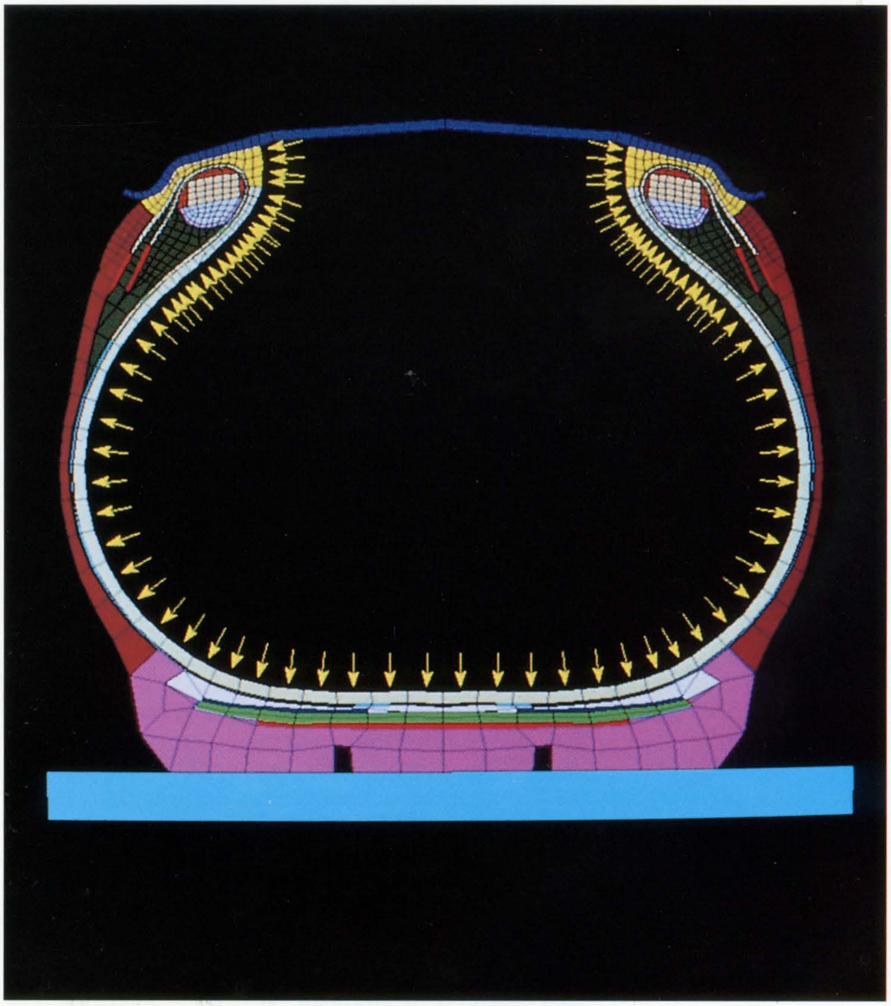


Figure 3 (top). Simulation of tire rolling over an obstacle, showing the strains around the obstacle. This type of analysis addresses the tire's ability to absorb an obstacle and is conducted as part of comfort studies.

Figure 4 (bottom). Close analysis of stresses in a cross section of the contact area. This type of analysis results in very fine meshes (56,500 nonlinear equations). Solutions typically are available from the Cray system within a few hours.

Simulations in acoustics and vibrations

Bruno Hazet and Thierry Lambert
P.S.A. Studies and Researches (Peugeot-Citröen)
La Garenne-Colombes, France

Predicting the acoustic comfort of car passengers is a principal design consideration for engineers at P.S.A. Studies and Researches (Peugeot-Citröen). An automobile's engine, exhaust pipe, and air filter produce varied noises and vibrations, and their effects are amplified by the car body, which propagates elastic (structure) waves and reflects acoustic (fluid) waves. These elasto-acoustic interactions are the primary source of noise propagation in the car passenger compartment.

Automotive engineers use two main computational approaches to study the complicated vibrational and acoustical behavior of automobiles. The classical technique involves using the finite element method. The second, presented in this paper, is more recent and less widespread and rests on the boundary finite element method. Regardless of the method chosen, the vibro-acoustic analysis of complex automobile structures involves several thousand degrees of freedom for up to several hundred frequencies. Hence, using our CRAY X-MP/14 computer system was absolutely necessary for accurate and timely simulation.

First, we will present the two methods used to describe the acoustic medium. Then, we will describe the two main classes of application: the prediction of the acoustic field radiated or diffracted by a structure of arbitrary shape, and the simulation of the vibro-acoustic reaction of these structures. Finally, we will approach the problems met, linked with the postprocessing of the results and their correlation to the experiment.

Two methods of analysis

When a structure vibrates in the presence of a fluid, it produces interaction between the elastic and acoustic waves. The equations of motion for the structure and fluid must be solved simultaneously, and are linked to the level of the interaction areas by two coupling conditions. In other words, if the fluid is assumed to be perfect, the continuity of the normal component of displacements and stresses occurs at the passage through these areas.

The finite element and the boundary finite element methods are used to solve the three-dimensional problems of radiation and fluid-structure coupling. As the names imply, these methods describe the structure by finite elements (thin shells); only the description of the fluid differs.

The finite element method describes the fluid by solid finite elements. This creates a symmetric algebraic system that couples the displacement field of the structure with the acoustic field in the volume occupied by the fluid. The first method generally is restricted to the study of internal problems in which the fluid is contained in the structure. Indeed, when the acoustic medium is spread out or unlimited, extremely long algebraic systems result, which complicates the numerical set and increases the cost of calculation. One advantage of the finite element method is that the calculated matrices are independent of the frequency. Moreover, the description of the fluid by solid elements allows the calculation of acoustic modes (cavity modes), but requires the building of two different meshes for the structure and fluid.

The boundary finite element method applies an integral representation for the pressure in the fluid that uses the trace of the acoustic field on the structure. Since only the boundary (surface mesh) is represented, this method has a clear advantage — it eliminates the need to describe the domain occupied by the fluid. A direct method involves solving the system by a method of collocation by points. This produces asymmetric algebraic systems and complex problems of singularities inherent to integral equations. The originality of the technique[1,2] lies in its use of a variational formulation. We look for a displacement field and a pressure or speed jump field at the fluid-structure interface, which makes the function of the coupled problem stationary.

Using boundary finite elements to describe the variational formulation leads to a small, symmetric, and fully algebraic system. However, the double integrations needed in the calculation of integral operators significantly increase the computing time. Moreover, the fluid matrices depend on frequency, which requires a new calculation when a frequency spectrum is analyzed. Because the structure and fluid meshes are of the same type (surface), they can correspond with one another — a significant advantage at the time of the numerical work. Finally, the boundary finite element method can solve either internal or external problems.

The two methods are complementary rather than contradictory. Their efficiency depends on the geometrical nature of the problem (ratio between the volume occupied by the fluid and the area of the structure), and on the acoustic domain studied (internal or external).

Software

The numerical work of the variational formalism by integral equations presented above is realized

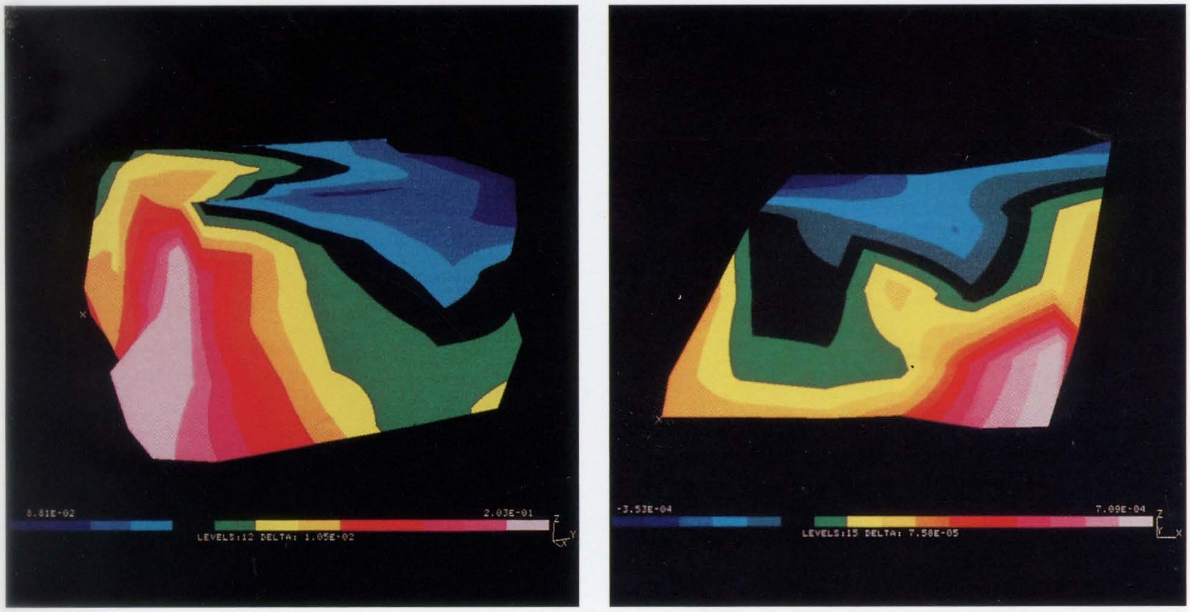


Figure 1. Exhaust pipe noise transmission in a car boot. In the left image, pressures are measured near the boot walls. The exhaust pipe is on the left side of the boot near the wheel housing, as indicated by the red area. The right image shows the normal velocities that have been computed for the left panel of the boot (wheel housing side).

by a boundary finite element method that calculates and assembles the integral equation operators appearing in the variational formulation. The RAYON code, which is a product of the Straco Corporation, is a modular code that executes a group of operations identical to those of a finite element code, such as

- ☐ Description of the geometry of the structure boundary (nodes coordinates, elements connections)
- ☐ Description of the physical constants of the problem (sound speed, density, frequency, etc.)
- ☐ Description of the physical properties (material impedance, number of degrees of freedom at nodes, etc.)
- ☐ Description of the acoustic sources
- ☐ Calculation of the elementary matric operators and assembly
- ☐ Solution of the symmetric linear matric system
- ☐ Calculation of the physical variables of the problem

The computation can be realized for one frequency or spectrum. All the variables are made adimensional to avoid the numerical problems due to an important ratio between the fluid (air) and structural variables.

The physical variables such as pressures, normal velocities, and acoustic intensities can be computed on the structure boundary and at any point of the fluid, internal or external to the structure studied. This computation can be made independently of the assembly and solution; several groups of points can be computed as the analysis is in progress in independent executions with restarts. This confers significant flexibility to the radiation and diffraction analyses. For problems of fluid-structure interaction, the RAYON code recovers the stiffness, mass matrices, and the force vector of the structure from a finite element code. The RAYON code then assembles those factors with the integral equation operators of the fluid and the coupling matrix of the displacement and acoustic fields.

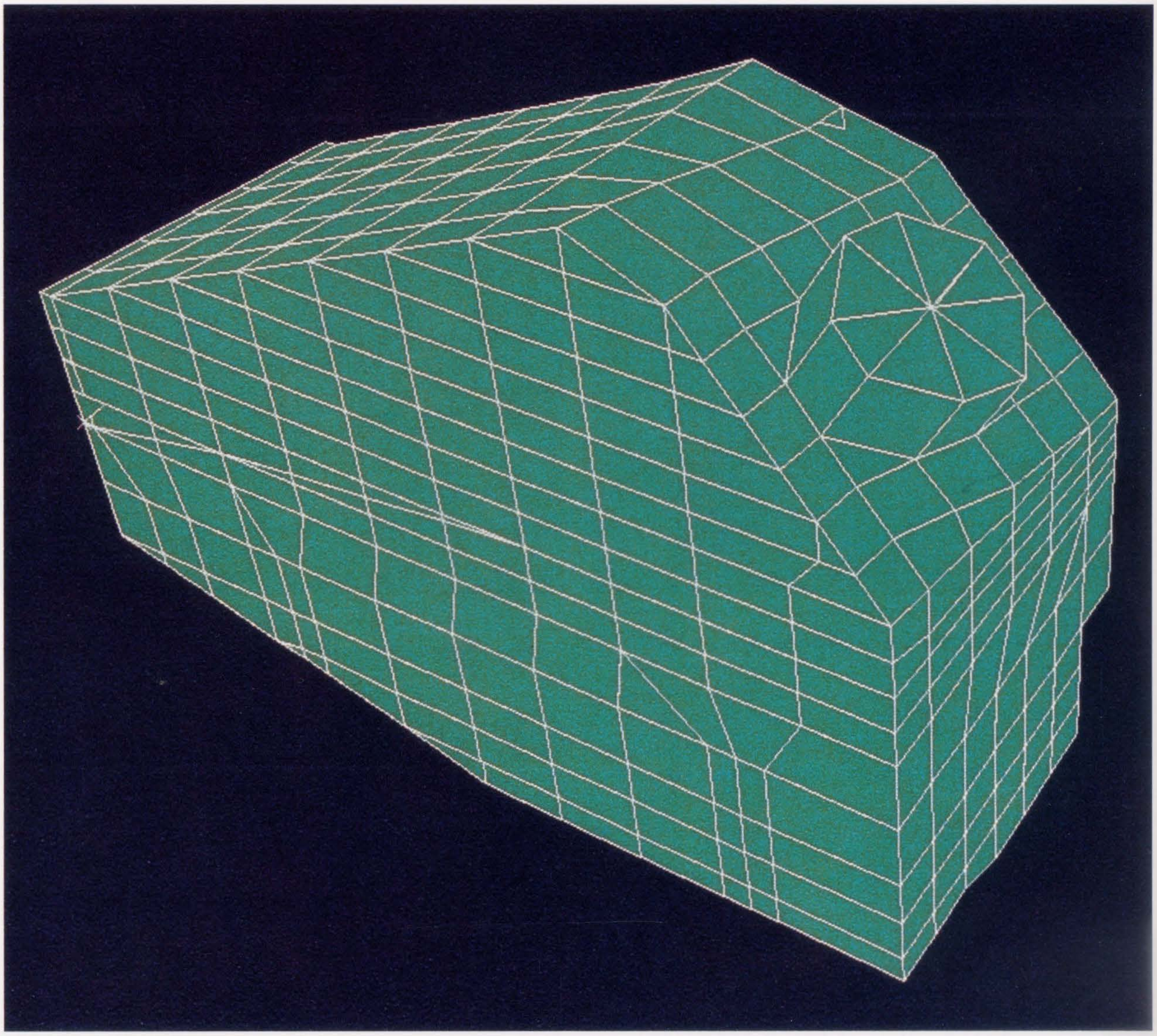
Computations of sound field

Calculating the sound field radiated or diffracted by arbitrarily shaped structures expands the acoustic analysis. From the measurement of the pressures on the structure surface, the pressures in the internal or external domain are calculated without complementary experiments. The number of analysis points is unlimited theoretically, but is limited in reality by computer memory and the cost of calculation. The acoustic sources can be of several kinds: punctual sources (spherical waves), plane waves, and distributed sources. In distributed sources, the measurement is realized either by microphones or accelerometers. Pressures or normal velocities obtained from these experiments are enforced at the center of the elements. Each element of the measure mesh must be an acoustic source. This involves a good correlation between the experimental phase and the phase of the constitution of the calculation mesh. The two models generally are identical. The nodes then correspond to the measurement points, and the acoustic variables are interpolated over the elements.

According to the kind of parts studied, the degree of the mesh acuteness depends on the length of the acoustic or elastic wave. In each case, each period must be described by at least four elements. This method is used in many car manufacturing applications, including studies of radiation (from road excitation or from components such as the engine, air filter, or exhaust pipe) and studies of transmission (including the passenger compartment and car boot).

For example, P.S.A. engineers are studying noise transmission and propagation with the boundary finite element method on the CRAY X-MP/14 computer system. Pressures are measured in the nearby car-boot walls and are enforced as acoustic sources. Then the code calculates the normal velocities to the partitions and the radiated field in the air domain (pressures, velocities, intensities) as illustrated in Figure 1. Also, we can measure the normal velocities, compute the pressure on the partitions, or mix the two types of measures.

Figure 2. Mesh of the air filter, which involves about 600 nodes and elements.



Using a supercomputer is essential for modeling fluid-structure interaction, especially when a frequency spectrum is studied.

To obtain the variables on the boundaries, a model of about 100 elements requires approximately one minute of time on a CRAY X-MP/14 computer system — the time increasing with the number of values desired in the air domain. Of course, this manipulation can be applied to a frequency range.

This second example illustrates a first approach of the vibro-acoustic analysis of structures, which will be discussed further in the next section. Following a modal analysis, the displacement field obtained at a natural frequency is enforced on the fluid boundaries as acoustic sources of normal velocity. The code computes the pressures on the structure walls and the radiated field. This example represents a simplified geometry of an air filter (Figures 2-5). The model is composed of approximately 600 elements. About 20 minutes of computing time are required for one frequency, without calculation points in the fluid domain. The benefit of using a supercomputer for these problems becomes very significant when calculating large frequency ranges.

Simulation of the vibro-acoustic behavior of a vehicle part

The above example concerns the study of sound transmission and radiation by the partitions of a simplified air filter (Figure 2) excited by the air flow that is modeled by a punctual source on a level with the opening.[1] The structure is clamped on certain points of its lowest part. A first calculation of modal analysis allows us to characterize the first natural frequencies and mode shapes of the structure on the border where the acoustic analysis will be realized (Figure 3).

The mesh is the same for interaction, fluid, and structure elements. This largely simplifies model preparation. The mesh is composed of about 600 nodes and elements. We use linear three- and four-noded boundary finite elements for the air, with one unknown quantity per node, which is the pressure jump. The structure is made with three- and four-noded thin shell elements, with six degrees of freedom per node. Thus, the model treated has 4,200 degrees of freedom. The code computes the pressures on the structure partitions as well as the radiated field. As we noted earlier, the algebraic system found is symmetric and small, but full for the fluid component.

For this study, the memory size of the CRAY X-MP/14 computer system (four million words) is greater than needed. For larger structures, such as an automobile's passenger compartment, the model must be treated by substructuring techniques that complicate the numerical work. Computing time is about half an hour for one frequency on the CRAY X-MP/14 computer system, which would be equivalent to several hours on a scalar computer. Using a supercomputer is essential for modeling fluid-structure interaction, especially when a frequency spectrum is studied.

Postprocessing

When analyzing the results, two main obstacles may arise. One difficulty is comparing model data with physical experimentation. Another difficulty is analyzing the volume of output for a model comprising several thousand elements and hundreds of computing points (such as the passenger compartment) on a large frequency range.

The first obstacle concerns the problem of compatibility between measurement computing results. A measurement point does not always correspond to a mesh node. Moreover, it becomes necessary to define and to borrow common graphic models for computation and experiment. These two phases can be executed on the same computer by one person.

The second difficulty may be overcome with the use of postprocessing tools that provide three-dimensional animation, quick display of contours in volume sections, frequency-by-frequency (spectrum) analysis, and selection criteria of physical values variable with frequency. To avoid reducing the result analysis to a lower rank (hundreds of drawings might be necessary but could not be done), we will conduct studies in the use of databases and dynamic postprocessing.

Conclusion

To predict car passenger acoustic comfort, studies of vehicle vibrational and acoustical behavior cannot be dissociated. One main challenge is characterizing and simulating the relationship of vibratory and acoustic sources, which are very difficult to identify exactly on a real vehicle. Another challenge is the treatment of physical and geometric singularities, such as imperfect links (welding joints) or damping. These complications confirm the necessity of supercomputer analysis and the need for close interaction between computation and experimentation.

About the authors

Bruno Hazet and Thierry Lambert are scientific research engineers at P.S.A. Studies and Research. Hazet joined P.S.A. in 1985, after graduating from Ecole Nationale Supérieure des Techniques Avancées in 1984. Lambert earned a Ph.D. degree in acoustics and vibrations from the University of Technology-Compiègne.

References

1. Hamdi, M. A., "Formulation variationnelle par equations integrales pour le calcul de champs acoustiques lineaires proches et lointains," These de Doctorat en Sciences, Universite de Technologie de Compiegne, June 23, 1982.
2. Hamdi, M. S., and T. Lambert, "Simulations par elements finis de frontiere des problemes d'environnement vibroacoustiques," *Colloque International des XIemes Journees Scientifiques et Techniques organise par l'Association pour le developpement des Sciences et Techniques de l'Environnement Paris-La Vilette*, March 15-18, 1988.

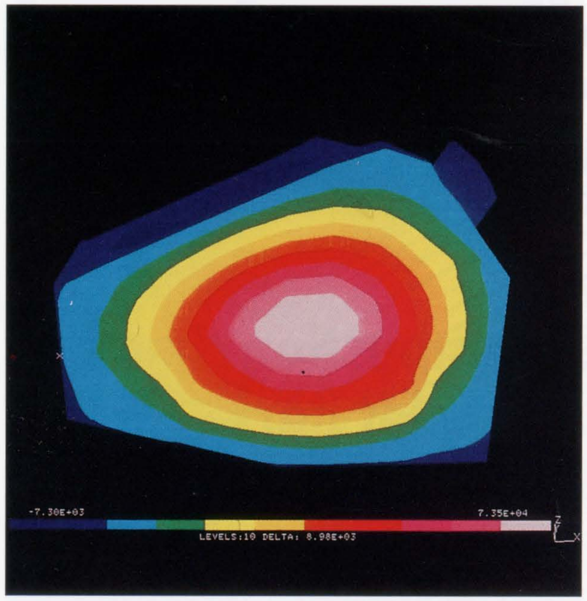


Figure 3. First mode shape for the air filters. Normal velocity repartition for the structural mode is shown.

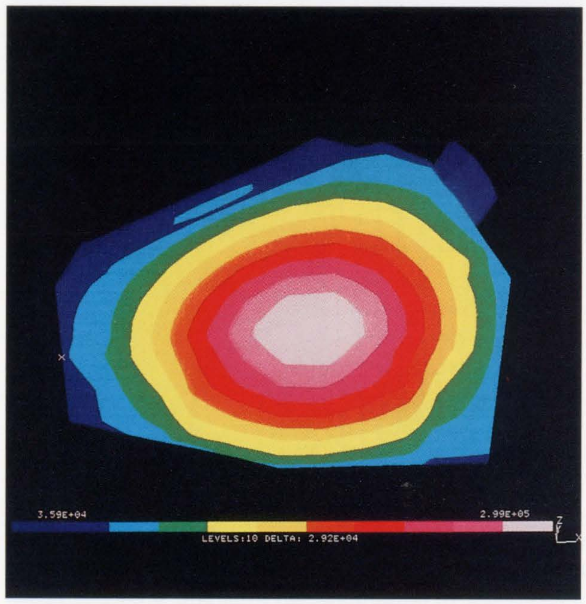


Figure 4. Sound field in the internal face of the air filter due to the normal velocities of the first structural mode.

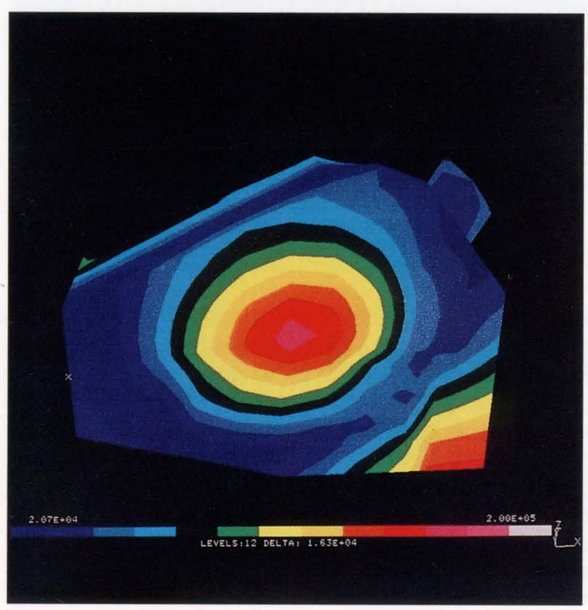


Figure 5. Sound field on the external face of the air filter due to the normal velocities of the first structural mode.

Multidimensional flow modeling

A combustion chamber and inlet port design guide

Reza Taghavi and Alain Dupont
R.N.U. Renault, Rueil Malmaison, France

Since Cray systems first became available to Renault engineers, computational fluid dynamics and combustion applications have demanded the greatest use of computational time and power. Renault engineers are using a CRAY X-MP/14 system to explore the possibilities and limitations of computational fluid dynamics codes for engine design. One specific application is the study of flow in a port and chamber under motoring conditions.[1] This study focuses on port and chamber flow because the phenomenon significantly influences the prevailing aerodynamic conditions at ignition and consequently affects combustion.

Several investigators have computed the steady flow in inlet ports and have compared the resulting velocity distributions in the valve passage with experiments.[2] Multidimensional flow simulations in complex combustion chambers under motoring conditions have been performed to a much larger extent.[3]

A major interest has been the computation of flow inside combustion chambers using experimentally determined boundary conditions at the valve. Simulations have been performed to study the effect of inlet ports in general[4] or multivalve intakes[5] on the development of in-cylinder flow. In these studies, satisfactory bulk swirls or horizontal plane velocity profiles were obtained in the case of helical ports.[2]

Multidimensional modeling of the flow within an inlet port and a combustion chamber, separated by a moving valve, addresses a number of problems that are not addressed when simple engine geometries or experimentally obtained valve velocity profiles are used.

The computer code

The computations were performed with the KIVA code, which was developed at the Los Alamos National Laboratory.[6] The code is designed to model internal combustion engines and uses cell and vertex flags to take into account computational domains composed of arbitrary hexahedrons. It uses an explicit scheme and acoustic subcycling to overcome Courant criterion time step limitations. The code has been modified to handle combustion chambers with realistic and complex shapes in the cylinder head and piston bowl.[2] Further modifications needed for this study are reported below.

Finite difference mesh generation for the combustion chamber/port assembly

Finite difference grid generation for industrial applications is a tedious task. Most of today's preprocessors are designed for finite element computations, for which the designer needs only to define the outer boundaries of an object. The mesh generator then fills the volume with the required elements and numbers them. The finite difference method requires that each cell layer be built carefully and that attention be paid to the numbering of cells and vertices.

The various parts of the computational grid used in this study were built using the preprocessors GIBI and SUPERTAB. The final grid was assembled according to the requirements of KIVA with an in-house renumbering procedure. Figure 1 shows a wire-frame representation of the final mesh with the position of the valve at its minimum and maximum lifts.

The grid at top dead center (TDC) is provided to the computer program. One of the features of the KIVA code is that as the piston moves downward, additional cell layers are added progressively between the piston bowl and the cylinder head to ensure a constant mesh density in the space between these components. In this example, at TDC the mesh contains 13,520 cells, about half of which are inactive. Active cells are added layer by layer so that at bottom dead center (BDC) the mesh contains 19,760 cells.

Grid handling and code modification

A moving valve is simulated by translating all cell vertices pertaining to the valve as prescribed by the lift function. The topology of the computational domain is not altered during valve movement because the connectivity of each computational cell is maintained in this operation.

A new iteration procedure is used, which when applied to an arbitrary but topologically accurate grid, yields a mesh that is much more adapted to the present flow computation and is less likely to yield negative volume cells than the common Laplace equation iteration procedure.[1]

The application of this procedure at every timestep of the simulation allows the computation to proceed normally as the main code lifts the valve according to the prescribed lift function. Furthermore, this grid-conditioning procedure can be used in any other circumstances in which the boundaries of an *I, J, K*-structured domain are severely distorted and deformed, or where negative volume cells are to be avoided.

Model boundary and initial conditions

A one-dimensional gas dynamics model of the complete single cylinder engine based on the method of characteristics is used to determine initial and time-varying boundary conditions. This model uses the geometrical characteristics of the intake and exhaust sides of the engine, as well as prescribed valve lift functions and a simplified combustion model (using Wiebe's law). It solves for pressure, mass flow rate, temperature, and various mass fractions at different points of the engine. A flat velocity profile is assumed at the upstream section of the inlet port and the initial velocity field at TDC is assumed to be quiescent.

Initially, the entire combustion chamber and intake port are assumed to be filled with residual burned gases resulting from previous cycles. The port is assumed to be filled with burned gas to simulate burned gas return in the intake due to an intake valve opening 20° crank angle prior to the induction TDC. Finally, it is assumed that the initial gas content of the combustion chamber equals 17 percent of the mass of gas that would be contained in the cylinder at BDC under atmospheric pressure.

Engine operating conditions and flow simulation

Due to a limitation in the minimum allowable cell size, valve movement cannot be simulated all the way to closure. The minimum valve lift for this simulation is 2.3 mm. Valve closure is enabled by assigning solid cell flags to all the cells in the valve clearance area. The maximum valve opening is limited to 5.6 mm due to a limitation whereby the valve cannot descend below the piston's maximum vertical position.

With these conditions, the design valve lift function is truncated. The original valve opening occurs at 20° crank angle before top dead center (BTDC) and closure occurs at 240° crank angle after top dead center (ATDC). Due to truncation, the actual opening occurs at 0° and closure at 220° crank angle ATDC.

The combustion chamber is composed of two parts: a piston bowl 4 mm deep and an ellipsoidal cavity in the cylinder head 6 mm deep. The spark plug is located on the edge of the cavity. The intake port is of a nonswirling, conventional type. The inlet valve is 40 mm in diameter with a seat angle of 45°. The valve stem in the inlet port also is modeled. The compression ratio is equal to 9.5. Simulation starts at induction TDC and proceeds down to BDC. The engine is run at 2000 rpm.

Discussion of results

At lower valve lifts (for example, −352.5° crank angle with 0° taken at compression TDC in all velocity and concentration plots) the velocity profile

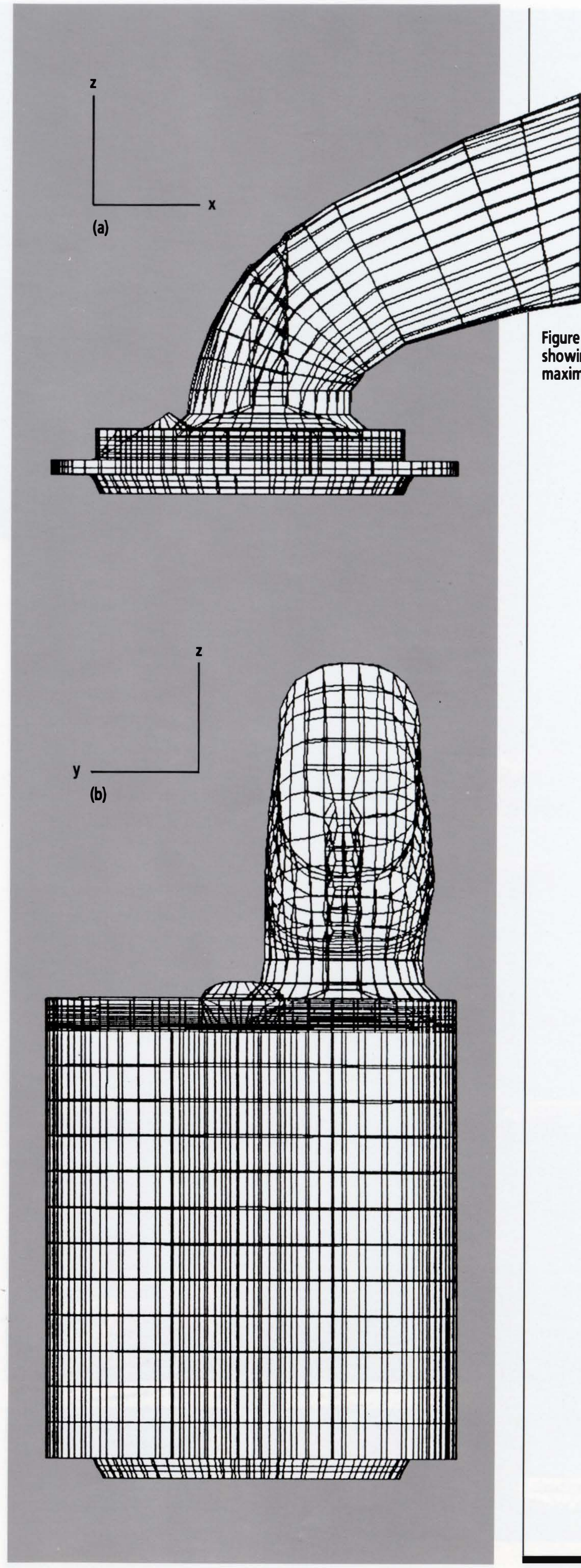


Figure 1. Port/cylinder assembly showing the minimum (top) and maximum (bottom) valve lift.

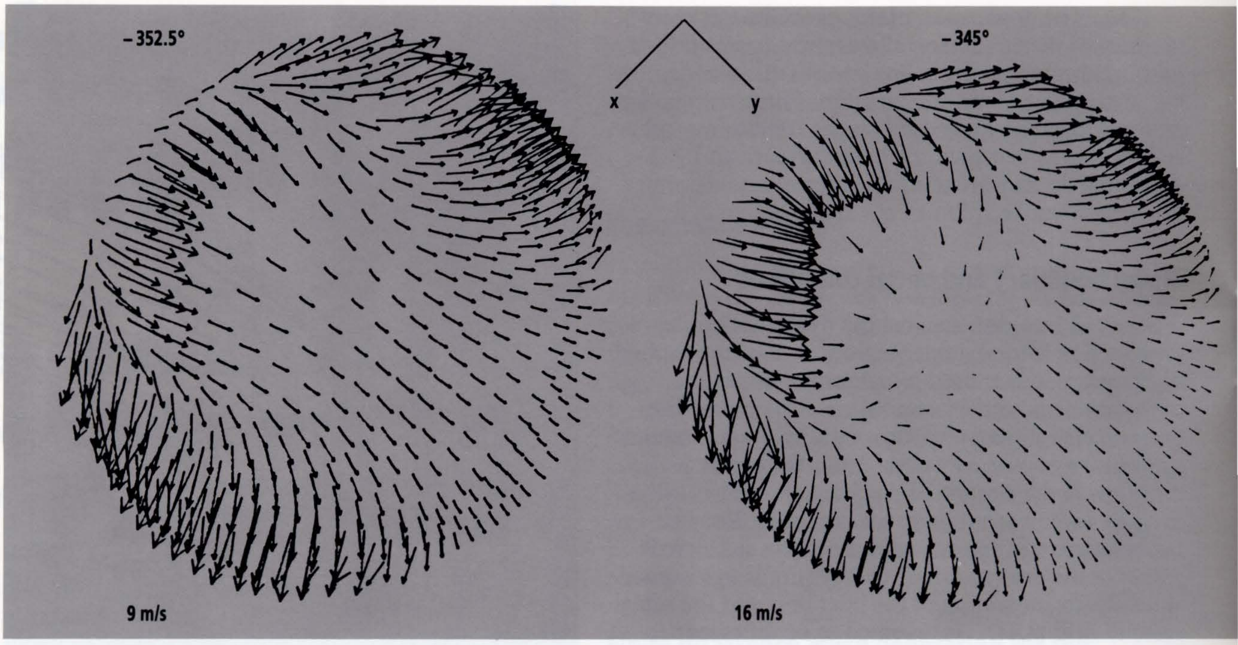


Figure 2. Velocity field 1 mm below the head gasket plane.

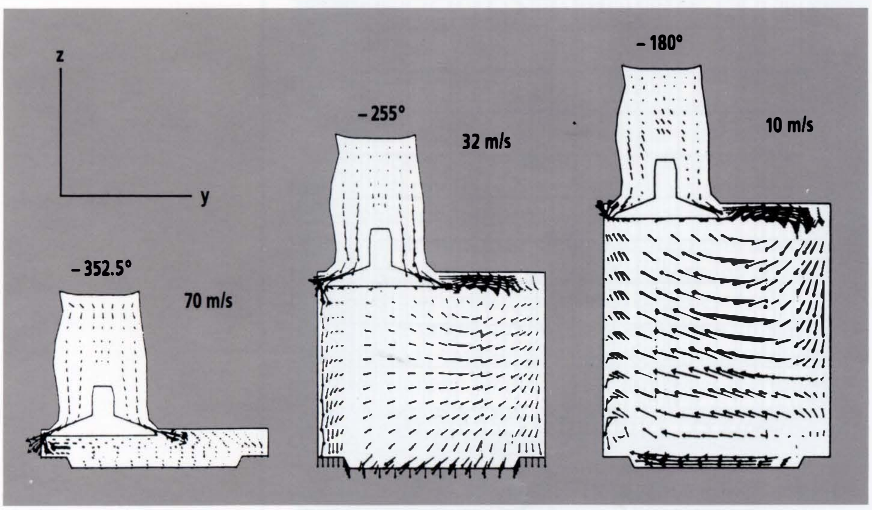
is less affected by the presence of the wall than at higher lifts. Soon after, as the valve lowers, wall effects become predominant and the flow is diverted to regions of the valve circumference that are farther from the wall.

The proximity of the wall creates an axis of symmetry along the longer axis of the cylinder head cavity that seems to supersede effects due to the inlet port's incoming flow direction. Figure 2 shows the flow field on a plane located 1 mm below the cylinder head gasket plane. The wall proximity effect is obvious at -345° crank angle. The jet issuing from the valve bounces back into the cylinder over at least the first 15° of intake. Later, two counter-rotating eddies are established which, at BDC, occupy more than half the view plane. Note that the axis of symmetry persists at least up to the BDC and seems to erase other effects such as the presence of the spark plug well or other geometric features of the cylinder head cavity.

Figure 3 represents the flow field on two vertical planes containing the valve axis. The projection plane used in Figure 3 actually passes through the center of the chamber (along the aforementioned axis of symmetry). At BDC the flow field is reversed and some gas returns into the inlet port.

Figure 3 shows clearly that, due to wall proximity, the flow field is essentially from left to right in the cylinder head cavity. The return flow occurs in the entire cylinder with the establishment of a single large eddy rotating clockwise. At BDC, the eddy persists but sends the flow upward toward the valve. Figure 4 illustrates the early stages of the intake process. A fresh air front advances through the inlet port, through the valve passage outlet, and along the walls of the combustion chamber. Note that in the plane containing the axis of symmetry, wall blockage causes the front to descend, while on the opposite side the front moves horizontally. The fresh air front reaches the region immediately below the valve last.

Figure 3. Velocity fields in an *x* = constant plane containing the valve axis.

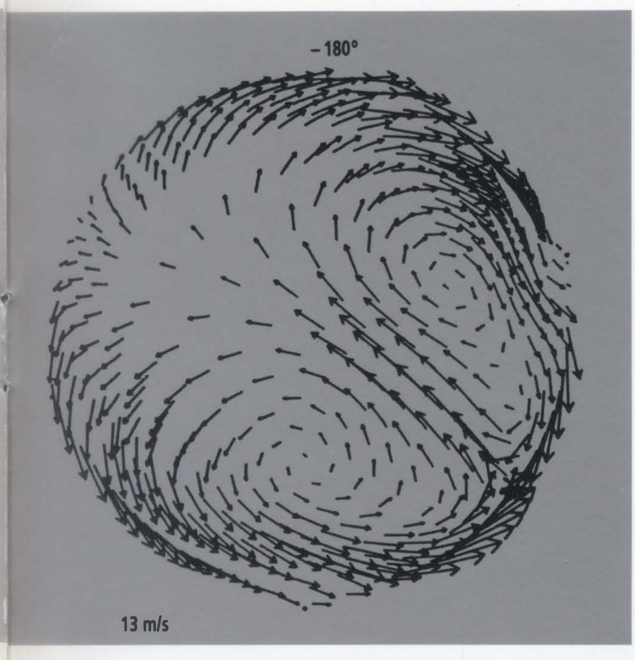


Conclusion

A new dynamic mesh-conditioning procedure made flow simulation possible in such a complex structured mesh. Effects due to the particular shape of the chamber and the direction of the inlet port were negligible when compared with the wall proximity effect, which imposed an axis of symmetry in the combustion chamber. The examination of fresh charge distribution plots indicated that the wall-to-valve proximity has an important affect on the filling of the combustion chamber with fresh charge.

Limitations in the turbulence model in this study and its inability to predict flow separation correctly may lead to inaccurate valve annular jet angles and head-loss coefficient. The prescription of the location and timing of flow separation could bring a solution to this problem.

Finally, it is important to note that mesh density and distortion effects are difficult to quantify and choices are still dictated by available memory, computation cost, and an investigator's judgment. The use of a CRAY X-MP/14 computer system was thus a determining factor that made the present computation possible.



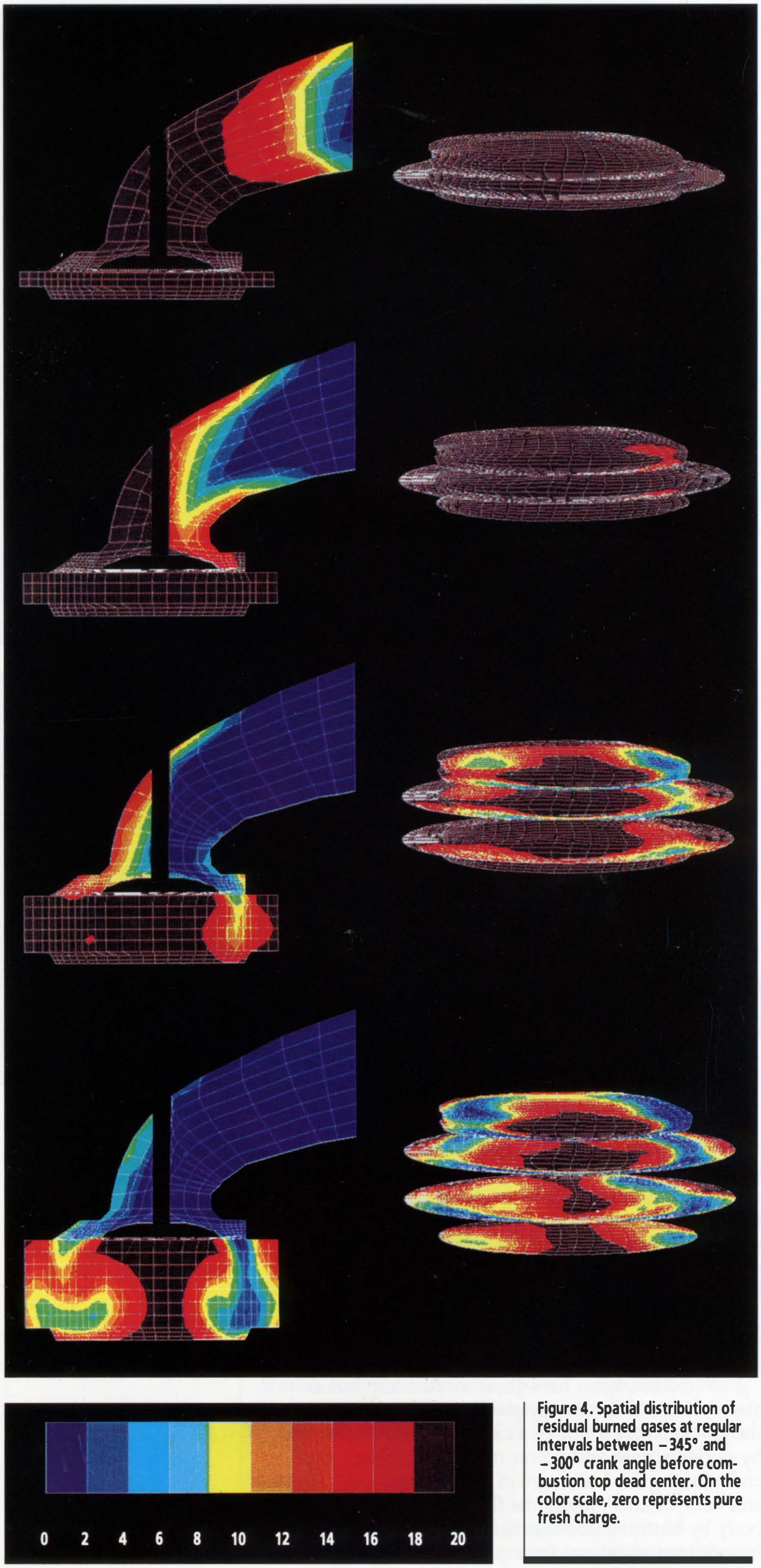


Figure 4. Spatial distribution of residual burned gases at regular intervals between −345° and −300° crank angle before combustion top dead center. On the color scale, zero represents pure fresh charge.

Acknowledgments

GIBI is developed by the C. E. A., the French Atomic Energy Authority. SUPERTAB is a product of SDRC CAE International.

About the authors

Reza Taghavi is the CFD and energetics group leader in the Scientific Analysis Department of Renault. He joined the Development Division of Renault in 1986. After earning an engineering diploma from the Ecole Nationale des Ponts et Chaussées in Paris, he obtained a Ph.D. degree in fluid mechanics from the University of Minnesota.

Alain Dupont works in the Scientific Analysis Department of Renault. He joined the Development Division in 1985. He received undergraduate degrees from the University of Rouen, where he also received a Ph.D. degree in energetics in 1984.

References

1. Taghavi, R., and A. Dupont, "Multidimensional Flow Simulation in an Inlet Port/Combustion Chamber Assembly Featuring a Moving Valve," *Proceedings from the ASME, Energy-Sources Technology Conference*, Houston, Texas, January 1988.
2. Argueyrolles, B., R. Taghavi, and M. Zellat, "Potentials of Today's CFD Codes for Modeling Real Engine Geometries," *Proceedings of the Second International Conference on Supercomputing in the Automotive Industry*, Seville, Spain, October 1988.
3. Wakisaka, T., Y. Shimamoto, and Y. Isshiki, "Three-dimensional Numerical Analysis of In-cylinder Flows in Reciprocating Engines," SAE Paper No. 860464, 1986.
4. Brandstätter, W., R. J. R. Johns, and G. Wigley, "Calculation of the Flow Produced by a Tangential Inlet Port," *Proceedings of the International Symposium on Flows in Internal Combustion Engines*, Vol. 3, FED-Vol. 28, 1986.
5. Wakisaka, T., Y. Shimamoto, and Y. Isshiki, "Induction Swirl in a Multiple Intake Valve Engine Three-dimensional Numerical Analysis," IMechE Paper No. C40/88, 1988.
6. 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 No. LA-10245MS, 1985.

Modeling undercarriage and power train components for crash simulation

H.-G. Höck, A. Poth, and W. Schrepfer
Bayerische Motoren Werke (BMW) AG
Munich, West Germany

Phase (see Figure 1)	Mechanism	Essential flux of force
1	Deformation of bumper system	Bumper — front rail — passenger compartment
2	Deformation of front end in front of the engine, relative forward motion of the engine	Front rail — passenger compartment, rising: radiator/fan — engine and front rail — engine suspension — engine
3	Backward motion of the engine, contact drive/body (can occur earlier, depending on construction), and wheel/wheel house, deformation of rear part of front end	Front rail — passenger compartment, drive-structure over contact area (can occur earlier, depending on construction), wheel house — wheel — suspension — structure, front rail — wheel house — side panel support — side frame
4	Deformation of passenger compartment (floor panel, fire wall, tunnel, sill beam, A-pillar), collapse of drive shaft	Front rail — passenger compartment, drive — fire wall/tunnel

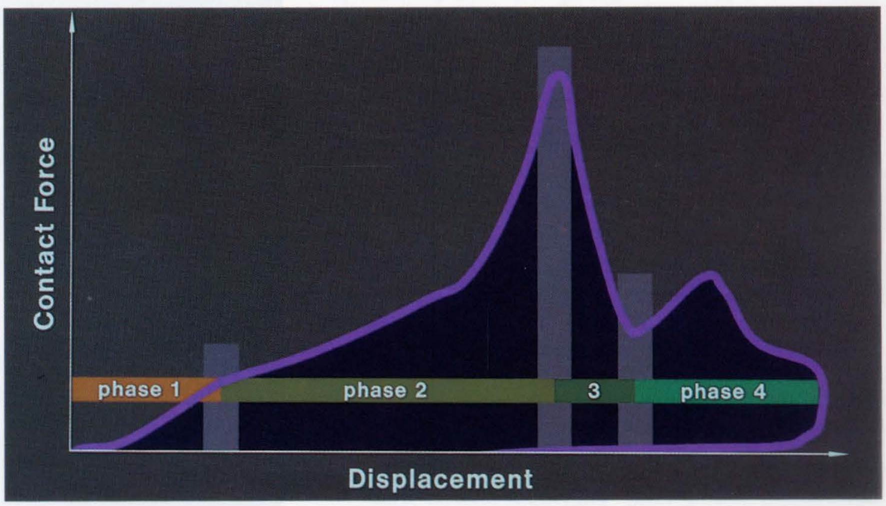


Table 1 (top). Typical crash mechanisms and essential fluxes of force during different phases of frontal crash of a BMW-type car.

Figure 1 (above). Contact force displacement of full car frontal impact during typical phases.

Numerical crashworthiness models used in the automotive industry comprise two types of components: sheet metal components and other crash-relevant components of the undercarriage and power train as well as related joints. Small but standard finite elements (FE) can be used for crashworthiness studies of sheet metal structures. Even large FE meshes can be solved in a reasonable time through the use of highly vectorized explicit FE codes and supercomputers such as the CRAY X-MP/28 system at the BMW research and engineering center in Munich.

Although a car's sheet metal components absorb most of the energy produced during a crash, the global failure mode of a car is affected significantly by undercarriage and power train components, especially during the late stages of frontal impact.[1,2,3] These stiff components change the fluxes of force in the car body by blocking possible deformation areas and by introducing additional force paths to the structure. We will discuss the technical requirements for modeling such components in the context of full car crash simulation.

Mechanisms during frontal impact

Figure 1 shows the relationship between the total contact force at a rigid wall and the displacement of a front-engine, rear-wheel-drive car during a 30-mph frontal impact. The contact force yields much information about the energy transformation in the complete car and is directly comparable to calculated results. The area under the curve in the figure corresponds to the absorbed kinetic energy. The sequence of events is divided into four typical phases; their significant mechanisms and related essential fluxes of force are described in Table 1.

The significant force peak shown between phases 2 and 3 in the figure is caused by the impact of

Component	Important properties			Modeling		Notes
	Elasticity	Plasticity	Contact	Standard FE structure	Characteristic elements	
Engine/ gear box	(2-4)*	(2-4)*	(3,4)**	X	(X)*	*Effect can be included in radiator/fan system, for example, if a rigid formulation is applied. **or earlier
Parts connected to engine	/	/	(2-4)*	O	O	*If important include in engine/gear box
Radiator/fan system	2-4	2	(2-4)*	O	X	*Only if modeled as separate structure
Drive shaft	3,4	3,4	(/)*	—	X	*Effect included in characteristic
Drive suspension	1-4	/	(2-4)*	X	X	*Stop at maximum deflection
Tires	(3,4)*	/	(3,4)*	—	(X)*	*Can be included in rims
Rims	3,4	/	3,4	X	X	Rigid formulation possible
Wheel suspension	3,4	/	3,4	X	(X)*	If local influence of kinematics is neglected
Other crash-relevant components						Decisions made case by case

1, 2, 3, 4 = Phases from Figure 1 X = Possible — = Impossible with element types usually available
= Negligible O = Possible, but not efficient

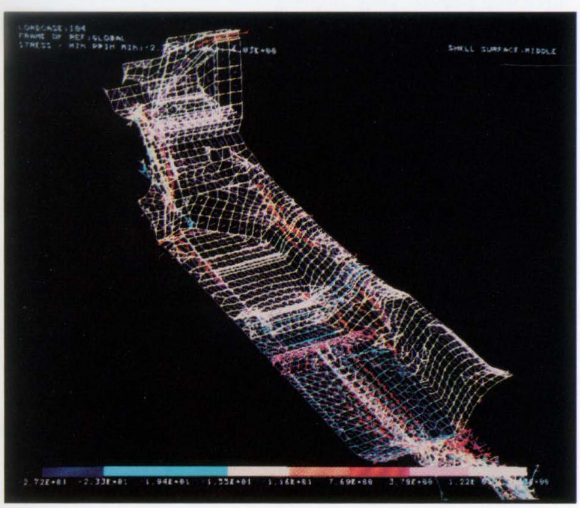


Table 2 (above). Power train and undercarriage properties and modeling methods for the crash phases identified in Figure 1.

Figure 2 (left). Distribution and orientation of minimal principal stresses in the floor panel of an axially loaded car model (static analysis); 45° orientation indicates the dominance of shear stresses.

the engine against the wall. The motion of the power train scarcely affects the car body as long as no essential flux of force arises between them. Consequently, the power train mass possesses its own kinematics. The power train comprises all of the components fixed to the engine, including the transmission, drive shaft, generator, starter, fuel feed system, and exhaust system. Components in front of the engine, such as the radiator and fan, absorb only their own energy and that of the power train. Consequently, the influence of these parts on the deformation of the body is very limited in phases 1 and 2. However, as deformation proceeds, the stiff engine block bridges large regions of the front-end structure. As a result, energy absorbtion is transposed to the area in front of the fire wall. Also, the drive shaft is compressed, leading to a deceleration peak. If the wheel contacts the wheel house, then the undercarriage influences the force distribution in phases 3 and 4. Although the steering column is displaced backward, it usually does not transmit large forces.

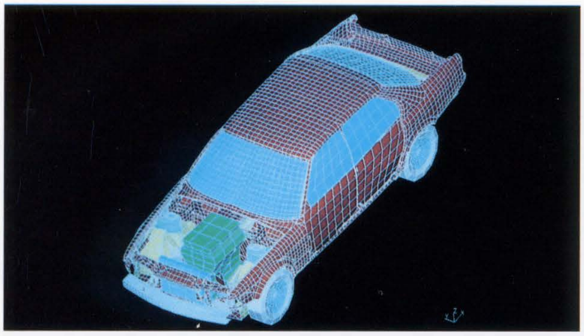
Modeling

To use an FE model of a car body, one need only describe the geometry and material behavior correctly and ensure that the mesh is fine enough to reproduce the real deformation kinematics. An entire body, or one-half of a body if symmetry is assumed, should be used for frontal crash simulation because a rigid rear-end structure will affect the force distribution. A static analysis of a car body under a longitudinal load reveals the dominance of shear forces in the floor panel, represented by a 45° orientation of the principal stresses (Figure 2). Doors and windshields also should be included in the model because of their stiffening effect.

Standard FE modeling typically is not possible or efficient enough for modeling undercarriage and power train components. This is true for two reasons. The first reason is that commercial crash codes do not yet contain appropriate element or material types for components such as the tires and drive shaft. The second reason is that too large an effort is required to establish a standard FE model. For example, modeling the radiator and fan system, which damps the engine impact at the wall, requires the creation of a model that has about 10,000 shell elements, with a minimum size of about two millimeters. For these reasons, simplified structures exhibiting equivalent characteristic behaviors must be used. These structures are called characteristic elements.

Characteristic elements

Table 2 specifies properties for crash-relevant components that are important during the



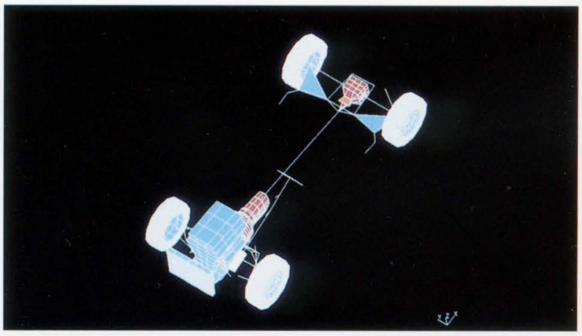


Figure 3 (left). FE crash model of complete car.

Figure 4 (right). FE crash model of undercarriage and power train components.

different phases shown in Figure 1 and the corresponding modeling philosophies. The effect of extended contact faces cannot be described sufficiently by characteristic elements, however, so their shapes must be modeled correctly, especially when friction is a factor. Total mass, rotational inertia, and the position of the center of gravity must be preserved for all components. This can be accomplished by the inclusion of additional mass points.

The formulation of characteristic elements for undercarriage and power train components creates some difficulties. The effect of the idealized component has to be defined for each degree of freedom of the boundary nodes acting at the residual structure. Even if different degrees of freedom are not coupled, the required relationships often are hard to obtain. Idealized characteristics are implemented directly into special elements as, for example, force-displacement relations. Unfortunately, the element libraries of commercial crash codes often are very limited. Therefore, a characteristic formulation modeled by FE depends strongly on the available features of the application code. Care also must be taken to control oscillation when using a system of characteristics. This can be accomplished by tuning the elastic properties or through mechanical damping. The following paragraphs discuss the modeling requirements for each component, along with information sources and problems.

Radiator and fan system

The axial compression behavior of these components must be defined to control engine rebound velocity. If the components are modeled separately, then a contact face to the engine must be included. Nodal motion in other than the axial direction can be suppressed or coupled conveniently because the cross behavior does not affect the remaining structure. Only shear motions must be restrained. Alternatively, the back nodes can be coupled to the engine front. Such a coupling does not require the definition of a contact problem, although allowance must be made for engine rotation. Such rotations can be induced by stiff components in front of the engine, a situation that can be represented by modifying the characteristic behavior of the radiator and fan system. In summary, the effect of all components in front of the engine must be accounted for across the entire front surface of the engine.

An estimation of effective power train mass is helpful in the determination of kinetic energy transfer. The difficulty lies in the unknown flux of force to the body during deceleration of the engine. Because a direct measurement of local contact force in the relevant area is not possible due to the influence of front cross members, the force portion of the car body during phase 2 in Figure 1 is assumed to be approximately constant according to the simulation of the body alone. More problems arise if local stiff parts cause the engine to rotate. More accurate data are needed and could be obtained by special tests and improved measurement methods.

Engine and gearbox

The characteristic behavior of these components can be modeled by choosing an appropriate elasticity modulus. If rigid modeling is applied, then the deformation behavior can be included in the radiator and fan system model. Preferably these parts are modeled as shell or brick structures because of the need to define contact faces.

Drive shaft

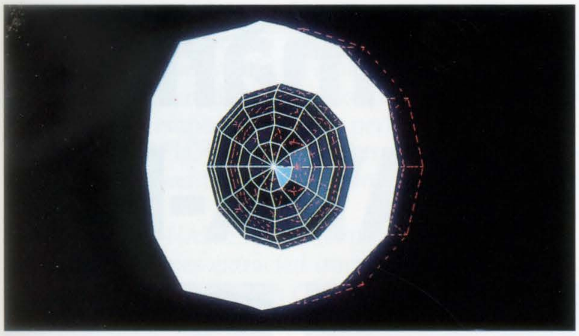
The drive shaft collapses under combined compression and torsion loading during phases 3 and 4 in Figure 1. A force-displacement relation between the back of the gearbox and the rear axle transmission can be implemented in a bar or beam element. A force history can be obtained in a full car crash test, along with the displacement histories of the end points, neglecting their rotations. A special measurement is necessary to determine the influence of shaft rotation.

Drive suspension

Depending on suspension kinematics, different modeling types are imaginable that define nonlinear elasticity between the body and the engine and gear box, or other components if necessary. The most general case requires that stiffness be defined with six degrees of freedom, although some degrees of freedom usually can be neglected or fixed. Data for drive suspensions can be obtained from full car crash tests only with much effort. Therefore, either a special test has to be developed, or the data can be extrapolated from operation data as a rough approximation. The simulation must be checked for the influence of possible deviations and dynamics.

Tires

The behavior of the tires during radial compression, if not neglected or included in the description of the rims, should be implemented in a structure that takes into account the surface that contacts the wheel house. A radial force deflection relation can be obtained from a simple test. The influences of the shape of the contact surface and other parameters usually are neglected.



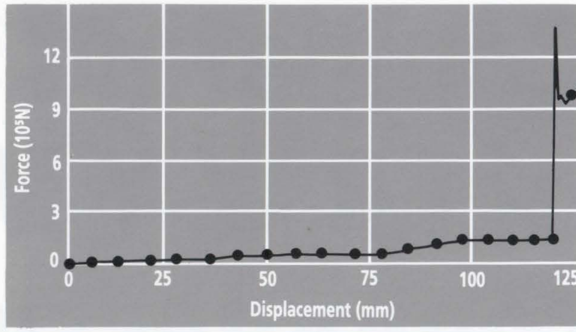


Figure 5 (left). Deformed geometry of wheel model.

Figure 6 (right). Force-displacement relation of wheel model.

Rims and wheel suspension

Standard modeling is preferable for these components. A rigid formulation for rims can be used if a tire model is included. The kinematics of wheel suspension should not be simplified if its influence on global behavior is unknown.

Examples

The commercial code PAM-CRASH[1] is used at BMW for numerical crash calculations. Although only a few element types and material laws suitable for modeling a characteristic behavior are available, all crash-relevant components of a complete car can be represented as shown in Figure 3. This model consists of about 30,000 elements and is used to simulate various crash types. The undercarriage and power train model is shown in Figure 4. The engine, gearbox, wheel suspension, and rims are modeled as elastic standard structures. A characteristic bar is used for the drive shaft. The effect of the drive suspension is handled by beam elements as a first approximation. The radiator representing all components in front of the engine consists of elastic-plastic shells with modified material behavior. The desired characteristic for the tires is attained by a combination of element deformation and kinematics (Figure 5). Figure 6 shows the output force-displacement relation.

Simplification models

Besides use in full car crash simulations, simplified models can be used in other applications, such as rough concept comparisons in the early stage of vehicle development or in situations where only partial vehicle structures or time domains must be analyzed.

If a finely meshed model of the car body exists, then crash simulation up to phase 2 in Figure 1 can be performed without including the undercarriage and power train, as long as fluxes of force to the body are assumed to be negligible. The separate analysis of partial body structures should be avoided if possible, otherwise the influence of undercarriage and power train components has to be determined, along with suitable boundary conditions.

Simplification methods can be tested after the fine-meshed model is verified. For the car body, for example, the behavior of single beams can be described by characteristic elements.[4] Since the time required for modeling the geometry of undercarriage and power train components as shown here is reasonable, simplification need not be a major concern. Nonetheless, more information is necessary to obtain reliable characteristics.

Conclusions

Undercarriage and power train components strongly influence the global failure kinematics of a car during the late stages of frontal impact, and therefore must be modeled suitably for accurate crash simulation. Detailed knowledge of crash mechanisms and of the particular kinematics of these components is necessary to model them with characteristic elements in an FE structure. Characteristic elements can be used to describe the drive shaft, wheels, and components located in front of the engine. However, undercarriage and power train components can be neglected during the early stages of the simulation if no essential flux of force to the body occurs.

About the authors

The authors work in the car body predevelopment group at the BMW research and engineering center in Munich, West Germany. H.-G. Höck received his Ph.D. degree in mechanical engineering at the Technical University of Aachen. A. Poth received his Ph.D. degree in mechanical engineering at the Technical University of Munich. W. Schrepfer received his Ph.D. degree in applied mechanics at the University of Karl Marx Stadt, East Germany.

References

1. Chedmail, J. F., P. DuBois, A. K. Pickett, et al., "Numerical Techniques, Experimental Validation and Industrial Applications of Structural Impact and Crashworthiness Analysis with Supercomputers for the Automotive Industries," *International Conference on Supercomputer Applications in the Automotive Industry*, Zürich, Switzerland, C. Marino, Ed., Computational Mechanics Publications, Southampton, Great Britain, 1986.
2. Bretz, G., W. Jarzab, and I. Raasch, "Berechnung eines frontalen Crashvorganges bie einer heckangetriebenen Limousine," *VDI Tagung Berechnung im Automobilbau*, Würzburg, West-Germany (VDI-Berichte 613, VDI-Verlag, Düsseldorf, West-Germany), 1986.
3. Haug, E., T. Scharnhorst, and P. DuBois, "FEM-Crash: Berechnung eines Fahrzeugfrontalaufpralls," *VDI Tagung Berechnung im Automobilbau*, Würzburg, West-Germany (VDI-Berichte 613, VDI-Verlag, Düsseldorf, West-Germany), 1986.
4. Höck, H.-G., A. Poth, and W. Schrepfer, "Analytische Abschätzung des globalen Frontalcrashverhaltens eines PKW," *VDI Tagung Berechnung im Automobilbau*, Würzburg, West-Germany (to be published in VDI-Berichte 699, VDI Düsseldorf, West-Germany), 1988.

Comparing the performance of CRAY Y-MP and CRAY X-MP computer systems

Gregory J. Faanes and James L. Schwarzmeier
Cray Research, Inc.

The CRAY Y-MP/832 system differs from the CRAY X-MP/416 system in three primary ways: it has twice as many processors, a clock that is 1.42 times faster, and four times as many memory banks. These differences translate into improved vector throughput by a factor of about 3.0 and improved scalar throughput by a factor of about 2.4 over the CRAY X-MP/416 computer system. We have compared the performance of these systems for several programs under various conditions and here present the results of these evaluations.

It is easy to understand and predict how uniprocessor job throughput scales with the number of processors and the clock period, but it is less clear how the throughput of multiple uniprocessor jobs or multitasked jobs is affected by the memory architecture. Therefore, we discuss the role of memory architecture in our comparisons of multiprocessing throughput, whether running multiple uniprocessor jobs or a single multitasked job. In the performance studies, various versions of the UNICOS operating system were used in a dedicated environment. The CRAY Y-MP system results are for a 6.0-nanosecond-clock system. The results here, except as noted, were obtained using CFT77 release 2.0.

First, single-CPU comparisons of the CRAY X-MP and CRAY Y-MP systems are discussed for the following programs: three SCILIB routines, the 24 Livermore Loops, the NASKERN kernels, the LINPACK routines, and some simple loops involving gather/scatter operations. Next, we show how uniprocessor job throughput is affected by memory design. Then we illustrate how system throughput is affected by job mixes composed of the three SCILIB routines, the NASKERN kernels, and the gather/scatter loops mentioned above. Finally, the performance of CRAY Y-MP and CRAY X-MP systems is compared for two microtasked application programs.

Table 1. Single-CPU results on various routines.

Code	CRAY X-MP system MFLOPS	CRAY Y-MP system MFLOPS	Ratio CRAY Y-MP system/ CRAY X-MP system
SAXPY	186.0	276.0	1.48
SSCAL	91.8	119.0	1.30
SSUM	107.0	151.0	1.41
Livermore average	71.2*	96.9*	1.36*
Livermore harmonic mean	21.5*	28.1*	1.31*
NASKERN, average (change 20 lines)	123.0	173.0	1.41
LINPACK, 100 by 100	59.0	74.0	1.25
LINPACK, 1000 by 1000	218.0	308.0	1.41
Loop-1	98.4	148.0	1.50
Gather-1	59.1	90.2	1.53
Scatter-1	59.1	93.8	1.59
Loop-2	165.0	233.0	1.41
Gather-2	74.8	95.9	1.28
Scatter-2	87.8	110.0	1.25

*Using FPP with CFT77 3.0

Single CPU comparison

The first test compared performance of three SCILIB routines: SAXPY (three-port code), SSCAL (two-port code), and SSUM (one-port code). SSCAL, for example, is called a two-port code because its statement V1(I) = S ⋆ V1(I) contains a vector read and a vector write. These tests used stride-one vector references, and the vector lengths were chosen long enough to achieve asymptotic performance on both computers. The results of this case are summarized in Table 1. Running these three vector codes on the CRAY Y-MP system yields an improved performance over the CRAY X-MP system that is approximately equal to the ratio of the clock periods.

The second performance test was run on the 24 Livermore Loops. For brevity, Table 1 lists the average MFLOPS and the harmonic mean of the 24 loops. The data for the 24 Livermore Loops were acquired using the newly released Cray Fortran Dependence Analyzer (FPP) with CFT77 3.0 to obtain improved performance. The average MFLOPS rate of the Livermore Loops is dominated by vector loops, whereas the harmonic mean emphasizes scalar performance. Thus, the average MFLOPS rate has improved by a factor (1.36) larger than that of the harmonic mean MFLOPS rate (1.31).

The NASKERN benchmark consists of seven subroutines extracted from programs used at the NASA Ames Research Center. These subroutines are highly vectorized and vary in vector lengths and strides. For the NASKERN results presented here the benchmarker was allowed to change up to 20 lines of the program. The average MFLOPS rates reported in Table 1 were obtained using the total number of floating point operations for the seven subroutines and the total amount of time to run the seven subroutines. Thus, the longest-running subroutines contributed the most to the average. The CRAY Y-MP system's CPU performance is about 1.4 times that of the CRAY X-MP system's CPU.

The LINPACK routines are a set of linear algebra subroutines for solving dense matrix problems. The results presented are for the case of a 100-by-100 matrix using the original Fortran code and one CPU, and for the case of a 1000-by-1000 matrix using one CPU and unrestricted optimization techniques (we used CAL-coded SCILIB routines). For the case of a 100-by-100 matrix, the ratio of CRAY Y-MP system MFLOPS rate to CRAY X-MP system MFLOPS rate is 1.25, indicating that the short vector lengths of this problem prevent the speedup from equaling the clock period ratio. On the other hand, the ratio of the speedups for the 1000-by-1000 case is 1.41 — the same as the ratio of the clock periods. This ratio is expected for long vector lengths.

The final single-CPU performance comparison between the CRAY Y-MP and CRAY X-MP systems is for some simple loops involving gather/scatter operations. Consider the loops in Figure 1. These loops perform constant stride vector and gather/scatter operations with a long vector length. In the gather/scatter operations the indirect index function IN(I) was chosen to access the banks of each computer randomly. (The results presented in Table 1 regarding the gather/scatter loops can be explained in terms of the memory architecture of the two computer systems, which will be addressed later.)

A few conclusions can be drawn from Table 1. First, on either system, a single, randomly indexed gather/scatter operation runs at about 60-65 percent of the speed of the comparable stride-one vector operation, with scatters slightly more efficient than gathers. Second, vector operations involving a single, randomly indexed gather/scatter memory access can run more than 50 percent faster on the CRAY Y-MP system than on the CRAY X-MP system. This difference, which exceeds the ratio of the clock periods, can be understood in terms of other improvements in the CRAY Y-MP system architecture (which will be explained later). A third conclusion can be drawn from Table 1: for a single CPU, random, multiple gather/scatter operations are relatively less efficient on the CRAY Y-MP system than on the CRAY X-MP system (after the ratio of the clock periods is taken into account).

Figure 1. Simple loops with gather/scatter operations.

```
Loop-1:            C(I)   = A(I) + S1
Gather-1:          C(I)   = A(IN(I)) + S1
Scatter-1:  C(IN(I))      = A(I) + S1
Loop-2:            C(I)   = A1(I) * B1(I) +
                            A2(I) * B2(I) +
                            A3(I) * B3(I) +
                            A4(I) * B4(I)
Gather-2:          C(I)   = A1(IN(I)) * B1(IN(I)) +
                            A2(IN(I)) * B2(IN(I)) +
                            A3(IN(I)) * B3(IN(I)) +
                            A4(IN(I)) * B4(IN(I))
Scatter-2: C1(IN(I))      = A(I) + S1
           C2(IN(I))      = A(I) + S1
           C3(IN(I))      = A(I) + S1
           C4(IN(I))      = A(I) + S1
           C5(IN(I))      = A(I) + S1
           C6(IN(I))      = A(I) + S1
           C7(IN(I))      = A(I) + S1
           C8(IN(I))      = A(I) + S1
```

In summary, a single CPU of the CRAY Y-MP computer system will perform typical scalar and vector codes about 1.2 and 1.4 times faster respectively than a single CPU of the CRAY X-MP computer system. A single processor of the CRAY Y-MP system will perform gather/scatter operations about 25-60 percent faster than a single CPU of the CRAY X-MP system.

Multiprocessing via multiple uniprocessor jobs

Next, CRAY Y-MP and CRAY X-MP system performance was compared for multiprocessor jobs on each computer. The goal was to understand how the performance of a given program on one CPU is affected by programs on the other CPUs. Because the jobs in this environment are independent except through their access to common memory, it is important to understand the memory architectures of the two computer systems in order to understand the comparison results.

For most applications, the throughput of the CRAY Y-MP system may be more than 3.0 times that of the CRAY X-MP system.

Memory architectures

One reason that the CRAY X-MP computer system has been so commercially successful is that its large memory bandwidth helps support the various levels of parallelism exploited by its CPUs. This allows CRAY X-MP systems to achieve near-peak performance more easily than would systems of similar CPU design, but with less memory bandwidth.

When the CRAY Y-MP system architecture was designed, one of the goals was to equal or exceed the memory performance of the CRAY X-MP system (on a per-processor basis). The CRAY Y-MP system performs better on a per-processor basis on memory intensive computations than does the CRAY X-MP system, even after the differences in the clock periods and CPU architectures of the two computer systems are taken into account.

The memory of the CRAY X-MP system is arranged into 64 interleaved banks using 64-Kbit ECL chips. Interleaving means that consecutive words in memory are stored in consecutive banks, so that a stride-one vector read on a given port would (in the absence of conflicts from other ports) access each of the 64 banks of the machine in successive clock periods before returning to the original bank.

Code	1 CPU	2 CPUs	4 CPUs	8 CPUs
SAXPY				
CRAY X-MP system:				
MFLOPS/CPU	186.0	179.0	133.0	--
efficiency	1.00	0.96	0.71	--
CRAY Y-MP system:				
MFLOPS/CPU	276.0	276.0	269.0	226.0
efficiency	1.00	1.00	0.98	0.82
SSCAL				
CRAY X-MP system:				
MFLOPS/CPU	91.8	91.5	90.6	--
efficiency	1.00	1.00	0.99	--
CRAY Y-MP system:				
MFLOPS/CPU	119.0	119.0	119.0	119.0
efficiency	1.00	1.00	1.00	1.00
SSUM				
CRAY X-MP system:				
MFLOPS/CPU	107.0	107.0	106.0	--
efficiency	1.00	1.00	1.00	--
CRAY Y-MP system:				
MFLOPS/CPU	151.0	151.0	151.0	150.0
efficiency	1.00	1.00	1.00	1.00
NASKERN average				
CRAY X-MP system:				
MFLOPS/CPU	123.0	118.0	110.0	--
efficiency	1.00	0.96	0.89	--
CRAY Y-MP system:				
MFLOPS/CPU	170.0	169.0	167.0	163.0
efficiency	1.00	0.99	0.98	0.96

Table 2. Performance results for multiple uniprocessor vector routines.

The 64 banks are grouped equally into four sections. A separate data path connects each port of each CPU to each of the four sections, but after leaving the section hardware, each bank has a single path from each CPU. Partitioning the banks into sections greatly reduces requirements for wiring and conflict resolution logic. Within each section is a local reduction in the number of connections from the 16 banks to the four CPUs.

On the CRAY X-MP computer system three memory conflicts may occur: bank conflicts, simultaneous bank conflicts, and section conflicts. Bank conflicts occur when any port in the system attempts to reference a bank that is still busy from a previous reference. The ECL chips used on the CRAY X-MP/416 system have a bank busy time of four clock periods. Thus, a bank busy conflict can cause a given port to be delayed from one to three clock periods, depending on the timings of the two conflicts. Simultaneous bank conflicts occur when two or more ports in different CPUs try to reference the same bank in the same clock period. When this happens, one port is given priority, and its reference proceeds, and all of the ports in the contending CPUs are disabled for one clock period. Section conflicts occur when two or more ports in the same CPU try to access the same section in the same clock period. The lower priority ports are held one clock period to resolve the conflict.

When a bank conflict occurs on the CRAY X-MP computer system, all active ports to memory in the waiting CPUs are disabled until the conflict is resolved. Stride-64 memory references are the slowest, since every element in the vector must wait three clock periods for the bank to finish referencing the previous element of the vector. Lesser degradations occur for strides 32 and 16. Also, each CPU can have only one gather/scatter access occurring at a time.

On the CRAY Y-MP computer, 256 banks are divided into four sections, and each section is divided into eight subsections. On this computer system, the bank busy time is five clock periods. The performance numbers presented here show that significant improvement has been made in the memory conflict resolution scheme of the CRAY Y-MP system compared to the CRAY X-MP computer system.

Unlike the CRAY X-MP computer system, when a memory conflict occurs on one port of a CPU on the CRAY Y-MP computer system, the other active ports of the CPU are not disabled. However, similar to the CRAY X-MP computer system, each CPU on the CRAY Y-MP computer system can have only one gather/scatter access occurring at a time.

For most vector codes the biggest difference between the CRAY Y-MP and CRAY X-MP system memory architectures is that the CRAY Y-MP system has four times as many banks as the CRAY X-MP system. Thus, programs that access memory heavily suffer less degradation due to memory conflicts on the CRAY Y-MP system compared to the CRAY X-MP system.

In some cases, such as the gather-2/scatter-2 loops in Table 1, the vector performance of the CRAY Y-MP system can be less than the ratio of the clock periods times the performance on the CRAY X-MP system. This happens because in a given clock period each processor of the CRAY Y-MP computer system sees 32 subsections with their associated subsection busy times. This is analogous to the bank conflicts experienced by a CRAY X-MP computer system with 32 banks. Subsection conflicts will dominate the performance of vector loops that tend to access the same subsection on the CRAY Y-MP system, such as stride-8 vectors or random gather/scatter references. However, when considering more commonly occurring vector references on all eight processors of the CRAY Y-MP system, the memory performance is dominated by its 256 banks.

Overall, for most user applications the throughput of the CRAY Y-MP system may be more than 3.0 times the throughput of the CRAY X-MP system. This increase is *larger* than the increase in the number of processors and clock period of the CRAY Y-MP computer system. For scalar codes the situation is different: when comparing the CRAY Y-MP system to the CRAY X-MP system, speedup per processor is *less* than the ratio of the clock periods. This primarily is because scalar processing is more sensitive to memory latency (latency is the time it takes to retrieve a single operand from memory) than is vector processing. Typically, scalar operations improve by about a factor of 1.2 per processor on the CRAY Y-MP system over the CRAY X-MP system.

Job mixes of constant stride vector operations

In this test we ran multiple copies of SCILIB routines or the NASKERN routines with the goal of understanding system throughput on the CRAY Y-MP and CRAY X-MP computer systems. First, multiple copies of three SCILIB routines were run: SAXPY (three-port code), SSCAL (two-port code), and SSUM (one-port code), all for unit stride and long vector lengths. Microtasking was used in the test program to spawn multiple copies of these routines over all the CPUs, and the programs were run long enough to guarantee that they were running concurrently. Table 2 shows the average MFLOPS for each program for one, two, and four CPUs on the CRAY X-MP and CRAY Y-MP systems, and the MFLOPS for eight processors on the CRAY Y-MP system. Also listed is the memory efficiency in each case, which is defined as the average MFLOPS for multiple copies of a program divided by the MFLOPS for a single copy of the program (with nothing else running on the computer). One minus this ratio represents the relative performance that is lost due to memory contention.

The table shows that the memory efficiency of eight copies of the SAXPY code on the CRAY Y-MP system is about 10 percent better than four copies of the SAXPY code on the CRAY X-MP system. Furthermore, neither computer system showed any measurable memory performance loss due to the two- and one-port SCILIB routines. The throughput of SAXPY on the CRAY X-MP system is 4 × 133 = 532 MFLOPS, while on the CRAY Y-MP system the throughput is 8 × 226 = 1808 MFLOPS — which is 3.4 times greater than the CRAY X-MP system's throughput.

For the second throughput test of various constant stride memory references, multiple copies of the NASKERN routines were run. This test provides a realistic picture of the performance improvement a typical mix of user jobs should see, since the NASKERN programs contain a variety of vector lengths and strides. Table 2 shows that the CRAY Y-MP system's memory efficiency is 0.96 for eight copies of the NASKERN routines, while the CRAY X-MP computer's memory efficiency is 0.89 for four copies of the routines. The throughput of NASKERN routines on the CRAY X-MP system is 4 × 110 = 440 MFLOPS, while on the CRAY Y-MP system the throughput is 8 × 163 = 1304, which is 3.0 times the CRAY X-MP computer value.

Multiple gather/scatter operations

Next, to reveal the effect of gather/scatter operations on system performance, we simulated the performance of a workload consisting mainly of gather/scatter operations. This was achieved by running multiple copies of the gather/scatter loops in Figure 1 and Table 1 on various numbers of CPUs on the two computer systems. Judging by the efficiencies shown in Table 3, the CRAY Y-MP system delivers from 10-20 percent more of its memory bandwidth for these operations than does the CRAY X-MP system. As mentioned before, this is because the CRAY Y-MP system has more memory banks per processor than the CRAY X-MP system.

The ratio of MFLOPS ratings for the two systems is higher for the gather-1 case (1.53) than for the gather-2 case (1.28 — see Table 1). This occurs because in the gather-1 loop, a vector load, a gather, and a vector store happen simultaneously on three memory ports, whereas in the gather-2 loop, primarily one memory operation happens at a time — a gather

Code	1 CPU	2 CPUs	4 CPUs	8 CPUs
Gather-1:				
CRAY X-MP system:				
MFLOPS/CPU	59.1	49.6	38.5	––
efficiency	1.00	0.89	0.68	––
CRAY Y-MP system:				
MFLOPS/CPU	90.2	86.4	80.6	72.4
efficiency	1.00	0.95	0.89	0.80
Scatter-1:				
CRAY X-MP system:				
MFLOPS/CPU	59.1	50.8	39.2	––
efficiency	1.00	0.89	0.68	––
CRAY Y-MP system:				
MFLOPS/CPU	93.8	91.8	88.0	81.3
efficiency	1.00	0.98	0.95	0.88
Gather-2:				
CRAY X-MP system:				
MFLOPS/CPU	74.8	68.8	58.5	––
efficiency	1.00	0.92	0.78	––
CRAY Y-MP system:				
MFLOPS/CPU	95.9	93.5	89.8	83.7
efficiency	1.00	0.98	0.94	0.88
Scatter-2:				
CRAY X-MP system:				
MFLOPS/CPU	87.8	80.1	67.6	––
efficiency	1.00	0.90	0.75	––
CRAY Y-MP system:				
MFLOPS/CPU	110.0	109.0	108.0	105.0
efficiency	1.00	1.00	0.98	0.95

Table 3. Performance results for multiple uniprocessor gather/scatter jobs.

on a single port. In the former case, on the CRAY X-MP system a memory conflict on the gather shuts down the vector-load and store ports until the conflict is resolved. On the CRAY Y-MP system a conflict on the gather port does not shut down the other ports. In the case of the gather-2 loop, where many consecutive gathers are executed (on either computer system), a memory conflict in a gather will slow down either computer system in a similar manner. On the CRAY X-MP system the throughput of gather-1 loops is 4 × 38.5 = 154 MFLOPS, while on the CRAY Y-MP system the throughput is 8 × 72.4 = 579 MFLOPS, which is 3.8 times the CRAY X-MP system value.

The CRAY Y-MP system's improved memory architecture over the CRAY X-MP system is demonstrated dramatically by tests of multiprocessing system throughput. Because the CRAY Y-MP system has a greater memory efficiency than the CRAY X-MP system, in a batch environment each processor will suffer less memory degradation due to the memory traffic of other processors, and less fluctuation will occur in the job-to-job performance of an application.

Multiprocessing via microtasking

Two microtasked application programs, ARC3D and SPECTL, were run using various numbers of CPUs on the two computer systems. The times reported are wall-clock times, not including the time for initialization work. (In realistic-sized application

Code	1 CPU	2 CPUs	4 CPUs	8 CPUs
ARC3D				
CRAY X-MP system:				
time (seconds)	68.9	35.5	18.9	––
speedup	1.00	1.94	3.65	––
efficiency	1.00	0.97	0.91	––
CRAY Y-MP system:				
time (seconds)	50.8	25.6	12.9	6.67
speedup	1.00	1.98	3.92	7.62
efficiency	1.00	0.99	0.98	0.95
Amdahl efficiency (99.5 percent parallel)	1.00	0.995	0.985	0.966
SPECTL				
CRAY X-MP system:				
time (seconds)	17.2	8.73	4.57	––
speedup	1.00	1.97	3.77	––
efficiency	1.00	0.99	0.94	––
CRAY Y-MP system:				
time (seconds)	12.1	6.15	3.17	1.70
speedup	1.00	1.98	3.83	7.16
efficiency	1.00	0.99	0.96	0.90
Amdahl efficiency (98.5 percent parallel)	1.00	0.99	0.96	0.905

Table 4. Performance comparison for two microtasked programs.

runs the initialization times are negligible compared to the times to run the rest of the programs.)

ARC3D

ARC3D, a computational fluid dynamics code, was more than 99 percent parallel for the sample problem. Table 4 defines the efficiency of a multitasked job as the speedup from using *P* processors on the job, divided by *P*. This measures the efficiency of using *P* processors in a multitasked job versus using them to process *P* uniprocessor jobs (apart from the effect of memory conflicts). The table also shows the efficiencies expected from Amdahl's law for 99.5 percent parallel work. There is a speedup of about 3.7 and a four-processor efficiency of 91 percent on the CRAY X-MP system, and a speedup of about 7.6 and an eight-processor efficiency of about 95 percent on the CRAY Y-MP system. Thus, running ARC3D on eight processors on the CRAY Y-MP computer system is more efficient than running ARC3D with four processors on the CRAY X-MP computer system. The efficiencies on both systems are consistent with Amdahl's law. However, we see that the efficiencies on the CRAY Y-MP computer system are much closer to the Amdahl efficiencies than are those on the CRAY X-MP system. This can be attributed to the improved memory architecture of the CRAY Y-MP system. ARC3D is dominated by large-stride memory references, which lead to frequent memory conflicts with multiple processors. These memory conflicts constitute overhead, which causes the actual efficiency on a given system to be less than the Amdahl efficiency. Table 4 shows that the overhead due to memory conflicts is much less on the CRAY Y-MP system than it is on the CRAY X-MP system.

SPECTL

SPECTL, a highly parallel weather code, was approximately 98-99 percent parallel for the sample problem. Table 4 shows a speedup of about 3.8 and a four-processor efficiency of 94 percent on the CRAY X-MP system, and a speedup of about 7.2 and an eight-processor efficiency of 90 percent on the CRAY Y-MP system. Also shown are the expected efficiencies from Amdahl's law for 98.5 percent parallel work. As was the case with ARC3D, the efficiencies of both systems for SPECTL are consistent with Amdahl's law. Unlike ARC3D, for SPECTL the efficiencies of both systems are close to the Amdahl efficiencies. This is because SPECTL stresses memory much less than ARC3D. The work in SPECTL is dominated by parallel stride-1 fast Fourier transforms, and by efficient matrix multiplies, which make repeated use of data in registers rather than frequently retrieving the data.

On the CRAY Y-MP computer, codes with sufficient parallelism easily can achieve speedups in excess of 7.0.

Conclusions

On the CRAY Y-MP system, the single-processor scalar and vector speedups are 1.2 and 1.4, respectively, over the CRAY X-MP system. However, the CRAY Y-MP system's better memory conflict resolution hardware and larger number of memory banks have increased its uniprocessor throughput over and above what would be expected due to its larger number of processors and faster clock rate alone.

The CRAY Y-MP system's performance advantage for gather/scatter operations is 35-70 percent per processor over the CRAY X-MP system. On the CRAY Y-MP computer, codes with sufficient parallelism easily can achieve speedups in excess of 7.0. The improved memory architecture of the CRAY Y-MP system also allows some multitasking programs to have a higher multitasking efficiency than they would on the CRAY X-MP system. These encouraging results bode well for properly designed, shared-memory, multiprocessor computer systems.

Each issue of CRAY CHANNELS includes a technical article that offers insights into the Cray environment. The editors thank Chris Hsiung and Jim Schwarzmeier for their regular technical advice.

Acknowledgments

ARC3D information can be obtained from Thomas H. Pulliam, Mail Stop 202A-1, NASA Ames Research Center, Moffett Field, CA, 94035. SPECTL is from the GFDL benchmark at Princeton, New Jersey.

About the authors

Gregory J. Faanes has worked in Cray Research's systems and performance analysis group in Chippewa Falls, Wisconsin, since 1985. He evaluates Cray system performance and studies present and future Cray system architectures. He earned a bachelor's degree in applied mathematics from the University of Wisconsin-Stout in 1983.

James L. Schwarzmeier is a member of Cray Research's computational chemistry group. He was formerly a member of the systems and performance analysis group. Since he joined Cray Research in 1987, he has worked mainly on the vector and multitasking optimization of application codes. He earned a Ph.D. degree in plasma physics from the University of Wisconsin-Madison in 1977. After serving as a postdoctoral fellow at the Courant Institute of Mathematical Sciences, New York University, he worked at the Los Alamos National Laboratory for six years.

CORPORATE REGISTER

Two research organizations order CRAY Y-MP systems

Kernforschungsanlage Jülich (KFA), West Germany's largest scientific research laboratory, has ordered a CRAY Y-MP/832 computer system. This is the first CRAY Y-MP system order from outside of the United States. The purchased CRAY Y-MP system will replace KFA's CRAY X-MP/22 computer system. KFA, a Cray customer since 1983, also is upgrading its CRAY X-MP/48 computer system to a CRAY X-MP/416 system and adding an SSD-31 solid-state storage device. The CRAY Y-MP system will be installed in the third quarter of 1989 at the West German Supercomputer Center, Hoechst-Leistungs-Rechen-Zentrum (HLRZ), pending export license approval. HLRZ serves researchers from universities and national laboratories. The Cray system will be applied to basic research, including elementary particle and solid-state physics.

The **Microelectronics Center of North Carolina** has ordered a CRAY Y-MP computer system for installation in the third quarter of 1989 at the North Carolina Supercomputer Center (NCSC) in Research Triangle Park, North Carolina. The purchased system will be used by North Carolina universities and research organizations for scientific research and engineering. Some of the research projects that will use the supercomputer system at NCSC will be supported by Cray Research's University Research and Development Grant Program.

Chrysler Motors, a new Cray customer, ordered a CRAY X-MP/14se computer system. All three of the major U.S. auto manufacturers are now Cray customers. The leased system was installed in the fourth quarter of 1988 at Chrysler's engineering center in Highland Park, Michigan. It is running Cray Research's UNICOS operating system, which is based on the AT&T UNIX System V operating system. The computer system is used for computer-aided engineering applications such as structural analysis, design optimization, sheet metal deformation studies, and aerodynamic simulation in support of the design and testing of Chrysler's next generation vehicles.

FIAT Auto ordered a CRAY X-MP/14se computer system for installation in the fourth quarter of 1988 at the FIAT Auto Computer Center in Torino, Italy. Alfa Romeo and Lancia, companies of FIAT Auto, also use the Cray system. The CRAY X-MP/14se system runs Cray Research's UNICOS operating system. FIAT Auto is using the Cray system for applications, such as crash simulation, design optimization, and aerodynamic simulation.

The **Finnish State Computer Center**, VTKK, has ordered a CRAY X-MP EA/416 computer system. The supercomputer will be the first Cray system installed in Finland. Pending export license approval, an interim CRAY X-MP/14se system will be installed in the first quarter of 1989, and will be upgraded to a CRAY X-MP/EA 416 computer system later in the year. VTKK is funded by Finland's Ministries of Finance and Education and will provide service to researchers at Finnish universities, the Technical Research Center of Finland (VTT), and industry. The system also will be used by the Finnish Meteorological Institute to run the operational weather forecasting service for Finland.

NASA Langley Research Center, a new customer for Cray Research, has ordered a CRAY-2S/4-128 computer system. The purchased system will be installed in the first quarter of 1989 at NASA Langley's Central Scientific Computer Complex in Hampton, Virginia. The system will be used to advance research in areas including aeronautics, space technology, electronics, and structures. The Langley Research Center, established in 1917, was the nation's first federal aeronautical research center. Today its researchers are developing techniques in computational fluid dynamics, computational structures mechanics, and earth and space sciences. NASA Langley recently held a contest to name the center's supercomputers. "Famous airplanes" was chosen as a naming convention to honor Langley's contributions to aircraft design. The new Cray system was named "Voyager" after the first aircraft to fly around the world without refueling.

The **P.S.A. Group (Peugeot and Citröen)** has ordered a CRAY X-MP/216 computer system. The leased system, which was installed at Citröen's computer center in the fourth quarter of 1988, replaces a CRAY X-MP/14 system that was installed in the first quarter of 1987. The system upgrade will be used to study crash simulation, combustion analysis, acoustics, aerodynamics, and vibration analysis. The upgrade also will facilitate P.S.A.'s migration from the Cray Operating System (COS) to Cray Research's UNICOS operating system.

Arizona State University (ASU) ordered a CRAY X-MP/14se computer system for installation in the fourth quarter of 1988 at ASU Computing Services, Tempe, Arizona. The computer is dedicated to research in areas such as astronomy; computational chemistry; and aerospace, electrical, and mechanical engineering. Currently, 22 Cray systems are installed at universities in 10 countries.

Cray Research announces executive promotions

In November, Cray Research announced the promotions of several executives. Marcelo A. Gumucio was promoted to the newly created position of president and chief operating officer. Gumucio also was elected to the company's board of directors. He had been executive vice president, marketing, since joining Cray in 1983.

Robert H. Ewald was elected executive vice president, software. Ewald joined Cray Research in 1984. He became vice president, commercial marketing, in 1985 and vice president, software division, in 1987. Ewald will continue to head the company's software development acivities.

Edward A. Masi was elected senior vice president, marketing. Masi came to Cray in 1980 as manager of the company's Eastern Region. He became vice president, marketing, in 1987. Masi's role will expand to include all corporate marketing activities.

Cray Research also announced the formation of a Technical Council to be chaired by company chairman and chief executiver officer John Rollwagen. The council will be responsible for setting long-term product development directions for the company. The other members of the council are Seymour Cray; Lester T. Davis, executive vice president, who continues to head operations in Chippewa Falls and Rice Lake, Wisconsin; and Ewald.

"We have become a large company, based on our continued technical excellence," Rollwagen said. "The time has come to fill the role of chief operating officer, who will focus on day-to-day activities and the accomplishment of objectives spanning one to three years, while I devote my attention to strategies for the longer term. Our corporate objectives or strategies will be built on the premise of technical leadership, and the Technical Council will advise me and the rest of the company in how to sustain that."

Cray Research announces Cray Link Software for UNIX Workstations

Cray Link Software for UNIX Workstations (CLS-UX) version 1.01 provides users of UNIX Workstations with low-cost access to Cray computer systems through an attached VAX/VMS system hosting the Cray Research VAX/VMS Station. CLS-UX is compatible with any Cray computer system running the UNICOS or COS operating system. CRAY Y-MP, CRAY X-MP EA, CRAY-2, CRAY X-MP, and CRAY-1 systems are supported.

Version 1.01 of CLS-UX can run on three different systems based on UNIX software: Sun Workstations, Apollo DOMAIN systems, and Digital Equipment Corporation (DEC) VAX systems running the ULTRIX operating system. Future releases will support additional systems based on UNIX software.

Sun and Apollo workstations communicate with the attached VAX/VMS system through TCP/IP and the machine-independent Cray Remote Station Protocol (CRSP). VAX/ULTRIX systems communicate with the attached VAX/VMS system through either TCP/IP or DECnet, and CRSP. No hardware interface is needed between the system based on UNIX software and the Cray computer system. The attached VAX/VMS system must host TCP/IP or DECnet or both (as needed) and version 4.01 or later of the VAX/VMS Station.

Through CLS-UX, users can submit jobs to the Cray system, monitor and control those jobs, stage (transfer) data to and from the Cray system, or use the Cray system interactively. Cray context, a menu-driven, full-screen interface, lets users submit, monitor, and control batch jobs. The Cray interactive facility lets users log in to Cray computer systems interactively, giving them the ability to access Cray files or data sets and execute UNICOS or COS commands directly. Proxy staging allows users to stage data directly to or from another account on other systems.

A CLS-UX system can function as either a remote system or as a server system that acts as a gateway for other CLS-UX remote nodes in a network of UNIX systems. An attached VAX/VMS Station can host as many as 50 CLS-UX systems, any number of which can act as servers, and a CLS-UX server can accommodate as many as 16 remote nodes.

CLS-UX contains sophisticated security features. Many security checks are performed when users access the Cray system: all commands entered by the user pass through a command filter controlled by the attached VAX/VMS Station and a filter controlled by the CLS-UX software administrator. Cray system status, log file messages, and input queues are controlled so that users can access only their jobs.

For more information on Cray Link Software for UNIX Workstations, contact the nearest Cray Research sales office.

VAX/VMS Station 4.01 includes remote interactive capability

Release 4.01 of the VAX/VMS Station adds significant enhancements, including remote interactive capability. The VAX/VMS Station runs on Digital Equipment Corporation VAX computers that serve as front-end gateways to Cray computer systems. The station software provides users with a wide array of communication facilities, including submission, status, and control of batch jobs; file transfer (dataset staging) between systems; and interactive access to the Cray system.

VAX/VMS Station software runs on either attached or remote VAX/VMS systems. A remote station provides users with virtually all of the capabilities available at attached stations. An attached VAX/VMS Station can host both remote VAX/VMS Stations and Cray Link Software for UNIX Workstations (CLS-UX) remote systems. The VAX/VMS Station is compatible with any Cray computer system running the UNICOS or COS operating system. CRAY Y-MP, CRAY X-MP EA, CRAY-2, CRAY X-MP, and CRAY-1 systems are supported.

New features

Version 4.01 of the VAX/VMS Station adds new features, including

- ☐ Full interactive processing capability for remote stations
- ☐ A considerably faster interactive user interface and several new commands to support interactive processing
- ☐ Support for the Cray Remote Station Protocol, which allows the attached station to accept remote station connections from non-VMS systems, such as UNIX Workstations running CLS-UX
- ☐ Automatic conversion of VMS files to UNICOS data-record format for UNICOS 4.0 systems
- ☐ The ability to share station images within a homogeneous cluster
- ☐ Asynchronous functioning of the operations manager
- ☐ The ability to run as many as four network listeners
- ☐ Enhancement of installation procedures

For more information about Cray Research's VAX/VMS Station, contact the nearest Cray Research sales office.

APPLICATIONS UPDATE

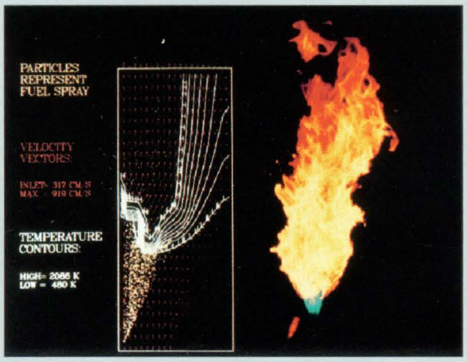
KIVA, KIVA-II simulate internal combustion engines

KIVA and KIVA-II are multidimensional fluid dynamics codes that have been written specifically for CRAY Y-MP, CRAY X-MP, and CRAY-1 computer systems running under all Cray operating systems. The codes, which extensively use the vector architecture of Cray systems, are used by engine companies throughout the world to design advanced internal combustion engines.

KIVA and KIVA-II were developed by Group T-3 at Los Alamos National Laboratory under sponsorship of the United States Department of Energy's Conversion and Utilization Technologies program. The codes model the interplay of complex physical and chemical processes, such as the mixing of fuel and air, ignition, chemical reactions, and heat transfer, which take place in a transient, three-dimensional, turbulent flow field. The flow field also may interact with an evaporating liquid fuel spray.

Although KIVA and KIVA-II were developed for internal combustion engine design, they also can be used for many other applications. The equations and the numerical solution procedure are general and can be applied to laminar or turbulent subsonic or supersonic flow, in two or three dimensions, with or without sprays or combustion. KIVA has been used for analyses of cold flows in complicated geometries, nonreacting and reacting sprays, Bunsen burner flames, hydrogen burning and debris dispersal in nuclear reactor accident simulations, and chemical explosives and their product gases.

KIVA solves the unsteady equations of motion for a chemically reactive mixture of ideal gases, coupled to the equations for a single-component vaporizing fuel spray. The number of gas-phase species and chemical reactions are arbitrary. They are limited only by computer time and storage constraints. The code distinguishes between slow reactions, which proceed kinetically, and fast reactions, which are assumed to be in equilibrium. Two models are available to represent the effects of turbulence. One is a standard version of the *k-e* turbulence model modified to include the effects of compressibility and spray/turbulence interactions. The other



KIVA simulation (left) and experiment (right) of a continuous spray combustor.

is a subgrid scale turbulence model that includes a transport equation for turbulent kinetic energy associated with subgrid scale motions.

The spray model is one of the best validated submodels of KIVA. Evaporating liquid fuel sprays are calculated by a discrete-particle technique, in which each computational particle represents a number of droplets of identical size, velocity, and temperature. The particles and gas interact by exchanging mass, momentum, and energy. Droplet collisions and coalescences also are accounted for, and a new model for droplet aerodynamic breakup has been installed in KIVA-II.

For more information about using KIVA and KIVA-II on Cray computer systems, contact Peter J. O'Rourke, Los Alamos National Laboratory, Los Alamos, NM, 87545; telephone: (505) 667-9091, or Jef Dawson, Cray Research, Inc., 1333 Northland Drive, Mendota Heights, MN, 55120; telephone: (612) 681-3670.

FLUENT models fluid flow

FLUENT is a computational fluid dynamics code for the design and analysis of fluid flow elements. The code is available from Creare Inc., and runs on all Cray computer systems under all Cray operating systems.

FLUENT software solves the steady two- and three-dimensional Navier Stokes equations for laminar or turbulent subsonic flows in the Cartesian or cylindrical polar coordinate systems. Turbulence is modeled using the standard two-equation *k-e* model. The code divides the flow domain into discrete computational cells using a staggered grid technique.

The code can be used for applications such as combustion design and engineering, aerodynamic design, particulate deposition, computer chip manufacturing, fire research, pollution control, and process engineering operations.

The code also solves the enthalpy transport equation, which enables it to solve heat-transfer problems. It has a six-equation radiation model that allows calculation of radiative heat transfer.

Three chemical species can be handled by FLUENT, in addition to an inert gas. Conservation equations are solved for each chemical species. If additional chemical species are required, they can be produced with custom modifications. The code also can solve single-step global chemical reactions.

FLUENT can model a dispersed second phase such as droplets or particles. That is, it can determine the dynamics of droplets or particles in the continuous gas or liquid flow field. Lagrangian trajectory computations are performed for groups of particles or droplets based on the computed continuous gas or liquid phase flow field.

The code has a built-in generator that automatically produces a uniform computational grid. It can handle nonuniform grids in which the spacing between the grids is not constant. Facilities are provided to compress or expand grids in regions of high gradients for improved solution accuracy.

FLUENT uses the SIMPLE algorithm, which recasts the continuity equation into a pressure-correction equation. The pressure correction equation is then used at each iteration to correct the velocity field obtained from the solution of the momentum equations. The nonlinear algebraic equations resulting from the discretization of the governing partial differential equation are solved using iterative line-by-line solution techniques. For more information about using FLUENT with Cray computer systems, contact Barbara Hutchings, Creare Inc., Etna Road, P. O. Box 71, Hanover, NH, 03755; telephone: (603) 643-3800, or Rob Vermeland, Cray Research, Inc., 1333 Northland Drive, Mendota Heights, MN, 55120; telephone: (612) 681-3401.

USER NEWS

Cray system plows through a snow job

The Rocky Mountain Forest and Range Experiment Station and two researchers at Colorado State University are using a CRAY X-MP/48 system to solve a snow job. No, their mission is not to discover if two snowflakes ever share the same shape. Instead, they are using their 70-hour National Science Foundation grant at the Pittsburgh Supercomputing Center to examine how the microstructure of snow influences water vapor movement in snow. Such knowledge will help predict when melting snow will release pollutants and even pinpoint the cause of avalanches.

"When seasonal snow covers the ground, pollutants such as sulfur dioxide are deposited from cars and other sources," explains Mark Christon, a CSU mechanical engineer involved with the project. "In the spring when the first melt water is released, it can have high levels of impurities and shock the ecosystem."

Pat Burns, a CSU professor who specializes in heat and mass transfer in CSU's mechanical engineering department, refers to the problem as "acid snow." "There's been a lot of talk about acid rain on the East Coast, but out here in the West where we have colder weather, we have an acid snow problem. In particular we're looking at its effect on the ecosystem at high elevations like the Rocky Mountains. Such areas have fragile ecosystems that are susceptible to damage at more moderate pH levels than ecosystems that exist at sea level."

> **"Our color images show just the opposite of what has been assumed for years."**

Although many ecologists have taken a macroscopic view of the problem to look at the effect of melt water on large hydrological systems, this project focuses on the basics. "Our approach has been to get down to a very small scale and answer fundamental questions about what is happening in the snow cover," says Christon.

Burns explains, "Researchers at the Rocky Mountain Forest and Range Experiment Station, led by Dick Sommerfeld, have been doing some experimental work measuring impurity pulses and we've been doing some simulation work." Burns and Christon initially created a two-dimensional model of snow metamorphism on a Cyber 205 computer, but needed the power of a CRAY X-MP system to create a comprehensive three-dimensional model. "The size of the problem requires a Cray system," says Christon. "It is a finite element model, which involves dynamic, deforming meshes. The Cray system is a good choice for this method because the algorithms lend themselves better to the Cray architecture."

By simulating snow evolution on a CRAY X-MP system, they are able to see how the structure of snow cover evolves throughout the winter. The images generated by the Cray system show how the temperature is distributed in the ice or solid phase of the snow. "In the technical literature it has been assumed that all of the horizontal ice crystals are isothermal," says Burns. "If this was the case, our images would show one solid color for the ice crystals at the top and bottom of the ice lattice. Our color images show just the opposite of what has been assumed for years."

So far, using the CRAY X-MP system, Christon and Burns have shown that for the three-dimensional ice lattices considered, the rate at which water molecules move through a snow cover is 1.02 to 1.2

times faster than the rate at which water molecules move through dry air. These results are for snow densities ranging from 91.6 to 458.0 kilograms/cubic meter and are for highly idealized ice lattices without protuberances.

"These results tell us how much influence the geometry and the phase change from solid to vapor (and back again) can have on the heat and water vapor transport for a given temperature gradient in the snow cover," says Christon. "This influence can be used to determine how the geometry and phase change will influence the movement of pollutants in the snow cover.

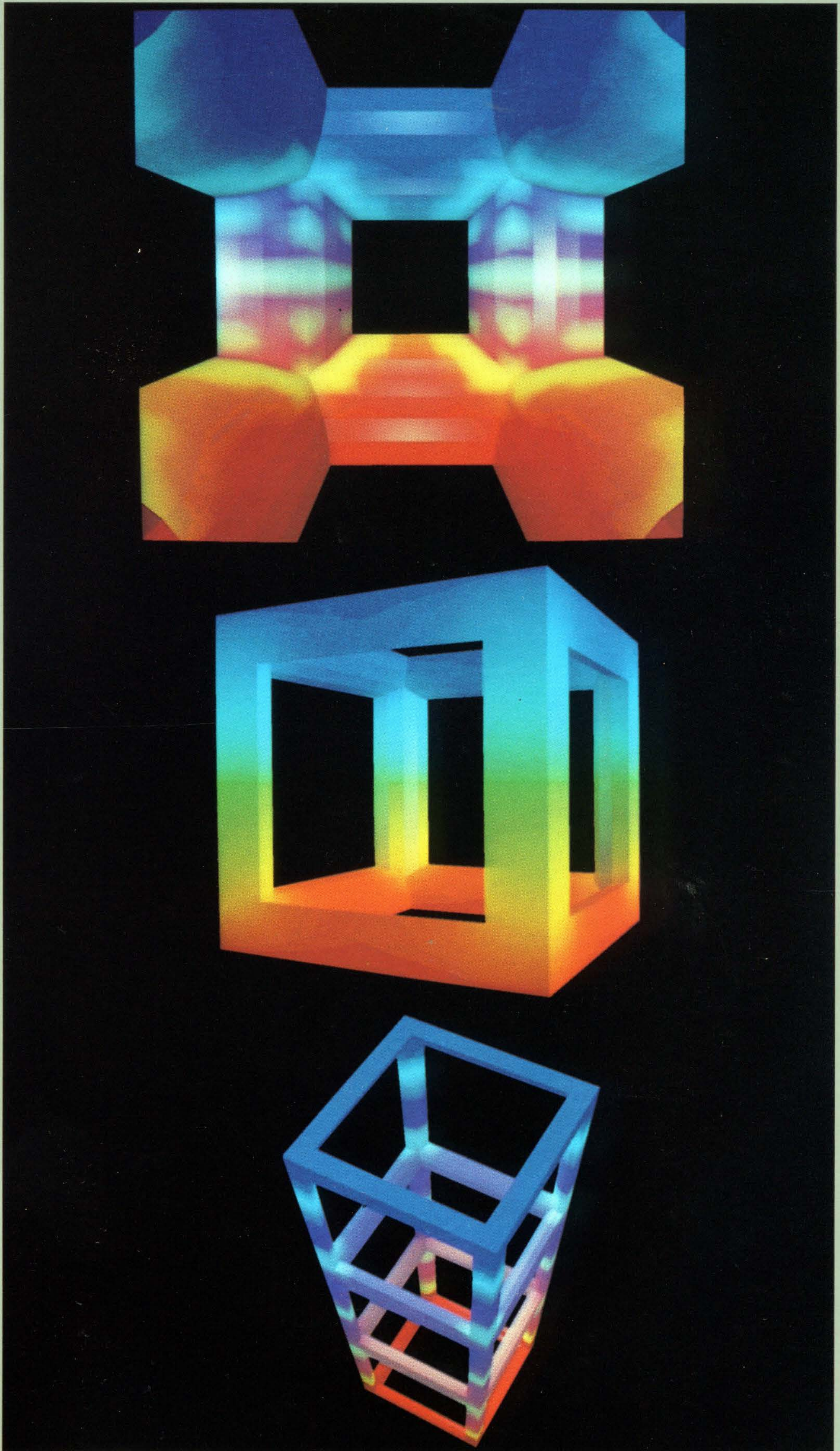
"When we enhance the model by adding a pollutant phase, we actually will be able to see how pollutants move through a snow pack, and verify some hypotheses about this," says Christon, adding that this information also may verify theories about the causes of avalanches. "As the ice lattice evolves, recrystallization continually occurs, and much of the structural integrity of the snow structure can be lost in a very short time, resulting in avalanche conditions."

When they animate their results, Burns and Christon will be able to "stand inside" the snow crystals to observe how the crystals change with time. Their findings will be used in larger hydrological models for predicting the timing and severity of an acid pulse. Explains Christon, "With that knowledge, we can do something. For example, we can recommend liming levels for the water system that will experience an acid shock. Right now it's a guess as to when the acid pulse is going to occur, and how much lime can be added to a small alpine lake."

Cray system bears timely results

Problem: Aerojet ElectroSystems needed to test three years of continuous wear on a bearing for a weather instrument, the Advanced Microwave Sounder Unit, AMSU-A, that would be attached to NASA's TIROS weather satellite. The high-precision bearing needed to endure 12 million starts and stops in a three-year period. One alternative would be to perform a life test, which would cost about $1 million and require a vacuum test chamber and three years of time. Engineers were pleased to find a computer code that could model accelerated, transient bearing loads, but were disappointed with its performance on their Cyber 180 computer.

"We took that code, converted it to run on our Cyber 180, and much to our chagrin, found that the running time for one cycle was months! A run that we thought wouldn't



(Top) View of the lattice (density 458 kg/m³) from the inside. Significant gradients are apparent along the surface (yellow). (Center) A single ice lattice (density 185 kg/m³) cell viewed from the front. Temperature gradients are illustrated along the horizontal ice crystals at the top and bottom of the cell. In both images, light colored strips are constant temperature bands that correspond to one-fifth of the total temperature difference along the ice lattice cell. The color spectrum goes from blue (cool) to red (warm). Stripes represent three-dimensional contour lines that have been laid on the three-dimensional ice lattice geometry. (Bottom) Three ice lattice cells with a total temperature difference of −0.012°C and density of 185 kg/m³.

take too long would take forever to complete," says Peter Welch, supervisor of applied mechanics at Aerojet ElectroSystems.

Solution: "We took the problem down to the CRAY X-MP system at the San Diego Supercomputer Center," says Welch. "Now we are getting results about 130 times faster."

This allowed engineers to focus their time on adding parameters to the Advanced Dynamics of Rolling Elements (ADORE) analysis code. A major adjustment was needed since the code originally was tailored to model bearings that run at a constant speed, while Aerojet's bearings perform 4 million starts and stops per year.

Lee Akin of Aerojet adjusted the code to accommodate preload, which is the axial load that sets the bearings against a race and keeps them in contact and prevents skidding, minimizing wear. "If the preload causes too much force on the bearing, then the bearing will never perform properly, causing it to wear out. If the bearing is free to slip excessively, then it will wear out from too much sliding," explains Welch. Also, modifications were made to include factors such as the geometry of the bearing, temperature, material mismatch, and the effect of lubricants.

"All of these parameters had to be integrated into the ADORE code to help obtain an estimate of what was going to happen to the bearing after three years of life," says Welch. Aerojet engineers varied the parameters in 12 runs on the SDSC Cray computer system to finalize the bearing design.

"The Cray system brought sophisticated analysis into a timely regime. It meant the difference between doing the analysis or not being able to do it at all."

"The result is that we have a very sophisticated bearing analysis code to run on a Cray system," says Welch. "The real savings is the reduction of risk. We have analyzed the worst case, and it looks fine." He adds. "The Cray system brought sophisticated analysis into a timely regime. It meant the difference between doing the analysis or not being able to do it at all."

A project that's off the beaten path

Wake up! It's Monday morning, 1998, and you're late for work again. So throw on your clothes, grab some toast, and climb into your car. O.K., now that you've secured the seat belt and wiped the crumbs from your face, it would be a good idea to check local traffic conditions on your dashboard video screen.

"It was not until we had the memory and speed of the Cray system that we were able to exercise our models in sufficent detail."

Just punch in your destination, and the computer will determine an efficient route with the least amount of congestion. As you travel toward your work place, the computer provides traffic updates every five minutes, warning you about accidents and shifts in traffic patterns. With the help of your dashboard traffic-monitoring system, you arrive at work with time to spare.

This electronic route-monitoring and guidance system is the vision of David Boyce, the director of the Urban Transportation Center at the University of Illinois, Chicago. Boyce is laying the groundwork for the project by working with specialists in mathematical programming and computer science to model traffic flow in urban areas.

Although transportation planners have attempted to model highway and mass transit systems since the mid-1950s, until today computers were not powerful enough to tackle the problem, says Boyce. "It was not until we had the memory and speed of the Cray system that we were able to exercise our models in sufficient detail," he explains.

Using the CRAY X-MP/48 computer system at the National Center for Supercomputing Applications at the University of Illinois, Urbana-Champaign, Boyce and his colleagues are able to examine extremely complex traffic networks with a mathematical technique called nonlinear programming, a method for optimizing a function subject to constraints. Boyce is using data from the Chicago Area Transportation Study to develop a model that divides the Chicago region into 1800 zones and 30,000 links, or segments of road. This model will be refined enough to study networks of arterial streets and highways.

Boyce and his colleagues not only develop transportation models, but they help other urban planners fine-tune their models. "We work hand-in-glove with transportation planners and planning agencies in large cities, particularly Chicago," says Boyce. "They use these models routinely for evaluating proposed additions or changes in transportation networks."

Recently, the Regional Transportation Authority serving northeastern Illinois asked Boyce's group to help examine the implications of discontinuing service over old parts of the rapid transit system in the Chicago region. New lines had been created, but older, redundant lines had not been closed. Boyce explains, "They wanted to look at how the people presently using the lines would react. Would they use the new lines? Would they switch to their cars? If they switched to their cars, how would that affect congestion?"

Boyce and his colleagues, together with CATS, evaluated several alternate line closings. The results were the basis of a financial analysis that compared the cost of keeping those lines in operation to the benefits the lines would provide to users.

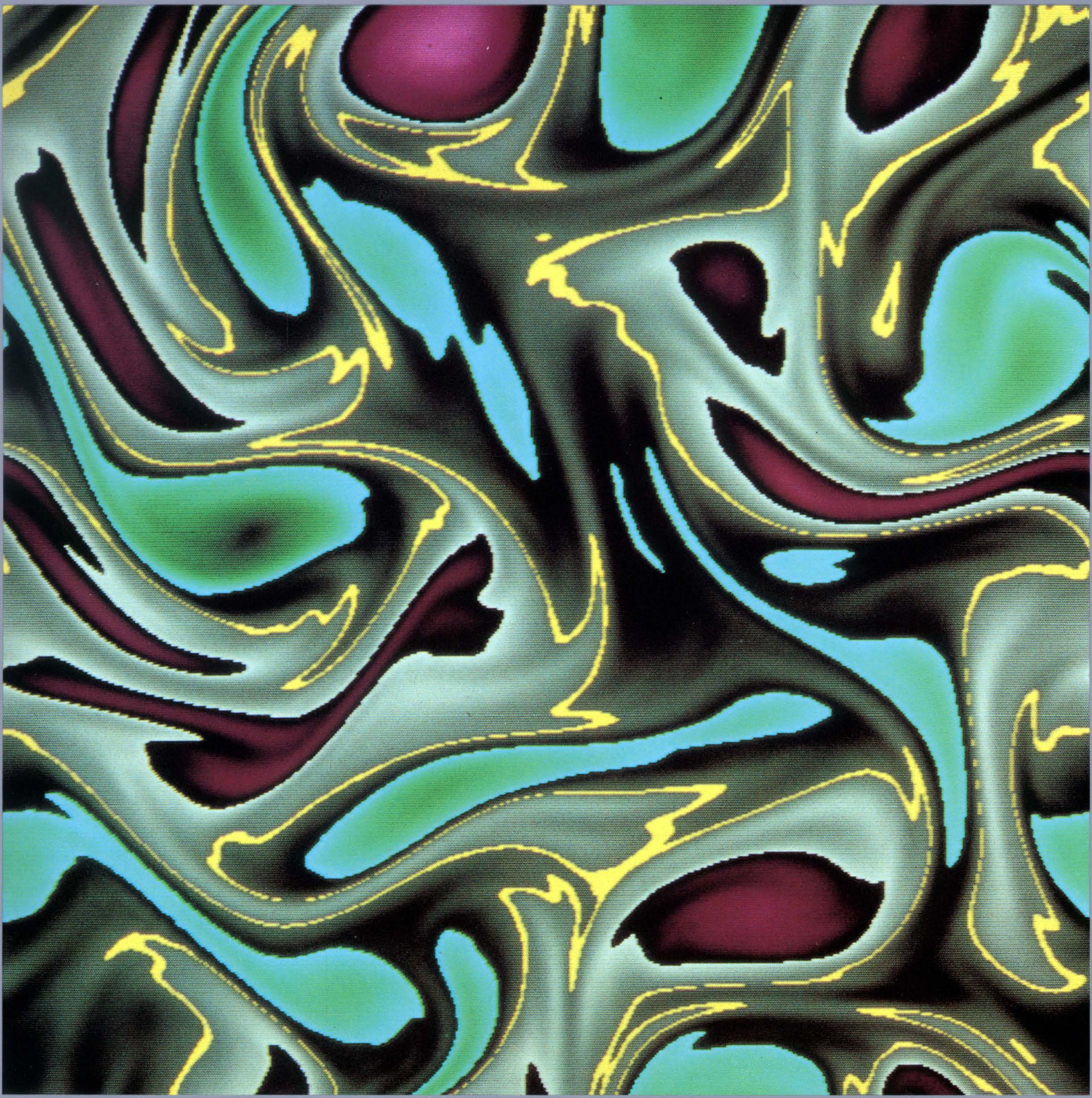
UI graduate students also are using models to consider conflicting traffic flows at intersections and optimal additions to network capacity. The focus of Boyce's future work is electronic route-monitoring and guidance systems that will help commuters determine their most efficient departure times and routes.

"There would have to be a very large central computer, like a CRAY-2 or its successor, that would continuously or periodically solve a very large network problem that would determine which routes vehicles in the network would use," says Boyce. This information would be transferred over a communications network such as a cellular telephone system, then transferred to video screens in vehicles.

"We also have thought about how we could use historical information from yesterday or last month to predict what would happen in the next hour or evening." He adds that the United States Department of Transportation has initiated a substantial contract to evaluate the potential for such a system.

"Actually, the Department of Transportation was working on the development of such a system in the 1960s — probably not realizing that the electronics for it did not exist," Boyce says. "With the supercomputer program at the National Science Foundation and the development of technology by Cray Research, now we have the resources to move ahead on this."

GALLERY



Potential vorticity field showing vortices without inertio-gravity waves, from a study of compressible two-dimensional turbulent flow. The 512-by-512 image was computed in about two CPU hours on the CRAY-2 system at the Centre de Calcul Vectoriel pour la Recherche (CCVR) in Palaiseau, France. The image was created by solving Saint-Venant's equations using a pseudospectral code developed by Marie Farge of the Ecole Normale Supérieure in Paris. The visualization was performed at LACTAMME in Palaiseau, on a prototype image processor developed by Jean-François Colonna of the Ecole Polytechnique and the Centre National des Télécommunications (CNET) using a normalized color scale adapted to the visualization of two-dimensional turbulent fields. © Marie Farge and Jean-François Colonna. Please send Gallery submissions to CRAY CHANNELS at the address inside the front cover.