CRAY CHANNELS

SCIENCE AND ENGINEERING SYMPOSIUM

Announcing the CRAY XMS minisupercomputer

CRAY CHANNELS

In this issue

A meeting of minds — supercomputing minds — convened from October 22 to 24, 1990, in London, England. The event was "Science and Engineering on Cray Research Supercomputers," the fifth international supercomputing symposium sponsored by Cray Research. Fifty presenters shared with the attendees results of their scientific and engineering projects involving Cray Research super-computers. The presentations demonstrated the advanced problem-solving capabilities of Cray Research systems in diverse fields of research, ranging from industrial engineering to weather forecasting and medical diagnostics.

In this issue we present five papers from the symposium, edited for CRAY CHANNELS. Our intent is to convey the breadth of application areas in which Cray Research supercomputers have become indispensible tools for acquiring knowledge and enhancing productivity. In this issue we also introduce Cray Research's first minisupercomputer, the CRAY XMS computer system. Our regular departments cover new networking technologies, a finite element analysis program, and the use of Cray Research systems to model metal forming, food processing, and potroom ventilation.

Cray Research regularly hosts symposia and forums to bring together users working on leading-edge research and engineering projects. These meetings give users a face-to-face opportunity to share research results, new algorithms, and programming techniques. The community of Cray Research system users constitutes a scientific and engineering elite, and Cray Research welcomes opportunities to facilitate the flow of information through this select community.

Features

2

Departments

Magnetic resonance image interpolation for three-dimensional data analysis and display

Raleigh F. Johnson, Jr., and Donald G. Brunder, University of Texas Medical Branch, Galveston, Texas

Magnetic resonance imaging (MRI) is one of the more informative medical imaging technologies. Its sensitivity to soft tissues, including tumors and other tissue abnormalities, makes it an invaluable tool for the diagnosis of certain conditions. MRI produces a series of two-dimensional images that represent anatomical slices of a finite thickness. Viewed on a computer display at a resolution of 256 by 256 pixels, the interpixel spacing represents approximately 1 mm. Each image, though only one pixel thick, represents a 3 to 10 mm-thick anatomical slice. Some MRI techniques include a gap or space between the slices; the third dimension (between image slices) thus is represented only by the number of slices acquired, which typically is less than 32. MRI is a unique medical imaging technology in that the orientation of the acquired image data can be selected along each of the principal orthogonal planes — coronal, sagittal, transaxial — and oblique imaging planes of the anatomy, as described by Johnson.[1-3] Each orientation requires a separate data acquisition. MRI can produce true three-dimensional data that can be viewed from any angle and with a spatial resolution nearly equal in all directions. However, this three-dimensional imaging method is very time consuming and physically uncomfortable for routine clinical use.

Multi-slice, two-dimensional imaging is routine clinically, and data interpolation methods can be used to produce a three-dimensional data set from two-dimensional data slices. But many investigators, including Udupa, who have used interpolation techniques for medical images, have dealt only with two-dimensional images acquired in one orientation, as with computed tomography.[4] The data interpolation method described here takes advantage of the multiorientation data acquisition capability of MRI. Images of a cadaver heart were recorded from three orthogonal orientations, and the interpolation methods were developed to compare the three-dimensional volume data sets derived from interpolation between images when considering one, two, and three orientations for linear interpolation. The algorithms for image interpolation, display, and analysis were written originally for Sun Microsystems and Silicon Graphics workstations, then were adapted to run on the CRAY X-MP/14se computer system, running the UNICOS operating system, at The University of Texas Center for High Performance Computing in Austin, Texas.

The three-dimensional interpolation method described here has evolved from a more comprehensive project to develop three-dimensional image display techniques, including spectroscopy and animation of MR images, particularly of the heart. The magnitude of the data computations requires the speed of supercomputers, such as Cray Research systems. The interpolation method described here produces a volume array of pixel data that represents the heart. The acquired image data are ported to the Cray Research system for all of the complex image computations including image data interpolation. The ultimate goal of the research is to provide the capability to produce stereoscopic image pairs of the heart seen from any spatial orientation and animated to present the organ for clinical diagnosis and review. Video recordings of the animation sequence would be made available for review, providing the physician with a stereoscopic, three-dimensional, and dynamic display of the heart and other interesting regions of the anatomy.

Figure 1. Linear interpolation between "real" image slices.
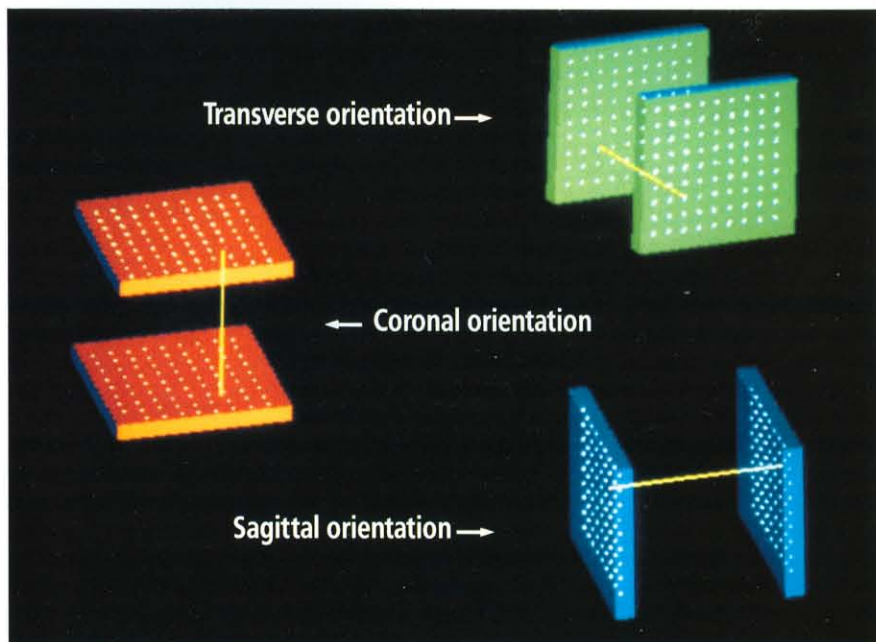
The purpose of the study was to assess the feasibility of constructing a three-dimensional data set from two-dimensional MR images acquired in one or more orthogonal orientations and to render and display the resulting three-dimensional images from any angle. Furthermore, an algorithm was developed to model the MR imaging method for simulating the two-dimensional data acquisitions, which provided a means for evaluating the interpolation methods. Unilinear, bilinear, and trilinear interpolations were evaluated by visual inspection of the resulting images and by computing the volume of the space between two concentric spheres of different diameters that were processed by a series of simulated MR imaging acquisitions performed by the MR imaging model.

Heart imaging

Multi-slice MR images of a cadaver heart were acquired in the transaxial, coronal, and sagittal orientations. The dimensions of each image were 256 by 256 pixels of "real" (acquired, as opposed to interpolated) data. The acquired MR data sets comprised 20 to 24 slices from each orientation. The heart images in the transaxial orientation represent a short axis view showing the muscular left ventricle as a doughnut-shaped chamber. The coronal and sagittal views demonstrate the cardiac anatomy parallel to the long axis of the heart. The in-plane resolution is 1 mm per pixel. For each orientation, the between-plane resolution in the third dimension is limited to 10 mm per slice.

The heart was imaged using a 0.6 tesla superconducting MRI system. (The cadaver heart was placed in the center of the magnet with the heart oriented such that the transaxial image would correspond to a cardiac short axis view.) The imaging pulse sequence was a spin echo with a repetition time, TR, of 500 msec and an echo time, TE, of 32 msec. These timing parameters affect image and data contrast. Optimal contrast is essential for distinguishing various tissues, including abnormal ones.

A cadaver heart was imaged to eliminate the need for synchronizing the images with the electrocardiogram. Furthermore, the three-dimensional image display and analysis methods developed using the current data are planned for adaptation to routine cardiac analysis in live patients. Interpolation of MR images to form three-dimensional volume data is expected to shorten the imaging procedure for clinical use and result in the display of dynamic three-dimensional cardiac images for enhancing clinical diagnosis.

Interpolation method

The interpolation between images includes simple linear interpolation between similar image data points contained in "real" slices acquired in a single orientation. The slope of a linear gradient is calculated between two "real" pixels and the estimated value of a specific pixel between them is interpolated along the gradient. Figure 1 illustrates the linear gradient between corresponding pixels between slices of "real" data. Interpolation is complete when a volume data set is constructed with spatial resolution equal in three directions and with dimensions of 256 by 256 by 256 pixels. The unilinear interpolation method results in a volume

Figure 2. Trilinear interpolation.
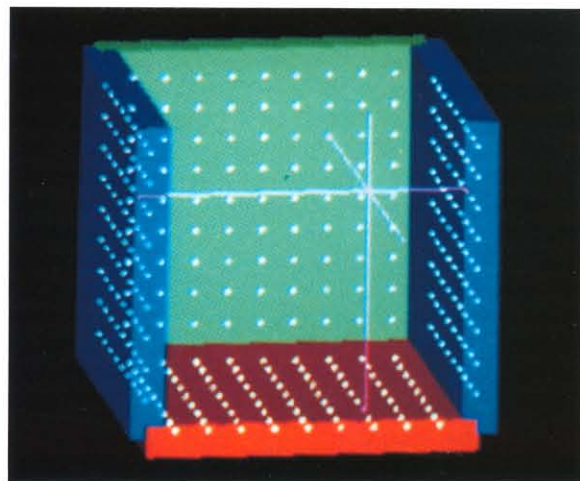
data set calculated from interpolations acquired in a single orientation. Bilinear interpolation is accomplished between images acquired in two orientations. Three orthogonal image orientations are included in the trilinear interpolation method, which represents the most complex method considered.

Enhancement of the interpolation performance was evaluated by considering the influence of additional neighboring pixels on the interpolated values. This effectively smooths the data and allows better results from edge-tracking algorithms used for three-dimensional image display.

Interpolation in three dimensions is illustrated in Figure 2, in which the pixel of interest, which lies between "real" image slices, is estimated by the linear gradients between three pairs of pixels. The interpolated value of the pixel determined by considering two slices is averaged with the computed values from the other pairs of slices. The volume data set then is used to perform the three-dimensional image display and analysis.

Figure 3 illustrates the image volume in which the "real" slices intersect and form a three-dimensional matrix of acquired image data from which the

Figure 3. Trilinear interpolation volume.

interpolation is performed. The intersection of images is especially important. The in-plane pixels may represent 1 mm of spatial resolution. However, the image is a single pixel thick, although the slice has a thickness of several millimeters. The pixel that represents the image thickness is the average value of several pixels when the same region is imaged in an orthogonal view. Thus the intersection of the orthogonal planes provides information about the values of the pixels that represent slice thickness.

MR image simulation

An MR imaging simulation algorithm was developed to evaluate the performance of the interpolation methods. A spherical phantom was simulated as a three-dimensional object contained within the three-dimensional volume represented by 256 by 256 by 256 pixels. The phantom was a test data set consisting of two concentric hollow spheres with different radii. The phantom was constructed by applying the formula for a sphere with no anti-aliasing performed. Pixels contained within the data volume were assigned known values.

The data volume was sampled to simulate the acquisition of the MRI system. The samples were multiple 5 mm slices each separated by a 1 mm gap. The average pixel value in each slice was calculated by integrating the pixels over the thickness of the slice. The sampling resulted in three sets of 20 to 40 slices representing the acquisition of MR image data from each of the three orthogonal orientations. Figure 4
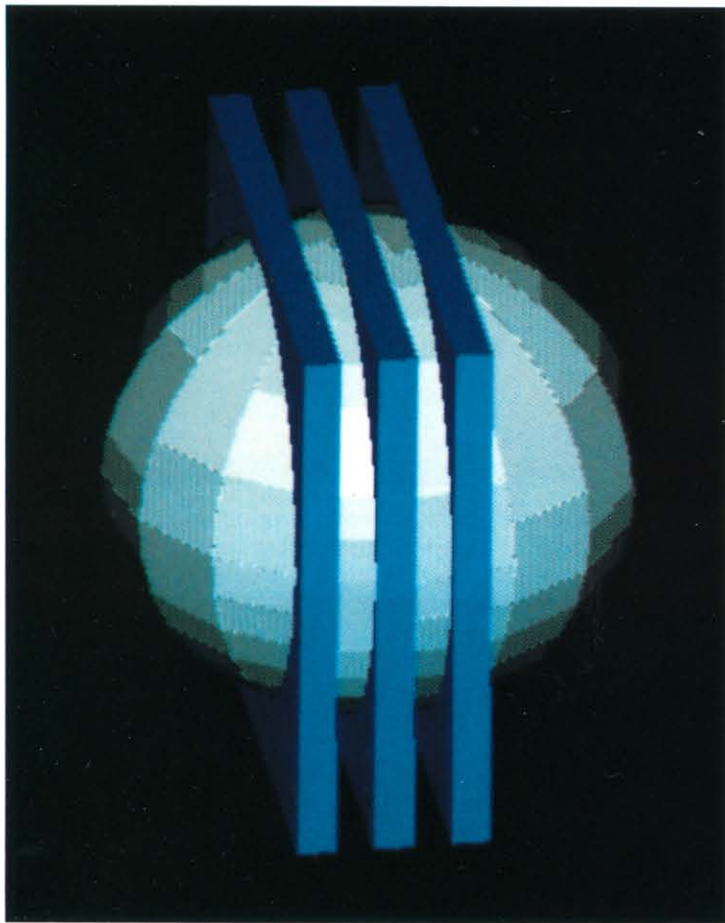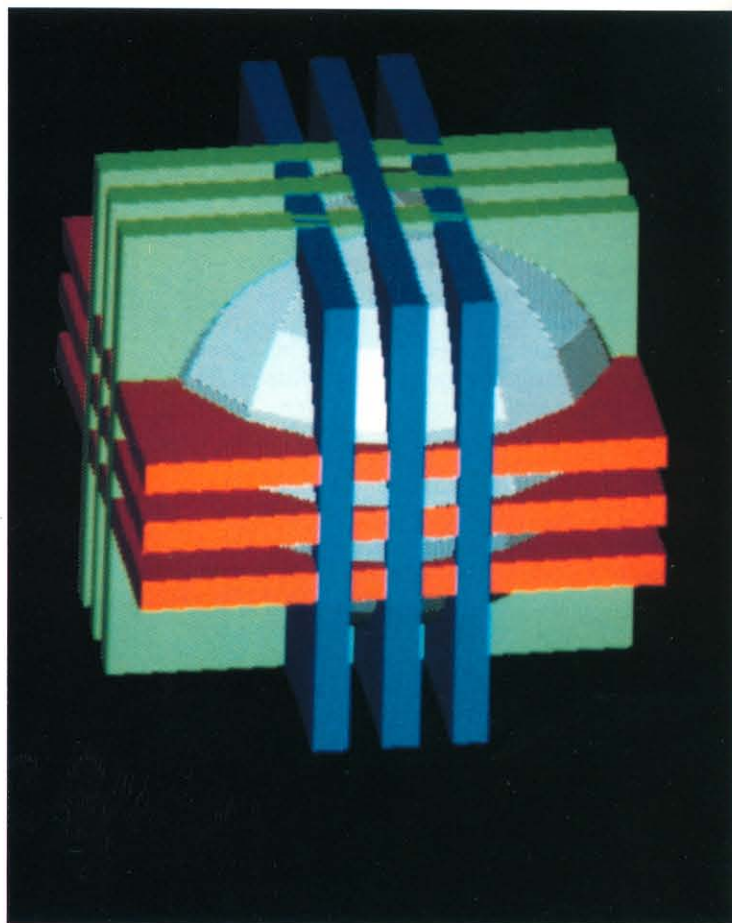
Figure 4 (left). MR image simulation of the spherical phantom from a single orientation.

Figure 5 (right). Trilinear MR image simulation and interpolation.

illustrates the MR imaging simulation from a single orientation of the slices through the spherical phantom. Each slice contains pixel data that represent the average throughout the slice thickness. Image interpolation was performed using the three methods previously described. The interpolated results would produce the three-dimensional data set that would be compared with the known data in the original volume data set. Each of the interpolation methods would be evaluated by comparing the results of a quantitative measurement, such as the volume of the spherical phantom, between the spheres. Figure 5 illustrates the trilinear imaging simulation and interpolation.

Volume statistics were calculated to assess the interpolation quality. Visual inspection of the three-dimensional images was performed to compare the methods. The marching cubes algorithm described by Lorensen and Cline[5] was used to render the volume data sets for display as three-dimensional images. This technique analyzes polygons formed by groups of adjacent pixels and determines the intersections of the polygons to form a surface.

Results

Interpolation of MR image data resulted in a volume of image data with dimensions of 256 by 256 by 256 pixels and equal spatial resolution in all directions. The images acquired of the heart were interpolated and rendered as three-dimensional images. Evaluation of the interpolation method was accomplished mainly by visual inspection of the three-dimensional
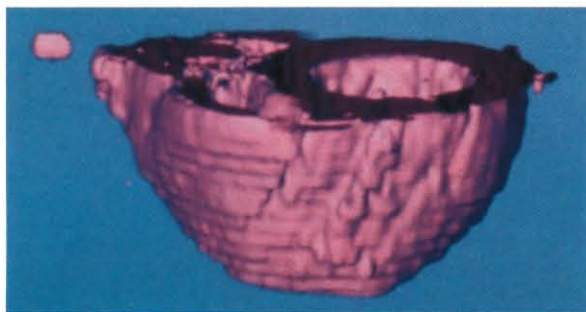
Figure 6. Section of three-dimensional heart rendered from unilinear interpolation.
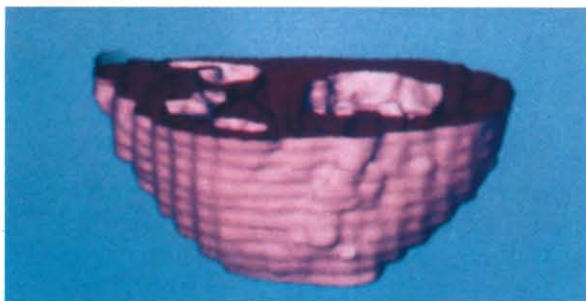
Figure 7. Section of three-dimensional heart rendered from bilinear interpolation.

Figure 8. Section of three-dimensional heart rendered from trilinear interpolation.

Interpolation results

Interpolation method	Thickness (mm)	Slice gap (mm)	Volume (cm³)	Error (%)
Unilinear	8.0	2.0	1596	−1.3
Bilinear	8.0	2.0	1612	−0.3
Trilinear	8.0	2.0	1613	−0.2
Unilinear	6.0	1.5	1595	−1.3
Bilinear	6.0	1.5	1606	−0.7
Trilinear	6.0	1.5	1609	−0.5
Unilinear	4.0	1.0	1617	0.0
Bilinear	4.0	1.0	1612	−0.3
Trilinear	4.0	1.0	1617	0.0

Table 1. Volume measurements determined by interpolation methods.

Conclusion

MRI data interpolation in three orthogonal orientations (trilinear method) provides an improved volume data set from which three-dimensional images can be generated. MRI is unique in medical imaging in that data (multislice, two-dimensional images) can be acquired from any orientation in a reasonably short time. Orthogonal images can be trilinearly interpolated and formed into three-dimensional images for enhanced clinical use. ▄

images. The spherical phantom for simulation of MR imaging was inspected visually also and was quantified to compare the volume of the spheres as determined by the three interpolation methods.

Heart image evaluation

The three-dimensional images of the heart were compared for relative smoothness. The unilinear interpolation method was used to produce Figure 6. The aliasing from the relatively few image slices is particularly noticeable in the horizontal plane. Figure 7 shows the results of bilinear interpolation. In this three-dimensional image, aliasing is obvious in both the horizontal and vertical planes. The best three-dimensional image of the heart was produced with the trilinear interpolation method, as shown in Figure 8. The image is smoother and more realistic than the others.

MR imaging simulation

The spherical phantom images were inspected visually, and the trilinear interpolation method gave smoother and more pleasing three-dimensional images. The motivation to develop this simulation algorithm was to provide the interpolation methods with a series of images from which parameters such as volume and surface area could be quantified. Table 1 lists the measurements of spherical volume for each of the interpolation methods. In each comparison of three slice thicknesses (8, 6, and 4 mm) the trilinear interpolation gave better results. One would expect thinner slices to yield better results. The MRI simulation model illustrates this, as seen in Table 1.

About the authors

Raleigh F. Johnson, Jr., is the director of the Magnetic Resonance Imaging Division of the Radiology Department at the University of Texas Medical Branch in Galveston, Texas. He received a B.A. degree in physics from Berea College in Berea, Kentucky, an M.S. degree in radiological physics from the University of Miami, Florida, and a Ph.D. degree in radiological physics and bionucleonics from Purdue University in Lafayette, Indiana.

Donald G. Brunder is a programmer/analyst for the Office of Academic Computing and Biostatistics at the University of Texas Medical Branch in Galveston, Texas. He obtained a B.S. degree in physics from Bemidji State University in Minnesota in 1974 and a Ph.D. degree in biophysics and neuroscience from Michigan State University in 1981.

References

1. Johnson, R. F., Jr., D. J. Dornfest, W. J. Prevost, Jr., and M. Ahmad, "Oblique NMR Myocardial Infarct Evaluation," *Journal of Nuclear Medicine,* Vol. 27, p. 312, 1986.
2. Johnson, R. F., Jr., M. Ahmad, E. G. Amparo, D. J. Dornfest, and W. J. Prevost, Jr., "Oblique Magnetic Resonance Imaging for Myocardial Infarct Evaluation," *Journal of Nuclear Medicine,* Vol. 27, p. 986, 1986.
3. Johnson, R. F., Jr., M. Ahmad, D. J. Dornfest, and W. J. Prevost, Jr., "Oblique Magnetic Resonance Imaging of the Heart with Cycle Gating for T2 Measurements," *Proceedings of The Society of Magnetic Resonance In Medicine,* New York, p. 768, 1987.
4. Udupa, J. K., "Computer Aspects of Three-Dimensional Imaging in Medicine: A Tutorial," *Proceedings of 3-D Imaging In Medicine,* Hospital of The University of Pennsylvania, Ch. 1, 1989.
5. Lorensen, W. E. and H. E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Computer Graphics,* Vol. 21, pp. 171-179, 1987.

Simulation of in-cylinder flow and combustion in reciprocating engines

Ben Adamson, David Gosman, Charles Hill, Chris Marooney, Behrooz Nasseri,
Michael Sarantinos, Theodor Theodoropoulos, and Henry Weller
Imperial College of Science, Technology and Medicine, London, England

Because industrial researchers and engineers require precise solutions to their technical problems, computational fluid dynamics (CFD) methodologies must accommodate real flow geometries. In many cases, including that of the reciprocating internal combustion engine, these geometries are difficult to mesh with conventional regular grids, which are susceptible to distortion. Unstructured body-fitted meshes best address such cases because they provide the flexibility to fit boundary and flow features without introducing large amounts of non-orthogonality into the mesh. This meshing strategy is used in the SPEED (Simulation Programme for Engine Evaluation and Design) code developed at Imperial College.[1] The development of the SPEED code has involved not only the adaptation for use on unstructured meshes of the finite-volume discretization method (FVM) and the conjugate gradient (CG) solution method for matrix equations, but also techniques for creating the mesh, describing its continuity, moving and restructuring it in response to boundary and flow changes, and visualizing the four-dimensional property fields generated by the code. The applications of the code described here were run on the CRAY X-MP/28 computer system at the University of London Computer Centre.

Discretization, numerics, and modeling

The SPEED code uses the finite-volume method to discretize the CFD equations for mass, momentum, internal energy, turbulence, and the chemical species for ignition and combustion. The equations are solved in a fully implicit manner so that large time steps can be used. This implicitness is necessary to accommodate the large computational demands made by engine simulation codes. These demands exist due to two primary factors: (1) the variable sets for the models of turbulence, fuel-spray, droplet motion, and combustion chemistry, as well as the variable set for the basic aerodynamic flow, are large and will increase as more detailed models come into use, and (2) the unstructured mesh requires a database of addresses that describe its connectivity.

This requirement, plus the larger supporting storage needed for efficient computation with unstructured meshes, also yields a large variable set. In combination, these factors create a substantial memory requirement, and as a result CPU time must be minimized to achieve an acceptable turnaround time.

The interactions among the various models are represented in a predictor-corrector manner after the PISO algorithm of Issa.[2] Corrector steps are solved neighbor-explicitly; that is, in the correction of a given node value, its neighbor's values are set to the latest values already available. For neighbor-explicit calculations, the algorithm is highly vectorizable because in an unstructured mesh the vector length equals the mesh size; the standard version of the code already is conditioned suitably for vector machines in this way. For parallel processing by subdomain decomposition, this part of the code also is well-conditioned for Cray Research hardware architectures, provided the neighbors on the subdomain boundary are duplicated in the data. The major architecture-conditioning challenge comes from the fully implicit predictor stages. The solution of the unstructured sparse matrix equations generated for these messages is best accomplished by a CG solver, which in principle is structurally neither vectorizable nor parallelizable. However, we have found that if the mesh node numbering is preconditioned such that the difference between neighbor numbers exceeds the vector length of the machine, then the CG solution algorithm can be substructured to yield a vectorizable inner loop, though the increased bandwidth increases the iterations to solution by about 20 percent. The technique is not available in parallel architectures, but research continues in this area.

For dispersed-phase calculations, Lagrangian equations of motion for a representative sample of droplets are used to model atomization, evaporation, breakup, and coalescence. Most of the Lagrangian code is independent of the mesh structure, and the extension to unstructured meshes involves defining a method of particle tracking. In the SPEED code, a local coordinate system is defined for each cell, on which a conventional search procedure can be used.
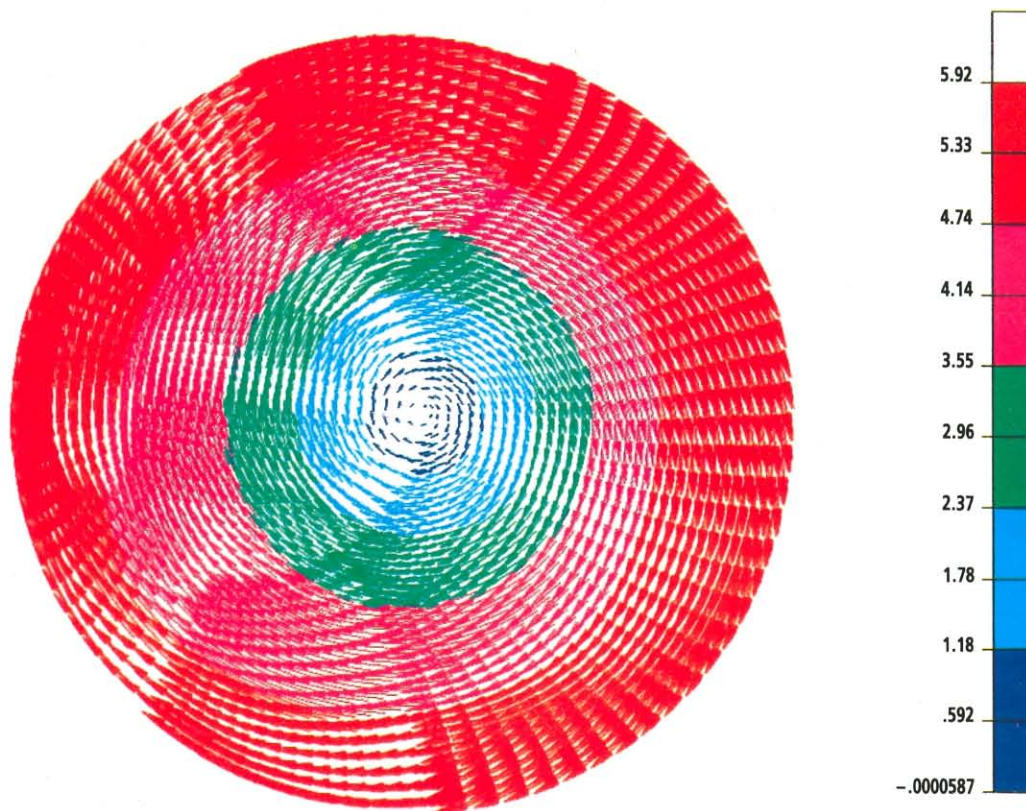
Figure 1. Disc chamber engine calculated swirl velocity field at top-dead-center.

Color scale values:
5.92
5.33
4.74
4.14
3.55
2.96
2.37
1.78
1.18
.592
– .0000587

Sample engine results

The engine calculations described here illustrate the range of phenomena that can be calculated with the SPEED code and the visualization options available. Figure 1 shows the top-dead-center swirl flow in a disc chamber geometry. The vector plot display of velocity demonstrates the value of color in decoding the basic velocity magnitude information conveyed by arrow length. Comparisons of calculations with experimental results are shown in Figure 2. The significant improvement seen in the swirl velocity comparison on moving from a coarse mesh of 8000 cells to a fine mesh of 30,000 cells is expected from the properties of the upwind differencing technique used to model convection. A similar improvement in the turbulence intensity (not shown) is due to improved resolution of the generating flow gradients.
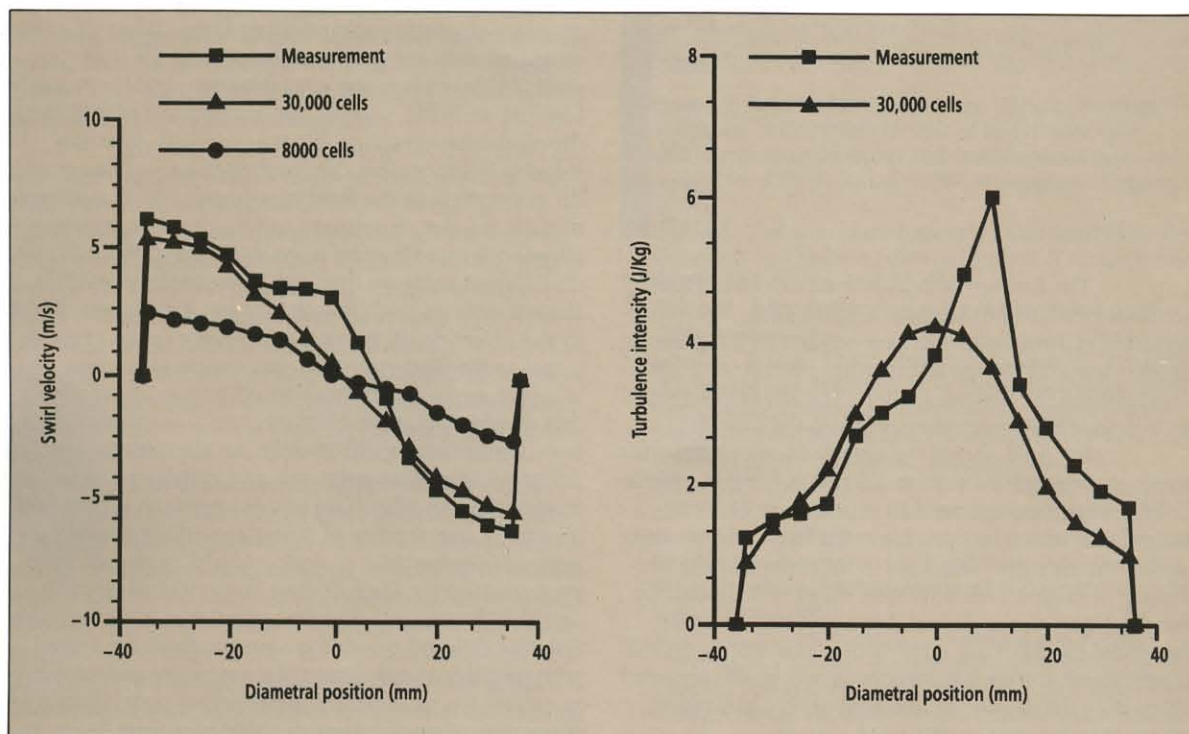
Figure 2. Comparison of measurements and computational results for top-dead-center compression in a disc chamber engine.
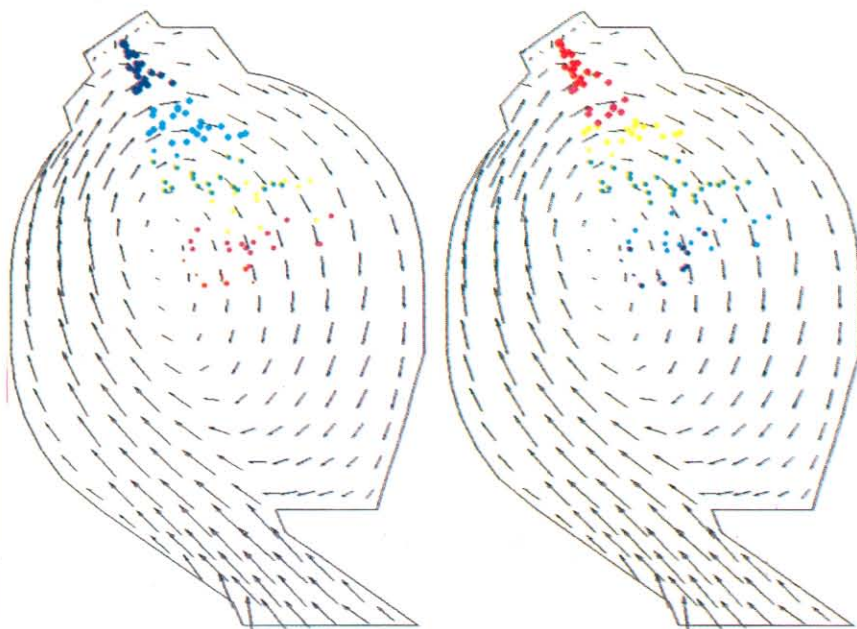
Figure 3. Vertical section through diesel prechamber showing swirl flow field and fuel spray colored for temperature (left) and drop size (right).

6.87-02
6.87-02
6.86-02
6.85-02
6.85-02
6.84-02
6.83-02
6.83-02
6.82-02
6.81-02
6.81-02
6.80-02
6.79-02
6.79-02
6.78-02
6.77-02

Figure 4. Vertical section through Renault F1800A spark-ignition engine showing onset of knock by depletion of fuel concentration in unburnt gas.

The fuel-spray calculations shown in Figure 3 are for a prechamber without a spark plug. The representation of the droplet flow, color coded for temperature in one view and for size in the other, reveals the correlation between the two parameters. The spray is imposed on a monochrome representation of the flow field.

Figure 4 shows the results of calculations for spark-ignited combustion in a Renault F1800A engine. Color fringe plots are used to visualize an Eulerian scalar field, which in this case is the fuel concentration in the unburnt gas. This field is important in the prediction of engine knock, whose onset is indicated by the depletion zones shown. Measurements suggest that these calculations agree with observations on the actual engine and are improvements over results obtained using only the ensemble-average fuel concentration, where knock is not predicted.[3,4]
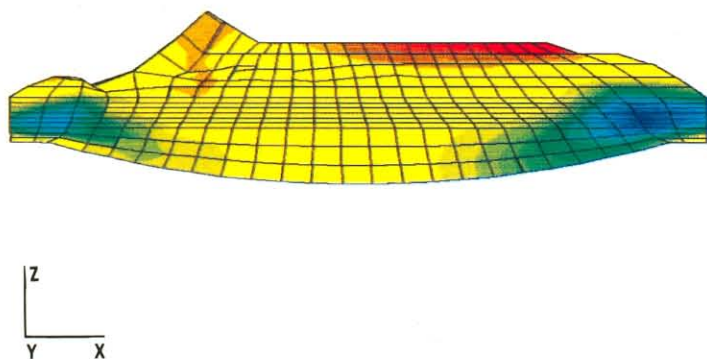
Figure 5 illustrates the three-dimensional complexity that can be obtained for the aerodynamic flow fields. The figure shows one axial and two diametral sections through the bottom-dead-center velocity field in the bowl-in-piston geometry. These sections through the flow were selected to make comparisons with measurements for this engine.[5,6] Evidently these sections do not exhaust the full complexity of the flow field, nor do they bring out changes over time. They thus point to the need for a more effective way to visualize the results.

Visualization of results

The plots of Figure 5 provide the basic information needed to evaluate the flow evolution. To understand how the flow fields evolved to this point, however, it is necessary to examine a temporal sequence of plots, using as many sections as required to map the features of the flow. Because such an evaluation can lead to the user drowning in a complexity of data, techniques are needed to package and summarize this information more directly, the most obvious one being animation. Once the problems of data storage (usually on the order of 10^5 points for each of 100-300 time steps) and automatic extraction of relevant data subsets are overcome, viewing the evolution of the flow dynamically is the most direct way to assimilate the code output.

Animated films have been made of the flow in the prechamber geometry, the evolution of a fuel spray injected into this flow, and the evolution of the flame front in a spark-ignition engine. The films were made from a sequence of frames taken directly from the calculations. An alternative method is to process the data to provide defined objects, which then are animated. This reduces storage by ensuring that only the objects of interest are stored and allows the animation package to improve image smoothness by appropriate time interpolation using the object definition.

A comparison of these films shows that the animated data are easy to comprehend for Eulerian scalar fields, such as fuel concentration, and for properly Lagrangian fields, such as the distribution of fuel spray droplets. However, a problem arises with Eulerian velocity fields, in that the fixed positions of the arrows do not represent the fluid motion very well. Although there is no simple solution to this problem, many alternative visualization techniques exist. For example, an Eulerian solution involves filling each cell with a striped pattern, such that the stripes are perpendicular to the direction of motion, and have a speed proportional to the flow speed. In this case it is unclear whether a minimum mesh resolution exists at which this display would be effective, and it is in any case a two-dimensional method only. An alternative is to adopt a Lagrangian approach and display particle trajectories. An advantage to this approach is that, with a suitably low density of particles and appropriate perspective, the time evolution could be shown in a static picture for the full three-dimensional domain. The disadvantage is that the particles misrepresent an Eulerian field by providing a nonuniform coverage, with particles accumulating in some places and depleting in others. A technique for providing generation and extinction of tracker particles also is required.
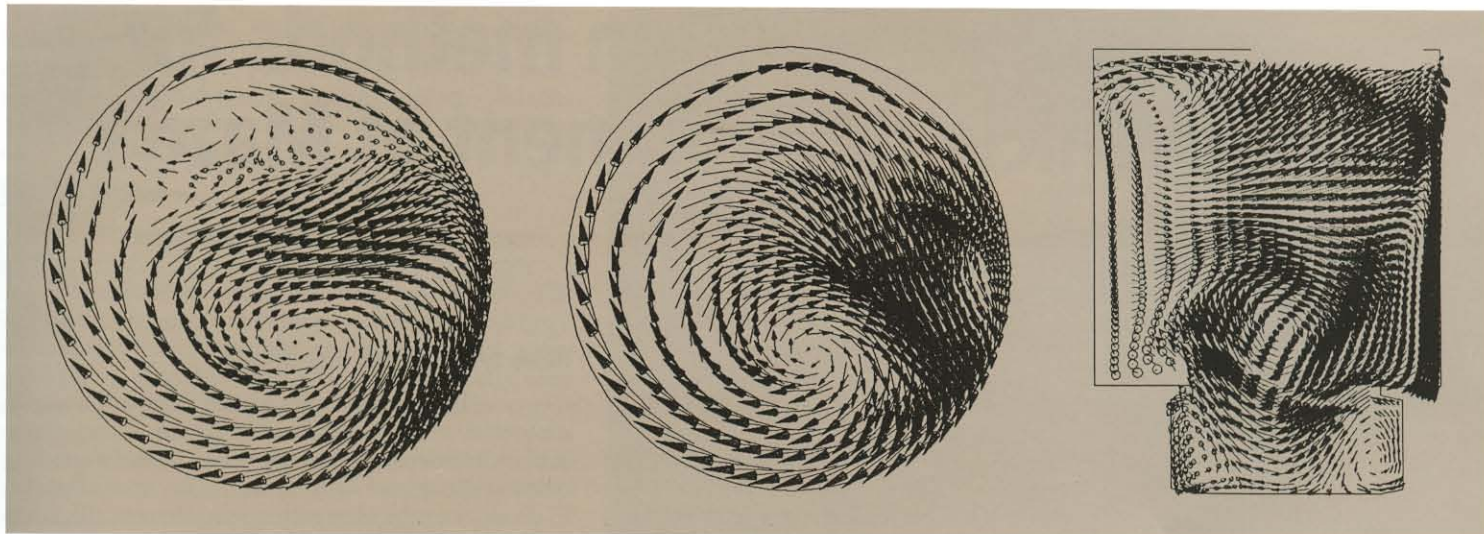
Future developments

As the number and detail of interacting models proliferate, major emphasis will be placed on improvement of the solution algorithm for the inter-action and on higher order discretization schemes to make the code more efficient. However, improvement of the output data evaluation process will remain a major area of development. The present state of development of the SPEED code relies heavily on existing standards in visualization, and it will have to be moved outside the existing capabilities of the PATRAN CAD system and of such commercially available systems generally. Although commercial CAD systems historically have been based in steady-state, often structural-mechanics based, applications, the industry increasingly is providing tools for time-dependent fluids applications. Nevertheless, the alternative of a tool-kit environment, which provides a standard graphics base on which the user can build the required applications, seems to be a more appropriately flexible way to explore visualization options, because the subjective worth of the end product is a major consideration and cannot always be evaluated prior to its realization.

Similar comments apply to mesh genera-tion. Although we would prefer to use the advancing front option, in view of its flexibility in the body-fitting of meshes, commercial packages with the full range of visualization facilities including this option are not yet available. In addition, the problems of closure of advancing fronts are more severe for hexa-hedral meshes than for tetrahedral meshes. However, this option provides the most straightforward means of mesh reconstruction for dynamic local refinement, and, when more widely available, is likely to become the standard.

All of the visualization possibilities men-tioned so far presume that a two-dimensional section of a three-dimensional field has to be observed as a whole. Alternative strategies based on viewing streamline trajectories as two-dimensional (space and time) objects also could provide a way of visualizing the flow. Here research is needed not only into the graphical represen-tation options, but also into the way that the mind understands such data fields. We hope that the existing successful application of the SPEED code environment to engine problems will provide further impetus for the development of such methods, which will be appli-cable to a wide range of problems. ▬

About the authors

The authors are faculty members in the fluids section of the Department of Mechanical Engineering at the Imperial College of Science, Technology, and Medicine in London.

References

1. Gosman, A. D. and C. J. Marooney, "Development and Validation of Computer Models of In-cylinder Flow and Combustion in Diesel and Spark-ignition Engines," C399/169, *IMechE*, Autotech '89, Birmingham, England, 1989.
2. Issa, R. I., "Solution of the Implicitly Discretized Fluid-flow Equations by Operator Splitting," *Journal of Computational Physics*, Vol. 62, pp. 40-65, 1986.
3. Zellat, M., Régie Nationale des Usines Renault, private communication.
4. Gosman, A. D., C. J. Marooney, and H. G. Weller, "Prediction of Unburnt Gas Temperature in Multidimensional Engine Combustion Simulation," International Symposium on Diagnostics and Modelling of Combustion in Internal Com-bustion Engines (Comodia), Kyoto, Japan, 1990.
5. Adamson, B., A. D. Gosman, C. J. Marooney, B. Nasseri, and T. Theodoropoulos, "A New Unstructured-mesh Method for Flow Prediction in Internal Combustion Engines," International Symposium on Diagnostics and Modelling of Combustion in Internal Combustion Engines (Comodia), Kyoto, Japan, 1990.
6. Vafidis, C., G. Vorropoulos, J. Whitelaw, and E. Nino, "In-cylinder Flowfield in a Motored Engine Equipped with a Helical Port and with Re-entrant and Square Piston-bowl Configurations," Imperial College, Department of Mechanical Engineering, Fluids Section Report F5/87/41, 1987.

Crash simulation methods for vehicle development at Mazda

Seiichi Ando, Koji Kurimoto, and Koji Taga
Mazda Motor Corporation, Hiroshima, Japan

At Mazda, developing new crashworthy vehicles involves optimizing the body structure as well as the occupant restraint system. Because physically testing these factors is expensive and time-consuming, Mazda began to use large-scale vehicle crash simulation methods for design and optimization several years ago. Today, the company models crashworthiness on a CRAY X-MP supercomputer; a cost-effective, efficient way to develop new structures for vehicle safety.

Figure 1. A comparison of deformation characteristics (original condition).
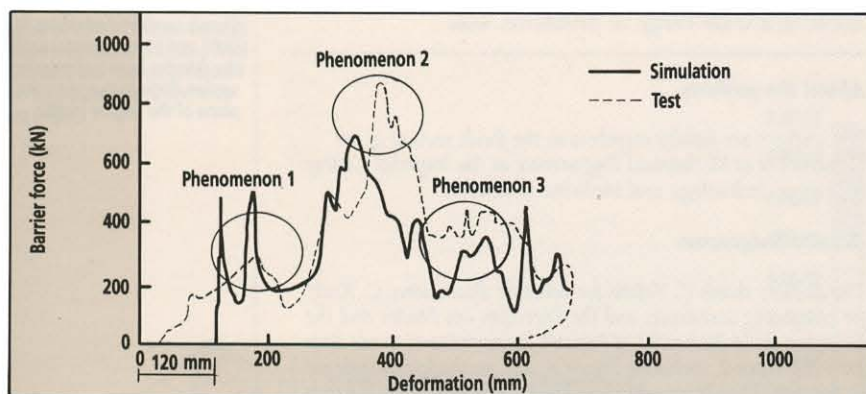
Figure 2. The effect of modifying a crossmember.
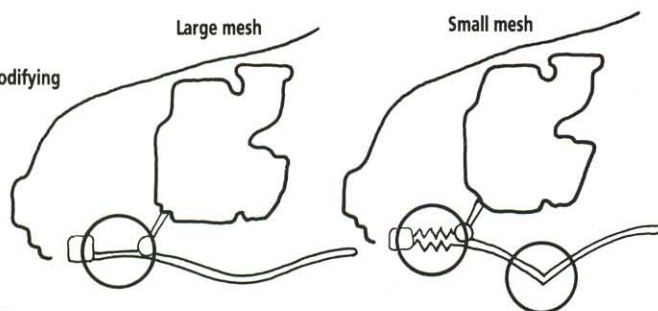
Figure 3. A comparison of deformation characteristics.
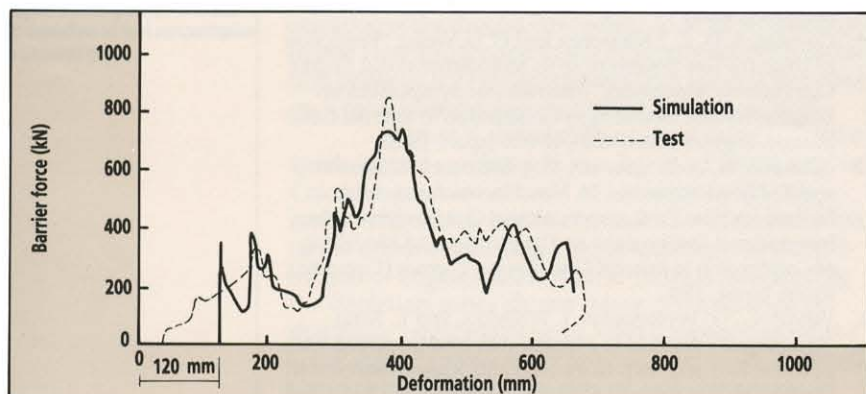
Three typical crash scenarios

Three of the most typical crashes simulated are frontal, rear, and side impact. The most important design consideration for high-speed frontal impact is lowering the impact force to reduce occupant injury. To do so, the vehicle structure must deform effectively to absorb energy during the crash. If the structure is too stiff, the body cannot crush, causing the impact force to be transmitted directly to the occupants. On the other hand, if the structure is too weak, the car body cannot absorb enough energy for the occupant compartment to be secure. Computer simulation is used to optimize the stiffness of the structure.

In the case of rear impact, the body structure should be designed to have enough strength and controlled deformation to satisfy two major requirements: survival space for the occupants and protection against fuel leakage. Designing the structure of smaller vehicles is difficult because they have a smaller crushable zone than do larger vehicles.

As with frontal impact, the most significant concern in side impact is the reduction of occupant injury. Side impact injuries are more difficult to prevent than frontal impact injuries, because seat belts and air bags are not as effective, and less crushable space is available to absorb the energy of the impact. Usually, more than 600 mm of crushable space is available for frontal crashes, but for side impacts this zone is only 200 mm or less. Therefore, the body structure and interior parts, such as a shoulder holster, must be highly optimized to reduce occupant injury from side impact.

Crash simulation methods

Mazda has used the PAM-CRASH simulation code since 1987.[1] PAM-CRASH is a three-dimensional Lagrangian explicit finite element program for analyzing the nonlinear dynamic response of structures. It is designed specifically for automotive crashworthiness analysis. Elements include three- or four-node shell elements, beam and bar elements, and four- to eight-node solid elements. Various material properties are available to model elastic, nonlinear, and failure conditions. The equations of motion are integrated explicitly by the method of central differences. A contact-impact algorithm permits gaps and sliding contact along element interfaces.

Because PAM-CRASH is vectorized to be executed on supercomputers, it calculates more than 10,000 elements during 100,000 time steps in a reasonable time. Mazda originally installed PAM-CRASH on an IBM 3090 system to verify the effectiveness of simulation methods. When the CRAY X-MP system was installed in 1989, engineers began to apply simula-

tion methods during the development of new vehicles. Now researchers run PAM-CRASH on the CRAY X-MP system to optimize the body structure for frontal and rear impact. MAZDA researchers currently are validating a side-impact model.

Frontal impact

To test frontal impact, Mazda researchers mathematically represented an ordinary passenger vehicle with a front engine and front-wheel drive. The vehicle was formed three-dimensionally, assuming it would crash into a rigid barrier. Finite elements with a mesh size of approximately 10 by 15 mm were used to model the sheet metal in the front half of the engine compartment. This was needed to capture the impact buckling modes. Because the rear portion exhibits little deformation, a rather coarse mesh was used to save calculation time. Static strain-stress curves of steel, linearly interpolated between eight points, were used to characterize the steel material, taking the strain rate effect into account by the Cowper-Symonds equation. The bumper was excluded to reduce calculation time. Researchers substituted an equivalent dimension of the rigid body for the residual element bumper deformation. Longitudinal deformation of the vehicle was expressed so that the difference in length between the bumper and its residual was added to the calculated deformation of the whole body structure.

A full-scale frontal barrier crash test at 56 km/hour revealed the barrier force, which is defined as the force measured from load cells set on the barrier, versus vehicle deformation. A computer simulation was executed on the full-scale vehicle model, taking into account the test results and the outcome of several component simulations. Figure 1 shows the comparison of tested and calculated results with respect to barrier force. Here, the lateral axis is shifted 120 mm for the calculation because the bumper model was excluded, and its residue was added to the results of the simulation. At this point, computed results did not agree with the results of the physical test. However, as described below, the discrepancies could be eliminated largely by the use of a more detailed finite element model.

Three results differ between the simulation and the physical test:

☐ The initial peak force is too high because the bumper is not included in the model. The value of the second peak force also is higher in the simulation than in the physical test. The difference between the test and simulation for the deformation of the crossmember (Figure 2) was caused by the mesh used to represent the crossmember. The mesh was considered too coarse to accurately show buckling at the point between the vehicle's front end and the engine support. Consequently, changing the mesh size for the crossmember from 100 by 100 mm to 30 by 30 mm reduces the difference between the physical test and simulation with respect to the deformation of the crossmember and the barrier force.
☐ The vehicle deforms less in the simulation than in the physical test when the peak force is produced as the engine contacts the rigid barrier (Figure 1). The results of the physical test and simulation may differ because the engine supports modeled by using

nodal constraints are too stiff to allow the engine to move parallel to the ground. Good agreement between test and simulation with respect to the behavior of the engine and barrier force is achieved when engine supports are modeled with bar elements, which are characterized by quasi-static stress-strain curves.

☐ The simulated barrier force attained after the engine contacts a barrier is lower than that achieved by physical testing. However, by adding sliding interfaces behind the engine and between the suspension tower and fire wall, good agreement is achieved between the physical test and the simulation with respect to the barrier force.

Figure 3 compares the physical tests to computer simulations for all of the above-mentioned conditions: at a crossmember, with engine supports, and with sliding interfaces. A finite element model and its deformation are shown in Figures 4 and 5. This model includes approximately 20,000 shell elements. Although this model is twice as large as the initial crash model, this increase is required to achieve
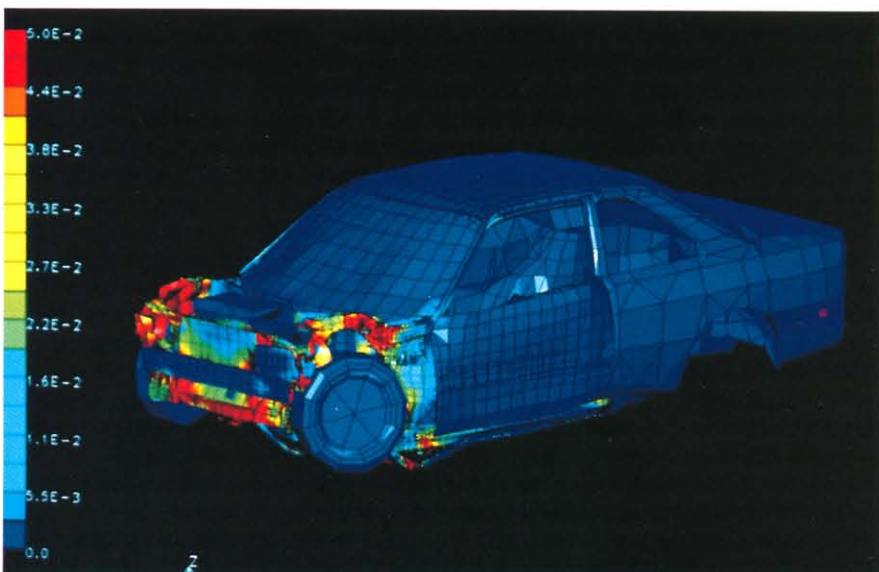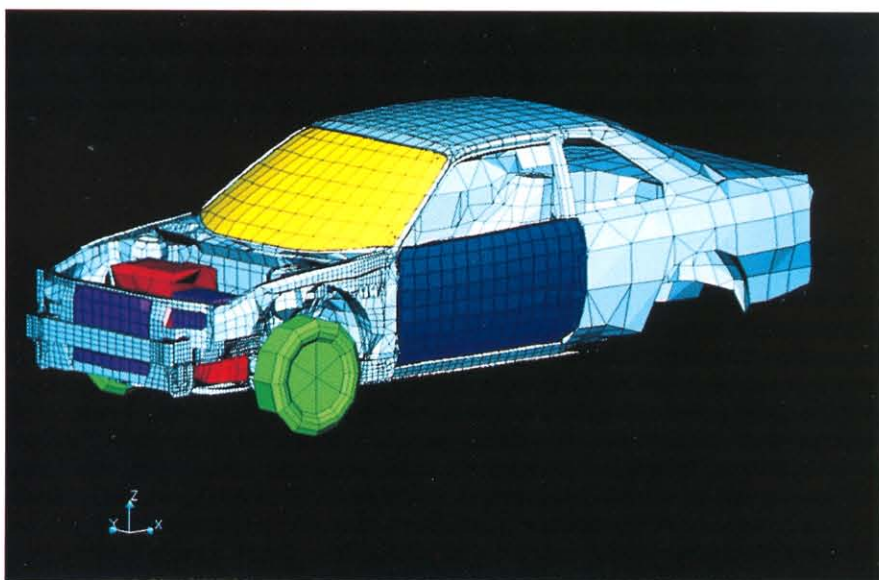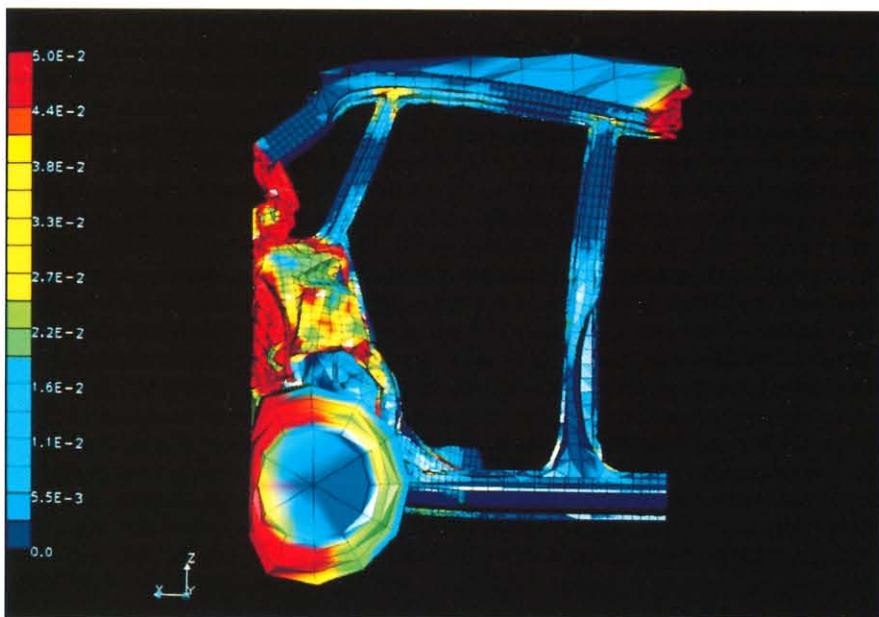
Figure 4 (top). A full-scale finite element model for frontal impact.

Figure 5 (bottom). The deformation and distribution of plastic strain at 60 msec.

Figure 6 (top). A rear crash finite element model.

Figure 7 (bottom). Deformation and distribution of plastic strain at 60 msec.

elements and requires six CPU hours to reach the maximum deformation point (60 msec) on the CRAY X-MP system. Figure 7 shows the deformation and distribution of plastic strain at 60 msec. Good agreement was achieved between the deformation results obtained by simulation and physical testing.

Using this model, we could estimate the improved crashworthiness that would be gained by substituting reinforced parts. These results were used to optimize the vehicle structure.

Side impact

At Mazda, researchers continue to develop models that adhere to impact standards proposed by the U.S. National Highway Traffic Safety Administration (NHTSA); a simulation based on the NHTSA honeycomb barrier almost is complete. To correlate the energy-absorbing characteristics of the honeycomb barrier, solid elements are used in the mathematical model. Defining the properties of the solid elements is difficult because the deformed shape of the solid elements easily becomes unrealistic, even if the barrier force is close to the test results. The barrier model is validated in terms of deformation and force by modifying each parameter, such as the solid element's volumetric strain and pressure, and by optimizing the mesh distribution. Figure 8 compares simulation and physical test results of the deformation-force curve of the crab barrier model into the rigid barrier. This validated barrier model simulates a full-scale side impact.

Full-scale side-impact simulations include some technical challenges not encountered in other types of crash simulations. For example, the crab barrier hits the door at high velocity, resulting in penetration if the contact area is small. The barrier also has a longitudinal velocity along the body, as it slides along the door to the rear part of the car body. The barrier sometimes penetrates the door, or other parts of the car body. Solving this problem requires a fine mesh; increasing the mesh resolution increases CPU time and modeling effort. Fortunately, the contact-impact algorithm in PAM-CRASH uses limitations to reduce calculation time.[?]

the necessary accuracy. The model is analyzed up to the point of maximum deformation (80 msec) in 18 CPU hours on the CRAY X-MP system. Based on our findings for frontal impact, we now routinely apply these methods to vehicle development.

Rear impact

Figure 6 shows the finite element model Mazda researchers use for rear-impact studies of a typical small vehicle. To reduce calculation time, only the rear half and the right side of the vehicle are modeled. The front half of the vehicle is considered rigid, based on experimental results. Instead of modeling the left side, symmetric boundary conditions are applied at the plane of symmetry between left and right. This does not reduce the accuracy of the results, because the deformation mode of the body structure is symmetric. A moving rigid-wall element is used to simulate the impact vehicle. The vehicle model includes 11,000

Figure 8. A comparison of deformation characteristics.
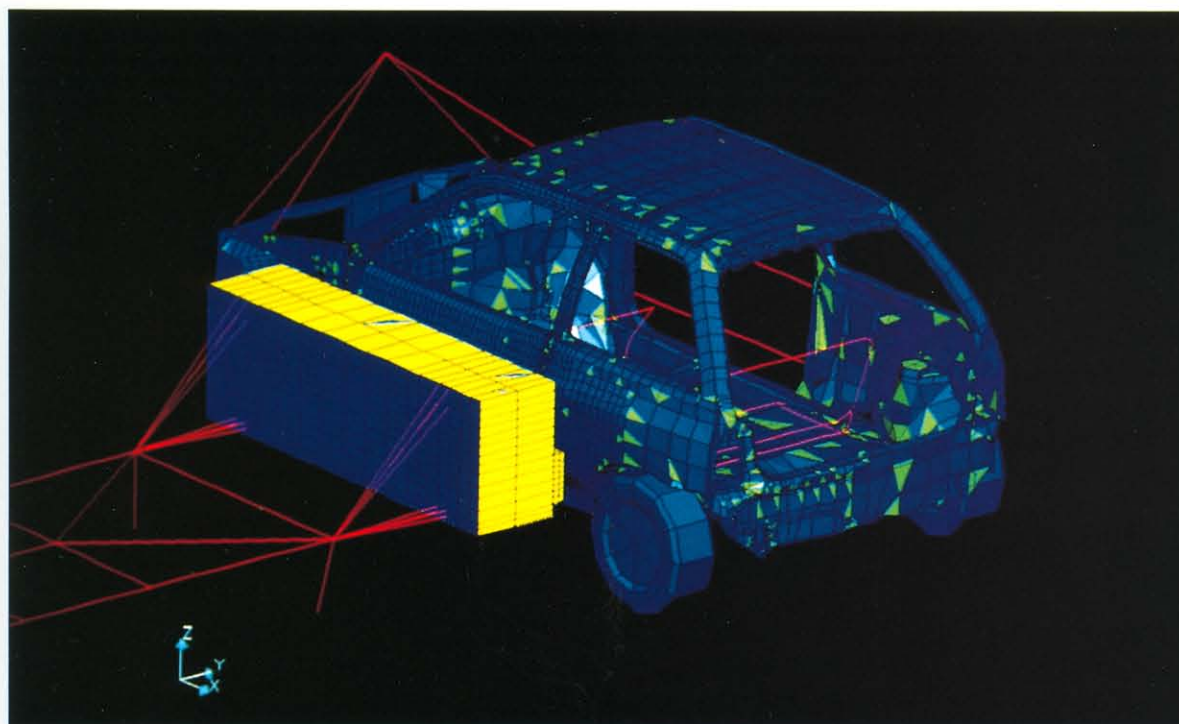
Another challenge is modeling the effects of vehicle occupancy on door deformation. The rate at which the door deforms is a significant factor in occupant injuries, and differs greatly depending on the location of the occupants.

Figure 9 shows the finite element model for the small vehicle side-impact simulation. For a small vehicle, modeling the front and rear suspension is necessary because the crab barrier first hits the front tire then slides along the doors to the rear tire. This model includes 24,400 shell elements for the body structure and 1600 solid elements for the crab barrier. This is one of the largest crash simulation models. This 40-msec. simulation takes 13 CPU hours on a CRAY X-MP system (Figure 10). In this example, the physical test and simulation are in good agreement.

Future applications

At Mazda, researchers actively apply supercomputer simulations to the development of new cars. Many useful results are obtained, and we have many research goals for the future. The correlated results for frontal impact are considered acceptable for practical applications, although we are working to improve models of components, such as bumpers, batteries, and tires. These components usually are incorporated into vehicle models.

The rear-impact results for the half model are useful when the vehicle is laterally symmetrical. For the full model, and when a vehicle is not symmetrical, the mesh size should be optimized so that it can be calculated in a reasonable time.

The honeycomb barrier model is available for full-scale side-impact simulation. The simulation and physical test results for body deformation are in good agreement. Next we will work on incorporating an occupant dummy into the vehicles to evaluate the effects of occupants on crash performance. ■
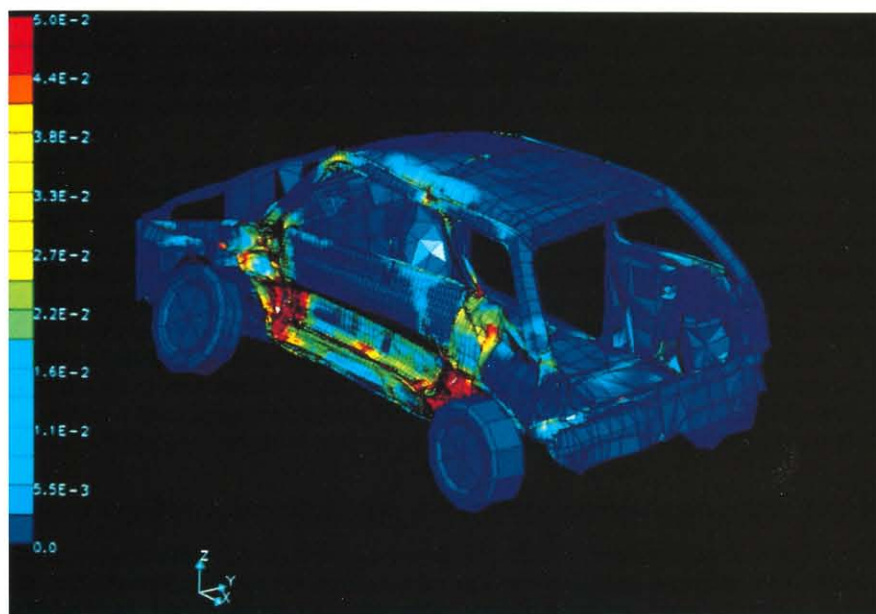
About the authors

Seiichi Ando, Koji Kurimoto, and Koji Taga are engineers in the Safety Development Group at Mazda Motor Corporation in Hiroshima, Japan. They develop vehicle crash simulations using Cray Research supercomputers.

References

1. Du Bois, P., and J. F. Chedmail, "Automotive Crashworthiness Performance on a Supercomputer," SAE International Congress and Exposition, Detroit, Michigan, 1987.
2. Haug, E., et al., "Compact-Impact Problem for Crash," Post-symposium short course of the Second International Symposium of Plasticity, Nagoya, Japan, 1989.
3. Kurimoto, K., et al., "Simulation of Vehicle Crashworthiness and its Application," The 12th International Conference on Experimental Vehicle Safety, Gothenburg, Sweden, 1989.

13

Autotasking forecast and climate models on the CRAY Y-MP system

Alan Dickinson, U.K. Meteorological Office
Bracknell, England

Operational weather forecasting has long been recognized as an application well-suited to parallel processing. In 1922, well before the advent of the first electronic computer, L. F. Richardson proposed a method for parallel numerical weather prediction.[1] Richardson described an operational weather forecasting system based on a factory of 64,000 human "computers," in which each person calculated the state of the atmosphere at his or her own grid point using data provided by neighbors. A conductor was responsible for synchronizing the calculations at each time step. Even 70 years later, this vision provides a perfect analogue to illustrate how the problem might be implemented on today's massively parallel computer systems.

A job for the CRAY Y-MP system

Because of the time-critical nature of numerical weather prediction and the large amount of data processing involved, weather models always have demanded the most powerful computers available. Today, these models require multiprocessor supercomputers. In fact,

The CRAY Y-MP system.

the accuracy of computer models has advanced in step with the amount of supercomputer power available.

Climate modeling also consumes a great amount of computer time. Growing international concern with atmospheric concentrations of greenhouse gases has led to renewed efforts to predict the effects of climate change. This research requires coupled models of the complete atmosphere-ocean system to be integrated forward a century or more. Although these models use much lower spatial resolutions than forecast models, a climate-model experiment can require many thousands of hours on the most powerful computer systems.

The U.K. Meteorological Office, which provides a national meteorological service, uses a CRAY Y-MP/832 computer system to perform operational weather forecasting using a variety of models of the atmosphere. These range from a global model with a horizontal resolution of 100 km, to a regional mesoscale model covering only the British Isles with a grid length of 15 km. The Meteorological Office also is at the forefront of an international effort in climate research, through its establishment of the Hadley Centre for Climate Prediction and Research. The U.K. Meteorological Office recently acquired a CRAY Y-MP/832 computer system and expects to acquire a second computer system with similar features and capabilities in early 1991. This second system will be used primarily for climate change predictions. These computers replace a Control Data Corporation Cyber 205 computer installed in 1981.

A new, unified climate-forecast model is being developed to run on the CRAY Y-MP system. It

is written in modular form using standard Fortran and includes a more accurate description of many important atmospheric processes than have previous models. A significant aspect is the use of Cray Research's CF77 compiling system with Autotasking, which enables users to parallelize codes automatically.

Autotasking the unified climate forecast model

Some of the model components are not yet in place, but examples from key areas of the atmospheric part of the code give an indication of the levels of performance that eventually might be achieved.

Multitasking considerations

The CRAY Y-MP system at the U.K. Meteorological Office has eight CPUs, a six nanosecond clock period, 32 Mwords of main memory, and 128 Mwords of SSD memory. Under the UNICOS operating system, the eight processors can operate independently on separate programs or concurrently on a single problem in multitasking mode.

The dedicated resources of all eight CPUs on the CRAY Y-MP system will be used for production runs of the forecast models. Climate modeling also will require some of the speed gains offered by multitasking, although in practice a small number of separate climate integrations may be run in parallel.

On a CRAY Y-MP system, code can be multitasked using three methods: macrotasking, microtasking, and Autotasking. Depending on the code structure and programmer preference, these techniques may be used separately or in combination.

Macrotasking allows a program to be partitioned into two or more tasks at the subroutine level. The programmer inserts library routine calls into the source code to initiate and synchronize tasks scheduled by the operating system. This technique is best suited to programs that exhibit large granularity. Macrotasking was the first implementation of multitasking on Cray Research computers and has been used with great success by several meteorological organizations, most notably the European Centre for Medium-Range Weather Forecasts.[2]

Microtasking, however, allows parallelism to be exploited at the DO-loop level and therefore is suited to problems that exhibit fine granularity. The programmer identifies parallel regions in the program and inserts directives accordingly. Code then is generated that allows the program to hand out loop iterations to each CPU as they become available for work, using a master-slave relationship. If no extra CPUs are available, the code will continue to execute on a single CPU. The code that controls the scheduling of additional CPUs is extremely efficient, using about 40 clock periods for each new iteration.

The CF77 compiling system with Autotasking identifies and exploits parallel regions in a program through the use of a preprocessor. This process may be completely automatic, but programmer intervention may be advantageous because not all types of parallelism can be detected. Autotasking is functionally equivalent to microtasking with the added advantages of automatic detection of parallelism, automatic scoping of variables, and the capability to exploit parallelism in individual DO loops.[3]

	Climate	Forecast
Number of levels	20	20
Number of points E-W	96	288
Number of points N-S	73	217
Time step (seconds)	1200	600
Grid box (degrees)	2.5 × 3.75	0.833 × 1.25

Table 1. Global model dimensions used on the CRAY Y-MP system for operational forecasting and climate studies.

Using Autotasking

Cray Research's CF77 compiling system comprises four elements: the Autotasking Fortran dependence analyzer (FPP), the Fortran translator (FMP), the UNICOS cf77 command, and the Cray Fortran 77 compiler (CFT77), which generates the code. The dependence analyzer looks for parallelism in the program and inserts directives, which are interpreted by the translator. The translator converts the directives into extra code that controls the parallel execution. During either analysis or translation, the programmer may insert directives to override how a particular block is handled by the compiling system.

Autotasking is very easy to use, and it is our experience that the vast majority of subroutines are multitasked without any user intervention. The design and structure of the model is intended to make use of the Autotasking feature to strip-mine inner loops. However, when implementing the option to strip-mine inner loops, FPP 3.1 did not recognize inner loops that would vectorize as well as CFT77. As a result, a vector loop occasionally was parallelized by allocating each loop iteration to a separate processor. This made the code less cost-effective than code run in single-tasked mode. This could be corrected by inserting a new directive (usually by replacing DO ALL with DO ALL VECTOR).

One consequence of Autotasking is that the final object code can be up to three times larger than its single-tasked equivalent. This is because Autotasking generates three copies of each parallel block: one for the master task, one for the slave tasks, and a unitasked version of the master task, which is executed if threshold tests indicate that single-tasking would be more efficient. The threshold test and the uniprocessor copy of the code can be removed through a compiler option.

The unified climate forecast model

Formulation

Forecast models and the atmospheric part of climate models may be viewed as different resolution versions of the same program. In forecast mode, a user selects the highest resolution that can be run in the time available. In climate studies a tradeoff must be negotiated between the accuracy in the representation of features and the length of time for which the model can be run. Therefore, coarser resolution models typically are used. Table 1 compares the climate and forecast resolutions used today, although enhanced resolution climate models undoubtedly will be needed soon to investigate the regional impacts of climate change.

The dynamic formulation of the model uses predictive equations on a latitude-longitude grid in the form of White and Bromley,[4] which includes specific terms important for treating large-scale flows. The finite difference scheme seeks to combine the accuracy and efficiency of the split-explicit scheme used in the forecast model run on the Cyber 205 computer with the conservation properties required for long-term climate integrations.[5] At each time step, the gravity-inertia terms are integrated in three short adjustment steps followed by a single "advection" step that includes the horizontal advection and diffusion terms. The effects of the physical parameterizations are added during the advection stage. These include convection,[6] large-scale precipitation, a comprehensive treatment of clouds and radiation, boundary-layer effects, and gravity wave drag. At high latitudes, stability filtering is applied to prevent the small east-west grid length from imposing unacceptable restrictions on the length of a time step.

Code design

On the U.K. Meteorological Office's previous computer systems, production code generally was written in a low-level language to extract the maximum performance from the hardware. On the Cyber 205 system, for example, Q8 calls were used, which allow direct access to hardware instructions. Although this approach led to fast, efficient code, maintenance and modification were difficult. Therefore, Fortran was chosen when writing the unified model and the CF77 compiling system was used to optimize the code. The small performance gains that might be obtained by judicious use of low-level language are now outweighed by the need to make code modifications as easy as possible. Apart from the control level, all sections that perform meteorological calculations are designed to be "plug compatible" and communicate information

with the rest of the model purely through their argument list. All routines correspond to the standard for exchanging meteorological code proposed by Kalnay et al.[7]

Because the model uses an explicit time-stepping scheme, only one copy of the model prognostic fields needs to be stored. This allows all the data to be retained in memory, even at the highest proposed resolution. The program stores data as horizontal fields, so that inner loops typically are over all points along a model level. This arrangement maximizes vector efficiency while allowing a range of multitasking options. The multitasking strategy currently adopted uses Autotasking to partition the innermost loops across all the available processors. With this approach, load balancing is assured by the dynamic scheduling algorithm used in Autotasking, provided the vectors are sufficiently long. It also removes the need to consider any horizontal (or vertical) dependencies inherent in the integration scheme by forcing multitasking to be performed from the inside out. Consequently, it is much easier to make fundamental changes to the model's formulation without redesigning the overall multitasking strategy.

Performance

To analyze the performance of the model it is convenient to consider the dynamics and physics routines separately. Note that in the present generation of models, the physics contributes approximately 55 percent to the cost of an integration and the dynamics contributes 45 percent. The trend is toward further increases in the relative cost of the physics.

In the dynamics, the same operations generally are performed at every point in the integration domain. This makes these algorithms easy to vectorize, but horizontal and vertical dependencies resulting from the discretization process limit the

Figure 1. The GFLOPS rate and speedup over single-tasked code of code distributed over eight processors using Autotasking. Results are shown for the global model dynamics (left) and the convection scheme (right) using a range of horizontal dimensions.
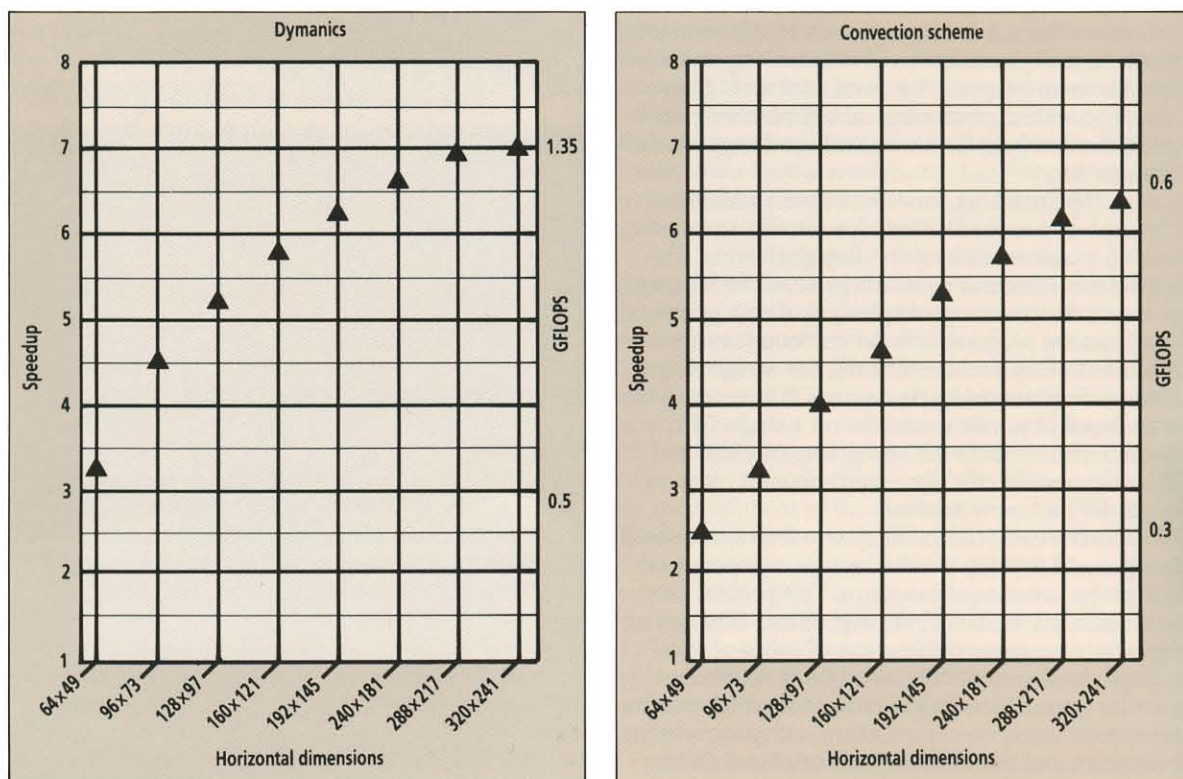
range of effective macrotasking options.[8] By adopting a multitasking approach that strip-mines inner loops, these dependencies can be ignored. But the success of this strategy depends on having large DO loops containing long vectors, because each loop incurs an overhead related to the cost of the code to control the multiprocessing. The performance of the dynamics routines, including the FFT code used for stability filtering, is shown in Figure 1. At the forecast resolution, a speedup factor of 6.9 over the single-CPU run without Autotasking is recorded with a respectable performance of 1.3 GFLOPS, while the climate resolution shows a speedup factor of only 4.5 and a performance level of 0.8 GFLOPS.

An estimate of the multitasking overhead can be obtained by computing the difference between the cost of a run using Autotasking limited to one CPU and the cost of the same run when Autotasking is not used. If this is inserted into Amdahl's Law, it gives an indication of the maximum speedup that can be obtained (Table 2).

Predictions of mt__ overhead for the climate and forecast resolutions of the dynamics were performed. By substituting these numbers into Amdahl's Law with number__of__cpus = 8, maximum speedups of 4.5 and 6.9 respectively are obtained — the same as given in Figure 1. Because the multitasking overhead is proportional to the number of parallel loops in the code, any further improvement in performance only can be obtained, logic permitting, by combining loops that use the same vector length. Any lack of load balancing appears to be insignificant, underlining one of the main benefits of the dynamic scheduling algorithm used to automatically multitask inner loops.

In contrast to the dynamics, the physics is data-dependent and localized, characterized by nested conditional blocks of code. Because no horizontal dependencies exist, a straightforward macrotasking approach, in which the integration domain is split into eight regions, each integrated in parallel by a separate processor, is a valid alternative strategy for using Autotasking on inner loops. If the conditional constructs are converted to use the Cray Research vector merge functions, such as CVMGT, then GFLOPS rates that surpass those obtained for the dynamics can be produced. This is a false measure of performance, however, because most of the physical processes apply only at subareas of the grid and a mask-merge approach leads to calculations being performed at all grid points without respect to where the final results need to be applied.

In the gravity wave drag parameterization, for example, vertically propagating mountain waves are modeled, so the effects need only be computed over land points. Because only one-third of Earth's surface is covered by land, the cost of this routine can be reduced significantly by first focusing on this set of points. This is achieved by using explicit gathers in the code based on index lists produced from simple tests. The other physics routines use this approach in varying degrees. For example, the cost of the convection scheme can be reduced by more than 50 percent by first removing stable points from the calculations (Figure 2). The performance of the convection scheme is shown in Figure 1. Note that the GFLOPS rate is less than half that of the dynamics, but the wall clock

$$mt\_overhead = 1 \text{ CPU automatically multitasked time} - single\_tasked\_time$$

$$max\_speed\_up = \frac{single\_tasked\_time}{mt\_overhead + single\_tasked\_time/number\_of\_cpus}$$

time of the code is significantly less than can be obtained with the mask-merge approach. Tests show that further improvements in wall-clock time, particularly at climate resolution, can be obtained by using the regional macrotasking approach described above, but with the addition of using Autotasking on inner loops to provide load balancing.

Even if a macrotasking layer were added to the physics, the performance of the climate resolution version of the model still would be rather poor, and it would be unacceptable to run this in a stand-alone mode. However, at least four parallel streams of work are identifiable in climate prediction: control runs, variously increased CO_2 scenarios, formulation changes, and runs from different initial data. Also, the size of the problem is such that four copies may fit easily into the available memory without the need to roll out a job.

Table 2. An estimate of multitasking overhead is inserted into Amdahl's Law to indicate maximum speedup.

Figure 2. Typical distribution of unstable points over the globe obtained from 11-level climate resolution data. These can be identified at each time step by a simple test on the temperature and moisture fields.
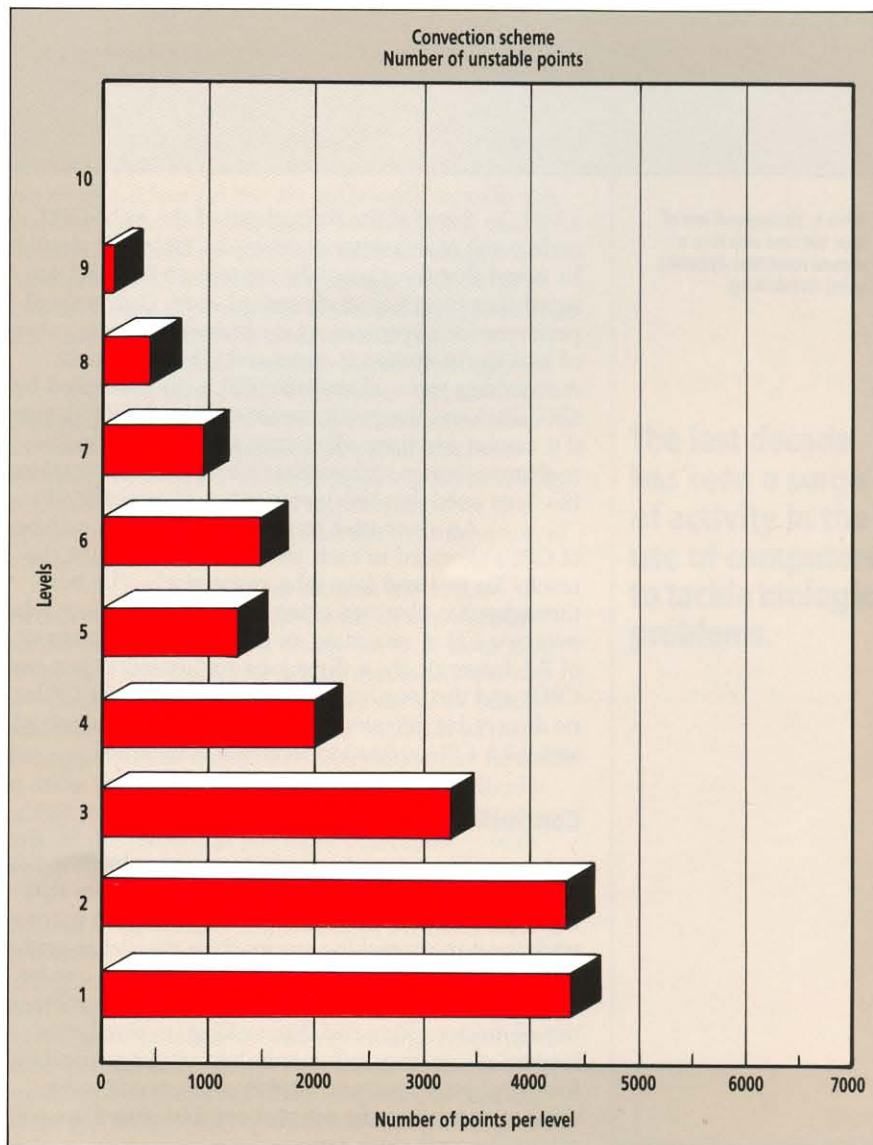

Convection scheme
Number of unstable points

(Levels vs. Number of points per level)

CRAY CHANNELS   FALL 1990

17

Table 3 (right). Throughput test of 100 time step run of climate resolution dynamics distributed over eight processors using Autotasking.

Table 4 (below). Throughput test of two 100 time step runs of climate resolution dynamics using Autotasking.

Number of copies of job in system	Average time per job in seconds
1	28.3
2	27.0
3	26.1
4	26.4

Number of CPUs used for Autotasking	Wall-clock time in seconds	Number of CPUs utilized	Number of CPUs allocated to work
1 (single-tasked)	114	2	2
2	64	3.5	4
4	36	6.3	8
8	54	4.3	8

Number of CPUs used for Autotasking	Wall-clock time in seconds	Number of CPUs utilized	Number of CPUs allocated to work
1 (single-tasked)	114	4	4
2	64	7.1	8
8	105	4.4	8

Table 5. Throughput test of four 100 time step runs of climate resolution dynamics using Autotasking.

A test of the throughput of the eight-CPU code using Autotasking is shown in Table 3. It should be noted that these test jobs contain no I/O and no significant pieces of single-tasked code. Only a small performance improvement is obtained as the number of jobs in the system is increased. This is because Autotasking grabs all available CPUs (as controlled by GETCPUS and the environment variable NCPUS), even if it cannot use them efficiently, and appears unable to share resources with another job in which Autotasking has been used that also wants to use all available CPUs.

An alternative strategy is to limit the number of CPUs allocated to each job. Tables 4 and 5 list the results for two and four jobs, respectively. The best throughput is obtained using Autotasking on four jobs over two CPUs, recording an overall CPU utilization of 7.1. Interestingly, if three jobs are limited to just two CPUs and the fourth is allowed to use all eight CPUs, no discernible difference in throughput can be observed, and a 7.1 CPU utilization level still is obtained.

Conclusions

The use of Autotasking on inner loops avoids the horizontal and vertical dependencies that usually need to be considered when using the more traditional macrotasking approach to parallelize grid-point models. Acceptable levels of parallelism can be achieved with large problem sizes, although the combined use of macrotasking and Autotasking (to give better load balancing) may well give the best results, particularly for the physics routines. Further work needs to be done to determine the optimal combination that minimizes wall-clock time.

Batch throughput tests at climate resolution suggest that running more than one eight-CPU automatically multitasked job in the system at the same time does not lead to a significant improvement in overall performance. Essentially, batch processing will make an inefficient eight-CPU automatically multitasked job relatively more efficient only if other jobs in the system are restricted to use fewer than eight CPUs. Acceptable levels of performance appear to be possible if climate runs are limited to two CPUs each.

Autotasking is easy to use. In the ideal case, only a small set of compiler options needs to be invoked. However, sometimes the preprocessor is unable to recognize that the inner loop vectorizes. In such cases, the user is required to hand-code some loops. Finally, the use of Autotasking on inner loops has allowed fine levels of granularity to be exploited in a user-friendly, almost transparent way. However, it should be stressed that this approach is effective as a primary multitasking strategy only if the program first is structured to use long vectors. ◾

About the author

Alan Dickinson is responsible for the application of climate models to monthly forecasting at the U.K. Meteorological Office. Since he joined the organization in 1976, he has worked on the development of numerical weather prediction and climate models. He received a B.Sc. degree in applied mathematics and a Ph.D. degree in fluid dynamics from Liverpool University. Dickinson is a Fellow of the Royal Meteorological Society.

References

1. Richardson, L. F., *Weather Prediction by Numerical Process*, Cambridge University Press, Cambridge, 1922.
2. Dent, D., "The ECMWF Model: Past, Present and Future," *Multiprocessing in Meteorological Models*, Editors G. R. Hoffmann and D. F. Snelling, Springer-Verlag, 1988.
3. Furtney, M., "Parallel Processing at Cray Research, Inc.," *The Dawn of Massively Parallel Processing in Meteorology*, Editors G. R. Hoffmann and D. K. Maretis, Springer-Verlag, 1990.
4. White, A. A. and R. A. Bromely, "A New Set of Dynamical Equations for Use in Numerical Weather Prediction and Global Climate Models," Met O 13 Branch Memo, Meteorological Office Internal Paper, 1988.
5. Bell, R. S. and A. Dickinson, "The Meteorological Office Numerical Weather Prediction System," Meteorological Office Scientific Paper No. 41, 1987.
6. Gregory, D., and P. R. Rowntree, "A Mass Flux Convection Scheme with Representation of Cloud Ensemble Characteristics and Stability Dependent Closure," *Mon. Weather Review*, June 1990.
7. Kalnay, E., M. Kanamitsu, J. Pfaendtner, J. Sela, M. Suarez, J. Stackpole, J. Tuccillio, L. Umscheid, and D. Williamson, "Rules for Interchange of Physical Parameterizations," *Bulletin of the American Meteorological Society*, Vol. 70, No. 6, 1989.
8. Dickinson, A., "Multitasking the Meteorological Office Forecast Model on an ETA-10," *The Dawn of Massively Parallel Processing in Meteorology*, Editors G. R. Hoffmann and D. K. Maretis, Springer-Verlag, 1990.

Gaussian 90

Applying quantum chemistry to biology

Douglas J. Fox, Pittsburgh Supercomputing Center
Pittsburgh, Pennsylvania

The large users of computational resources traditionally have come from physics and chemistry, where mathematical models play a central role. Problems in these areas lend themselves naturally to computational methods, and as a result the power of computer models can keep pace with increasingly powerful computational resources, such as supercomputers. Biology has lagged behind physics and chemistry as a consumer of computer cycles, however. In some cases this has been due to the lack of a clear computational model that addressed the problems of interest. In others, it has been because the size of realistic biological models outstripped both the capability of the method and the available computational resources.

The last decade has seen a surge of activity in the use of computers to tackle biological problems. Recent applications have ranged from the study of genetic functioning, via searches of protein databases, to the use of finite element and fluid-flow methods to model the human heart. At the molecular level, molecular mechanics has proven to be a very powerful tool for addressing questions of the structure, dynamics, and thermodynamics of macromolecules. Molecular dynamics builds on and complements the traditional tools of computational chemistry so that improvements in these tools will aid in the continued growth of molecular mechanics. This article covers some of the problems of using ab initio quantum chemical techniques to study biological systems and the ways in which the applicability of these methods can be expanded. Included are examples of studies of malonate complexes with calcium and magnesium that were performed on the CRAY Y-MP8/832 computer system at the Pittsburgh Supercomputing Center.

Characteristics of molecular mechanics

Molecular mechanics has been one of the more widely used methods in the study of molecular structure. It has provided a route to understanding macromolecular structure, dynamics, and thermodynamics. Molecular mechanics has been successful because classical potential functions that govern the motion of the nuclei can be defined. These potential functions describe chemical bonds that are inherently quantum mechanical but are sufficiently transferable so that the classical analog does not need to be modified for each chemical environment. The developers of these potential functions have used a wide range of theoretical and experimental sources to find parameters that reproduce nature at a reasonable cost.

As molecular mechanics has been applied to an ever widening field of problems, the need for improved force fields has become evident. Improvements are needed where sufficiently powerful quantum chemistry methods could be applied to great advantage. Examples of areas that need improvement are the treatment of atomic charges and the determination of energy barriers.

The charge on an atom in molecular mechanics is a parameter, whereas in a molecular orbital calculation the net charge on each atom is a property. Thus molecular orbital methods are a way of defining atomic charges and are useful as benchmarks for evaluating approximate methods of introducing polarization into molecular mechanics. However, to make the benchmarks meaningful, the molecular orbital method must be able to handle systems as large as the molecular mechanics calculations with which they are compared.

Acceptable energy barrier heights are extremely difficult to determine. Experimental methods exist to determine rotational barrier heights but often create ambiguities as to what barrier actually has been measured. Similarly, energy differences for intramolecular rearrangements often are subject to large amounts of experimental uncertainty. Theoretical methods offer unambiguous measures of these quantities that can fill the gap for missing experimental data and help clarify ambiguous cases.

> The last decade has seen a surge of activity in the use of computers to tackle biological problems.
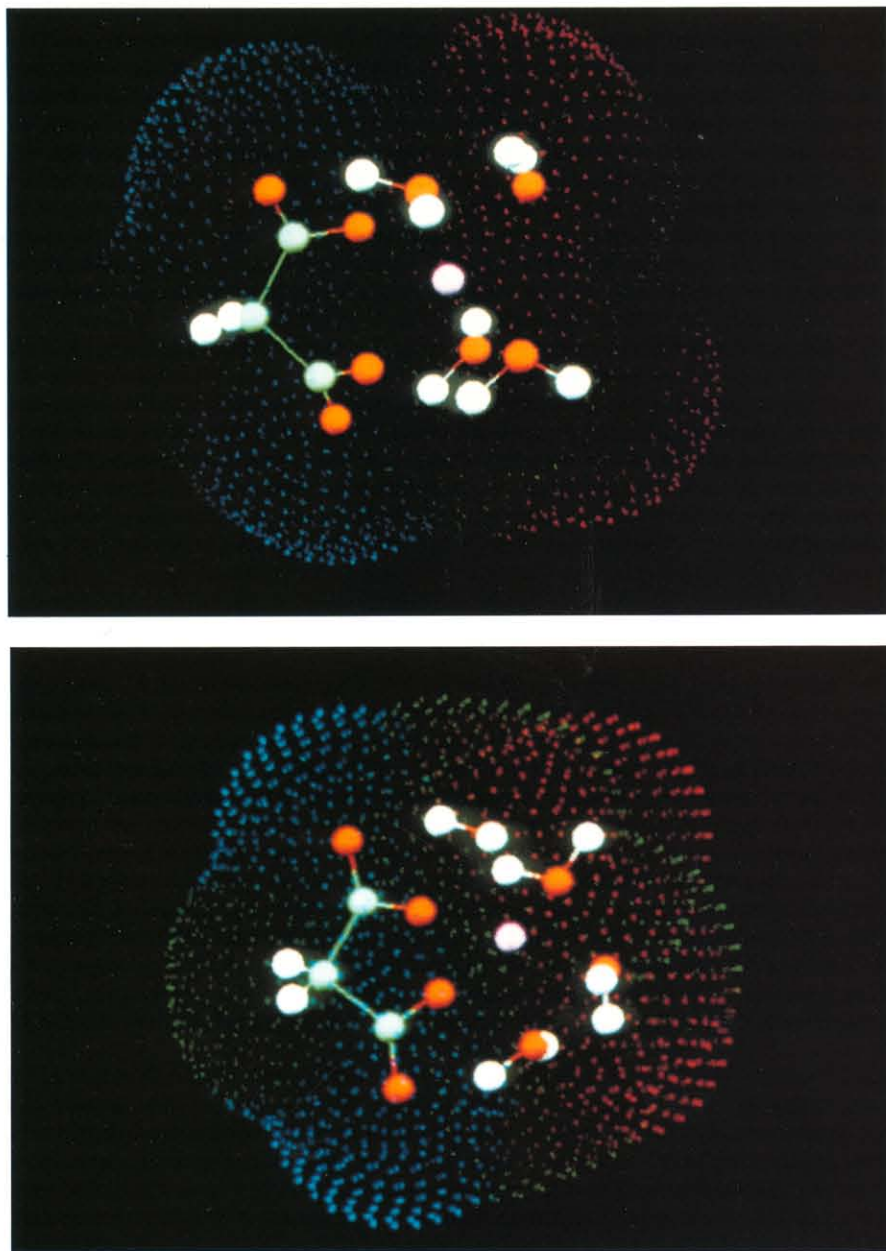
Figure 1. Calcium (top) and magnesium (bottom) complexes, each with four waters of hydration. The electrostatic potential is mapped onto a solvent-accessible surface about each complex.

Characteristics of ab initio molecular orbital methods

Ab initio molecular orbital methods apply quantum mechanics to molecular structure and energetics by solving the Schrödinger equation for an electrostatic Hamiltonian. The problem is expanded in a set of atomic basis functions that ideally span the spatial and angular momentum dimensions of the problem space. The Roothan-Hartree-Fock method uses linear combinations of basis functions to approximate independent one-electron molecular orbitals, and the linear combinations are iterated to obtain a self-consistent field (SCF) description. The quality of the wavefunction produced can be improved by expanding the function space and by correcting for errors in the independent electron model, a process called electron correlation. Ab initio methods originally were directed at determining molecular structure and energetics for small molecules that could be studied in detail experimentally.

Studies like these have shown that the methods reproduce molecular structures and frequencies to within a few percent, making them useful in the study of systems that could not be studied accurately by experiment.

Because these methods are computationally demanding, the codes have been ported regularly to high-performance hardware including Cray Research supercomputers. The most computationally costly aspect is the treatment of electron repulsion integrals. Because there are $o(n^4)$ electron repulsion integrals, where n is the size of the atomic orbital expansion, the cost of computing them is so high that the traditional method is to compute the list of integrals once, store it on disk, and read through the list during each iteration of the self-consistency procedure. Because computational power has improved faster than disk performance, this method often makes disk space the limiting resource, and the I/O costs can exceed the CPU costs. In addition, attempts to vectorize the evaluation of these integrals were largely unsuccessful for several years.

Features of Gaussian 90

The Gaussian system of programs has developed over the past two decades into a very user-friendly package for highly accurate searches of molecular potential energy surfaces. The new work included in Gaussian 90 reflects two general directions in which the programs are moving: extension to very large molecular systems and improved electron correlation models.[1] The treatment of very large molecules is addressed by the continued extension of direct SCF methods first proposed by Amlof and coworkers.[2] The treatment of electron correlation for biological systems is addressed by the development of direct and semi-direct second order Moller-Plesset perturbation theory.

The direct SCF method

Direct SCF is based on a simple concept. Instead of storing the full n^4 set of integrals, the Hamiltonian matrix is formed by computing a group of electron repulsion integrals. Once the full Hamiltonian is formed, it is diagonalized and the wavefunction is updated. If the process has not reached self-consistency, it is repeated, and the electron repulsion integrals are recomputed. This was proposed initially as a route to studying very large molecules when the integrals could not be stored, but it has proven to be a viable route even to medium sized molecules.

The method is economical for two principal reasons. First, Amlof showed that for large molecules the list of integrals does not continue to grow as n^4, because the electron repulsion integrals tend toward zero for widely separated parts of the molecule. Second, with careful consideration of the integral prefactors it is possible to avoid evaluation of large blocks of integrals that would not significantly change the wavefunction on the current iteration. These two factors combine to make the actual computational cost scale more like $n^{\sim 1.5}$, causing a crossover in costs for large enough molecules. On scalar machines this crossover is seldom reached, but on vector processors, such as Cray Research systems, the crossover is at about $n = 100$, which translates to a molecule such as glycine described with a good basis, 6-31G**.

Vectorized electron repulsion integral methods

Vector processors benefit from a lower crossover because, when vectorizing integral computations, the improved performance is magnified by the number of iterations. The method currently in use in Gaussian 90 was developed by Head-Gordon and Pople,[3] who extended the recursion relations of Obara and Saika.[4] The recursion relations allow the identification of large groups of related integrals. The inner loops for these groups often exceed 10,000 even in less than two megawords of memory. Direct computation of the electron repulsion integrals has been integrated across all SCF methods in Gaussian 90. First and second derivative integrals can be computed, as can perturbations to the molecular orbitals based on the Atomic Orbital Coupled Perturbed Hartree-Fock method, which are necessary for the second derivative of the SCF energy.

Direct Moller-Plesset perturbation theory

The concept of computing integrals as required has been extended also to second order Moller-Plesset perturbation theory.[5] This use does not reflect the economy of computing only part of the integral list, but because it is not an iterative procedure the loss is not serious. The code also has been implemented so that, when memory is available, the integrals can be computed and processed in batches of $o(n^2)$, $o(n^3)$ or $o(n^4)$. This makes the computation of electron correlation corrections for barrier heights or multiply bonded structures computationally accessible for a wide range of molecules.

Excited states

Economical methods are needed to search excited-state electronic potential surfaces because many biological processes occur through conformation changes on such surfaces. A new feature of Gaussian 90 is the ability to search excited-state surfaces using the single excitation configuration interaction method, CIS.[6] This method produces a compact wavefunction that is orthogonal to the ground state, yet allows variational relaxation of the excited state wavefunction. Moreover, the energy is analytically differentiable, making potential surface searches much more economical. It also has been implemented as a direct algorithm and so should be applicable to a wide range of molecular systems.

Parallelization

The absence of disk I/O during the direct SCF procedure naturally brings up the possibility of applying multiple processors. The preliminary work has been done, and an SCF calculation with 192 basis functions showed that about 94 percent of the computation is performed in the parallel region. The small memory required to achieve good vector performance allows each parallel process to be very efficient as well.

Application to Ca and Mg complexes with malonate

Calcium and magnesium atoms are divalent metal ions that bind in chelation bidentate orientations in a significant percentage of their known structures. However, in coagulation processes, where a number of substituted malonic acid groups exist, the binding of calcium and magnesium produces very different physiological effects. Accurate modeling of these bound metal ions in solution requires consideration of the hydrated complex. The use of direct SCF methods on the Cray Research system allowed us to include four water molecules to make the metal ions hexa-coordinate (Figure 1). Using the 3-21G* basis, the smallest basis likely to be chemically correct, these systems contained about 135 basis functions, and we observed about a tenfold improvement in CPU time for a structure optimization with direct SCF over disk-based methods on the Cray Research system.[7] Using this computer system reduced the time for this project from a few months on scalar machines to a few days.

Conclusion

Ab initio molecular orbital methods have proven to be a very useful complement to experimental methods and a useful route to properties that cannot be measured easily by experiment. Ab initio methods previously had been applied sparingly to biological systems because the methods did not scale to the sizes of molecules relevant to biological systems, such as several amino acid residues of a protein. The Gaussian 90 system of programs includes a number of new algorithms that offer the possibility of treating serious biological fragments realistically while making excellent use of vector processors such as Cray Research systems. ▬

About the author

Douglas J. Fox has been active in the ab initio quantum chemistry field for more than a decade, starting with doctoral work at the University of California, Berkeley, in the development of derivative techniques for computing configuration interaction wavefunctions. He became involved with the Gaussian system of programs while doing postdoctoral work at Carnegie-Mellon University.

References

1. Frisch, M. J., M. Head-Gordon, G. W. Trucks, J. B. Foresman, H. B. Schlegel, K. Raghavachari, M. Robb, J. S. Binkley, C. Gonzalez, D. J. Defrees, D. J. Fox, R. A. Whiteside, R. Seeger, C. F. Melius, J. Baker, R. L. Martin, L. R. Kahn, J. J. P. Stewart, S. Topiol, and J. A. Pople, *Gaussian 90*, Gaussian, Inc., Pittsburgh, Pennsylvania, 1990.
2. Amlof, J., J. Faegri, and K. Korsell, *Journal of Computational Chemistry*, Vol. 7, p. 385, 1982.
3. Head-Gordon, M. and J. A. Pople, *Journal of Chemical Physics*, Vol. 89, p. 5777, 1988.
4. Obara, S. and A. Saika, *Journal of Chemical Physics*, Vol. 84, p. 3963, 1986.
5. Head-Gordon, M., J. A. Pople, and M. J. Frisch, *Chemical Physics Letters*, accepted for publication.
6. Foresman, J. B. and J. A. Pople, submitted to *Journal of Chemical Physics*, 1989.
7. Deerfield, David W. III, Douglas J. Fox, Martin Head-Gordon, Richard G. Hiskey, and Lee Pederson, *Journal of the American Chemical Society*, submitted, 1990.

Announcing the CRAY XMS minisupercomputer system

Cray Research now offers a Cray-compatible minisupercomputer, the CRAY XMS system. With the addition of this system to its product line, Cray Research provides an unparalleled range of supercomputing capability, from a leading price/performance minisupercomputer to the world's most powerful computer system. The CRAY XMS system is a 64-bit minisupercomputer based on the proven Cray Research vector architecture, and, like all Cray Research systems, it runs the powerful Cray Research UNICOS operating system.

The CRAY XMS minisupercomputer system can serve as

☐ a minisupercomputer pathway into the full Cray Research product line
☐ a cost-effective UNICOS application development platform
☐ a Cray Research-compatible entry-level supercomputer at a price an individual department can afford
☐ a Cray Research-compatible platform for secure operations

The CRAY XMS system represents power in a small package, delivering 36 MFLOPS peak vector and 18 MIPS peak scalar performance. The system is available with one CPU and 64 or 128 Mbytes of four-ported, 16-way-interleaved Error Correction Code (ECC) memory. The system is compact, easy to install, and its low power requirements, high reliability, and high serviceability help minimize operating costs.

Powerful system software

The CRAY XMS system runs the UNICOS operating system, which is based on the AT&T UNIX System V operating system; UNICOS is used throughout the Cray Research supercomputer product line. The CRAY XMS system also runs the same powerful and productive compiler software as other Cray Research systems, including the CF77 Fortran compiling system and Standard C, along with a variety of useful library routines and utilities. Binaries from the CRAY XMS system will run on CRAY X-MP and CRAY Y-MP systems, allowing easy application transportability to Cray Research supercomputers.

Networking

The CRAY XMS system can be used as a stand-alone machine or networked into a Cray Research supercomputing environment. The system is supported by communications software and hardware interfaces, including the TCP/IP protocol, a widely accepted protocol for interconnecting UNIX systems, NFS, and the X Window System. Connectivity is supported by Ethernet and HYPERchannel connections.

A broad range of applications

The CRAY XMS system can run the same application programs that run on other Cray Research systems, with little or no modification. Cray Research is aggressively working with third-party applications vendors in many areas of engineering and science to license their software programs for the CRAY XMS system.

Modular I/O system

A wide selection of peripherals can be interfaced to the CRAY XMS system through a VME-based Input/Output Subsystem that communicates with the CPU via a 40 Mbyte/sec channel. ESDI disk drives and disk arrays operate at up to 16 Mbyte/sec and provide affordable, high-performance disk I/O.

Reliable support

System support is "on-call" and is enhanced by Cray Research's SCANbus diagnostic facility, which allows local or remote diagnosis using scan paths designed into the CPU boards. With SCANbus, faults can be detected quickly and accurately down to the component level.

Cray Research supercomputer users will appreciate the ease with which the CRAY XMS system integrates into their existing computer networks. For new customers, the system provides an affordable introduction to Cray Research's powerful UNICOS user environment. With the CRAY XMS minisupercomputer system, Cray Research provides users with a new high-performance computing option and extends its leadership in large-scale computing to a broader range of requirements. ▄▄▄

CORPORATE REGISTER

Cray Research gains new customers in Canada, France, Korea, United States

Thomson-CSF of Paris, France, has installed a CRAY X-MP/216 supercomputer at its Radar Countermeasures Division. The electronics firm is a new customer for Cray Research. The system is being used for scientific applications in electromagnetism, optics, and mechanics. The system runs Cray Research's UNICOS operating system, which is based on AT&T's UNIX System V, an industry standard.

Kia Motors Corporation, a Korean automobile manufacturer, has purchased a one-processor CRAY Y-MP4 computer system. This is the first commercial supercomputer order from Korea and the second sale in Korea by Cray Asia/Pacific, a wholly owned subsidiary of Cray Research. The system will be installed in Seoul in the fourth quarter of 1990, pending export license approval, and will be used for automotive design. The system will use Cray Research's UNICOS operating system.

Grumman Data Systems, under prime contract with the **Naval Oceanography Command** in Bay St. Louis, Mississippi, has ordered an eight-processor CRAY Y-MP computer system. The system is scheduled for installation in the fourth quarter of 1990 and will be used in the development and operation of global oceanographic and meteorologic models. The system will run Cray Research's UNICOS operating system.

Advanced Integrated Technology, under prime contract with **Eglin Air Force Base** in Fort Walton, Florida, has ordered a two-processor CRAY Y-MP8 computer system. The system, scheduled for installation in the third quarter of 1990, will be used for computational fluid dynamics and structural analysis research.

Exxon Corporation has ordered a CRAY Y-MP8 computer system, to be installed in the fourth quarter of 1990. The system will replace Exxon's CRAY X-MP/14 and CRAY-1S systems. The new system will run Cray Research's UNICOS operating system and will be used primarily for reservoir modeling, seismic processing, and finite element analysis.

Hydro-Québec, the principal Canadian supplier of hydroelectric power, has pur-chased and installed a CRAY X-MP super-computer system. Hydro-Québec is a new customer for Cray Research. The system is being used at the Hydro-Québec Center for Numerical Network Analysis in Montreal to simulate the behavior of the power transport network of Hydro-Québec as well as the network in the northeastern part of the continent. The system will run Cray Research's UNICOS operating system.

Cray Research announces new networking products and technologies

In July, Cray Research introduced the UNICOS Storage System, a new file server technology; the DS-41 Disk Subsystem, a new high-capacity disk subsystem; and new networking capabilities to support its network supercomputing strategy.

The UNICOS Storage System is a high-performance network storage management system that provides high-speed access to many files on diverse storage media. It is a feature of the current version of Cray Research's UNICOS operating system and can run on all Cray Research supercomputers. When run on the CRAY Y-MP2E system, the UNICOS Storage System transfers data up to four times faster than traditional file servers at prices competitive with traditional servers. The UNICOS Storage System supports industry standards such as UNIX and TCP/IP, enabling users to access equipment from multiple vendors. The UNICOS Storage System works with existing Cray Research disk technology, as well as with the new DS-41 Disk Subsystem.

The DS-41 Disk Subsystem incorporates highly reliable eight-inch disk technology to provide large storage capacity, fast transfer rates, low power consumption, and a small footprint. The unit was developed for use with CRAY Y-MP, CRAY X-MP, and CRAY-2 systems. The DS-41 Disk Subsystem offers users the option of disk "daisy chaining," a method of connecting multiple disk storage units to a single channel on a Cray Research system to maximize storage capacity. Each DS-41 Disk Subsystem can store up to 19.2 Gbytes of data and can transfer data to and from disks at a sustained rate of 9.6 Mbytes per second.

Cray Research also announced several new networking technologies: the HIPPI external communications channel, support for the FDDI networking standard, the Cray-based Network Monitor, and support for the OSI Version 1.0 interconnection standard.

The HIPPI (High-Performance Parallel Interface) external communications channel connects Cray Research systems to each other and to peripheral equipment, such as network adapters and raster display devices, at rates of 100 million bytes per second. FDDI (the Fiber Distributed Data Interface) is a proposed industry networking standard that enables users to interconnect high-performance computer systems and provides a high-speed environment for medium-performance local area networks. The Cray-based Network Monitor is an X Window System application suite that enables users to examine network connections and pinpoint problem areas. It enables an administrator to manage and control network connections that are local to the Cray Research system. The OSI (Open Systems Interconnection) Version 1.0 logically connects Cray Research systems to other systems running OSI protocols. The U.S. government has mandated the use of OSI protocols for all U.S. government systems, effective September 1990.

Symposium proceedings available

The papers presented at the fifth international supercomputing symposium sponsored by Cray Research, which was held in London from October 22-24, 1990, have been bound into a single volume, *Science and Engineering on Supercomputers*. The book is available through the publisher, Computational Mechanics Publications. To obtain ordering information in North America contact Linda Ouellette, Computational Mechanics, Inc., 25 Bridge Street, Billerica, MA, 01821; telephone: (508) 667-5841; fax: (508) 667-7582. Outside of North America contact the Order Department, Computational Mechanics Publications, Ashurst Lodge, Ashurst, Southampton, SO4 2AA, United Kingdom; telephone: 44 (0) 703 293223; fax: 44 (0) 703 292853.

⧉PPLICATIONS UPDATE

ADINA analysis system serves diverse applications

The ADINA system, from ADINA R&D, Inc., is a finite element package for the analysis of structural, heat-transfer, field, and fluid-flow problems. The system has been developed as a tool for linear and nonlinear practical state-of-the-art analysis and runs on all Cray Research computer systems under the UNICOS and COS operating systems. The ADINA system comprises three solution programs:
☐ ADINA — for displacement and stress analysis of solids and structures
☐ ADINA-T — for analysis of heat-transfer and field problems
☐ ADINA-F — for viscous fluid flow with or without heat transfer

These programs are fully supported by the ADINA-IN preprocessor, for free-format command language input, generation, and display of input data for ADINA, ADINA-T, and ADINA-F, and the ADINA-PLOT postprocessor for graphical and alphanumeric display of program input and output data for ADINA, ADINA-T, and ADINA-PLOT.

For the analysis of solids and structures, the ADINA system can be used to perform
☐ efficient linear analysis
☐ small or large deformations with linear and nonlinear material models, plasticity, high-temperature creep, rubber, and concrete
☐ automatic load stepping for static and dynamic analysis
☐ contact problems with variable contact areas and friction
☐ dynamic response
☐ frequency or response spectrum analysis
☐ heat-transfer analysis
☐ fluid-structure interactions

For the analysis of fluids, the ADINA system can be used to model
☐ flow in porous media
☐ inviscid and viscous laminar and turbulent flows
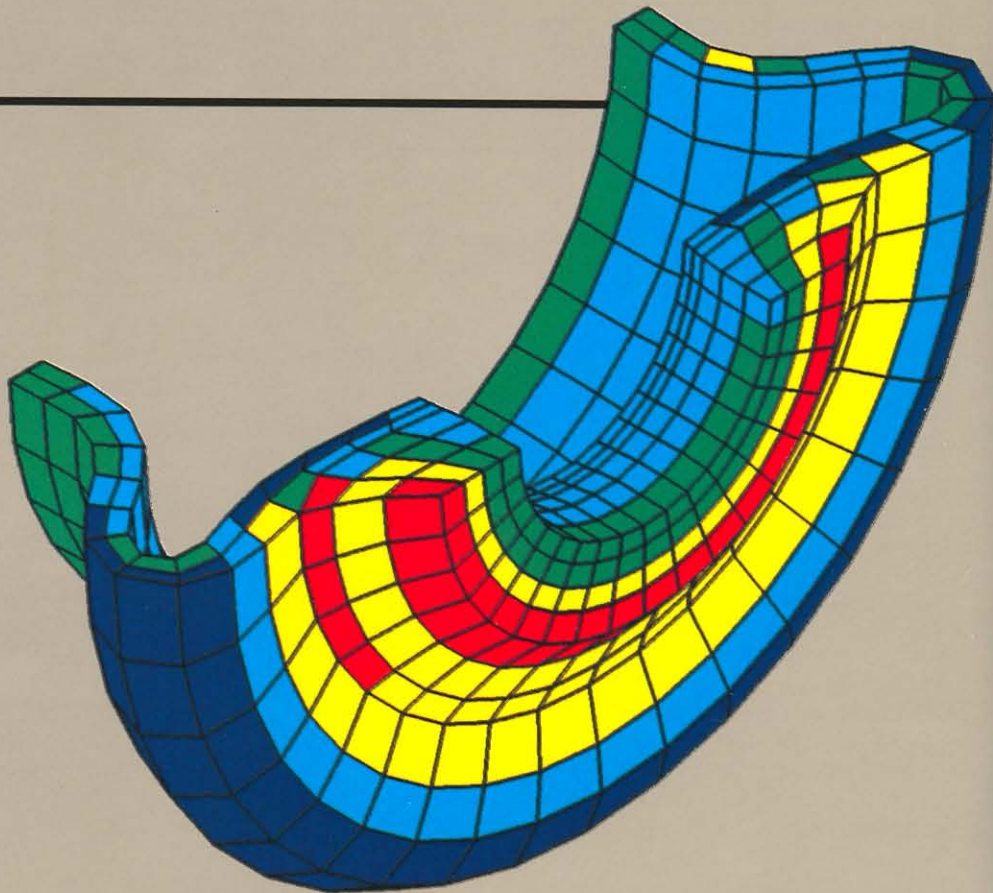
Pressure band plots of hydrostatic pressure modeled with the ADINA software package, from an analysis of a rubber coupling on a steel axle.

☐ heat transfer in fluids: forced and natural convection
☐ fluid flow coupling with solids

Enhancements incorporated into recent releases of the ADINA system pertain to
☐ analysis of composite structures
☐ the analysis of two- and three-dimensional rubberlike components
☐ the solution of metal-forming problems
☐ the analysis of reinforced and pre-stressed concrete structures
☐ the analysis of fluid-structure interactions
☐ the solution of viscous laminar and turbulent fluid-flow problems
☐ the assessment of the quality of the solution results

☐ general pre- and postprocessing procedures in ADINA-IN and ADINA-PLOT that interface to the PATRAN and SUPERTAB packages

The ADINA system is applicable to general three-dimensional analysis and can be used in many areas of engineering, including civil, mechanical, aerospace, automotive, nuclear, geomechanical, and offshore. For more information about using ADINA on Cray Research computer systems, contact Jan Walczak, ADINA R&D, Inc., 71 Elton Avenue, Watertown, MA, 02172; telephone: (617) 926-5199; or Mike Long, Cray Research, Inc., 655-E Lone Oak Drive, Eagan, MN, 55121; telephone: (612) 683-3656.

A breath of fresh air

Researchers from Cray Research and Canada's Alcan International, one of the world's largest aluminum companies, recently solved a key environmental problem in Alcan's production environment. By running computational fluid dynamics (CFD) simulations on a CRAY Y-MP computer system, the researchers determined the ventilation patterns in a potroom, a large room in which smelting operations are carried out. The research team included C. Mark Read, Andrew M. Kobos, and Marc Dupuis of Alcan's Arvida Research and Development Centre in Quebec, and Kent Misegades of Cray Research.

The electrolysis process used to smelt aluminum gives off a mixture of gases and fumes, a small part of which may be released into the potrooms. These fumes must be controlled by the potroom ventilation system. The research problem involved modeling air flow around large, hot obstacles (up to 120° C) variously arranged. After investing heavily in physical modeling, Alcan engineers decided to extend their knowledge with mathematical modeling. The results, particularly when presented through advanced visualization and animation, made a significant contribution to the company's understanding of potroom air flow.

The FIDAP software package, version 4.0, developed by Fluid Dynamics International, was among those used to solve the Reynolds-averaged Navier-Stokes equations. Air pressure, velocity, temperature, and fume concentrations were computed for a wide range of inlet flow conditions and floor plans. The FIDAP program is the first commercial CFD code to be developed primarily on a Cray Research computer system. The result is a high-performance code that has a reduced memory requirement and can take advantage of the Autotasking automatic parallel processing feature of Cray Research's CF77 Fortran Compiling System. As a result, the code runs at more than 100 MFLOPS on some problems.

The ventilation problem was modeled in two and three dimensions. When solving the two-dimensional problem, the FIDAP code ran at a sustained 42 MFLOPS, 26 times the performance achieved in single, 32-bit precision on a Silicon Graphics Personal IRIS workstation. The much larger three-dimensional case ran at approximately 150 MFLOPS. "Where lower productivity is acceptable, the Silicon Graphics system might suffice for two-dimensional runs involving a small number of iterations," explained team member Kobos, a super-

Concentration of fumes from aluminum smelting operations in a potroom, modeled by researchers from Alcan International and Cray Research.

computing consultant at Alcan. "But several hundred iterations are required to model transient flows, and three-dimensional runs clearly require the power of a Cray Research system."

The project team used Cray Research's MPGS graphics package to postprocess the FIDAP files into animation sequences. The team members translated the result files into MPGS-format files by using translation programs already written for these packages. The resulting data sets were stored on the CRAY Y-MP supercomputer. The Cray Research system did the computation, and the results were displayed pictorially on a Silicon Graphics workstation and recorded on videotape for future study.

By visualizing and animating the results through many time steps, the researchers were able to identify regions of poor ventilation and places where fumes tended to linger. The effects of changes in the ventilation regime also were readily apparent. "The use of a Cray Research supercomputer is now a fact of life for us; it's a production tool that we plan to continue using," said research team member Read. Information gained from this study will play a key role in improving environmental conditions at Alcan's production operations.

Pipe dreams come true

In the past, efforts to optimize metal forming processes, such as tube and pipe rolling, relied exclusively on experimental and simplified theoretical methods. During complex metal-forming processes such as stretch reducing, mandrel rolling, or cold pilgering, tubes are hot rolled, extruded, centrifugally cast, continuously cast, cold rolled, or welded to achieve perfect round-

ness, constant wall thickness, and optimal elongation.

Numerous factors complicate or exclude accurate measurements, which are essential for optimizing processing parameters. Mandrel rolling, for example, requires temperatures of between 900° and 1000° C. Other considerations are variations in tension within the metal, compressive stress, deformation caused by the rollers, wall thickness tolerances, and axial stresses in the pipe or tube.

Mannesmann Demag, a subsidiary of the German industrial conglomerate Mannesmann, had been exploring ways to improve the quality of the pipes and tubes manufactured on the pipe and tube mills built by the company's Equipment Division in Mönchengladbach, West Germany. As part of its efforts to investigate variations in the parameters used during metal forming, the company was back-renting milling equipment from clients, a testing method that puts obvious limitations on the speed of innovation.

In the fall of 1989, Mannesmann Demag and Cray Research GmbH, Munich, initiated a joint hero project and embarked on a one-year cooperative effort to explore the feasibility of using finite element methods to model the milling process and to design milling equipment. In April 1990, after completing the first half of the pioneering work, it was obvious to Mannesmann Demag and Cray Research that the results exceeded expectations. The software consulting team

from Abacom (the European supplier of ABAQUS) and Marc Europe, whose support was essential for the success of this project, agreed.

Three-dimensional visualizations of the stretch reducing and mandrel rolling processes, produced by the Mannesmann Demag technical staff and Cray Research's Industry, Science & Technology department, not only proved the feasibility of using advanced computational methods for these complex industrial processes, but also represented a quantum leap in accelerating the optimization of production and design methods. "The numerical simulation of these metal forming processes is comparable in significance to numerical crash simulation in the automotive industry," said Axel Krapoth, an account manager with Cray Research GmbH in Munich.

The stretch reducing simulation was run on a CRAY Y-MP4/432 computer system in 39 CPU hours using the MARC software package. It analyzed the stresses and strains in the stretch reducing of hot steel tubes. The mandrel rolling simulation was run on a CRAY X-MP EA/464 system in 18 CPU hours using the ABAQUS software package. It analyzed the influence of wall thickness on a predeformed tube in a six-stand mandrel rolling mill. Preliminary computations performed between Munich, Mönchengladbach, and Cray Research in Mendota Heights, Minnesota, with the help of high-speed data networks and via satellite took over 200 CPU hours.

During the second phase of cooperation, between August and November 1990, Mannesmann Demag and Cray Research plan to optimize the code for the stretch reducing process further. In addition, a third, and considerably more complicated tube rolling process, cold pilgering, will be analyzed computationally, and a three-dimensional visualization will be completed.

Impressed with the preliminary results, Mannesmann Demag already anticipates shifting from experimental to computational methods for future design processes. "Our experience with this project has been so positive that Mannesmann Demag has decided to base the design of our proposed new tube rolling mill largely on numerical simulation methods and use traditional experiments only for the verification of these computed results," said Ralf Passmann, a mechanical engineer at Mannesmann Demag. Complete results of the year-long project will be published and presented at the beginning of 1991.

Researchers explore food processing applications

The U.S. food industry thermally processes over 100 billion cans of food each year to kill disease- and spoilage-causing micro-organisms. To reduce the energy costs associated with sterilization and improve food quality, food companies are exploring ways to reduce the over-processing that often occurs.

To investigate alternative processing methods and methods to scale processes from laboratory to production, researchers at the Southeast Dairy Foods Research Center (SDFRC) in the Department of Food Science at North Carolina State University are simulating processing conditions on the CRAY Y-MP computer system at the North Carolina Supercomputing Center.

New processing procedures must be approved by the U.S. Food and Drug Administration (FDA), which requires assurance that the procedures provide adequate thermal treatment. The present safety standard calls for a reduction of *Clostridium botulinum* spores by 12 log cycles. The corresponding heat treatment can be severe, but is justified when one considers the potential danger of the organism. A single surviving spore might replenish a sufficient bacterial population to produce enough toxin to kill a person within hours of ingestion. Overprocessing, however, costs the food industry time and energy and reduces the nutrient content of foods; as a result, thermally processed foods must be nutritionally fortified, an additional cost factor. Processing methods that would reduce overprocessing
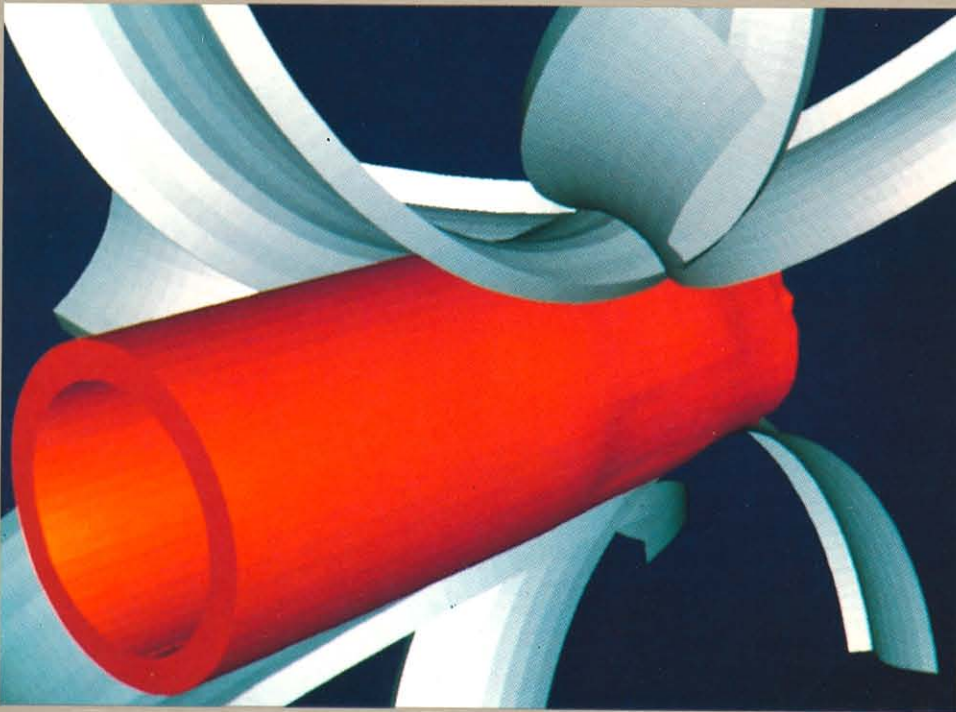
Simulation of the stretch reducing process run on a CRAY Y-MP4/432 computer system by researchers from Mannesmann Demag and Cray Research. One roller was removed to visualize the gradual diametric compression and elongation of a tube fed through the consecutive sets of rollers at temperatures of between 900° and 1000° C.

even slightly could provide a food company with a competitive edge.

Some food products, such as soups, and broths, are sterilized in pressure vessels called still-retorts, which use natural convection to spread heat throughout their contents. The complexity of natural convection heating makes it impossible to predict analytically the point in a can receiving the least thermal treatment. Determining the location of this region is crucial to the design of an optimal thermal process.

To better understand convection during thermal processing, researcher Ashwini Kumar, now at SDFRC, ran simulations using the fluid dynamics package FIDAP on a CRAY-2/4-512 computer system at the Minnesota Supercomputer Center, in Minneapolis. The work was part of his doctoral research at the University of Minnesota and was funded by a supercomputing grant from the Minnesota Supercomputer Institute.

The images at right show the computed temperature and velocity profiles for one of the simulations. This transient problem required between seven and eight hours of CPU time for completion. The simulation represented 2754 seconds of real-time heating of a model liquid in a cylindrical container in an upright position in a still-retort. A total of 297 time steps were required with two to five iterations at each step. Each iteration took from 23 to 24 seconds of CPU time. The model included the axisymmetric case of the governing, continuity, motion, and energy equations, along with temperature and a description of shear thinning viscosity. The grid was nonuniform with 850 nine-node isoparametric elements and 3519 nodal points. Because the FIDAP program is only partially vectorized, it did not take full advantage of the CRAY-2 system's extensive vectorization capabilities.

Nevertheless, the system's speed and memory capacity made practical the solution of the problem. Kumar also ran one of the simulations on a Hewlett-Packard 9000/840 minicomputer, but the processing time required made it infeasible to complete many runs. The CRAY-2 system's large memory made it possible to solve the problem without external storage and thus eliminated I/O time.

The results showed that no fixed point in the domain could be called the coldest point. Computed particle paths also revealed the movement pattern of the liquid in the can.

Increasingly, experts in the food industry are showing interest in a novel technology called aseptic processing and packaging (APP). This process involves sterilizing the food and the container separately, then bringing them together in a sterilized environment. The brick-style juice and milk
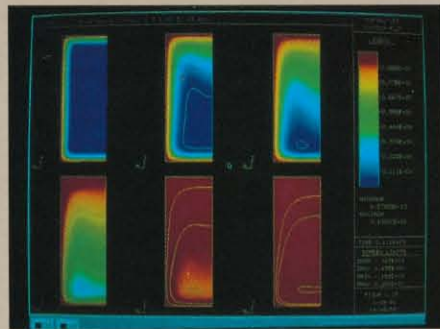
Temperature contour plots for natural convection heating of canned liquid food, computed by researchers from North Carolina State University. The model is axisymmetric; the right side of each figure is the axis of symmetry. Temperature is shown on a dimensionless scale where T = 0 at 313 K and T = 1 at 394 K.
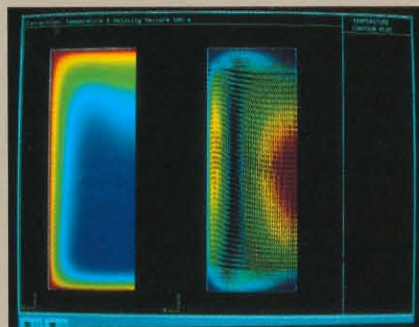
Temperature (left) and velocity (right) contour plots at 180 seconds of real-time heating. The model is axisymmetric; the right side of each figure is the axis of symmetry. Temperature is shown on a dimensionless scale where T = 0 at 313 K and T = 1 at 394 K. Red corresponds to high temperature or high velocity; blue corresponds to low temperature or low velocity.

cartons now appearing in grocery stores are among the first commercial applications of this process. APP has many advantages over conventional processes. It can improve product quality, allow food companies to introduce new package forms, and provide convenience by acting as a microwavable container.

However, the processing of foods that exhibit two-phase flow, such as soups with vegetable chunks, is not well understood according to Kenneth R. Swartzel, director of the Center for Aseptic Processing and Packaging Studies (CAPPS), also in the Department of Food Science at North Carolina State University. The major challenge in finding wider applications for APP lies in predicting the center temperature (cold spot) of the fastest moving particle in two-phase flow. This knowledge would aid in the design of APP treatments acceptable to the FDA.

The power of Cray Research systems coupled with appropriate software may provide a practical way to model moving particles in two-phase flows. This capability would benefit food companies and consumers by making quality processed food less expensive. To help develop this capability, researchers including Swartzel and Kumar are using the CRAY Y-MP system at the North Carolina Supercomputing Center to model food processing operations.

Another area of importance to the food industry, and the dairy industry in particular, is the scaling-up of processes from laboratory to pilot plant and commercial operations. Products from commercial processes often differ from those produced on the laboratory scale due to the lack of a reliable methodology and to variable thermal and physical conditions. Through modeling and experimentation, the researchers are developing methodologies to define a point of overall equivalent thermal and physical treatment that can be duplicated on a commercial scale to produce products identical to those produced on a laboratory scale. "Other areas in food processing and handling could benefit from simulations using Cray Research computer systems," said Kumar. "These include extrusion, freezing, thawing, moisture diffusion in packaged foods, and drying. Supercomputer modeling could lead to a better understanding of the processes involved and to improved process designs, which would lead to better quality food products." (Kumar and Swartzel presented a paper on food-processing applications on supercomputers at the Cray Research science and engineering symposium held in London in October.)

Reinventing the wheel

Engineers at Du Pont have successfully combined space-age composite material technology and Cray Research supercomputer power to make an aerodynamic wheel that is helping bicycle racers break performance barriers. While the wheel could only have been built using today's advanced technology, its roots are in an idea that is nearly a century old.

Developed in a joint venture of Du Pont's Advanced Composite Division and Specialized Bicycles, this composite wheel may be the most aerodynamically efficient wheel available. The wheel has three spokes and is made of carbon, glass fiber, and Du Pont's Kevlar aramid fiber molded over a foam core with an aluminum rim. Described by *Bicycling* magazine as a "rolling wing that slices the air," the wheel enables cycle racers to reduce their 100-mile race times by as much as 15 minutes.

"The original concept for this project really goes back to before the turn of the century and a man named Archibald Sharp." explained Mark Hopkins, one of the Du Pont engineers who designed the composite wheel. "He wrote a book in 1896 — *Bicycles and Tricycles: An Elementary Treatise on their Design and Construction*, that I keep on hand as a reference." In the book, Sharp described a four-spoked wheel that he claimed would have less air resistance than wheels with wire spokes.

"Of course, Sharp didn't have the materials or the Cray Research computer that we have to make it happen," said Hopkins. "Building a similar wheel presented Du Pont with new opportunities for the application of composite technology."

From the beginning, Hopkins and fellow designer Frank Principe established the safety of the rider as the primary design consideration for the wheel. The wheel had to be stronger than a standard wheel and provide good bicycle handling under all road conditions, including crosswinds. In addition, the wheel, including the axle and bearing hardware, had to weigh under 1250 grams. A conventional wire spoke wheel weighs about 910 grams.

"Our analysis showed that the performance benefit of making our wheel more aerodynamic was more than an order of magnitude greater than the penalty for the small weight increase," said Hopkins. "It's much more important for the wheel to be aerodynamic than it is for it to be light."

At racing speeds (25 to 35 mph) most of the relative resistance on the rider is caused by aerodynamic drag. To achieve an optimum profile, the designers used NASA data tables that indicate the lowest-drag airfoil shape for given air velocities. The resulting wheel design has about 50 percent open area, allowing crosswinds to pass through the wheel with little effect on the bicycle's handling.

After visualizing the external geometries of the wheel, Hopkins and Principe carved a model from a piece of Hopkin's plywood workbench. They used this model to build a prototype made of aluminum. Extensive wind tunnel testing of the prototype showed that it performed significantly better than any other available wheel, registering 75 grams of drag at 30 mph. By comparison, 36-spoke wheels registered 260 grams and solid disk wheels registered 100 grams.

An extensive analysis was conducted to engineer the layout of composite fibers required to achieve the target weight and to meet or exceed the structural strength of wire spoke wheels. The modeling was accomplished using PATRAN and MSC/NASTRAN finite element modeling software on Du Pont's CRAY X-MP/28 supercomputer.

Composite bicycle wheel designed by Du Pont researchers.

The detailed wheel model required 30 minutes of dedicated processing time.

The supercomputer analysis showed how various material factors affect wheel stiffness and stress distributions. The designers used this information to engineer the orientation of fibers within the composite to withstand the stress distribution.
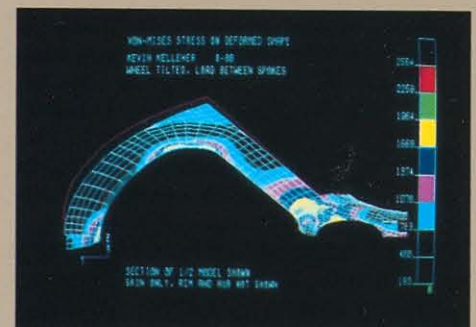
After the computational analysis, prototype composite wheels were built, stress tested, and test ridden. The prototypes were subjected to vertical loads of 1000 pounds without damage. Du Pont's tests showed that top quality, 36-spoke wheels buckle between 600 and 900 pounds.

The prototypes also provided excellent racing performance. Because of the reduced drag, many of the test riders claimed they could shift to a higher gear without increasing their efforts.

The composite wheels designed by Du Pont have been in production for over a year and are permitted in triathlons and U.S. Cycling Federation time trials, but not mass-start races. This restriction is under review.

The composite wheels have proven their speed in competition. Terri Schneider, an internationally known bicycle racer,


Du Pont's CRAY X-MP system predicted stresses on this partial wheel model. The scale shows pounds per square inch; colors correspond to loads.

tested a pair of Du Pont wheels in the 1989 Hawaii Ironman Triathlon Women's event. She improved her time in the 180 km (112 mile) cycling leg of the event by 15 minutes over her performance in the same event in the previous year.

Involvement in this project has reaffirmed Du Pont's success as a manufacturer of high-performance composite parts. By taking a ride back to the past with a supercomputer, the company has arrived at a state-of-the-art solution for the future.
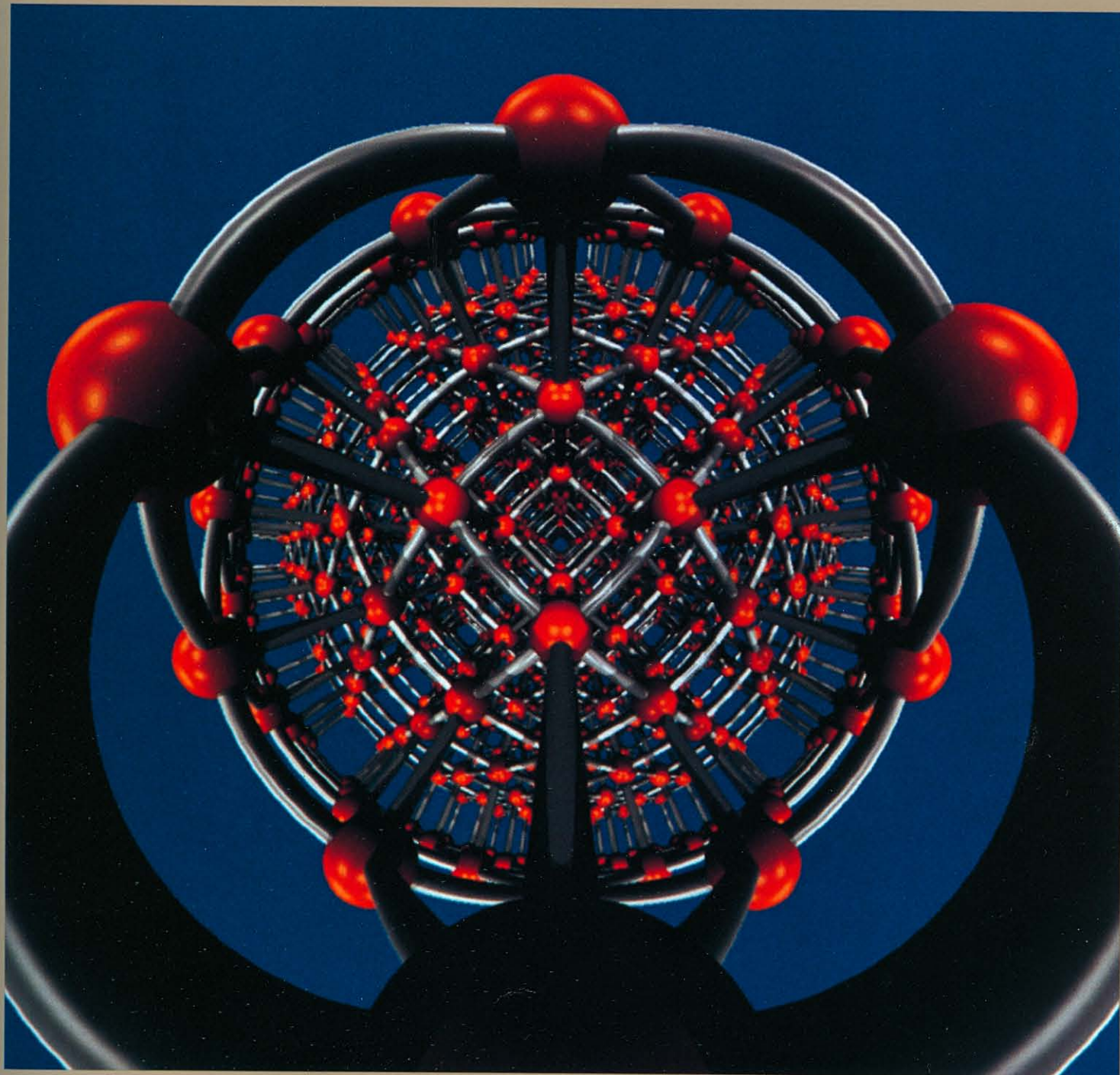
Seen as if traveling toward the viewer at 99 percent the speed of light, a rectangular lattice of spheres and rods is distorted by relativistic effects in this image produced on the CRAY Y-MP computer system at the Pittsburgh Supercomputing Center. The 1000-by-1000 pixel image required about 1000 CPU seconds to compute. Each pixel contains 24 bits of color. The image was produced by researchers Robert Dunn, P. K. Hsiung, and Mike Wu of Carnegie-Mellon University, and Joel Welling, a scientific specialist at the Pittsburgh Supercomputing Center.

CRAY CHANNELS welcomes Gallery submissions. Please send submissions to the address inside the front cover.