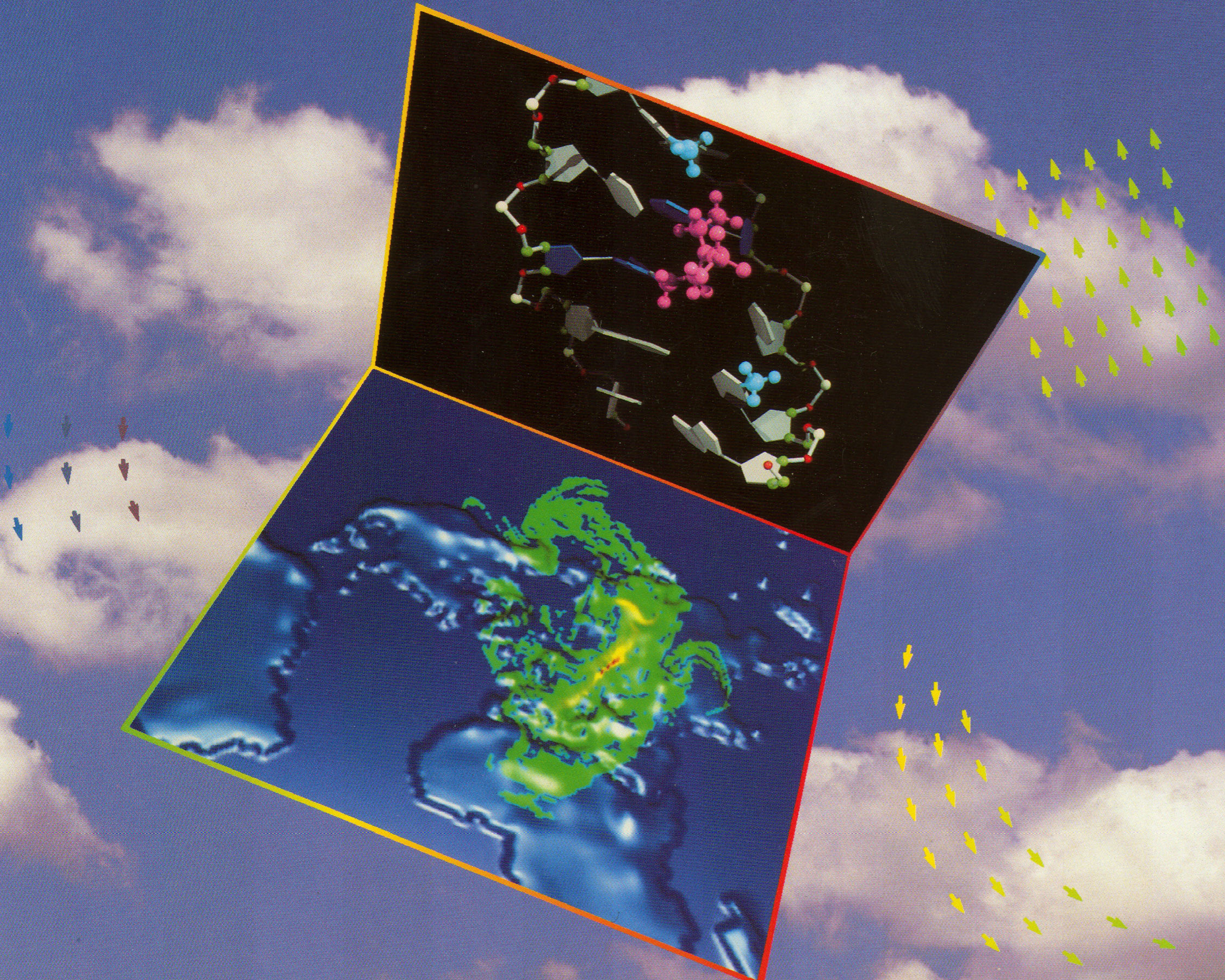
CRAY CHANNELS

FALL 1988 · A CRAY RESEARCH, INC., PUBLICATION

Health and environment



CRAY CHANNELS

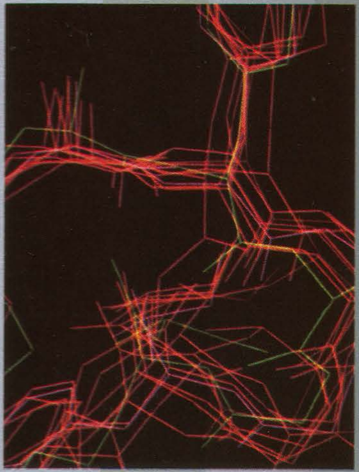
In this issue

The insights provided by supercomputers have launched a second scientific revolution. Today, the results of these scientific advancements are impossible to ignore — especially advancements that are improving the quality of our health, our environment, and our lives. For example, viruses that scientists were once unable to examine in the laboratory, now are scrutinized from workstation screens. And on a more macroscopic level, researchers are beginning to use supercomputers to trace the complex spread of those same viruses through the human population.

In this issue of CRAY CHANNELS we look at advancements in health and environmental research through the eyes of scientists at prominent research institutions. Researchers at the Los Alamos National Laboratory report on their efforts to map the spread of the AIDS epidemic computationally by studying complex social interactions. At the National Cancer Institute, scientists are using their Cray system to understand cancer and other diseases, while researchers at the Atmospheric Environment Service of Canada are studying the spread of radiation that was released from the Chernobyl nuclear reactor accident. Scientists from Yale University and Scripps Clinic and Research Foundation share their computational methods for studying chemical structures and processes. Also, we take a look at Cray Research's goals and visions for the UNICOS operating system. Our regular departments feature new system orders and announcements of Cray Ada and Common Lisp software. The User News department highlights computational research into cosmic strings and radio imaging of the Crab Nebula supernova remnant.

Whether they are used to examine minute molecules or the spread of epidemics, supercomputers are providing insights that will continue to improve the quality of our lives. Here's to your health!

Features



16

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 10, Number 3

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: A meeting of macroscopic and microscopic worlds symbolizes the diversity of supercomputer applications related to health and the environment. The top image shows a model of an anticancer drug bonded to DNA (image courtesy Frederick Hausheer, National Cancer Institute). The bottom image shows results from a study of the spread of radionuclides from the Chernobyl accident (image courtesy Janusz Pudykiewicz, Atmospheric Environment Service of Canada).

Supercomputing in biomedical research: applications to sequence analysis and molecular biology

Jacob V. Maizel, Jr., National Cancer Institute, Frederick, Maryland
NCI researchers run volumes of genetic sequence data through their Cray system to study cancer and other diseases.

Computer visualization of radionuclides from the Chernobyl accident

Janusz Pudykiewicz, Atmospheric Environment Service of Canada, Dorval, Quebec
William Samayoa, Cray Research, Inc.
A large-scale atmospheric circulation model accurately tracks material released during the Chernobyl accident.

Building large-scale models to understand the AIDS epidemic

James M. Hyman, E. Ann Stanley, Stirling A. Colgate, and Scott P. Layne, Los Alamos National Laboratory, Los Alamos, New Mexico
Supercomputer models provide a framework for studying the complicated social interactions that lead to the spread of AIDS.

Hydrophobic hydration by supercomputer: a free energy perturbation study

B. Govinda Rao and U. Chandra Singh, Scripps Clinic and Research Foundation, La Jolla, California
Scripps' Cray system provides insights into the hydration of a variety of solute molecules.

Crystallographic refinement by simulated annealing on supercomputers

Axel T. Brünger, Yale University, New Haven, Connecticut
A new refinement technique simplifies the manual refitting of structures.

A blueprint for the UNICOS operating system

Steve P. Reinhardt, Cray Research, Inc.
Cray Research continues to enhance the UNICOS operating system for maximum supercomputer performance.

Corporate register

Applications update

User news

Gallery

Supercomputing in biomedical research

Applications to sequence analysis and molecular biology

Jacob V. Maizel, Jr., National Cancer Institute, Frederick, Maryland

Modern biology is increasingly dependent on powerful computers for genetic, structural, biophysical, and engineering studies. Molecular biology, the study of genes and gene products at the molecular level, provides a rich field in which to explore the use of supercomputing in the biological sciences. Genetic sequence data, for example, are accumulating at a rapid pace, are ideally suited to computer analysis, and pose important problems of a magnitude that require supercomputing. The National Cancer Institute (NCI) has developed the Advanced Scientific Computing Laboratory (ASCL) to exploit this data for basic research in the study of cancer and other diseases.

The facility

From the end of 1983 to early 1986 the National Cancer Institute defined the need for a supercomputer devoted to biomedical applications. In early 1986 NCI acquired and installed such a system in the ASCL under the oversight and scientific direction of the institute's Laboratory of Mathematical Biology. Although situated in the environment of a scientific laboratory, the supercomputer is available to members of other NCI, National Institutes of Health (NIH), and academic institutions. Two broad-based scientific groups composed of intramural and extramural scientists provide further oversight for time allocation and management.

The system installed initially was a CRAY X-MP computer system with two million words of memory. Within a month of its installation it was saturated, and over the course of the following two years it was upgraded to its maximum capacity as a CRAY X-MP/24 system. The supercomputing environment is enriched by the presence of VAX 8600 and 785 front-end minicomputers, and an Apollo workstation network linked directly to the Cray system by a Network Systems Corporation HYPERchannel network. A heterogeneous Ethernet, TCP/IP-based network links additional terminal servers, and several high-performance graphics workstations from Silicon Graphics, Evans and Sutherland, Sun Microsystems, AT&T, and others, along with other peripheral devices.

The long-term goal of the ASCL is to use supercomputing in the study of scientific problems of a magnitude or urgency that are not readily treated in routine computing centers. Although the dedication of a supercomputer to biomedicine is in itself a pioneering experiment, it is based on the availability of proven computer technology, large sources of readily available data, and significant problems. The emphasis on molecular biology is justified by the rapid expansion of primary data, the match between the data and current computer technology, and the enticing potential for analysis and discovery of fundamental biological laws.

Sequence comparison

Because of the emergence of recombinant DNA technology and nucleic acid sequencing techniques, genes and their products increasingly are defined by sequences of four letters representing the chemical units from which they are built. The best way to discover relationships between genes is through analysis and comparison of their nucleic acid and protein sequences, conveniently represented as extended strings from an alphabet that by convention uses the four letters a, c, g, and t for nucleic acids and a 20-letter alphabet for proteins. So far, only a few viruses have had their gene sequences entirely determined. That situation is likely to change as the scientific community moves toward the inevitable project of sequencing the genes of complex organisms, including entire human genomes. Soon after the era of rapid sequencing began it became clear that computation would play an important role in handling and understanding these new kinds of data.

Published nucleic acid sequences are entering the collections maintained by NIH, IntelliGenetics, Inc., and Los Alamos National Laboratory at a rate of nearly one million bases per month. The present size of the collection is approximately 10 million bases. The comparison of all new sequences with those already in the database is essential. If we are to understand the complex interaction of macromolecules in normal and abnormal cells, tests for identical sequences are needed along with ones that detect subtle relationships indicating functional similarities between genes and their products. Examples arise daily of newly discovered relationships between cancerous and normal genes, of protein classes with diverse origins but common structural motifs, of insights into molecular evolution, and of mechanisms to generate the seemingly enormous diversity that organisms show. A new gene may be part of a known superfamily, may be the start of a new superfamily, or may share limited domains with one or more groups. A recent example of the usefulness of genetic sequencing in cancer molecular biology is the classification of a family of oncogenes that have similarity to protein kinases.[1] Earlier work had shown that these enzymes modify many cellular enzymes with profound regulatory effects. Molecules with this kind of activity are ideal candidates for central involvement in malignant behavior.

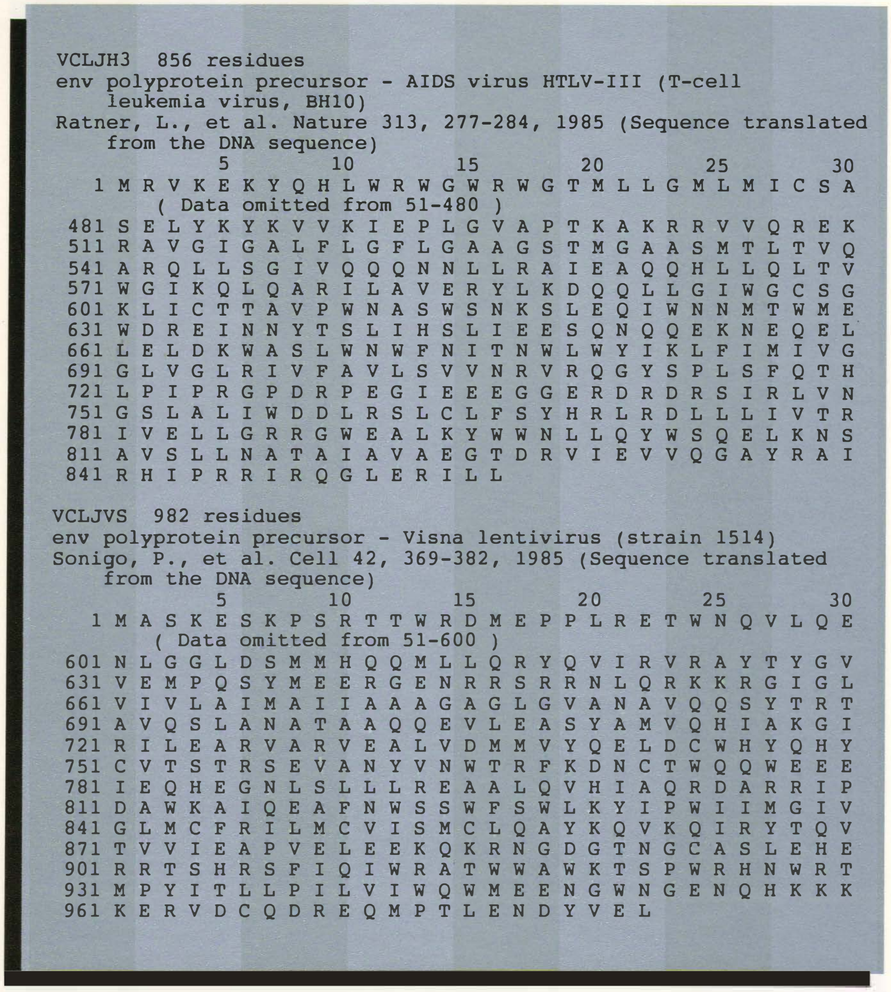
The most useful strategies and algorithms for comparing sequences are designed to find localized regions of similarity between sequences rather than a single global alignment. This approach has emerged with the realization that macromolecules are composed of domains of distinct structure and function that are sometimes translocatable and independent. At the same time, the overall context in which individual domains are located is known to affect detailed behavior. These facts imply that the simple weighting methods for mismatches, insertions, and deletions now used universally in scoring local alignments between sequences will need refinement and sophisticated improvement.

Analysis methods

Nevertheless, two basic and efficient approaches are used widely. One of these, called here the "matrix" method, aims to compare all individual elements of one sequence to those of the other, using a matrix proportional in size to the product of the two sequence lengths, along with dynamic programming techniques to store the local similarities.[2] A traceback through the matrix extracts the details of local similarities that meet a given criterion. Execution time is dominated by the matrix-filling process and is proportional to the product of the two sequence lengths.

A second approach, called here the "lexical" method, converts the original sequences into lists of "words" of chosen size, and notes their locations. For example, words of three nucleic acid characters with four possible letters at each place lead to 64 possible words. The lists then are examined for similar words at constant or nearly constant stride through the sequences. In a popular, fast implementation of this method, the aligned subregion is derived directly using the words. This overlooks certain important kinds of similarities and has been improved and further optimized in a version using the lexical stage to localize potential regions of interest, followed by a very much constrained matrix method to produce a final alignment. With these constraints, the lexical phase dominates and requires storage proportional to the sums of the sequence lengths, and time proportional to the product of one sequence length and logarithm (base two) of the other.

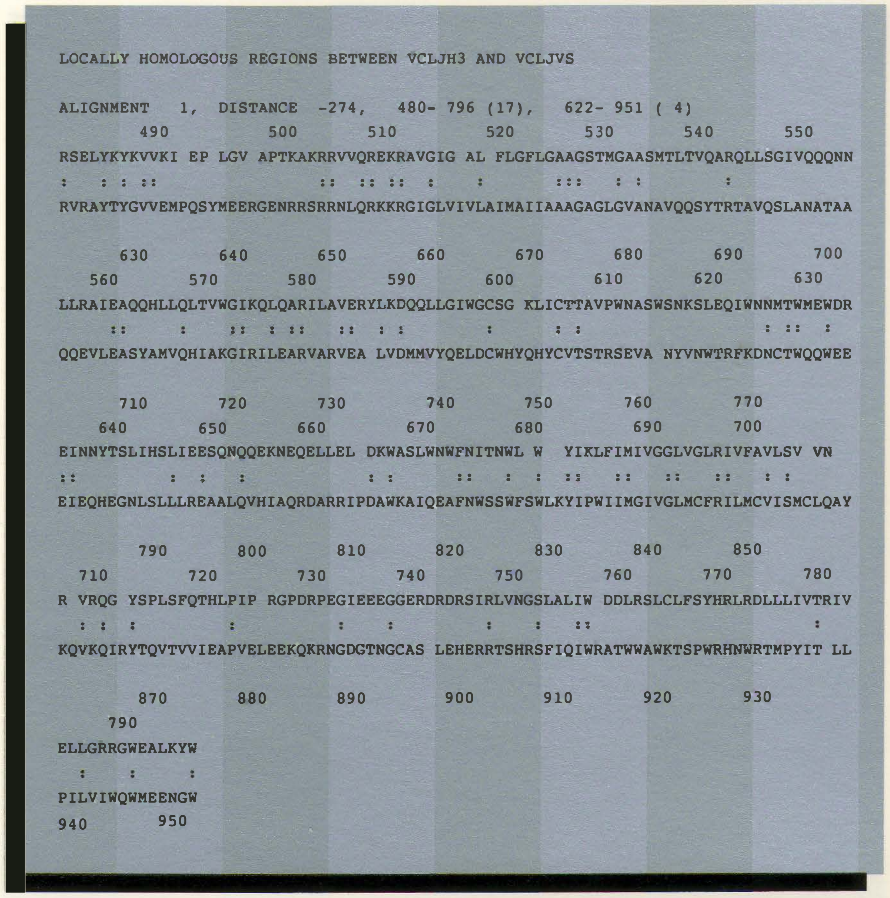
Both of these methods are used widely on various types of computers. The matrix stage can be vectorized on vector computers such as Cray systems and, in principal, could be parallelized on many architectures. The lexical stage may have parallelism, but it has not been optimized by us for vector architecture. Nonetheless, the Cray system is used to treat large problems because it has the fastest scalar speed of any available system. The matrix method for proteins is implemented in M. Kanehisa's SEQHP program, and a hybrid lexical/constrained-matrix method is implemented in his SEQFP program.[2] Comparable programs SEQH and SEQF are for nucleic acids. At the ASCL they are implemented on the VAX front-end and Cray computers within a special program environment called the IDEAS package with many other utilities for sequence analysis. An option at run time decides if a problem is large enough to consider use of the Cray system and asks if the user wishes to submit it. If the user decides to do so, it creates the necessary input and submits it to an appropriate batch queue on the supercomputer, otherwise it is run interactively or in batch mode on the VAX computer. Many of the most heavily used programs have been optimized considerably for the Cray system. The matrix algorithms generally have been vectorizable as a result of array indexing schemes that address them antidiagonally rather than diagonally. Because the high precision of floating point computation is not used and large integers are unnecessary for many operations, the packing of small data items in large words has been accomplished where memory limitations are more important than execution speed.



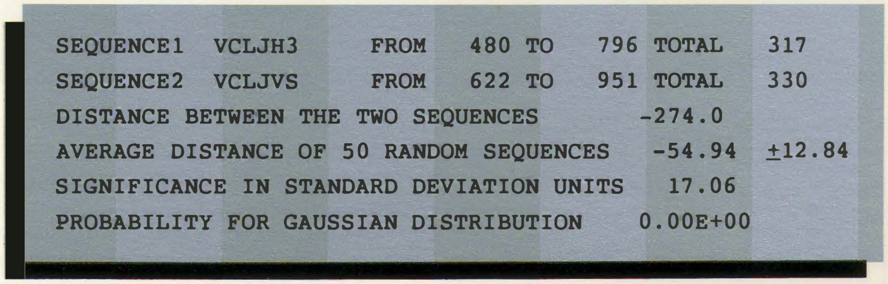
```
VCLJH3  856 residues
env polyprotein precursor - AIDS virus HTLV-III (T-cell
    leukemia virus, BH10)
Ratner, L., et al. Nature 313, 277-284, 1985 (Sequence translated
    from the DNA sequence)
            5        10        15        20        25        30
  1 M R V K E K Y Q H L W R W G W R W G T M L L G M L M I C S A
        ( Data omitted from 51-480 )
481 S E L Y K Y K V V K I E P L G V A P T K A K R R V V Q R E K
511 R A V G I G A L F L G F L G A A G S T M G A A S M T L T V Q
541 A R Q L L S G I V Q Q Q N N L L R A I E A Q Q H L L Q L T V
571 W G I K Q L Q A R I L A V E R Y L K D Q Q L L G I W G C S G
601 K L I C T T A V P W N A S W S N K S L E Q I W N N M T W M E
631 W D R E I N N Y T S L I H S L I E E S Q N Q Q E K N E Q E L
661 L E L D K W A S L W N W F N I T N W L W Y I K L F I M I V G
691 G L V G L R I V F A V L S V V N R V R Q G Y S P L S F Q T H
721 L P I P R G P D R P E G I E E E G G E R D R D R S I R L V N
751 G S L A L I W D D L R S L C L F S Y H R L R D L L L I V T R
781 I V E L L G R R G W E A L K Y W W N L L Q Y W S Q E L K N S
811 A V S L L N A T A I A V A E G T D R V I E V V Q G A Y R A I
841 R H I P R R I R Q G L E R I L L

VCLJVS  982 residues
env polyprotein precursor - Visna lentivirus (strain 1514)
Sonigo, P., et al. Cell 42, 369-382, 1985 (Sequence translated
    from the DNA sequence)
            5        10        15        20        25        30
  1 M A S K E S K P S R T T W R D M E P P L R E T W N Q V L Q E
        ( Data omitted from 51-600 )
601 N L G G L D S M M H Q Q M L L Q R Y Q V I R V R A Y T Y G V
631 V E M P Q S Y M E E R G E N R R S R R N L Q R K K R G I G L
661 V I V L A I M A I I A A A G A G L G V A N A V Q Q S Y T R T
691 A V Q S L A N A T A A Q Q E V L E A S Y A M V Q H I A K G I
721 R I L E A R V A R V E A L V D M M V Y Q E L D C W H Y Q H Y
751 C V T S T R S E V A N Y V N W T R F K D N C T W Q Q W E E E
781 I E Q H E G N L S L L L R E A A L Q V H I A Q R D A R R I P
811 D A W K A I Q E A F N W S S W F S W L K Y I P W I I M G I V
841 G L M C F R I L M C V I S M C L Q A Y K Q V K Q I R Y T Q V
871 T V V I E A P V E L E E K Q K R N G D G T N G C A S L E H E
901 R R T S H R S F I Q I W R A T W W A W K T S P W R H N W R T
931 M P Y I T L L P I L V I W Q W M E E N G W N G E N Q H K K K
961 K E R V D C Q D R E Q M P T L E N D Y V E L
```

Figure 1. Examples of protein sequence data. Part of the information included with each sequence in the National Biomedical Research Foundation protein sequence database is shown for the envelope proteins of two lentiviruses. The sequence identified VCLJH3 is from the human immunodeficiency virus implicated in AIDS. The sequence identified VCLJVS is from a sheep visnavirus, of related general structure, that causes respiratory disease.

Figure 1 shows regions of local similarity between the envelope proteins of a human immunodeficiency virus (HIV), believed to be the causative agent of AIDS, and a partially analogous sheep visnavirus. The sequences are typical of the data available in the National Biomedical Research Foundation protein sequence database. It is difficult to recognize similarity between these sequences visually, but several local



```
LOCALLY HOMOLOGOUS REGIONS BETWEEN VCLJH3 AND VCLJVS

ALIGNMENT   1,  DISTANCE  -274,   480- 796 (17),   622- 951 ( 4)
        490          500       510         520       530       540       550
RSELYKYKVVKI EP LGV APTKAKRRVVQREKRAVGIG AL FLGFLGAAGSTMGAASMTLTVQARQLLSGIVQQQNN
:   : : ::                ::  :: ::  :    :       :::   : :        :
RVRAYTYGVVEMPQSYMEERGENRRSRRNLQRKKRGIGLVIVLAIMAIIAAAGAGLGVANAVQQSYTRTAVQSLANATAA

      630       640       650       660       670       680       690       700
   560       570       580       590       600        610       620       630
LLRAIEAQQHLLQLTVWGIKQLQARILAVERYLKDQQLLGIWGCSG KLICTTAVPWNASWSNKSLEQIWNNMTWMEWDR
     ::     :    ::  : ::   ::  : :        :      : :                  : ::  :
QQEVLEASYAMVQHIAKGIRILEARVARVEA LVDMMVYQELDCWHYQHYCVTSTRSEVA NYVNWTRFKDNCTWQQWEE

      710       720       730        740       750       760        770
    640       650       660        670        680         690       700
EINNYTSLIHSLIEESQNQQEKNEQELLEL DKWASLWNWFNITNWL W  YIKLFIMIVGGLVGLRIVFAVLSV VN
::         :  :   :            : :      ::   :  :  ::   ::   ::   ::   : :
EIEQHEGNLSLLLREAALQVHIAQRDARRIPDAWKAIQEAFNWSSWFSWLKYIPWIIMGIVGLMCFRILMCVISMCLQAY

        790       800       810       820       830       840       850
  710        720        730       740       750        760       770       780
R VRQG YSPLSFQTHLPIP RGPDRPEGIEEEGGERDRDRSIRLVNGSLALIW DDLRSLCLFSYHRLRDLLLIVTRIV
  : :  :         :          :    :         :    :   ::                      :
KQVKQIRYTQVTVVIEAPVELEEKQKRNGDGTNGCAS LEHERRTSHRSFIQIWRATWWAWKTSPWRHNWRTMPYIT LL

        870       880       890        900       910       920       930
     790
ELLGRRGWEALKYW
  :    :     :
PILVIWQWMEENGW
940       950
```



```
SEQUENCE1  VCLJH3     FROM    480 TO    796 TOTAL    317
SEQUENCE2  VCLJVS     FROM    622 TO    951 TOTAL    330
DISTANCE BETWEEN THE TWO SEQUENCES         -274.0
AVERAGE DISTANCE OF 50 RANDOM SEQUENCES     -54.94  ±12.84
SIGNIFICANCE IN STANDARD DEVIATION UNITS     17.06
PROBABILITY FOR GAUSSIAN DISTRIBUTION       0.00E+00
```

Figure 2 (top). Comparison of HIV and visnavirus envelope protein sequences. The full matrix program, SEQHP, was used to produce about 50 regions of local similarity; the region of greatest similarity is shown.

Figure 3 (bottom). Results of a test of statistical significance for the similarity between the HIV and visnavirus envelope protein sequences. The test, performed by the SEQDP program, shows little chance that the degree of similarity is due to random occurrences.

similarities found by the SEQHP program are shown as aligned in Figure 2. A more detailed statistical analysis by a Monte Carlo technique using the SEQDP program shows a low probability of occurrence by chance (Figure 3). Observation of similarities rapidly provides a broadened viewpoint and a base of knowledge from previously known systems that greatly helps researchers understand new diseases.

Performance

Although sequence comparisons have consumed 5 percent or less of the CPU cycles on the NCI Cray system, it is revealing to consider the potential magnitude of the problem, especially in light of the current interest in increasing our study of human and other complex genomes. Execution times are inversely related to speeds, and a commonly accepted speed description is millions of instructions per second (MIPS), with the VAX 11/780 called by some a 1-MIPS processor. For this article we will arbitrarily define it that way and note that in those terms the CRAY X-MP single-processor system averages one hundred times the performance of the VAX 11/780 computer, consequently defining the ASCL two-processor CRAY X-MP/24 system as 200 MIPS. From practical experience the matrix methods are executed at from 1000-10,000 "matrix elements" per second per MIPS over a wide range of problem sizes and machines. Lexical methods for a problem of 100 elements of query sequence and 2,000,000 elements of database execute at 100,000 to 1,000,000 elements per second per MIPS. The following two examples show the magnitude of these computational problems.

Comparison of hypothetical HIV proteins to a reference protein database

A typical present-day problem could be the comparison of all, instead of the above-mentioned single, hypothetical proteins encodable by HIV with all of the proteins of a typical database. This is a minimum of about 3000 amino acids worth of viral protein sequence. If information from the complementary strand and from all possible translational reading frames is considered, this quantity grows by a factor of six. This test makes sense because biological systems are known to exist in which proteins arise from genes spliced in all possible triplet reading frames, and in which information from both strands is translated separately. The translation sometimes overlaps the two strands and several reading frames. Present reference databases, such as that of the National Biomedical Research Foundation, contain more than 2,000,000 amino acid residues. One can argue about the exact make-up desired for such a reference database, but this size is likely to be within an order of magnitude. On a 1-MIPS processor, the matrix method would then require from one to ten weeks of CPU time, but less than 10 hours at 200 MIPS. The lexical method requires about 20 hours at one MIPS and only 10-20 minutes on the Cray system. Either of these tasks would be feasible, and a variety of such partial studies has been pursued.

Comparison of human and other complex genomes

Biological researchers have formulated several decisive plans to sequence the human genome. The sequencing of mouse, fruit fly, yeast, and other complex genomes will be undertaken as well, because these organisms serve as experimental models when human experimentation is inappropriate. With sizes on the order of three billion bases for the human or mouse genome, on the order of 10^{19} elements will have to be compared. Even if computers capable of tera-operations per second (10^{12} ops/sec) become available, hundreds of years would be needed to complete the matrix methods. If the lexical algorithm scales successfully, however, it could be feasible on such a system in a few hours. An encouraging note is that the memory capacity of the largest supercomputers today, capable of storing 256 million or more 64-bit words, is useful for many sequence analysis problems. Since only four nucleic acid bases exist, they can be encoded directly in two bits per base. This would allow the storage of several complex genome sequences directly in dynamic memory. This capability would be extremely beneficial because the data movement that is a major part of most of these analyses would be most efficient using dynamic memory. Although algorithms and computers will improve, sequence data will continue to challenge supercomputers.

Nucleic acid secondary structure prediction

Another computationally intensive area of increasing importance is the prediction of RNA secondary structure. This property is important for regulation of expression and processing of messenger RNA, interactions between proteins and nucleic acids, and understanding the remarkable catalytic properties directly expressed by some nucleic acids. Efficient dynamic programming algorithms have been developed using various empirical rules to evaluate structures for optimal stability.[3] Execution time of these algorithms varies approximately as the cube of the number of bases, and storage is proportional to the square of the number of bases. Prediction of the optimal secondary structure of 1000 bases typically requires about an hour on a 1-MIPS computer. As with the sequence comparison programs mentioned above, optimization using similar techniques gives speedups of 100-fold for a single CRAY X-MP processor. As a result, less than one minute of computation time is required for problems that would require one hour on a 1-MIPS computer. Memory limits the size of the problems to 1200 bases on a four-million word machine without packing, or about 4000 bases with packing. Using a CRAY-2 system with large memory, it has been possible to fold molecules approaching 10,000 bases. Although researchers want to know the structure of large molecules, present algorithms have unpredictable reliability. For many small problems they are encouragingly successful, but the results must be considered speculative. Consequently, considerable development and testing of modified algorithms needs to be undertaken.

Monte Carlo simulations

Researchers working on DNA and protein sequence comparison and RNA secondary structure prediction want to know how far from random their results are. To find out, they frequently use a powerful test that randomly shuffles the order of characters in a sequence while keeping the overall composition the same. They then repeat the computer analysis. Usually a comparison of the average of many random experiments with the original experiment will detect instances of high significance. Unfortunately, biological systems are not entirely describable by statistical and mathematical terms alone. This leads to many instances, especially at the borderline of significance, that can be resolved confidently only by further experimentation.

Implementation of the shuffling test can increase the computer time in proportion to the number of random samples. Its use has led to a number of interesting observations, especially in conjunction with RNA structure prediction. Using subregions of RNA sequences taken successively along a molecule, researchers have found that most regions are not distinguishable from random. However, regions exist with more stable structures than expected, while others have less stability or no structure. In several cases these regions correlate with known biological features that already are suspected to be structure dependent.[4] Data exist that show a correlation between the highly variable regions of different strains of human immunodeficiency viruses and regions on the RNA where no stable structure can be predicted. This observation may lead to an understanding of the mechanism and limits of hypervariability, and may have implications for the design of vaccines and for other antiviral strategies.

Conclusion

The feasibility of developing a powerful computer facility dedicated to biomedical research has been demonstrated. The system assembled at the Frederick Cancer Research Facility has great capability and is delivering at or ahead of expectations. In the physical sciences, the availability of a tool with tens to hundreds of times the usual capabilities not only increases the amount of work done, but also qualitatively alters the way people view their science. Advanced computing is having a similar effect on modern biology.

Acknowledgments

The author thanks Frederick Hausheer for discussions, Robert Jernigan for discussions and comments in general, and Charles Crum and the staff of the Advanced Scientific Computing Laboratory for assistance in providing an operating environment favorable to scientific computing. Minoru Kanehisa kindly adapted and provided the IDEAS sequence analysis package from which the SEQHP and SEQDP programs were used in this paper. Michael Zuker provided the original RNA folding program, which was optimized for the CRAY X-MP system by Cray Research staff, J. Chen of the ASCL, and S. Le of NCI.

About the author

Since 1983, Jacob V. Maizel, Jr., has been chief of the Laboratory of Mathematical Biology of the National Cancer Institute. He previously was chief of the Section on Molecular Structure of the Laboratory of Molecular Genetics of the National Institute of Child Health and Human Development. Maizel received B.S. and M.S. degrees in biochemistry from Pennsylvania State University in 1955 and 1957 respectively. He received a Ph.D. degree in biochemistry from the California Institute of Technology in 1959.

References

1. Hanks, S. K., A. M. Quinn, and T. Hunter, "The Protein Kinase Family: Conserved Features and Deduced Phylogeny of the Catalytic Domains," *Science*, Vol. 241, 1988, pp. 42-52.
2. Goad, W. B. and M. Kanehisa, "Pattern Recognition in Nucleic Acid Sequences. I. A General Method for Finding Local Homologies and Symmetries," *Nucleic Acids Research*, Vol. 10, 1982, pp. 247-263.
3. Rossman, M. G., E. Arnold, J. W. Erickson, E. A. Frankenberger, J. P. Griffith, H. Hecht, J. E. Johnson, G. Kamer, M. Luo, A. G. Mosser, R. R. Rueckert, B. Sherry, and G. Vriend, "Structures of a Human Common Cold Virus and Functional Relationship to Other Picornaviruses," *Nature*, Vol. 317, 1985, pp. 145-153.
4. Le, S., J. Chen, M. J. Braun, M. A. Gonda, J. V. Maizel, Jr., "Stability of RNA Stem-Loop Structure and Distribution of Non-random Structure in the Human Immunodeficiency Virus (HIV-1)," *Nucleic Acids Research*, Vol. 16, 1988, pp. 5153-5168.

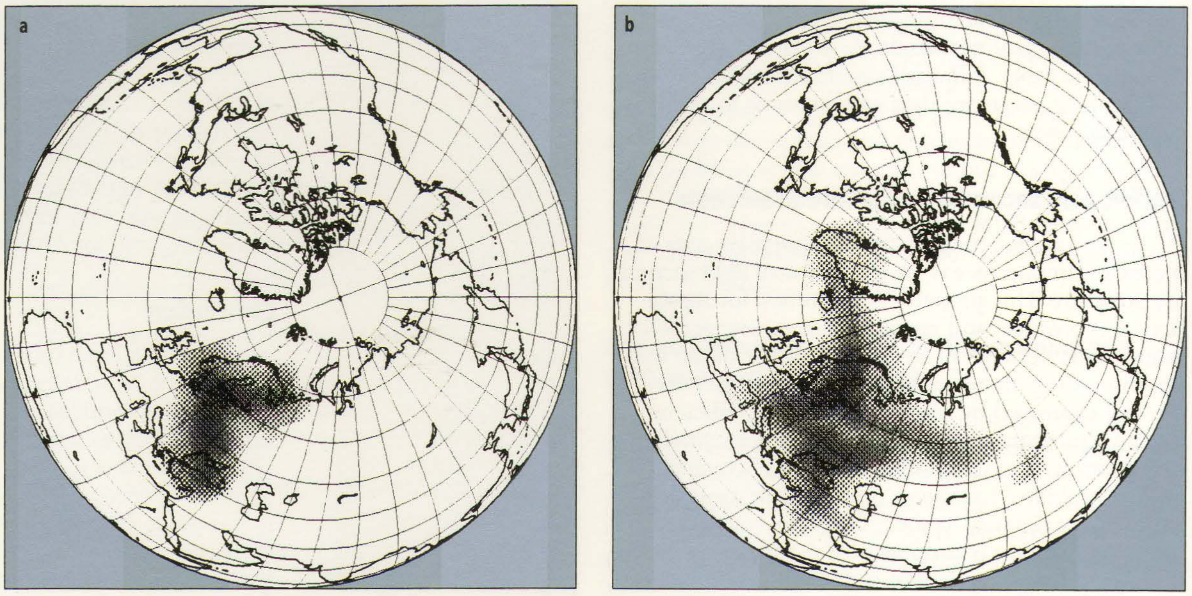


Figure 1. The Chernobyl cloud at $\sigma = 1$ (anemometer level) on April 27 (a) and April 29 (b), 1986. The activity on the figures is represented by the density of the black marker particles.

Computer visualization of radionuclides from the Chernobyl accident

Janusz Pudykiewicz, Atmospheric Environment Service of Canada, Dorval, Quebec
William Samayoa, Cray Research, Inc.

The numerical simulation of environmental processes is a tool used increasingly for practical environmental problem solving. A well-known example is the use of numerical models to track atmospheric chemical tracers for studies of acid rain and stratospheric ozone depletion. We recently used such a tracer model to simulate the dispersion of nuclear debris from the Chernobyl nuclear reactor accident. The research was conducted using the CRAY X-MP/28 computer system with SSD solid-state storage device at the Canadian Meteorological Center (CMC). We used a hemispheric model because we believed that the effects of the accident might extend around the globe.

The models

The current version of the atmospheric circulation model used in this research is fully three-dimensional and could be executed on a grid with a horizontal resolution of 50 kilometers over the Northern Hemisphere.[1] The Cray system executed the CMC's Objective Analysis (OA) series of programs. These programs are used operationally by the CMC to analyze meteorological data, and were used in this research to solve the radioactive dispersion problem. The computational speed and large memory available with the Cray system were the critical factors that made the simulation possible.

The three-dimensional tracer model used in this study is based on the mass conservation equation written in the terrain-following coordinate system σ. The model equations were transformed to the polar stereographic projection system true at 60° north latitude to simplify their solution. The horizontal and vertical advection and gravitational settling were modeled using the semi-Lagrangian algorithm developed by A. Robert[2] and subsequently used for pollution problems by J. Pudykiewicz and A. Staniforth.[3] The vertical diffusion equations were solved using the implicit method described by P. Concus et al.[4]

The horizontal grid of the model is uniform with a resolution of 150 kilometers on the polar stereographic projection. The corresponding grid domain spans 180-by-180 points and is centered on the North Pole. Eleven vertical levels were used for the numerical solution of the model equations. The layer corresponding to $\sigma = 1$ is located at the anemometer level $-z_a$. The thin layer between z_a and Earth's surface was treated analytically using surface layer theory (Businger et al.[5]).

The target grid of the analysis at the moment of the accident was a Gaussian latitude-longitude grid with a resolution of 128-by-32 points over the Northern Hemisphere. The temporal resolution of the meteorological fields from the OA system was six hours. The meteorological fields for intermediate time levels were obtained by time interpolation.

Results of the simulation

The hemispheric tracer model was executed for the four-week period beginning April 25, 1986. During the first two days after the Chernobyl accident, the radioactive cloud moved mostly toward Scandinavia. A second southern segment of the cloud had spread southeast over the Black Sea toward the Middle East (Figure 1a).

On April 28, the northern part of the radioactive cloud moved southward, crossing Poland, Germany, and Austria. On April 29, after the dissipation of the blocking high-pressure system that was

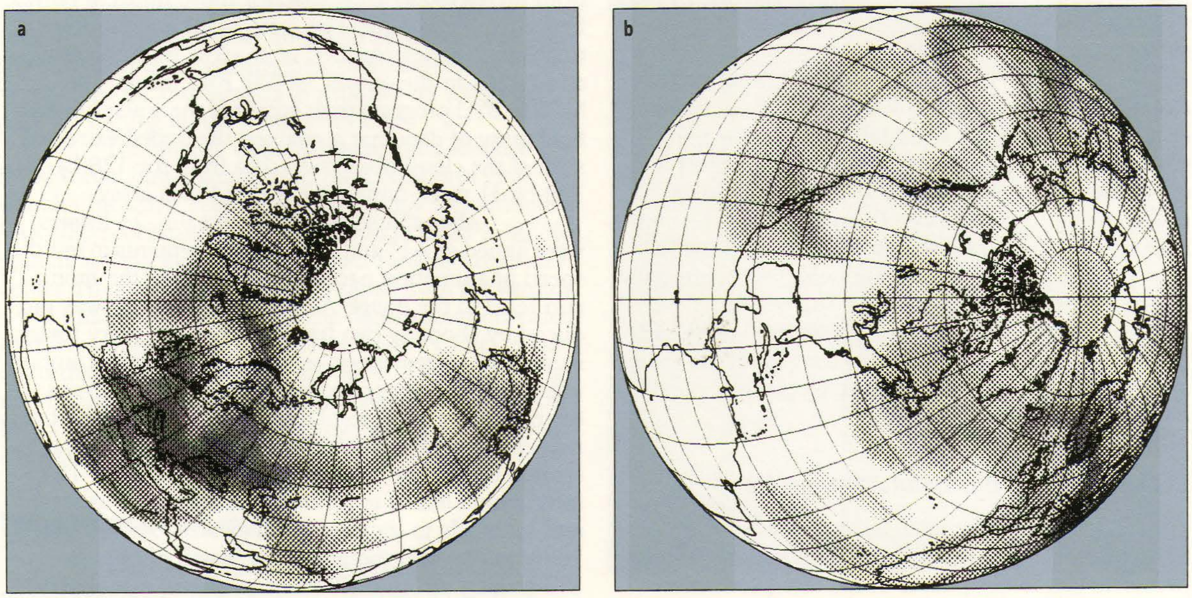


Figure 2. The Chernobyl cloud at $\sigma = 1$ (anemometer level) on May 2 (a) and May 8 (b), 1986. The activity on the figures is represented by the density of the black marker particles.

centered northeast of Chernobyl, a well-developed westerly flow began to transport radioactive material across the USSR. The surface values of the activity of the ^{131}I isotope on April 29 are shown in Figure 1b. The cyclonic circulation system related to the deep low centered near Iceland began to move radioactive material from Scandinavia and the eastern part of the North Sea toward Greenland. The analysis shows that the two major directions of transport to the Western Hemisphere were established at the end of April 1986 (Figure 1b). During the following days, the cloud moved along these two routes toward North America.

On May 2, the cloud spread over Greenland and the region of small activities approached the northern part of Quebec (Figures 2a and 3a). The part of the cloud traveling along the westerlies was separated from the Arctic part by the high-pressure system that had developed over the North Pole. At this time much radioactive material was transferred to Greenland. This occured despite the fact that the Icelandic low, so critical to the transfer of radioactivity across the Atlantic, became weaker and moved northeastward.

On May 4, the eastern part of the cloud at the anemometer level ($\sigma = 1$) had spread across Asia and the Pacific Ocean. The material traveling much faster at high levels was brought down by descending large-scale motion and appeared at the surface around southern California. The flow associated with the low-pressure system centered over northern Quebec and the high-pressure system over the North Pole transported the radioactive material above the Canadian Arctic westward. This transport feature is an interesting example of the easterly flow in the polar region.

On May 6, after the dissipation of the low over northern Quebec, the radioactive material originally spread over the Arctic region moved rapidly southeastward. The Atlantic part of Canada was thus affected initially by radioactive material injected from the Arctic regions on May 6. The main part of the cloud to affect the Atlantic region arrived later, around May 8 (Figures 2b and 3b). The descending motions on the western side of the cloud brought down some

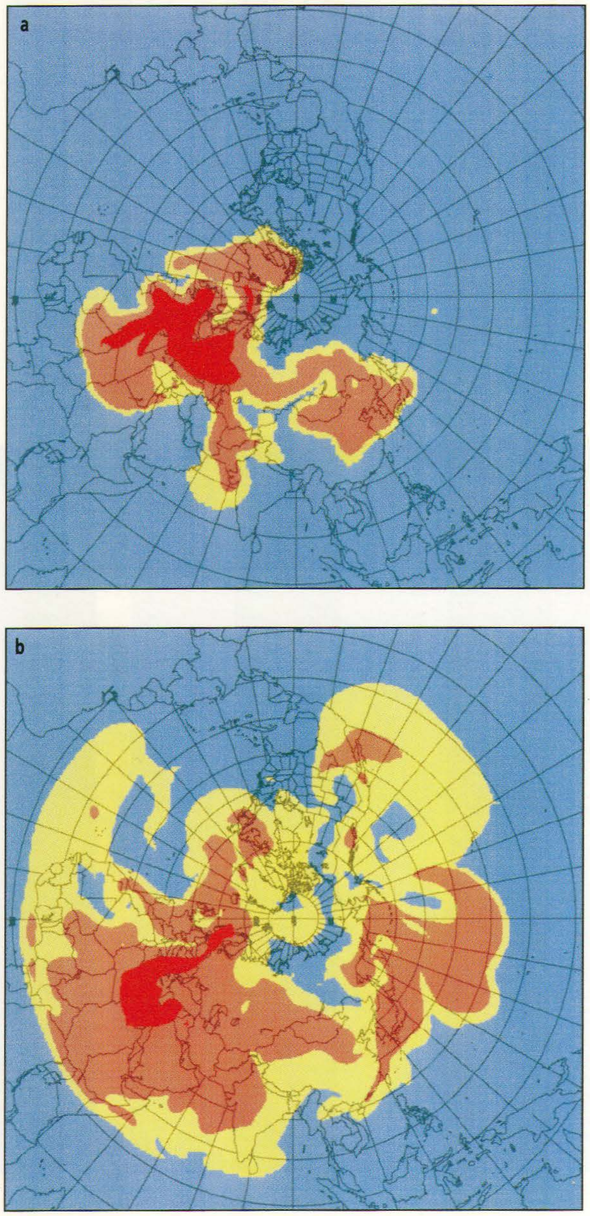


Figure 3. Activity fields (bq kg^{-1}) for ^{131}I at $\sigma = 1$ for May 2 (a) and May 8 (b). Activities between 10^{-4} and 10^{-2} (bq kg^{-1}) are shown in yellow, activities between 10^{-2} and 100 (bq kg^{-1}) in pink, and activities larger than 100 (bq kg^{-1}) in red.

material from the higher levels, which appeared along the west coast from Vancouver to Alaska and over the southwestern United States.

On May 8, the scale of the radioactive cloud became comparable to the scale of the Northern Hemisphere because eddylike mixing by the synoptic-scale low-pressure systems superposed with the average zonal flow spread radioactive material toward the equator (Figures 2b and 3b). On the hemispheric scale the cloud shows a pattern essentially analogous to turbulent diffusion, with low pressure systems acting as mixing eddies. The transport over Asia and the Pacific was actually a superposition of the zonal and eddy mixing, whereas transfer of the radioactive material across the Atlantic Ocean was purely eddylike.

Visualization of the simulation results

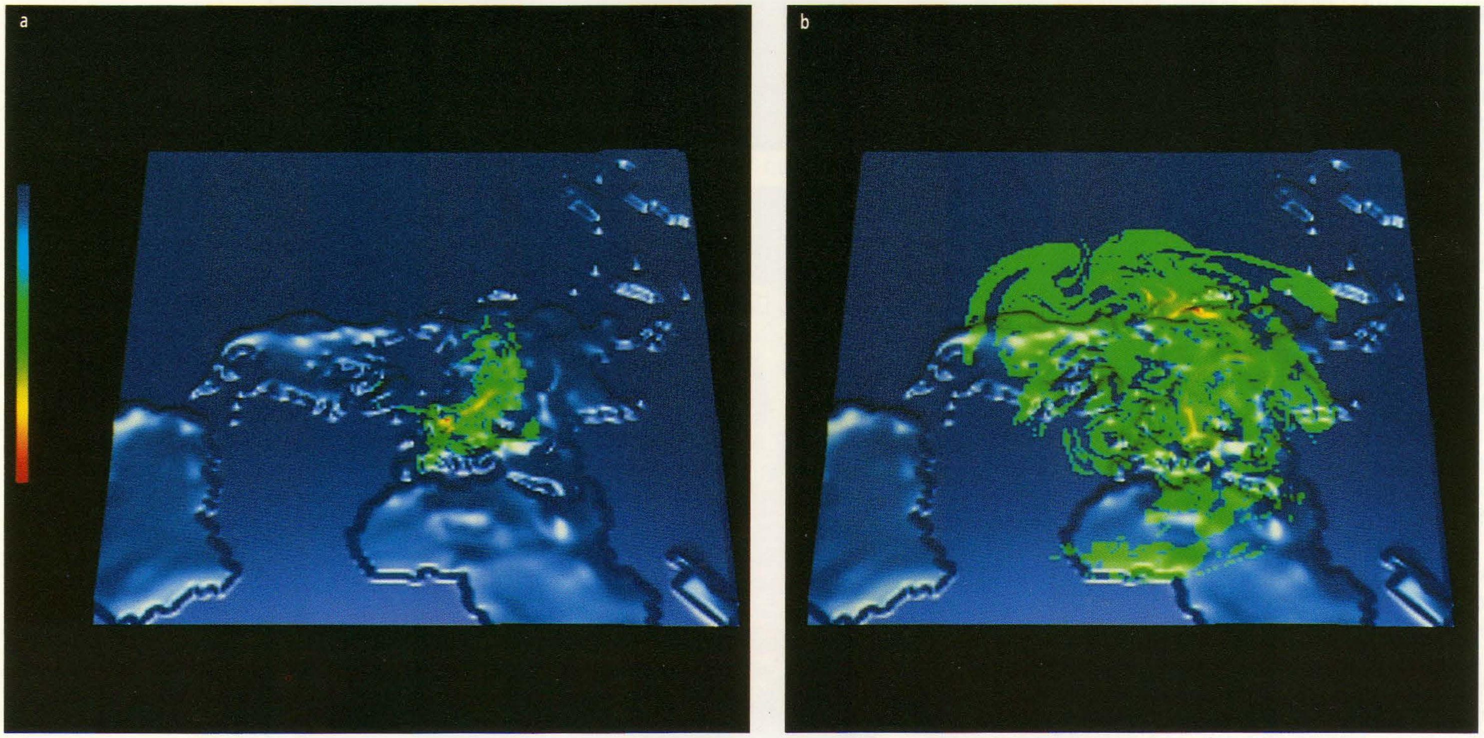
A film of the simulation results shows the sequence of events in a format that is easy to interpret. The first major phases of the film production began with a one-week project to read-in the three-dimenional cube array of cloud data and the associated land/sea and elevation maps. This first phase was attempted by decoding a 128-by-128-by-1 slice of data by writing a data conversion program to produce the necessary geometry and scalar function files for input to the MOVIE.BYU graphics package. The MOVIE.BYU program would read in the 128-by-128 point structure file of *x*, *y*, and *z* coordinate values that was constructed by the translation program.

The translation program first processed the land/sea mask data array to provide a zero value if the grid points were located on the sea and an artificial positive elevation bias of 300 meters for the grid points on land. The second part of the translation involved the addition of the scaled mountain elevation data from the elevation array map for those nonzero points in the land/sea mask. The scale factor was adjusted to produce a pleasing level of shadow from the mountain ranges in the output MOVIE.BYU image. The MOVIE.BYU program then accepted the scalar function field array of data that contained the scaled extracted slice of the cloud dispersion data from the 128-by-128-by-11 three-dimensional cloud data array.

After adjusting the color scales, 24 frames were filmed at +1° and then −1° increments to test the speed of animation necessary to enhance the mountain relief in the image. A 48-frame sequence was produced by shooting two frames of film for every image in six-hour increments to test the cloud temporal motion rates. This smaller scaled test film also pointed out the need for greater resolution in the final product to reduce the source block character look of the data created by the low resolution 128-by-128 grid. This information led us to refine the color scale further and to decide to increase the data size to a 180-by-180 grid. We also increased the length of time to 132 frames, which enabled us to extend our observations to include mixing in the atmosphere over the Pacific Ocean.

The production of the second film involved processing a 180-by-180-by-11 array per six-hour time step with 132 increments. This large 210-Mbyte data file involved the processing of three 6250 bpi magnetic tapes of the model output produced by the CMC's Cray system for the 2.5 hours of the simulation run. The program to translate the cloud grid array into the MOVIE.BYU point files and scalar function file had to be expanded to handle the larger 180-by-180-by-11 array. Also, the adjustment scaling was added to set up the proper color scale in the production of the scalar function file. The new translation program produced 132 sets of point and scalar function files that were archived on the CRAY X-MP/48 disk system. Then

Figure 4. Frames from the movie showing the Chernobyl cloud at $\sigma = 1$ (anemometer level) on May 1 (a) and May 4 (b), 1986.



each of these sets of files had to be processed by the MOVIE.BYU program for about 40 seconds, each with the instruction list settings programming the angle of view, rotation, and lighting to produce an image display for filming by the 16mm camera.

The images were displayed on a Ramtek 9460 frame buffer attached to the UNIBUS of Cray Research's Industry, Science & Technology Department's VAX/785 processor, which front-ended a CRAY X-MP/48 system at Cray Research's Mendota Heights computer center. The analog 30 Mhz video signal from the Ramtek 9460 frame buffer was fed to the high-resolution video monitor to produce a 1280-pixel-by-1024-line 24-bit color display and to a MATRIX 3000 film camera. Figure 4 shows the positions of the radioactive cloud on May 1 and May 4. The movie shows the cloud on a polar stereographic projection.

Verification of the model

The model results were verified against measurements from a network covering most of the Northern Hemisphere. The verification indicated substantial improvement when compared to the simple models used immediately after the Chernobyl accident. Verification of the model results was performed in this study only for the surface activities of ^{131}1. Data for the American receptors were derived from Larsen et al.[6] The Canadian measurements were provided by Tracy.[7] The network considered in this verification study consisted of 17 stations.

Statistical verification of the model is presented in Table 1 by correlation coefficient. The activities considered in the hemispheric scale simulation are of the order of millions of Bequerell/kg close to the source and fractions of mBq/kg in the remote locations. Because of this extreme variability of activities, the correlation coefficient was computed also for the logarithms of the activities. The correlation coefficient for the logarithmic case $\varrho_{\log}$ is displayed in Table 1.

New values of the correlation coefficient are between 0.73 and 0.95 with an average value of 0.88. The fact that the $\varrho_{\log}$ is so large shows that the magnitude of the activity field predicted by the model agrees very well with observation. The linear correlation coefficient is shown in the second column for control purposes.

Conclusions

The simulation described here demonstrates that the computing power now available and high-resolution numerical models make possible the accurate treatment of advection-dominated processes. The three-dimensional tracer model used in our computer animation work could be used as an emergency response system in the case of nuclear accidents. The operational version of the system is very flexible and could be executed in a diagnostic mode, using objectively analyzed meteorological fields, or in a predictive mode. The development of the research version of the tracer model as well as the work with the operational version of the system clearly indicate that the application of the Cray supercomputer was critical to ensure the promising results obtained with the models simulating the long-range transport of atmospheric tracers.

Station	$\varrho_{\log}$	ϱ
Barrow, Alaska	0.84	0.38
Beaverton, Oregon	0.91	0.47
Chester, New Jersey	0.95	0.94
Chiba, Japan	0.96	0.85
Fredericton, New Brunswick	0.84	0.76
Greenwood, Nova Scotia	0.94	0.81
Helsinki, Finland	0.73	0.70
Moosonee, Ontario	0.95	0.80
Paris, France	0.73	0.24
Rexburg, Idaho	0.93	0.52
Stockholm, Sweden	0.80	0.78
Winnipeg, Manitoba	0.91	0.65

Table 1. Correlation coefficient computed for logarithms of the activity versus previous correlation coefficient.

About the authors

Janusz Pudykiewicz is a research scientist at the Atmospheric Environment Service of Canada. He was born and educated in Poland and received an M.S degree in physics of the atmosphere from the University of Warsaw in 1978 and a doctorate from the Polish Academy of Sciences in 1982. He was a postdoctoral fellow at the Recherche en Prevision Numerique research center in Montreal, where he developed computer models of atmospheric chemistry.

William Samayoa is a senior graphics analyst in Cray Research's Industry, Science & Technology Department. He received a B.S. degree in physics from the College of St. Thomas in St. Paul, Minnesota, in 1968 and an M.S. degree in computer science from the University of Minnesota in 1974. He joined Cray Research in 1984 and works primarily on the conversion of image processing applications to Cray systems.

References

1. Pudykiewicz, J., "Simulation of the Chernobyl Dispersion with a 3-D Hemispheric Tracer Model," 1988, to appear in *Tellus*.
2. Robert, A., "A Semi-Lagrangian and Semi-implicit Numerical Integration Scheme for the Primitive Meteorological Equations," *Journal of the Meteorological Society of Japan*, Vol. 60, 1982, pp. 319-325.
3. Pudykiewicz, J. and A. Staniforth, "Some Properties and Comparative Performance of the Semi-Lagrangian Method of Robert in the Solution of the Advection-Diffusion Equation," *Atmosphere-Ocean*, Vol. 22, 1984, pp. 283-308.
4. Concus P., G. H. Golub, and D. P. O'Leary, "A Generalized Conjugate Gradient Method for the Numerical Solution of Eliptic Partial Differential Equations," Rep. No. STAN-CS-76-533, Computer Science Department, Stanford University, 1976.
5. Businger J. A., J. C. Wyngaard, Y. Izumi, and E. F. Bradley, "Flux-Profile Relationships in the Atmospheric Surface Layer," *Journal of the Atmospheric Sciences*, Vol. 28, 1970, pp. 181-189.
6. Larsen R., C. G. Sanderson, W. Riviera, and M. Zamichielli, "The Characterization of Radionuclides in North American and Hawaiian Surface Air and Deposition Following the Chernobyl Accident," Department of Energy Report EML-460, 1986.
7. Tracy, B., personal communication, 1988.

Building large-scale models to understand the AIDS epidemic

James M. Hyman, E. Ann Stanley, Stirling A. Colgate, and Scott P. Layne
Los Alamos National Laboratory, Los Alamos, New Mexico

Understanding and halting the spread of acquired immune deficiency syndrome (AIDS) is the most urgent public health objective today. Scientists at the Los Alamos National Laboratory (LANL) are studying the problem using Cray supercomputers and data from the Centers for Disease Control, the National Institutes of Health, the armed services, state and local health agencies, and other research organizations around the world. Using the lab's experience in data analysis, computer science, biology, and the modeling of other infectious diseases, we began developing a set of models that can elucidate how social and medical factors enhance or inhibit the AIDS epidemic. These insights also will be used to analyze the economic and social impact of the disease, including medical care costs and national productivity losses due to illness and death.

A job for large-scale simulation

Mapping the spread of AIDS is a significantly more complex project than studying the spread of other epidemics, because a culture's social structure plays a significant role in the spread of AIDS. While the spread of influenza occurs on a random basis, the spread of AIDS is determined by strong biasing — by intimate contact between partners that have similar risk behaviors. As a result, epidemiological models of AIDS must be of sufficient sociological complexity before they can portray reality.

Most predictions of the AIDS epidemic are based on simple exponential or polynomial extrapolations of current trends. These curve-fitting methods cannot be used reliably for long periods of time, nor can they provide understanding of the interactions that lead to the epidemic's spread. During the long asymptotic period after the infection with the human immunodeficiency virus (HIV), which causes AIDS, changes in the environment of viral transmission occur continuously, causing complex interactions.

Only models that are founded on the transmission methods of HIV can show how the early infection of high-risk groups, behavioral changes, and future medical advances will affect the spread of AIDS. By developing large-scale simulations on the Cray systems at LANL, researchers are creating a logical structure that organizes existing information about AIDS into a coherent framework and suggests needed information about a variety of topics including drug use, sexual activity, and the interactions between HIV and the immune system. The models provide qualitative insights, even when the data are lacking, and help prioritize data collection.

The large memory and speed of the Cray system play an important role in this project because they provide us with immediate feedback, enabling us to interact with the data as they are rendered as contour plots and lines on a graphics workstation. Running this job in batch mode would not allow us to adjust the parameters interactively as the Cray system does.

Although large-scale models of AIDS epidemiology are too complex to be run in reasonable time on today's supercomputers, formulating computer models defines a framework that enables researchers to make assumptions and parameter estimates. This allows researchers to methodically reduce the full model to submodels, each of which can be carefully investigated. Information obtained about the behavior of these submodels, such as sensitivity to variations in parameters, will show which aspects of the full model are crucial and which are not. For example, dividing the country into regions may be important only for understanding interregional spreading, or it may prove that the barriers to mixing provided by distance are crucial to slowing the spread of AIDS. All of these submodels fit into the same general framework, and LANL researchers are developing a computer program that allows the models to be changed easily to incorporate new ideas or to compare one version to another. The different submodels will be compared by how well they predict the future and model the past. This approach will help ensure the consistency of the submodels and improve our knowledge of the history of the AIDS epidemic.

A multitude of variables

Interaction networks that lead to the spread of sexually transmitted diseases are complex. Hundreds of thousands of differential equations must be solved to describe the path of the disease, making the power of a supercomputer necessary for accurate analysis. In future AIDS models we hope to include various population groups chosen according to age, frequency and type of sexual contact, drug use, ethnic or social group, population density, and region of the country. Types of sexual contact would include not only the heterosexual/homosexual differentiation, but also repeated contacts with such primary partners as spouses. Another important factor to consider is that unlike most viruses, the HIV may lie undetected in the body for many years. The disease therefore can be spread long before a carrier even suspects illness, making it difficult to determine how many people already are infected and how many future AIDS cases the nation may face. Models will be expanded to account for behavior changes that could result from AIDS-prevention education, for new knowledge about drug-use patterns, and for the consequences for age distribution in high-risk groups (for example, younger individuals may have more sexual contacts than older individuals).

Developing reliable models

As a first step toward developing a reliable model, we have used a simple deterministic model to

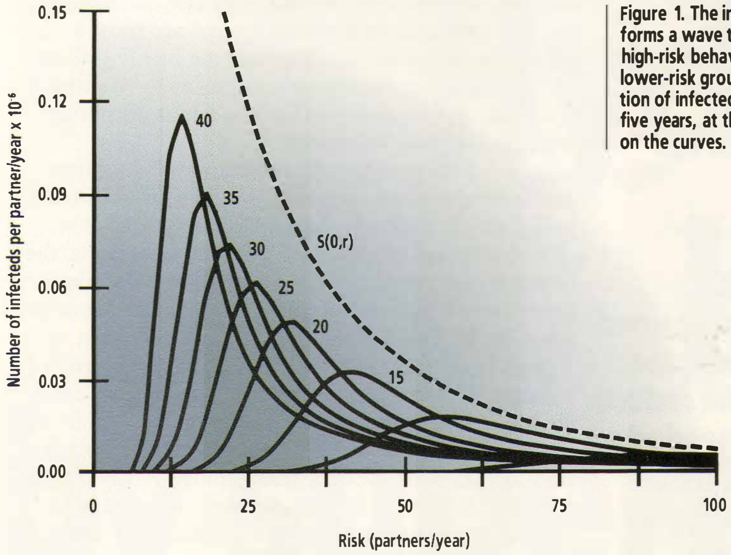


Figure 1. The infected population forms a wave that sweeps from high-risk behavior groups into lower-risk groups. The distribution of infecteds is shown every five years, at the times marked on the curves.

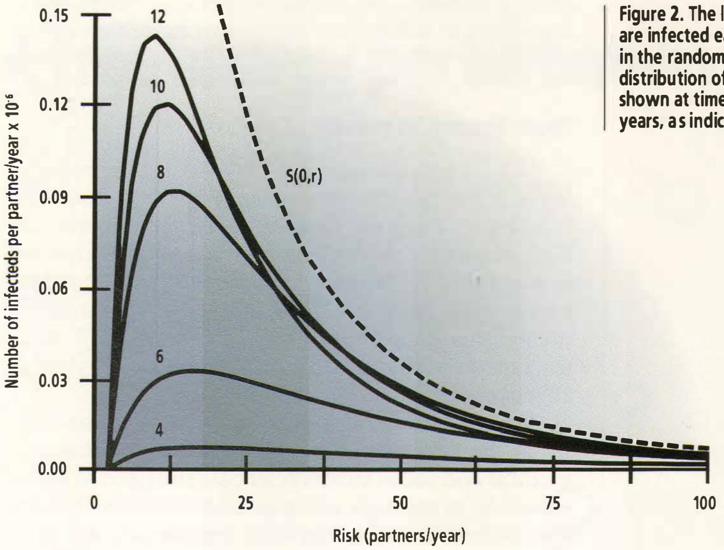


Figure 2. The low-risk susceptibles are infected early in the epidemic in the random mixing model. The distribution of the infecteds are shown at times 4, 6, 8, 10, and 12 years, as indicated on the curves.

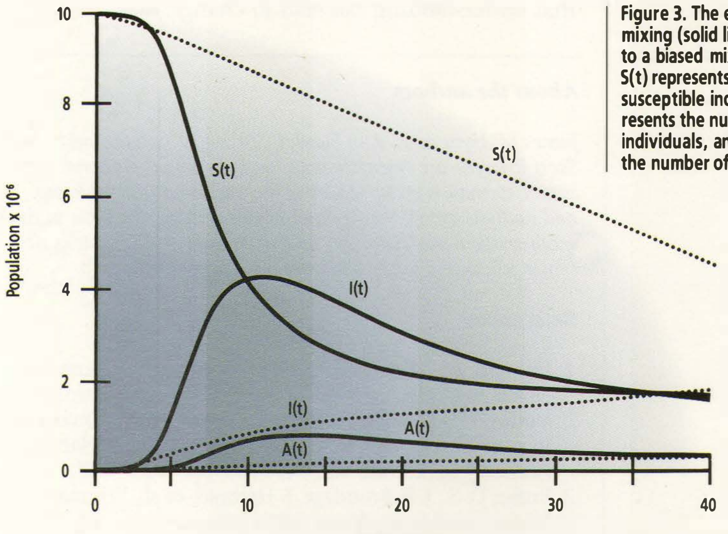


Figure 3. The effects of random mixing (solid lines) as compared to a biased mixing (dotted lines). S(t) represents the number of susceptible individuals, I(t) represents the number of infected individuals, and A(t) represents the number of AIDS cases.

explore the impact of various plausible results for the infectivity as the time since infection increases. These calculations, which use an average-risk behavior, point out the importance of measuring variability of the infectiousness during the disease.

We have then used models that stratify the population according to the number of sexual partners per year and have compared random partner choice with a strong bias of likes prefer likes. The two mixing patterns result in radically different epidemics. The difference indicates that before accurately predicting the AIDS epidemic will be possible, much more must be known about the interactions that spread the AIDS virus. The number of sexual partners that people have, the partner-selection process, and the amount and type of contacts between partners must be understood and correlated with sociological information about the partners, such as how many partners each person has. Similarly, patterns of needle-sharing by drug users and the effect of this drug abuse on sexual behavior strongly affect this epidemic.

Our models have shown that the amount of sexual contact and needle-sharing between high-activity and lower-activity individuals determines who becomes infected and the speed with which the epidemic progresses. If little mixing occurs between these groups, then nearly all of the individuals in high-risk are infected before the infection moves to the lower-risk groups. However, if a mixing is large, many more lower-risk individuals will become infected in the early stages of the epidemic (Figure 1). AIDS moves much faster when mixing is large because a larger pool of lower-risk individuals feeds the epidemic. In a model in which partners are chosen randomly, regardless of their partner-change rate, the total number of infected low-risk individuals quickly exceeds the number of infected high-risk individuals. The distribution of the infected population by the random mixing model is shown in Figure 2 for a sequence of times. This result is contrary to experience and reflects an urgent need to collect and analyze the information on mixing patterns to estimate critical model parameters. Figure 3 shows differences between random and biased mixing patterns.

In our analysis we have focused on the initial growth of the epidemic. If we are to predict where this epidemic is going, we must fully understand its transient dynamics, including the response to changes in the environment of the epidemic. The epidemic will not reach an equilibrium epidemic state for a long time, partly because of the long conversion times from infection to AIDS, during which a person can transmit the virus. The time factor makes AIDS unlike many other epidemics, including measles, gonorrhea, and syphilis. Another reason the epidemic will not reach an equilibrium epidemic state for a long time is that medical advances and changes in lifestyle will modify the epidemic greatly. Education programs are being launched to encourage people to reduce their number of sexual partners, and to promote practices such as using condoms, nonoxynol-9, and sterile needles. The infectiousness and susceptibility of high-risk individuals in the heterosexual community may be significantly reduced if programs are initiated to identify and treat other sexually transmitted diseases quickly. Models can be used to investigate the effects of each of these programs on the course of the epidemic only if they can capture the transients of the epidemic.

In developing models, we also must decide which questions to answer. If public health officials are to attack this epidemic efficiently, then they need to know which groups of people are most at risk of infection. Models that distinguish between behavioral groups may help predict where the infection is likely to spread. Our risk-based model is aimed at answering this question.

In our preferential-mixing model we can choose parameters that ensure that AIDS cases in the numerical simulations match the past history in the United States. Many other reasonable models also can fit these cases quantitatively, but may predict very different futures. Quantitatively matching past AIDS cases is not, therefore, sufficient to distinguish between models. Other data such as the risk behavior of AIDS cases or the infected population must be considered. Qualitative discrepancies between AIDS cases and the model need to be explained; for example, models with exponential growth do not fit the current United States AIDS case data.

Models must be compared with data from studies on seroprevalence and risk behavior versus infection. For example, we plan to compare our preferential mixing model to the San Francisco Hepatitis B study. This study of sexually active homosexual men was started in 1978. Numerical data were collected from steady and nonsteady sexual partners, including information about the number of contacts per partner. A series of serum samples was stored from a subset of men. Many of these samples have been tested for HIV, and so a correlation between sexual behavior and time of infection can be made and compared with our model. Inconsistencies will be seen, and the model must be revised to account for them. Limitations of the data will influence the accuracy of the models. Many of the sensitive parameter values, such as the magnitude and variability of infectiousness, will be known accurately only after years of study.

Additional insights

Unless significant medical breakthroughs change the course of the epidemic, the large number of currently infected people (.75 to 1.5 million in the United States) will increase, and most eventually will display AIDS symptoms. As the epidemic progresses, the need for accurate predictions of health care needs and the full economic impact of the epidemic will increase.

Many more Americans are apt to become ill and die from this disease before either a vaccine or an effective treatment become available.[1] Most of these people will be in the 20-40 age group.[2] The loss of productivity and the burden on America's health care system will be enormous. In the long run, even the military preparedness of the United States and other countries may be affected. The more accurately we can estimate the extent and distribution of the epidemic, the better prepared the nation will be to deal with the care of those affected.

So little is known about the spread of sexually transmitted diseases that the next groups most at risk are not intuitively obvious. We hope that our modeling will help ascertain which groups are most at risk before those groups actually become highly infected. For example, a controversy is raging as to whether this epidemic can be sustained in a group with purely heterosexual risk behaviors.

The insights gained from large-scale models will reduce the time needed to determine which strategies are effective.

Preliminary calculations show the safety of the heterosexual community is largely an illusion based on the different time scales associated with different sexual behavior patterns. In these runs, the heterosexual (less active) susceptible trends mimic the current homosexual trends with a time lag of only a few years. This is true, even though a fraction of 1 percent of the heterosexual population currently is infected. The spreading of HIV by heterosexual contacts already can be seen in the most recent figures from the Centers for Disease Control. In the adult black population, heterosexually transmitted AIDS already accounts for 12 percent of all cases. In military test recruits, the fractions of infected males and females are nearly equal.[2] Besides the obvious desire to predict accurately the future numbers of the infected, ill, and dead, a primary goal of this project is to investigate the effects of behavioral changes, treatment methods, and vaccines on the future course of the epidemic.

From theory to reality

Epidemiological AIDS models are producing as many questions as answers. We may find that a risk factor previously thought important has relatively little impact on the forecasts, or we may find that a new category of information must be collected to obtain meaningful predictions from the models. Ultimately, the insights gained from large-scale models will reduce the time needed to determine which strategies are effective and which are not.

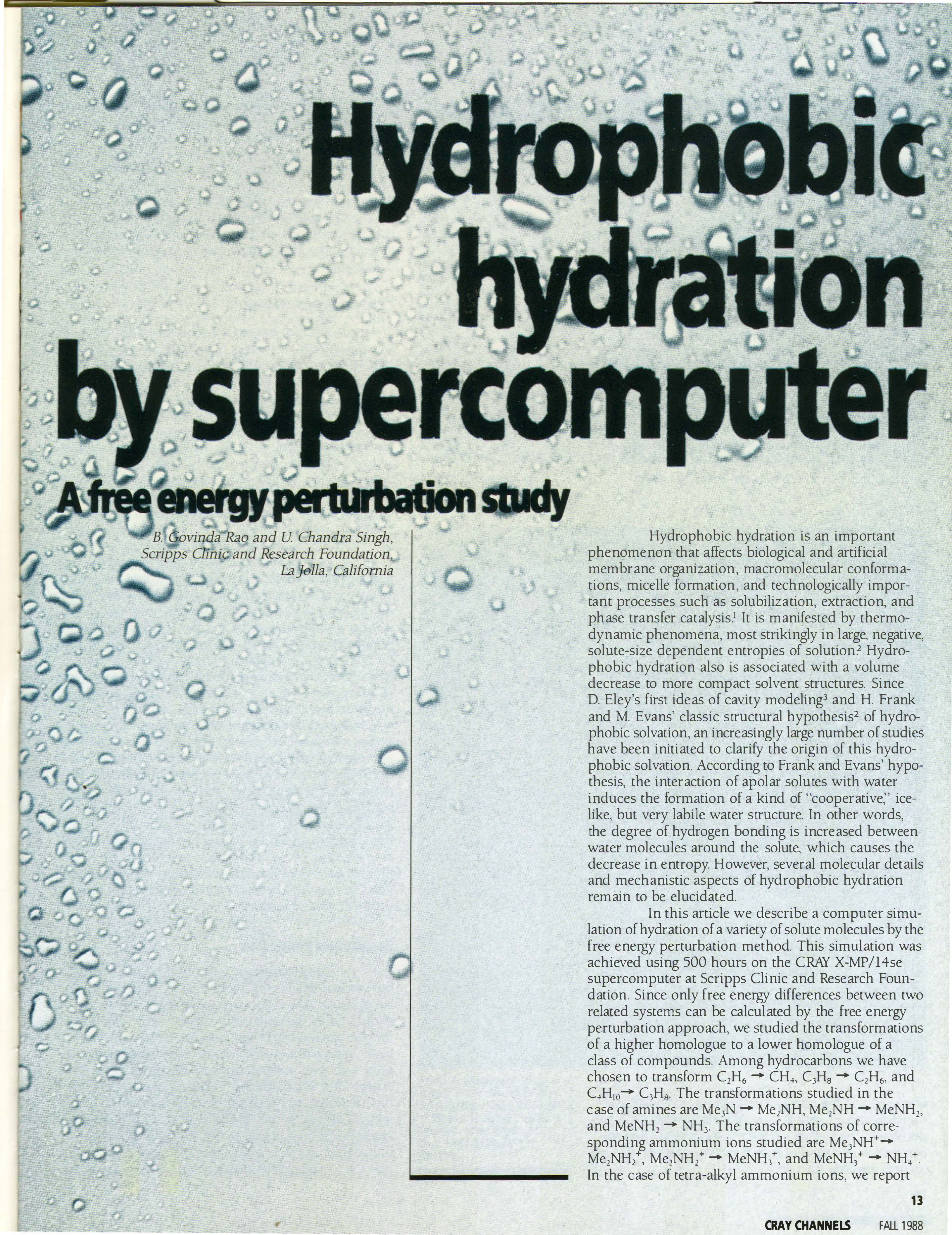
Fighting AIDS may be an effort as challenging as the fight against cancer because of the many medical and social factors scientists still need to understand. AIDS research will require an interdisciplinary approach and can benefit from the interactions of large-scale supercomputing centers, such as LANL. Our goal is to show that theory can elucidate reality, and that understanding can lead to change. ▬

About the authors

James M. Hyman, E. Ann Stanley, Stirling A. Colgate, and Scott P. Layne are researchers at the Los Alamos National Laboratory. Hyman is group leader of the mathematical modeling and analysis group, Stanley and Layne are staff scientists in the mathematical modeling and analysis group, and Colgate is a senior fellow and group leader of the astrophysics group.

References

1. Hahn, B. H., M. A. Gonda, G. M. Shaw, et al., "Genomic Diversity of the Acquired Immune Deficiency Syndrome Virus HTLV-III: Different Viruses Exhibit Greatest Divergence in their Envelope Genes," Proceedings from the National Academy of Science, Vol. 82, 1985, pp. 4813-4817.
2. Burke, D. S., J. F. Brundage, J. Herbold, et al., "Human Immunodeficiency Virus (HIV) Infections Among Civilian Applicants for the United States Military Service," *October 1985-March 1986: Demographic Factors Associated with Seropositivity*, unpublished report, November 1986.



Hydrophobic hydration by supercomputer

A free energy perturbation study

B. Govinda Rao and U. Chandra Singh,
Scripps Clinic and Research Foundation,
La Jolla, California

Hydrophobic hydration is an important phenomenon that affects biological and artificial membrane organization, macromolecular conformations, micelle formation, and technologically important processes such as solubilization, extraction, and phase transfer catalysis.[1] It is manifested by thermodynamic phenomena, most strikingly in large, negative, solute-size dependent entropies of solution.[2] Hydrophobic hydration also is associated with a volume decrease to more compact solvent structures. Since D. Eley's first ideas of cavity modeling[3] and H. Frank and M. Evans' classic structural hypothesis[2] of hydrophobic solvation, an increasingly large number of studies have been initiated to clarify the origin of this hydrophobic solvation. According to Frank and Evans' hypothesis, the interaction of apolar solutes with water induces the formation of a kind of "cooperative," ice-like, but very labile water structure. In other words, the degree of hydrogen bonding is increased between water molecules around the solute, which causes the decrease in entropy. However, several molecular details and mechanistic aspects of hydrophobic hydration remain to be elucidated.

In this article we describe a computer simulation of hydration of a variety of solute molecules by the free energy perturbation method. This simulation was achieved using 500 hours on the CRAY X-MP/14se supercomputer at Scripps Clinic and Research Foundation. Since only free energy differences between two related systems can be calculated by the free energy perturbation approach, we studied the transformations of a higher homologue to a lower homologue of a class of compounds. Among hydrocarbons we have chosen to transform $C_2H_6 \rightarrow CH_4$, $C_3H_8 \rightarrow C_2H_6$, and $C_4H_{10} \rightarrow C_3H_8$. The transformations studied in the case of amines are $Me_3N \rightarrow Me_2NH$, $Me_2NH \rightarrow MeNH_2$, and $MeNH_2 \rightarrow NH_3$. The transformations of corresponding ammonium ions studied are $Me_3NH^+ \rightarrow Me_2NH_2^+$, $Me_2NH_2^+ \rightarrow MeNH_3^+$, and $MeNH_3^+ \rightarrow NH_4^+$. In the case of tetra-alkyl ammonium ions, we report transformations of $Bu_4N^+ \rightarrow Pr_4N^+$, $Pr_4N^+ \rightarrow Et_4N^+$, and $Et_4N^+ \rightarrow Me_4N^+$. For comparison we also report the transformations of corresponding neutral hydrocarbons: $Bu_4C \rightarrow Pr_4C$, $Pr_4C \rightarrow Et_4C$, $Et_4C \rightarrow Me_4C$, and $Me_4C \rightarrow CH_4$. Finally, the transformations of some aromatic compounds, namely, aniline → benzene, phenol → benzene, and toluene → benzene, are reported.

The free energy perturbation method

The free energy perturbation method is based on the statistical perturbation theory developed initially by R. Zwanzig.[4] This method has been implemented into the framework of molecular dynamics by U. Singh et al.[5] A brief description of the method is given below.

The total Hamiltonian as described in this method may be separated into two parts,

$$H = H_0 + H_1$$

where H_0 is the Hamiltonian of an unperturbed system and H_1 is the perturbation. The Gibbs free energy contribution due to the perturbation is given by

$$G_1 = -1/\beta \ln < \exp(-\beta H_1) > 0$$

where $\beta = \frac{1}{RT}$ and the average of $\exp(-\beta H_1)$ are computed over the unperturbed ensemble of the system. Obtaining the ΔG between two states requires defining the perturbed group Hamiltonian for states A and B as

$$H_\lambda = \lambda H_A + (1 - \lambda) H_B \qquad 0 \leq \lambda \leq 1$$

where H_A is the Hamiltonian for A, and H_B is that for B, when $\lambda = 1$, $H_\lambda = H_A$ and when $\lambda = 0$, $H_\lambda = H_B$. At intermediate values of λ, the state is a hypothetical mixture of A and B. This type of coupling ensures the smooth conversion of state A into state B. Even if the two states are significantly different in chemical structure, the conversion can be made smooth by choosing smaller values of λ and by allowing the system to equilibrate sufficiently longer after each perturbation. If range of λ is divided into N windows, $\{\lambda_i, i = 1, N\}$, at each window λ_i, the solute state is perturbed to λ_{i+1} and λ_{i-1} states and the free energy difference between the states of A and B is computed by the summation over all of the windows as

$$\Delta G = \sum_{i=1}^{N} G_I (\lambda_i)$$

To eliminate some sampling difficulties during the conversion of a polar solute to a nonpolar solute in water in which a large change exists in the molecular volume, the free energy is decomposed into two components called electrostatic and van der Waals components. If the total energy is written as

$$E_{tot} = E_{badh} + E_{ele} + E_{vdw}$$

where E_{badh} is the energy due to bond, the angle and dihedral and E_{ele} and E_{vdw} are the electrostatic and van der Waals interaction energies respectively. Then the conversion of state A to state B can be achieved through an intermediate state A'.

$$A \rightarrow A' \rightarrow B$$

$$\Delta G_{AB} = \Delta G_{AA'} + \Delta G_{A'B}$$

In state A' the solute has the same intramolecular and van der Waals parameters as in state A. However state A' assumes the charge distribution of state B. Thus $\Delta G_{AA'}$ corresponds to the electrostatic contribution to the free energy difference, whereas $\Delta G_{A'B}$ corresponds to the van der Waals contribution to the free energy difference.

Computational details

The calculations have been performed using the AMBER (Version 3.1) program on the CRAY X-MP/14se computer system.[6] The molecules were built from the known geometric information. These molecules were solvated by placing them at the center of a rectangular box containing three cubes of 216 water molecules, and discarding any water molecule that was farther than a cutoff distance of 12.0 angstroms along x, y, or z direction from any atom of the solute. The partial charges of these molecules necessary for the simulation were calculated from the electrostatic potential around the molecules obtained by the ab initio program Gaussian –82 (QUEST) with a 6-31G* basis set. The AMBER forcefield parameters were used.

Each system was minimized first by conjugate gradient method in three stages; first only the solvent was minimized with periodic boundary conditions for a maximum of 800 cycles, then the whole system was minimized for another 800 cycles, and finally the whole system was minimized using SHAKE for 100 cycles. The system initially was equilibrated for 6.5 ps at constant temperature (300° K) and pressure (1 atmosphere) using a time step of 0.002 ps. The mutation was achieved in two stages and the molecules were mapped one onto the other by direct atom-to-atom assignment. As already described in the previous section, first the charges were mutated, then the van der Waals parameters were mutated, thus separately calculating the electrostatic and van der Waals contributions to free energy change. For the electrostatic run, 21 windows were used. At each window the system was equilibrated for 1.0 ps and the data were collected over the next 1.0 ps with a time step of 0.002 ps. For the van der Waals run, the simulations were divided into 201 windows with 0.2 ps of equilibration and 0.2 ps of data collection.

Results and discussion

The electrostatic contribution, ΔG_{ele}, and the van der Waals contribution, ΔG_{vdw}, and their sum are given in Table 1 for each simulation. The experimental values available in the literature also are listed for comparison in the last column. It may be noted from Table 1 that the calculated free energies agree quite well with the experimental values in all cases except for normal alkanes. The discrepancy between the calculated and the experimental values for alkanes may be due to the fact that the expected free energy changes are comparable to the fluctuations expected in these simulations. It also may be noted from Table 1 that the van der Waals contribution to the total free energy dominates in all cases except for neutral amines and aromatic compounds.

The changes in ΔG_{ele} with λ for the simulations do not show any discernible pattern. It is probably dictated by the partial charge distributions of the solute molecule at the beginning and the end of mutation. The changes in ΔG_{vdw}, on the other hand, show a definite pattern. The variations of ΔG_{vdw} with λ are shown in Figure 1 for a representative conversion from each class of compounds. In the case of ammonium ions ΔG_{vdw} decreases, in the case of polar compounds it increases, and in the case of neutral hydrocarbons it initially decreases and then slightly increases. The initial decrease in free energy, with the mutation of van der Waals interaction parameters observed for the cases of ammonium ions, suggests that the solvent water molecules around the solute are held tightly by the solute due to the presence of unit positive charge on the solute. This results in configurations wherein the solute and the solvent are in the repulsive regions of their interaction potential surface, which, therefore, leads to negative free energies as the solute shrinks in size. The same trend is seen initially for alkanes, but the effect is not so amplified because these molecules are neutral. Further, the free energy change shows an increasing trend for alkanes at higher values of λ, suggesting that the solvent molecules do not continue to get pulled toward the solute beyond a certain a value of λ. The variation of free energy with λ for amines and aniline → benzene conversion shows almost linear increase without any prominent dip. This may be due to the presence of groups capable of forming directional hydrogen bonding with the solvent. A detailed analysis of these results will be published elsewhere.

About the authors

B. Govinda Rao received a Ph.D. degree in solid and structural chemistry from the Indian Institute of Science, Bangalore, India. He is a postdoctoral research associate at Scripps Clinic and Research Foundation. U. Chandra Singh received a Ph.D. degree from Madurai University, Madurai, India. After several years of postdoctoral experience at the Indian Institute of Science, Bangalore, and at the University of California, San Francisco, he joined the faculty of the Department of Molecular Biology at Scripps Clinic and Research Foundation.

References

1. Dogonadze, R. R., E. Kalman, A. A. Kornyshev, and J. Ulstrup, Eds., *The Chemical Physics of Solvation*, Part A, Theory of Solvation, Elsevier: Amsterdam, 1985.
2. Frank, H. S. and M. W. Evans, *The Journal of Chemical Physics*, Vol. 13, 1945, p. 507.
3. Eley, D. D. *Transactions of the Faraday Society*, Vol. 35, 1939, p. 1421.
4. Zwanzig, R. W., *The Journal of Chemical Physics*, Vol. 22, 1954, p. 1420.
5. (a) Singh, U. C., F. K. Brown, P. A. Bash, and P. A. Kollman, *Journal of the American Chemistry Society*, Vol. 109, 1987, p. 1607.
   (b) Bash, P. A., U. C. Singh, R. Langridge, and P. A. Kollman, *Science*, Vol. 236, 1987, p. 564.
6. AMBER (version 3.1) is a fully vectorized version of AMBER (version 3.0) by U. C. Singh, P. K. Weiner, J. W. Caldwell, and P. A. Kollman, University of California, San Francisco, 1986. AMBER (version 3.1) also includes coordinate coupling and intra/inter decomposition.

Solute	Reference	ΔG_{ele}	ΔG_{vdw}	Total	Expt
C_2H_6	CH_4	-0.09 ± 0.01	-0.33 ± 0.02	-0.42 ± 0.03	0.17
C_3H_8	C_2H_6	0.01 ± 0.00	0.02 ± 0.01	0.03 ± 0.01	-0.12
C_4H_{10}	C_3H_8	0.05 ± 0.00	0.01 ± 0.00	0.06 ± 0.01	-0.12
Me_4C	CH_4	-0.05 ± 0.01	-0.81 ± 0.07	-0.86 ± 0.08	-0.50
Et_4C	Me_4C	0.05 ± 0.00	0.35 ± 0.01	0.40 ± 0.01	0.28
Pr_4C	Et_4C	0.44 ± 0.01	-1.03 ± 0.01	-0.59 ± 0.02	
Bu_4C	Pr_4C	0.41 ± 0.01	-0.92 ± 0.02	-0.51 ± 0.03	
$MeNH_2$	NH_3	-0.21 ± 0.00	0.28 ± 0.13	0.07 ± 0.13	0.27
Me_2NH	$MeNH_2$	-2.21 ± 0.01	0.28 ± 0.07	-1.93 ± 0.08	-0.27
Me_3N	Me_2NH	-1.47 ± 0.01	0.30 ± 0.05	-1.17 ± 0.06	-1.07
$MeNH_3^+$	NH_4^+	-3.35 ± 0.02	-5.73 ± 0.10	-9.08 ± 0.12	-7.30
$Me_2NH_2^+$	$MeNH_3^+$	1.36 ± 0.04	-7.71 ± 0.09	-6.35 ± 0.13	-6.40
Me_3NH^+	$Me_2NH_2^+$	-0.11 ± 0.00	-5.76 ± 0.05	-5.87 ± 0.05	-7.00
Me_4N^+	NH_{4+}	-0.65 ± 0.00	-29.17 ± 0.52	-29.82 ± 0.52	-31.70
Et_4N^+	Me_4N^+	-0.64 ± 0.01	-6.10 ± 0.25	-6.74 ± 0.26	-7.00
Pr_4N^+	Et_4N^+	-1.33 ± 0.01	-4.26 ± 0.10	-5.59 ± 0.11	
Bu_4N^+	Pr_4N^+	0.35 ± 0.02	-3.10 ± 0.06	-2.75 ± 0.08	
$C_6H_5CH_3$	C_6H_6	-0.32 ± 0.01	-0.10 ± 0.02	-0.42 ± 0.03	-0.86
C_6H_5OH	C_6H_6	2.52 ± 0.01	1.00 ± 0.00	3.52 ± 0.01	2.20
$C_6H_5NH_2$	C_6H_6	1.05 ± 0.01	1.16 ± 0.01	2.21 ± 0.02	1.68

Table 1. Experimental and calculated free energy changes. All values are in kcal/mol.

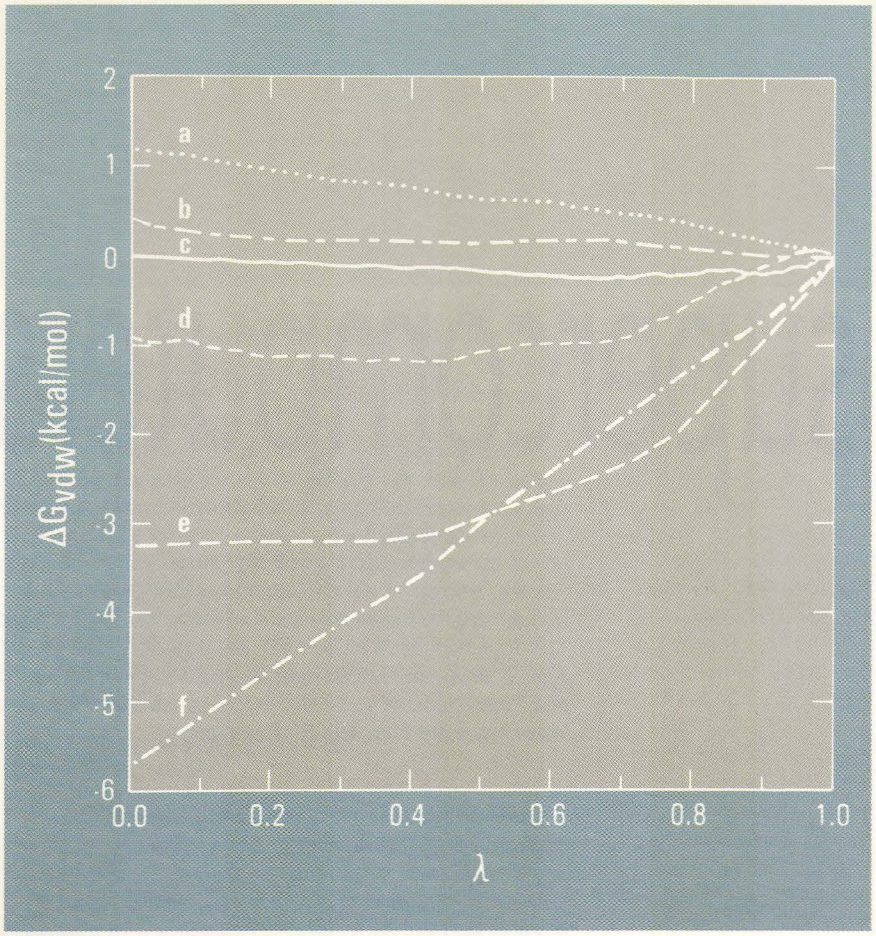
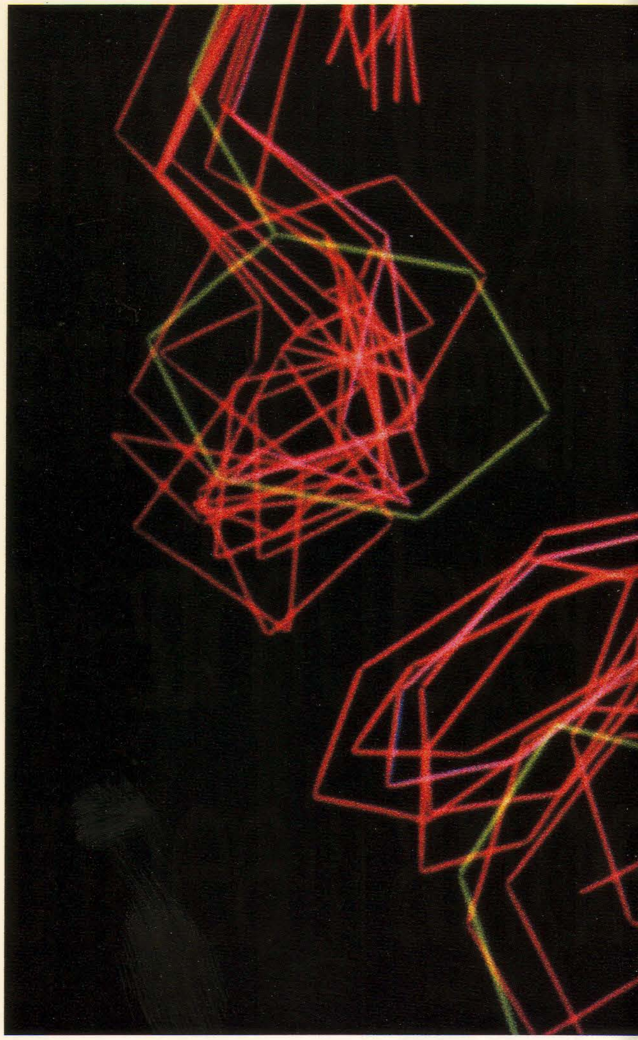


Figure 1. Variation of ΔG_{vdw} with λ for the conversion of (a) $C_6H_5NH_2 \rightarrow C_6H_6$, (b) $Me_3N \rightarrow Me_2NH$, (c) $C_4H_{10} \rightarrow C_3H_8$, (d) $Bu_4C \rightarrow Pr_4C$, (e) $Bu_4N^+ \rightarrow Pr_4N^+$ and (f) $Me_3NH^+ \rightarrow Me_2NH_2^+$.

Crystallographic refinement by simulated annealing on supercomputers



Axel T. Brünger, The Howard Hughes Medical Institute and Department of Molecular Biophysics and Biochemistry, Yale University, New Haven, Connecticut

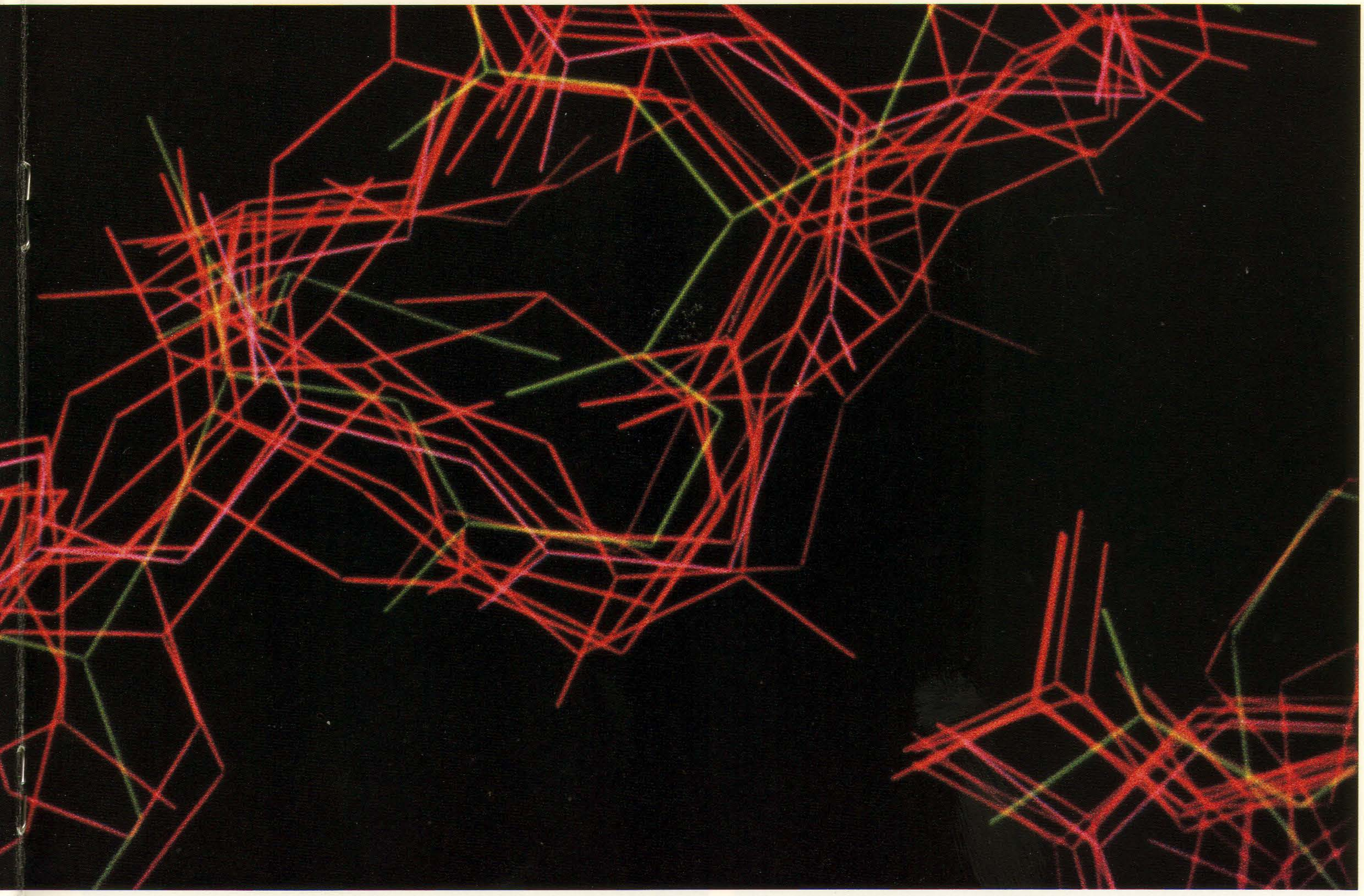
The ability to determine the structures of crystallized proteins is an important tool for protein engineering and rational drug design. With the increased availability of fast diffraction data collection devices and the successful crystallization of increasingly large macromolecular complexes, crystallographic refinement becomes a bottleneck for structure determination. As a result, more efficient refinement processes are needed. A major portion of the research in our laboratory focuses on developing computational techniques for automated crystallographic structure determination and refinement. Using a CRAY X-MP system from the Pittsburgh Supercomputing Center, we have developed a new refinement technique that simplifies the manual refitting of structures.

Crystallographic refinement

The goal of crystallographic refinement is to achieve a best fit between observed data and simulated data of an atomic model, while maintaining good stereochemistry and packing interactions of the atomic model. In particular, one wants to minimize the difference between the observed ($|F_{\mathrm{obs}}(\vec{h})|$) and calculated ($|F_{\mathrm{calc}}(\vec{h})|$) structure factor amplitudes, which usually is expressed as a weighted sum of the residual

$$\sum_{\vec{h}} W_{\vec{h}} \left(|F_{\mathrm{obs}}(\vec{h})| - k|F_{\mathrm{calc}}(\vec{h})|\right)^2 \qquad (1)$$

where $\vec{h} = (h, k, l)$ are indices referring to the reciprocal lattice points of the crystal. The conventional refinement process is trapped easily in a local minimum and does not correct residues that are misplaced by more than about one angstrom, making human intervention necessary. This involves refitting the model structure to electron density maps with interactive computer graphics.[1] The procedure must be repeated quite often and can take several months for just one protein structure determination.



Simulated annealing

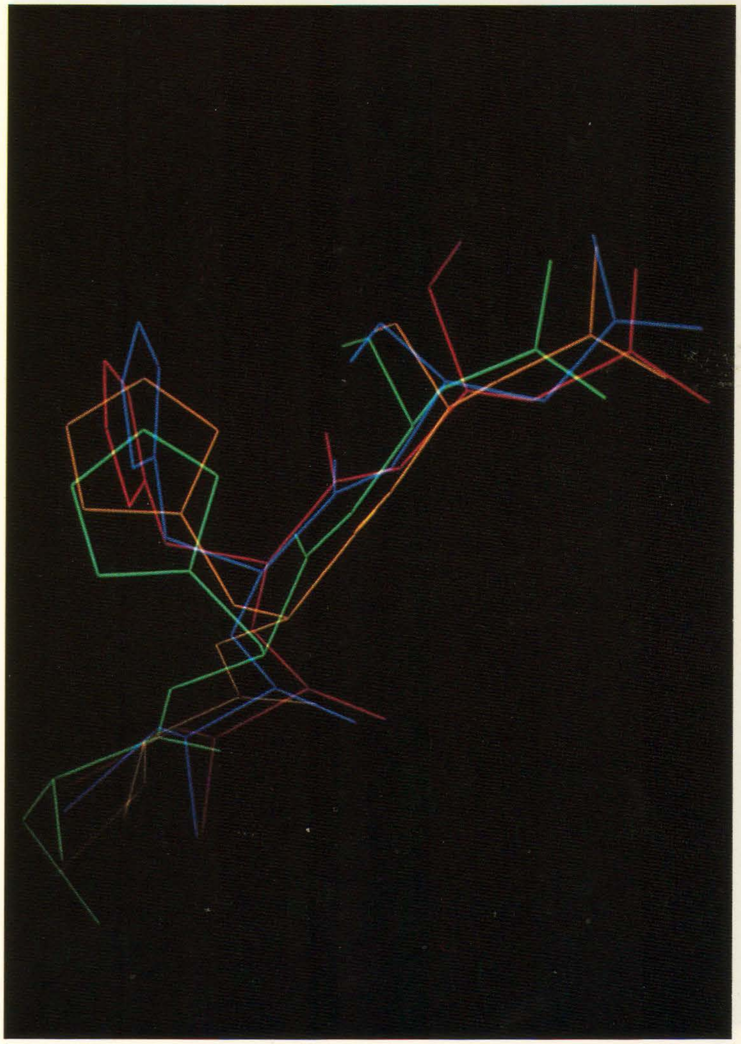
The new refinement technique we have developed, referred to as SA-refinement, is an approach related to the simulated annealing technique used for nonlinear optimization problems.[2] Crystallographic refinement can be understood as a nonlinear optimization problem with the aim of finding a minimum close to the global minimum of a target function $f(X) = f(x_1, x_2, \ldots, x_n)$ containing the residual sum (Equation 1), and stereochemical and other interactions of the macromolecule. The quantities $x_1, x_2, \ldots, x_n$ represent the variables of the system, such as the atomic positions or individual atomic temperature factors.

The simulated annealing technique usually consists of simulating the many-parameter system by a Monte Carlo algorithm that involves trial moves of $x_1, x_2, \ldots, x_n$. Gradient descent methods, such as restrained least-squares refinement, accept only moves that reduce $f(X)$ and, therefore, cannot escape from local minima. Monte Carlo algorithms, on the other hand, accept certain moves that increase or decrease $f(X)$. A move that increases $f(X)$ is accepted with a probability given by $\exp(-\Delta f(X)/kT)$ where $\Delta f(X)$ is the difference between the values of $f(X)$ before and after the trial move; T is the "temperature" of the system; and k is Boltzmann's constant. In fact, T should be understood not as a physical temperature but rather as a control parameter that determines whether the system can escape certain local minima. Thus, very high values of T may have to be introduced if the barriers between local minima are large.

The success of simulated annealing depends on the "annealing" schedule, which determines how the temperature is modified during the simulation. Initially, the temperature is kept very high and the system is then "annealed" by slowly reducing the temperature. In other words, a coarse search is carried out at high temperatures and a local minimum is approached during the cooling stage. This procedure may have to be repeated to reach the global minimum.

Simulated annealing through molecular dynamics

Methods to introduce simulated annealing into crystallographic refinement are explained in references 3 through 5. A major difference from the optimization problems discussed in S. Kirkpatrick et al.[2] is that one optimizes a single unit with internal degrees of freedom representing a macromolecule rather than a fluidlike system consisting of many identical subunits. A direct application of the Metropolis algorithm to macromolecules turns out to be inefficient if all degrees of freedom are included, as the covalent bonds of the system will lead to rejection of most steps



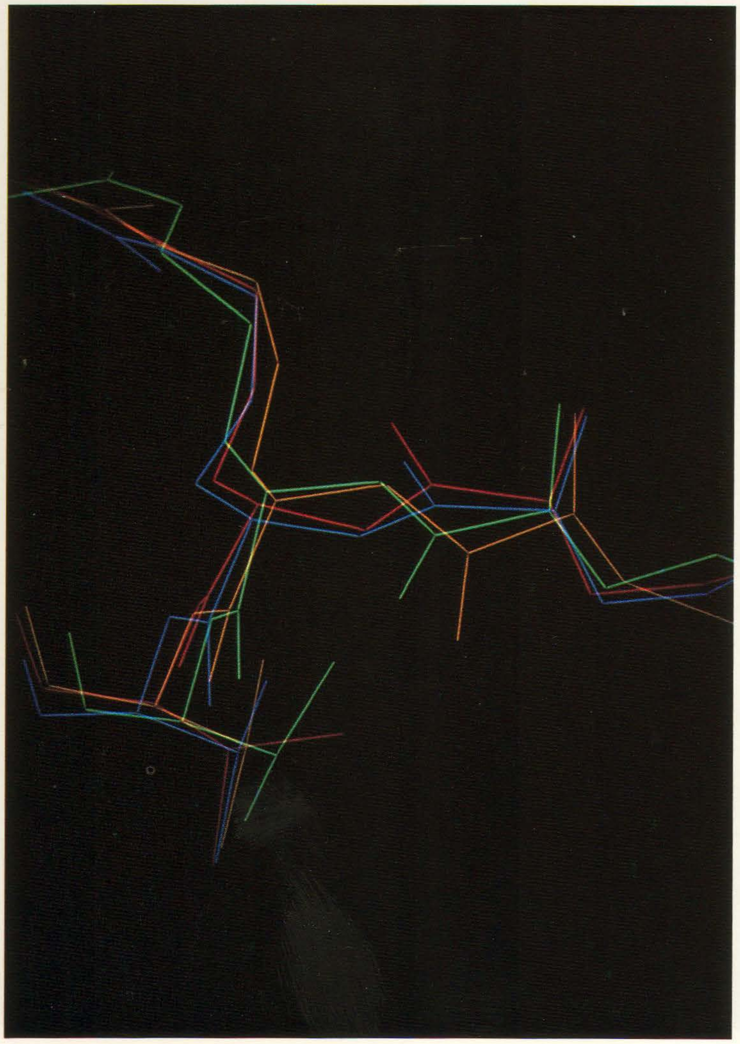


Figure 1 (left). Segment (Cys-192, His-193, Asn-194) of AATase. Superimposed are the manually refined structure (blue lines), the final SA-refined structure (red lines), the restrained least-squares refined structure (orange lines), and the initial structure (green lines).

Figure 2 (right). Segment (Glu-265, Arg-266, Val-267) of AATase. Superimposed are the manually refined structure (blue lines), the final SA-refined structure (red lines), the restrained least-squares refined structure (orange lines), and the initial structure (green lines).

taken by the algorithm. Instead, molecular dynamics can be used to follow the gradients of a target function and to introduce a temperature into the system to escape from local minima. The target function consists of an empirical potential energy describing the internal energy of the molecule and the crystallographic residual (Equation 1) describing the diffraction data. The resulting effective potential energy is similar to the function used in refinement by least-squares minimization.[6] Refinement by simulated annealing with molecular dynamics proceeds in the same way as with the Monte Carlo algorithm: a heating stage and a cooling stage occur, with heating and cooling possibly repeated several times.

Task	VAX 8700	Convex C-1/xp	CRAY X-MP/48 one processor
Electron density calculation	56	41.3	2.97
P_1 –FFT	373.2	21.8	1.67
Symmetry operations, factoring	34	8.1	0.64

Table 1: Run times in seconds on various computers for the calculation of structure factors and first derivatives with respect to atomic coordinates for aspartate aminotransferase at 2.8-angstrom resolution.

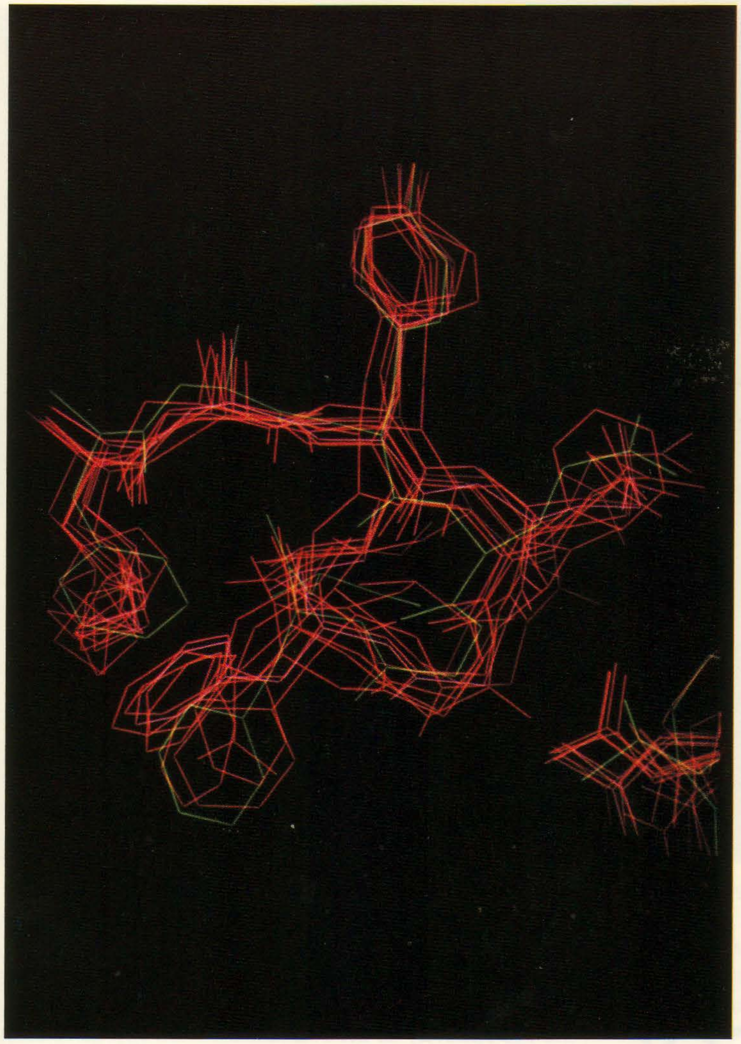
Vectorization of the algorithm

The most CPU-time-consuming step in SA-refinement is the evaluation of the structure factors of the atomic model and the first derivatives with respect to the atomic coordinates of the system. Evaluation of the atomic electron density on a finite grid followed by fast Fourier transformation (FFT) greatly speeds the calculation. The program X-PLOR (which is available upon request through the author or through Polygen Corporation) uses a vectorized FFT algorithm.[5] As Table 1 demonstrates, the availability of simultaneous FFT routines on the CRAY X-MP/48 system made the algorithm very efficient (Table 1).

SA-refinement requires relatively modest amounts of supercomputer time, typically 20 minutes to 10 CPU hours on a CRAY X-MP system. Most important to the crystallographer is turnaround time. What would take weeks on a VAX computer can be done in a few hours on a Cray system.

Examples

SA-refinement has a radius of convergence larger than that of conventional restrained least-squares refinement. SA-refinement reduces the need for manual model building. Figures 1 and 2 show examples of errors in the initial structure that SA-refinement can



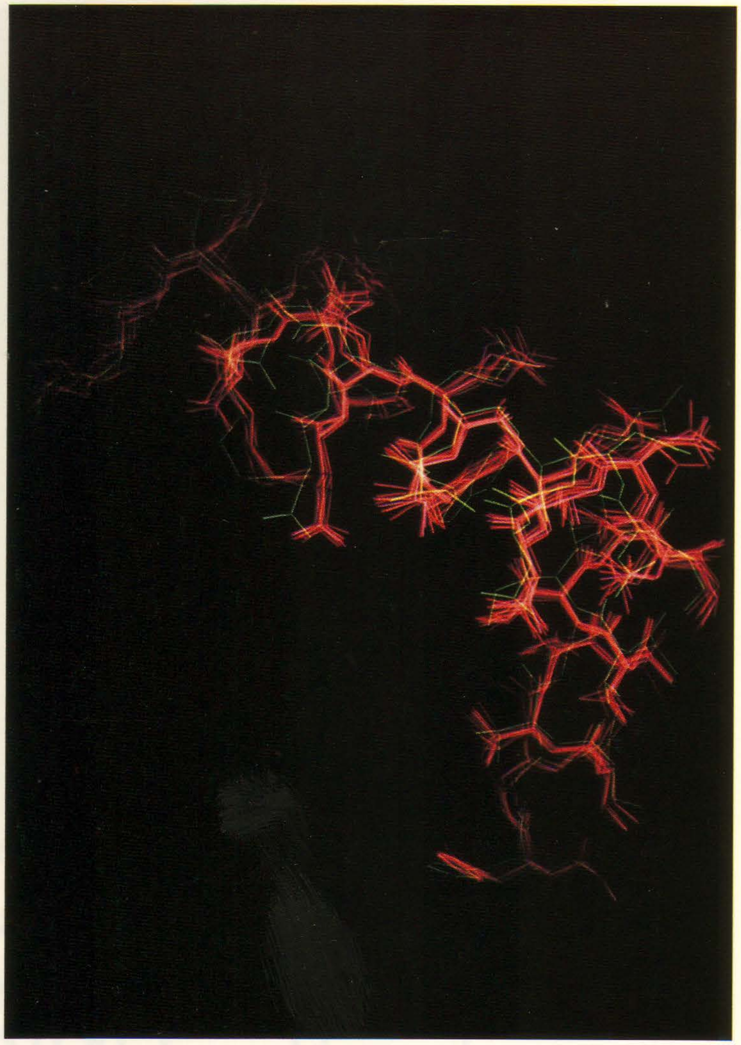


Figure 3 (left). Segment of AATase, consisting of residues 223 to 229 (Phe-223, Ala-224, Tyr-225, Gln-226, Gly-227, Phe-228, Ala-229). Various SA-refined structures (red lines) are superimposed on the initial structure (green lines).

Figure 4 (right). Segment of AATase, consisting of residues 300 to 330. The segment is composed of two alpha-helices that are connected by a turn. Various SA-refined structures (red lines) are superimposed on the initial structure (green lines).

correct.[4] In the case of Figure 1, the histidine-193 ring of the protein aspartate aminotransferase (AATase) has undergone a 90° rotation around the χ_1 bond during SA-refinement. This rotation was accompanied by rearrangements of the backbone atoms. These structural changes were not possible by using conjugate gradient minimization. In the case of Figure 2, SA-refinement has flipped the peptide bond of Glu-265 of AATase by 180° similar to the result obtained by manual refitting. Again, the structural changes were not accomplished by restrained least-squares refinement without refitting.

Crystallographic refinement by simulated annealing can generate an ensemble of structures, each of which agrees with the diffraction information. Regions of large variations of the ensemble indicate either erroneously fitted or disordered segments of the macromolecule. Figure 3 shows a region of AATase that exhibits large variations indicating a problem in the structure, whereas Figure 4 shows an α-helical region that exhibits few variations, or convergence.

Future directions

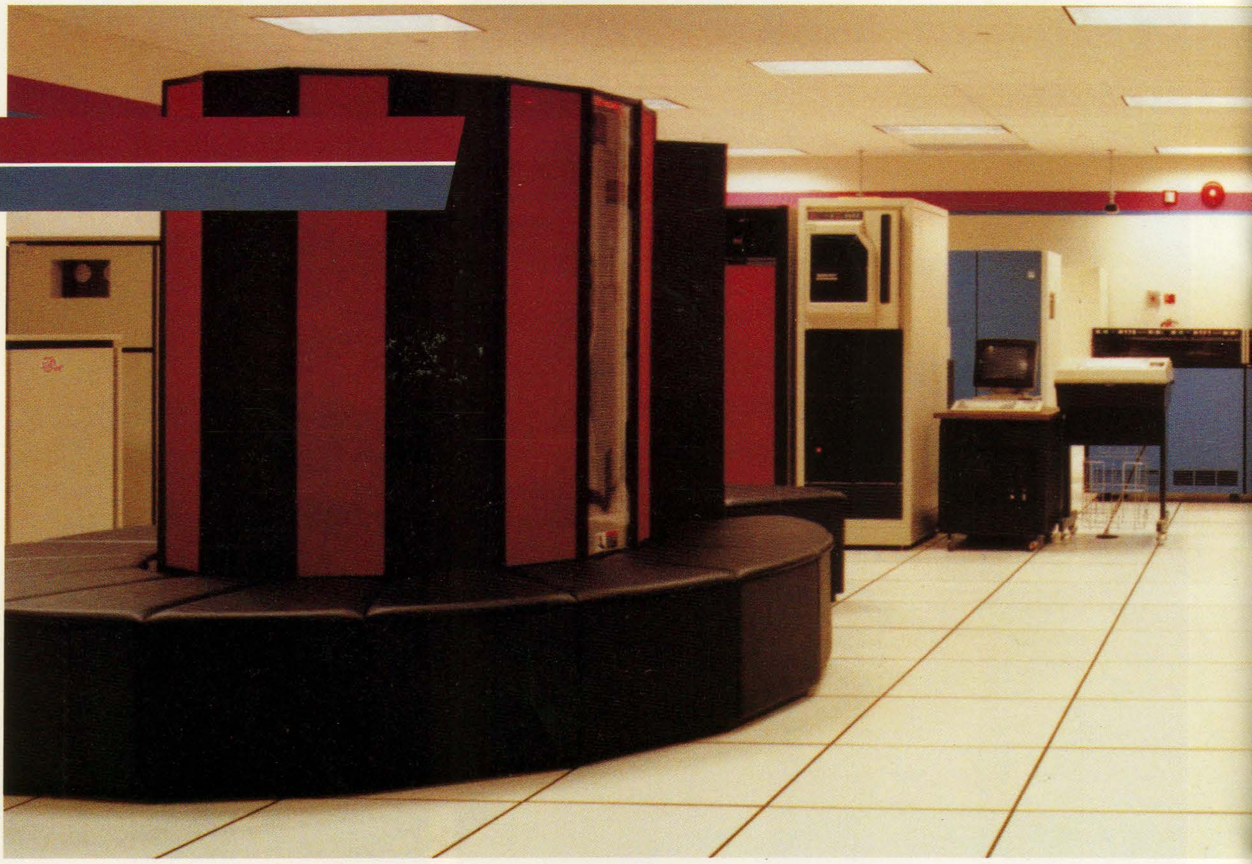
SA-refinement does not provide fully automatic structure refinement at present, but it greatly simplifies the manual refitting of the structure. Current research may determine whether the method is useful in cases when conventional refinement is "locked," that is, when parts of the structure are difficult to interpret. Preliminary results indicated that the methodology of SA-refinement helps to determine the structure in difficult regions of the molecule. ▬

About the author

Axel T. Brünger holds a joint appointment as assistant professor in the Department of Molecular Biophysics and Biochemistry at Yale University and as assistant investigator at the Howard Hughes Medical Institute. He received a Ph.D. degree in biophysics from the Technical University of Munich, West Germany. He spent five years as a postdoctoral research associate in the group of Martin Karplus at Harvard University. His research interests include computational x-ray crystallography and NMR-spectroscopy.

References

1. Jones, T. A., *Journal of Applied Crystallography*, Vol. 11, 1978, pp. 268-272.
2. Kirkpatrick, S., C. D. Gelatt, Jr., and M. P. Vecchi, *Science*, Vol. 220, 1983, pp. 671-680.
3. Brünger, A. T., K. Kuriyan, and M. Karplus, *Science*, Vol. 235, 1987, pp. 458-460.
4. Brünger, A. T., *The Journal of Molecular Biology*, in press.
5. Brünger, A. T., *Acta Crystallographica*, in press.
6. Jack, A. and M. Levitt, *Acta Crystallographica*, Vol. A34, 1978, pp. 931-935.



A blueprint for the UNICOS operating system

Steve Reinhardt, Cray Research, Inc.

Cray Research began developing the UNICOS operating system in 1982 to answer the need for an efficient, high-performance operating system that could be integrated smoothly into diverse computing environments. UNICOS, which is based on the AT&T UNIX System V operating system with some Berkeley extensions, not only provides functions of the Cray operating system COS, but also offers great potential for improving performance, ease-of-use, user productivity, and portability. The Cray software groups developing UNICOS have been pursuing these goals with significant progress.

This article summarizes Cray Research's goals for UNICOS, offers a progress report on where the operating system is today, and looks into the future of UNICOS development. As a foundation for this discussion, we will examine several dimensions of user productivity, since the ultimate goal of UNICOS is to maximize the productivity of Cray system users.

Measuring user productivity

As the computational research community has matured, the definition of computer performance has expanded from MFLOPS ratings alone to include the productivity of the researchers who use the computers. Because Cray users have diverse needs, productivity can be defined in many ways.

Productivity may depend on the ease with which high performance can be achieved. Many users require high performance but do not want to spend time learning the details of hardware and software to squeeze the last drop of performance from a system. Cray Research's UNICOS operating system shoulders the burden of providing Cray power to less sophisticated users and for applications that are not highly tuned for Cray systems.

Other Cray system users find that productivity is limited by the computer's raw performance. For these users, UNICOS must allow a program to harness the peak speed of the underlying hardware. This means driving all of the disks and tapes at peak speeds, using all CPUs efficiently, and running the SSD at peak speeds (on CRAY X-MP, Cray Extended Architecture, and CRAY Y-MP systems). Even when running in a shared environment, devices without contention should run at peak speeds.

Users developing applications may find productivity limited by the speed and effectiveness of the edit-compile-debug cycle. Often these users work best using the Cray system interactively, so UNICOS must provide excellent editors, compilers, debuggers, and other program analysis tools. Additionally, interactive sessions require a network system flexible enough to let users transparently operate remote computers.

Other factors affect users in more subtle ways. Integrating graphics output quickly into a program and displaying data in a useful way lets users get their work done in less time. The ability to develop new features when needed helps many users. To achieve this, software developers must devise a system that can be modified quickly. Allowing hardware designers more flexibility in pursuing high performance also requires an easily modified system.

UNICOS goals and achievements

Before UNICOS can increase the productivity of its users, their diverse requirements must be converted into a real system. This means defining the user requirements as a coherent group of software development goals. Cray Research's goals for UNICOS are derived from many sources, including the philosophies underlying the UNIX and COS operating systems and the demanding requirements of today's high-performance supercomputing environments.

Tracking standards

To stand on the shoulders of other software developers (and not on their fingers), Cray Research must be able to use outside software with few changes. The UNIX system is at the root of this objective, and Cray Research tracks the major standards of the UNIX world, such as AT&T System V, Berkeley BSD, and IEEE/POSIX.

Using standards sometimes means that Cray Research has to choose between competing standards. When two standards cannot be supported simultaneously by Cray Research, we choose the one that will provide the greatest benefit for the largest number of our users. The widespread use of standards allows many of our users and their applications to move easily to Cray systems from other UNIX systems. Standards for graphics packages also are emerging. By using standards we easily can provide the functions that users need, freeing development time for other projects that are more specific to the needs of supercomputer users.

Networking: here to stay

Networking is an area where our emphasis on standards has shown significant benefits. Transmission Control Protocol/Internet Protocol (TCP/IP) is the clear choice today as a network protocol for multivendor sites. It provides a base that supports a wealth of other software, including the X Window System, a standard network window system for bit-mapped workstations, and Sun Microsystem's NFS, a widely used network file system. The Network Queuing System (NQS) developed at NASA/Ames is being considered as a standard as well. Cray Research is following developments from the International Standards Organization. In summary, we believe networking is here to stay.

UNICOS needs to allow users access to different computer systems in a network to perform specialized tasks. Cray systems are superb numeric processing machines, while workstations are better suited for character handling. Disk space for infrequently used data may be less expensive on computers other than a Cray system.

Layered software

Writing software in layers allows greater flexibility to put code in at the optimal level; that is, the level at which one can achieve the optimal compromise between performance, consequences of failure, frequency of change, and the need for customization. For example, many sites have unique requirements for accounting. Writing most accounting code as user-level programs allows users to change the code more easily without hurting its effectiveness or performance. By contrast, disk software needs to be written mostly at the operating system kernel level because system effectiveness depends on disk performance. Such issues sometimes boil down to whether a feature should be implemented in the kernel or as a user process.

Cray Research favors keeping new software functions out of the kernel whenever possible. The operating system kernel is the lowest-level software on the computer. The kernel is responsible for scheduling processes, handling interrupts, initiating I/O, and enforcing protection of resources. Because it controls so much of the software, its speed is critically important. System reliability depends largely on kernel reliability and fault tolerance. We believe high speed and reliability are achieved by keeping the kernel as small as possible. New features have been implemented carefully in the UNICOS kernel, and the kernel has remained fast, small, and reliable. A possible exception to our ideal of keeping complexity out of the kernel is that we have put much of the TCP/IP networking code in the kernel for performance reasons.

Features that are part of the COS operating system often are implemented by UNICOS as privileged user processes. Much of traditional "systems programming" resides in privileged subsystems in UNICOS. The subsystems have proven to be an effective way to accomplish tasks. The rapid pace of development and maturation of NQS and online tape support attests to the soundness of this approach. Keeping code out of the kernel by assigning major tasks to subsystems has enhanced the reliability of the kernel. Even though UNICOS has been released for only two years, it has matured rapidly, and reliability numbers are improving accordingly. For example, UNICOS 3.0 has exceeded its mean-time-to-interrupt goals.

Some features that run as regular user processes with extra privileges include the UNICOS Station Call Processor (which talks to front-end stations with the SCP protocol), the batch subsystem, large parts of the tape subsystem, parts of the TCP/IP networking system, parts of the security system, and a number of miscellaneous processes. The result provides several benefits to users. User-level processes can be developed and debugged, in the words of one developer, "from nine to five — A.M. to P.M." Failures of subsystems are isolated to users using that subsystem. Rather than requiring systems time, test versions of subsystems can be run in parallel with production versions. These factors combine to provide more robust, more easily maintained software.

Reaching peak performance

On performance measures, UNICOS displays the full power of the underlying hardware. On a CRAY-2 system, one program can stream 34 DD-49 disk drives at peak rate. On a CRAY X-MP, CRAY X-MP EA, or CRAY Y-MP system, a program can drive the SSD storage device at its theoretical limit even with moderately sized transfers. On a CRAY-2 system, a multi-

Case study: use of the SSD solid-state storage device

The SSD solid-state storage device is a powerful tool for improving the speed of I/O-bound applications. The device can be used with most CRAY-1, CRAY X-MP, and CRAY X-MP EA systems, and with the CRAY Y-MP system. The evolution of SSD usage demonstrates Cray Research's commitment to providing accessible high performance.

In order to use the SSD more efficiently we must recognize that it typically contains less than 5 percent of a system's secondary storage, while the SSD channels represent more than 95 percent of the total I/O bandwidth of the system. Ideally, almost all I/O transfers (systemwide) would occur across the SSD channel. However, arranging the data to achieve this goal is a challenge.

Initially, the SSD was used under UNICOS as a disk. (The COS operating system supports the SSD only in this way.) This method is simple to integrate into the system, but users must identify the most heavily used files. When the SSD is used as a disk, a method is needed to pick which user files should be SSD-resident. Under COS, this situation led to the idea of the SSD as a controlled device, and then to SSD pre-emption (a method of granting higher-priority jobs SSD access by rolling out lower priority job data to other media). Viewing the SSD as a disk has two major drawbacks: users are required to assign files to the SSD, and it is necessary to arbitrate among users for SSD space. Internally, treating the SSD as a disk means executing much code that is appropriate for disks but not for a fast, low-latency device like the SSD. As a result, this use of the SSD compromises the device's performance.

In 1985, Larry Schermer of Cray Research implemented secondary data segments (SDS), which treat the SSD as extended memory under UNICOS. This type of use provided the peak speed of the SSD by simplifying the mapping scheme and inventing new system calls that were specific to SDS. This resulted in astonishing speed. For example, SDS allows a user to initiate a request to read 512 words (a sector) of SDS and continue with the data transferred in only 25 microseconds. A transfer of 18 sectors transfers at 65 percent of the theoretical channel speed, even including system overhead. SDS meets the needs of users who want to push the limits of the machine. The main drawback to SDS is that it requires the same user intervention as does using the SSD as a disk, and it presents the same arbitration problems for the system when many users want SSD space.

In 1986, Steve Reinhardt of Cray Research implemented a feature to provide transparent high performance, in contrast to the peak performance goals of SDS, by using the SSD as a giant disk cache. Known as the SSD cache, this caching mechanism mimicked the disk cache common to most UNIX systems with a second level of cache blocks that resided in the SSD. The SSD cache was released with UNICOS 3.0 and proved that a caching scheme can provide high payoffs for some job mixes. One advantage was that caching was transparent to the user; normal programs automatically used the cache, and data used most frequently remained in the SSD, while less-frequently used data migrated to disk.

The SSD cache had several difficulties, however. All caching was done in chunks of a single sector, which meant that disk transfers were small, and hence did not make good use of the disk bandwidth. (A good scheme would take advantage of SSD memory to make large, high-bandwidth transfers to and from disks, then use low-latency SSD requests to move data to and from main memory.) The tasking program has achieve 1830 MFLOPS on four CPUs. On a CRAY Y-MP system, the same program will achieve 2497 MFLOPS with a 6.0-nanosecond clock on eight CPUs. Both of these numbers are within one-half percent of the theoretical peak speeds of the machines. On a CRAY X-MP system, a minor system call takes about 17 microseconds under UNICOS, versus about 55 microseconds under COS for a comparable system call.

Disk striping under UNICOS is more flexible than under COS and allows better user control. Memory-resident file systems ensure predictable, fast access for frequently used data, as benchmarks under UNICOS have shown. Under UNICOS, there are a variety of ways to configure disks, the SSD (on CRAY-1, CRAY X-MP, CRAY X-MP EA, and CRAY Y-MP systems), and file systems.

UNICOS also has made other aspects of performance more accessible to users. The variety of I/O caching schemes that exists has provided transparent fast I/O for most programs. Cray Research is developing an autotasking feature to be released with CFT77 3.0 that will move parallel processing into the mainstream of Cray software. Autotasking not only will detect and exploit parallelism automatically, but also will allow knowledgeable users to direct the dependence analyzer to exploit more parallelism in a program.

User environment

Tools should have graphics interfaces, and operator and administrator information should be displayed graphically. While ASCII interfaces are important for many sites and users, the advantages of graphics output are sufficient that we can help users take a quantum leap in productivity by making standard windowing software available everywhere possible. Interactive use of Cray systems will increase steadily because it increases user productivity. High-quality editors and debuggers are critical to many Cray users, especially during program development.

Cray Research provides software for a diverse range of users and sites, and we need to respond to the various needs of different sites. For example, most sites probably will not use all available network protocols. Some may want interactive access to maximize responsiveness to program-generated data, while others may find the security feature appropriate for their computing environments. Cray Research's goal is to provide easily configurable features.

Program environment

Many new UNICOS facilities have been developed to better support user environments with production (batch) workloads. The principal need of these users is an ability to control the rate at which new work enters the system, and the way that machine resources are distributed among jobs that are executing. The main objective is to provide administration with a variety of tools that can be used to control the flow of work through the Cray environment. In this way, users can define policies that work best for them. The most prominent tools in UNICOS for controlling workflow are the Network Queuing System for classification and job entry, the User Data Base for aligning privileges with individual users, the Share Scheduler for the equitable allocation of CPU resources, and a

single-sector approach also used excessive main memory for table space. The SSD cache proved the feasibility of a cache scheme, but several problems had to be overcome to make it a general-purpose product.

By working to overcome the problems with the SSD cache, Larry Schermer developed the scheme now known as the logical device (LD) cache. Although similar to the SSD cache in many respects, the LD cache has several improvements over the SSD cache. First, because different file systems may have different I/O needs, the SSD space is allocated (by a system administrator) to each file system in an amount appropriate to that file system. For instance, file systems that are used for temporary scratch space often will have much I/O done on them, so it makes sense to use considerably more SSD space as cache to speed up transfers to them. Second, these file-system-dependent slices can be split into different chunk sizes, allowing often-used file systems to have chunks of several tracks or more. This technique uses disk bandwidth more effectively. Third, because the attachment of SSD space to buffers is dynamic, SSD space can be detached from one file system and added to another as the system is running, allowing sites to tune the SSD allocation without rebooting the operating system.

The performance results for the LD cache were much better than for the SSD cache. In the simple case in which data resides on the SSD, a transfer of 50 sectors runs at 600 Mbytes/sec, which is about 25 percent of the channel speed. Numbers from a recent benchmark are more useful for seeing how well LD cache performs in a mix of programs. Seven jobs with a total of 91 million words of data were run on a system with a 32-million-word SSD. By running all I/O to disk we observed 1815 seconds elapsed time. By assigning files to the SSD and making each job wait until enough space was free for its files (that is, using it as a "COS-controlled device") we observed a run time of 640 seconds. By running the mix with all data moving through the LD cache we observed a run time of 467 seconds, which is a factor of four speedup over disk I/O and a 27 percent speedup over traditional SSD I/O.

With UNICOS 4.0 the SSD is supported in three ways: as a file system (disk), as SDS, and as LD cache. (The SSD cache was replaced by the LD cache.) Given these options, the majority of sites should consider using the SSD as LD cache. Swapping, reading system binaries, and other miscellaneous functions will exercise the cache automatically and run faster. Problems arising from contention do not exist; the most frequently used data reside in the SSD. Many sites will have a few users and applications that need the full I/O bandwidth of the SSD to run in a timely manner and will use SDS. LD cache and SDS coordinate such that a request for SDS space can be accommodated by shrinking the amount of space used for LD cache. Thus, no space needs to be dedicated to SDS, but will be available when needed. If SDS requests oversubscribe the SSD, processes can be checkpointed to free their SDS space. With the new, larger SSDs (a 512-million-word SSD has about as much space as four DD-49 disk drives) it is unclear how many sites will wish to use that much disk space to hold checkpoint files.

SSD handling in the UNICOS operating system typifies many of the goals we have for the UNICOS system. For example, available hardware should be used to make programs run fast, without requiring user program changes. Also, different uses of the same hardware may be better for various groups of users. Finally, being able to implement a solution quickly allows developers to try different possibilities to find the best solutions.

tunable memory scheduling algorithm. Apart from workflow concerns, other facilities have been developed that implement industry-standard support for ANSI- and IBM-labeled tape as well as a full complement of data and format conversion utilities.

Program development

So far, the biggest UNICOS success story has been program development. The tools to develop and optimize programs are flexible and varied. Under UNICOS, working with a multi-CPU machine is much simpler than under COS. Interactive use lets many users work more productively without hampering other large programs in the system. Full-screen text editors run across a network to bit-mapped displays on workstations. From a front-end system or workstation a user can log onto a Cray system and interactively edit and compile a program. Visualization is available both through the X Window System for moderate data rates and through Cray Research's HSX high-speed external communications channel for extremely high data rates. A shell script that makes up a job can be run interactively or through the job queuing system. While the tool set is not yet complete, the structure is in place for new programs that will satisfy user needs.

The biggest UNICOS success story has been program development.

Feature development

Software adaptability under UNICOS has met Cray Research's expectations. For example, we were able to support 16 million words of memory on the CRAY X-MP system much more easily with UNICOS than with COS. With COS, which is written entirely in CAL, it was a painstaking task to switch from using A- registers to S- registers for address calculations. However, UNICOS is written almost entirely in C. Since the C compiler already had been changed to generate CAL using the new addressing scheme, converting UNICOS was a simple task.

Most of the UNICOS commands are shared between CRAY-2 and CRAY X-MP systems. Cray Research has been able to implement a given feature of the operating system from two to ten times more quickly with UNICOS than COS. Some new features have taken only a few days to develop. Today UNICOS development is limited as much by time spent identifying user requirements and working with users to set priorities for developing the software, as by how long it takes to implement code, once decided upon.

Interactive access

The question of interactive access to Cray systems finds few people without strong opinions. Cray Research ports as much standard software as possible and preserves standard interfaces. This allows customers to port programs from other UNIX systems to UNICOS without rewriting them. Unfortunately, this approach provides freedom to use the Cray system poorly. For example, many standard UNIX programs do single-character I/O to a screen. This can be very expensive if the character is making several hops across a network to the Cray system and back to the user's terminal. Software developers hold a wide range of attitudes about this issue. Some favor showing a stronger prejudice against single-character I/O to Cray systems (to help users avoid stepping in a trap), while others believe we should focus on allowing easy porting of codes. As in other areas, we prefer to provide alternatives (with clear descriptions of the pros and cons) rather than dictate solutions; all the interactive features can be disabled at a site's discretion.

The future of UNICOS

Looking ahead, most users will work more productively in an integrated programming environment using windowing. Editors, debuggers, compilers, and optimizers will use a common database to hold program information. Using parallel processing will be easier with this scheme; a parallel dependence analyzer will be merely another program that restructures the database. Because our user base is so varied, Cray Research will provide sites and users with more options for tailoring software for their needs. To do this we first must do three interrelated things: continue to work with users to determine what they want, choose those that are appropriate to send to customers after developing and testing choices internally, and explain what needs the software addresses, including how it solves a particular problem and how it compares with other solutions to the same problem.

We believe that graphics and windowing features will be used more widely in tools and products. Debuggers, performance analyzers, and system configuration and monitoring tools will have graphical output. For example, a display may have a top view of the computer room, with a Cray mainframe, disks, tapes, and network devices all drawn in position. Each peripheral device would display the amount of bandwidth being used, the number of requests received, and would signal if too many errors occurred in a time interval. Thus operators, customer engineers, and systems administrators could identify a failing device or a performance bottleneck more quickly.

In many ways, the success of Cray Research software depends on how quickly software allows us to adapt to the changing requirements of the underlying hardware and the changing needs of our users. On this scale, UNICOS is fulfilling its expectations. Because our goal is to provide the most productive computing environments for our users, we believe that UNICOS will prove to be a continuing success in giving Cray system users what they need.

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

Acknowledgments

The author thanks Jim Harrell, Tim Hoel, Vic Lee, Jim Miller, and Larry Schermer of Cray Research for their ideas, reviews, and comments.

About the author

Steve Reinhardt is a senior programmer/analyst in Cray Research's Software Division. He worked on the Cray operating system COS for four years and on the UNICOS kernel for three years. He currently works in the multiprocessing software group. Reinhardt received a B.S. degree in computer science from Yale University in 1980.

CORPORATE REGISTER

NASA Ames receives first CRAY Y-MP system delivery

In August Cray Research delivered the first CRAY Y-MP computer system to the **National Aeronautics and Space Administration's Ames Research Center** in Mountain View, California. The system replaces one of two CRAY-2 supercomputers at the NASA Ames site. It is the second High-Speed Processor (HSP-2) in the Numerical Aerodynamic Simulation (NAS) program. NASA will use the system to simulate advanced aerodynamic designs. John Barton, HSP-2 manager at NASA Ames, said, "Installation of the CRAY Y-MP system maintains the NAS Systems Division at the leading edge of supercomputer technology."

The **National Center for Supercomputing Applications** at the University of Illinois, Urbana-Champaign, has ordered a CRAY-2S/4-128 computer system, Cray Research announced in June. The system will be installed in the fourth quarter of 1988. The University of Illinois will be the first center funded by the National Science Foundation to own and operate two Cray computer systems. Cray Research Chairman John Rollwagen said, "With the addition of the CRAY-2 system, computing power at NCSA will more than double. We congratulate the University of Illinois for its success in bringing supercomputing to a broad range of researchers in academia and industry."

The Ohio State University, acting as fiscal agent for the **Ohio Supercomputer Center** in Columbus, has ordered a CRAY Y-MP computer system to be installed in the fourth quarter of 1989. Ohio State is the first university to order a CRAY Y-MP system. The system will replace a CRAY X-MP/24 system that was installed in 1987. Researchers at more than 20 colleges and universities are using the CRAY X-MP system in such disciplines as aerodynamics, chemistry, astronomy, and physics. Charles F. Bender, director of the Ohio Supercomputer Center, said, "The CRAY Y-MP system will be a research tool; it also will help us recruit researchers to our colleges and universities and allow us to compete more effectively for federal grants." He added, "By the time we get the CRAY Y-MP system next year, our CRAY X-MP system will be saturated." Cray Research announced the order in June.

Also in June Cray Research announced that the Martin Marietta Corporation Information and Communications System Division, acting as prime contractor for the **U.S. Air Force**, had ordered a CRAY-2S/2-128 computer system valued at approximately $17 million. The system was installed in the third quarter of 1988 at the National Test Bed facility at Falcon Air Force Station near Colorado Springs, Colorado. The U.S. Air Force will use the system to simulate a layered ballistic missile defense system for the National Test Bed program.

In July Cray Research announced that **Shell Research, B.V.** had ordered a CRAY X-MP EA/164 system to be installed at the Koninklijke/Shell Exploratie en Produktie Laboratorium in Rijswijk, the Netherlands. The computer system will be installed in the fourth quarter of 1988, and will be the second Cray system to be installed at the Shell KSEPL site. Shell will use the system for seismic data processing and reservoir simulation.

Deutsches Klimarechenzentrum GmbH (DKRZ) located in Hamburg, West Germany, has ordered a CRAY-2 computer system to be installed in the fourth quarter of 1988. DKRZ, formerly known as the Max Planck Institute for Meteorology, will use the CRAY-2 computer system for climate research and long-range environmental studies. Wolfgang Sell, general manager of DKRZ, said that the CRAY-2 system was selected for its vector performance and large, directly addressable memory. "This will open new research opportunities for advanced weather and climate modeling," he said. DKRZ will share the system with several other oceanographic and meteorological institutes in West Germany. Cray Research announced the order in July.

Also in July Cray Research announced that Digital Equipment Corporation, acting as a major subcontractor for the **U.S. Army**, Kwajalein Atoll (USAKA), had ordered a CRAY X-MP/14se computer system. This will be the first Cray system dedicated to a real-time application. The system is scheduled for installation in the fourth quarter of 1988 at the USAKA site in the U.S. Marshall Islands. The system will be used for radar image analysis.

Two universities in Brussels, Belgium, **Universite Libre de Bruxelles** and **Vrije Universiteit Brussel** (ULB/VUB), have ordered a CRAY X-MP/14se computer system. The Cray system, which is the first to be installed in Belgium, is scheduled for delivery in the first quarter of 1989. ULB is a French-speaking university with an enrollment of about 14,000 students, while VUB is a Dutch-speaking university with an enrollment of about 7,000 students. Cray Research announced the order in July.

In August Cray Research announced that the **Pittsburgh Supercomputing Center**, Pittsburgh, Pennsylvania, had signed a long-term $30 million agreement to purchase a CRAY Y-MP computer system for delivery by the end of 1988, and to upgrade to a CRAY-3 supercomputer, which is under development. The eight-processor CRAY Y-MP/832 system will be installed at the center in the fourth quarter of 1988, and will replace the CRAY X-MP/48 system installed since 1986. The CRAY Y-MP will be the first installed in a National Science Foundation Center, and will be made available to scientific and engineering researchers in the first quarter of 1989. Richard Cyert, president of Carnegie-Mellon University, said, "The Pittsburgh Supercomputing Center has been a tremendous success story. The commitment we have made will ensure that the PSC stays at the leading edge of supercomputing technology for the foreseeable future."

In August Cray Research announced that **Lawrence Livermore National Laboratory** had ordered a CRAY-2S/4-128 computer system. The computer will be installed at the National Magnetic Fusion Energy Computer Center in Livermore, California. Installation is scheduled for the third quarter of 1988. The laboratory is operated for the U.S. Department of Energy by the University of California. This will be the fifth Cray computer system to be installed at LLNL's Energy Computer Center; two of those, both CRAY-1 systems, will be retired.

Announcing UNICOS 4.0

Release 4.0 of the Cray operating system UNICOS, which is based on AT&T's UNIX system V operating system with extensions from 4.2BSD, is now available. The UNICOS 4.0 release runs on all Cray systems, including CRAY X-MP EA and CRAY Y-MP systems. This release adds many enhancements and new features, including expanded hardware support. The FEI-3 network interface and the CNT LANlord network connection are supported. The DS-40 Disk Subsystem is supported for CRAY X-MP EA and CRAY Y-MP systems. Support for the HSX high-speed external communications channel has been enhanced. Also, the Cray hardware performance monitor is supported for CRAY X-MP systems. The operator workstation, which replaces the IOS peripheral expander and system consoles, also is supported.

Enhancements

UNICOS 4.0 includes the following enhancements:

☐ A new security feature has been added to support concurrent processing of sensitive information at multiple security levels (This feature will be available with UNICOS 5.0 for CRAY-2 systems).

☐ The UNICOS tape subsystem supports production tape operations.

☐ Sun Microsystems' Network File System (NFS) is supported for CRAY-2 systems (This feature will be available with UNICOS 5.0 for other Cray systems).

☐ The Network Queuing System (NQS) improves efficiency, supports user/group limits, and now can be networked between UNICOS hosts.

☐ The accounting feature has been enhanced for the accounting of multitasking programs.

☐ New commands, **cf** and **cf77**, provide an enhanced interface to Cray Fortran compilers.

☐ Job and process recovery capabilities have been extended.

☐ The UNIX System V.3 file-locking feature is implemented.

☐ Microtasking has been improved and now supports CRAY-2 systems.

☐ The assign command is now available on all Cray systems.

☐ On CRAY-2 systems, users now may assign a file as temporary and be assured that the file space is released when the job or process leaves the system (This feature will be available with UNICOS 5.0 for other Cray systems).

☐ File system backup utilities dump and restore now are available for all systems.

☐ A portion of the SSD solid-state storage device (for CRAY Y-MP systems, CRAY X-MP EA systems, and appropriately equipped CRAY X-MP systems) can be used as a logical device cache.

☐ The File System Switch (FSS), which provides a framework for assessing various types of file systems, has been implemented (FSS was provided with UNICOS 3.0 for CRAY-2 systems).

☐ C 2.1, which produces optimized, vectorized code, is now the default C compiler for CRAY-2 systems (C 2.1 integers consist of 64 bits with 46 bits of precision, while the previous C compiler had 32 bits of precision).

☐ Many system calls, commands, and library routines are new or upgraded.

Languages

UNICOS 4.0 supports CFT77 2.0, Pascal 3.3, and CAL version 2, release 3.2 on all Cray computer systems. CFT2 4.0 and C 2.1 are supported on CRAY-2 computer systems. CFT 1.15 BF2 and C 3.1 are supported on CRAY X-MP and CRAY-1 computer systems. All of these languages, except CFT77 2.0, are included with UNICOS 4.0. The following compilers will provide support for Cray Extended Architecture systems: CFT77 3.0, CFT 1.15 BF3 (24-bit mode only), C 4.0, and Pascal 4.0.

Libraries and products

UNICOS 4.0 supports many libraries, including Fortran, C, math, scientific, microtasking, macrotasking, and the X Window System. UNICOS 4.0 supports the tools and utilities in the common set developed from the AT&T System V.3 and 4.2BSD releases of the UNIX operating system. In addition, UNICOS 4.0 supports Cray products for source code maintenance, symbolic debugging, parallel processing, performance analysis, I/O Subsystem (IOS) maintenance, and other uses.

On CRAY X-MP systems, CSIM now can simulate the Y-mode instruction set to support Cray Extended Architecture systems. Other enhanced products for UNICOS 4.0 include PREMULT 1.2, which now is supported on all Cray computer systems, and the **csort** command, which has been ported to UNICOS from the Cray operating system COS. A feature called **hpm** has been implemented to provide the equivalent of the COS **perfmon** command for reporting the results of the hardware performance monitor feature of the CRAY X-MP architecture. A new facility to link and load the object modules by using traditional overlay techniques now is available with the **ldovl** command. The **yadb** debugger, which can analyze multitasking programs and use the X Window System, now is available on CRAY X-MP and CRAY-2 systems. On-line diagnostics have been enhanced for all Cray systems.

Communications

UNICOS 4.0 running on CRAY-2 systems now supports Sun Microsystems' NFS client and server mode, which allows users to access and manipulate files transparently between remote file systems and the CRAY-2 file system.

Enhancements have been made to the transmission control protocol/internet protocol (TCP/IP), including performance and maintenance enhancements and HSX support. Also, the UNICOS Station Call Processor (USCP) has been enhanced. UNICOS 4.0 supports version 10 and version 11, release 1, of the MIT X Window System.

Migration

Cray Research is working to simplify the process of converting from COS to UNICOS. The improved COS Guest Operating System (GOS) will allow CRAY Y-MP, CRAY X-MP EA, and CRAY X-MP system multiprocessor users to run COS 1.17 and UNICOS 4.0 concurrently (if memory and disk resources permit), providing a valuable aid to migration. A special version of the CFT 1.14 compiler allows the migration of applications to UNICOS while maintaining the same language processor as COS. A set of migration tools and a special migration team also help customers make a smooth transition to UNICOS.

For more information about using the UNICOS operating system on Cray computer systems, contact the nearest Cray Research sales office.

Common Lisp now available for Cray systems

Cray Research is pleased to offer Allegro CL® (Common Lisp) for Cray computer systems. Cray Allegro Common Lisp enhances industry-standard Common Lisp with extensions that take advantage of the power of Cray computer systems, enabling Cray customers to apply supercomputer power to large real-world problems in artificial intelligence. Version 1.0 of Cray Allegro Common Lisp runs under the UNICOS operating system (release 4.0 and later) on CRAY X-MP systems, CRAY X-MP EA systems, and CRAY Y-MP systems running in X-MP mode. Future releases of Allegro Common Lisp to run on other Cray systems are under development.

Lisp, which stands for list processing, is quite different from imperative and procedural computer languages. It enables programmers to build computational models that can adapt and learn. In Lisp, program behavior can be modified during execution as new information is obtained, providing an effective problem-solving tool. Programmers using Lisp can use functions that are built into the language and functions defined by the programmers. Newly defined functions can be used interpretatively or compiled for efficiency. Lisp processes lists and symbols as well as numbers, thus providing power and flexibility. Lisp's strength lies in its ability to manipulate ordered collections of objects without requiring programmers to reduce them to numeric quantities. Recursion, a natural programming technique, allows a concise and elegant statement of problems and algorithms.

Cray Research's Allegro Common Lisp incorporates features and extensions, including

☐ A standard implementation of Common Lisp as described in *Common Lisp: the Language*, by Guy Steele, Jr. (Digital Press, 1984)
☐ Compliance with the ANSI X3J13 technical committee standards for Common Lisp
☐ The ability to use the powerful EMACS editor for editing, formatting, and syntax checking of code
☐ An extensible set of top-level commands
☐ Error handling
☐ Debugging, stepping, and tracing
☐ Object-oriented programming support (Flavors and Portable Common Loops)
☐ Support for the industry-standard X Window System

The software environment also includes Cray Research's loader SEGLDR, and profiling tools that are useful in analyzing program performance during optimization. Cray Allegro Common Lisp provides an interpreter and a compiler; interpreted and compiled functions can be intermixed freely. This capability allows a programmer to switch from the compiler to the interpreter when developing a challenging section of code, thus attaining interactive convenience and compact, fast, compiled code.

Many applications are particularly well-suited to Cray Allegro Common Lisp, such as artificial intelligence; natural language processing, including speech recognition; image processing; automatic theorem proving; rule-based expert systems that emulate human experts in fields such as aerospace, geology, and automotive design; software prototyping and automatic programming; diagnostics and "what if" scenarios; and coupled symbolic-numeric problems. Also, because it is an extensible language, Lisp can be used to create special-purpose languages.

For more information on Cray Allegro Common Lisp, contact the nearest Cray Research sales office.

Cray Ada compiler handles diverse needs

An Ada language compiler now is available for Cray computer systems. Cray customers now have the option of combining the strengths of Ada with the large memories and extremely fast processing speeds of Cray computer systems. Ada release 1.0 runs on CRAY-2 and CRAY X-MP computer systems under the UNICOS operating system (release 4.0 or later). This release provides an Ada environment to meet the requirements of Cray customers and encompasses the compiler and tools that are necessary to support the Ada programming language.

Version 1.0 of Cray Ada conforms to the Ada language as defined in the *American National Standard Reference Manual for the Ada Programming Language*, ANSI/MIL-STD-1815A-1983. It is a validated Ada compiler, having passed the Ada Compiler Validation Capability 1.9 test suite of the Department of Defense. In addition to standard features, release 1.0 of Cray Ada allows users to call routines written in Fortran, C, Pascal, and the Cray assembly language (CAL).

Release 1.0 of Cray Ada includes

☐ Compiler
☐ Library management tools
☐ Linker
☐ Source dependency lister and formatter
☐ Cross-referencer
☐ Source-level debugger

Cray Research is developing Ada in two phases. The first phase (release 1.0) emphasizes early availability, the second phase (release 2.0) emphasizes improved performance.

Release 1.0 of Cray Ada provides a preliminary Ada environment. Release 2.0, which also will run on CRAY Y-MP systems, will support scalar optimization, vectorization, and instruction scheduling while retaining a user interface compatible with release 1.0.

For more information about using Ada with Cray computer systems, contact the nearest Cray Research sales office.

Symposium papers invited

Authors are invited to submit papers for "Supercomputers: Emerging Applications in Manufacturing," a symposium cosponsored by Cray Research that will take place September 11-13, 1989, in Minneapolis, Minnesota. The symposium will concentrate on significant new applications in manufacturing in areas such as process design, prototyping and evaluation through manufacturing processing, production control, and quality assurance.

Cray Research is cosponsoring the symposium with ETA Systems, Inc., the Minnesota Supercomputer Center, Inc., the Minnesota State Department of Trade and Economic Development, the National Bureau of Standards, and the University of Minnesota Productivity Center. Topics of interest include applications in mechanical, electrical, chemical, and manufacturing engineering. Papers with specific examples that highlight research and development efforts are encouraged.

One- or two-page abstracts are due by November 1, 1988, and full papers or extended abstracts are due by February 1, 1988. They may be sent to the program review committee, Minnesota Supercomputer Center, Inc., 1200 Washington Avenue South, Minneapolis, MN, 55415; telephone: (612) 626-1888. For more information contact Carlos Marino, Cray Research, Inc., 1333 Northland Drive, Mendota Heights, MN, 55120; telephone: (612) 681-3652.

APPLICATIONS UPDATE

ACSL models dynamic systems

ACSL is a continuous system simulation language from Mitchell and Gauthier Associates that runs on all Cray systems under the COS, CTSS, and UNICOS operating systems. The code is designed to help engineers and scientists mathematically model and analyze the behavior of continuous systems described by time-dependent, nonlinear differential equations or transfer functions. Typical application areas include control system design, aerospace simulation, fluid-flow analysis, and heat-transfer analysis.

ACSL is based on Fortran, and its structure follows the specifications of a standard continuous system simulation language established in 1967. An ACSL simulation consists of two sections: the model definition and the run-time commands. The model definition involves using ACSL statements to establish mathematical specifications for the continuous simulation system being modeled. The model then is analyzed by instructions interpreted during the run-time command section. The advantage of this two-part structure is that once a model is defined, it can be kept on file and analyzed repeatedly by the run-time commands. All variables in the model definition can be displayed, plotted, or printed, and any constant or parameter can be changed before a simulation is run.

The language helps eliminate extraneous statements and provides extensive data output formats. ACSL complements the control synthesis tools used with linearized systems such as root locus and frequency response by evaluating performance in a nonlinear real-world situation. ACSL features include

☐ Interactive command structure for data output
☐ Full compatibility with user-generated Fortran libraries
☐ Choice of integration routines including user's own routine
☐ Gear's integration algorithm for stiff systems
☐ Complete access to variable dictionary via ACSL commands
☐ No limit to program size, number of names, states, or labels
☐ Low CPU and memory operation
☐ A DISCRETE section for modeling online digital controllers
☐ Array capabilities complemented by vector and matrix integration operators
☐ Simplified structure with default options for the novice

For more information about using ACSL with Cray computer systems, contact John Rodrigues, Mitchell and Gauthier Associates, 73 Junction Square Drive, Concord, MA, 01742-3096; telephone: (508) 369-5115, or contact Doug Petesch, Cray Research, Inc., 1333 Northland Drive, Mendota Heights, MN, 55120; telephone: (612) 681-3654.

VIP-EXECUTIVE applies new solution techniques

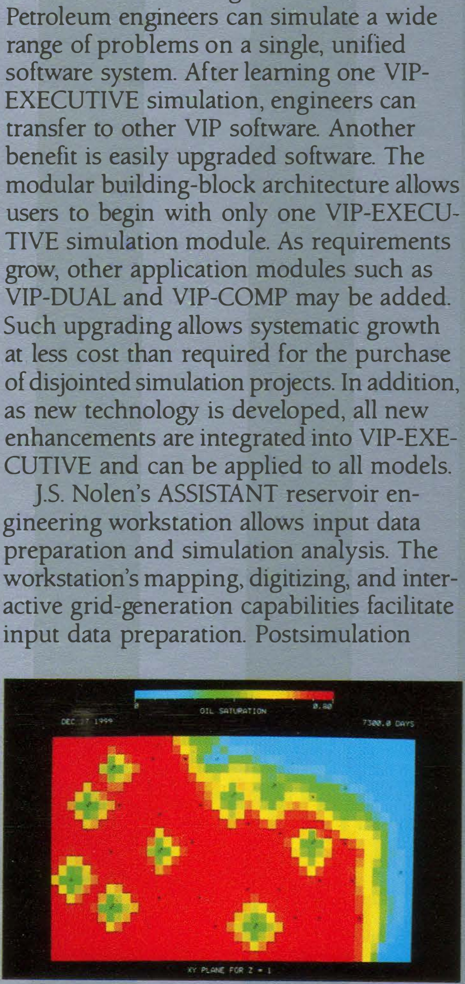
J.S. Nolen & Associates offers VIP-EXECUTIVE™ petroleum reservoir simulation software, an integrated suite of simulator modules accessible from a single interactive graphics pre- and postprocessor, ASSISTANT. VIP software was designed specifically for vector computer systems and runs on all Cray systems under the COS and UNICOS operating systems.

Modular design is a key feature of the VIP-EXECUTIVE software. VIP-CORE, the initialization module of VIP-EXECUTIVE, provides user control and access to various application modules including

☐ VIP-COMP, the N-component, equation-of-state simulator for compositional studies
☐ VIP-ENCORE, three-dimensional, three-phase "black oil" simulator with ability to handle multicomponent-dimensional systems in which PVT properties are described adequately by pressure-dependent K-values
☐ VIP-DUAL, the fractured-matrix model for treating reservoirs with dual-porosity, dual permeability characteristics (VIP-DUAL may be coupled to either VIP-COMP or VIP-ENCORE modules)
☐ VIP-POLYMER, the polymer option available for VIP-ENCORE and VIP-COMP
☐ VIP-THERM, the thermal simulator for hot-water or steam-flood modeling
☐ EOS-PAK, the numerical model for phase-behavior simulation

The linked software and sharing of computer code provides multiple benefits for users of VIP-EXECUTIVE software. One benefit is that training needs are reduced. Petroleum engineers can simulate a wide range of problems on a single, unified software system. After learning one VIP-EXECUTIVE simulation, engineers can transfer to other VIP software. Another benefit is easily upgraded software. The modular building-block architecture allows users to begin with only one VIP-EXECUTIVE simulation module. As requirements grow, other application modules such as VIP-DUAL and VIP-COMP may be added. Such upgrading allows systematic growth at less cost than required for the purchase of disjointed simulation projects. In addition, as new technology is developed, all new enhancements are integrated into VIP-EXECUTIVE and can be applied to all models.

J.S. Nolen's ASSISTANT reservoir engineering workstation allows input data preparation and simulation analysis. The workstation's mapping, digitizing, and interactive grid-generation capabilities facilitate input data preparation. Postsimulation



Simulation of oil saturation, as modeled by VIP software.

analysis is enhanced by high-resolution color graphics displays that include *x*, *y* plots, contour maps, and color-shaded maps.

For more information about using VIP-EXECUTIVE software on Cray computer systems, contact Debbie Fogarty, J.S. Nolen & Associates, Inc., 16225 Park 10 Place, Suite 560, Houston, TX, 77084; telephone: (713) 578-3210, or contact Bill Kamp, Cray Research, Inc., 1333 Northland Drive, Mendota Heights, MN, 55120; telephone: (612) 681-3665.

IMCOMP: an image compression and conversion algorithm

IMCOMP is a set of image conversion and compression routines that provides a useful scheme for data communications, data storage, and downloading images to frame buffers with various memory configurations. The advantages of image compression include reduced memory requirements, increased transmission speeds, high-quality in the resulting images, and cost savings over networks and phone lines. The package can convert images consisting of 24, 16 or 8 bits/pixel to 16 or 8 bits/pixel, then compress them further to 2 or 3 bits/pixel while maintaining a reasonable full-color representation. IMCOMP runs under the CTSS and UNICOS operating systems. The National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign supports IMCOMP for its users on the center's CRAY X-MP/48 system. The package also is available for licensing to non-NCSA users.

The IMCOMP package was developed by Charnchen Dai under the supervision of Tom DeFanti at the Electronic Visualization Laboratory at the University of Illinois at Chicago. IMCOMP consists of two processes: color cluster conversion and color cell compression (CCC). The color cluster conversion algorithm enables users to display computer-generated images on a variety of frame buffers, regardless of the number of bits/pixel. IMCOMP reduces the number of colors in an image from 24, 16, or 8 bits/pixel to 16 or 8 bits/pixel (8-bit systems with color look-up tables). The algorithm for reducing the number of colors in an image intelligently selects a set to represent the original color distribution. Pixels defined by 24 or 16 bits contain RGB information; pixels defined by 8 bits are pointers to a 256-color map look-up table. The color cluster algorithm accounts for the different encoding schemes when converting 24, 16, or 8 bit/pixel images to 16 or 8 bits/pixel.

The CCC compression algorithm converts 24, 16, or 8 bit/pixel images to 3 or 2 bits/pixel. The resulting images can be transmitted over networks or phone lines, then decoded in seconds using personal computers and displayed on 16- or 8-bit frame buffers, such as Truevision Inc.'s TARGA-16 or TARGA-M8 boards, and Number Nine Computer Corporation's Revolution Board.

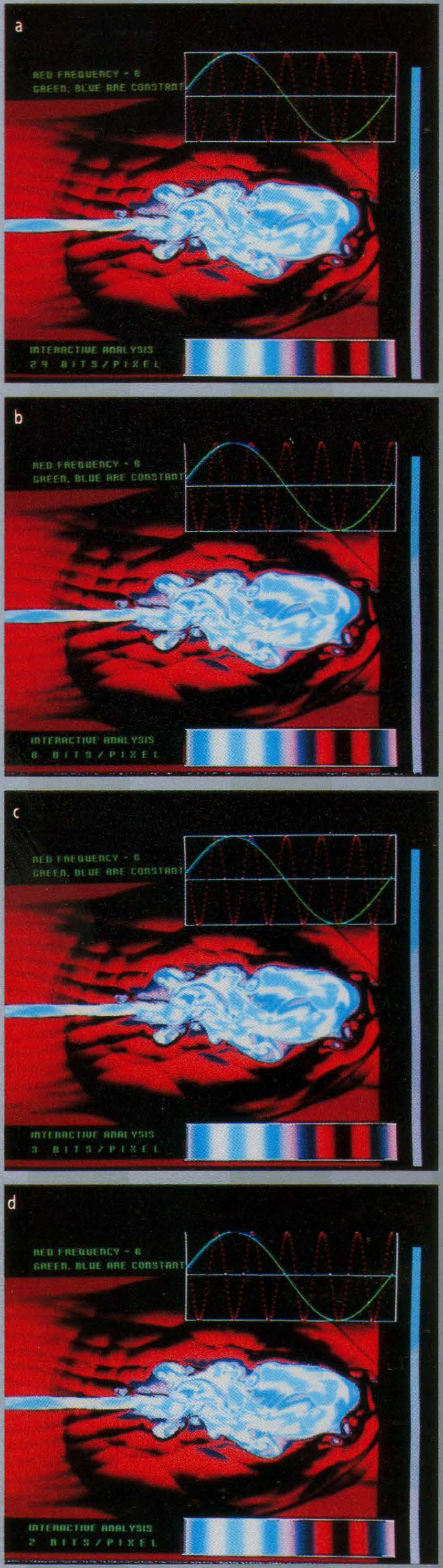
The compression algorithm first divides a color image into 4-by-4-pixel areas. It then generates a 16-bit binary bitmap and two 24-bit representative colors — a high and a low color — for each pixel in a given 4-by-4 pixel area. If a pixel's bit value is ON (1), then the corresponding pixel takes the high color as its color value; if its value is OFF (0), then it takes the low color. This 4-by-4-pixel area representation constitutes 8 bytes of information, or 4 bits/pixel.

For 3 bit/pixel image compression, the color cluster program maps these representative colors (24 bits of information) into 16 bits by throwing away the lowest-order 3 bits for each 8-bit R, G, and B and then packing the remaining 5 bits each into 2 bytes. 3 bit/pixel compression is valid only if the image is displayed on 16-bit systems. This new 4-by-4-pixel area representation constitutes 6 bytes of information, or 3 bits/pixel.

For 2 bit/pixel image compression (valid only if the image is to be displayed on an 8-bit system that has a color look-up table), the color cluster program creates a 256-color map look-up table that contains the "best selection" of high and low representative colors. It replaces the color values in the 4-by-4 pixel area description with high and low representative color pointers. A 768-byte color map also must be transmitted with the image. This new 4-by-4 pixel area representation is 4 bytes of information, or 2 bits/pixel.

Each method offers advantages and disadvantages. The 3 bit/pixel compression scheme allows more colors than the 2 bit/pixel scheme (32K versus 256 colors), and the choice of the 32K colors is computationally trivial, so it operates much faster. 3 bit/pixel images, however, contain 50 percent more data, which requires more transmission time and storage. At the receiving site, 3 bit/pixel images must be displayed on a 16 (or greater) bit/pixel buffer; 2 bit/pixel images can be displayed on an 8 bit/pixel buffer.

The IMCOMP software is supported by NCSA for its users. For more information on using IMCOMP with Cray computer systems, non-NCSA users can contact David Shu at Frederiksen and Shu Laboratories, Inc., 531 West Golf Road, Arlington Heights, IL 60007; telephone: (312) 956-0710, or contact Cal Kirchhof, Cray Research, Inc., 1333 Northland Drive, Mendota Heights, MN, 55120; telephone: (612) 681-3662.



"Astrophysical Jet Streams" by Michael Norman and Donna Cox, National Center for Supercomputing Applications. The same image has been processed using the IMCOMP software: the 24 bit/pixel, full-color image (768 Kbytes) (a), the 8-bit color clustered image (256 Kbytes) (b), the compressed 3 bit/pixel image (96 Kbytes) (c), and the compressed 2 bit/pixel image (64 Kbytes) (d). Note that the text remains perfectly readable in all cases.

USER NEWS

Thin strings yield big things

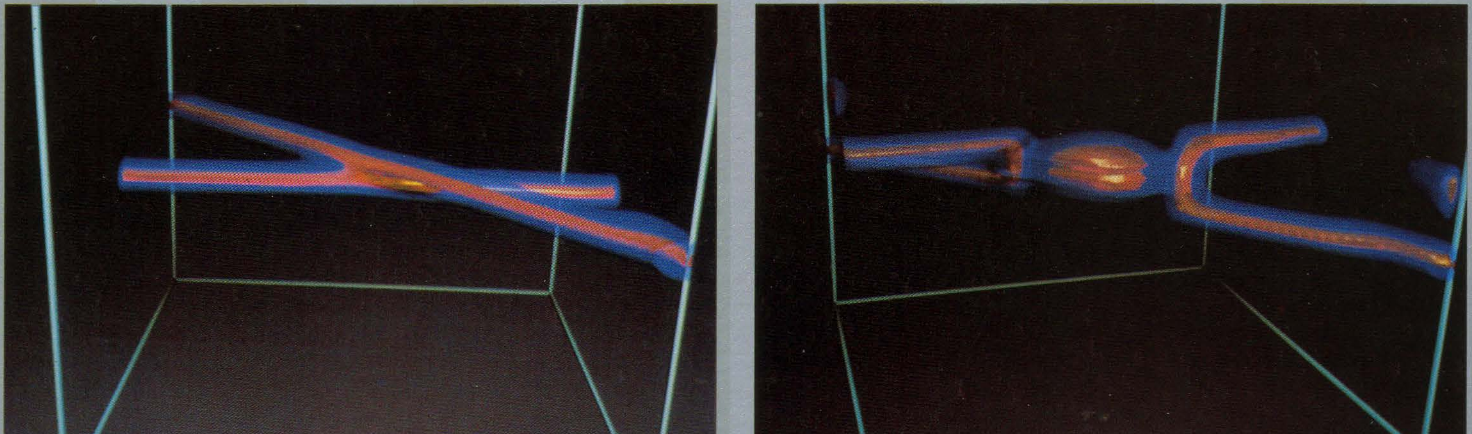
The big bang theory is the cosmology favored by most astrophysicists today. Competing theories of the origin of the universe fell away quickly after the cosmic microwave background radiation was discovered in 1965. The microwave background radiation, which permeates the entire universe, is believed to be a remnant of the earliest moments of the big bang and to astrophysicists stands as convincing evidence that the big bang indeed occurred. But the theory still has some loose ends, like galaxies and clusters of galaxies; the big bang doesn't satisfactorily account for the formation of such large structures. To tie up these loose ends, physicists are exploring the possibility that another type of remnant persists from the big bang: cosmic strings.

Although much thinner than the diameter of a single atomic nucleus, a cosmic string, if such a thing exists, wields enough gravity to bring dispersed matter together to form galaxies and other large structures. Richard Matzner, at the Center for Relativity at the University of Texas at Austin, is using the CRAY X-MP/48 computer system at the University of Illinois to model the interactions of cosmic strings. He is testing a hypothesis about strings: if a string loops back and intersects itself, then the looped part will break off and the two remaining ends of the string will rejoin. The string loops shed during this process could form gravitational "seeds" around which galaxies and other structures could form.

By solving equations from field theory that describe the physics of cosmic strings and rendering the results graphically, Matzner is able to create animated models of intersecting strings. His results show that in most cases the loop-shedding hypothesis is sound; a string bent back to intersect itself will shed a loop and then rejoin its free ends. Only when a string falls back upon itself at a 180° angle (that is, when two antiparallel segments pass through each other) does loop formation and reconnection fail to occur. "The calculations support an assumption about strings that had not been tested very well," Matzner says. "The results would have been more dramatic if I had discovered that they do not reconnect, but according to my calculations they do."

The calculations were carried out within a three-dimensional grid having 64 cubes per edge. The memory-resident program occupies about 4.5 million words of memory on the Cray system. "A sequence of 200 to 250 frames takes about two-and-one-half hours to generate," Matzner says. "It takes this long because we write out the whole volume of cubes every cycle and we dump out a number of fields that have to be computed from the fundamental fields. As a result, I/O makes up at least half of the total run time. A lot of time also is spent updating boundary conditions, which are two-dimensional instead of three-dimensional, and so are more difficult to vectorize."

The next step in this research involves extending the model to test the behavior of superconducting strings, which, if they exist, may explain another aspect of the structure of the universe. The intense electromagnetic fields that would emanate from superconducting strings might repel matter, explaining the large voids found between galactic clusters. The complexity of these simulations will require a larger memory than is available on the CRAY X-MP/48 system. To access the needed memory, Matzner will transfer his calculations to the CRAY-2S/4-128 system scheduled for installation at the University of Illinois at Urbana-Champaign in the fourth quarter of 1988. "It's possible that if we add the complication of current, or even superconductivity without current, it might change the way the strings evolve," explains



Two frames from a movie of a simulated cosmic string collision. In the left image the strings have just touched. Their speed is 0.75 times the speed of light relative to the box. The strings are oriented 45° from antiparallel. The right image shows the initial result of the collision: an annihilation of the central parts of the strings, with a release of radiation (the central bubble). The strings "cross connect," creating new strings with straight segments joined by kinks. The kinks travel out into the undisturbed segments at the speed of light.

Matzner. "We also want to check the possibility of huge quantities of electromagnetic radiation being produced when superconducting strings break, which would have significant astrophysical implications."

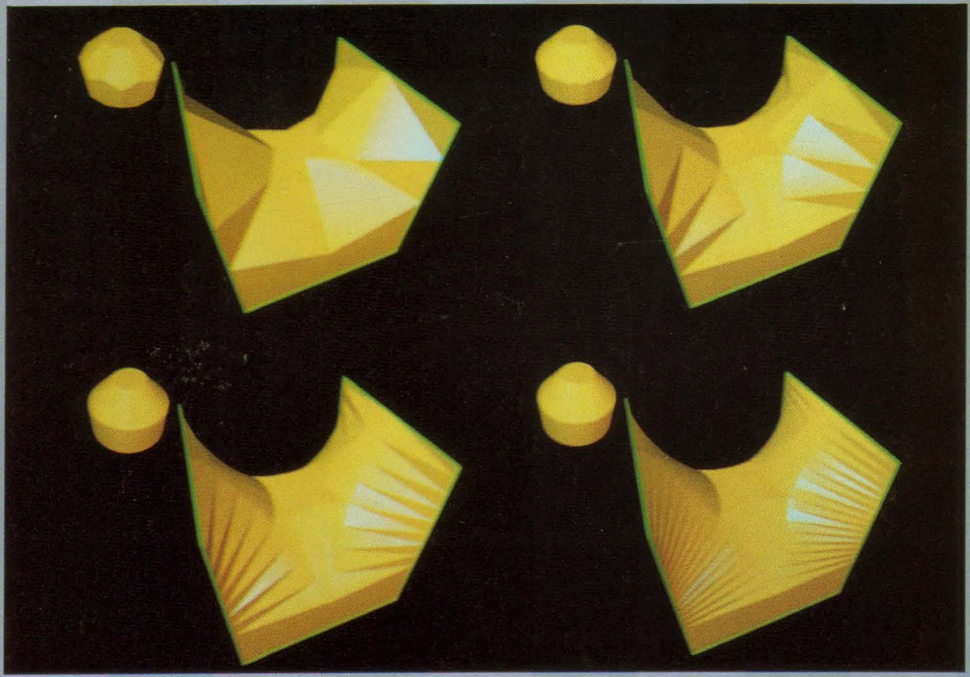
The nature of the universe, whether or not it is built on a lattice of cosmic strings, will continue to intrigue and confound human curiosity. When that curiosity encounters the limitations of experimental science, computational science can continue to provide insight. Although nature itself finally judges scientific theories, large-scale computations can help researchers separate the possible from the impossible, and point the way toward the probable.

Optimal surfaces shape up

The shape of the interface joining two soap bubbles may not seem like a compelling inspiration for high-level mathematical research, but it raises questions about surface shapes generally. When more than a few bubbles are joined, the shapes of their interfaces become staggeringly difficult to predict. Discovering the rules that determine the natural, or optimal, shapes of such surfaces is the hope of mathematicians Frederick J. Almgren, a professor of mathematics at Princeton University, and Jean E. Taylor, a professor of mathematics at Rutgers University. Uncovering the rules that govern the geometry of optimal surfaces in complex configurations, whether of bubbles, biological cells, crystals, or grains in a metal, would benefit many fields of natural science and engineering.

Almgren and Taylor believe they can solve some longstanding problems in the prediction of optimal surfaces with the help of the University of Minnesota's CRAY-2 computer system. The researchers head the Minimal Surface Team of the University of Minnesota's Geometry Supercomputer Project. Organized by U of M mathematics professor Albert Marden, the project brings together geometricians from the United States and Europe to address problems that have languished unsolved for years, or centuries, for lack of computing power. The project has received a three-year $1.5 million grant from the National Science Foundation and receives additional support from the University of Minnesota, Cray Research, and Sun Microsystems.

The research on optimal surfaces proceeds from the observation that nature strives to minimize free energy; that is, it strives toward equilibrium. Energy minimization, therefore, to some degree governs natural shapes, such as that of the surface shared by adjoining bubbles. In the case of bubble interfaces, the challenge of computing optimal, or least energy, surfaces involves computing the surfaces of minimum area. "The simple principle of surface area minimization makes a single bubble spherical," Almgren explains, "but when several bubbles are joined, fantastic geometries emerge as they assume the lowest energy configuration. We want to describe and predict these complex geometries mathematically."



Four crystalline minimal surfaces. The shape to the upper left of each surface represents a single crystal at equilibrium.

But minimal surface calculations will not always reveal the shapes of surfaces found in nature. Crystals, for example, have preferred cleavage planes that represent low-energy surfaces that may not be least-area surfaces. Nonetheless, the researchers believe that the principles governing the shapes of bubble interfaces can be used to predict the shapes of crystal surfaces as well. "We should be able to feed the energy considerations derived from chemistry and physics into the computation and ask 'Now what are the surfaces of minimal energy?,' that is, we should be able to determine the analogs of soap films for a crystal structure," Taylor explains.

Similar considerations determine the shapes of interfaces between biological cells and between grains in metals. In the case of metals, studying these geometries should provide a better understanding of why metals are strong, when they might fail, and how a metal will change when heated near its annealing point.

Optimal surface calculations are related to a family of similar mathematical problems, the so-called optimal partitioning problems. The solutions of these problems, for simple cases, have led to such conjectures as that honeycomb chambers are hexagonal simply to provide the most efficient means for storing honey. "However, the corresponding problem in three dimensions has not been solved formally," Almgren says. "The problem has been completely inaccessible until just recently, because we now have supercomputers. Lord Kelvin proposed a geometry for the three-dimensional honeycomb problem, and we hope to test his conjecture."

The central long-term challenge of the research is learning how to navigate computationally through the mathematical spaces of varying geometries in which optimal configurations are embedded. "Given a geometrical starting point, we want to understand how to sail either with or across the energy-driven flows in such spaces," Almgren says.

To arrive at this goal the researchers are solving their computational problems with Sun 360 Workstations, a Pixar image computer, and the CRAY-2 system. In addition to custom code written primarily in C, the researchers use a quadratic and linear programming package. Graphics are rendered on the Pixar computer and, with the graphics package CHEYENNE from Princeton University's computer science department, on the Sun Workstations.

One of the most useful tools developed by the team is a "surface evolver" algorithm, which is largely the creation of team member Ken Brakke of Susquehanna University in Pennsylvania. The evolver accepts surfaces and solids of fixed combinatorial geometry and deforms them to minimize area or other surface energies subject to the constraint of preserving the volumes of solid regions or taking into account pressure energies. A similar program, a "solid evolver" algorithm, is being developed to handle the simultaneous evolution of many regions.

A recent test case for the solid evolver modeled the evolution of a dumbbell-shaped oil drop suspended in another viscous fluid. The simulation showed that the drop breaks apart into smaller drops before arriving at its final spherical geometry. For this problem the locations of the two fluids were defined by 5000 control points of two colors (representing the two liquids). The CRAY-2 system computed the remaining geometry in only four minutes. The geometry comprised 37,939 facets, 65,562 edges, and 32,834 vertices (omitting those going to infinity). Once the starting geometry was defined, the CRAY-2 system computed the desired motions of the control points to minimize most rapidly the interface area between sites of different colors, while preserving the total volume of each color. The algorithm discards sites along the way that are unused, substantially speeding the geometry recomputation at each step. "Overnight the Cray system can generate a complete evolution for analysis that would take a month of full-time computing on one of our Sun Workstations," Almgren says.

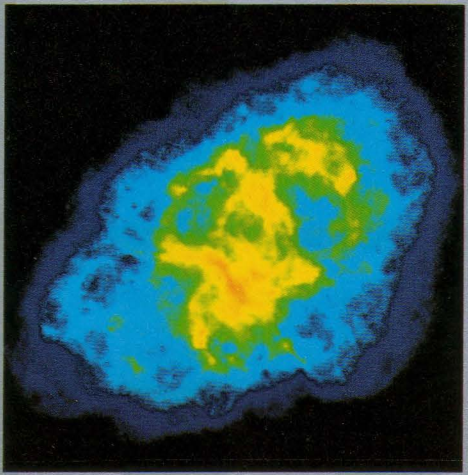
As the research develops, the researchers will be able to predict increasingly complex geometries. More and more real-world problems will become solvable with the knowledge gained, such as one that Almgren encountered some years ago. "An engineer working on a rocket fuel tank called me with a problem," he recalls. "The tank held liquid fuel and had to work in space, but without gravity the fuel might end up stranded on the side of the tank opposite the fuel port. The engineering solution was to use capillary attraction — to use baffles to attract the fuel — and this person asked if I could compute the best configuration. At the time I had to tell him no, but I think that with our new knowledge and tools we could do it today."

Cray system reveals stellar insights

About 930 years ago a bright star exploded, leaving behind its incredibly dense spinning nuclear core and a turbulent gas cloud. The star's remains formed the Crab Nebula, a supernova remnant that has become a fascinating laboratory for gas dynamics and high-energy plasma processes. Today, with a Cray computer system and the world's most powerful radio telescope, astronomers are creating the most detailed images of supernova remnants ever observed. The images already have provided insights into the creation and distribution of elements that form planets and life.

The supernova images were gathered by the Very Large Array (VLA) telescope at the U.S. National Radio Astronomy Observatory (NRAO) in Socorro, New Mexico. The VLA collects readings from 27 satellite dishes spread over a 35 kilometer area. But because the telescope's eyes can see more than the observatory's computers can produce graphically, the astronomers there turned to the University of Toronto's CRAY X-MP/22 supercomputer to process the data.

The University of Toronto (U of T) Radio Astronomy Group has been developing, adapting, and optimizing special image-processing software for the Cray system, such as the NRAO's Astronomical Image Processing System code (AIPS) and the WERONG suite of programs developed by Robert Sault of the University of Illinois.



Radio image of the Crab Nebula, a remnant of a supernova that exploded in 1054 AD. This is the most detailed image of the Crab Nebula that has been produced to date.

"We process the data here to make some very detailed maps of complex phenomena," says U of T astronomer Philipp Kronberg. "These images are most easily manipulated on a Cray system." He adds, "Using the Cray system enables us to make discoveries in astronomy that we otherwise were not in a position to make."

So far, NRAO astronomers Rick Perley and Robert Braun have used the U of T supercomputer to produce three sequential observations of Cassiopeia A, a supernova explosion that happened nearly 300 years ago. "Thanks to the very detailed map that we can get from the Cray system, we actually can see changes in detail from 1983 to 1987 from Cassiopeia A," says Kronberg. One of Kronberg's students, Michael Bietenholz, is making images of the Crab Nebula, which has a pulsar in its center that continuously feeds energy into the surrounding gas cloud.

The images enable the astronomers to explore the detailed gas dynamical processes that occur in the supernova. For example, the Crab Nebula's pulsar is about as big as the city of Toronto and spins about 30 revolutions per second. The pulsar is composed of nuclear material with a density of 100 million tons per cubic centimeter, which explains how it can spin so quickly and not fly apart.

"This spinning pulsar also has a strong magnetic field linked to it — so what you have is an enormous dynamo that accelerates gas at relativistic speeds and somehow feeds it into the exploding gas cloud," Kronberg explains. "Furthermore, the conversion of energy appears to be near 100 percent. How that can happen in nature is a mystery. If we knew more about that process, we might be able to generate electricity better than we do now," he adds.

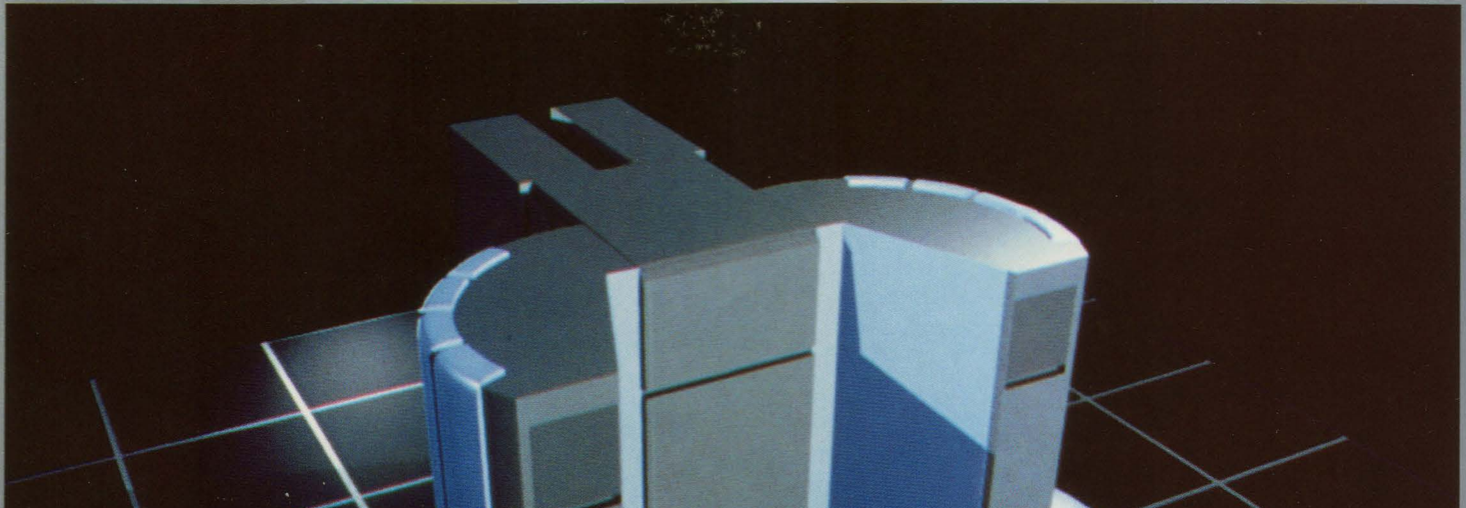
The supernova images produced by U of T's Cray system are important for another reason: supernova explosions create the vast bulk of heavy elements, from nickel and copper to lead and uranium. The pressure and heat from the explosions cause atoms of light elements to fuse and form heavier elements. "The solar system was formed from these heavy elements," says Kronberg. "We are witnessing the process with these images produced on the Cray system. The images show how heavy elements are being fed into the rarified gas between the star from the factory that makes them. The factory is, in effect, the explosion of the star, which is really a super nuclear bomb."

According to Kronberg, the key to producing these high-resolution images has been the efficient WERONG code, the Cray BENCHLIB routines, and the efforts of U of T's Patricia Monger, who optimized the code in a COS environment. "By fine-tuning the code — mostly by optimizing management of the I/O and our SSD solid-state storage device — we have cut costs and increased speed by a factor of eight," Kronberg says.

"Besides handling the I/O, the key to optimizing the code is using the fastest algorithms for doing convolutions and fast Fourier transforms (FFTs). We have found it very efficient to use Cray BENCHLIB canned routines, which are prewritten routines to perform FFTs."

Next, the team will go from two-dimensional image processing to three-dimensional image processing, which few sites are capable of performing, according to Kronberg. "When we upgrade our computer we will be able to tackle yet another level of detail and complexity," he says.

GALLERY



This image of a CRAY Y-MP computer system was produced on a Cray computer system by Gray Lorig and Nancy Rowe of Cray Research, Inc., with the Oasis animation software package. Please send Gallery submissions to CRAY CHANNELS at the address inside the front cover.

Worldwide business centers

Corporate Headquarters
608 Second Avenue South
Minneapolis, MN 55402
(612) 333-5889, telex 6879144

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

Domestic

Central Region Office
3375 Mitchell Lane
Boulder, CO 80301
(303) 447-8855
Albuquerque
Chicago
Colorado Springs
Denver
Detroit
Los Alamos
Minneapolis/St. Paul
St. Louis

Eastern Region Office
4041 Powder Mill Road, Suite 600
Calverton, MD 20705
(301) 595-5100
Atlanta
Bernardsville
Boston
Cincinnati
Darien
Huntsville
New York
Pittsburgh
Tampa
Washington, D.C.

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

Western Region Office
3130 Crow Canyon Place
Suite 400
San Ramon, CA 94583
(415) 866-0600
Huntington Beach
Los Angeles
Salt Lake City
Scottsdale
Seattle
Sunnyvale

International

Cray Asia/Pacific Inc.
Suite 1708
One Pacific Place
88 Queensway
Hong Kong
(5)-8109321, telex 89765

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

Cray Computadores do Brasil, Ltda.
Av Armando Lombardi 205
Salas 307 A 310
22621 Rio de Janeiro, Brazil
(21) 399-7808, telex 02136111

Cray Research (Australia) Pty. Ltd.
Victorian Technology Centre
283 Normandy Road
Port Melbourne, Australia 3207
(03)-647-9833, telex 32111

Cray Research France S.A.
7 Rue de Tilsitt
75017 Paris, France
(01)-47-66-01-55, telex 660568

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

Cray Research Japan, Limited
Ichibancho Eight One Building
6-4 Ichibancho
Chiyoda-Ku, Tokyo 102, Japan
(03)-239-0711, telex 28893

Cray Research S.A.E.
Pasco de la Castellana, 141, Plta. 20
28046 Madrid, Spain
(01) 459-0150, telex 48998

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

Cray Research (Suisse) S.A.
Route de Pre-Bois 20
P.O. Box 534
1215 Geneva-15, Switzerland
(022) 980-091, fax 4122980516

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

CRAY, CRAY-1, SSD, UNICOS, and CRAY CHANNELS are registered trademarks and CFT, CFT77, CFT2, COS, CRAY-2, CRAY X-MP EA, CRAY X-MP, CRAY Y-MP, CSIM, HSX, IOS, SEGLDR, and SUPERLINK are trademarks of Cray Research, Inc. Apollo is a registered trademark of Apollo Computer Inc. The UNICOS operating system is derived from the AT&T UNIX System V operating system. UNIX is a registered trademark of AT&T. UNICOS is also based, in part, on the Fourth Berkeley Software Distribution under license from The Regents of the University of California. LANlord is a trademark of Computer Network Technology Corporation. VAX and UNIBUS are trademarks of Digital Equipment Corporation. X Window System is a trademark of the Massachusetts Institute of Technology. HYPERchannel is a registered trademark of Network Systems Corporation. Ramtek is a trademark of Ramtek corporation. Sun Workstation is a registered trademark of Sun Microsystems, Inc. Ethernet is a registered trademark of Xerox Corporation.