A CRAY RESEARCH, INC. PUBLICATION

# CRAY CHANNELS

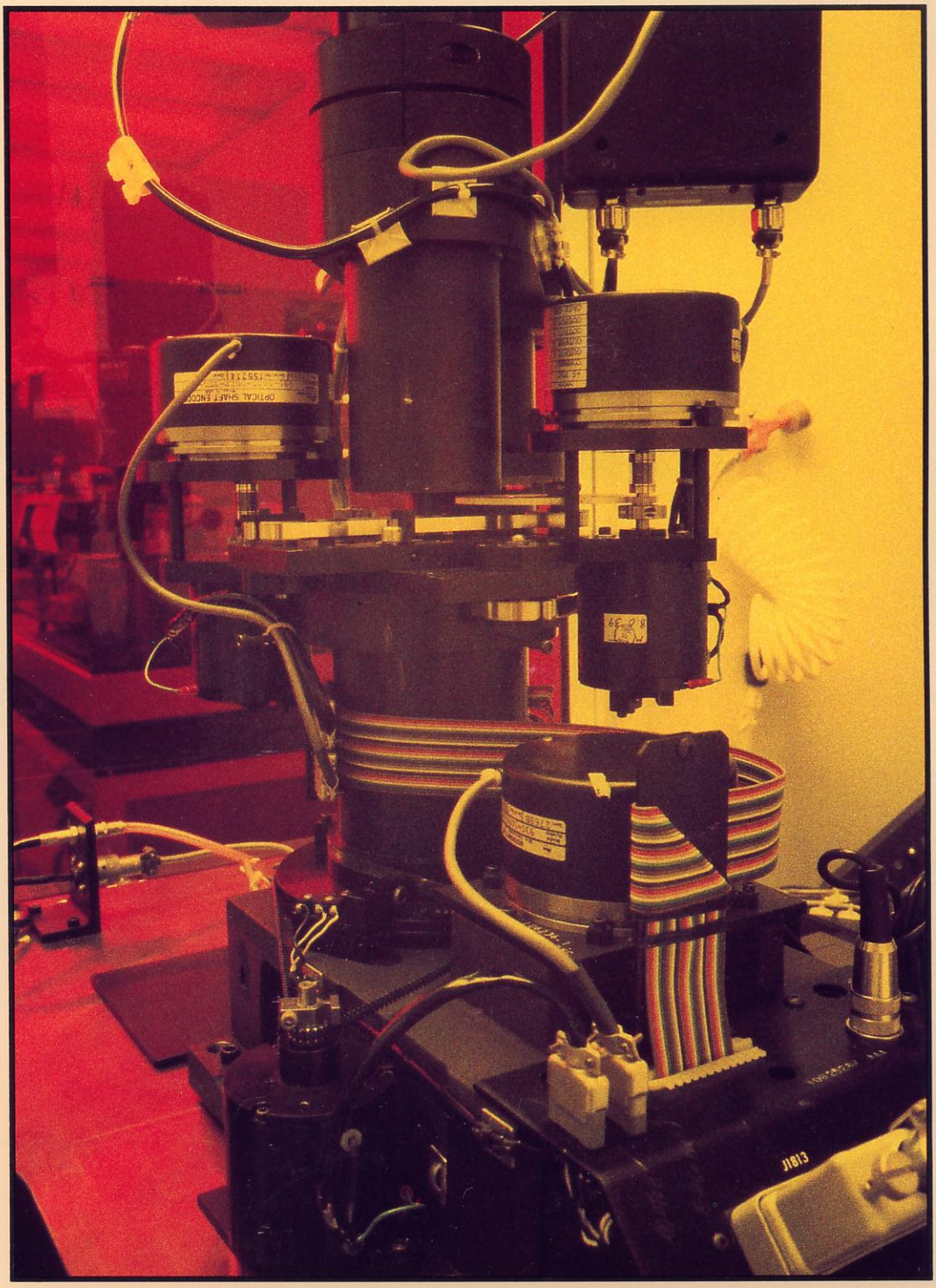
Spring 1985

## FEATURE ARTICLES:



## DEPARTMENTS:



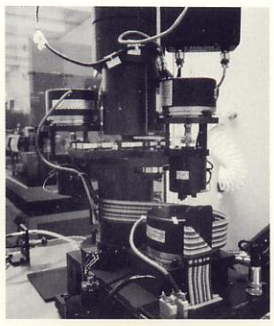


## IN THIS ISSUE

The evolution of chemistry from alchemy was an important passage from magic to science. But today a kind of magic is returning to chemistry — the magic of computers. Computer simulations and graphic displays provide a window into the microscopic world of molecules and molecular systems. Simulating the activities of this world can now answer some of the "how" and "what if" questions chemical and genetic engineers have been asking. The combination of mathematical molecular models and high-speed computers promises to greatly benefit diverse industries, ranging from plastics to pharmaceuticals and biotechnology.

CRAY computers can make molecular modeling accurate enough to be useful and rapid enough to be practical. This issue of CRAY CHANNELS features two articles on molecular modeling using CRAY computers. With these we hope to give some indication of the value CRAYs bring to the arena of computational chemistry. In another feature article, we highlight the Cray Pascal Compiler. Our regular departments look at a tribute to Seymour Cray and the netting of a remotely piloted vehicle.

Although magic has been largely discredited by science, seen in the light of today's technology, it may have been ahead of its time.

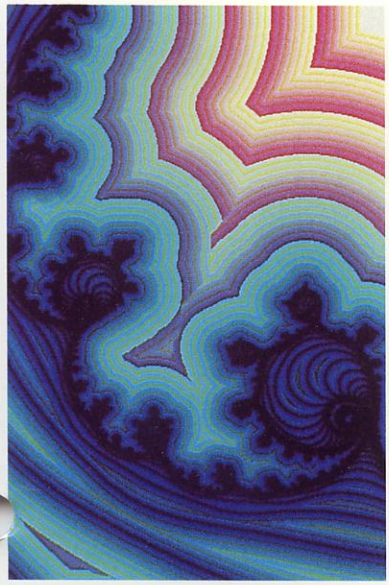


**On the cover** a pattern generator for making reticles is shown. Reticles are patterns, similar to photographic negatives, that are used as master images for printing electronic circuitry. This pattern generator is currently being used to create reticles for gallium arsenide circuits at Cray's Advanced Research Project Facility in Chippewa Falls, Wisconsin.

# CRAY CHANNELS

**A Cray Research, Inc. publication**

**Spring 1985**



12

## FEATURES



## DEPARTMENTS



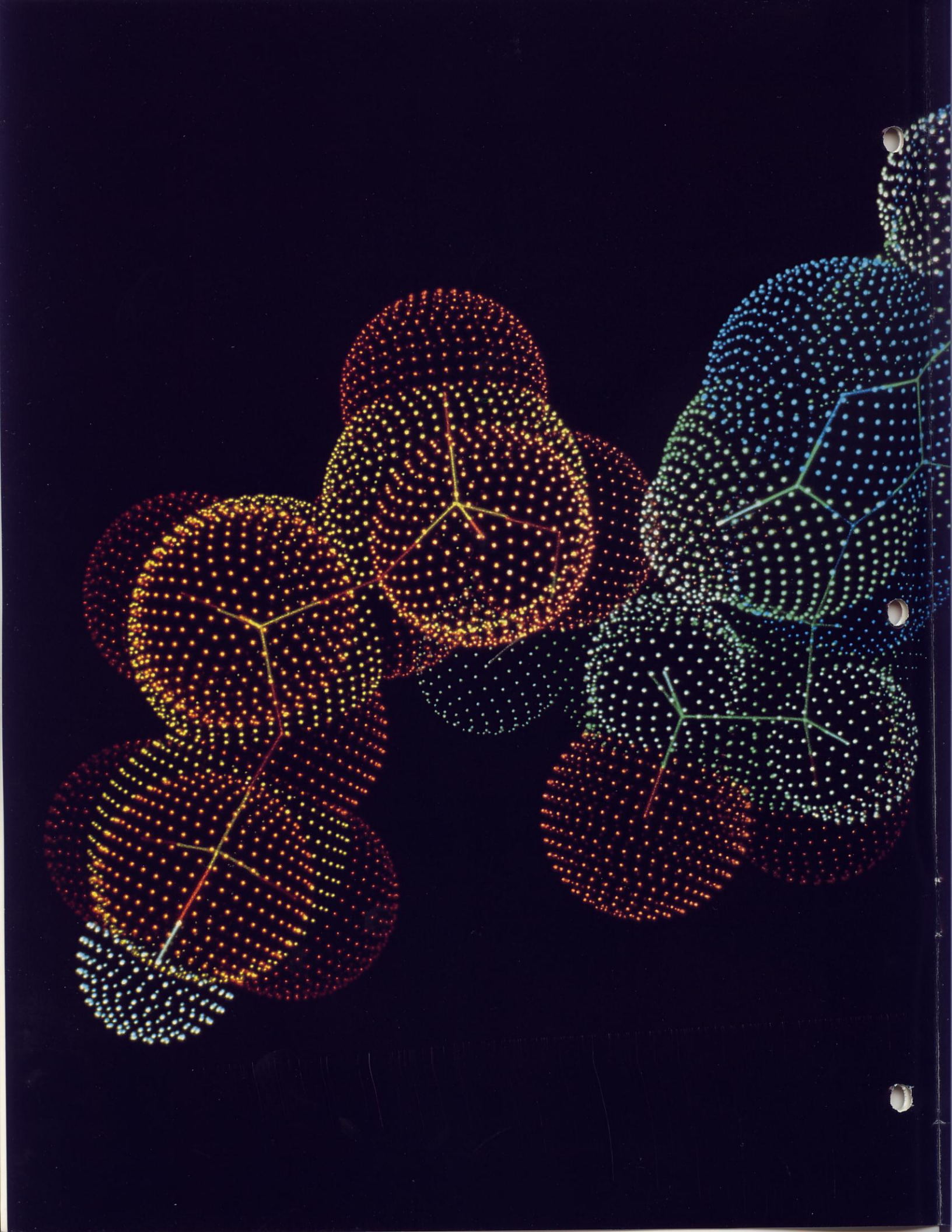
CRAY CHANNELS is a quarterly publication of Cray Research, Inc. intended for users of Cray computer systems and others interested in the company and its products. Please mail subscription requests, feature story ideas and news items to CRAY CHANNELS at Cray Research, Inc., 608 Second Avenue South, Minneapolis,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

Volume 7, Number 1
Spring 1985

**Editor**
Carol Smith

**Contributing Writer**
Ken Jopp

**Design**
Cindy Erickson
Eric Hanson



# Computational chemistry of biomacromolecules

*by George Seibel*
*Department of Pharmaceutical Chemistry*
*University of California, San Francisco*

Computational chemistry is gaining recognition as a powerful branch of the chemical sciences. Although its roots predate computers (the earliest quantum chemical self-consistent field calculations were performed with pencil and paper by the father of theoretician D.R. Hartree), high-speed computing has now made it possible to simulate simple chemical systems and calculate useful approximations of their behavior in various environments.

The development of computational chemistry parallels the growth of numerical methods in other fields, notably physics, mathematics and engineering. For years, numerical methods were considered less reputable than analytic theory, and any use of empirical parameters was viewed with distaste by strict physical chemists.

The rightful place of computation in chemistry is still being determined. Classically, experimental chemists produced data, to which theorists applied mathematics in an attempt to construct comprehensible models of an invisible world. In the early 1950s, a few scientists began to take empirical numbers from the experimentalists and mathematical models from the theorists and put them together on the computers that were becoming available. They began to learn things about the molecular world that could not be learned in other ways.

Progress was slow with the limitations of early hardware, but the speed and capacity of hardware grew, expanding the scope of problems that could be addressed. The introduction of virtual architectures and supporting software in a package that an academic researcher could afford allowed the field of computational chemistry to flourish. Significant discoveries have already been made by computational chemists. An example is the uncovering of a never before seen low energy conformation of a spherand ionophore during molecular mechanical simulations by Peter Kollman at the University of California at San Francisco (UCSF). The predicted molecular structure has since been verified experimentally. This is one example of computation pointing the way for experimentalists.

## Computational methods survey

The computational chemist today has a spectrum of tools to apply to chemical problems. At one extreme are the ab initio (from first principles; literally, from the beginning) quantum mechanical methods. These employ iterative algorithms to calculate the distribution of electrons about a molecule's nuclei, and make use of no empirical parameters. They require tremendous amounts of memory and computation even for molecules with only a handful of atoms. A number of variations exist within this genre, each introducing successively more approximations in order to treat larger systems. The quantum chemical programs are essential for treating systems in which chemical reactions are occurring, that is, where the electronic structure of a molecule is being rearranged.

In situations where bonds are not being made or broken, however, much simpler empirical energy calculations have been used very successfully. These empirical energy approaches, known as molecular mechanics and molecular dynamics, represent the other end of the spectrum from ab initio methods. While ab initio quantum mechanics calculations are limited to a maximum of about 20 atoms because of memory and computational limits, molecular mechanics and dynamics can treat systems of thousands of atoms. These methods are used to calculate the most stable structures of molecules, the energies of interaction between molecules and, in the case of molecular dynamics, the detailed motions of the atoms in a molecular system at a specified temperature. Because these methods can be used to describe very large molecules, they are of great use in the study of macromolecules found in biology.

In addition to quantum mechanics and empirical energy programs, the computational chemist employs a host of tools in the exploration of the microscopic world. Computer graphics plays an important role in allowing one to visualize complex molecular structures and interactions. Interactive molecular graphics programs such as MIDAS, FRODO and MOGLI allow the user to manipulate three-dimensional chemical structures in real time on high-speed graphics devices. These programs allow chemists to visualize the results of their calculations, as well as to graphically build new model systems for input to various computational programs. The Connolly surface algorithm is often used, generating a series of dots on the solvent-accessible surface of a molecule. These dot surfaces may be manipulated in real time in concert with the more common stick figure structural representations. The surfaces show the shape and topology of the molecule, and may be color coded to show the electrostatic potential at the surface of the molecule. This is useful in evaluating the electrostatic complementarity of drug-macromolecule interactions.

## Molecular mechanics and molecular dynamics

### Background

In order to quantitatively model the structural aspects of molecular systems in a reasonable amount of time, it is necessary to use empirical energy techniques such as molecular mechanics or molecular dynamics. In this section, we will examine the theory that underlies both of these methods.

In this type of calculation, molecules are simply considered as collections of atoms held together by chemical bonds, behaving as particles held together by springs. In the types of molecules that biochemists are typically interested in, chemical bonds are known to have very characteristic "natural" lengths and angles that depend on the identity of the two atoms making up the bond and upon their immediate environment. The energies and forces involved in distorting a molecule slightly from its natural bondlengths and angles can be represented very effectively by a simple harmonic potential such as:

$$E_{stretch} = K_{AB}(r_{ab}-r_{eq})^2$$

where the energy due to bond stretching, $E_{stretch}$, is equal to a parameter $K_{AB}$ whose value depends on the two atom types A and B making the bond, multiplied by the square of the difference between the actual bond length and the "natural" or "equilibrium" length of a bond between atom types A and B. A similar equation describes the energy involved in distorting three atoms from their "natural" bond angle. Terms that describe the potential energy contributions of dihedral angles, electrostatic interactions, van der Waals forces, and hydroden-bonding are also included in the total potential function (Figure 1).

$$E_{total} = \sum_{bonds} K_{AB}(\tau_{AB} - \tau_{eq})^2 + \sum_{angles} K_{\Theta}(\Theta - \Theta_{eq})^2 + \sum_{dihedrals} \frac{V_n}{2}[1 + \cos(n\varphi - \gamma)] +$$

$$\sum_{i<j} [\frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^{6}} + \frac{q_i q_j}{\epsilon R_{ij}}] + \sum_{H\text{-}bonds} [\frac{C_{ij}}{R_{ij}^{12}} - \frac{D_{ij}}{R_{ij}^{10}}]$$

*Figure 1. Potential energy equation, or "force field," used to calculate the potential energy of a particular conformation of atoms in a molecular system. It is typically evaluated 100,000 times in a molecular dynamics simulation.*

Atoms making up the system are assigned "types" based on their environment. For example, a carbon atom bonded to oxygen behaves differently than a carbon atom bonded to hydrogen, so the two kinds of carbon are given different types. Each of the potential terms contain one or more parameters that are assigned on the basis of the types of the atoms that are interacting. The values of the parameters are selected empirically to duplicate the energy and structural features of known molecules before they are applied to new systems.

*Computational expediency is very important in treating macromolecular systems, as a single term in the potential may be evaluated hundreds of millions of times in the course of one dynamics simulation.*

One will note that all of the terms in the potential function are mathematically simple. More importantly, they are analytically differentiable, so that forces (the derivative of energy with respect to coordinates) may be evaluated quickly. Computational expediency is very important in treating macromolecular systems, as a single term in the potential may be evaluated hundreds of millions of times in the course of one dynamics simulation.

When a potential function is fitted with appropriate empirical parameters, it is known as a "force field". Because of the subtle interplay of the various terms in their total effect on the final structure of a molecule, the correct parameterization of a force field is a mammoth undertaking. Researchers tend to parameterize a force field to deal well with a certain class of compounds. The potential function illustrated in Figure 1 is from the macromolecular simulation package AMBER developed at UCSF by the Peter Kollman group. It is intended for treating proteins, nucleic acids, and their solvent and counterion interactions, and is also capable of simulating drug-macromolecule interactions and water-water interactions. Other force fields have been developed to treat unusual or highly strained small molecules and typically have more complicated potential functions.

**Methodology**

Molecular mechanics and molecular dynamics are related in their use of the empirical force field. Molecular mechanics is used to calculate quickly the positions of the atoms in a molecule such that its potential energy is minimized. The conformation of atoms with the minimum energy is the molecule's most stable structure and represents the molecule as it would most likely exist in nature. The low energy conformation is calculated by using the force field as a response function, and applying standard numerical optimization methods to minimize the potential energy with respect to the atomic coordinates. Essentially, the atoms are moved by small increments in whatever direction will reduce the energy of the structure, until further movement produces no further energy decrease. The physical analogy to molecular mechanics energy minimization is the immediate quenching of the starting structure to zero degrees Kelvin.

However, this method of finding a molecule's lowest energy conformation has the disadvantage of dropping the molecule into the first low energy conformation encountered. One is not guaranteed of finding the ultimate minimum energy conformation. A more effective search of conformational space may be performed through the use of molecular dynamics. With this technique, the force on each atom of the system is evaluated using the force field. Through the solution of Newton's equations of motion, each atom is allowed to move in the direction of the force it experiences for an extremely small time step, on the order of one to two femtoseconds ($10^{-15}$sec). The force on each atom is then re-evaluated and the process is repeated. The very small time step is needed to ensure an accurate numerical integration of the equations of motion. Thus 500 to 1000 force evaluations are required for the simulation of one picosecond of real-time motion. This highly reiteritive process is well suited for processing on a supercomputer like the CRAY.

A simulation of this type is physically analogous to equilibrating a molecule at a fixed temperature, which can be adjusted up or down in the simulation. Thus by "heating up" (with molecular dynamics) then "cooling down" (with molecular mechanics) a molecular system, it is possible to "anneal" the system, driving it out of local energy minima in which it may be trapped. Molecular dynamics allows one to get a feel for the magnitudes of the allowed motions of the system. This is useful, for example, to see if a protein is flexible enough to allow a drug molecule access to an occluded binding site. Globular proteins were at one time considered to be fairly rigid molecules, due to their high density and also probably due to the influence of x-ray crystallography on biochemical thinking. Molecular dynamics is one tool that shows these molecules actually have remarkable flexibility.

*.... 500 to 1000 force evaluations are required for the simulation of one picosecond of real-time motion. This highly reiteritive process is well suited for processing on a supercomputer like the CRAY.*

One of the goals of computational chemistry is to provide information about molecules that cannot be obtained experimentally. An area where this has

been achieved is in the simulation of molecules in their natural environment. Most experimental structure determination methods examine molecules in the solid phase or in the gas phase. However, biomolecules exist in an aqueous environment. In addition, they are very sensitive to the presence of dissolved ions.

Using high-speed computers, it is now possible to perform molecular mechanics or molecular dynamics on large systems of one or more molecules interacting in a bath of water molecules with counterions and other dissolved solutes. The total number of atoms in such a system is very great, and the need for computational power is dramatic. Such a simulation is described below.

## AMBER on the CRAY

Recently, the molecular dynamics and molecular mechanics modules of the AMBER (Assisted Model Building with Energy Refinement) software package were put up on the CRAY. The AMBER package consists of a sequence of programs that allow the user to construct computer models of chemical systems ranging in complexity from small gas phase molecules to large proteins and nucleic acids. In addition, the models may be "immersed" in baths of simulated water taken from Monte Carlo calculations on pure water. These systems then serve as input for the molecular mechanics and molecular dynamics modules. Other modules analyze the results of the simulations and prepare graphical output.

The study of the structural properties of DNA has been a long-standing interest in our group. Working on a VAX 11/780 at UCSF, the most realistic dynamics simulation of DNA we were able to perform required us to leave the water molecules out of our model, and use several artificial constructs to mimic their presence. Working on the CRAY X-MP/48 at Cray Research's Mendota Heights, Minnesota facility, 114 picoseconds of real time motion were computed for a base paired DNA pentamer. The model consisted of a short piece of double helical DNA, with sodium counterions and a nine angstrom shell of explicit water molecules. Preliminary calculations showed that this job would have realistically taken from one to two years on the shared UCSF VAX.

## Code conversion

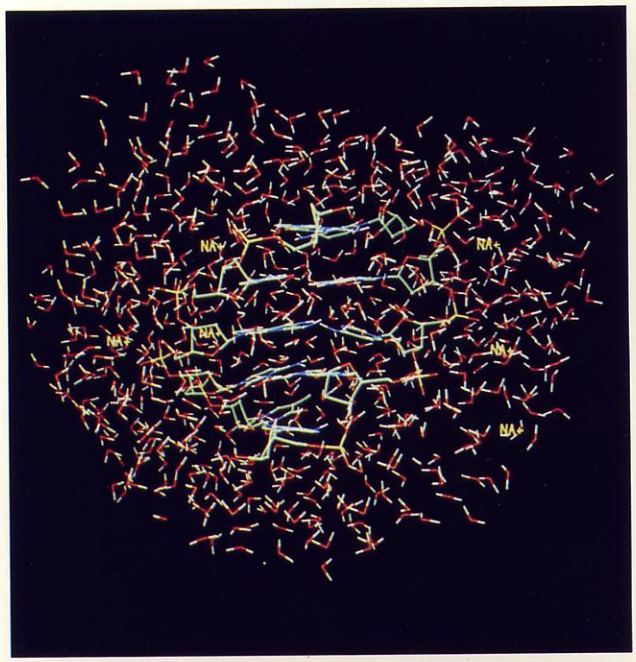
Installing the CPU-intensive modules of AMBER on the CRAY proved to be quite straightforward, as they were written in standard FORTRAN. Empirical energy calculations of this type have until now been called "non-vectorizable problems," and indeed they present an optimization challenge for vector machines. This challenge has been effectively met with the CRAY X-MP/48. By far the rate determining step in empirical energy calculations is the evaluation of the nonbonded interactions. This involves calculating the effect of each atom on every other atom in the system. Within the subroutine that performs this task are over thirty gather and scatter operations. A gather occurs whenever the following FORTRAN construct occurs:

```
A(I) = B(J(I)).
```

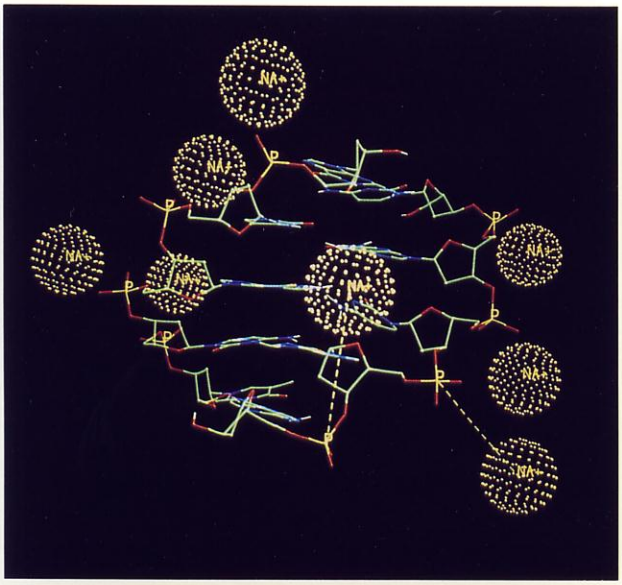
In other words, a pointer array J() points to random elements of B() that must be placed in A(). A scatter is simply the reverse assignment. These operations are implemented in hardware on the X-MP/48 (on previous CRAY computers they were performed in software). The complete optimization of the nonbonded evaluation routine took only about a day. This mainly involved breaking a large scalar loop into a series of smaller vector loops. The optimal vector version of the subroutine was written entirely in FORTRAN. No library calls to gather/scatter routines were needed, as these tasks were handled by the new vectorizing compiler, CFT 1.14. Upon seeing the FORTRAN construct illustrated above, the new compiler automatically generates the appropriate gather instruction. The vectorized subroutine is not only fast, but clean and transportable. When first put up on the CRAY, the original entirely scalar code ran forty times as fast as on the VAX. After the simple optimization, it ran 150 times VAX speed. The ease with which this performance was achieved illustrates what a well-balanced machine the CRAY X-MP/48 is for real-world codes.

## Results

Our DNA dynamics simulation required 20 hours on the CRAY. Analysis of the voluminous data (about 175 megabytes) showed that the shape of the double helix was preserved through the course of the simulation. Various structural features were in good agreement with experiment and much interest-



*DNA-water-counterion system used in molecular dynamics simulation on the CRAY X-MP/48.*



*Same model, with water removed for clarity.*

ing base and deoxyribose motion was observed. Extensive hydrogen bonding was observed between the water and DNA, as well as realistic solvation of the sodium ions. Two of the ions were displaced from the DNA backbone by water molecules and drifted through the solution. Compared to the simulation without water, the DNA helix was destabilized slightly by solvation, as expected.

## Applications

An important area of application for these methods exists in the pharmaceutical industry. As the science of biology has shifted to a more molecular basis, it has been recognized that the action of drugs is most often due to interaction of the drug molecule with a specific site on a biomacromolecule, termed the drug receptor. As the structures of more of the body's macromolecules become known, the prospect of designing new drugs for specific functions emerges.

By creating a molecule with the right shape and conformational flexibility to fit in a cleft on the surface of an enzyme, with the correct electrostatic complementarity to make it stick, it is possible to switch off the activity of that enzyme. An example of such an enzyme inhibitor is the drug methotrexate, used in cancer chemotherapy. This drug binds tightly to the enzyme dihydrofolate reductase, blocking its normal function in the synthesis of DNA. Rapidly dividing cancer cells are particularly susceptible to this blockade. Molecular mechanics can be used in favorable cases to estimate the binding energy of drugs in receptor sites, and can even be performed in real time as the medicinal chemist manipulates the two molecules on a high-speed vector graphics terminal.

Recent advances in the field of biotechnology have made it possible to alter the genetic makeup of common bacteria, directing them to manufacture proteins of any desired amino acid sequence. Thus proteins may be "engineered" to perform new functions. Molecular mechanics is now being used to evaluate the effects of amino acid changes on the binding properties and stability of proteins, in order to select the most promising modifications of a protein's amino acid sequence or to rule out bad choices.

Enzymes define one group of proteins that researchers are attempting to modify. They hope to enhance their specificity and catalytic rate, and to make them function at higher temperatures for use in industrial processes. Ultimately it should be possible to build enzyme systems to perform complicated chemical syntheses that may not be possible using the methods of classical chemistry.

Computational methods are also making inroads in areas of material design. Molecular mechanics and computer graphics have recently been proposed for use in the design of a membrane that would separate ethyl alcohol from water. In addition, simulations based upon polymer statistical mechanics show promise in the prediction of mechanical properties of new plastics. The design of structural materials at the molecular level would allow materials scientists to tune new materials to their applications in ways never before possible.

## Conclusion

In our lifetimes, we are witnessing a revolution in science; a "third way" of doing science: the numerical simulation of complex systems by computer. Numerical simulations in organic chemistry and molecular biology can effectively point the way for experimental scientists. For the more complex simulations, supercomputers are needed. They are the laboratories of the future, currently taking their place alongside the experimental laboratories we now possess. It is critically important that supercomputers be made available to the scientific talent of today, for new breakthroughs in science will be had by putting the best tools we can build into the hands of today's researchers. □

---


**About the author**

*George Seibel received his bachelor of science degree in chemistry at the University of California, Santa Cruz in 1979, where he worked in the molecular collisions laboratory. After teaching chemistry for several years at the community college level, he received his masters degree in chemistry at California State University, Fullerton, where he developed a multi-user molecular modeling system. He is currently working toward a doctorate degree in pharmaceutical chemistry at the University of California, San Francisco, where he is researching computer simulation of macromolecular systems. Part of the research described in this article was conducted during an internship at Cray Research in the summer of 1984.*


*The computer-generated image on pages 2 and 3 shows the van der Waals surface of an adenosine triphosphate molecule. All photos courtesy computer graphics laboratory, UCSF.*

# RNA modeling for biotechnology

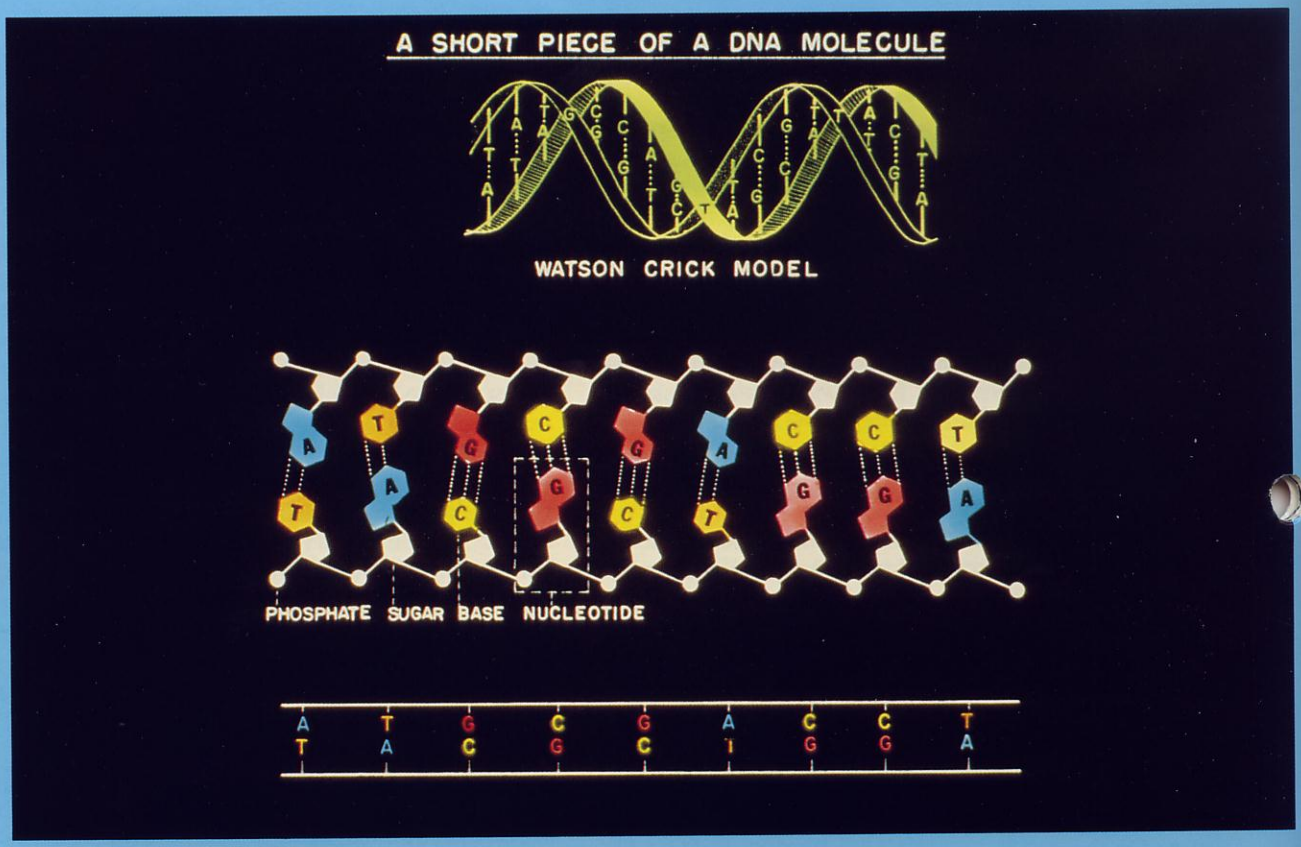



*Figure 1. Structure of DNA.*

In 1953 James Watson and Francis Crick made history and opened the door to biotechnology by discovering the double helix structure of the DNA molecule. The intervening years have witnessed a landslide of breakthroughs in our understanding of, and ability to manipulate, genetic material. By the early 1970s it was apparent that one could purify DNA from an organism, use enzymes to cut it at predetermined sites, and then reconstruct the DNA correctly in a test tube. By 1974, the first evidence of artificial gene transfer in bacteria was obtained, setting the stage for the development of commercial biotechnology. In 1976, the first genetic engineering company was established. Since then, biotechnology has matured to the point where the latest in computer technology is necessary for practical problem solving. One important area that requires high-speed computers is the modeling of secondary structures of messenger RNA (mRNA) molecules. The shape of mRNA molecules may limit the rate at which proteins are assembled in a cell. Computer modeling can be used to help find the optimum structure of mRNA to increase the efficiency with which valuable biological products are produced.

## Genetic structure

Biotechnology revolves around DNA, the chemical "database" from which all organisms are constructed. The DNA molecule (Figure 1) is a polymer composed of two strands chemically linked to form a double helix. When stretched out and laid flat, the DNA structure resembles a ladder, where the rungs are the elements of the genetic code and the outside

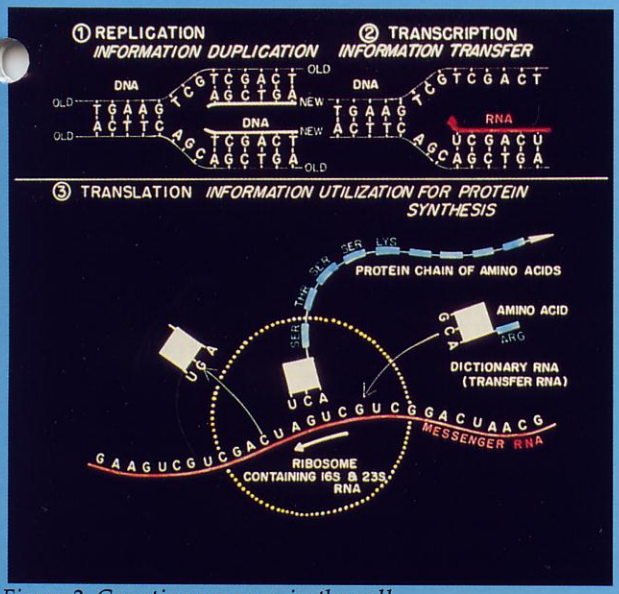



*Figure 2. Genetic processes in the cell.*

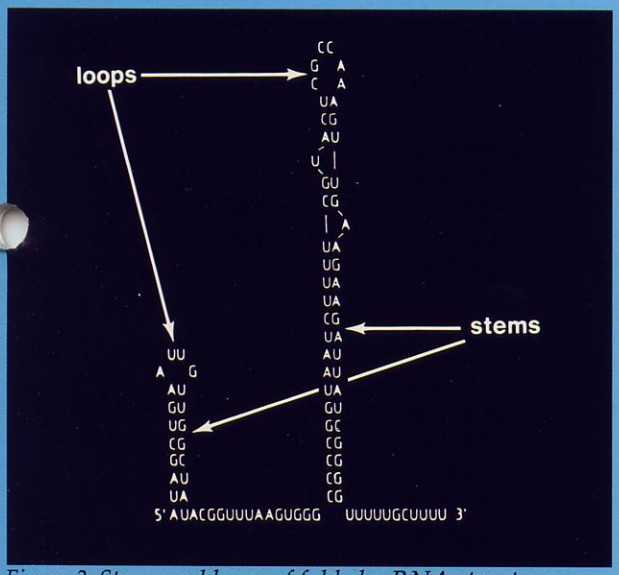



*Figure 3. Stems and loops of folded mRNA structure.*

supports are the structural parts that hold the ladder together.

Each strand of DNA consists of a linear sequence of elements called nucleotides. A nucleotide is composed of a phosphate molecule, a pentose sugar and an organic base, either adenine, thymine, guanine or cytosine. The nucleotides on one DNA strand pair with those on the other strand to form the rungs of the DNA ladder. A fundamental rule of this pairing is that nucleotides containing adenine (A) always pair with those containing thymine (T), similarly, those with cytosine (C) and guanine (G) always pair. Weak hydrogen bonds hold the nucleotide pairs together: two hydrogen bonds for an A-T pair and three for a C-G pair. These hydrogen bonds are easily broken, allowing the DNA to split open lengthwise for its own replication and for the synthesis of mRNA.

## Protein synthesis in the cell

The process of building a protein from its gene involves a number of steps (Figure 2). First, the DNA splits lengthwise at the location of that gene, allowing the cell to manufacture a complementary strand of mRNA containing the molecular code for that particular protein. Messenger RNA is a molecule that allows the cell to transcribe the information in the DNA database without destroying or damaging any of that information. Hence, this process is called transcription. Messenger RNA is much like DNA, though it differs chemically in that it is composed of a slightly different sugar molecule called ribose. In addition, it contains uracil instead of thymine, a chemically similar base.

Once the region of the genetic code that the cell requires has been copied via mRNA, the mRNA leaves DNA and participates in a process called translation. This process involves the deciphering or interpretation of the genetic code. The mRNA, like a piece of recording tape, must be passed over a "tape head" in a linear direction in order for the genetic code to be read. The reading element or "tape head" is called a ribosome and is the cell's protein assembly site. The mRNA molecule binds to the ribosome and rachets through it, moving three nucleotides at a time. These three-nucleotide sequences are called codons. Each codon specifies a single amino acid. Amino acids are joined together in the ribosome in a linear sequence to form proteins.

The key molecule involved in deciphering the genetic code is a "dictionary" molecule called transfer RNA (tRNA). Transfer RNA acts much like a dictionary in that each type of tRNA corresponds to a single codon. Each tRNA has three nucleotides complimentary to every codon and carries a specific amino acid to the ribosome. As each codon is read, the corresponding amino acid is brought into the ribosome by tRNA and chemically linked to the sequence of amino acids to form the protein.

## Production rate factors

Cost-effectively maximizing production of proteins is a primary concern of biotechnology companies interested in producing protein-based commercial products such as hormones or viral vaccines. But since nature need not concern itself with commercial markets, its production lines are not geared to producing commercial quantities. One limiting factor in the cellular process of protein synthesis may be the ease with which a ribosome can read the nucleotide sequence of an mRNA strand. The same hydrogen bonds that hold nucleotide pairs together in DNA cause mRNA to bond to itself, resulting in a molecule with an irregular shape (Figure 3) characterized by loops (areas without paired nucleotides) and stems (areas where the nucleotides have

bonded to each other). This is called the molecule's tertiary structure. Stems and loops appear to be important for regulating gene expression, but they can also serve as "road blocks" to efficient protein production. If ribosomes ratcheting down the mRNA one codon at a time run into an energetically stable stem-loop structure, the ribosomes may stop and protein production will cease at that point.

A challenge to genetic engineers is determining modified mRNA sequences that will display minimal folding in their tertiary structures while preserving the information encoded in the original mRNA nucleotide sequence. Tertiary mRNA structures with minimal folding can significantly speed up protein synthesis by saving ribosomes the time it takes to "unzip" each hydrogen bond of a folded mRNA molecule. Determining alternate mRNA sequences that code for a given gene is theoretically feasible due to the redundancy inherent in the genetic code.

Since there are four "letters" — the bases A, T, G and C — in the genetic alphabet and since all of the "words" in the genetic code — the codons — are composed of three letters, there is a total of 4 x 4 x 4, or 64, words that comprise the genetic code. However, this scheme is redundant because there are only 20 amino acids that need to be coded for. Since each amino acid is coded for by more than one codon, it is possible to interchange certain codons and still specify the same amino acid. This interests biotechnology companies because it presents the possibility of substituting certain codons for one another, via gene splicing, to reduce the folding of mRNA molecules while preserving their orignial "meanings." In other words, genetic engineers can create modified mRNA sequences that will produce the same protein as the original sequence but will increase the efficiency of production due to a decrease in the number and stability of stems and loops that a ribosome will encounter.

## Enter the computer

Molecular modeling via computer can help researchers determine modified mRNA structures that will minimize folding while retaining the molecule's information content. The problem is to determine the molecule's tertiary structure from its primary structure. The molecule's primary structure is simply the one-dimensional sequence of nucleotides of which it is composed. The three-dimensional folding and twisting of these long molecules is described by the laws of chemistry and thermodynamics. The underlying principle in mRNA modeling is that physical systems tend to assume configurations of minimum energy. The computer is used to calculate the thermodynamic stability of all possible combinations of stems and loops until a single secondary structure is predicted which has minimal energy, i.e., the most stable structure. It is assumed that the configuration of lowest energy is the configuration that describes how the molecule will actually fold.

In the case of mRNA the number of possible tertiary configurations is so large and the energy calculation for each configuration so long that biologists have had to settle for predicting only the secondary structure. This is the shape of the folded chain when constrained to lie in a two-dimensional plane. Secondary structures can provide researchers with enough information to make useful decisions about modifying mRNA structures to speed up commercial production of biological products.

Several methods have been developed to assist the molecular biologist in modeling RNA secondary structure. One such method is the dot matrix analysis, which is suitable for a variety of computers including microprocessors. The analysis is fairly rapid but the results are often difficult to interpret and the model gives no information about the energy stability of the predicted molecule. A second method is the stem and loop analysis. This was one of the first methods that incorporated the energy associated with stem and loop structures into the prediction. A third method is the stem completion method which is a relatively new rapid analysis that attempts to simulate the various pathways which an RNA molecule could fold. The stem completion method generates a diverse family of possible structures, but it does not always find the most energetically stable structures.

A fourth method used to predict secondary structure was developed by Paul Stiegler and Michael Zucker of the National Research Council of Canada. Their program, FOLD, uses a technique called "dynamic programming" based on combinatorial mathematics. The program computes the configurations of lowest energy for all nucleotide sequences of length $n$ and then uses that information to compute the configuration of lowest energy for all sequences of length $n$+1. It proceeds from the smallest possible subchains through a recursive buildup to the one configuration of lowest energy for the entire chain of nucleotides.

Recently, Dr. David W. Mount, working at the Department of Biochemistry and Biophysics at the University of California at San Francisco, used FOLD running on a CRAY-1/S, to predict the minimal energy structures of a 359-base RNA strand found in the potato spindle tuber viroid, and 1376- and 1807-base transcripts from the human intestinal bacteria E. coli. The computation times on the CRAY-1/S for these problems were 16 seconds, 15 minutes and 34 minutes, respectively.

"The running times were proportional to $n^{2.7}$, where $n$ is the number of bases in the sequence," explained Mike Ess, an analyst who has optimized the code for the CRAY. "However, those times would be faster by a factor of at least two if we ran them today on the CRAY X-MP, considering improvements in machine and compiler performance and code optimization. But memory is really the bottleneck for this type of problem, and since the CRAY has the

largest memory, it's needed to model the largest molecules."

A commercial firm, Molecular Genetics, Inc., of Minnetonka, Minnesota, has used a CRAY X-MP/48 for mRNA secondary structure analysis. "The CRAY was used to predict the probable secondary structure of several gene designs for the bovine growth hormone (BGH) messenger RNA molecule," explained Dennis Anderson, director of information systems at Molecular Genetics. "Based on the predicted models, our molecular biologists chose an optimal sequence construction for BGH that they believed to have limited secondary structure in the front end of the mRNA molecule yet would produce a bovine growth hormone identical to the native molecule (Figure 4).

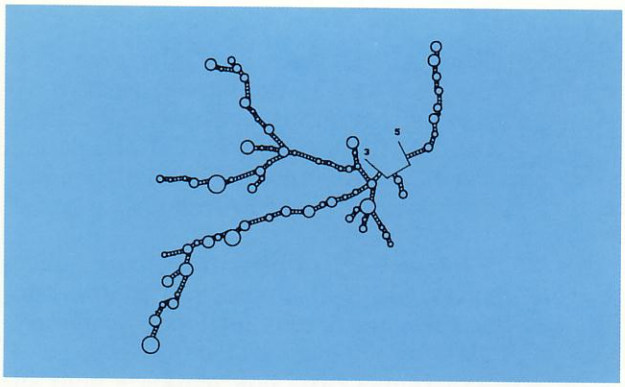



*Figure 4. Calculated secondary structure of bovine growth hormone mRNA.*

The expression of the hormone increased substantially using a gene sequence that limited secondary structure and utilized preferred codons. With the original sequence, production was so slight, we could detect it only by tracing it with radioactive isotopes. With the modified sequence, the production rate increased by factors of 1000 to 10,000. Using the CRAY to predict the secondary structure of BGH was a real advantage in that the design possibilities were limited to a manageable number that had a significantly higher probability of expressing high amounts of the hormone."

Molecular Genetics also used the CRAY to apply analysis of RNA secondary structure to the study of the Rift Valley Fever Virus (RVFV). RVFV is a zoonotic virus that causes death in humans and animals and is a significant health problem to populations living in the Middle East. Unlike mammals or bacterial cells, the genetic material of this virus is composed of a single strand of RNA. The virus' genetic material, or viral genome, is known to form a circular structure that is closed by a double-stranded "panhandle" region at the end (Figure 5). It is believed that this panhandle structure is the result of two regions of the RNA molecule binding to itself to form a duplex region resembling that found in DNA. Although biochemical evidence indicated the existence of the double-stranded region in RVFV, no sequence data was available until recently. Scientists at Molecular Genetics determined the 3884 nucleotide sequence of RVFV and subjected it to secondary structure analysis on the CRAY.

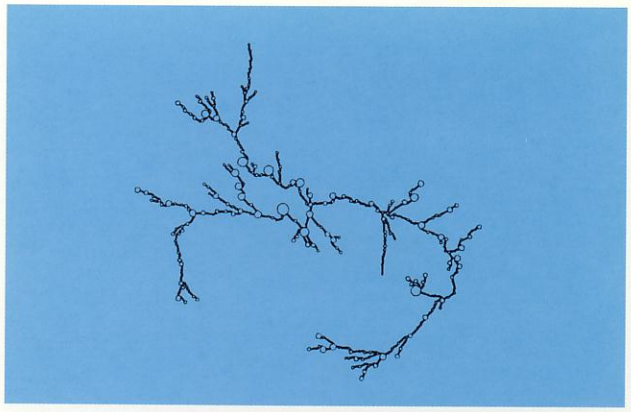


*Figure 5. Calculated secondary structure of Rift Valley Fever Virus RNA.*

"The computer model revealed a double stranded region composed of 51 nucleotides in which the beginning of the molecule is paired with a complimentary region at the very end of the viral genome," explained Anderson. "To our knowledge, this is the largest RNA sequence ever analyzed via the Stiegler and Zucker algorithm. It required between five and six hours of CRAY CPU time to calculate the most energetically stable structure and it is the first report of a proposed model for the genome of a bunyavirus like RVFV. Although the validity of the prediction remains to be determined by further biochemical and genetic tests, this model of RVFV has been useful in the design and execution of additional experiments to understand the molecular biology of this virus."

## Conclusion

Commercial biotechnology promises to improve production methods for valuable medical and agricultural products, thereby greatly increasing their availability. Biotechnology also promises eventually to design novel products through techniques such as protein engineering. But to reap the full benefits of this new technology, high-speed scientific computers like the CRAY are needed. Supercomputers make practical the modeling of the large macromolecules, like mRNA, that biotechnology manipulates. As additional predictive models are developed and refined, they will provide the bench scientist working in commercial biotechnology with the same type of CAD/CAM capabilities employed in aerospace, electronics and other engineering industries. Computer modeling of biomacromolecules in a practical time frame is an important expedient to realizing biotechnology's vast potential benefits. □

**Acknowledgement**

*Special thanks to Dennis Anderson at Molecular Genetics, Inc. for his help in preparing this article.*

# Versatility enhanced with Cray Pascal

To increase the options of CRAY computer users in their choice of programming languages, Cray Research released a Pascal compiler in the Fall of 1983. An updated version with added extensions was released in September 1984.

Pascal was first drafted in 1968 by Niklaus Wirth, a professor at the Federal Institute of Technology in Zurich, Switzerland. Wirth intended the language to be used as a tool for teaching programming skills. Its rigid grammar and strict syntax forced students to write programs that were well-structured and well-organized.

But Pascal has since gained a wider audience. Its versatility and straightforward logic have brought it into contention with C language for popularity among the next generation of computer users. It is a language particularly suited for use in writing systems development programs such as utilities, tools and compilers. Debugging of programs written in Pascal is relatively simple since the language's control structures are rich enough to avoid extensive use of GOTO statements. In addition, a considerable amount of error checking is performed during compilation, allowing programmers to spot errors quickly. These features have all contributed to spreading Pascal's appeal.

## The Cray variety

Work on the Cray Pascal Compiler was initiated at the University of Manchester in the United Kingdom and completed at Cray's Mendota Heights, Minnesota facility under the direction of Cray's Software Development group. Cray Pascal is a fully supported Cray product and will be used as the implementation language for the new Cray FORTRAN Compiler and some other software products.

"Pascal was chosen for the new FORTRAN compiler and the new version of the Cray assembler because of its many clean programming features," explained Karen Spackman, Manager for New Compiler Development. "It has a good structuring capability, a nice way of identifying records and a strong typing capability, which are all useful in error checking."

Cray Pascal complies with the Level 1 requirements of standard ISO 7185, defined by the International Standards Organization (ISO), with some extensions and restrictions. "Some of the extensions were added to make it easier to use Pascal as an implementation language and to help in handling large programs," commented Spackman. "We also include some common Pascal extensions." The compiler will issue messages identifying these extensions to help transport a program to a machine running a different implementation of the language.

The Cray Pascal Compiler is invoked by the PASCAL control statement. The user selects compiler parameters either explicitly by listing them on the control statement or implicitly by accepting the default values. The result of the compilation is relocatable code used as input to the relocatable and segment loaders.

Input and output statements move data between local files or external devices, such as disks, and data structures within a Pascal program. Pascal does not directly support tape I/O; however, FORTRAN-callable library routines to perform tape I/O can be invoked.

The Cray Pascal Compiler accepts two types of compilation units: program and module. A module is a stand-alone routine, such as a library routine, that can be accessed and executed by other modules and programs. A module allows the separate compilation of encapsulated code and data.

Generated code performance has been improved for the second release of the compiler, and research has already begun on future enhancements. "Right now we're working on further increasing the speed of the generated code as well as adding vectorization," said Spackman.

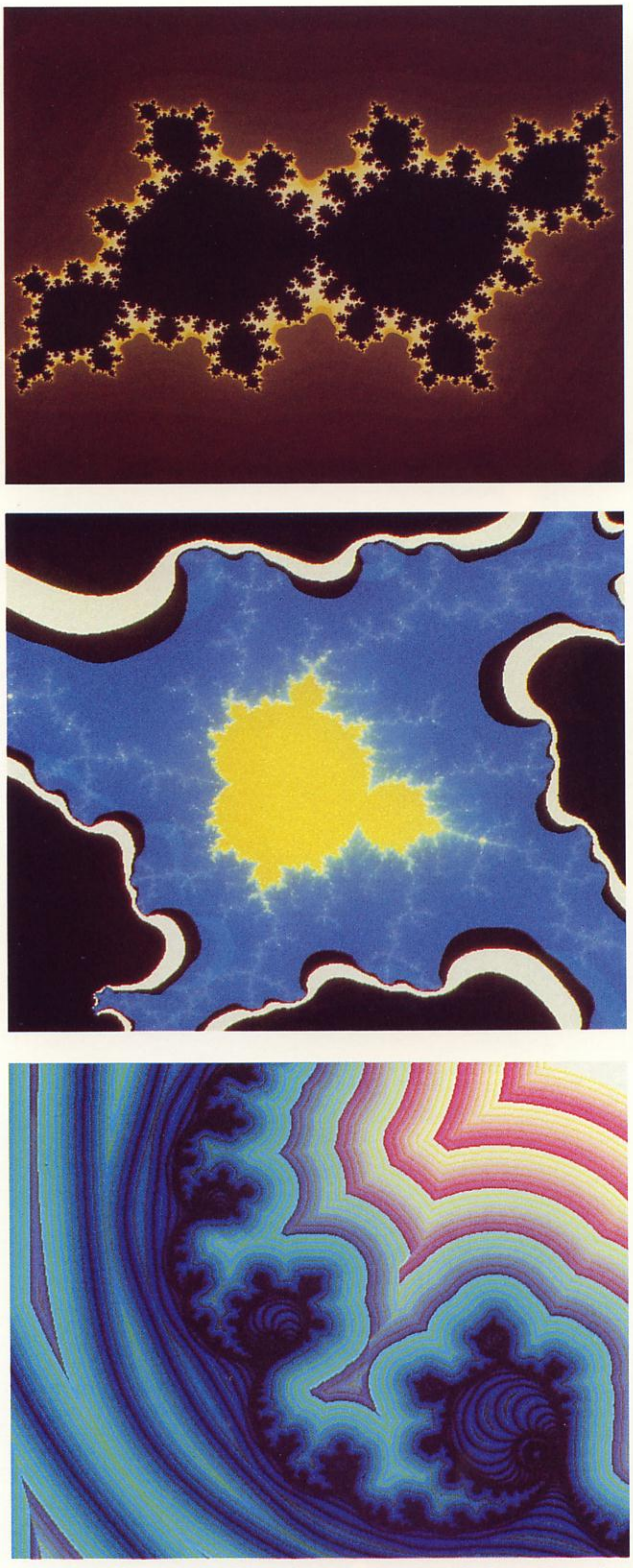


*Figure 1. Geometric fractals generated with Cray Pascal.*

## Cray Pascal-generated fractals

Along with aiding in software development, the Cray Pascal compiler has lent itself to some interesting applications. An example is a collection of geometric fractals (Figure 1) generated using a Pascal code running on a CRAY X-MP. The algorithm was written by Greg Turk, while researching computer graphics at West Coast University in California, and the program was adapted to the Cray by David Coons of the Jet Propulsion Laboratory at the California Institute of Technology. Fractals are generated by successively inputting the product of a mathematical function into that same function and graphing the results. Points put through this iterative process either move away from their origin continually and are said to diverge, or remain within a given region near their origin, in which case they are said to be bounded. The complex patterns shown here resulted from iterating the function F(z)=lambda * z *(1-z), where lambda is a complex number. Different colors were used to label diverging and bounded points.

The original fractal program was written to run on a VAX 11/750 coded in VAX-11 Pascal. "My original plan was to convert the code to FORTRAN to run on the CRAY because I knew FORTRAN was implemented well and I didn't expect Pascal to be," explained Coons. "But when I found out how well-supported Pascal was on the CRAY, I decided to stay with it. The final fractal algorithm tallied to about 3600 lines of Pascal source, took under eight X-MP CPU seconds to compile, and ran between 60 and 100 times faster in scalar mode than the original program running on the VAX 11/750."

The goal of getting the program on the CRAY was to demonstrate the Ramtek 9465 graphics display device tightly coupled to the CRAY X-MP for the 1984 ACM/SIGGRAPH conference. The final Pascal program was used to generate high resolution (1024 x 1280 pixel) images, save them to disk files, and later load them back into the graphics display at high speed. The program could be used in either batch mode or interactive mode, depending on the available display devices.

## The future

The Cray Pascal Compiler promises to broaden the range of other software products available to CRAY users. Currently, Network Systems Corporation (NSC) is planning to convert its NETEX™ data communications software system to Cray Pascal. NETEX is an access method that allows high-speed host-to-host communications between like and unlike mainframes. It is offered as an adjunct to NSC's HYPERchannel, an adapter for connecting mainframes, minicomputers and peripherals. "We chose Pascal for NETEX because Pascal is about the only language that has sufficient ability to create data structures and to define a communications protocol," explained NSC marketing representative Terry Mikkelson. "And it gave us access to the widest range of computers and computer operating systems."

From its versatility and straightforward logic to its extensive error checking, Pascal is a language whose time has come. And now, the availability of Pascal on the CRAY provides new opportunities for all CRAY computer users. □

**Cray Research extensions to the ISO Level 1 Pascal standard include the following:**

- OTHERWISE label specifies the action to take when no other label in a CASE statement is selected.
- MODULE compile units allow the user to define Pascal procedures, functions, and data without a PROGRAM module.
- IMPORTED and EXPORTED procedures permit the use of previously compiled modules and FORTRAN or CAL subroutines.
- IMPORTED and EXPORTED declarations allow sharing of variables between compile units.*
- COMMON declarations allow FORTRAN common blocks to share data across compile units.*
- STATIC declarations allow a local variable to keep its value between calls to a routine.*
- VALUE definitions initialize variables at compile time.*
- VIEWING statement allows a variable to be used as a different type.*
- EXTERNAL and FORTRAN procedure and function directives.
- Expansion of the predefined procedure and function list to include the following:

— BAND, BOR, BOXR and BNOT are bit-string Boolean functions that accept integer arguments and return integers.
— LSHIFT and RSHIFT shift an integer argument left or right by a specified number of places.
— SIZEOF gives the size of a dynamic variable.*
— LOC returns the address of a variable.*
— LOG calculates the common logarithm.
— TAN calculates a tangent.
— ARCSIN and ARCCOS calculate the inverse of the sine and cosine.
— SINH, COSH and TANH are the hyperbolic functions.
— HALT terminates the execution of a program when encountered.
— CONNECT associates a COS local dataset with a PASCAL file.

- Octal numbers can be used in integer form.
- ALFA is a predefined type specified as an eight-member packed array of characters.
- I24 is a predefined 24-bit integer data type.
- Special characters ($ % @ ) are used in identifiers.

**Cray Pascal includes the following restrictions to the ISO Level 1 Pascal standard.**

- The commercial "at" sign (@) cannot be used as a substitute character for the circumflex (^), as specified by the standard. The @ is implemented as a valid character in an identifier.
- The maximum line length for a text file is 140 characters. The ISO standard does not impose restrictions on the line length of a text file.
- Some errors specified by the ISO standard are not detected. Cray Pascal documentation details these errors.

*added in latest release

## What is Pascal?

A descendant of Algol 60, Pascal is a high-level, general purpose language that emphasizes the virtues of structure, simplicity and portability. Using it, a programmer can implement algorithms and data structures in a high-level, machine-independent manner without sacrificing efficiency.

Pascal is a block-structured language with programs consisting of two parts: a heading which names the program and specifies which local datasets will be used, and the body of the program, called a block.

Blocks consist of six sections. The first four state the labels, constants, data types and variables. Labels identify statements for future reference. Constants equate numbers with names, e.g. pi = 3.14. Data types are many and structured ones can be defined to include sets, records, arrays and files. Variables must be followed by a type designation.

The fifth section of a block names a procedure or function. Procedures can be placed within each other, and each procedure must be preceeded by the keyword "begin" and followed by "end." The sixth section of a block contains the executable code for the given procedure or function. Arithmetic, relational and logical operators are defined, and numerous control statements are allowed.

# CORPORATE REGISTER

## New COS and CFT versions released

The 1.14 versions of the Cray Operating System (COS) and the Cray FORTRAN Compiler (CFT) and associated libraries and utilities were released in January 1985. This release provides support for new Cray hardware and offers improved performance, enhanced functions and several new software products.

The 1.14 COS and CFT software allows user access to the following new hardware products and features:

- Larger memories — up to 8 million words for mainframes and 128 million words for the Solid-state Storage Device (SSD). Extended Memory Addressing (EMA) is supported in all language processors (CAL, CFT, and Pascal). All standard Cray software can be built to execute on CRAY X-MP systems with programs up to 8 million words of memory.
- Four processors on the CRAY X-MP/48 — a single job can be multitasked on four processors.
- Two 1000-megabyte-per-second channels to the SSD (CRAY X-MP/48 with 128-million-word SSD)
- Gather/scatter and compressed index instructions — CFT compiles the new gather/scatter and compressed index instructions where available and when appropriate.
- The Hardware Support Monitor (HSM)
- DD-49 Disk Drives

In addition to the new hardware support, the 1.14 software release includes enhancements that can significantly improve system performance. The new CFT can decrease execution time with improved instructions for exponentiation, two-version loops, loop unrolling and conditional loop code. With the new version of COS, system call overhead and I/O overhead can be reduced.

This release also includes the following enhancements:

- CFT will now optionally flag statements which do not conform to the ANSI FORTRAN 77 standard.
- CFT and the libraries support an extended character length variable of up to 16,383 characters.
- CFT and the loaders (LDR and SEGLDR) recognize a new type of common block (task common) that is allocated separately for each task in a multitasking job.
- CFT supports recursion in stack mode; that is, SUBROUTINES or FUNCTIONS will be able to call themselves.
- COS with an I/O Subsystem now supports several on-line magnetic tape enhancements. These include partial implementation of IBM multifile capability, user processing of end-of-volume, read/write ring specification, multiple tape mark processing, parallel tape mounting and read/write of very large tape blocks.
- User programs can specify the next control statement to be executed.
- User jobs can send and receive messages to and from other jobs.
- User jobs can suspend themselves until an event (for example, a message from another user job) occurs.

Several new software products are included in this release. They include:

- A sort/merge package available through calls to the library
- The Kernighan and Plauger Software Tools
- A Table Diagram Generator to generate the *COS Table Descriptions Internal Reference Manual*. This allows sites with local table modifications to have the manual include their changes.
- AUDPL to catalog UPDATE program libraries
- SKIPU to position unblocked datasets
- FTREF to cross-reference FOR-

TRAN common block usage and to produce an application's static calling tree.

## New MVS and NOS/BE versions released

The 1.13 versions of the IBM MVS and the NOS/BE Software Service packages were released by Cray Research in January 1985 and December 1984, respectively.

Major features of the new IBM MVS Software Service release include:

- Support in an MVS/JES2 SP1.3.4 environment. A new JES2 modification CRFDO6 is provided.
- The station generates System Management Facility (SMF) records at various points during its processing. An installation can select the records to be written to the SMF dataset.
- Support is provided for Generation Data Groups (GDGs) on transfers to and from COS.
- The MVS format restrictions on dataset transfers to and from COS transparent datasets are removed.
- Segment sizes up to 28K bytes are now supported. The segment size can be selected by parameter in the Link Options Table. This feature can improve both transfer rates and CPU utilization.
- Datasets migrated by the IBM Hierarchical Storage Manager (HSM) are explicitly recalled by the station before dynamic allocation. This should reduce the contention during dynamic allocation.

The NOS/BE Software Service supports the following new features:

- Binary blocked and binary deblocked transfer modes; a binary blocked dataset contains COS control words and is specified by using the DF=BB parameter; a binary deblocked dataset does not contain control words and is specified by using the DF=bb parameter. In the DISPOSE, ACQUIRE and FETCH statements, the BD parameter is the same as the TR parameter. Character conversion does not occur.
- MULTI status to the CSTATUS station command; a new job status which states that the COS job is a multitasking job.
- Page numbers for station displays; the page number appears in the upper right corner of each pageable display.
- Enhanced error processing, including station auto-relog capabilities and more informative error messages.
- CSTATUS display of all queue entries and queue count.
- Numerous performance improvements including speed up of transparent transfers and coded transfers.
- Alternate output file specification and optional repeat parameters allowed on CSTATUS and CJOB.

## Cray hosts Washington seminar

Bruce Kasson, Cray Research's vice president for government marketing, in cooperation with Cray's Eastern region office, hosted three executive seminars on supercomputer technology in Washington, D.C. in mid-January. The half-day briefings were designed to update senior government and military officials on the supercomputer industry today. Each presentation was tailored to educate those in the audience who were not overly familiar with supercomputer technology, as well as to update those with some background knowledge.

Marcelo Gumucio, Cray's executive vice president for marketing, opened the seminar with a brief overview of the company, noting that the U.S. Federal government is Cray's largest customer. Bruce Kasson reviewed the development of supercomputer technology and Les Davis, executive vice president for engineering and development continued with a discussion of technology development at Cray. Derek Robb, manager of marketing development, and Carl Diem, manager of benchmarking, familiarized attendees with the wide variety of customer applications of CRAY computers, illustrating their presentations with numerous CRAY-generated graphics and a short CRAY-generated video. Bob Ewald, director of commercial marketing, concluded the seminar with a discussion of price/performance issues, government procurement strategies and his experiences in directing the Los Alamos computer facility.

Over 125 participants from more than 25 agencies attended the seminar, including representatives from the U.S. Army, Navy and Air Force, the White House Office of Science and Technology Policy, the Departments of Commerce, Energy and State, the National Science Foundation and the National Aeronautics and Space Administration (NASA). Additionally, a number of executives from various Department of Defense contracting companies attended. Because of the positive comments regarding the quality and value of the session, additional briefings will be planned for the Washington area.

"We were able to expose some important people to the potentials of supercomputing," said Dick Harris, marketing representative for Cray's Eastern region. "Often these people get requests or recommendations for supercomputers, but they don't really appreciate a supercomputer's capabilities. We got some very favorable feedback from the attendees and some important exposure."

## Exhibit honors Seymour Cray

Computers, computers everywhere. That was the feeling at The Computer Museum's grand re-opening in Boston last November. The museum, housing the world's largest collection of historic computing

equipment, moved to Boston after outgrowing its previous site in Marlboro, Massachusetts. The reopening featured exhibits highlighting the modern era of computing, including a tribute to Seymour Cray.

"We wanted to feature a personality important to modern computing," explained Gwen Bell, the museum's director. "There aren't that many people you could call heroes in the field, but Cray was an obvious choice."

The Cray exhibit, ***A Man and his Machines: Seymour Cray's Contributions to Computing,*** traces Seymour Cray's career as a computer designer from 1950, and his work with Engineering Research Associates on the ERA 1103, to his current work on supercomputers with Cray Research. "The ERA 1103 was really the first scientific machine," notes Cray in a videotaped 1974 lecture that accompanies the exhibit.

Examples from several machines Cray had a hand in designing are displayed, including parts of the Naval Tactical Data System from 1957, the Little Character (a prototype of Control Data's first computer series), and a CRAY-1 supercomputer.

"Highlighting an individual this way helps to demystify things," Bell noted. "You can come in and see that someone really went to high school (the exhibit includes a photograph of Cray from his high school yearbook), that they're a flesh-and-blood person. And when you look at the machines you can see what 'computer architect' means. The continuity in design can be seen in a way that looking at pictures just doesn't get across."

The exhibit concludes with a posting of Cray quotes that captures his unique philosophy of computer design ("There has never been a good product designed by a committee.") and his sometimes self-effacing style ("They certainly aren't supercomputers. They are kind of simple, dumb things.").


*CRAY-1 module under glass at The Computer Museum.*

The Computer Museum's other current exhibits include the Air Force's 25,000-vacuum-tube SAGE (Semi-Automatic Ground Environment) system, which ran continuously from 1958 to 1983, and interactive PC exhibits demonstrating graphics and voice synthesis.

"Our purpose is really to preserve computing technology and to show things that are bigger than life in a friendly and interesting way," said Bell. "Other industries have their museums, and we think it's important that computing have its history preserved too." The Seymour Cray exhibit will be displayed for about a year, after which the museum will highlight "a software giant," Bell said.

## Rockwell orders CRAY

Rockwell International Corporation recently ordered a CRAY X-MP/12 computer system to be installed in the second quarter of 1985 at Rockwell's Information Systems Center in Seal Beach, California. The system will be used by Rockwell engineers and scientists to perform complex aerodynamic and structural analyses. Rockwell said the major benefits to be realized from the CRAY system are faster turnaround, improved design accuracy and reduced prototype and experimental testing.

## LLNL to get two CRAYs

In December, Cray Research announced an order for its two newest and largest computer systems, the CRAY-2 and the CRAY X-MP/48, by Lawrence Livermore National Laboratory (LLNL). The laboratory's National Magnetic Fusion Energy Computer Center (NMFECC) will receive the CRAY-2 in the second quarter of 1985. It will replace a prototype CRAY-2 with a single central processing unit currently installed at NMFECC. NMFECC provides large-scale computing services to researchers in magnetic fusion and energy research at 150 laboratories and universities across the nation.

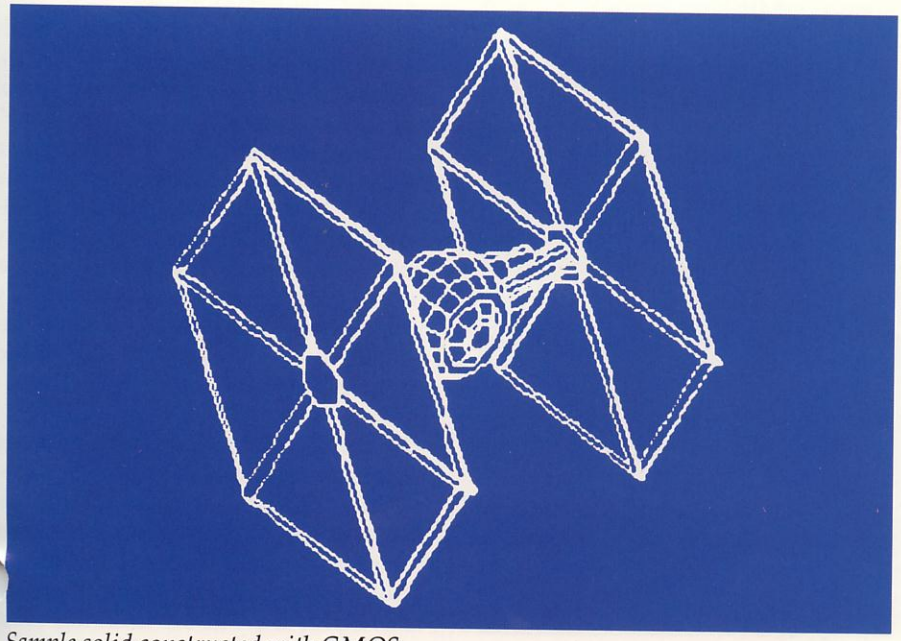
The CRAY X-MP/48 was recently installed at the LLNL computer center, which is operated by the University of California for the U.S. Department of Energy. More than 4000 users are linked to the national center by satellite and land lines.

# APPLICATIONS IN DEPTH

## Solid modeling primitives on the CRAY

A versatile general-purpose geometric modeling program, GMOS (Geometric Modeling of Solids), is now available on CRAY machines.

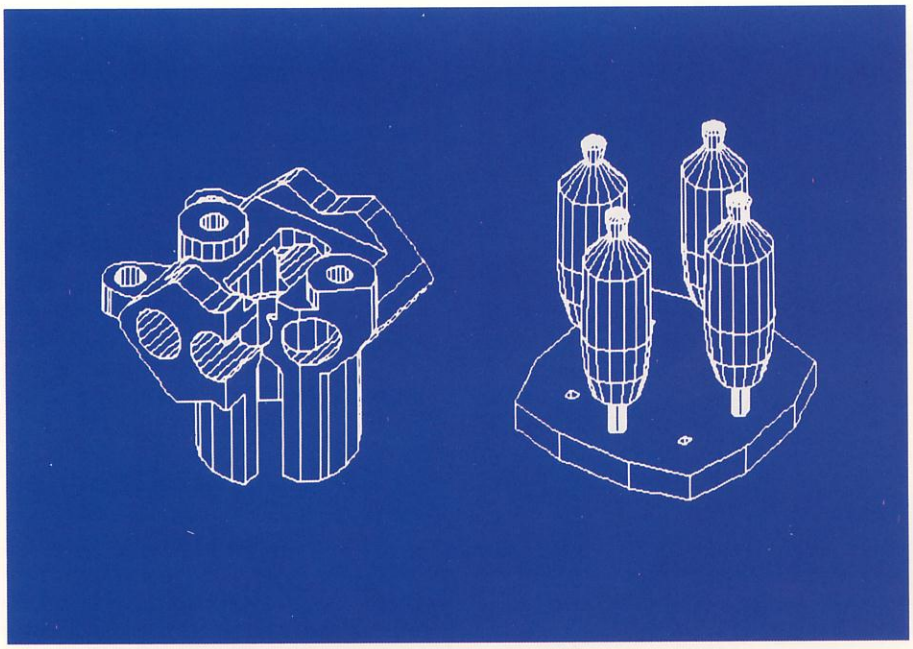
GMOS is a general-purpose interactive processor for the construction and display of three-dimensional solid models. By the use of either a screen menu or simple keyboard commands, users can construct complex models by creating primitive objects through the MAKE command and manipulate these objects with the BOOLEAN operations. Other commands allow such features as three-dimensional view transformations, hidden line removal and the calculation of mass property information.



*Sample solid constructed with GMOS.*

GMOS gives users the capability of creating primitive objects such as blocks, cylinders, cones, spheres, tubes, objects of extrusion and objects of revolution (both with arbitrary cross-sections that may be described by keyboard input or cursor/tablet input). The most powerful single operator is the Boolean operator. This is actually a set of four operations, allowing two objects to be joined, the first object to cut the second, the second to cut the first and the two objects to intersect.

Various view operations may be made to allow an object to be viewed from any position desired. It is also possible to see several views of one object at one time using the MULTIVU command. In addition,



*Sample solids constructed with GMOS.*

the surface area, volume, weight, mass moments of inertia and principle moments of inertia can be obtained for any object generated by GMOS.

For more information on GMOS on the CRAY contact: JML Research, Inc., 5713 Crabapple Lane, Madison, WI, 53711; telephone (608) 274-2524.

## Large 3-D FFT now feasible

The computation of large three-dimensional Fast Fourier Transforms (FFTs) has long been a problem generally sidestepped because it was not feasible on conventional computing equipment. But in 1983, Cray analysts successfully computed a three-dimensional FFT on a 1024 x 1024 x 1024 sample data set in an elapsed time of 55 minutes on a CRAY 1/S using approximately 90 percent capacity on eight DD-29 disk units. Last year, Cray analysts ran a multitasked version of the same code on a CRAY X-MP/48 with the Cray Solid-state Storage Device and eight DD-49 disk units. This last demonstration ran in an elapsed time of less than eight minutes, an increase in performance by a factor of 14 over the previous year's, considering it included a forward three-dimensional FFT followed by an inverse three-dimensional FFT, representing twice the computational work of the earlier demonstration. This increase in performance establishes the feasibility of performing large three-dimensional FFTs on the CRAY X-MP/48.

FFTs are computational tools for breaking down complex functions into their component frequencies. Performing computations in the Fourier domain generally requires significantly less time than equivalent computations in the spatial or spatial-time domain. FFTs can be used in virtually every mathematical computational area. One of the most common uses is in the extensive seismic data processing conducted by petroleum companies and geophysical contractors.

The 1024-cubed FFT demonstrated in 1983 required three data passes. The first pass computed a two-dimensional FFT, the second pass transposed the second and third storage indices and the third pass computed the final one-dimensional FFT. Essentially the same method was used in the most recent demonstration, but with several novel programming strategies included to take advantage of the CRAY X-MP/48's multiple CPUs and the SSD's high-speed data transfer capability. In this demonstration, forward and inverse transforms required three data passes, with transposition passes eliminated by using the SSD.

"We used a method we call 'microtasking', which is a way of multitasking to deploy four CPUs on one problem," explained Mike Booth, an analyst who worked on the new code. "We also used asynchronous queued I/O that significantly reduces system overhead to transfers into and from the SSD." (Microtasking differs from standard multitasking in smaller task overhead or start-up and task granularity or time duration. Asynchronous queued I/O initiates multiple I/O calls with a single operating system call.)

Achievements like the latest three-dimensional FFT calculation invariably introduce new goals. "Demonstrating forward and inverse 1024-cubed FFTs in less than eight minutes is a mandate to the scientific computing community for the establishment of four-dimensional FFT computation as a new goal," commented Mickey Edwards, a Cray analyst who worked on the code. "Computation on a 1024-quadrupled data set is obviously not possible today, but, for example, a 1024 x 1024 x 64 x 64 data set only represents a fourfold storage increase over the current 1024-cubed problem."

Establishing the feasibility of large three-dimensional FFTs also establishes the feasibility of many two-step three-dimensional algorithms which process all constant y planes followed by all constant x planes. Examples include various migration algorithms with and without FFTs.

# USER NEWS

## CRAY helps explain Namoratunga II

Remember the long hours in college spent cramming for that archaeoastronomy final? Probably not. Archaeoastronomy is a relatively new science, being the study of ancient systems of astronomy. But it's not so new that it hasn't already benefited from a CRAY computer.

Using a CRAY X-MP at the NASA Ames Research Center, Dr. Laurance Doyle of Information Management International, Inc., has calculated the probable function of Namoratunga II, an arrangement of stone pillars recently discovered at an archaeological site in northern Kenya. Like the Stonehenge and Avebury sites in Britain, Namoratunga II is a highly organized arrangement of megaliths set up by ancient people for an apparent purpose. But that purpose is unknown and, as with the Stonehenge and Avebury sites, the topic of much debate.

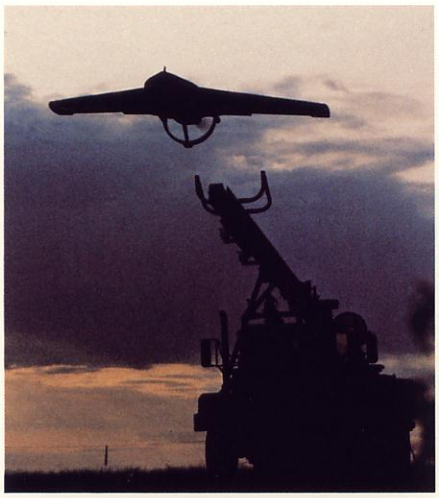
Based on his computed results, Doyle proposes that the site, dated at 300 B.C., was the basis for a lunar calendar that is today used by the Borana people of southern Ethiopia. The Borana Calendar relies solely on lunar phase observations in conjunction with seven specific stars. Recent measurements of the site found 25 two-pillar alignments with the horizon rising positions of the seven calendrical stars, taking into account the Earth's precession since 300 B.C.

Doyle evaluated the possible calendrical function of the site via statistical analysis. "We wanted to calculate the probability of obtaining 25 or more alignments with seven random positions on the sky using the horizon rising alignments found with the nineteen pillars at the site," explained Doyle. "We wrote a program to generate seven random numbers between -90 and +90 degrees (horizon rising positions from the equator) and compared this with the pillars' horizon alignment positions."

The number of alignments found for each such experiment was tabulated and the experiment was run 10,000 times. The results showed that the probability of getting 25 or more random alignments was 0.0041, or about three standard deviations from the mean. "The 0.41 percent chance that the alignments were accidental makes us think it is very likely that the pillars were used specifically to align with the 300 B.C. positions of the Borana Calendar stars," concluded Doyle, who said he plans to submit his findings to the journal *Archaeoastronomy.*

"Using the CRAY was a little like hunting flies with a bazooka," said Doyle. "Then again, the random number generator was useful and the available routines on the CRAY were the best for my application. Of course, I appreciated the CRAY's speed." Doyle noted that if Namoratunga II is an ancient astronomical observatory, then it is the first archaeoastronomical site to be discovered in sub-Saharan Africa.

*Stones at the Namoratunga II site. Photo courtesy E. William Frank.*



*RPV being deployed.*

## CRAY helps net RPV

Pilots flying reconnaissance missions for the military have the unenviable job of crossing enemy lines to survey enemy activities. Once that task is complete, they must concern themselves with achieving a safe return, something which they, unfortunately, do not always accomplish. To minimize endangering human pilots, the U.S. Army and Lockheed are developing designs for a remotely piloted vehicle (RPV). The RPV is equipped with surveillance equipment and transmitters for sending real-time video images of enemy installations and activities back to a control station. Most significantly, the RPV will be unmanned: its direction and speed will be controlled by pre-programmed instructions. Lockheed has developed a test flight recovery simulation code to run on the CRAY and is using it to help design the RPV.

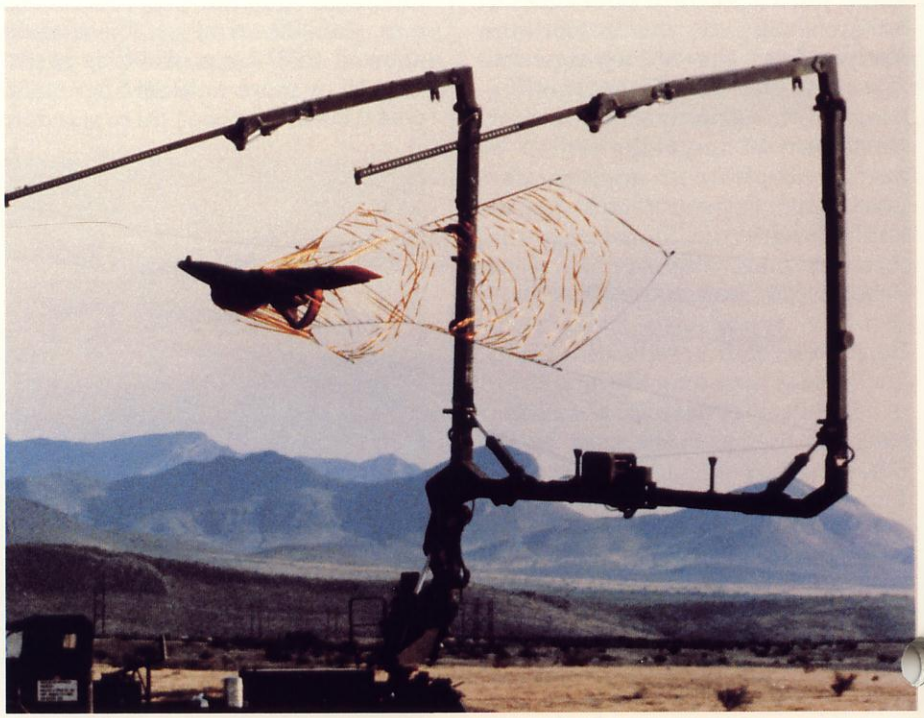
Specifically, Lockheed's code simulates the impact of the RPV flown into a net. Nets are used to recover the unmanned vehicle since it is not equipped with brakes or landing gear. "We needed a simulation of the force that the net's straps apply to the wing," explained Jim Nicholson, advanced systems engineering specialist at Lockheed's Austin, Texas division. "We wanted to know what loading force is being applied to each point on the wing during each of the 700 milliseconds or so it takes to stop a plane. We thought we'd try to come up with a simulation rather than risk breaking a lot of wings."

The RPV has a 13-foot wingspan and weighs approximately 200 pounds. It is intended for use in a tactical environment where it will have to be launched and quickly recovered in a relatively small area. The code Nicholson and his collaborators developed to simulate the RPV recovery has three basic parts; it simulates the plane in flight, the stretching and shaping of the net around the RPV and the dynamics of the net's braking system.

"Simulating the first part, the RPV's flight dynamics, isn't that complicated," Nicholson said. "There are six degrees of freedom: the x, y and z coordinate axes and the rolling, pitching and yawing of the plane. There are two coordinate systems, one fixed on the plane and one fixed on the ground, so the code has to perform matrix coordinate transformations back and forth between them."

The second part of the code is the most time consuming. The force balance calculations needed to describe the deformation of the net can involve thousands of iterations. "Each time the equilibrium for a particular point on the net is calculated, it can throw off other points that have already been calculated," explained Nicholson. "So the computer has to keep sweeping over a row of points until it finds where they all balance out within our required range of precision. Then it feeds the forces back into the simulation of the flying RPV, lets it fly for another millisecond, sees what it does, then goes through the force balance calculations again. The CRAY is an excellent machine for this kind of repetitive calculating."

The third part of the code simulates the net's braking system. The four corners of the net are attached to steel cables which are wound around a rotating drum positioned behind the net. Brake shoes hold the drum stable and keep it from rotat-



*RPV and recovery net.*

ing unnecessarily. "This was the hardest part to model," said Nicholson." The equations governing aerodynamics and the net dynamics are pretty well understood, but we really ended up modeling this to match our results. It was hard to model for a number of reasons. For example, the amount of cable that unwinds and the rate at which the drum turns depend to some extent on the elasticity of the cable. We had to consider three elasticity coefficients, one for the cable and one each for the vertical and horizontal straps of the net."

The flight recovery simulation code is currently running on a CRAY-1/S System at Lockheed Integrated Computer Services in Sunnyvale, California.

Development of the remotely piloted vehicles is currently in the testing stage. Additional testing needs to be done to see if the RPV can handle varying recovery conditions. Because of the CRAY's speed it is practical to run simulations for many different conditions. "For example," Nicholson said, "we can run it to see what would happen if a number of straps bunch up on one point on the wing or if we change speed or change the material the net is made of. The RPV must be able to be recovered with a crosswind of so many miles per hour, and that's another thing we can simulate."

Testing of the RPV will continue for some time yet, but, without the help of the CRAY, the recovery simulation might never have gotten off the ground.

## CRAY BLITZ does it again

What was once a contest last year became no contest. After victory at the 1983 World Computer Chess Championship, CRAY BLITZ went on in 1984 to pick up the Association for Computing Machinery (ACM) computer chess championship.

The ACM contest pitted the top 14 chess programs in the country against each other. "After three rounds we had it clinched, even with one round left to go," said Bob Hyatt, CRAY BLITZ's developer and Chief of Systems at the University of Southern Mississippi. "Nothing in the world comes close to the speed we've got now, not even the special-purpose machines."

In 1982 CRAY BLITZ tied BELLE, Bell Laboratories' special purpose chess-playing computer, for the ACM championship. Running on a CRAY X-MP the following year, CRAY BLITZ unseated BELLE at the World Computer Chess Championship. Last year, a multitasked CRAY BLITZ running on an X-MP/48 also stole the ACM championship, held in San Francisco.

The multitasked version of CRAY BLITZ can calculate from eight to nine moves ahead during its turn, depending on the number of pieces on the board. That ability requires analyzing about 100,000 possible positions per second, or 6 to 7 million each minute. During an end game, CRAY BLITZ can calculate up to 45 moves ahead.

"There are really two schools of thought on how to approach computer chess." explained Hyatt. "The 'brute force' approach we use takes advantage of the speed available on a computer like the CRAY. It just calculates every legal move and checks its outcome. The 'selective search' approach, on the other hand, tries to emulate human thought and so saves time by rejecting obviously bad moves from the start. But now the CRAY is so fast that our brute force method is getting the same results as selective search."

Hyatt explained the multitasking procedure used on CRAY BLITZ: "To make use of the four processors, we redesigned the tree search algorithm to make use of a parallel search of the game tree. The CFT multitasking primitives (TSKSTART, TSKWAIT, LOCKON, LOCKOFF, EVWAIT, EVPOST, and others) are easy to understand and to use. The reentrancy feature of CFT also works well and causes no problems. The actual multitasking algorithm causes many problems as the testing, debugging, and trying to reproduce a bug make life very interesting. Simpler algorithms for straight number crunching are much easier to implement and debug.

"We chose to implement multitasking in the most effective way we could envision, where each processor starts searching a portion of the game tree and all processors communicate with each other relative to best move(s) found, best/worst scores, etc. Our initial approach followed research done at McGill University and the University of Alberta. However, they are both using networks of multiprocessors, not a tightly-coupled symmetric architecture like the X-MP. We are now diverging from their research path since we have a much better architecture to work with in terms of ease of use, low communication/synchronization overhead and ultra-high computation speed.

"The most critical problem we have encountered and are still working on is load balancing among the processors. If one processor is given more work to do than the other three, there will be a time interval when it is the only one doing anything and the other three are waiting on it to complete its task. This reduces overall throughput and can sometimes be quite significant. In San Francisco, we carefully kept statistics on the first game and found that from the potential factor of four available from four processors, we were able to average 3.6 processors busy for the entire game. Our algorithm is very good about avoiding duplicate work between two or more processors, so we essentially saw the program run 3.6

times faster than with one processor. Further testing has confirmed this. However, for the next game, due to our complex search algorithm, we dropped to approximately 3.1 processors average because the game was more tactical, which tends to defeat our load balancing and yield poorer results.

"We are still studying the problem and have several new ideas that seem to promise even better utilization of the processors. Obviously, we are also anticipating a greater use of multiprocessing as new hardware becomes available. This has forced us to design an algorithm than can balance the load among *n* processors, a non-trivial problem. The algorithm has been written and tested. On the X-MP/48 we have run up to twenty 'logical CPUs' with the present algorithm with no problems."

Computer chess competitions aren't quite like contests between human players, Hyatt added. During a computer match, players move pieces on an actual game board according to instructions fed to them by the competing machines. "During a game, I'm essentially reduced to an I/O peripheral for the computer," Hyatt said. "In other ways, though, it's a lot like a regular tournament — except nobody's going 'Shhh'."

## CRAY resuscitation successful

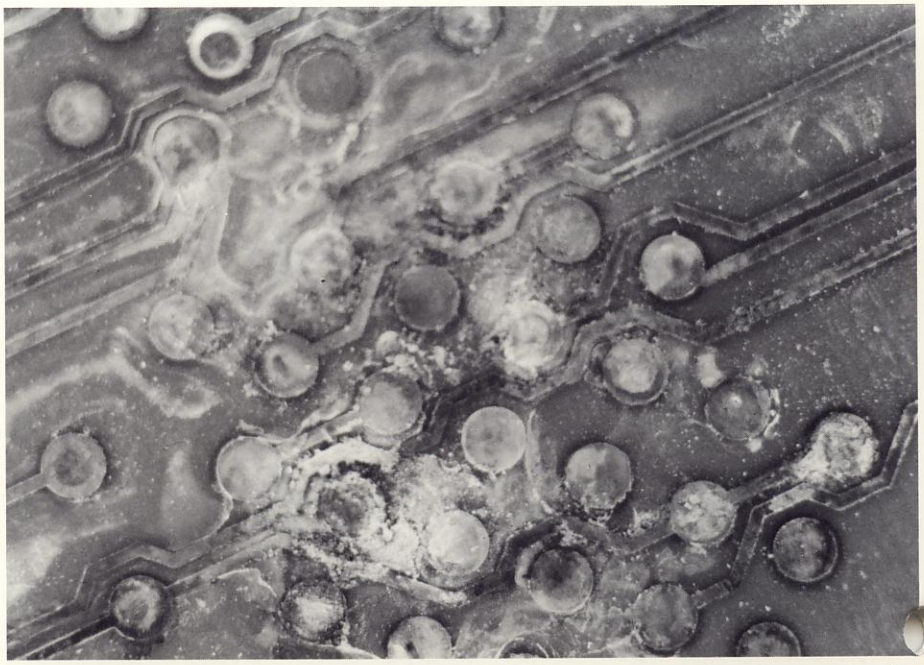
It may be a shock, but electricity and water don't mix. That lesson was learned last Fall when a CRAY-1/S at Lawrence Livermore National Laboratories (LLNL) ran into a plumbing problem. A defective water pipe leaked onto the machine, the machine went down and a major revival operation was launched.

The CRAY was rescued by a combined effort of Cray customer engineers at the Lab and personnel from Cray's manufacturing, systems test and international technical support departments in Chippewa Falls, Wisconsin. All of the machine's modules had to be removed and about half of them were sent to Chippewa Falls for inspection and cleaning. "From sampling modules and observing water patterns on the machine, it was determined which modules were likely to have gotten the most exposure. These were brought back to Chippewa, about 500 in all," explained Lou Saye, chief engineer of reliability and quality at Chippewa Falls. "The water had rust-like contaminants in it and had migrated under the printed circuit boards, between the boards and the cold plate."

The modules that were brought back to Chippewa Falls were torn down, cleaned up, baked out, put back together, tested then shipped back to the Lab, along with a cadre of Cray manufacturing personnel to help reassemble the lobotomized machine. "Chippewa Falls responded with the reinforcement that was needed," said Art Unger, engineer in charge at the Lab. "It was a full complement of people and the situation was treated almost like a new installation, including wiring and testing support."

The doused computer was disassembled, cleaned, reassembled and back up and running within 50 days. "At the time, that machine accounted for about 15 percent of our total computing capacity," commented Dr. Robert Borchers, associate director for computation at LLNL. "At first, we were sort of terrified, but the rescue operation turned out better than expected." As a precaution against repeat incidents like this one, permanent protective umbrellas are being installed over all of the Livermore computers.

Borchers wrote to express his delight at the successful recovery: "Serial 43 is back up and running, apparently as solidly as ever ... your organization's response to this incident was outstanding and attests to the responsibility Cray feels to keep the machines running, as well as to the robust nature of the I/S design and hardware. I hope you will see that all people involved are made aware of our appreciation of the quality and speed with which Cray responded."



*Rust and residue on a printed circuit board after exposure to water and contaminants.*

## WORLDWIDE BUSINESS CENTERS

### Domestic

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

**Government Marketing/**
**Government Relations Office**
1828 L Street NW, Suite 201
Washington, DC 20036
(202)775-0155

**Central Region Office**
5330 Manhattan Circle, Suite F
Boulder, CO 80303
(303)499-3055
Albuquerque
Chicago
Minneapolis

**Eastern Region Office**
11710 Beltsville Drive, Suite 500
Beltsville, MD 20705
(301) 595-5100
Atlanta
Boston
Laurel
Pittsburgh

**Petroleum Region Office**
5858 Westheimer, Suite 500
Houston, TX 77057
(713) 975-8998
Dallas
Tulsa

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

### International

**Cray Canada Inc.**
4141 Yonge Street, Suite 302
Toronto, Ontario
Canada M2P 2A8
(416) 229-2729

**Cray Research France, S.A.**
7 Rue de Tilsitt
75017 Paris, France
01-766-01-55

**Cray Research GmbH**
Perhamerstrasse 31
8000 Munich 21, West Germany
89-56014-0

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

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

CRAY RESEARCH, INC.

**Corporate Headquarters**
**Cray Research, Inc.**
608 Second Avenue South
Minneapolis, MN 55402
Tel: (612) 333-5889