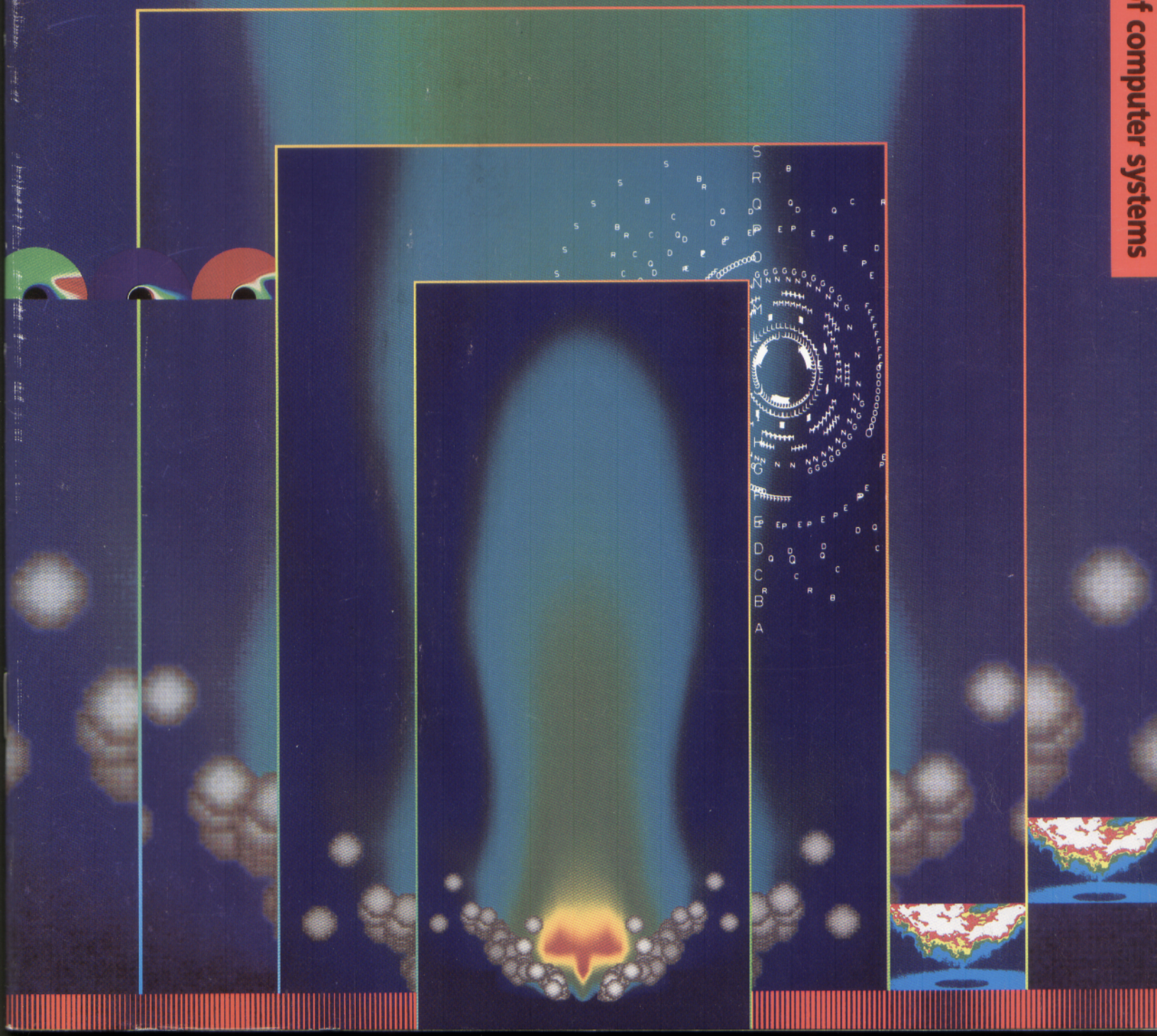
CRAY CHANNELS

SPRING 1989 · A CRAY RESEARCH, INC., PUBLICATION

Energy



Introducing the CRAY Y-MP series of computer systems

CRAY CHANNELS

In this issue

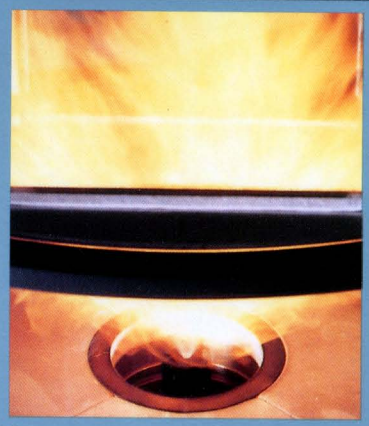
The energy crisis of the mid-1970s changed the world's attitude toward energy. We were reminded that Earth's natural resources are limited — and that we must efficiently use the resources we have as we develop alternatives.

Since 1978, when the National Magnetic Fusion Energy Computer Center at Lawrence Livermore National Laboratory became Cray Research's first energy research customer, Cray systems have played a significant role in the field. In fact, about one-fourth of Cray systems installed today are devoted to energy applications such as seismic data processing, reservoir simulation, and nuclear energy research and development. The result: petroleum companies are improving their exploration techniques and maximizing their production processes, nuclear energy is safer and cheaper to produce, and researchers are exploring long-term alternatives to our traditional energy sources.

This issue of CRAY CHANNELS is dedicated to energy applications of Cray systems. We look at advances in fuel combustion research at Sandia National Labs, nuclear plant safety and operator training at Electricité de France, and reservoir modeling at ARAMCO in Saudi Arabia. In addition, this issue introduces the CRAY Y-MP series of computer systems, a field-upgradable line of supercomputers that extends the availability of the CRAY Y-MP/832 system introduced last year. Our regular departments cover new system orders, applications software, and unusual applications in meteorology, music, and politics.

Cray systems are making energy production easier, safer, cheaper, and more environmentally sound. When energy producers apply supercomputer technology to their research and production, everyone benefits and the groundwork is laid to ensure an adequate energy supply for future generations.

Features



4

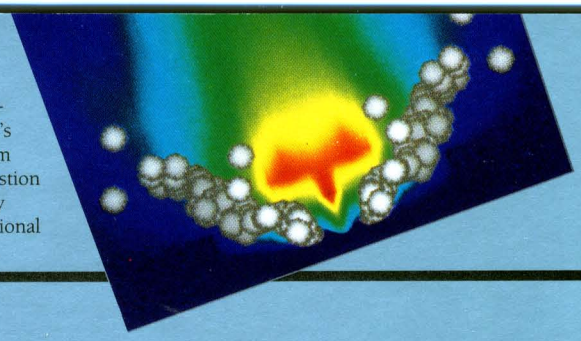
Departments

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

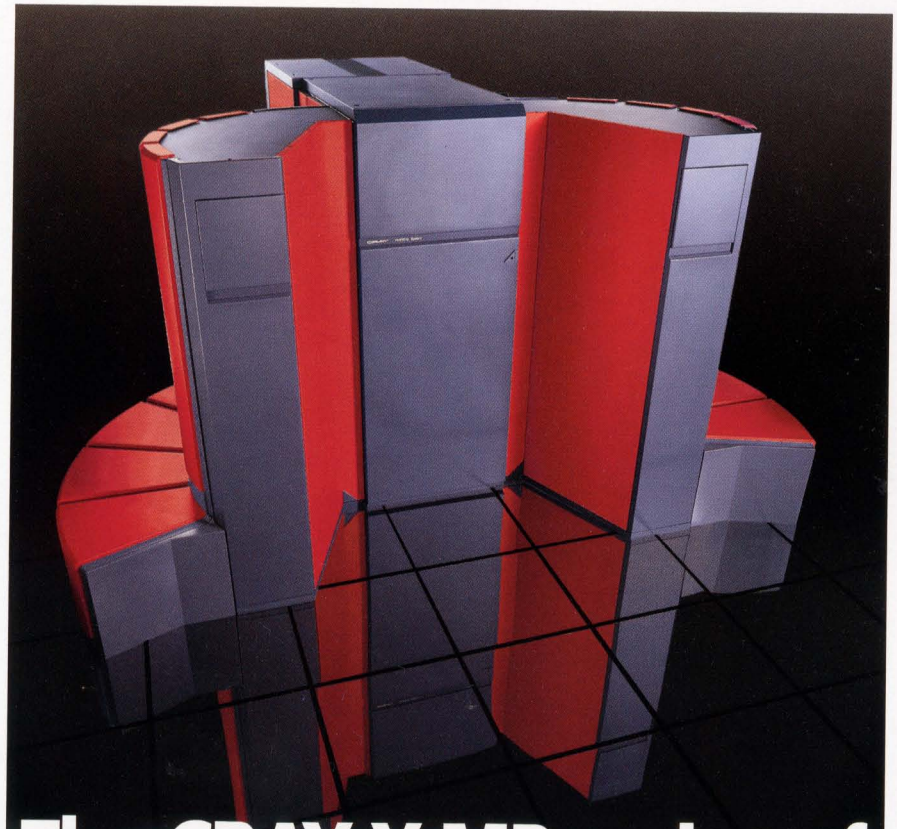
Volume 11, Number 1

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

Design and production
Barbara Cahlander
Eric Hanson
James Morgan
Cynthia Rykken



On the cover: Scientists and engineers are using Cray systems for research into hydrocarbon combustion, nuclear fusion, and other important energy-related applications. Combustion research aims to help preserve the world's critical hydrocarbon resources, while fusion research aims to find long-term solutions to the world's energy needs. Combustion images courtesy Combustion Research Facility, Sandia National Laboratories. Magnetic field plot courtesy National Magnetic Fusion Energy Computer Center, Lawrence Livermore National Laboratory. Solar corona courtesy Larry Rudnick, University of Minnesota.



The CRAY Y-MP series of computer systems

The next generation arrives

The CRAY Y-MP/832 computer system announced in February 1988 initiated a new generation of Cray Research supercomputers. The system brought together new hardware technologies and design innovations to provide users with unequaled supercomputer performance. Now the successful CRAY Y-MP design has been expanded into a series of systems, bringing this technology to the broad community of supercomputer users. The CRAY Y-MP series ranges from a single-processor system with 16 million words of memory to a top-end eight-processor system upgradable to 128 million words of memory.

Each model in the CRAY Y-MP series includes 2500-gate-array chip technology, a six-nsec clock cycle, and very high I/O bandwidth for fast data transfer. The result is markedly improved performance over previous systems, as demonstrated by benchmark runs for the NASA Ames Research Center. The center received the first CRAY Y-MP system delivered to a customer site: an eight-processor, 32-million-word system delivered in August 1988. To benchmark the system, NASA programmers assembled a suite of CFD codes approximating the site's workload. In all cases, the system exceeded the performance requirements set by NASA. In a peak performance test, the CRAY Y-MP system surpassed two billion floating point operations per second, or two GFLOPS.

"The CRAY Y-MP system repeatedly demonstrated that it could run at over one GFLOPS, both on multitasked individual codes and on a collection of codes approximating the site workload," said John Barton, advanced high-speed processor manager at the NASA Ames Research Center.

CRAY Y-MP systems deliver higher total throughput, greater reliability, and more flexible upgrade options than any previous series of Cray systems. A CRAY Y-MP processor is implemented on a single module comprising a stack of four 21.2-by-11-inch 12-layer printed circuit boards. Similarly, up to four million words of memory are implemented on a single memory module. The systems' modular design enables customers to add processors and memory capacity as their computing needs grow. In this way, customers can target their present needs precisely, yet retain the flexibility to expand their supercomputing resources in the future.

The CRAY Y-MP series comprises three mainframe models, each with several processor-number and memory-size options. The CRAY Y-MP8 system can be configured with up to eight processors and with 32, 64, or 128 million words of memory. The CRAY Y-MP4 system can be configured with up to four processors and with 16, 32, or 64 million words of memory. The CRAY Y-MP2 system can be configured with one or two processors and with 16 or 32 million words of memory. Each model is field-upgradable to its maximum processor number and memory size.

In addition, three single-processor CRAY X-MP EA/se models are available with 4, 8, or 16 million words of memory. These systems implement the CRAY Y-MP instruction set and have an improved clock cycle time of 8.5 nsec. The CRAY X-MP EA/se systems are specially packaged and priced to meet the needs of some first-time supercomputer users and of dedicated projects in large-scale computing environments.

I/O and external storage

All CRAY Y-MP systems come standard with an Input/Output Subsystem (IOS). The IOS for the CRAY Y-MP2 computer system is housed in its own stand-alone cabinet. The IOS for the CRAY Y-MP8 and CRAY Y-MP4 systems comes in a cabinet that forms one arm of the system's Y-shaped configuration. A second IOS can be configured with the CRAY Y-MP8 system. The IOS that comes standard with all CRAY Y-MP systems has a 4-million-word buffer memory that can be upgraded to 8 or 32 million words. Each IOS can access nearly 250 Gbytes of rotating mass storage using the advanced Cray DS-40 disk subsystem.

For expanded data-storage capacity, a CRAY Y-MP system can be configured with Cray Research's SSD solid-state storage device. This option is useful for solving problems that require extensive I/O or out-of-memory solution techniques. The SSD storage device is available with 32, 128, 256, or 512 million words of storage capacity. A CRAY Y-MP2 system can be configured with a 32- or 128-million-word SSD. CRAY Y-MP4 and CRAY Y-MP8 systems can be configured with any size SSD.

Cray Research's DS-40 disk subsystem and DD-49 disk storage units also are available for use with CRAY Y-MP systems. These magnetic storage

devices can sustain transfer rates of 9.6 Mbytes/sec at the user job level with an average seek time of 16 msec. The DS-40 disk subsystem comprises four DD-40 disk storage units, and has a storage capacity of 20.8 Gbytes. A daisy-chaining option enables users to double the storage capacity of the subsystem to 41.6 Gbytes. Cray DD-49 disk drives have a capacity of 1.2 Gbytes. When combined with the data handling and buffering capability of the IOS, these disks provide superior I/O performance.

Software

Cray Research offers a comprehensive set of software products, including the UNICOS operating system, language compilers, networking products, and a set of tools, utilities, and libraries. Cray Research's UNICOS operating system is an interactive system based on AT&T's popular UNIX System V operating system. The UNICOS operating system functions across the entire line of Cray Research hardware products. The availability of a single operating system across all Cray computer systems makes it easy for users to transport their programs from one Cray system to another. About 200 application programs have been written for or converted to run under the UNICOS operating system.

Cray Research's CFT77 Fortran compiler provides automatic vectorization, scalar optimization, and a capability called *Autotasking*, which automatically partitions individual programs so that the parts will execute simultaneously on multiple processors. On a four-processor system this automatic parallel processing feature can result in performance improvements of up to 3.9 times, and on an eight-processor system in improvements of up to 7.8 times, that of single-processor execution. Cray Research also offers compilers for the Ada, C, Common LISP, and Pascal programming languages.

Hardware

- ☐ Up to eight central processing units
- ☐ 16, 32, 64, or 128 million words of directly addressable central memory
- ☐ Six-nsec clock cycle
- ☐ Powerful I/O with optional second IOS with CRAY Y-MP8 systems
- ☐ CRAY X-MP compatible instruction capability

Software

- ☐ UNICOS operating system
- ☐ COS operating system with the UNICOS system as a guest operating system
- ☐ Compatability with CRAY X-MP and CRAY-2 UNICOS software products
- ☐ Vectorizing Fortran, C, and ISO Level-1 Pascal compilers
- ☐ Automatic parallel processing in Fortran
- ☐ Software for versatile network connectivity
- ☐ Optimized Fortran mathematical and I/O subroutine libraries
- ☐ Optimized scientific subroutine libraries
- ☐ The Cray macro assembler, CAL2
- ☐ A rich assortment of public-domain and third-party applications software

CRAY Y-MP CPU and memory configurations

Frame	CPUs	64-Mword	32-Mword	16-Mword
8	8	CRAY Y-MP8/864[1]	CRAY Y-MP8/832[2]	
	4	CRAY Y-MP8/464[1]	CRAY Y-MP8/432[2]	
4	4		CRAY Y-MP4/432[3]	CRAY Y-MP4/416[4]
	2		CRAY Y-MP4/232[3]	CRAY Y-MP4/216[4]
	1		CRAY Y-MP4/132[3]	CRAY Y-MP4/116[4]
2	2			CRAY Y-MP2/216[5]
	1			CRAY Y-MP2/116[5]

[1]Memory will be upgradable soon to 128 Mwords.
[2]Memory is upgradable now to 64 Mwords and will be upgradable soon to 128 Mwords.
[3]Memory will be upgradable soon to 64 Mwords.
[4]Memory is upgradable now to 32 Mwords and will be upgradable soon to 64 Mwords.
[5]Memory will be upgradable soon to 32 Mwords.

CRAY Y-MP systems deliver higher total throughput, greater reliability, and more flexible upgrade options than any previous series of Cray systems.

Connectivity

CRAY Y-MP systems can be integrated smoothly into multivendor networks. Cray Research offers network interfaces that provide point-to-point communication at the I/O channel level between Cray systems and computing equipment from many other manufacturers. In addition, multiple front-end systems can be configured with CRAY Y-MP systems by using channel adapters such as Network Systems Corporation's HYPERchannel and Computer Network Technology's LANlord. DEC offers a VAX Supercomputer Gateway that provides a high-performance direct connection between the DEC VAXcluster environment and CRAY Y-MP systems. The Cray Research fiber optic link allows a Cray network interface (FEI) to be separated from a CRAY Y-MP system by a distance of up to one kilometer (about .6 miles).

Support and maintenance

Cray Research recognizes the need to maintain high system reliability while delivering the highest level of system performance. The use of high density integrated circuits and an overall high level of component integration minimizes the number of components and connections in CRAY Y-MP systems, and thereby enhances system reliability. As a result, the reliability of CRAY Y-MP systems meets or exceeds that of previous Cray systems.

Before installation and throughout a system's lifetime, hardware engineering and system software support are provided on-site and by technical centers throughout the company. Cray Research also provides comprehensive user documentation for hardware and software products. Technical software training is offered to customers on-site and at Cray regional and corporate training facilities.

The CRAY Y-MP series of computer systems reflects Cray Research's commitment to provide scientists and engineers with comprehensive supercomputing solutions. High levels of multiprocessing; abundant I/O capability; large, fast-access memories; and field upgradability are combined in the CRAY Y-MP series to provide unsurpassed computational performance. Cray Research provides balanced computing capabilities, reliable hardware and software, and a high level of customer support to ensure that all Cray system users derive the most enlightening and profitable results possible.

Simulating fuel spray combustion

A look to the future

David Dandy, Harry Dwyer, Chris Edwards,
Ken Marx, and Billy Sanders
Combustion Research Facility
Sandia National Laboratories, Livermore, California

Figure 1. Steady kerosene spray flame in Sandia's Optical Access Research Furnace.

Obtaining a clear picture of the processes that characterize liquid hydrocarbon fuel sprays is one of the most challenging problems in combustion research. These complex, often unsteady reacting flows pose unique computational difficulties because they cannot be described adequately by continuum equations. However, some of the fundamental spray subprocesses are yielding to detailed numerical simulations, and promising computational models for overall spray behavior are under development.

A large fraction of the world's energy — about 30 percent — is derived through the combustion of liquid hydrocarbon fuel sprays. Given the environmental and safety issues associated with nuclear power and the abundance of petroleum that can be processed inexpensively, the demand for liquid hydrocarbon fuels is not likely to decline during the remainder of the twentieth century. Although fuel sprays play an important role in energy production, the ability to predict spray behavior in practical combustion devices is severely restricted by the lack of a fundamental understanding of spray transport processes. Even for well-defined, relatively simple laboratory spray flames, the question of what happens to millions of discrete liquid droplets as they interact with a turbulent air stream is beyond our ability to answer experimentally

or theoretically. However, large-scale scientific computations, carefully validated by experimental measurements, are beginning to fill the gaps in our knowledge.

Experimental studies of the structure of a steady kerosene spray flame in Sandia National Laboratory's Optical Access Research Furnace (Figure 1) provide important clues to the complexity of even the simplest spray flames.[1] Figure 2 is a composite of 500 exposures of flame luminosity digitally processed to provide information on the mean structure and spatial fluctuations of the flame zone. These two figures clearly show that the instantaneous flame contains a great deal of flow structure, and even the averaged picture shows a ragged, asymmetric spray flame zone. Attempting to model this flame structure computationally is similar to simulating the flame structure of burning logs in a fireplace.

The goal of our spray modeling research is to develop a true predictive capability for spray evolution and combustion, beginning with the initial conditions upstream of the fuel nozzle and terminating with the combustion of the vaporized droplets in real combustor environments. Clearly, such a predictive model will require enormous computational power beyond the capacity of present-day supercomputers. However, even if the computer power was available today, we lack a clear physical picture of a spray to serve as a basis for modeling. To build more complete models, we are researching spray evolution and transport using Sandia National Laboratory's CRAY X-MP/24 and CRAY-1 computer systems.

Defining the computational problem

By dissecting a spray we can examine some of its major components. This process will provide guidance as to how to separate the complex spray into computational tasks that are tractable with supercomputers — tasks that will provide crucial information for more comprehensive spray models.

Upstream of the fuel nozzle a flow of liquid is supplied either at steady velocity, as in a furnace or gas turbine, or in a highly unsteady flow pulse, as in a fuel-injected internal combustion engine in which the flow rate is high for a short period of time. Immediately downstream of the nozzle is a distribution of droplet sizes, velocities, and directions that depend on fuel flow rate, nozzle geometry, disruptive aerodynamic forces acting to break up the liquid issuing from the nozzle, and liquid surface tension forces that tend to oppose the effects of the aerodynamic forces. Slightly farther downstream, large droplets and liquid ligaments continue to break up under aerodynamic and surface tension forces, resulting in progressively smaller droplet sizes. At some point not far from the nozzle, the droplets would reach a steady-state size distribution in the absence of evaporation. The exchange of heat from the hot combustion environment to the cool liquid droplets provides the basis for overcoming the latent heat of vaporization, resulting in gaseous fuel that mixes and reacts with the hot oxidizing environment of the combustor.

At Sandia National Laboratories we are using our Cray computers to focus on several aspects of fuel sprays. We have adopted an approach developed by researchers at the Los Alamos National Laboratory,[2] in which the history of a large ensemble of droplets representing a spray is followed in a quasi-statistical manner. The gas-phase flow field solution in the Los Alamos procedure is obtained from the full set of unsteady partial differential equations representing mass, momentum, and energy transport, coupled to a two-equation turbulence model. This procedure treats the liquid phase by assigning statistical droplets to represent large numbers of similar droplets. The statistical droplets, or parcels, are transported throughout the computational domain by a Lagrangian procedure, and the interaction of liquid and gas phases is coupled in the overall solution procedure.

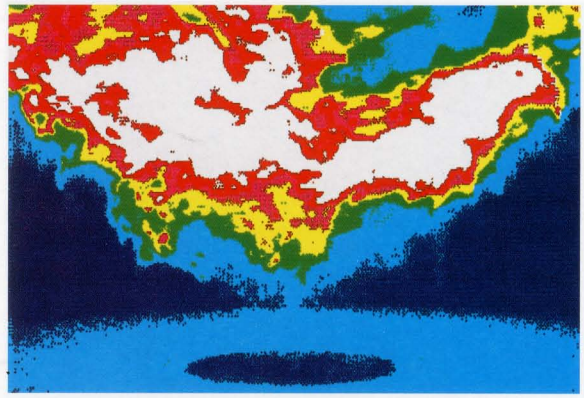


Figure 2. Composite image of flame location probability from 500 individual photographs of the flame in Figure 1. The highest probability region has been designated as red, and the lowest portion as black.

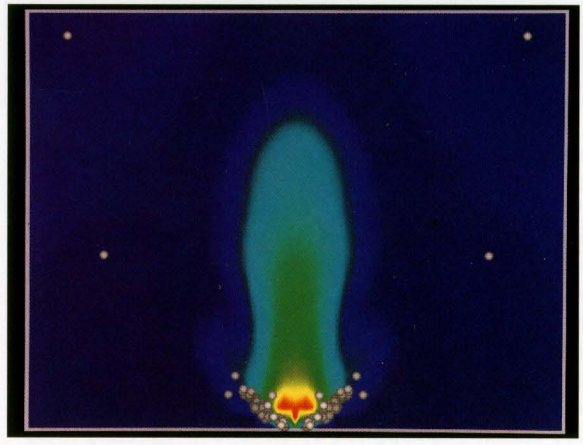


Figure 3. Simulation of a spray flame. Computational droplets, representing many individual droplets each, are superimposed on the continuous-color plot of the temperature field resulting from the combustion of vaporized liquid fuel.

Figure 3 shows a representative spray flame using this model, which was run on Sandia's CRAY X-MP system. The statistical droplets, each representing thousands of individual droplets, are shown superimposed on the color-coded temperature field. The fuel spray enters at the bottom of the computational domain, and here the number of droplets is so large the computational parcels overlap. Farther from the liquid injection region, the smaller droplets have evaporated completely, and parcels representing the larger droplets are less closely spaced. Combustion of the vaporized fuel was modeled through a single global kinetics step of fuel and oxygen reacting to produce combustion products and heat. The temperature field has been assigned a color scale in which blue is the lowest temperature and red the highest temperature in the computation. These results are representative of our initial attempts at modeling the laboratory spray flame shown in Figures 1 and 2.

This model will represent more accurately real spray physics and chemistry through the increase

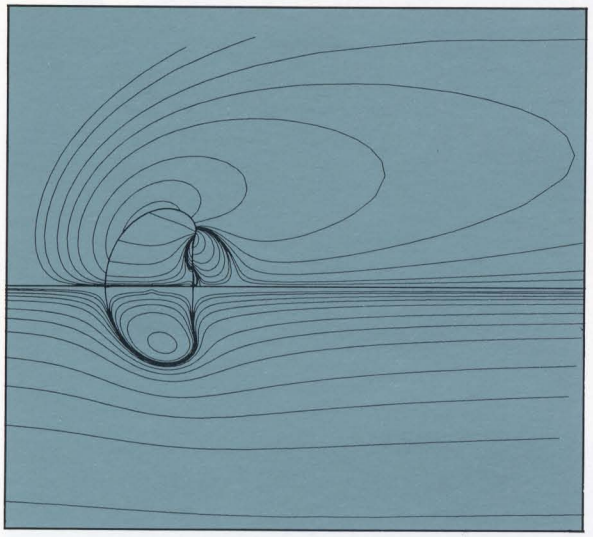
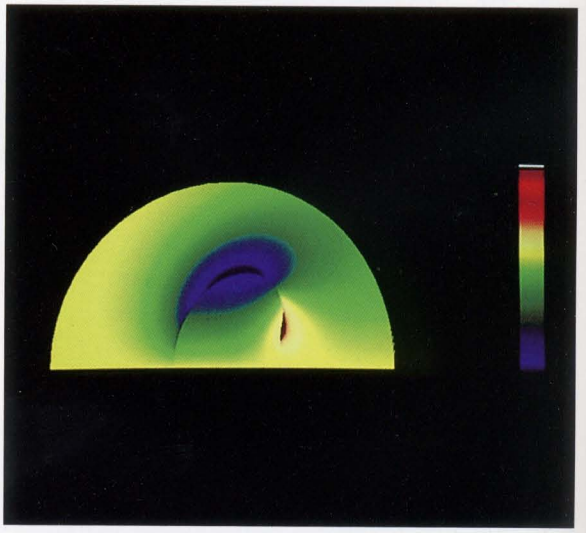


Figure 4 (Left). Simulations of a deformable droplet in an air stream. Vorticity contours are shown in the top half, streamlines in the bottom half.

Figure 5 (Right). Continuous color plot of the vorticity solution presented in Figure 4.



in computational resolution possible with larger memories, faster cycles, and multiprocessing capability. While we believe the Los Alamos procedure to be the best available upon which to build a comprehensive spray model, a number of areas need better physical descriptions of liquid-phase processes. For example, the current procedure requires a description of the droplet sizes, directions, and velocity distributions at the inlet boundary. Ideally, we would prefer to predict this input from first principles rather than relying on boundary conditions supplied by either experimental measurements or our best guesses. In the far field of the spray, droplets are transported into the combustion zone by their own momentum and by the influences of the surrounding mixture of fresh air and products of combustion. To describe the trajectories of the droplets accurately, or the statistical packets representing many droplets, one must specify relationships for drag on the droplets, the rate of heat transfer to the droplet liquid, and the resulting rate of evaporation. This information then is fed into the global spray model via algebraic relationships, which have been formulated using correlations of single-droplet experimental data or derived from detailed numerical simulations of droplet flow fields.

Liquid breakup near the nozzle

To provide information on initial liquid core breakup and subsequent breakup of large ligaments and droplets into smaller droplets, we have begun computations of idealizations of these subprocesses. Figure 4 shows the steady-state solution for an isolated, nonvaporizing, deformable liquid droplet in an air stream.[3] The conditions for this case are Reynolds number of 10, Weber number of 6, ratio of liquid to gas viscosities of 2.0 and ratio of liquid to gas densities of 0.9. The top half shows contours of vorticity and the lower half shows streamline patterns. Figure 5 is the vorticity contours of Figure 4 displayed in a continuous color plot, which clearly indicates the buildup of vorticity in the wake region of the deformed droplet. The computations in Figures 4 and 5 are two-dimensional due to the axisymmetric nature of the flow. Figure 6 is a complementary calculation in which a third spatial dimension is included: the liquid flow recirculation is omitted because the droplet is rigid.[4] This computed solution gives the surface pressure distribution on a nonspherical liquid ligament at an angle of attack with the oncoming stream. The next step is to add interfacial surface tension and follow the dynamics of the particle as it breaks into two or more smaller droplets. The results presented in Figures 4 through 6 demonstrate the direction in which we are moving, namely toward the detailed solution of a three-dimensional liquid element undergoing time-dependent deformation due to the unbalanced interfacial forces of surface tension, gas-phase-induced shear, and normal pressure imbalance. Additional physical processes will be incorporated in the liquid breakup model. Ultimately, it will take into account aerodynamic surface forces on liquid elements that are substantially modified by mass blowing at the surface due to evaporation, and internal liquid circulation, which strongly affects the temperature distribution at the droplet surface. The computations we have performed to date use a substantial portion of the CRAY X-MP system's memory, and require between 5 to 40 minutes each to execute. The entire solution, incorporating unsteady deformation and all of the other physics described, will require hours of computational time, and will be more properly suited to the Cray Extended Architecture system planned for installation at Sandia in October 1989.

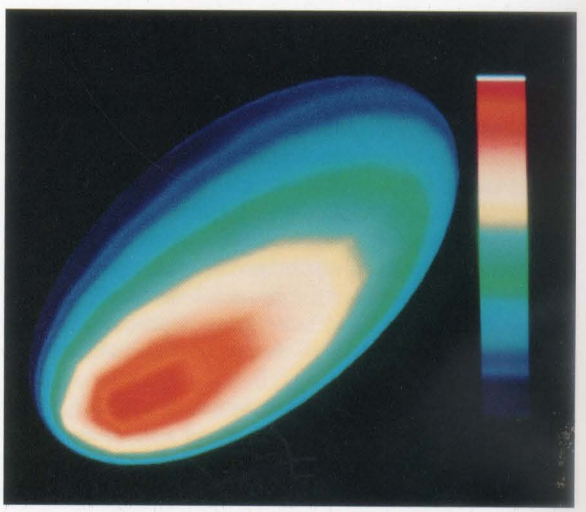


Figure 6. Three-dimensional pressure field on an elliptic liquid element at 45° angle of attack relative to the oncoming air stream.

Individual droplet processes

Over the past several years we have studied the two-phase flow associated with unsteady droplet vaporization and combustion.[5] We have isolated a single fuel droplet and followed its trajectory after it had been impulsively introduced into a quiescent, hot combustion environment. In these, like previous simulations, we have chosen to perform a Galilean transformation whereby the droplet was held fixed in space and the direction of the convective flow is left to right past the initially cold droplet. As heat was transferred to the cold droplet, the surface temperature increased and evaporation of the liquid fuel resulted. Due to the resistive forces of the approaching gas stream, the droplet velocity relative to the gas flow was reduced continually, causing a reduction in the relative Reynolds number. The decrease in droplet diameter from evaporation also aided in the Reynolds number unsteadiness over the droplet lifetime. The computation was made highly nonlinear by the influences of heat transfer to the droplet, internal liquid circulation, droplet boundary layer modification due to the addition of fuel and momentum into the boundary layer around the droplet, and the variable transport properties of liquid and gas due to species mixing and temperature variations. Because of this nonlinearity, great care is required to obtain accurate solutions.

The sequence in Figure 7 shows a representative computation in which gas-phase, single-step chemistry also has been included in the computation. Figure 7a shows the temperature field at a time when the Reynolds number has decayed to approximately 50, based on instantaneous droplet diameter and gas properties far from the droplet. Note that combustion heat release at this Reynolds number is primarily in the droplet wake, since at this condition the mixing of fuel and free-stream oxygen is not fast enough to allow combustion at the front of the droplet. Figures 7b and 7c show respectively the fuel and oxygen mass fraction contours at the same Reynolds number. Figure 8 shows the temperature field for the same droplet later, when the Reynolds number is approximately 19. For this state the reaction rate is sufficiently fast relative to gas-phase convection to produce combustion in front of the droplet, resulting in what is known as an envelope flame.

The future

In the not-too-distant future, comprehensive spray flame predictions will be possible, partly due to increased computing power, and partly due to better physical and chemical descriptions of the fundamental processes involved in liquid breakup, transport, evaporation, and combustion. At present, the framework for a spray flame model is in place, and computer simulations of individual droplet processes are providing detailed information about behavior near the nozzle and far downstream of the nozzle. Future studies must include the more complicated region between the nozzle and combustion zone, where the droplet density is too great to ignore droplet-droplet interactions. This region also offers the greatest experimental challenge because of the difficulty in penetrating this zone with modern, nonperturbing optical diagnostics. This article has focused only on the two-phase flow aspects of spray combustion; the chemistry of combustion and pollutant formation is being pursued concurrently at Sandia and other laboratories.

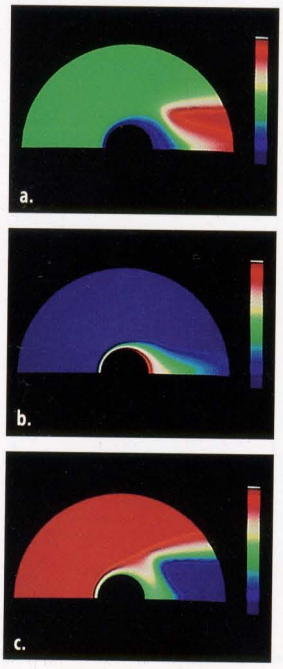


Figure 7. Computer solutions of an octane droplet at a Reynolds number of 50 burning in air: (a) temperature contours, (b) fuel mass fraction contours, and (c) oxygen mass fraction contours.

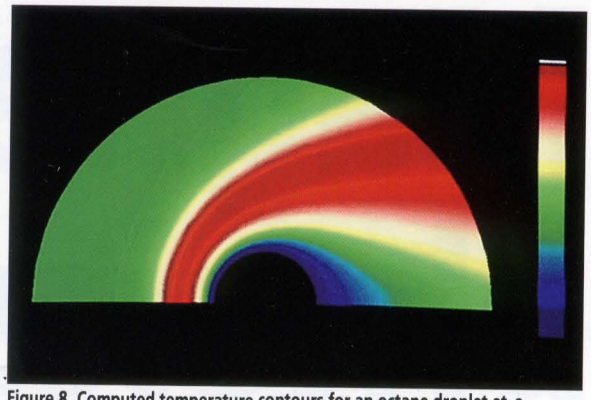


Figure 8. Computed temperature contours for an octane droplet at a Reynolds number of 19 burning in air.

Acknowledgments

The work presented in this article was supported primarily by the U.S. Department of Energy, through programs managed by the Energy Conversion and Utilization Technologies Program, and the Office of Basic Energy Sciences, Division of Engineering and Geosciences. The thesis work of David Dandy was supported by the National Science Foundation.

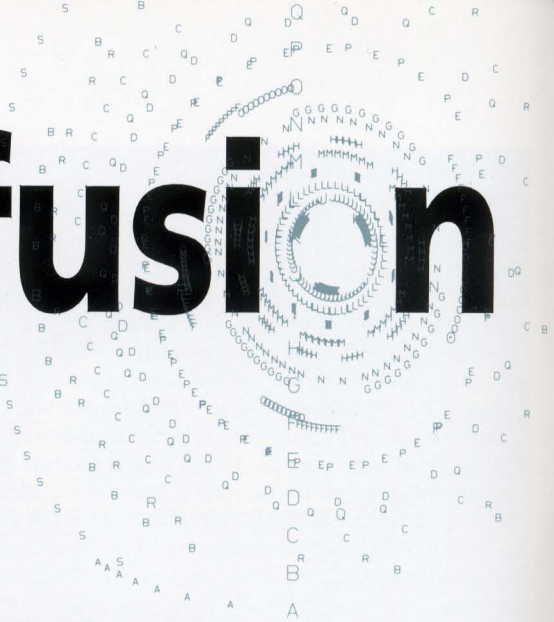
About the authors

The authors are employed at the Combustion Research Facility, Sandia National Laboratories. David Dandy received a Ph.D. degree in chemical engineering from the California Institute of Technology in 1987. Harry Dwyer, a professor of mechanical engineering at the University of California at Davis, is a consultant to Sandia. He received a Ph.D. degree in mechanical engineering from Rutgers University in 1966. Chris Edwards received a Ph.D. degree in mechanical engineering from the University of California at Berkeley in 1986. Ken Marx received a Ph.D. degree in applied science from the University of California at Davis in 1968. Billy Sanders received a Ph.D. degree in mechanical engineering from the University of California in 1974.

References

1. Edwards, C. F., "Investigation of Spray Flame Structure in Near-Axisymmetric Optical Access Research Furnace," Paper WSS/CI88-45, Western States Section/The Combustion Institute, March 1988.
2. Dukowicz, J. K., "A Particle-Fluid Numerical Model for Liquid Fuel Sprays," *The Journal of Computer Physics*, Vol. 35, Number 229, 1980. See also: Amsden, A. A., J. D. Ramshaw, P. J. O'Rourke, and J. K. Dukowicz, "KIVA: A Computer Program for Two- and Three-Dimensional Fluid Flows with Chemical Reaction and Fuel Sprays," Los Alamos National Laboratory Report LA-10245-MS, 1985.
3. Dandy, D. S., "Intermediate Reynolds Number Free-Surface Flows," Ph.D. thesis, California Institute of Technology, 1987.
4. Dwyer, H. A. and D. S. Dandy, "The Influence of Particle Shape in Lift, Drag, and Heat Transfer," Paper WSS/CI 88-93, Western States Section/The Combustion Institute, March 1988.
5. Dwyer, H. A. and B. R. Sanders, "Calculation of Unsteady Reacting Droplet Flows," *Proceedings of the Twenty-Second Symposium (International) on Combustion*, The Combustion Institute, 1988.

Magnetic fusion physics on Cray systems



Arthur A. Mirin
National Magnetic Fusion Energy Computer Center
Lawrence Livermore National Laboratory
Livermore, California

Nuclear fusion is one of several technologies that could help humanity meet its long-term energy needs. Most of the world's energy today comes from fossil fuels, which are being consumed more rapidly than they are being replenished. Providing for our long-term needs, therefore, requires the development of nuclear or solar energy. Present nuclear power facilities operate according to the principle of nuclear fission, in which large atomic nuclei divide into smaller ones. The breeding of fuel is necessary for nuclear fission to be a viable long-term energy source, however, because the natural supply of fissionable materials is limited. The fission breeder reactor, whose technology is much more at hand than that of fusion, is one possible solution. However, because of safety and security issues, many people view fusion as a more attractive long-term alternative.

Nuclear fusion is the natural process by which stars generate heat and light. It involves the fusing of lightweight atomic nuclei into heavier ones and typically results in the release of large amounts of energy. For the past 35 years scientists have been working toward building a power plant in which fusion is carried out in a safe and controlled way, with the resulting heat being used to generate electricity. Computer models have played a critical role in the fusion power plant design studies that have been conducted thus far. At the National Magnetic Fusion Energy Computer Center (NMFECC) at the Lawrence Livermore National Laboratory in Livermore, California, researchers are using a CRAY-2 system to model the complex physics that would operate in a nuclear fusion plant.

Physicists have identified two ways in which controlled nuclear fusion might be achieved — inertial confinement and magnetic confinement. Inertial confinement involves the bombardment of small fuel pellets with high-powered beams, causing the pellets to become very dense and very hot (as in a star), thereby initiating fusion. Magnetic confinement involves the injection of the fuel into a container that is surrounded by magnetic coils; the fuel ionizes, and the resulting charged particles must follow the direction of the magnetic field, which is oriented so that the particles will not escape from the container; heating of the fuel mixture then results in fusion. This article describes research into the physics of magnetic confinement.

Computer models of magnetic fusion

The first step in designing a fusion device is choosing a container vessel along with a configuration of magnets to provide the confining magnetic field. Once a vacuum magnetic field is given, it is necessary to determine whether or not the ionized gas, called plasma, can form an equilibrium state — that is, a balance between hydrodynamic and magnetic pressure. Because the plasma is a collection of moving charged particles, it forms its own magnetic field, which tends to cancel the applied field; hence a given configuration can hold only so much plasma. This limitation is quantified by the parameter β, defined as the ratio of the plasma pressure to the magnetic field pressure. Once an equilibrium is found, it must be determined whether or not that equilibrium is stable. This is a formidable task, because many types of instabilities must be investigated. Even if the plasma is stable, particle collisions can cause the plasma to drift across the magnetic field, leading to a degradation of confinement. This transport phenomenon has been one of the more difficult to understand and model. Because a confined, stable plasma will not undergo enough fusion without sufficient heat, one must determine how to heat the plasma in such a way that it maintains a confined, stable equilibrium.

In summary, the fusion physics issues that must be addressed are: (a) vacuum magnetic fields, (b) equilibrium, (c) stability, (d) confinement, and (e) heating.

The above problem areas are addressed with a wide variety of computer models, a comprehensive discussion of which is beyond the scope of this article. Areas (a) and (b) typically involve the solution of elliptic partial differential equations (PDEs), and areas (d) and (e) typically involve parabolic PDEs. Area (c) covers a wide range of phenomena and hence a variety of solution methods; in some cases an eigenvalue approach is used and in other situations an evolutionary approach is superior. For the most part these models involve the solution of coupled systems of nonlinear partial differential equations. Very often the plasma is treated as a fluid, and one evaluates the typical macroscopic fluid variables along with the magnetic and electric fields. In other cases the fluid approximation is not valid and a kinetic description is necessary. For example, to examine microstability or finite Larmor radius (FLR) effects, a particle-in-cell method may be used; and to model collisional processes, the plasma often is represented by distribution functions in phase space. No "supercode" exists that can be used to describe an entire fusion device because the spatial scales and time scales of relevance range over many orders of magnitude.

In this article we will examine more closely two state-of-the-art computer programs developed at NMFECC for use on the CRAY-2 computer system. The first of these codes, TEMCO,[1] uses the magnetofluid approximation and analyzes the stability and transport of the plasma. The second code, HQC,[2] examines fast kinetic effects by representing the plasma as a collection of "particles" and solving the appropriate equations of motion.

Modeling plasma stability and transport with TEMCO

TEMCO is a three-dimensional, single-fluid code that solves the primitive equations of compressible magnetohydrodynamics. We use the code to model the evolution of a plasma on a time scale relevant to magnetohydrodynamic (MHD) stability. The code's physics model includes resistivity, viscosity, and thermal conductivity, and the Ohm's law contains Hall terms. The code uses cylindrical coordinates (r, ϕ, z) and is applicable to either cylindrical or toroidal geometry. Coupled evolutionary partial differential equations for the density, temperature, velocity, and magnetic fields are integrated in time; the pressure, electric field, and current density are expressed in terms of the other dependent variables.

Finite difference approximations on a variably spaced mesh are used in the r and z directions, and a Fourier expansion is used in ϕ. Fourier convolutions are performed using a pseudospectral technique. The discretized equations are time-integrated using an explicit (leapfrog) algorithm with either operator splitting or implicitization of the diffusive terms. To allow larger time steps, a partially implicit algorithm may be used.

The explicit portion of the time advance is accomplished as follows. Both the current density and pressure are computed everywhere. The z fluxes and their derivatives then are computed at all values of z.

The CRAY-2 system makes possible fully three-dimensional electromagnetic particle simulations.

The dependent variables then are advanced one z line at a time for all r, ϕ; all other coefficients are computed as needed. Vectorization generally is performed in the r direction.

The partially implicit time advance is designed so that the various ϕ harmonics decouple (taking the viscosity and resistivity to be functions of at most r and z) and the z components decouple from the r and ϕ components. Fourier transforms are used in the z direction, and the resulting tridiagonal systems (either scalar or 2-by-2 blocks) are solved using standard techniques. The decomposed system matrices are recomputed only when necessary.

Most of the main time-integration loop is multitasked. In undertaking such a strategy, one must decide where to place synchronization points — points at which the code will wait for all outstanding tasks to complete. From a performance standpoint, the work is best partitioned between synchronization points into N tasks of approximately equal duration, where N is the number of CPUs. For the CRAY-2, N is equal to four.

For the problem at hand, it is most convenient to place a synchronization point immediately after the explicit solve and before the implicit coefficient computation and time advance. This is because the explicit solve is carried out one z at a time, whereas the coefficients of the implicit systems require updated information at several values of z. It was decided to place another synchronization point between the implicit coefficient computation and the implicit time advance since the coefficients could be computed most optimally one z at a time (a number of the linear systems share similar coefficients), whereas the linear systems could be inverted most efficiently system-by-system. In the explicit portion of the code, partitioning the mesh generally is preferable to partitioning the dependent variables, because several variables share similar coefficients and because we do not always want to advance all of the dependent variables. It is necessary to surround the computation of z derivatives with synchronization points, because for cylindrical problems the computation of z derivatives invokes the use of Fourier transforms, which couple all z mesh points. It is convenient to place a synchronization point after the current density and pressure computation because that particular module is used elsewhere in the code. All in all, this results in a division of the main time integration loop into six multitasked sections.

TEMCO has been applied to a wide variety of fusion devices. One of these devices, called a spheromak, is considered an "alternate concept" device. As compared to the tokamak, which is the conventional approach to magnetic fusion, the spheromak is more compact and does not have magnetic field coils in its interior, making it advantageous from an engineering standpoint. We present an example of an application of TEMCO to the spheromak.

As pointed out earlier, one of the more important physics issues is that of confinement. Charged particles tend to follow the direction of the magnetic field. Therefore, configurations consisting primarily of closed, nested magnetic surfaces are good for confinement. Changes in magnetic field topology are likely to result in an increased loss rate and hence a degradation of confinement.

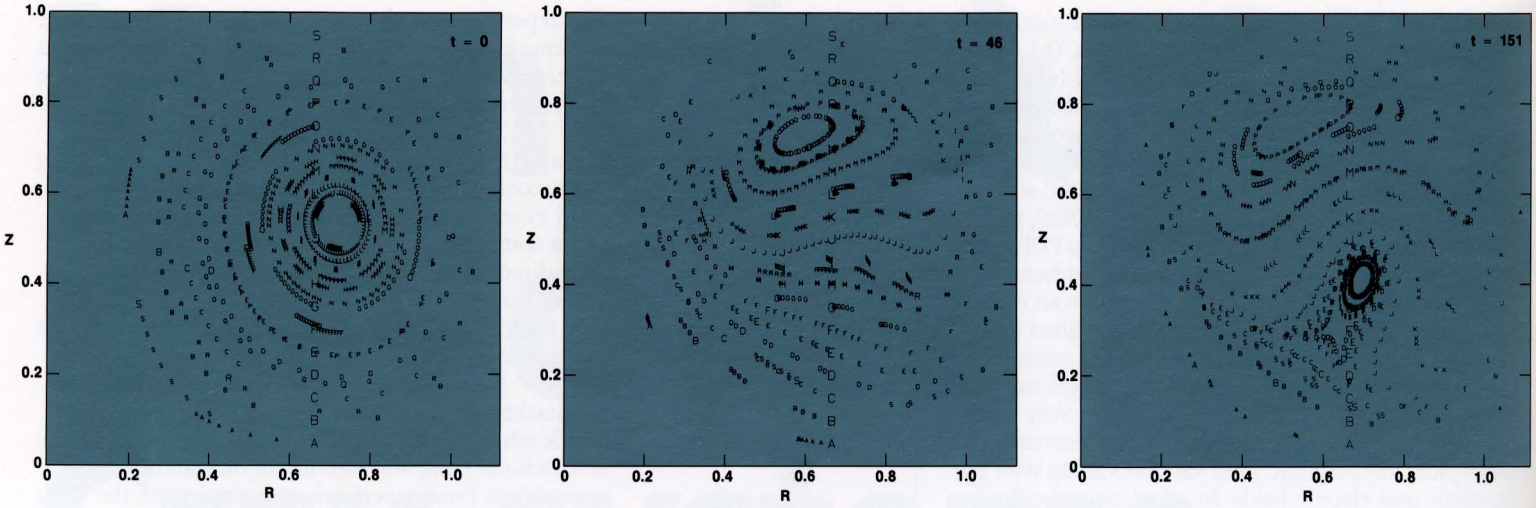


Figure 1. Poincare plots of the magnetic field in a spheromak, as computed by the magnetohydrodynamics code TEMCO. Results are at (left) t = 0, (middle) t = 46, and (right) t = 151.

Figure 1 depicts the evolution of the magnetic field in a spheromak, based on a simulation using TEMCO. Each image plots the intersections of representative magnetic field lines with a cross section of the torus. All points of the same letter correspond to the same magnetic field line and the collection of such points represents the intersection of that "magnetic surface" with the cross section. At $t = 0$ (t stands for an arbitrary time unit), the magnetic surfaces are closed and nested. As time progresses, the magnetic surfaces move upward and begin to deform. By $t = 46$, the upward motion of the primary confining surfaces has ceased. As time increases, the upper magnetic surfaces diffuse and the lower ones pinch upward. By $t = 151$, magnetic reconnection has taken place; a new magnetic axis forms and is surrounded by nested closed surfaces. As time increases further, the original surfaces diffuse and the new nested structure dominates. During this time, the plasma, which is confined in the original structure by the magnetic field lines, will have been deposited near the walls of the device. This leads to a severe loss of energy from the system.

Modeling kinetic effects with HQC

We now investigate the simulation of a plasma on a time scale relevant to studying ion kinetic effects. We use HQC, which is a three-dimensional particle-in-cell code written at NMFECC expressly for the CRAY-2 system. The type of fusion device under study is known as a field-reversed configuration (FRC). The FRC, like the spheromak, is an alternate concept device that does not have magnetic field coils in its interior. The FRC is more elongated than the spheromak and admits somewhat simpler equilibrium magnetic field configurations.

Magneto-fluid codes of the type described in the previous section predict that the FRC should be unstable with respect to the "tilt mode;" that is, on a time scale shorter than the desired experimental lifetime, the FRC is expected to tilt on its "side" and collapse. Yet existing experiments do not display this phenomenon. The prevailing theory is that the MHD approximation leaves out important physics, and that the effects of the ion gyro-orbits cannot be ignored. Furthermore, it is believed that the extent to which the MHD theory disagrees with the kinetic theory depends on the quantity s, which is the ratio of the plasma radius to the ion gyroradius. In present FRC devices, s is relatively small so that the ion kinetic effects are likely to be quite relevant. However, in reactor-sized plasmas, s tends to be large, and such plasmas are much more realistically described by the magneto-fluid model, so fears exist that an FRC reactor would be unstable. This possibility has led to the use of a kinetic model to try to explain the effect and to ascertain its severity.

The ions, instead of being modeled as a fluid, are represented as a collection of particles whose equations of motion obey the Newtonian laws of motion. Each particle (which is on the order of a billion actual ions) is represented in six-dimensional phase space by three spatial coordinates and three velocity coordinates. The rate of change of a particle's position is equal to its velocity, and the particle's velocity changes according to its acceleration, which is due to both magnetic and electrical effects. In this particular application, the electron kinetic effects are not important. This, together with other factors, allows the electrons to be represented as a massless, pressureless fluid.

To compute the forces on a given particle, one must take into consideration the effects of all of the other particles in the simulation. It becomes prohibitive to do this directly as the number of particles gets larger (although such schemes are being looked at with massively parallel computers). The way around this problem is to represent the magnetic and electric fields on a grid and to update the grid quantities in accordance with the motion of the particles. In HQC this is made even simpler by ignoring light waves. In order for the grid to communicate with the continuum, interpolation must be performed. Thus, particle codes such as HQC have four main phases: particle integration, interpolation from particle positions to grid, field update (on the grid), and interpolation from grid to particle positions.

Over the years, particle codes have been limited in scope by the computational facilities available. Adequate representation of the plasma required that the particle data be buffered to and from disk every time step, and the interpolation phases of the calculation could not be vectorized efficiently due to the random correspondence between the grid and particle indices. However, the availability of the CRAY-2 system makes possible fully three-dimensional electromagnetic particle simulations. This is due to three important features of the CRAY-2 system: its large and fast random-access

memory, its multiprocessing ability, and its ability to vectorize gather-scatter constructs.

With HQC, data on several million particles may be kept in main memory. The tedious coding of triple-buffering (moving particles to and from disk while effectively utilizing the CPU) is eliminated along with the resulting waste of CPU time. Since the equations for advancing the Cartesian components of the particles' positions and velocities are of similar form and independent of one another, that phase of the calculation is easily vectorized and multitasked.

Because of the random correspondence between the particle and mesh indices, the DO loops that appear in the interpolation phases of the code use indirect indexing. These constructs are vectorizable on the CRAY-2 system and run about six times faster than the equivalent scalar coding. Moreover, all of the particle information being available at once (rather than being stored on disk) allows interpolation with still more efficient techniques. As with the "particle push," the independent nature of the interpolation phase leads to an easy division of labor among the CRAY-2 system's four CPUs.

The evolutionary equation for the magnetic field is principally a combination of Faraday's law and Ampere's law. It is second order and parabolic and couples the three components of the magnetic field. The vacuum region (which surrounds the plasma) is modeled as a resistive medium. To concentrate the grid points in the plasma region, cylindrical coordinates are used to represent the magnetic field components. A Fourier expansion is performed in the angular variable ϕ, and measures are taken to ensure that the field remains solenoidal. An explicit difference algorithm combined with an iteration step to handle the nonlinearities is being used, but because of time step constraints, use of an implicit method is being actively investigated. The large memory and multiprocessing ability of the CRAY-2 system should allow a wider choice of solution methods.

HQC is now applied to an FRC scenario in which s is equal to 1.6. Figure 2 shows density contours at $t = 0$ and $t = 3$ microseconds (μsec). The z direction is along the axis of the containment cylinder, and the toroidal structure is oriented in the x-y plane. Each figure contains two slices through the torus, 90° apart. At $t = 0$ the configuration is in equilibrium, but by $t = 3$ μsec it is tilting. The computed rate of tilt is 0.57 per μsec, which is half that computed by the MHD model of the previous section.

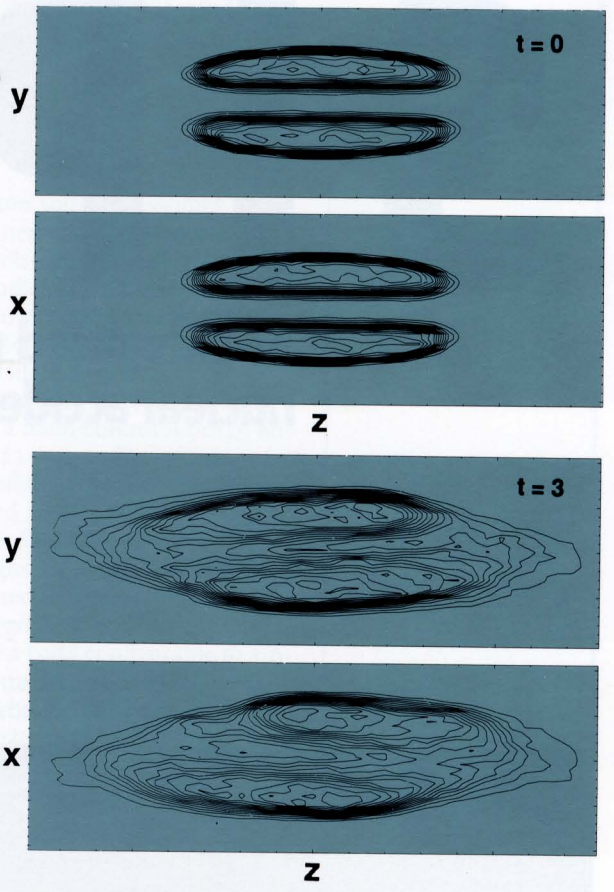


Figure 2. Density contours for the simulation of a field-reversed configuration plasma by the particle code HQC. Results are at (top) $t = 0$ and (bottom) $t = 3$ μsec.

Looking forward

If research into magnetic confinement fusion is to be successful, then improved models and computer hardware must be brought to bear on the understanding of fusion physics. Computational methods are critical to the advancement of this science because of the high cost of experimentation and the long lead times needed to build the experimental equipment. Only by applying the latest hardware and software tools can researchers study in a practical way the complex physics involved in fusion phenomena. And through their studies, fusion researchers hope to develop a clean and abundant energy source that will help meet humanity's long-term energy needs.

Acknowledgments

The application of TEMCO to the spheromak was carried out in collaboration with A. G. Sgro of Los Alamos National Laboratory. The HQC code evolved from a code developed as a doctoral thesis project by E. J. Horowitz of the University of California at Davis (now at the University of Maryland) under the supervision of D. E. Shumaker, D. V. Anderson, and J. Killeen of NMFECC. The work described in this article was performed under the auspices of the United States Department of Energy by the Lawrence Livermore National Laboratory under contract W-7405-ENG-48.

About the author

Arthur Mirin joined Lawrence Livermore National Laboratory in 1969 and has worked at the National Magnetic Fusion Energy Computer Center (NMFECC) since 1974. He is leader of the NMFECC Computational Physics Group and teaches at the Applied Science Department of the University of California at Davis. Mirin received an A.B. degree in mathematics from the University of California at Berkeley in 1969 and a Ph.D. degree, also in mathematics, from the University of California at Berkeley in 1974.

References

1. Mirin, A. A., "Predicting Multiprocessing Efficiency on the Cray Multiprocessors in a (CTSS) Time-Sharing Environment/Application to a 3-D Magnetohydrodynamics Code," *Computers in Physics*, No. 2, 1988, p. 62.

2. Horowitz, E. J., D. E. Shumaker, and D. V. Anderson, "QN3D, A Three-Dimensional Quasi-Neutral Hybrid Particle-in-Cell Code with Applications to the Tilt Mode Instability in Field Reversed Configurations," accepted by *Journal of Computational Physics, Lawrence Livermore National Laboratory Report No. UCRL-97685.*

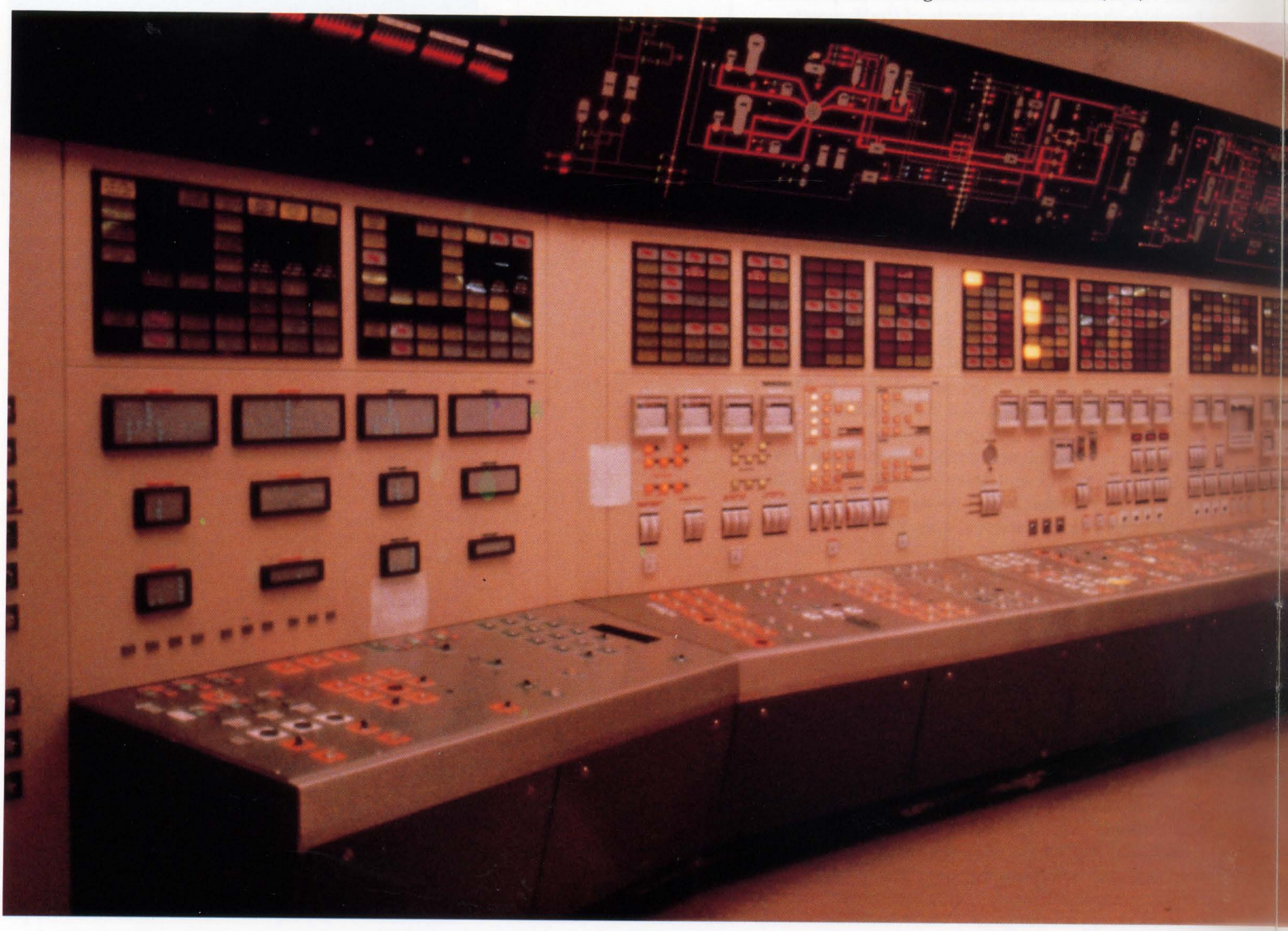
SIPA

An advanced project for nuclear accident simulators

J. Cohen, P. Cordel, F. Poizat, A. Sekri, J. M. Bernard, and J. P. Kirchgessner, Electricité de France

France's nuclear power industry supplies about 70 percent of the country's total electrical output. The industry's capacity derives almost exclusively from pressurized water reactors, which collectively produce about 50,000 megawatts (MW) of electricity. The French program is unusual in that it includes only one prime contractor and owner, Electricité de France (EDF), and only one nuclear steam supply system manufacturer, Framatome. This monolithic structure promotes a standardization that is unique in the world. More than 30 identical 900 MW reactors already are completed in France; 20 identical 1300 MW reactors are nearing completion; and four new "French-type" 1400 MW reactors are under construction. This level of standardization makes it much easier for the French safety authorities to supervise the design and operation of the plants. France's highly integrated arrangement constitutes what is generally referred to in the nuclear energy industry as "French cooking."

This article addresses the SIPA project, a design study and training simulator for post-accident conditions being implemented by EDF. The design studies involving SIPA will be conducted by EDF's design department for thermal and nuclear projects (SEPTEN), and the training will be conducted by the nuclear and fossil generation division (SPT).



The feasibility of the SIPA simulator arises from recent advances in several technical areas. These include increased and more cost-effective computing capacities; progress in networks, user-friendliness, and software engineering; improved sturdiness of numerical analysis; reduced calculation times; and the CATHARE research and development program for thermal and hydraulic analysis.

EDF expects the SIPA project to bring together systems design engineers, operations design engineers, operating advisers, and safety analysts from the French government.

Motivations

The SIPA project was initiated to meet a two-fold objective. The simulator will be used to assist in the training of power plant operators, especially shift safety advisers. This training function will include crisis drills. SIPA also will be used in design studies to better define safety margins, optimize operating points, upgrade automatic control devices, and for incident analysis and the development of rules and procedures covering incident and accident conditions.

Consequently, SIPA is designed to simulate severe transients or hypothetical accidents leading to fully developed two-phase thermodynamic conditions that are beyond the scope of the conventional training simulators used by the SPT. The heart of the SIPA project is the CATHARE-SIMU software, which is derived from CATHARE, the CEA-EDF thermal/hydraulic advanced code. The code has been optimized to achieve real-time computation.

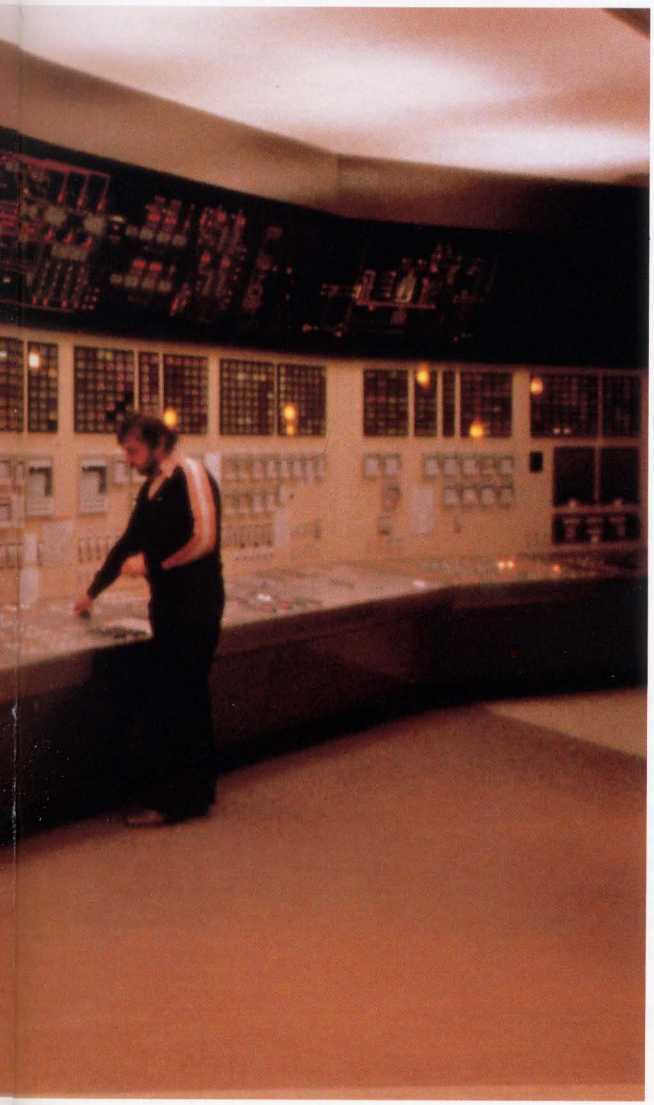


Figure 1. Main control room of a 900MW pressurized water reactor at Tricastin, France.

Accident simulation software assumes that the reactor core undergoes neither distortion nor meltdown. Core integrity is precisely the objective aimed for by design and safety studies and by the training of shift teams, whose task is to prevent an accident from becoming uncontrollable. Moreover, SIPA will allow the simulation of normal or incidental transients.

SIPA is not a "full-scope" simulator; that is, it is not an exact replica of a main control room of an actual nuclear unit (Figure 1). It is, rather, a computerized system whose flexibility allows the simulated configuration to be modified at will; this will put SIPA in a position to handle both 900 MW or 1300 MW units and preliminary designs for future reactors while varying the number of connected systems represented.

SIPA will be the first post-accident simulator validated by an advanced code configurable to any plant unit and dedicated to training and design studies. It will be installed at Villeurbanne close to Lyon, and run on a CRAY X-MP/28 system at EDF's research and development division (DER) research center, at Clamart near Paris. The French atomic energy commission (CEA) decided to partly duplicate the EDF project, with the aim of purchasing a "SIPA 2" that will be installed at Fontenay aux Roses near Paris and will run on the CRAY X-MP system at the CEA's Saclay nuclear research center.

Computing capacity needed by SIPA

Preliminary studies indicated that SIPA would require the processing power of a single central processing unit of a CRAY X-MP system (the CRAY X-MP/216 and X-MP/28 systems at Clamart are both two-processor systems). This level of computing power is required because of the complexity of the physical phenomena under simulation. Those phenomena relating to transients that result from a fully developed two-phase accident, such as the fast fracture of a primary pipeline, are particularly complicated. The high level of computing power is needed to compute simulations in real-time as much as possible. This determination was facilitated, in addition, by the fact that the CATHARE-SIMU code, which will be the basis for SIPA, already is running on Cray systems.

The estimated computing capacity equivalent to one CRAY X-MP processor is not an overestimation. Even that level of computing power will not allow real-time computations in every case. During the simulation of certain complex transients, the simulation will lag somewhat behind the physical phenomena under real conditions. This is referred to as the simulator "dawdling" period, which will be minimized. Having two Cray systems available for mutual back-up will ensure that SIPA is never unavailable, especially for training.

SIPA's computer hardware architecture (Figure 2) comprises two parts: the real-time simulator itself and an "engineering" Ethernet network with

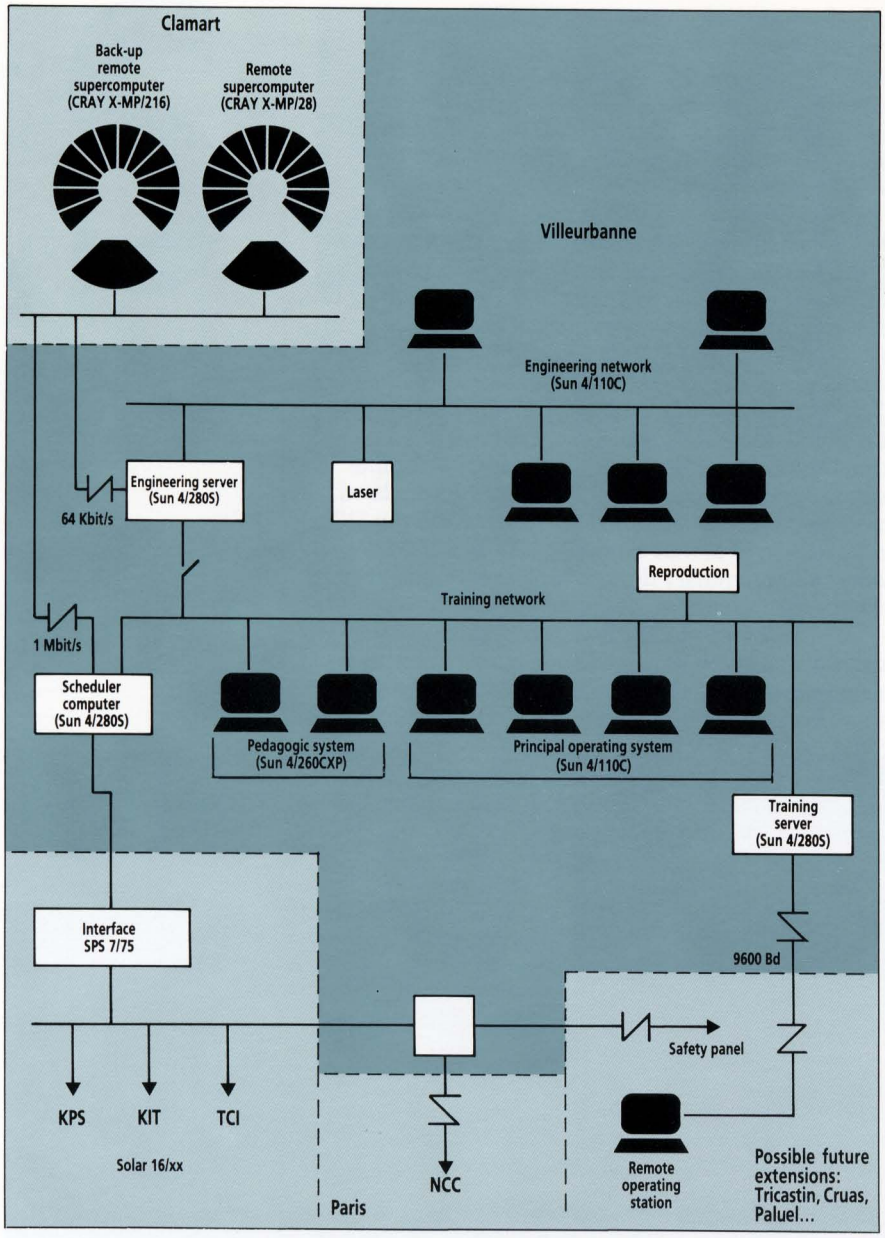


Figure 2. SIPA hardware configuration.

access to the Cray system through a server. The real-time simulator consists of the following parts:

- ☐ A scheduler computer
- ☐ A "training" Ethernet network supporting the principal operating system and the teaching system that displays the primary circuit and is serviced by a server
- ☐ A plant unit status logging computer (KIT) and a safety panel synthesizing key safety parameters (KPS) for 900 MW units and their equivalents for 1300 MW reactors (TCI), which are interfaced with the interface computer; this hardware is connected to the National Crisis Center to allow for realistic drills
- ☐ A CRAY X-MP computer system

The engineering Ethernet network will allow EDF engineers to make the usual job batch transfers to the Cray system and to have access to interactive functions outside the real-time simulator operating periods.

The scheduler computer, principal operating system, pedagogic system and the machines of the engineering network will consist of Sun Workstations running the UNIX operating system. The interface, KIT, KPS, and TCI computers will be systems manufactured by Bull Information Systems and will be served by an ARLIC network. Consequently, they will be identical to the hardware currently installed in French nuclear power plants. EDF is retaining the possibility of installing decentralized operating systems within French nuclear power plants for decentralized training sessions.

Operating principle of the real-time simulator

The scheduler computer orchestrates the simulations. It manages simulator time signals from the clock and synchronizes several operations. It assimilates the commands issued by the operator(s) of the principal operating system and accordingly submits requests to the Cray system to calculate the plant unit status at the next time step. The scheduler computer receives the unit status for the ongoing time step from the Cray system and disseminates part of this status to each of the principal-operating-system, pedagogic system, KIT and KPS stations, for updating the corresponding readout devices. The nominal simulator time step in SIPA is 500 ms, even though substeps are needed for CATHARE-SIMU running on the Cray system to carry out physical/numerical iterations of the software program.

Real-time use of the Cray system

The real-time application of the Cray system involves several steps. The scheduler computer at Villeurbanne uses the Cray system at Clamart as a "peripheral" that supplies it with CPU power. At each simulator time step of 500 ms, a request is sent to the Cray system in the form of a 1-Kbyte data block. The Cray system then must return its reply in the form of a 24-Kbyte data block, and this reply must reach the scheduler computer by the end of the next time step (outside of "dawdling" periods). This constraint raises two technical issues: the Cray system must be able to provide a "real-time" service with a response time of less than one second, and data transmissions (hardware and software) must guarantee the necessary flow rate (24 Kbytes/500 ms = 384 Kbits/sec).

Analysis has shown that these problems can be overcome with the use of Cray's UNICOS operating system and the design of fast network links. The UNICOS operating system allows a real-time application to be managed on a standard basis by the system. The effective 384 Kbits/sec needed for transmission can be obtained with a safety margin by designing an appropriate network using, for example, an NSC HYPERchannel network and a transmission line for 1 Mbit/sec (TRANSFIX line). This analysis has been verified on a mock-up highly representative of the final configuration, which was assembled at Clamart with the assistance of the companies chosen to implement the project.

Software structure

SIPA is a computer-based system for the management of simulation software programs. In other

words, it is a workshop for making special-purpose simulators. Its intrinsically evolutionary software architecture will be developed in two stages.

The first stage addresses the two configurations referred to as shift safety advisers because they are used for adviser training; these are the software programs handling the three- or four-loop primary circuit (CATHARE-SIMU) and those representing connected circuits, mainly the 900 MW and 1300 MW unit safeguard systems. These "physical" software programs are combined with

- ☐ Human-machine dialog software used to control simulated transients from the principal operating system
- ☐ Software for visualizing the two-phase behavior of the reactor coolant and used to serve the pedagogic system (Figure 3)
- ☐ CAD and graphical display software used both for illustration of phenomena and for the generation of Fortran modules for connected circuits and, finally, for their assembly as required
- ☐ Software for management of the data in the KIT, KPS, and TCI stations
- ☐ Utility programs for submitting applications (on a real-time, interactive, or batch basis)

The main software objectives of the SIPA project are portability and modularity, for both the calculation and the presentation of results.

The second stage, scheduled to finish slightly after the first, will be conducted by EDF and will complete and extend the design study configurations beyond the shift safety adviser configurations described above.

Execution

Execution of the first stage, involving both hardware and software, was awarded by SEPTEN and SPT partly to two French companies and partly to the applied mathematics and computer operations department (IMA) of DER, for networking to the two existing Cray systems. The companies are THOMSON-CSF, which will develop the scheduler computer and the engineering and training networks, and SEMA GROUP, which will develop the KIT-KPS-TCI stations. These manufacturers are the suppliers of all the EDF full-scope simulators.

SEPTEN, and probably the CEA, will be responsible for developing additional software. The time span allotted for development and acceptance of SIPA 1 is 34 months. The starting date was April 1, 1988, so a simulator should be operational in Villeurbanne by the beginning of 1991. The CEA-run SIPA 2 should follow SIPA 1 by less than one month.

The overall cost (outside contracts and EDF in-house costs for exclusive Villeurbanne supplies) is slightly over 100 million francs (U.S. $16 million), of which three-fourths is for software development. This is not such a high figure if it is compared with the cost of the CATHARE development and safety study program run by EDF, CEA, and Framatome (about 1000 million francs, or U.S. $160 million), of which SIPA is one outcome. The cost is all the more reasonable considering the investment figure for the gigantic French nuclear program, which is close to 50,000 million francs or about U.S. $8 billion.

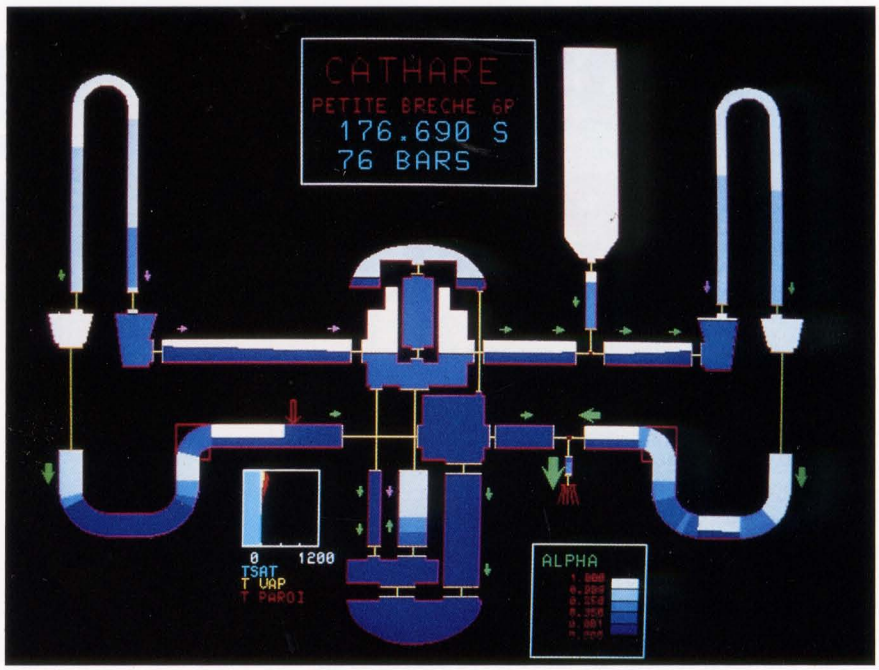


Figure 3. CATHARE image showing a 6-inch loss-of-coolant accident in a 900 MW pressurized water reactor.

Conclusion

The SIPA project is the outcome of recent advances in simulator techniques. Its configurability is the result of state-of-the-art techniques for computer-based project implementation, including modularity of software and portability, and of the choice of workstations in independent, connectable networks. Real-time and remote use of Cray systems will provide the necessary realism. The drive for portability, modularity, and adaptability to customer requirements should translate into quality assurance for design studies and for operator training. ▬

About the authors

The authors are researchers at Electricité de France. José Cohen has an engineering degree from INSA Lyon and joined EDF in 1982. He is in charge of the computer and methodological research aspects of the SIPA project. Philippe Cordel has an engineering degree from the Institut Polytechnique of Grenoble. He joined EDF in 1976 and since 1984 has been in charge of the Software Development Group at SEPTEN. He contributed to the SIPA project definition. François Poizat has an engineering degree in nuclear science from Ecole Supelec. He is the SIPA project manager and has worked on numerous power plant design and construction projects at EDF. Ahcène Sekri has a doctorate degree in engineering from ENSP in Strasbourg. He worked on the CATHARE project prior to joining SPT in 1984. He is the SPT training division representative to the SIPA project. Jean-Marie Bernard has an engineering degree from Ecole Polytechnique. He is a computer engineer in the IMA Department, where he has managed several computer networking projects. He represents the DER on the SIPA steering committee. Jean-Paul Kirchgessner has an engineering degree from Ecole Centrale. He joined the IMA Department in 1971, where he has worked on finite element application programs. He is the IMA correspondent to the SIPA project team.

A front-tracking method for reservoir simulation

Brent Lindquist and Qiang Zhang, Courant Institute of Mathematical Sciences, New York, New York; and Yogeshwar Sharma, Cray Research, Inc.

When flowing fluids mix in a reservoir, sharp interfaces may develop. In such cases, the dynamics of those interfaces may dominate the characteristics of the flow. Viscous fingering and channeling are two such phenomena that are of concern to the oil industry. Viscous fingering occurs when the interface between two fluid phases, such as oil and water, becomes unstable due to a tangential velocity difference between the fluids.[1] This may cause premature injection fluid breakthrough, which often occurs at producing wells during secondary or tertiary stages of oil recovery. As a result, the displacement efficiency is reduced dramatically. Channeling is a related phenomenon in which the fingering is shaped or channeled by the higher conduction (more permeable) regions of the reservoir. This is particularly important in the vertical plane of a reservoir that may have layers of rock with widely varying permeabilities.

This article addresses the recent success in compiling and running the front-tracking code of Glimm et al. on the CRAY X-MP/416 computer system.[2] This represents one of the largest C codes that has been compiled and run successfully on a Cray system. More than 200,000 lines of C coding, excluding subsidiary FORTRAN libraries such as LINPAK and EISPAK, are linked into the resulting executable. The results for the simulation of a two-phase immiscible displacement process through a two-dimensional vertical section of a layered reservoir shows the capability of the front-tracking algorithm running on the CRAY X-MP system.

The example demonstrated here shows the displacement of an *in situ*, incompressible reservoir fluid (oil) by an injected, incompressible fluid of greater density, but lower viscosity (water). The displacement process is characterized by an interface that separates the regions of unflooded reservoir from the water-flooded area. In the flooded area, the two fluid phases, oil and water, coexist in a two-phase region that stretches behind the interface, back through the flooded region to the source of injected fluid. The injected fluid is not present ahead of this interface in the unflooded region, except for the naturally present connate bound water in the reservoir. Additionally, a certain percentage of the oil in the flooded region is immovably bound, and is termed residual oil. Assuming these are incompressible fluids, the amount of each fluid can be described in terms of its local volume fraction or saturation.

Most commercial simulators track saturation changes only. As a result, the sharp change in fluid volumes that occurs across the interface separating the two-phase flooded region from the single undisplaced phase is resolved only to within the size of a single discretization mesh element. For simulations of entire oil fields, a single mesh block may be tens to several hundred meters on a side. Any resultant fluid behavior connected with the interface and its motion effectively will be lost. In contrast, the front-tracking algorithm demonstrated here employs moving curves (in a two-dimensional calculation) of zero width to track these fluid interfaces. The interfaces are propagated by an exact solution to the flow locally in the vicinity of the interface.

The fluid saturations in the regions of the reservoir away from these interfaces are computed by conventional finite difference methods. The front-tracking method is thus a hybrid, allowing enhanced resolution of the fluid interfaces.

The front-tracking method

A large class of physical problems involves important discontinuities in the physical quantities. Front-tracking has been developed for the computation and resolution of such problems. A successful scheme has been developed that combines the flexibility and efficiency of general purpose hydrodynamics methods with the power to resolve these discontinuities accurately. In this scheme, front-tracking shows no appreciable inefficiency, as it represents only a small fraction of the overall computation.

The large class of physical problems that involves such discontinuities is governed by sets of hyperbolic equations. Such systems of equations evolve discontinuities in the solution even if the initial data are smooth. We observe that in the direction locally normal to the interface, the discontinuous profile in the computed variables resembles a step function. The specification of a system of hyperbolic equations for a problem in one spatial dimension having initial data consisting of a simple step function discontinuity in one or more of the physical variables is known as a Riemann problem. The propagation of the discontinuity interfaces then is achieved by the solution to the Riemann problem.

Much is known about the properties of such Riemann problem solutions. In some cases (the Euler equations of gas dynamics or several model reservoir systems) the solution is known exactly or is computable. In particular, the Riemann problem solution presents the full spatial and temporal chromatographic separation of the dependent variables in the hyperbolic equations. The terminology of chromatographic separation is most apt when considered in terms of reservoir flow, in which the dependent variables are the saturations of the fluid components. The solution of the Riemann problem gives the one-dimensional separation of the initial static data in terms of moving phases, in which the fluids mix, accompanied by the

appearance of discontinuity interfaces between different phase regions. The Riemann problem solution gives the one-dimensional speed of propagation of these discontinuities and the composition of the fluid phases in between.

When applied to a two-dimensional calculation, the Riemann problem approximation is assumed to hold at each point along the discontinuity interface and to be valid for only a short interval of time. Once each interface point has been propagated in the normal direction by the correct speed determined from the Riemann problem solution and the new fluid compositions set on each side of the interface, the process is repeated, with new local data at each point of the interface. Riemann problem solutions can be computationally expensive, since the solution in the regions away from the discontinuity interface is obtained by conventional finite difference schemes that employ as "boundary" data the solution developed at the interfaces from the Riemann problems.

In summary, the essential features of the front-tracking implementation are:

- ☐ The use of a one-dimensional, time-dependent grid to track the position of the discontinuity interfaces in a two-dimensional calculation
- ☐ The use of an accurate analysis (Riemann problem solutions) on nonlinear wave modes and interactions to propagate these interfaces
- ☐ The coupling of the tracked interfaces with a finite difference method to compute the solution in the "interior" regions away from the discontinuities
- ☐ The use of coherent, computationally efficient, and user-oriented data structures for the storage and manipulation of the discontinuity data

In addition, for reservoir flows, the system of hyperbolic equations is coupled to a parabolic or elliptic equation for a pressure field. The presence of sharp discontinuity interfaces in the hyperbolic variables implies discontinuous coefficients in this pressure equation. The current implementation of the front-tracking method for reservoir equations treats the pressure equation as parametrically coupled to the hyperbolic system through the saturation variables, and solves the pressure equation separately using a finite element formulation on a mixed mesh of triangles and rectangles that align with the discontinuity curves.[3]

An example calculation

To demonstrate the front-tracking method, we show a two-dimensional calculation of a water displacement process in the vertical plane of a layered reservoir. The reservoir cross-section is assumed to be rectangular, 100-by-1000 meters, tilted at 10° to the horizontal (rising from left to right). It consists of three layers of depths 40 m, 20 m, and 40 m. The rock properties (such as porosity, permeability along the layer and vertical to the layer, relative permeability to the flowing fluids, and percentages of immobile oil and water) are constant for each layer but differ from one layer to the next. In particular, the upper layer is much more permeable than the middle layer, which is more permeable than the lower layer. An initial aquifer of water lies at the bottom (left) of the tilted plane. Water is then injected steadily through the aquifer (that is, through the left edge of the bottom layer) and fluid production from the reservoir is allowed to occur from the right side of the top layer. The initial oil-water interface is thus present and its development is tracked. The finite difference scheme resolves the two-phase (oil, water) region that develops behind the interface.

A successful scheme has been developed that combines the flexibility and efficiency of general purpose hydrodynamics methods with the power to resolve these discontinuities accurately.

Assuming the oil and water are incompressible, the set of equations governing the flow under consideration is:

$$\alpha\phi \frac{\delta s}{\delta t} + \nabla(\vec{v}_{\text{water}}) = 0 \qquad (1)$$

$$\nabla \cdot \vec{v} = 0 \qquad (2)$$

The saturation of the water phase is represented by s; $1 - s$ is thus the oil saturation. The thickness of the reservoir slab in the third dimension and the reservoir porosity are α and ϕ respectively, and are held constant. Equation (2) describes the divergence free character of the total volumetric flow field $\vec{v} = \vec{v}_{\text{water}} + \vec{v}_{oil}$. The water phase velocity $\vec{v}_{\text{water}}$ is related to the total velocity by

$$\vec{v}_{\text{water}} = f(s)\vec{v} + \overleftrightarrow{K} f(s)\, \lambda_{\text{oil}} (\varrho_{\text{water}} - \varrho_{\text{oil}})g \qquad (3)$$

where ϱ is the relevant fluid density and g the gravitational acceleration. The total fluid velocity is specified by Darcy's law, which, neglecting surface tension effects between the two fluids, is

$$\vec{v} = -(\lambda_{\text{water}} + \lambda_{\text{oil}})\overleftrightarrow{K} \cdot \nabla P - (\lambda_{\text{water}}\varrho_{\text{water}} + \lambda_{\text{oil}}\varrho_{\text{oil}})g\overleftrightarrow{K} \cdot \hat{z} \qquad (4)$$

Equations (2) and (4) thus determine an elliptic equation for the pressure field, which is coupled to the hyperbolic equations (1) through the water saturation.

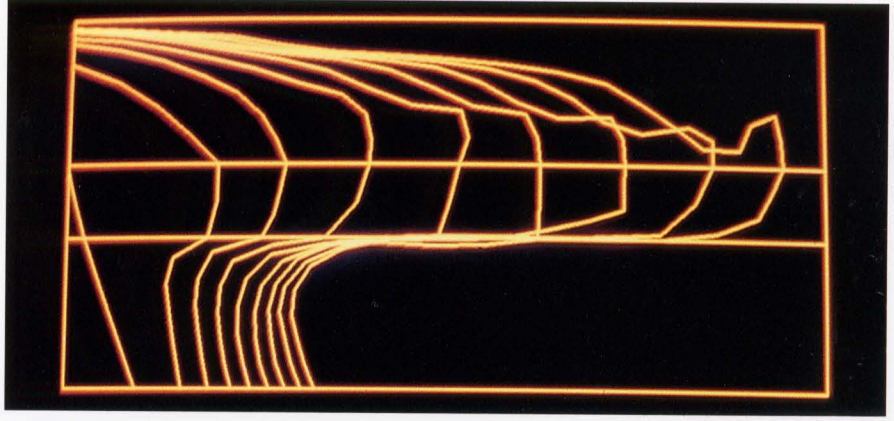
The rock permeability tensor $\overleftrightarrow{K}$ is a 2-by-2 tensor describing the spatial variation of the rock pore structure and hence its permeability to flow. λ_i is the relative transmissibility of the rock to the particular fluid i, in the presence of the other fluid. In a region where both fluids exist, the immiscibility of the two phases with respect to each other implies that the presence of the water phase in the pore channels nonlinearly reduces the ability of the oil phase to flow, and vice versa. The forms of λ_{water} and λ_{oil} used in this computation were determined from experimental data.

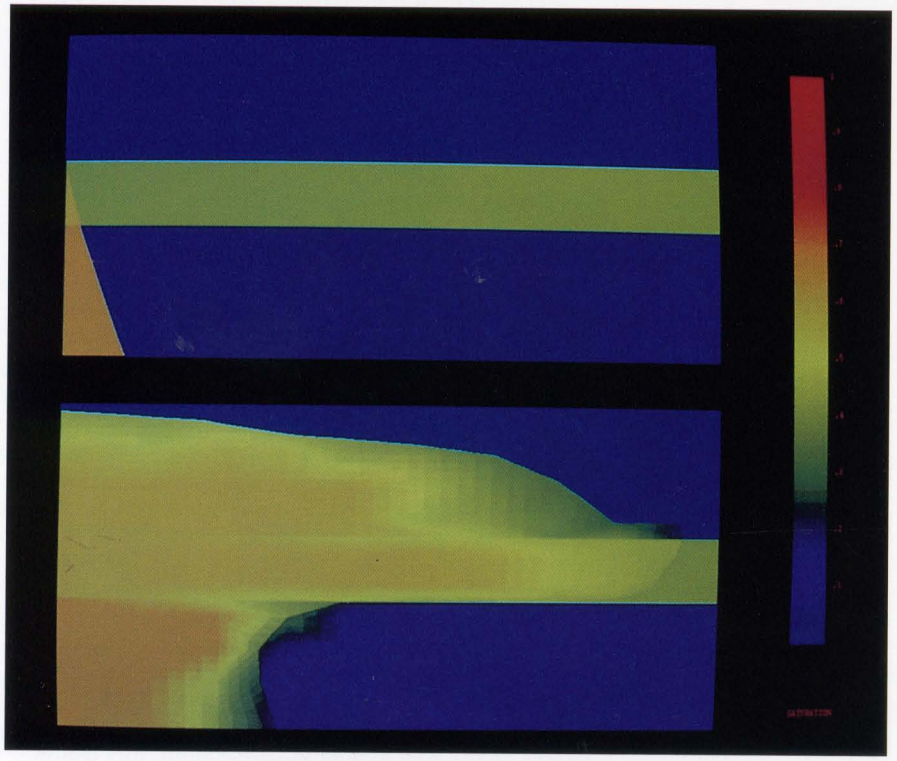
The fractional flow function, which relates the water phase velocity to the total phase velocity in the absence of gravity, is given by a ratio of fluid transmissibilities,

$$f(s) = \frac{\lambda_{\text{water}}}{\lambda_{\text{water}} + \lambda_{\text{oil}}} \qquad (5)$$

Conclusion

Figure 1 (following page) displays the tracked interface, which separates the unflooded oil phase from the water-swept region. The interfaces at one-fourth-year intervals for two years total simulation time are superimposed on each other, along with curves indicating the boundaries between the rock layers. For clarity, the reservoir vertical section is not drawn to scale, nor is it shown tilted at the 10° of the





calculation. Thus, in reality the initial oil-water interface at time 0 is horizontal.

Figure 2 displays the water saturation as a function of position in the reservoir at the initial time using a color scale for saturation values. The initial aquifer position is shown. In addition, the middle layer contains a large fraction (40 percent) of immobile, connate water. Small immobile fractions (less than 8 percent) of water also exist in the other two layers. Additionally, a given volume fraction of the oil (35 percent in the bottom layer, 40 percent in the other two) is irretrievable and bound to the rock; hence the initial water saturation in the aquifer is 65 percent, of which 59 percent is mobile and 6 percent is connate. Figure 2 displays the water saturation at the final time of two years.

Several physical phenomena are observable from Figure 2. First, the interface is seen plainly as a sharp discontinuous change in the water saturation. (In the middle layer, the color differential across the interface is less pronounced, though still visible about four-fifths of the way along the layer.) The size of the jump in the water saturation changes abruptly as the interface travels from one layer into the next, while the size of the jump changes smoothly along the interface within each layer due to gravitational effects. This interface would not be resolved cleanly by a conventional simulator; the discontinuous change in water saturation would be smoothed out over a distance, and the water-swept region would appear to be more advanced than shown by the front-tracking method.

Figure 1. Injection front evolution after two years.

Figure 2. (Top) initial reservoir conditions, (bottom) water saturation distribution after two years.

The displacement process has channeled largely into the more permeable top two layers. The bottom edge of the channel is plainly coincident with the boundary between the middle and lower layers over at least half of its length. A conventional simulator would show that the channel extended farther into the bottom layer.

Because of the density differences between the fluids, the water is prevented from channeling to the top of the reservoir by a lighter layer of oil, which rides on top. This gravity override is an important consideration in the design of vertical displacement strategies.[3] Again, while a conventional simulator would show some override, the computation would not be as clean, and the water-swept region would appear more advanced along the upper boundary.

Finally we point attention to the high resolution obtained from the front-tracking procedure. This calculation was performed using a tracked interface to follow the fluid boundary, two tracked, but unmoving curves to locate the layer edges, and a 10-by-20 rectangular grid to compute the solution away from the interface. The pressure solution was computed on a finite element mesh of 20-by-40 nodes. A total of 73 time steps were taken to run this problem for a total CPU time of 460 seconds. ▬

About the authors

W. Brent Lindquist is an associate professor in the Applied Mathematics and Statistics Department at the State University of New York at Stony Brook. He previously was a research associate professor at the Courant Institute of Mathematical Sciences in New York City, where he spent seven and a half years. He received a Ph.D. degree in theoretical physics from Cornell University in New York in 1981.

Qiang Zhang is an associate research scientist at the Courant Institute of Mathematical Sciences. He received a Ph.D. degree in physics from New York University in 1986.

Yogeshwar Sharma is a petroleum industry specialist for Cray Research in Dallas, Texas. Before joining Cray Research, he was a research scientist for Schlumberger Doll Research. He received a bachelor's degree in mechanical engineering from the University of Alberta in 1973, and completed postgraduate work in chemical engineering at the University of Calgary in 1977.

References

1. King, M. J., W. B. Lindquist, L. Renya, "Stability of Two-dimensional Immiscible Flow to Viscous Fingering," DOE/ER/03077-244, March 1985.
2. Glimm, J., J. Grove, W. B. Lindquist, O. A. McBryan, and G. Tryggvason, "The Bifurcation of Tracked Scalar Waves," SIAM, Vol. 9, Number 1, January 1988.
3. Fayers, F. J. and T. J. M. Newly, B. P. Research, "Detailed Evaluation of an Empirical Model for Viscous Fingering with Gravity Effects," *Proceedings of the Ninth Annual SPE Symposium on Reservoir Simulation*, February 18, 1987.

Oil field modeling in Saudi Arabia

William M. Brummett, ARAMCO
Dhahran, Saudi Arabia

Saudi Arabia's state oil company, Saudi ARAMCO, is the world's largest oil company. It manages the kingdom's hydrocarbon resources, which are produced from the world's largest oil fields. To ensure that these vast resources are developed and exploited efficiently, ARAMCO's petroleum engineers rely heavily on numerical reservoir and surface facility simulation models. These simulations currently run on a CRAY-1M/4400 system at the company's EXPEC Computer Center in Dhahran, which is devoted exclusively to exploration and petroleum engineering computing.

Cray system marks three firsts

In late 1984, ARAMCO installed the Cray system, marking several firsts in the field of reservoir simulation. The system was the first supercomputer installed in the Middle East, ARAMCO's first venture into supercomputing, and the first supercomputer devoted exclusively to reservoir and oil field facility simulation.

ARAMCO chose the CRAY-1M system over other available Cray machines for several reasons. First, the CRAY-1M's MOS technology was deemed to be more reliable at that time. (This has proved to be a good assumption as the machine has accumulated one of Cray's best reliability records, completing 1988 with 99.7 percent availability for the full year.) Second, during the initial period of model transfer and in-kingdom model construction, the CRAY-1M would provide sufficient computing power. Third, because this system would be the first supercomputer obtained by a Middle Eastern country, Cray Research and ARAMCO believed that obtaining the U.S. export license for a CRAY-1M system would be easier than obtaining an export license for a larger Cray system.

The system originally was configured with four Mwords of main memory, four I/O processors, four Mwords of buffer memory, and 16 DD-29 disk drives. It was upgraded to eight Mwords of buffer memory and 32 DD-29s in 1986. The system is front-ended by a Cray MVS Station running on an IBM 3090-400E with two Vector Facilities. The installation runs under version 1.16 of Cray Research's COS operating system.

Building a comprehensive dynamic model

The function of the branch of petroleum engineering known as reservoir engineering is to estimate the recoverable reserves from a reservoir, design the most efficient withdrawal strategy to optimize recovery, decide where and when to drill wells and install surface facilities, and predict the rate of production. This requires a clear understanding of reservoir energy distribution and fluid movements under any given operational scenario or market-demand situation. No other single technology has the ability to provide this insight as well as modern reservoir and facility simulation, because it brings all pertinent geological and engineering data together into a single, comprehensive, dynamic model.

Oil and gas typically are found in the pore spaces of rock formations, such as sandstone or limestone, thousands of feet below the earth's surface. The hydrocarbons reside at a high structural point in the formation (the reservoir) and normally are underlain and surrounded by water (the aquifer). The ability of fluids to flow through the reservoir and aquifer rock is governed by the rock permeability. This is a measure of the ability of the rock to permit flow of fluids through the interconnections of the pore spaces.

Reservoir simulation

Reservoir simulators are among the most complicated and difficult computer applications ever devised. They model fluid flow and pressure behavior over time with complex partial differential equations that simulate fluid flow in porous media. These take the form of parabolic pressure equations derived from Darcy's Law and conservation principles, and hyperbolic fluid saturation equations derived from mass balance relationships. Mass transfer between fluids is described by an equation of state. Where mass transfer is limited, these functions are fairly simple. Reservoir simulators in this category are called black oil simulators, and are the most commonly used type.

The equations must be solved numerically, usually by finite difference techniques. This requires spatial discretization of the reservoir into a grid system. This can be in one, two, or three space dimensions. Typically two-dimensional models describe a single layer divided into an areal grid. Three-dimensional models further divide the reservoir into multiple vertical layers. Each grid block or cell describes geologic, fluid, and energy parameters. Some parameters remain static (porosity and permeability) and some change with time (pressure and fluid saturations). A cell may contain a well that can either withdraw fluid (a producer) or inject fluid (an injector). Fluids may move freely between cells, depending upon permeability relationships and pressure gradients.

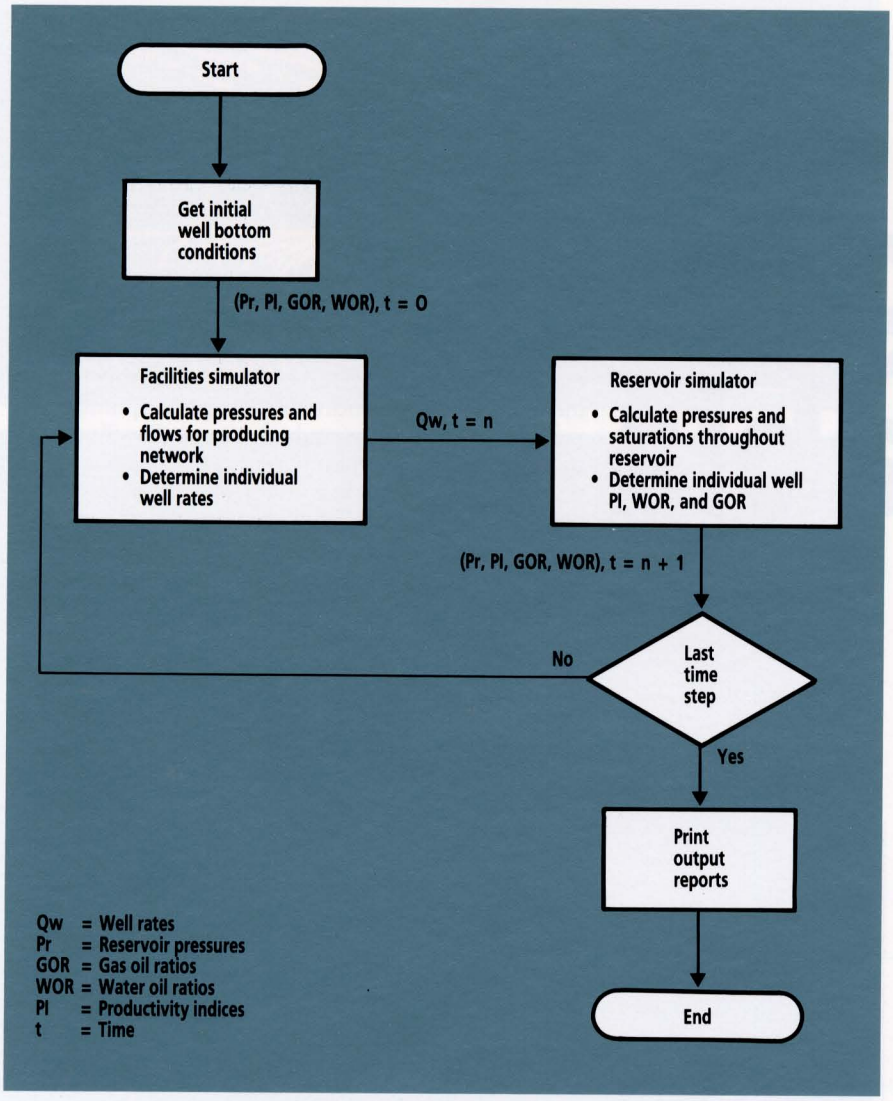


Figure 1. Flow diagram for a typical reservoir/facilities simulation program.

It is important that the cell sizes are small enough to describe adequately the variations in data. If the grid is too coarse, the parameter averaging necessary to assign single values can give erroneous results. Grids that are too coarse also can impact the mathematical solution due to excessive truncation error. For fieldwide models, the ability to describe a grid with sufficient density often is limited by computer resources because these models may require tens of thousands of cells for minimally adequate definition. Fortunately, advances in supercomputer technology are removing such barriers rapidly.

Facility simulation

An important but often overlooked factor in oil field simulation is the effect of wellbore and surface piping systems on reservoir production performance. The producing facilities can have a significant and time varying effect on reservoir behavior due to changing piping system flow resistances caused by variations in fluid mixtures that may be flowing.

Facility simulators model the flow behavior of multiphase fluids in piping systems as functions of pressure and temperature. The mathematical relationships are based on classical energy balance concepts from fluid dynamics, modified to account for complex density, viscosity, and friction variations caused by the interaction of the multiple fluid phases flowing. The various piping components such as well tubing, surface flow lines, trunk lines, and loops form a network. Solution of the complex flow rate-pressure-temperature relationships through a single flow path, such as a single wellbore and flow line, are solved iteratively, either with finite element or finite difference techniques. Combinations of flow paths form the entire network. A global solution of the network is superimposed over the multiple single flow path calculation using various network solution algorithms.

When reservoir simulators are used in a stand-alone mode, certain assumptions must be made about producing wellbore pressures. If a facility simulator is linked dynamically to a reservoir simulator, the facility simulator can calculate the wellbore pressure and remove one of the more critical assumptions. Thus, a reservoir simulator and facility simulator, dynamically linked, form a complete oil field simulator that can provide more realistic forecasts. Figure 1 illustrates a typical linked reservoir-facility oil field simulator.

Simulator application

Normally an oil field simulation study comprises two distinct phases. In the first phase, history matching, the best available data are input and the wells are produced at known historical rates for the oil phase. Water and gas phases are calculated according to interphase and petrophysical relationships. Calculated rates and pressures are compared to known historical data. Certain input parameters, such as cell porosity and permeability, are adjusted iteratively by the engineer until all known and calculated data compare within achievable tolerances.

In the second phase, prediction, a theoretical production strategy is specified and the simulator is allowed to produce wells at rates determined by calculated productivity for all fluid phases. The production strategies can be very complex in sophisticated simulators with the simulator allowed to "drill" new wells and "recomplete" existing wells to meet or maintain specific production criteria. In this manner, various exploitation and facility construction schemes can be evaluated.

Oil field simulation at ARAMCO

ARAMCO historically has used reservoir simulation as the technology evolved. Although early models were beneficial, the sheer magnitude of Saudi Arabian reservoirs and computing limitations restricted engineers to very coarse grid sizes.

The emergence of supercomputers such as Cray systems provided a major breakthrough in reservoir modeling. For the first time, ARAMCO engineers could envision models that more faithfully represented physical systems. ARAMCO's engineers and geologists initiated new data-gathering and interpretation efforts aimed at providing more detailed descriptions of the company's major reservoirs. They developed large and comprehensive data bases to accumulate and manipulate the data to form the initial basis for a new generation of oil field models.

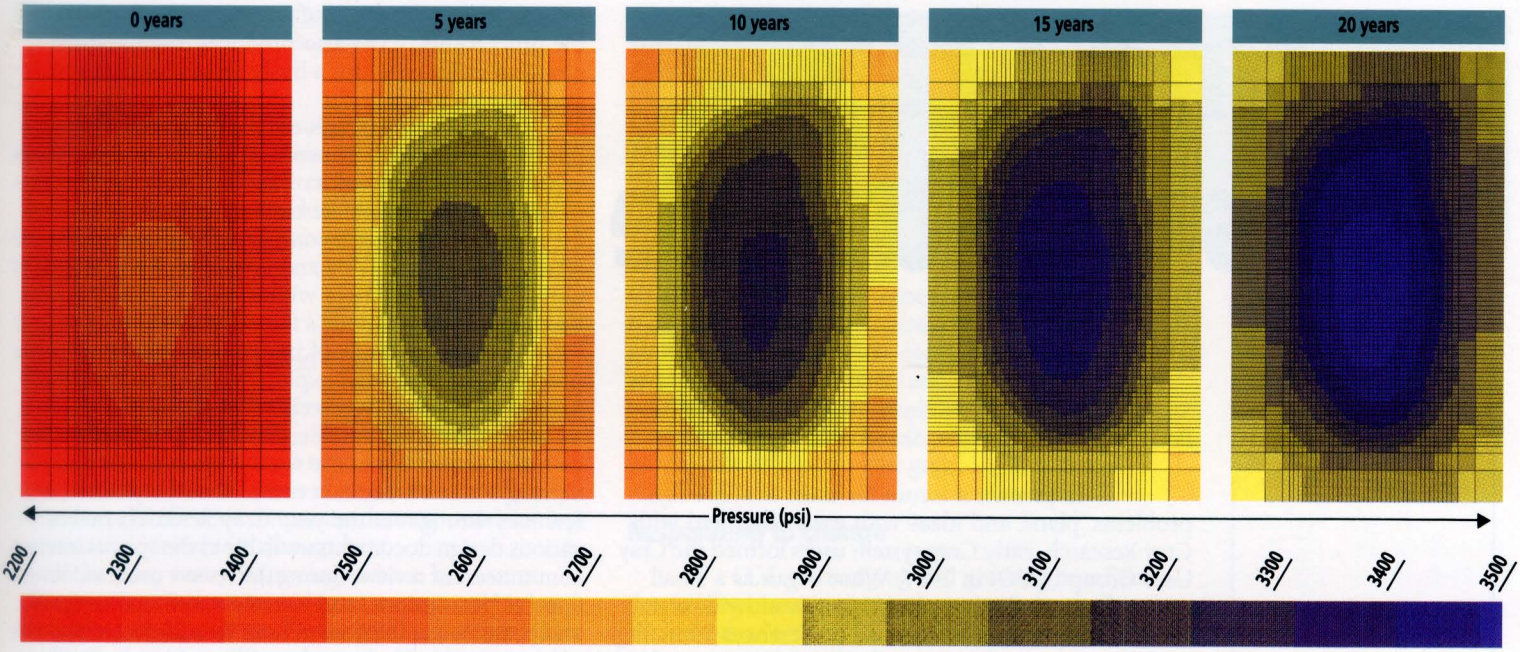


Figure 2. Snapshot examples of SIMPLOT displays.

Application software

ARAMCO uses two state-of-the-art reservoir simulators — both in "black oil" mode — and the industry standard multiphase network piping facility simulator. The MARS reservoir simulator is licensed from the Exxon Corporation. The CRS-3D reservoir simulator and the PIPEFLOW-2 facility simulator are licensed from the Chevron Corporation.

In addition to the standard support software that is part of these licensed packages, ARAMCO's ECC Simulation Systems Division has developed a wide variety of pre- and postsimulation support applications to provide a more engineer-friendly way to use the technology. The cornerstone of this software is SIMTRAN, an ARAMCO-developed framework that provides a simulator-independent user interface.

The key components of SIMTRAN are SIMEASE and SIMPLOT. SIMEASE, which is licensed from Scientific Software Intercomp, is an interactive graphics interface for reservoir simulators. SIMPLOT is an ARAMCO-developed graphics display system that uses special IBM image processing hardware to download large cell matrix displays (such as reservoir pressure) and display them in near movie quality animation as they change through time. Figure 2 shows snapshot examples of SIMPLOT displays. These represent the pressure behavior of a typical reservoir as oil is produced over a 20-year period. The snapshots were taken at five-year intervals. However, the actual SIMPLOT display that an engineer views at the image processing workstation shows all time steps (in this example, about 200 intervals) as a smooth animated sequence.

Workload

The sheer size of Saudi Arabian reservoirs indictates that the reservoir models will be tremendously large. ARAMCO model sizes currently range up to over 50,000 cells with even larger models currently under construction. Up to 25 models may be in active use at any time with many larger than 20,000 cells. An even larger number of producing facility models may be active at any time and several of these may be linked dynamically to reservoir models.

Managing such a workload requires careful attention to software efficiency and scheduling. A series of enhancements has been made to the simulation software to increase throughput. When the software first was installed in 1984, the average wallclock-to-CPU time ratio was about nine to one. Steady improvements were made, and in late 1988 the average was down to about two to one in spite of larger job sizes. A single thread scheduling scheme has proven to provide the maximum throughput since the job mix is essentially homogeneous. The system is currently running at near full seven-day, 24-hour capacity.

ARAMCO engineers and geologists would like to add more resolution to current models to take advantage of better basic data that have been accumulated. In addition, new models will be constructed for undeveloped reservoirs. This indicates that additional computer power will be needed. ECC's hardware planners are evaluating various upgrade strategies to provide the necessary supercomputing resources to address these needs. ▬

Acknowledgment

The author would like to express appreciation to the Saudi Arabian Ministry of Petroleum and Mineral Resources, to Saudi ARAMCO, and to Chevron Corporation for permission to publish this article.

About the author

Bill Brummett is Manager of the Petroleum Engineering Application Services Department in ARAMCO's EXPEC Computer Center. This department supports all of ARAMCO's simulation software. He has been on loan to ARAMCO from Chevron Corporation since 1982. He holds B.S. and M.S. degrees in petroleum engineering from Louisiana Tech University. His 27 years with Chevron have included a wide variety of assignments including drilling engineering, production engineering, reservoir engineering, operations research, and computer applications.

Uniting Cray users worldwide

Stephen Niver, Cray User Group president
Boeing Computer Services, Seattle, Washington

To provide a forum for users to share their problems, plans, and ideas with each other and with Cray Research, early Cray system users formed the Cray User Group (CUG) in 1977. What began as a small group of users today has grown into a worldwide organization that sponsors semiannual conferences attracting hundreds of attendees from the broad spectrum of Cray customers.

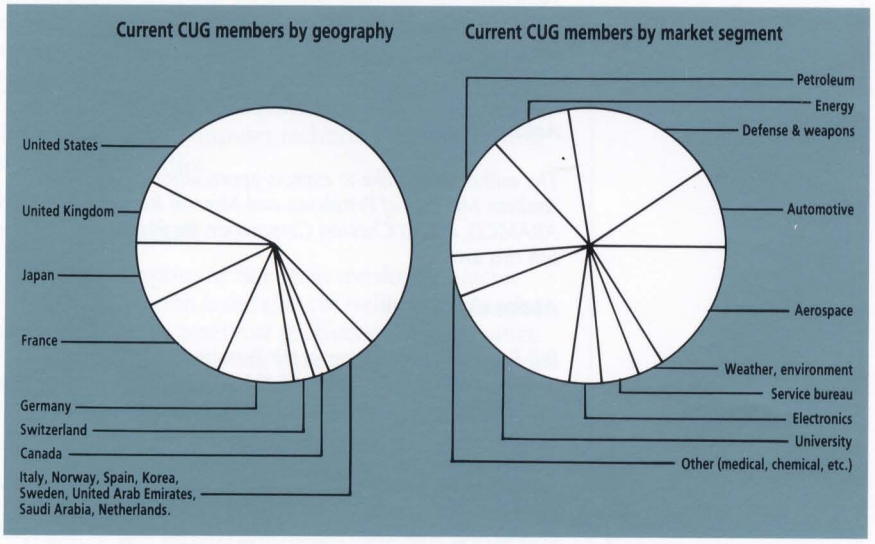
CUG today

Members of CUG own or lease at least one Cray supercomputer. Today, Cray users from diverse industries and organizations all over the world belong to the group (Figure 1). CUG's diverse membership means that a wide range of opinions and interests is represented.

The most visible aspect of CUG is its semiannual conferences that bring together all types of computer professionals including data center managers, system programmers, applications developers, and end users. Attending these conferences is an excellent way to meet other professionals in similar areas of expertise. Many personal and professional friendships have been started or renewed at the conferences.

The conference proceedings provide valuable information to those unable to attend. The spring 1988 CUG meeting produced a 500-page publication written by Cray users for Cray users. In addition to the proceedings, CUG also produces a semiannual newsletter that provides a lively forum for discussion and information.

Figure 1. CUG membership breakdown.



The conferences often provide the first opportunity to discuss new products. Often, CUG members can discuss a new product with its developer. Sessions dealing with new Cray Research products are among the most popular at each conference. The last conference featured talks on the Ada and Lisp compilers, as well as users' initial experiences with the CRAY Y-MP system. Conferences also provide a forum for discussion of new ideas, such as methods of implementing new software products. Sessions that explore UNICOS conversion from COS are also very well-attended.

From the outside, CUG might appear to be functioning only at the time of each conference, although much work is accomplished between conferences throughout the year. Cray Research makes various design documents available to the special interest committees for review; during the review process, design approaches, options, and alternatives are considered and presented at the conferences. In addition to reviews of near-term products such as file archiving, CUG members become involved in discussions and reviews of new products that may introduce significantly different or new capabilities or may pose problems. While Cray Research representatives and CUG members share technical discussions at the conferences, the CUG board of directors also meets with Cray Research executives at the conferences.

The organization

The Cray User Group has adopted a flexible policy to facilitate discussion and provide some degree of guidance and management. The board of directors is responsible for the administrative operation and general direction of CUG; directors are elected each year with staggered terms to minimize transitional problems. The special interest committees and mutual interest groups were formed to focus discussion on specific technical topics or on various topics of mutual interest. These committees and groups that meet and sponsor the majority of the talks given at each conference may be created or disbanded depending upon interest. CUG's special interest committees and mutual interest groups include:

- ☐ Applications/algorithms — end user applications/mathematics
- ☐ Operations — operational aspects of computer centers
- ☐ Software tools — compilers, multiprocessing tools, utilities, etc.
- ☐ Communications — front ends, gateways, networking, etc.
- ☐ Performance — software performance-related issues
- ☐ Graphics — graphics features, devices, connectivity
- ☐ Operating systems — COS, UNICOS operating systems
- ☐ CTSS — the Cray Time Sharing System
- ☐ Management — management issues related to computing

☐ Aerospace — issues related to aerospace business
☐ CRAY-2 system — issues related to CRAY-2 system sites

The other CUG subgroup, the advisory council, consists of the chairs of the special interest committees, the Cray Research executive representatives, the board of directors, the local arrangements chair for the current and next conferences, and several members-at-large. The advisory council "runs" each conference.

Technological change and CUG

Because of the constant state of technological and sociological change, CUG has attempted to be flexible enough to respond to the evolving needs of its membership. Several recent changes have had a major impact on CUG, the companies that its members represent, and its individual members.

In the past, many technologies have been semi-independent. Today technologies are not only related, but also interdependent. This interdependence adds to the confusion and complexity in developing and using technology. This means that Cray Research's customers can no longer view their supercomputer technology separately from other computational technologies. Cray Research recognizes this, and has been developing business relationships with companies that produce complementary products. CUG always has offered a broad perspective of options to its members by sponsoring presentations by companies that offer products that fit into the Cray supercomputing environment. While participation from companies other than Cray Research will continue to increase, CUG's challenge is to insure that our primary interest in Cray Research products and services is not compromised.

In an organization with a membership as diverse as that of CUG, the relationships among companies and with Cray Research have changed. A number of companies have joint agreements with each other, some companies are customers of others, and some have joint agreements with Cray Research. These new relationships and arrangements are complementary in technology, yet competitive in business. CUG must continue to provide a useful forum for all of these companies, while respecting their complex and sometimes conflicting relationships.

While the pace of technological change is frequently breathtaking, the rate at which individuals and organizations absorb and react to these changes is much slower. For example, consider the many technologies that are changing the fundamental focus in the computing environment to a network focus, where the traditional data center represents only one node in a computational network. In this particular case, the technologies are providing networking capabilities more quickly than people in the organizations can understand how their roles and missions are changing. At CUG, the end user, applications developer, system programmer, data center manager, and Cray Research developer can discuss their roles and enhance the resulting computing environment.

One method of systematizing and managing the many technologies is through development of standards that describe how functions behave and interact. Several areas are especially important at this time, because their evolution will play a significant role in our computing environments in the future: window interfaces (icon standardization for point and click computing), new network protocols for higher-speed communication, Fortran bindings (as in the POSIX proposals), and math and graphics libraries.

Date	Location	Sponsor
April 24-28, 1989	Los Angeles, California	Lockheed Advanced Aeronautics Company
September 18-22, 1989	Trondheim, Norway	University of Trondheim
October 1-5, 1990	Austin, Texas	University of Texas
Spring 1991	London, England	University of London
Fall 1991	Sante Fe, New Mexico	Los Alamos National Laboratory

Figure 2. Upcoming CUG conferences.

Responding to change

CUG is changing and growing to reflect the developments in technologies and the marketplace. We have made some mechanical changes to the conference program to facilitate attendance at parallel sessions. We also have added flexibility to the length of talks to allow for added emphasis on certain topics.

Tutorials introduce many new subjects to attendees and provide an opportunity for detailed discussions on very specific topics. CUG also created a management special interest committee to provide a forum for managers to discuss the many technical and nontechnical management issues in supercomputing. Today, several CUG members participate in various standards activities. We are considering methods of disseminating standards information and how CUG should become involved in the various standards/open software efforts.

Many of the challenges ahead are nontechnical. CUG officers and Cray Research executives have established a closer relationship, and now meet once a year outside the conferences. This dialogue will benefit both Cray Research and CUG in better understanding future directions and how both organizations can work together.

The Cray User Group is involved in many exciting issues aside from the normal technical discussions relating to Cray Research hardware and software products. For each conference, we attempt to select a timely theme. "The Challenge of Change" is the theme for the next conference, which will be held in Los Angeles in late April. This theme is especially appropriate for CUG today, both for individual members and for the organization as a whole.

As president of CUG, I hope to see interest in the organization continue to grow. Participation in the Cray User Group is the best opportunity for addressing technical needs and for further integrating supercomputing into the mainstream of computing. ▬

CUG is changing to reflect developments in technologies and the marketplace.

About the author

Stephen Niver, CUG president, is the manager of computing technology for the Commercial Engineering and Scientific Group at Boeing Computer Services. He earned a bachelor's degree in electrical engineering from the University of South Carolina in 1966.

Basic linear algebra subprogram optimization on the CRAY-2 system

Qasim Sheikh and Jong Liu
Mathematical Software Group, Cray Research, Inc.

By effectively using the CRAY-2 system's main and local memories, Cray Research's Mathematical Software Group has optimized Level 3 basic linear algebra subprograms (BLAS) for complex data. The memory management scheme described throughout this article minimizes degradation in performance due to even-strides in memory access for complex data.

Level 1, Level 2, and Level 3 BLAS define subroutines to perform basic vector-vector, matrix-vector, and matrix-matrix operations, respectively.[1,2,3,4] Level 1 BLAS kernels have been used successfully in a wide range of software, including LINPACK, EISPACK, the ACM algorithms, and more specialized application packages. However, to approach the potential performance of vector supercomputers such as the CRAY Y-MP, CRAY X-MP, and CRAY-2 systems, one needs to optimize at least on the matrix-vector operations level. Level 2 BLAS were defined in an attempt to address this issue on vector processors. Unfortunately, using Level 2 BLAS often is not the best approach for computers with a memory hierarchy (such as global memory, cache or local memory, and vector registers), or parallel processing computers, in which excessive movement of data to and from main memory should be avoided. To address this problem, Level 3 BLAS, targeted at matrix-matrix operations, were proposed.

Scope of Level 3 BLAS

In brief, the Level 3 BLAS perform five basic types of matrix-matrix operations, as shown in Table 1. (More detailed description of Level 3 BLAS scope, naming conventions, and storage conventions can be found in the BLAS 3 proposal that was published in Argonne National Laboratory's Technical Memorandum 88.[4])

The matrices can be general, symmetric, Hermitian, or triangular (where appropriate). Table 2 summarizes the names of the routines available in our implementation of Level 3 BLAS.

Matrix-matrix products
$C \leftarrow \alpha AB + \beta C$
$C \leftarrow \alpha A^H B + \beta C$
$C \leftarrow \alpha AB^H + \beta C$
$C \leftarrow \alpha A^H B^H + \beta C$
where C is an m-by-n matrix. If A is not transposed it is an m-by-k matrix; otherwise it is an n-by-k matrix. Similarly, if B is in not transposed it is a k-by-n matrix; otherwise it is an n-by-k matrix. Also, A and B can be symmetric or Hermitian matrices.

Rank-k updates of matrices
$C \leftarrow \alpha AA^T + \beta C$ (C symmetric)
$C \leftarrow \alpha A^T A + \beta C$ (C symmetric)
$C \leftarrow \alpha AA^H + \beta C$ (C Hermitian)
$C \leftarrow \alpha A^H A + \beta C$ (C Hermitian)
where C is an m-by-m symmetric or complex Hermitian matrix and A is an m-by-n or n-by-m matrix.

Rank-$2k$ updates of matrices
$C \leftarrow \alpha AB^T + \alpha BA^T + \beta C$ (C symmetric)
$C \leftarrow \alpha A^T B + \alpha B^T A + \beta C$ (C symmetric)
$C \leftarrow \alpha AB^H + \bar{\alpha} BA^H + \beta C$ (C Hermitian)
$C \leftarrow \alpha A^H B + \bar{\alpha} B^H A + \beta C$ (C Hermitian)
where C is an m-by-m Hermitian matrix, A and B are m-by-k or k-by-m matrices.

Multiply a matrix by a triangular matrix
$B \leftarrow \alpha TB$, $B \leftarrow \alpha T^T B$
$B \leftarrow \alpha BT$, $B \leftarrow \alpha BT^T$

Solve a triangular system with multiple *rhs*
$B \leftarrow \alpha\ T^{-1}B$, $B \leftarrow \alpha(T^T)^{-1}B$
$B \leftarrow \alpha BT^{-1}$, $B \leftarrow \alpha B\ (T^T)^{-1}$
where B is an m-by-n matrix and T is an n-by-n or m-by-n matrix.

Table 1. Five basic types of matrix-matrix operations.

Level 3 BLAS

Complex	Real	MM	RK	R2K	SM
CGE	SGE	*			
CSY	SSY	*	*	*	
CHE		*	*	*	
CTR	STR	*			*

Table 2. Routines available in Cray implementation of Level 3 BLAS.

Implementation strategies

Cray Research's Mathematical Software Group has provided optimized Cray Assembly Language implementation for 40 different routines in the set of Level 2 and Level 3 BLAS. Taking into account the programming differences between the CRAY Y-MP, CRAY X-MP, and CRAY-2 system architectures, we actually have optimized 80 different subroutines. If each subroutine had been optimized in isolation, this project would have required great effort. Through careful analysis we identified a small set of functions that are common to the different BLAS 2 and 3 subroutines. These common functions then were implemented as CAL macros and were used as building blocks in implementing optimized subroutines.[5] Following are examples of some of our BLAS 2 macros:

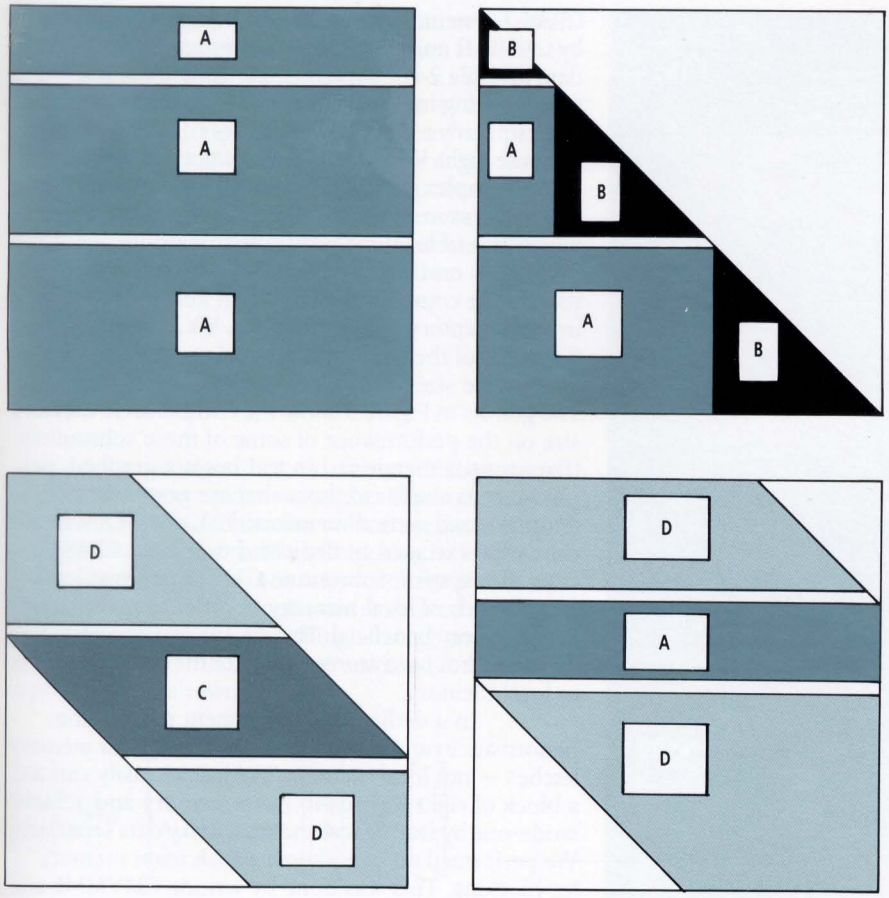


Figure 1. Four types of matrices split into blocks to use BLAS-2 macros. *A* represents macros for rectangular matrices, *B* represents macros for triangular matrices, *C* represents macros for parallelogram matrices, and *D* represents macros for trapezoidal matrices.

Multiply an m-by-n, $m \leq 64$, rectangular matrix by a vector, adding the result to the contents of a known vector register.

Multiply an n-by-n, $n \leq 64$, triangular matrix by a vector, adding the result to the contents of a known vector register.

Multiply an m-by-n, $m \leq 64$, trapezoidal matrix by a vector, adding the result to the contents of a known vector register.

Multiply an m-by-n, $m \leq 64$, parallelogram matrix by a vector, adding the result to the contents of a known vector register.

Figure 1 shows four types of matrices and how they can be split into blocks to use the macros defined above, where *A*, *B*, *C*, and *D* represent the macros for rectangular, triangular, parallelogram, and trapezoidal matrices, respectively.

Using these macros yields several advantages, including correctness by construction and performance portability across subroutines. Note that we are not sacrificing any performance by using macros because they are basically treated as text substitution at subroutine assembly time. Also, these macros were designed to prefetch data for the following macro in the calling subroutine. These macros also allow software developers to optimize these subroutines with ease for new systems designed by Cray Research. Our hope is that only the macros will need reoptimization.

Macros described above exploit the main memory and vector registers as a memory hierarchy. On the CRAY-2-type architecture, it is important to exploit local memory as well. For this reason, our definition of BLAS 3 macros for complex data was extended. Following are examples of some of our BLAS 3 macros that exploit local memory:

CMVLM1 fetches an m-by-n, $n \leq 64$, $m \leq 64$, matrix A from main memory, performs $y \leftarrow y + Ax$, where x and y are vectors of sizes n and m, respectively, and stores A in local memory.

CMVLM2 performs the same function as CMVLM1 does, except that matrix A is fetched from local memory.

The macro CGEMMM performs $C \leftarrow \alpha AB + \beta C$, where A, B, and C are m-by-k, k-by-n, and m-by-n matrices, respectively, also $m \leq 64$ and $k \leq 64$. This is done by multiplying each column of B by matrix A and adding it to the corresponding column of C.

CMVLM1 is used to multiply the first column of B by A. In this process, A is fetched from main memory and stored in local memory. CMVLM2 is used for multiplying the rest of the columns of B by A and A is fetched from local memory.

CMVLM1 performs calculations that are memory-bound with a memory stride of two. This leads to significant performance degradation in CRAY-2 systems. However, when matrix A is stored in local memory, we separate the real and imaginary parts. This leads to very efficient stride-one fetches from local memory. Calculations performed by CMVLM2 are vector-register-bound. We use "loop unrolling" and "register rotation" to transform this calculation to a functional-unit-bound calculation and achieve performance close to the peak performance of the CRAY-2 system.

All complex routines in Table 1 are built using CMVLM1, CMVLM2, CGEMMM, and other macros that use similar strategy to exploit local memory. However, if we do not wish to use local memory we can use macros from Level 2 BLAS to construct Level 3 BLAS subroutines. Figure 2 compares the performance of CHEMM using straightforward Fortran Level 2 BLAS macros (local memory not used), and Level 3 BLAS macros (eight Kwords of local memory used).

Performance vs local memory size

The implementation of CHEMM using local memory runs at speed near peak performance of the

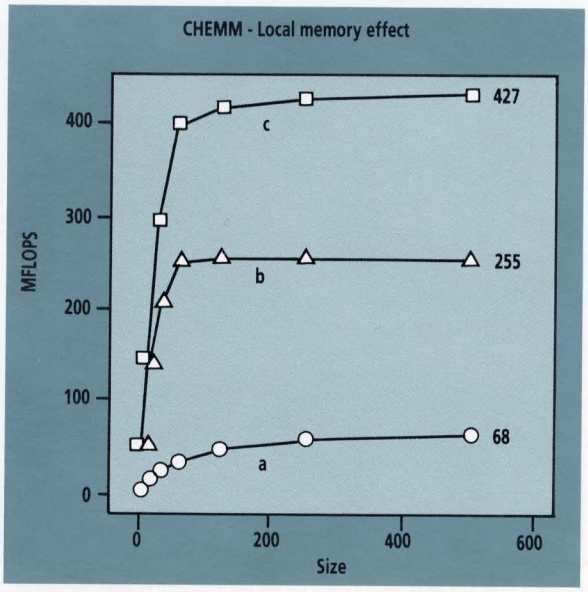


Figure 2. Performance comparison of CHEMM: a) using straightforward Fortran, b) using Level 2 BLAS macros (local memory not used), and c) Level 3 BLAS macros (8 Kwords of local memory used).

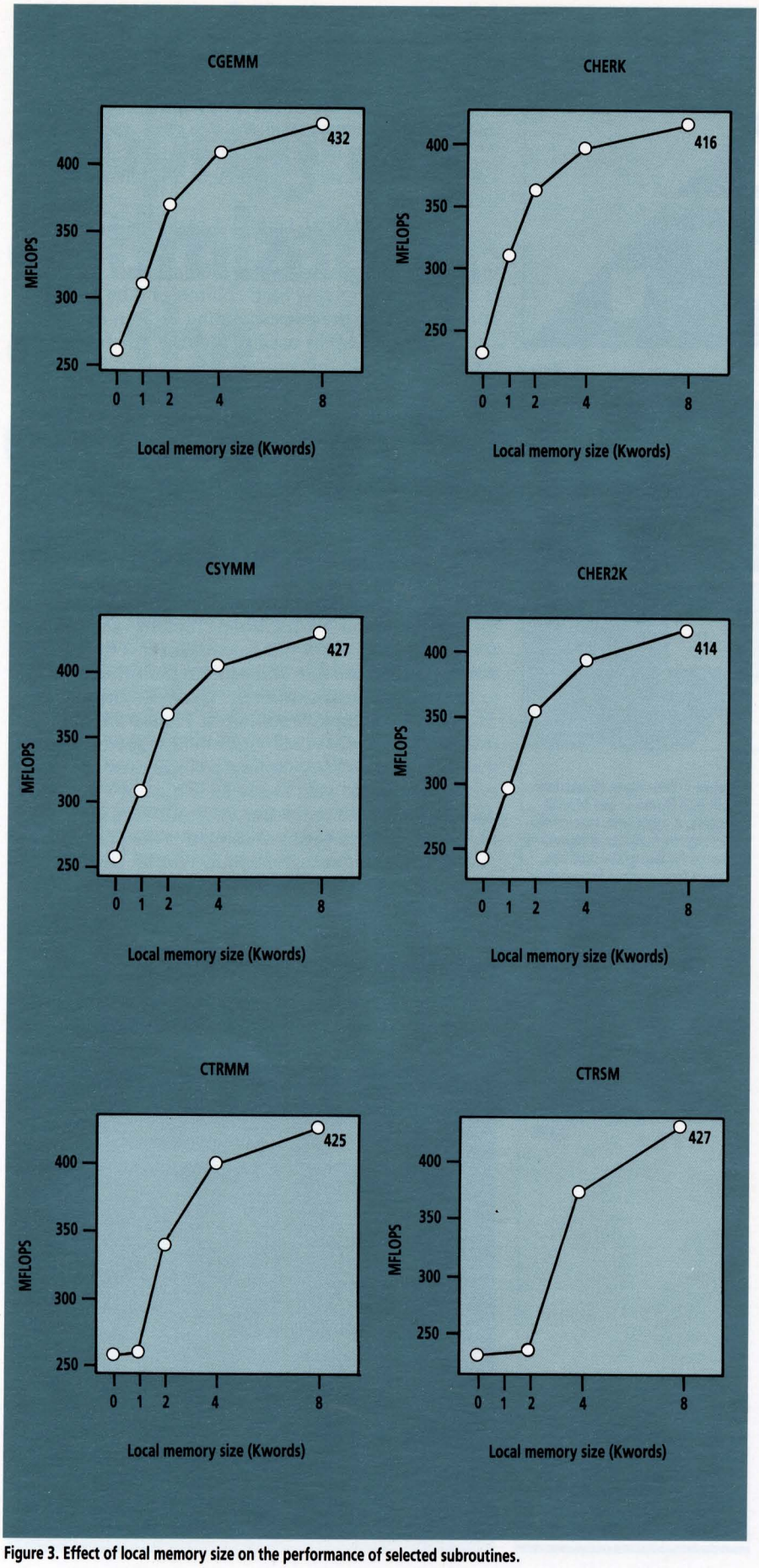


Figure 3. Effect of local memory size on the performance of selected subroutines.

CRAY-2 system. Most of the computation is performed by CMVLM1 and CMVLM2. All subroutines for complex data in Table 2 are implemented using these and other macros using local memory, and the subroutines perform similarly to the performance of CHEMM. However, they use eight Kwords of local memory to store a 64-by-64 complex matrix. The size of local memory on a CRAY-2 system is 16 Kwords. Situations may occur in which size of local memory available for Level 3 BLAS routines is much less than eight Kwords, since the user or the compiler may find that keeping some data in local memory is advantageous. We studied the performance of the subroutines mentioned above by varying the size of the block stored in local memory. The graphs in Figure 3 show the effect of local memory size on the performance of some of these subroutines (For routines mentioned in text but not graphed, performance is similar to those that are graphed). All matrices used were dimensioned 512-by-512, and all runs were executed in dedicated time on a CRAY-2S computer system. Subroutine CTRSM needs at least four Kwords of local memory in order to make using local memory beneficial. This exception occurs because the algorithm used stores a copy of the solution matrix in local memory.

In a dedicated environment most of the performance can be attributed to stride-one for memory fetches — not local memory. We just as easily can use a block of eight Kwords in main memory and achieve stride-one by storing real and imaginary data separately. We performed an experiment to use main memory for blocking. This was done by writing CMVMM1 and CMVMM2 and other macros that use main memory instead of local memory. CMVMM1 uses a temporary work space in main memory to store matrix *A*, so that real and imaginary parts of columns of A can be accessed with a memory stride equal to one. CMVMM2 fetches matrix A from this temporary storage in main memory. As expected, we achieve performance very similar to the performance achieved by using local memory. However in a nondedicated and parallel processing environment, using local memory pays off.

These experiments led us to choose our blocking strategy dynamically. At load time we checked the total amount of local memory used by all other subroutines. If the size of local memory available was less than one Kword, we used eight Kwords of main memory. The cost is a waste of eight Kwords of main memory allocated in stack mode. This is done without any user intervention. If desired, the user can easily learn the block size and the kind of memory used.

Parallel processing and local memory use

To exploit the potential of the CRAY-2 system, we have implemented a version of CGEMM that uses all four procesors and uses both local and main memory for blocking. The basic idea is to partition matrices *A*, *B*, and *C* as blocked matrices. Each block of a matrix is a matrix itself. We perform $A * B$ by multiplying blocks of *A* with corresponding blocks of *B*. Let *A*, *B*, and *C* be *m*-by-*k*, *k*-by-*n*, and *m*-by-*n* rectangular matrices, respectively. Also, let n_c be the number of complex columns of size 64 that can be stored in the local memory of each processor, let n_p be the maximum number of processors in the system, and n_b =

k/n_c. We assume, for simplicity, that k and m are divisible by n_c *and* n_p, respectively. We partition A in blocks A_{ij}, where each block is of size m/n_p-by-n_c. Similarly, we partition matrices B and C in blocks of sizes n_c-by-n and m/n_p-by-n, respectively. If more than one processor accesses the same block of a matrix at the same time, memory bank conflicts may occur. In this case, only one processor will proceed and the rest must wait. The following algorithm is used to distribute memory access by processors as evenly as possible.

For $j = 1$ to n_b do
  For $i = 1$ to n_p do
    $C_i \leftarrow \alpha A_{i,\,(i+j-1) \bmod n_b} B_{\,(i+j-1) \bmod n_b} + \beta\, C_i$
  if $j = 1$ then $\beta \leftarrow 1$

In this algorithm the choice of block size is determined by the maximum number of processors available in the system and the size of local memory. Graphs in Figure 4 show the results of blocking in local and main memories. These runs were made for 256-by-256 matrices on the CRAY-2 system in dedicated time using all four processors. The blocking strategy and the sequence in which the blocks are accessed for the above experiment are shown below.

Let

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix} \leftarrow \alpha \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} + \beta \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix},$$

where $n_p = 4$, $n_c = 64$, A_{ij} are 64-by-64 matrices, and B_i and C_i are 64-by-256 matrices. The first iteration of the outer loop performs $C_1 \leftarrow \alpha A_{11} B_1 + \beta C_1$, $C_2 \leftarrow \alpha A_{22} B_2 + \beta C_2$, $C_3 \leftarrow \alpha A_{33} B_3 + \beta C_3$, and $C_4 \leftarrow \alpha A_{44} B_4 + \beta C_4$. The second iteration of the outer loop performs $C_1 \leftarrow \alpha A_{12} B_2 + C_1$, $C_2 \leftarrow \alpha A_{23} B_3 + C_2$, $C_3 \leftarrow \alpha A_{34} B_4 + C_3$, and $C_4 \leftarrow \alpha A_{41} B_1 + C_4$, and so on. Note that in a dedicated environment no significant difference in speedup exists for local or main memory blocking. However, in a nondedicated and parallel-processing environment, blocking using local memory will yield better results than could be achieved using main memory for blocking.

Effects of improvements in CRAY-2 architecture

Newer CRAY-2 models will have improved memory modules so that memory fetches with stride two will not be problematic. This should improve the performance of CMVLM1. Also, by exploiting "tailgating," a feature planned for implementation in future CRAY-2 systems, Cray Research's software developers will not have to use "loop unrolling" and "register rotation" in CMVLM2. Clearly, as systems with 16 and 64 processors are designed, use of local memory for blocking will become more important.

This paper presents results from an extensive study of local memory use for Level 3 BLAS for complex data. Local memory block size is selected dynamically and users are freed from the responsibility of managing local memory. It is important not to load subroutines that statically reserve local memory but are not called. Loading such subroutines can force our routines to use main memory for blocking.

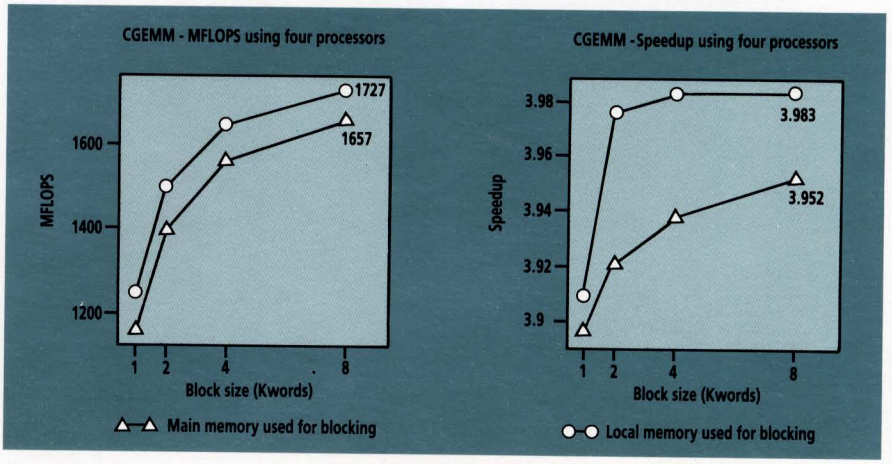


Figure 4. Results of blocking in main and local memories.

An area for future exploration is the use of local memory for real data. Studies have been conducted to solve systems of linear equations for such data.[6] For a real case it may be possible to store blocks of temporary results in local memory. ■

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

About the authors

Qasim Sheikh and Jong Liu are senior programmers in Cray Research's Mathematical Software Group. Before joining Cray Research in 1986, Sheikh earned a master's degree in mathematics from Punjab University in Lahore, Pakistan, and master's and Ph.D. degrees in computer science from the University of Illinois, Urbana-Champaign. Liu joined Cray Research in 1986 after teaching mathematics and computer science at Coker College in Hartsville, South Carolina, and the University of Central Florida in Orlando. He earned a master's degree in mathematics from Taiwan University, and a Ph.D. degree in applied mathematics from the University of Cincinnati in Ohio.

References

1. Lawson, C., R. Hanson, D. Kincaid, and F. Krogh, "Basic Linear Algebra Subprograms for Fortran Usage," *ACM Transactions on Mathematical Software 5*, 1979.
2. Dongarra, J. J., J. Du Croz, S. Hammarling, and R. J. Hanson, "An Extended Set of Fortran Basic Linear Algebra Subprograms," *ACM Transactions on Mathematical Software 14*, 1988.
3. Dongarra, J.J., J. Du Croz, S. Hammarling, and R.J. Hanson, Algorithm 656: "An Extended Set of Fortran Basic Linear Algebra Subprograms: model implementation and test programs," *ACM Transactions on Mathematical Software 14*, 1988.
4. Dongarra, J.J., I. Duff, J. Du Croz, and S. Hammarling, "A Set of Level 3 Basic Linear Algebra Subprograms," Technical Memorandum 88, Argonne National Laboratory, May 1988.
5. Vu, P., J. Liu, Q. Sheikh, and C. Yang, "Implementation of BLAS 2 & 3 Kernels and their Applications in a New Linear Algebra Library," *Proceedings of the Cray User Group 1988 Fall Meeting.*
6. Calahan, D. A., "Block-oriented, Local Memory Linear Equation Solution on the CRAY-2," Technical Report, Supercomputer Algorithm Research Laboratory, Department of Electrical Engineering & Computer Science, University of Michigan.

CORPORATE REGISTER

Cray Research's customers span the globe

Lawrence Livermore National Laboratory (LLNL) installed a CRAY Y-MP/832 supercomputer at the Livermore Computer Center in Livermore, California. The laboratory is operated for the U.S. Department of Energy by the University of California. Marcelo Gumucio, president and chief operating officer of Cray Research, said, "We are gratified to see Lawrence Livermore acquire our latest technology. The CRAY Y-MP system has demonstrated sustained performance in excess of 1 billion calculations per second on customer applications. Some programs even have achieved sustained performance in excess of 2 billion calculations per second." Gumucio added, "We know of no other general purpose computer system that has achieved this level of sustained performance."

Du Pont Company, acting as prime contractor for the **U.S. Department of Energy's Savannah River Laboratory**, ordered a CRAY X-MP EA/132 computer system. The computer was installed at the Savannah River Laboratory, Aiken, South Carolina, in the fourth quarter of 1988. The system is running Cray Research's UNICOS operating system, which is based on AT&T UNIX System V. The Department of Energy is using the Cray system for nuclear reactor safety studies, environmental impact statements, chemical analyses, and plant operation support.

Standard Oil Production Company has accepted delivery of a CRAY X-MP EA/416 system at Standard Oil's new scientific computing center in Houston. The Cray system was installed in the third quarter of 1988 to replace a CRAY X-MP/28 system that was located in Standard Oil's Technical Data Center at Dallas. The complete configuration includes a 128-million-word SSD solid-state storage device and 60 Gbytes of disk storage. The computer system is used for seismic processing and reservoir simulation. Jerry D. Bullock, Standard Oil's executive vice president, technology, said, "Ours is the largest Cray supercomputer configuration installed to date in the petroleum industry. Instead of performing exploration and production supercomputing on separate machines, we have found it possible to meet both needs with a single resource." Standard Oil Production Company is the wholly-owned exploration and production subsidiary of BP America Inc., Cleveland, Ohio. BP America is the U.S. affiliate of The British Petroleum Company p.l.c.

The **University of Kiel** in West Germany installed a CRAY X-MP/216 computer system in the fourth quarter of 1988 at the university's computer center in Kiel. The purchased system replaced a CRAY X-MP/18 system that was installed in the first quarter of 1988. The computer is used for basic research in fields such as chemistry, physics, geophysics, and oceanography. Peter Grosse, director of the university's computer center, said, "The installation of the CRAY X-MP/216 system doubles the center's supercomputer capacity within a year, allowing more researchers supercomputer access. In particular, the West German Institute for Oceanography will run enhanced simulations of the circulation of the North Atlantic and the Baltic Sea."

Mitsubishi Motors Corporation has ordered a CRAY X-MP EA/116 computer system. This was the fourth Cray system to be purchased by a Japanese automotive customer in 1988. The purchased system will be installed in the third quarter of 1989 at Mitsubishi's Research Center in Okazaki, Japan. Mitsubishi will use the system for structural analysis, aerodynamic simulation, crash simulation, and combustion simulation.

The **University of Linköping** in Sweden ordered a CRAY X-MP/48 computer system to be installed at the computer facilities of Saab-Scania, a Swedish manufacturing company. The purchase was initiated by the Swedish Natural Sciences Research council. The system will be used to provide a national supercomputing facility for academic research, and will be accessible through the university's SUNet network, which connects all of the universities in Sweden and is linked to other Scandinavian university networks. The system will be applied primarily to academic research in fields such as physics and chemistry, and also will be used by Saab-Scania for aircraft and automotive design.

Scripps Clinic and Research Foundation in La Jolla, California has ordered a CRAY X-MP EA/116se computer system. The leased system, which will be installed in the second quarter of 1989, will replace a CRAY X-MP/14se system that was installed in the fourth quarter of 1987. The supercomputer will be used for biotechnology applications, such as designing and deciphering the shapes and dynamics of peptides, proteins, and other biological materials. Scripps Clinic was founded in 1924 and is one of the largest private, nonprofit biomedical research institutions in the country.

Lockheed Missiles and Space Company, Inc., installed a CRAY X-MP EA/232 system in the fourth quarter of 1988 at the Lockheed Engineering Computing Facility in Sunnyvale, California. The leased system replaced a CRAY X-MP/28 system. Lockheed is using the system for finite element modeling, as well as aerodynamics and fluids computations. Lockheed recently has converted to Cray Research's UNICOS operating system.

The **Naval Weapons Center** in China Lake, California, has ordered a CRAY X-MP EA/116 computer system. The center, which is a new customer for Cray Research, will install the purchased system in the first quarter of 1989 at the Scientific and Engineering Facility at the Albert A. Michelson Lab in China Lake, California.

The **University Corporation for Atmospheric Research** has ordered a CRAY X-MP/18 system to be installed at the National Center for Atmospheric Research in Boulder, Colorado, in the second quarter of 1989. The leased CRAY X-MP/18 system will be used to convert users' codes from the Cray operating system COS to Cray Research's UNICOS operating system, which is based on AT&T UNIX System V. The used system will replace NCAR's CRAY-1A system, which has served NCAR for 11 years. NCAR will continue to apply its CRAY X-MP/48 system to climate and weather research.

TECNATOM, S.A., a company that serves the major Spanish electrical utility is scheduled to install a CRAY X-MP/14se computer system in Madrid, Spain, in the second quarter of 1989. The purchased system will be used for real-time applications, including nuclear reactor simulation and plant operator training. TECNATOM will use the real-time function of Cray Research's UNICOS operating system.

APPLICATIONS UPDATE

RENDER presents molecular structures

The RENDER graphics program graphically displays molecular structures. The program, which is written in Fortran, runs on all Cray computer systems under all Cray operating systems.

RENDER reads one file to define one molecular image. The first part of the file defines control parameters that will be used by the program. Examples of these parameters include the size and aspect ratio of the image, background color, secondary light contribution, ambient light contribution, eye position, and light source position. The second part of the file is a list of atoms that constitute the molecule. Each atom is defined by its color, size in angstroms, and position in three-dimensional Cartesian space.

The output from RENDER is a generalized raster image of a molecule. From this point the data can be written to graphics devices with minimal effort. RENDER has been used extensively on workstations for generating images of proteins. Since it has been converted to run on Cray systems, RENDER has been used to create still images of molecules and to produce molecular animation sequences.

For more information about using RENDER with Cray computer systems, contact David Bacon, Department of Biochemistry, University of Alberta, Edmonton, Alberta, Canada; telephone: (403) 432-4575, or contact John Haslip, Cray Canada, Inc., 800 Royal Trust Tower, Edmonton Centre, Edmonton, Alberta, T5J 2Z2, Canada; telephone: (403) 426-4220.

Geovecteur solves seismic tasks

Geovecteur is a complete processing software package for two-dimensional and three-dimensional land and marine seismic analysis. The code, developed by Compagnie Generale de Geophysique (CGG), runs on CRAY X-MP/EA, CRAY X-MP, and CRAY-1 computer systems under the COS operating system. CGG plans to release a UNICOS version of Geovecteur in mid-1989.

By using the Geovecteur package with Cray systems, users can achieve productivity and perform tasks that cannot be accomplished on other computer systems. For example, CGG has been able to accomplish one-pass, three-dimensional depth migration, the only method available to give an exact image of subsurface structures. Geovecteur allows geophysicists to define their own seismic processing, including their input and output data, in a simple and concise language that is independent of Cray machine language. The package is an open system, allowing users to modify the application program or add new ones.

Geovecteur offers a wide range of job-oriented task modules that are applicable to two-dimensional and three-dimensional surveys, with two-dimensional surveys considered a special case of three-dimensional surveys. For each type of processing — deconvolution, migration, residual static corrections, antimultiples, or others — the system offers several methods of solution. Examples include migrations in the FX and FK domains, migration by finite difference, and depth migration.

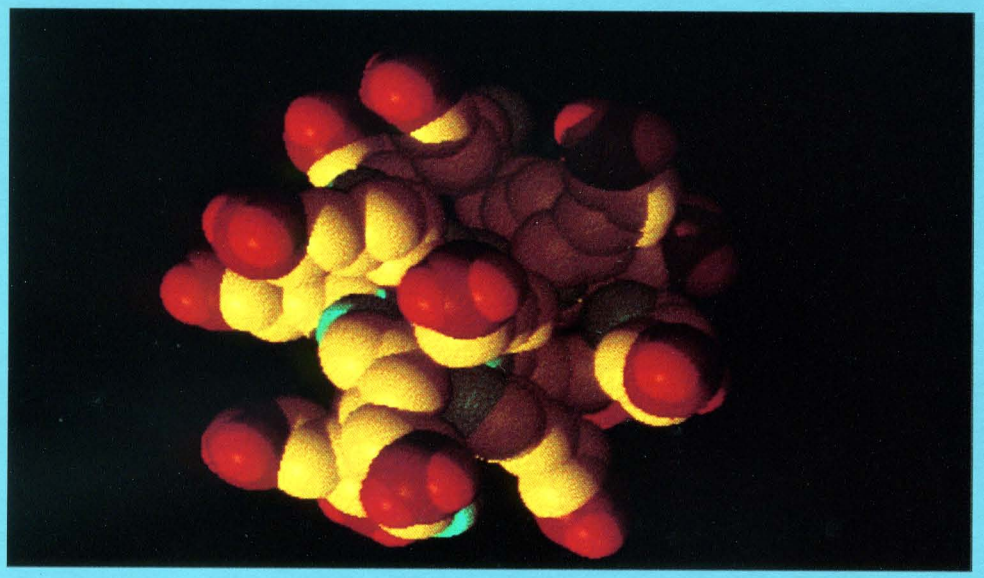
One program, TDMIG, is a true one-pass, three-dimensional depth migration program. The code requires great computational capability to process very large amounts of data quickly and simultaneously to avoid the risk of interruption.

TDSAT is a supercomputer version of the SATAN 3-D program. For each shot point and each geophone, the program searches for the correction that provides the maximum amplitude and spatial coherency (x, y) for the common midpoint (CMP). The starting point is a strip of subsurface CMP lines from the data volume. A combined in-line and cross-line dipscan then is performed, leading to three-dimensional selection of planar vectors. The results are written to tape for harmonization. Using a supercomputer allows the survey to be divided into very wide runs. A certain degree of overlap between these runs can eliminate the harmonization phase.

INPOL is used to create interpolated traces between existing traces in order to avoid spatial aliasing, which deteriorates migration results. This intelligent multidip interpolation problem is applicable to both land and marine data.

WEMUL, an antimultiple processing program, was developed by CGG for offshore seismic surveys in the North Sea.It is considered to be the most efficient tool for eliminating multiples due to resonance in the water layer.

For more information about using the Geovecteur software package on Cray systems, contact Dick Maxwell, CGG American Services, Inc., 2500 Wilcrest, Houston, TX, 77042; telephone: (713) 784-0740, or contact Bill Kamp, Cray Research, Inc., 1333 Northland Drive, Mendota Heights, MN, 55120; telephone: (612) 681-3665.



Stable micelle structure composed of 16 octanoate chains as displayed with the RENDER graphics program. Data were generated by a molecular dynamics simulation performed at the National Research Council in Ottawa.

USER NEWS

Supercomputing votes in the U.S. Senate

When examining the voting patterns of the U.S. Senate, many people would assume that decisions are made along party lines — with Democrats and Republicans voting in step with their fellow party members. However, a recent supercomputer study shows that the Senate's voting patterns are determined by alliances of coalitions that represent four distinct world views.

These four groups — the liberal Democrats, the conservative Democrats, the moderate Republicans, and the conservative Republicans — form unique partnerships to pass legislation through Congress, explains Jacob Stampen, a professor and political analyst at the University of Wisconsin at Madison.

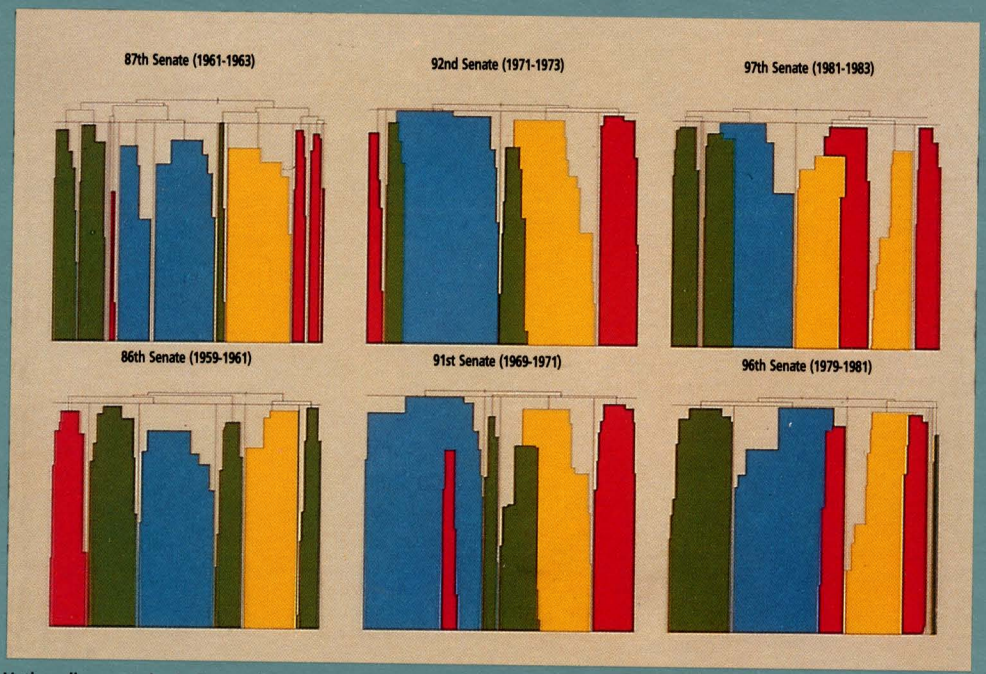
Stampen first became interested in voting behavior and public policy in the late 1960s. Using an unconventional method of factor analysis, he determined voting patterns and even predicted legislation. "But when using factor analysis on conventional computers, there are always methodological problems — the voting data aren't set up on a ratio/interval scale, and standard computers just don't have the capacity for the problem," he says.

In 1986 Stampen teamed up with John Davis, a UW political science methodologist, to study the voting alignments in six U.S. Senates between 1959 and 1981. Because this problem involved mapping hundreds of relationships, Stampen and Davis applied for an allocation of time on the CRAY X-MP system at the San Diego Supercomputer Center.

"The Cray system could handle the large matrices that caused capacity problems for other computers," says Stampen. "Another reason we needed a supercomputer was that the director of our experimental design lab at Madison, Frank Baker, had developed a cluster analysis program that was extremely computationally intensive."

The program compares voting records of each Senate member to every other Senate member, matching the representatives with the most similar voting records. "The two people with the most similar records are matched first, and then the others are connected until all the members are defined as one group," says Stampen. The program identifies where the most basic breaks occur between core groups. These multi-issue coalitions combine in many ways to form larger voting coalitions.

Each multi-issue coalition has unique characteristics that determine how it will interact with the other groups. Conservative Republicans tend to support strong



Voting alignments in six Senates between 1961 and 1983. Green represents conservative Democrats (Boll Weevils), blue represents liberal Democrats (Honey Bees), yellow represents conservative Republicans (Yellow Jackets), and red represents moderate Republicans (Gypsy Moths).

defense and aggressive foreign policy, but tend to vote against domestic spending, while liberal Democrats tend to vote against defense spending and aggressive foreign policy, but favor domestic spending. "The conservative Democrats tend to be the big spenders in Congress because they support both guns and butter — they are willing to support a strong defense as well as domestic policy," he says. "Moderate Republicans tend to be the opposite. They are very skeptical because they do not believe they will see results from their investments."

Conservative Republicans are more likely to join forces with conservative Democrats than moderate Republicans or liberal Democrats, while moderate Republicans often choose sides with the liberal Democrats. "The two groups that don't interact much are the conservative Republicans and liberal Democrats. They rarely team up — only once in a while for a civil liberties issue," says Stampen.

Stampen and Davis have observed a unique balance of power in this decade's Congress. "When President Ronald Reagan was elected in 1980, the faction of the Senate that grew the most was the conservative Republicans, and the faction that was hurt the most was the moderate Republicans," explains Stampen.

Although the majority of senators were Republican, they found it difficult to gain the support they needed from other factions to pass Reagan's agenda. "These moderates who did not fare well in the elections found that they had a lot of power because they could work out agreements across party lines," says Stampen. "If a few more moderates had lost in the election of 1980, I think our history would have been different. As it turned out, the moderates became the catalysts for compromises that stopped the Republicans from getting a lot of things that they wanted."

But perhaps the most useful application of this research is its explanation of public policy. "If I tried to explain the issue of student financial aid in terms of individuals, I would come up with a very muddled explanation," Stampen says. "Explaining it in terms of Republicans and Democrats isn't much clearer because the two parties split on the issue. However, if I explain the issue in terms of these subparty coalitions, the outcome is very clear.

"When Reagan came to office he wanted to dismantle the Department of Education and pass its responsibilities back to the states," explains Stampen. However, he was up against an alliance of moderate Republicans and liberal Democrats that had formed 20 years before to support federal student financial aid. The moderate Republicans and the liberal Democrats worked in committee and on the floor to prevent major changes in the system.

Next, Stampen and Davis would like to carry their research to the House of Representatives, and then to the American voters. "We could take opinion polls and compare the groups that appear in the legislative bodies to the constituency. We could find out if the same divisions appear, if they are well-matched, which groups are best represented, and which groups are least represented — a problem only possible to solve with supercomputers," says Stampen.

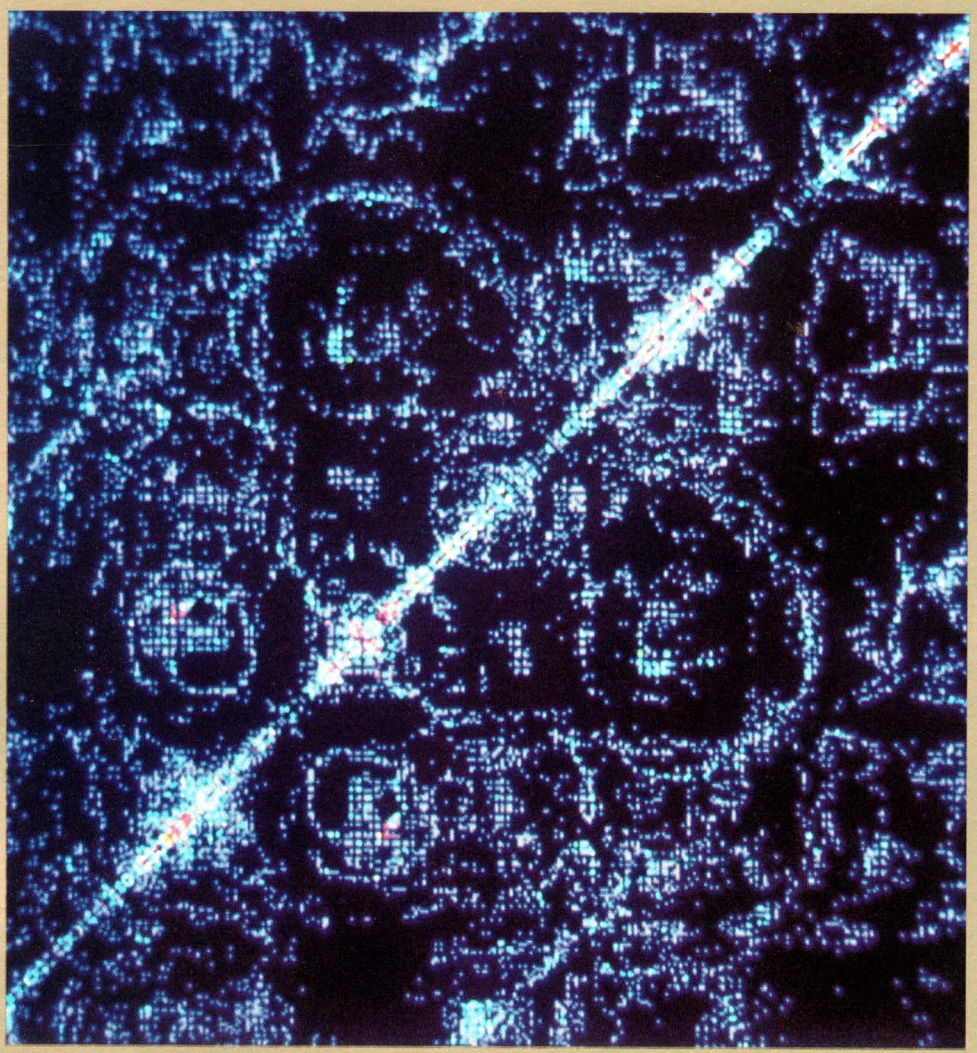
"When we do our analysis of the House, that will be the first legislative study of its size with a matrix of 435 times 300 highly contested votes," he says. "When we use the same method on both the House and Senate, that will be another first." He adds, "This study is nice from a supercomputing perspective because it is taking us to new frontiers."

Life-long violinist meets supercomputer

For hundreds of years, violin makers have judged the suitability of a piece of wood for a particular violin by applying a variety of qualitative tests. A popular method of testing the wood was to bend it in two directions, then twist it to determine its elasticity.

Robert Schumacher, a life-long violinist and a physics professor at Carnegie Mellon University in Pittsburgh, wondered if this method was sufficient. "The theory of elasticity says that a material like wood requires nine independent elastic constants for a complete description. For the most part, violin makers have been content to describe violin wood with only three elastic constants — which leaves six unspecified elastic constants unmeasured," he says.

"The purpose of my research was to see if it was necessary to know what these six other elastic constants are. In other words, are violin makers — even modern scientific ones — fooling themselves by specifying



Data taken from an accelerometer attached to the top plate of a violin while it was being played. The data were processed on a Cray system using a recurrence plot method. The output shows the effects of vibrato on the periodicity of a bowed string note.

three elastic constants as being sufficient to describe the wood for their purposes?"

The nine elastic components would be difficult to determine experimentally, especially in the audio frequency region, so Schumacher turned to the CRAY X-MP/48 system at the Pittsburgh Supercomputing Center (PSC). Schumacher computed the normal mode frequencies of a representative arched shell, using the ABACUS code, a commercial finite element package. "I tried to perform the calculation on a VAX 780 computer, and it was hopelessly slow — and if a calculation is long enough, you will never get it done. What's worse, using the VAX system cost an incredible amount of money," he says. "I could not have done this without a Cray system or some other extremely fast computer with the financing that was available."

Schumacher's results told him that for the most part, violin makers had been right, within qualitative limits. "When you take a piece of Norway spruce from which the top plates of most high-quality violins are made, and you cut it into the configuration of a violin, the other six elastic constants don't matter very much. The three elastic constants — the two Young's moduli and the sheer modulus that is relevant to a flat or slightly curved plate — are sufficient to explain the frequencies of vibration in the top plate within a few percent."

Schumacher won't use his results to optimize the modern violin, but to understand its form and evolution. "An awful lot of very clever and perceptive people have put a lot of time into making violins," Schumacher explains. "If you look at the first violin in modern form, which dates from about 1530, the evolution appears to be very slow and very minor — but this certainly is not a player's point of view." Schumacher's results will enable violin makers to substitute some synthesized components if the European wood used to make violins becomes an endangered resource. Understanding the violin's structure also will be useful for those who simulate musical instruments by computer.

Schumacher, who is currently on leave from Carnegie Mellon University, is working at Stanford University's Center for Computer Research in Music and Acoustics. He currently is using the CRAY X-MP/48 system at PSC to study how periodic wave forms are produced by bowed string instruments, other acoustic instruments, and other sources.

"If you try to put the physics of a bowed string into a computer program to produce output like a real instrument, you need to know how successful you are at producing this output," he explains. One of the characteristics of a real instrument is that its output is never completely periodic; a certain amount of fluctuation appears in the wave form. Schumacher is trying to determine the physics needed to produce the same sort of slightly nonperiodic output that is produced by a real instrument. The results will help other researchers who are trying to simulate instruments by computer.

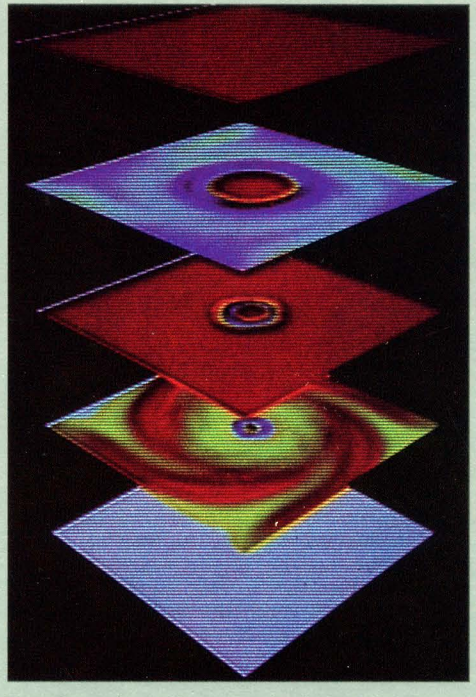
What's next? Schumacher can't say what his next project will be, but it would be safe to guess that it will involve musical instruments and computers.

Cray system looks inside a hurricane

Although hurricanes typically develop over the oceans, in the last year many of these tropical storms have been observed in central Ohio. And while most of these weather systems produce damaging winds and torrential rains, the hurricanes in Ohio have peacefully come and gone.

These hurricanes are video models that have been produced with the Ohio Supercomputer Center's CRAY X-MP/28 system. The video hurricanes allow researchers to travel to the eye of a storm in the safety of the laboratory, observing movements of air, moisture, and energy.

Without supercomputers, studying hurricanes can be a difficult and expensive task, says Jay Hobgood, assistant professor of geography at Ohio State University. "Despite the fact that you hear about hurricanes in the press a lot, they are relatively infrequent weather systems. Generally they occur over tropical oceans where it is very difficult to collect data." While satellites yield some clues about hurricane development, the data they provide lack detail. Another method of data collection — flying special aircraft into the storms — is costly and inconvenient.



Temperature changes in a mature hurricane. Areas of greatest warming are shown in red, and areas that experience cooling are shown in blue. The four levels represent from top to bottom, the upper troposphere, middle troposphere, lower troposphere, and the planetary boundary layer. The small red area in the center represents the eye of the hurricane.

One part of Hobgood's research has been the simulation of hurricanes as they might have developed over the Atlantic Ocean during the last Ice Age. He then compares the results of those simulations with simulations of today's hurricanes. "As we expected, the storms from the Ice Age were much weaker than those we see today," says Hobgood.

He explains that hurricanes develop from the release of latent energy, such as the energy that is released when water vapor condenses from a gas to a liquid. The cold, dry atmosphere of the Ice Age was less conducive to the release of latent energy.

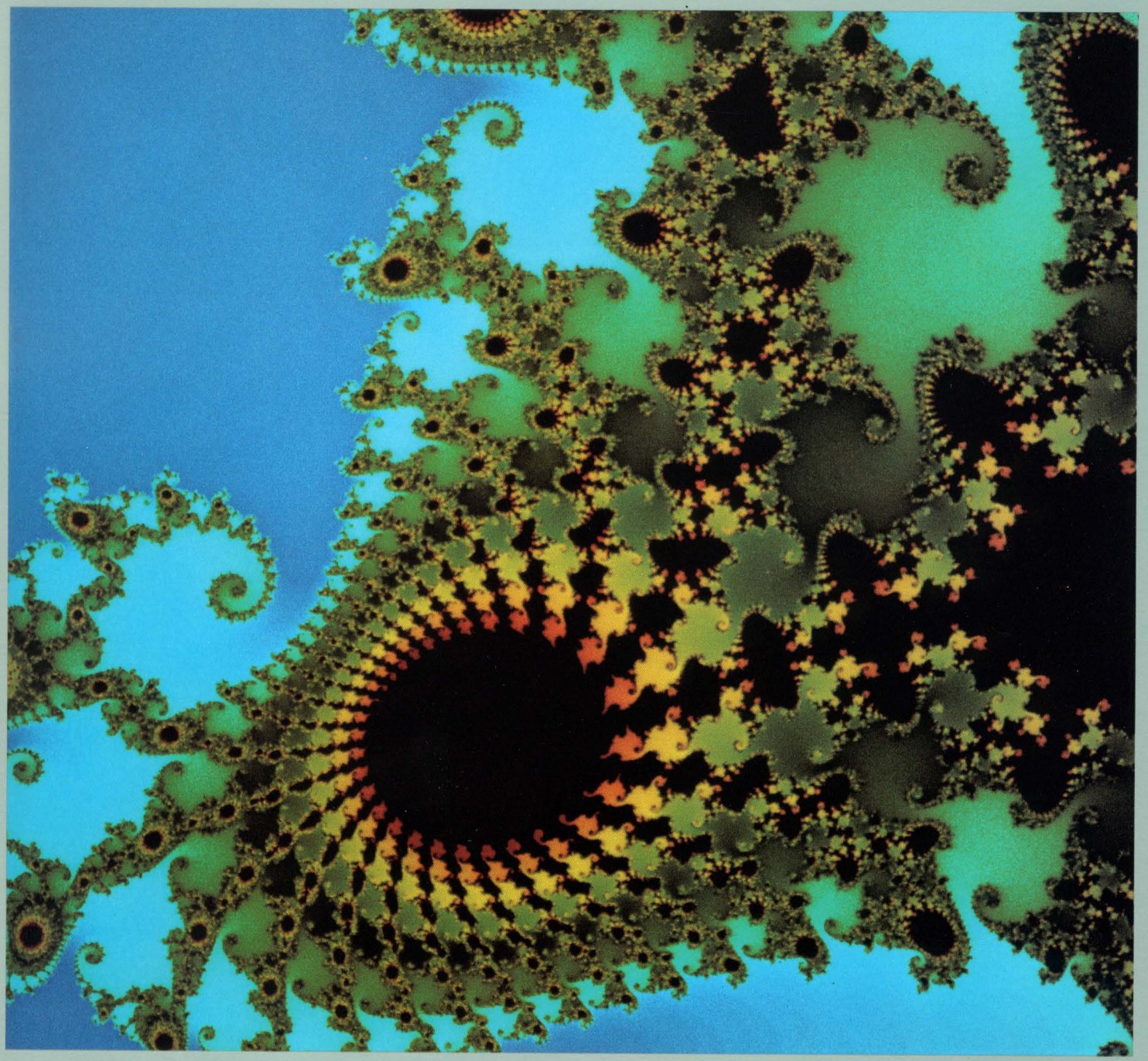
Studying Ice Age hurricanes will help researchers predict how weather systems might change if the Earth's climate changes. "We want to look at the greenhouse effect, and see how that might affect strong tropical cyclones around the world," says Hobgood. "We also plan to extend the model to the Pacific and Indian Oceans — other places where tropical storms develop."

Simulating hurricanes involves sifting through massive amounts of data and performing thousands of calculations — requiring the speed and memory of a supercomputer, says Hobgood. "Before I came to Ohio State I was using a Control Data CYBER computer at the University of Nebraska. Now I can do in a couple of days what used to take me the better part of a month to complete."

Hobgood has been working with the Ohio State Supercomputer Graphics Project to produce unique video animations of the results. "We have produced in three dimensions the temperature structure and relative humidity of the storm. Now we are working to show the movement of the air through various trajectories," says Hobgood. "We also are working on a split screen approach — so we can compare the developments of two storms on the same screen simultaneously. That way we can see which one is intensifying more quickly, or which one is stronger at a particular moment in time.

"The results of this research will give us a better understanding of what is going on inside hurricanes," says Hobgood. "And the more we learn about these systems the better we will be able to forecast them."

GALLERY



A region of a Mandelbrot set computed on a CRAY-2 system by Joel Neisen, graphics project leader at the Minnesota Supercomputer Center in Minneapolis. The Mandelbrot set is an aesthetically pleasing and useful way to depict patterns in natural phenomena. The power of a CRAY-2 system enables mathematicians to create these complex, high-resolution images in seconds. Please send Gallery submissions to CRAY CHANNELS at the address inside the front cover.