At Last There Is a Way to Take It with You

At a week-long workshop, crystallographers struggle to write a portable program that will run on any computer system

Computers are supposed to help researchers save time, not waste it. For the most part, the machines serve their masters admirably, but in one important respect they fail miserably. Because computers are a lot like people—they are all individuals—programs written to run on one machine must often be extensively revised before they will run on another. And researchers therefore waste countless hours hunched over a keypunch or a remote terminal reworking their programs whenever they buy a new computer or borrow a program from a colleague with a different machine.

Neither computer manufacturers nor federal funding agencies have shown much interest in finding ways to resolve this widely appreciated lack of portability. But last November a group of 11 crystallographers engaged in a unique experiment aimed at overcoming the problem. In a week-long workshop, the researchers wrote a program designed to run on any medium- or large-sized computer. Testing of the program in many laboratories over the next several months will determine how successful the experiment is. One thing that may help the portable program gain wide acceptance is that it will be one member of a set of crystallography programs being assembled by Sid Hall of the University of Western Australia, James Stewart of the University of Maryland, and numerous collaborators. When completed, the set will allow a crystallographer armed only with experimental diffraction data and a computer to determine the structure of any crystal without having to write special programs to accommodate the peculiarities of either the computer or the data.

The workshop was held at the National Resource for Computation in Chemistry (NRCC), which is quartered at the Lawrence Berkeley Laboratory.* The participants wrote a program called multiple isomorphous replacement phasing, a fancy name for a technique widely used to determine the phases of the diffracted waves in x-ray or neutron crystallography. Participants met every day for eight straight days, working from 8 a.m. to 12 midnight. At the end, an all-new, nearly functioning program with 3000 lines of code was in hand. A second get-together is scheduled for this spring to iron out any remaining wrinkles; then the testing begins. One expectation is that the concepts used in the writing may find their way into attempts to construct portable computer programs in fields other than crystallography.

Nonportability of computer programs is hardly a new issue. Since the day when the second computer was born, researchers have had to confront the problem. Now, there are dozens of manufacturers, each with several models. All in all, estimates Bruce Barnes, a computer scientist at the National Science Foundation (NSF), among any 100 major computer centers, there may be 40 different configurations of computers that cannot run one another's programs. Moreover, the prevailing sentiment in the research community is that computer makers have shown little inclination to do anything to help. In fact, manufacturers are said to fear that standardization makes it too easy for buyers to switch brands.

Although no statistics are kept, the cumulative amount of time spent in modifying computer programs sometimes seems to rival the time actually doing research. Numerous anecdotal accounts exist of postdoctoral associates devoting several months to converting programs written for one machine to run on another. Even entire laboratories have dropped out of the data processing business for similar periods of time during conversion of programs to run on a new computer system.

For crystallographers, two recent developments have made nonportability a more critical issue than before. One such event is the phasing in of a new version of Fortran, the computer language most often used in large-scale scientific computation. Despite pleas by researchers, an American National Standards Institute committee in 1978 issued a new standard, called Fortran 77, which had been formulated so that programs written containing certain statements allowed under the old rules would not run on machines operating under the new ones. Stewart says that this turn of events is one of the factors that motivated him to begin developing a set of portable crystallography programs. Says Stewart, "I rewrote my codes in 1958, 1963, and 1972, and now I am going to have to do it again. But this time I want them so they will never have to be redone again because of changes in computers."

Just as important is the trend, begun about 4 years ago, toward the use of high-performance mini (also called midi) computers for large-scale computation or number crunching. The machines are sufficiently inexpensive (a few hundred thousand dollars) that individual departments or even groups of investigators can afford them. Researchers often find the smaller machines less expensive and more convenient than a larger computer in a computer center. In particular, in the last 2 years, several crystallography groups have acquired such machines and have been faced with the need to convert their programs to run on them. In the past, each group would have proceeded on its own, an approach with an inherently large degree of duplication of effort.

Although there is no inherent reason why portable programs have to be written by committee, in the case of the crystallographers the two concepts seemed to go hand in hand. Says Arthur Olson of the NRCC, we thought "that if agreement could be reached on some [standards and definitions] the huge task might be broken down into a group effort with many people contributing to a common comprehensive set of 'workhorse' programs. These programs would be general in scope, efficient, well documented, and transportable from laboratory to laboratory and machine to machine." The specific idea for a workshop

*The participants were: Richard Alden and Steven Freer, University of California at San Diego; Robert Munn and James Stewart, University of Maryland; George Reeke, Rockefeller University; Steven Sherriff, University of California at Los Angeles; Jurgen Sygush, University of Sherbrooke, Canada; Lynn TenEyck, University of Oregon; Keith Watenpaugh, University of Washington; Sid Hall, University of Western Australia; and Arthur Olson, NRCC.

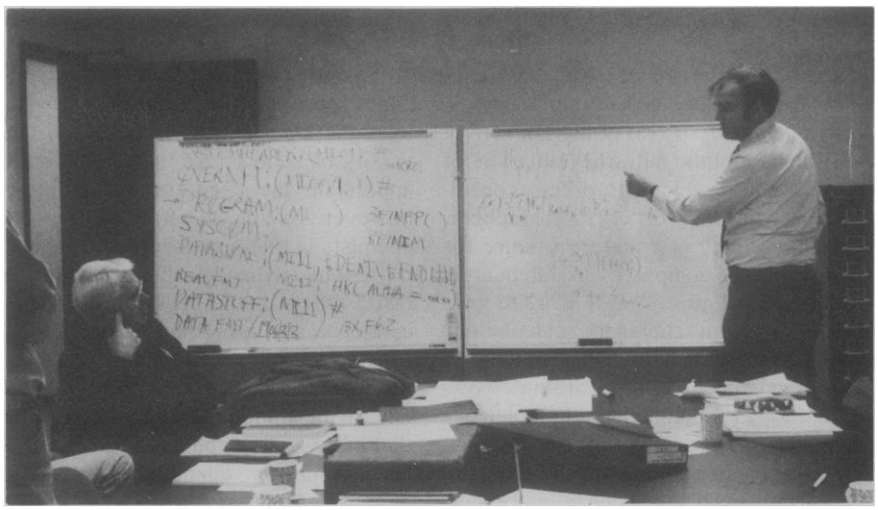
0036-8075/80/0215-0746$00.50/0

in which computer programs would be developed and checked out rather than just discussed came from Keith Watenpaugh of the University of Washington, who had urged similar exercises in the past, but without success.

Choice of multiple isomorphous replacement phasing as the program to be written and tested was meant to supply as stiff a challenge as possible to the concept of portable programs. In crystallography, one obtains the positions of the atoms in a crystal by Fourier transformation of the diffracted x-ray or neutron waves. A problem arises, however, because detectors record only the intensities of the diffracted waves, whereas their phases must also be known in order to do the transformation. A common solution to the problem in very complex structures, such as those of proteins, is to record diffraction data on several crystals: a parent and several derivatives that differ in that a few atoms are replaced by heavy metals. From the small differences in the intensities, crystallographers can estimate the phases. Further analysis, called refinement, produces the most likely phase values. As Watenpaugh wrote at the close of the workshop, "While other fields of crystallography will place different demands on the [portable set of programs], they will not be greater than those required in this experiment."

The approach to portability followed at the workshop was to use certain ideas developed by Robert Munn of Maryland and Stewart for the still-to-be-completed set of crystallography programs, XTAL 80. There are three key concepts that lend portability to XTAL 80: structured programs, preprocessors, and macros. All, to be understood, require a quick look at how current programming practices inhibit portability.

A major influence on programming is that, while manufacturers offer machines that will run programs written in standard Fortran, they also give these same computers the option of operating with Fortran having certain extra capabilities, so-called extended Fortran. Since no two computer makers design their machines and the programs that run them (systems software) in exactly the same way, one manufacturer's extended Fortran is seldom compatible with that of another. Expert programmers naturally try to squeeze the last drop of performance out of their machines. To do this requires taking advantage of the peculiarities of each individual computer system. Converting a program written in such a way may mean much more than just changing a few lines; it may mean reorganizing a large part of the program.



Workshop participants tended to be specialists in either computer programming or in crystallographic theory. Here a programming expert, James Stewart of the University of Maryland, sits beside a board covered with computer code, while theorist Jurgen Sygush of the University of Sherbrooke discusses his mathematical formulation of the phasing problem in crystallography. [Photo by Arthur Olson, NRCC]

Structured programming begins the process of enhancing portability by forcing a discipline upon the writer. In particular, the 10-year-old concept requires the programmer to divide a program into modules that perform different functions and to keep the modules clearly separated. This approach ensures that changes can be made to any module without major alterations in the organization of the program.

An extra feature of modularity is that it made the work of the NRCC workshop participants easier because groups of two or three crystallographers could write the modules separately. Because each module of the program had to be able to communicate with others, however, the programmers had to adhere to precise but arduously worked out conventions in their writing. As described by Olson, the process was an iterative one, with alternate sessions of writing of modules by the subgroups and full workshop meetings to discuss ways to handle intermodule communication. Moreover, says Munn, because each participant wanted the program to be compatible with his machine, close attention was paid to making sure that the conventions adopted really were conducive to portability.

Structured programming is made considerably easier with the use of a preprocessor because certain non-Fortran commands and other constructs are permitted that enhance the programmer's ability to design modules and keep their functions separate. The preprocessor is a portable program that acts like a translator to convert a program written in the structured programming language into the specific form of Fortran accepted by a particular machine.

To make the structured program independent of a particular computer system, an additional feature, the macro, is required. A macro is a sequence of instructions that is identified with a key word. Whenever that word appears in the structured program, the sequence of instructions is inserted in the Fortran program by the preprocessor. Macros enhance portability because all the peculiarities of a given machine can be isolated and defined in a few macros; it is not necessary to write the entire program to fit a specific computer system.

Macros have another role. Programs written in Fortran could be portable if only features of the language available on every machine were used, but such programs would be inefficient. The use of macros makes the portable program efficient because the preprocessor translates the key words defined in macros to Fortran commands optimized for the particular computer system being used.

The relevance of these concepts to solving the problem of nonportable computer programs will be determined by how widely used they become. For the moment, says NSF's Barnes, the multiple isomorphous replacement phasing program written at the workshop serves as a test-bed that will allow researchers to see if ideas produced over the years by computer scientists work when applied to real computational problems. One factor that can affect the future use of these ideas in disciplines other than crystallography is the nature of the computation.

Among computational chemists, for

example, there are two other large groups: those who calculate the electronic structure of molecules and those who simulate collisions between molecules. But, as Donald Secrest of the University of Illinois points out, the situation of these quantum chemists differs considerably from that of crystallographers. Crystallography is a mature science for which today's computers are adequately powerful and pretty well worked out. Some quantum chemistry calculations are routine; but at the forefront of research existing computers are only powerful enough to handle simple molecules in detailed, first-principles calculations. Moreover, programs are continually being improved to squeeze more performance from the machines or to try a new computational method. George Jeffrey of the University of Pittsburgh estimates that the situation of quantum chemists is somewhat akin to that of crystallographers 15 years ago, when all computers were very much slower.

For these reasons, quantum chemists do not seem to feel the sense of urgency about portable programs that crystallographers do. Frank Harris of the University of Utah thinks that the failure to appreciate the need for portability is, however, having its effect on quantum chemists in that new theoretical ideas do not circulate through the community as fast as they should because researchers cannot easily reproduce one another's programs. Thus, even though their situations differ, the need of quantum chemists for portable programs is as great as that of crystallographers. Nonetheless, it seems unlikely quantum chemists (or anybody else) will adopt the crystallographers' solutions without modifying them to fit their own circumstances.

It would be a feather in the NRCC's cap if the portable program experiment does turn out to be successful. One of the major reasons portable programs are only now being constructed has to do with the way in which federal agencies support research. In chemistry, for example, most computer programs have been developed by scientists as part of their ongoing research projects, with funding provided for the overall project but not specifically for program development. In the past, says Edward Hayes of NSF, proposals to develop new computer codes generally have not been well received by peer reviewers. Since acceptance or renewal of proposals has depended primarily on new chemistry results, investigators understandably have not placed a high priority on refinements of computer programs such as portability. In some cases, such work may be "bootlegged" on grants not intended for the purpose. The NRCC, which is barely 2 years old and is still seeking its niche, is perhaps filling a gap in its sponsorship of such efforts as the portable program workshop.—ARTHUR L. ROBINSON

Concern Rising About the Next Big Quake

"Interesting" phenomena in southern California are giving the new Earthquake Prediction Evaluation Council something to ponder

The earthquake that struck last month east of San Francisco near Livermore—the second moderate quake in the Bay Area in 5 months—raised familiar questions. Does that mean that the Big One is coming? Is California finally going to slip into the ocean? Other than assuring everyone that California will never slip into the ocean, scientists cannot say what, if anything, the Livermore quakes or the larger Imperial Valley earthquake of last October imply about the expected great earthquake, which would be ten thousand times more powerful than these moderate ones.

The Livermore quake appears to say little about future earthquakes, but researchers have observed unusual phenomena in southern California that, they admit, would have already prompted an official prediction of some sort if a more empirical approach were taken here, as it is in the People's Republic of China. Americans familiar with the situation are expressing a "heightened concern," but, having been burned before by phenomena that are poorly understood, they remain wary of drawing any conclusions. The data from southern California will be closely examined by the new National Earthquake Prediction Evaluation Council that will aid the director of the U.S. Geological Survey in issuing any formal predictions, but no action is expected to be taken by it soon.

What has happened is that the San Andreas fault, with Los Angeles on one side and the Mojave Desert on the other, began to be pulled apart in 1979 instead of being increasingly squeezed together, as it had been in the past. At the same time, the strain that tends to force the two sides to slip past one another continues to build.

Although it is "definitely not a good sign," the expansion in southern California comes as almost a relief to researchers because it has some predictable consequences. Studying it "is so much better," says one specialist, "than chasing precursors"—those poorly understood phenomena such as barking dogs, bubbling springs, or bulges in the crust.

Researchers are taking the abrupt change in the behavior of the San Andreas fault seriously because it appears to be undeniably real and widespread. Two groups of researchers have detected the expansion using two methods that could hardly be more unlike. James Savage's group at the U.S. Geological Survey at Menlo Park measured the lengths of 20- to 30-kilometer long survey lines across and along the San Andreas fault to an accuracy of 1 centimeter with a portable laser device called a geodolite. A. E. Neill and his colleagues at the Jet Propulsion Laboratory measured the distance between two radiotelescopes 190 kilometers apart on either side of the fault with an accuracy of 5 centimeters. They determined the distance between the two by comparing the precisely measured arrival times of radio noise from a quasar millions of light years away.

Both methods showed about the same change at about the same time. The geodolite first detected the switch from compression of the fault to an expansion in January 1979 near the Salton Sea southeast of Los Angeles, or about 30 kilometers from the center of October's Imperial Valley earthquake. A similar expansion was also detected on the San Andreas due north of Los Angeles, and an even larger one was observed near Palmdale, which is northeast of the city.

Although the next large earthquake on the San Andreas fault would arrive sooner if the present expansion contin-

0036-8075/80/0215-0748$00.50/0