Crystal Data


Crystal data for uranium phosphorus pentoxide, UPO₅, and uranium arsenic pentoxide, UAsO₅.*

By H. BARTEN,† Netherlands Energy Research Foundation ECN, Petten, The Netherlands

(Received 18 March 1986; accepted 29 July 1986)

Abstract

Guinier powder patterns \( d(Cu Kα_1) = 1.5406 \) Å of UPO₅ and UAsO₅ have been indexed. UPO₅ is triclinic. The cell dimensions are: \( a = 5.472(2), b = 5.646(2), c = 7.027(1) \) Å, \( α = 87.39(2), β = 93.39(2), γ = 106.51(1)° \). \( V = 207.7 \) Å³, \( Z = 2, D_x = 5.58 \) g cm⁻³. The new compound UAsO₅ appears monoclinic. The unit cell has dimensions: \( a = 8.833(2), b = 6.858(1), c = 7.151(1) \) Å, \( β = 96.88(1)° \). \( V = 430.1 \) Å³, \( Z = 4, D_x = 6.07 \) g cm⁻³. The space group is \( P2_1/c \). The data show the presence of a strong triclinic \( C \) subcell: \( a = 5.591(1), b = 5.591(1), c = 7.151(1) \) Å, \( α = 84.57(1), β = 95.43(1), γ = 104.35(1)° \), which suggests that both compounds are structurally related. The JCPDS Diffraction File No. for UPO₅ is 37-1462 and for UAsO₅ 37-1461.

© 1986 International Union of Crystallography

Computer Programs


Use of a new array processor in the restrained least-squares procedure of Hendrickson and Konnert.

By GERSON H. COHEN, Laboratory of Molecular Biology, NIDDK, National Institutes of Health, Bethesda, MD 20892, USA

(Received 1 May 1986; accepted 13 July 1986)

Abstract


Introduction

The restrained least-squares refinement program PROLSQ (Konnert, 1976; Hendrickson, 1985) is one of the more common refinement tools used by protein crystallographers today. With structure factors and derivatives calculated by conventional summation it is a time-consuming procedure. The run time for this procedure may be reduced dramatically by the use of a ‘super’ computer or a vector processor. Hendrickson (1985) originally developed and maintained the PROLSQ program on a Texas Instruments ASC which was a mainframe computer with efficient vector processing capabilities and a vectorizing Fortran compiler. More recently, Furey, Wang & Sax (1982) have implemented the program on a Floating Point Systems AP190-L array processor linked to a Digital Equipment Corporation PDP-10 while Hendrickson’s group (Hendrickson, 1985) has used a Floating Point Systems AP 120B array processor to improve the run time on a VAX 11/780.

This is the first reported use of the Numerix MARS-432 array processor for crystallographic applications. Our experience suggests that this hardware will permit one to exceed the computational speed of a large mainframe computer at a fraction of its cost.

Description of the array processor and its software

The Numerix MARS-432 array processor is a pipeline type of machine. Its arithmetic units include two adders and one multiplier and its addressing units can reference 16 Mwords (1 Mword = 1024² words) of 32-bit memory. Results from the arithmetic units, memory, as well as some registers, are available through FIFO (first in, first out) queues. The data memory is arranged in such a manner as to permit the
Two varieties of data memory are available. The first allocation of 64 Kwords (1 Kword = 1024 words) is restricted to a fast, 100 ns, static memory. Data memory may be increased by the addition of another 64 Kwords, but a slower, 300 ns, dynamic memory is also available in larger increments. Owing to the organization of the slower memory into banks, serial addresses can still be delivered each 100 ns. Our system consists of 64 Kwords of static memory and 1 Mword of dynamic memory. The system also contains the minimum allocation of four Kwords of program memory of which the PROLSQ program uses about one third.

The structure of the MARS-432 data word is essentially the 32-bit code used by Digital Equipment Corporation on their VAX and PDP-11 series with the exception that the two half-words of floating-point items are reversed. This reversal is transparent to the user and occurs automatically during data transfer.

We use the Fortran Development System available from the vendor. The compiler generates code optimized for the hardware from Fortran sources which adhere closely to the specifications of Digital Equipment Corporation's implementation of Fortran 77. A large number of highly optimized vector routines are available in the vendor-supplied software package.

The array processor has been interfaced to a VAX 11/780 equipped with a floating-point accelerator and running the VMS (version 4.3) operating system. Communications between the host and the array processor are managed in the host by a series of vendor-supplied subroutine calls. This communication can also be controlled from the array processor after the code has been loaded and the program is running.

**Implementation**

The modifications to the PROLSQ program were relatively straightforward. During the structure-factor refinement phase of the program, the calculation of structure factors and derivatives is carried out in the subroutine CALC. The derivatives are then accumulated into a sparse normal matrix in the subroutine SFREF. This subroutine is also responsible for the input and output of the reflection data and the results of the calculation. The matrix accumulation portions of SFREF were transferred to the CALCl subroutine and the resultant whole was recast into an independent main program for the array processor. The remainder of SFREF was modified to provide the appropriate linkages to load and start the array processor and perform the necessary communications with it to complete the calculation. Fortran loops representing vector operations in the array processor code have been replaced by calls to appropriate vendor-supplied subroutines wherever possible to improve the execution performance. The resulting array processor program is, therefore, a Fortran main program containing Fortran statements for scalar calculations and calling several of the special vector subroutines for most array operations. Data transfer from the VAX to the array processor and return involves only reflection data, 250 per packet, once the calculation is under way.

**Results and comparisons**

We have tested the modified program on several structures: the tripeptide Gly-Phe-Gly (Marsh & Glusker, 1961), bovine pancreatic trypsin inhibitor (Wlodawer, Walter, Huber & Sjolin, 1984), ribonuclease A (Wlodawer, Bott & Sjolin, 1982) and Rhizopus pepsin. We will quote here our results with Rhizopus pepsin (P, 212121, a = 60.31, b = 60.60, c = 106.97 Å, nominal resolution = 1.75 Å), the refinement of which has recently been completed in our laboratory (Suguna, Bott, Padlan, Sheriff, Cohen & Davies, 1986). Coordinates of 2791 atoms with individual thermal parameters were varied and 31915 reflection data to 1.75 Å were included in the calculations. Run times with different versions of PROLSQ are summarized in Table 1. In all cases, the times shown and discussed below are for the portion of the program which calculates the structure factors and derivatives only. For the VAX implementations, the total machine time will be longer than the CPU time and depend on a variety of factors, such as other activity on the host CPU and the file input/output system.

The array processor implementation showed a 27-fold improvement in execution speed for the space-group-general version of the program and an 18-fold improvement for the P, 212121 space-group-specific version. The discrepancy in the run-time ratios arises because the VAX space-group-specific code had been optimized to run on the VAX whereas the VAX space-group-general version is essentially the distribution version as received from the original program author. By comparing the best code for each hardware environment we may conclude that there is an approximate 20-fold average performance enhancement. Most significantly, the host CPU time required is reduced to only one minute. This time includes all data transfer to and from the array processor as well as an amount of input from and output to disk in the host itself.

The program set (host program and array processor program) is designed so that no alteration of the host program is required to use a different space-group routine; the appropriate array processor routine is selected by means of a VAX/VMS command language statement which causes the host program to load the array processor with the desired version. The array processor programs have been constructed to permit the simultaneous consideration of coordinates, isotropic temperature factors and occupancy factors for as many as 10000 atoms without recompilation or relinking.

| **Time required to calculate structure factors and derivatives for one cycle of PROLSQ refinement of Rhizopus pepsin, P, 212121, 2791 atoms, 31915 reflections** |
|---------------------------------|-----------------|-----------------|
| VAX, space-group-general        | 1055            | 1055            |
| VAX, P, 212121                 | 453             | 453             |
| MARS-432, space-group-general  | 1               | 39              |
| MARS-432, P, 212121            | 1               | 25              |

*All times are expressed in min.
†Dependent upon other activity in the VAX.
The modified space-group-specific version of PROLSQ is currently being applied to the high-resolution refinement of the native form of the enzyme cytosolic aspartate aminotransferase, AAT \([P2_1_2_1_2_1, a = 124.8, b = 130.8, c = 55.8\ \AA]\) (A. Arnone & C. Hyde, personal communication). The structure consisting of 6578 atoms, with individual temperature factors, is being refined against 91 146 structure-factor amplitudes extending to 1.6 \(\AA\) resolution. Preliminary refinement cycles carried out with the version of PROLSQ modified for \(P2_1_2_1_2_1\) by P. Briley (see Hendrickson, 1985) required more than 48 h of VAX 11/780 CPU time to complete and often twice that amount of clock time owing to other use of the VAX. In contrast, duplicate refinement cycles using our array processor version of PROLSQ were complete in just under three hours, which included 156 minutes of array processor time for the summation during which only 16 min of host CPU time were used, and only 30 min of host CPU time for the complete job.

The ratio of run speeds for a VAX 11/780 and an IBM 3081 was observed by L. Sjolin (personal communication) to be 1:11 while making some alterations to the same PROLSQ program with which we have been working here. J. L. Sussman (personal communication) observed a similar ratio when comparing the speed of a VAX 11/780 and an IBM 3081 with the program CORELS (Sussman, 1985) as a benchmark. If we may be permitted to extrapolate from these observations, the current results suggest that our hardware configuration running optimized code can perform faster than the IBM 3081 by a factor of two.

Our source codes are available for use by the general crystallographic community. The latest versions, including currently available space-group-specific routines, will be sent to any requestor upon the receipt of a magnetic tape.

The author expresses sincere appreciation for the critical reading of the manuscript by Drs S. Sheriff and C. Hyde. He also appreciates the encouragement of Dr D. R. Davies.

References


**XFPS, a program for automatic Fourier, Patterson and superposition methods.** By František Pavelčík, Department of Analytical Chemistry, Faculty of Pharmacy, Komenský University, 832 32 Bratislava, Czechoslovakia

(Received 19 December 1985; accepted 14 August 1986)

**Abstract**

A program is presented that solves crystal structures completely using Patterson function, symmetry minimum function, atomic minimum superposition and advanced Fourier methods. The program works most efficiently for structures with one or several heavier atoms in the asymmetric unit.

**Introduction**

The superposition methods (Beevers & Robertson, 1950; McLachlan, 1951; Buerger, 1959) were regarded in the 1950's and 1960's as the most powerful tool for the solution of the phase problem (Lipson & Cochran, 1966), but later the appearance of sophisticated direct-methods programs caused considerable loss of interest in these methods. Nevertheless, superposition methods can still play an important role in structure determination (Simonov, 1982), particularly in a case with a small number of heavier atoms in the asymmetric unit.

The symmetry minimum function (Simpson, Dobrott & Lipscomb, 1965, henceforth SDL) offers a very convenient method for the determination of atomic positions from the Patterson function, particularly in the heavy-atom case, but computer applications are rather rare (Hubbard, Babich & Jacobson, 1977), probably because it requires storage of the Patterson function. Nowadays, existing mainframe computers permit fast access to sufficient memory, so there is no reason not to use this effective method.

**XFPS** was designed to fill the gap in crystallographic software as well as making a contribution to the automation of structure determination. In any case it offers a powerful alternative to direct methods.

**Procedure**

Our purpose was to develop a procedure that needs only marginal information to solve the phase problem and to obtain the complete structure automatically in a single computer job. To achieve this we used Patterson and superposition functions to solve the phase problem with trial...