

# Busting out of crystallography's Sisyphean prison: from pencil and paper to structure solving at the press of a button: past, present and future of crystallographic software development, maintenance and distribution

Lachlan Michael David Cranswick

National Research Council of Canada, Chalk River Laboratories, Chalk River, Ontario, K0J 1P0, Canada. Correspondence e-mail: lachlan.cranswick@nrc.gc.ca

The history of crystallographic computing and use of crystallographic software is one which traces the escape from the drudgery of manual human calculations to a world where the user delegates most of the travail to electronic computers. In practice, this involves practising crystallographers communicating their thoughts to the crystallographic program authors, in the hope that new procedures will be implemented within their software. Against this background, the development of small-molecule single-crystal and powder diffraction software is traced. Starting with the analogue machines and the use of Hollerith tabulators of the late 1930's, it is shown that computing developments have been science led, with new technologies being harnessed to solve pressing crystallographic problems. The development of software is also traced, with a final caution that few of the computations now performed daily are really understood by the program users. Unless a sufficient body of people continues to dismantle and re-build programs, the knowledge encoded in the old programs will become as inaccessible as the knowledge of how to build the Great Pyramid at Giza.

## 1. Introduction

The following concentrates on the early history of crystallographic computing as the first attempts in anything, by going from nowhere to somewhere, are the most difficult, most important, and often set precedents and create traditions for future development. Where possible, these pioneers speak their own words, as many of their overall situations and strategies still maintain a modern resonance. However, the crystallographic wisdom behind much software, program philosophies and source code can be obscured to modern view. Significant history can be distributed sparsely in the literature so as to resemble a shattered crystal that can be reconstructed in many and various ways. Thus what follows should be considered as a partial brass rubbing of a much larger richer history where there is still much room for elaboration and exploration.

### 1.1. Trends in crystallography

In the proceedings of the 1960 Glasgow crystallographic computing conference, a paper on 'Small-Scale Computers in X-ray Crystallography' by Niggli displayed a triangle diagram detailing Availability *versus* Convenience *versus* Reliability, where for crystallographic calculations the 'slide rule' was given high marks for availability, but low marks for reliability

and convenience (Niggli, 1961). The technology and emphasis of crystallographic computing has changed greatly, with an overall trend of the practising crystallographer delegating more and more to the electronic computer. Before the 1940's, the crystallographer was required to be both the data-processing drudge and thoughtful scientist to solve a crystal structure to completion. From the 1940's to the 1980's, the trend was to delegate the crystallographic drudge work to the electronic computer, but the scientific thought and brainpower generally remained with the individual crystallographer. From the 1980's to the present, the trend has been to delegate not only the drudge work but also the scientific thought to the computer from crystal screening through to structure validation. Delegation of scientific thought to the computer is in practice the delegation of thought to the crystallographers who wrote the analysis software and whose expertise is incorporated within the running programs.

## 2. Escaping the Sisyphean prison: from the late 1930's to the personal computers of the 1990's

Crystal-structure solving and refinement before routine access to electronic computers could be described as a severe Sisyphean challenge, taking a determined effort involving drudgery over months and maybe years. On completion, the

Sisyphian task would repeat with the next crystal structure problem. The history of crystallographic computing can be explained metaphorically as continual attempts to escape incarceration from different Sisyphian prisons, starting from the soul-sapping maximum-security-confinement of the pencil, paper and adding machine. Developer of the X-RAC crystallographic analogue computer (Pepinsky, 1947), Ray Pepinsky (see Fig. 1), stated that he was not in the mood for a lifetime of traditional crystallographic prison cuisine; that after World War II: *'There were enough fascinating structural problems, Siren-like, to lure one back to a life-time of X-ray work; but the vision of life at a hand calculator was like the wax Ulysses prescribed for the ears of his sailors.'* (Pepinsky, 1952e.) As elaborated below, Pepinsky was predated in these general sentiments by British crystallographers of the late 1930's (Beevers, 1939) and Linus Pauling at the California Institute of Technology (Caltech) in the USA (Hughes, 1952a). The crystallographers represented within the British paper entitled 'A Machine for the Rapid Summation of Fourier Series' (Beevers, 1939) were quite articulate that they were not intending to escape the Sisyphian dungeon only for themselves but were also aiming to spring everyone else out as well.

### 2.1. Late 1930's: plans for design and use of custom analogue computers

The intended method by which computational crystallographers of the late 1930's were planning to escape the drudgery of manual data-analysis methods was *via* building their own affordable custom crystallographic analogue computing apparatus. Much effort was based around transferring the manual pencil-and-paper-based methodology of Beevers–Lipson Strips (Lipson & Beevers, 1936) to analogue computing apparatus. By the technological and financial conditions of the time, this was a progressive strategy to consider. Analogue computers have a long history, where

clever (and usually mechanical) designs allow tedious calculations to be performed easily. There are many examples of recent and ancient analogue computers (see Bromley, 1990). One is the ingenious ancient Chinese 'South-Seeking Chariot', reported as designed and built in the third century AD by Ma Chun; it being a non-magnetic compass vehicle, and the oldest known self-regulating mechanism (Santander, 1992). Another is the shoebox-sized *Antikythera* mechanism of ancient Greece (Freeth *et al.*, 2006), *ca* second century BC, the world's oldest surviving astronomy computer. The designs of effective analogue computers are generally associated with much elegance, brilliance, ingenuity and economy on the part of the inventors.

Unlike computer software and algorithms, information for construction of analogue computers can be more difficult to disseminate. There is much confusion on the history of working versions of the South-Seeking Chariot and how many times it was reinvented due to previous descriptions being too vague and subject to interpretation (Santander, 1992). The unique *Antikythera* mechanism was found in a shipwreck and it is postulated the effects of the Roman conquest of ancient Greece extinguished this computing technology (Freeth *et al.*, 2006). The extremely powerful X-RAC analogue crystallographic computer of Ray Pepinsky, built in the late 1940's, seems to have been similarly affected, *'The plans were offered to European crystallographers as early as 1951, in case the decision was reached to duplicate the machine in some center or centers; and funds could have been made available from America to support the construction in Europe. I have always felt that more wisdom should have been exerted, in selecting an experienced electronics man with crystallographic interests, to advise British crystallographers of the possibilities inherent in the basic X-RAC design. How many man-years of dull computations and contour-map delineation would probably have been avoided, if bad advice had been avoided!'* (Pepinsky *et al.*, 1961b.)

### 2.2. The influence of Leslie John Comrie (1893–1950) and Wallace J. Eckert (1902–1971)

Although British and American crystallographers of the late 1930's were proposing their intent for custom crystallographic analogue computing solutions, two of the main pioneers of scientific computation using general-use electronic computers (Grosch, 2003) had an influential effect on the early direction of crystallographic computing. These were the UK-based Leslie John Comrie (1893–1950) (da Cruz, 2006a) and the American, Wallace J. Eckert (1902–1971) of Columbia University, New York (da Cruz, 2006b) (see Fig. 2).

As a consequence of being seriously wounded and losing his left leg in France during WWI (Massey, 1953; McLintock, 1966), the New Zealand born Leslie John Comrie changed his interest from chemistry to the computational side of astronomy and in the process became a strong proponent for the use of commercial calculation equipment for scientific computation (Massey, 1953). He was employed as Deputy Director of the British Association for the Advancement of



**Figure 1**

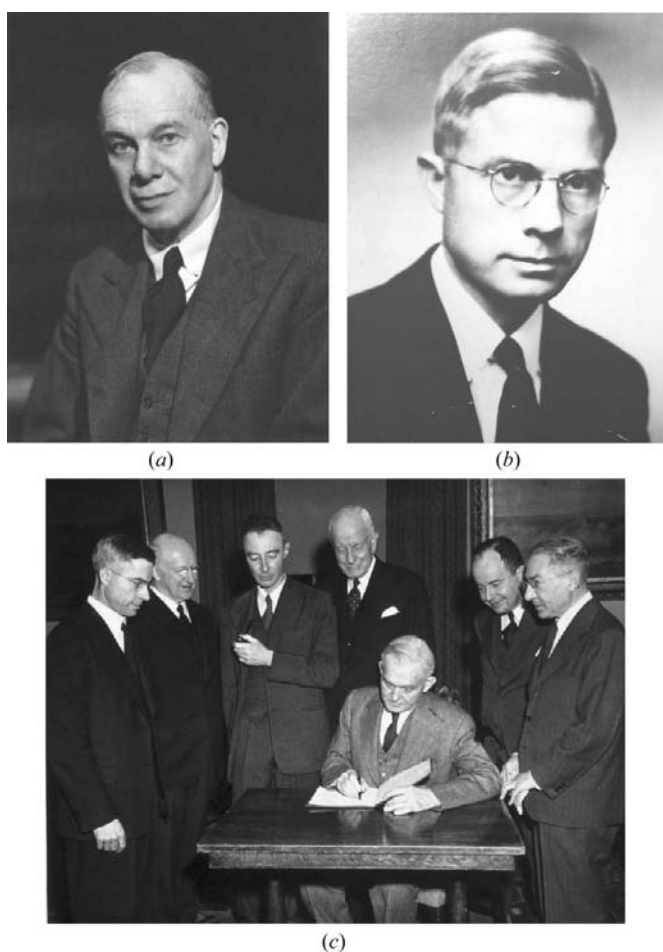
Ray Pepinsky explaining the X-RAC analogue computer system for Fourier syntheses and display of contour maps. Left to right: Fankucken, Robertson, Low, Hughes, Friedman, Wrinch, Pepinsky. Inset left: assembly of S-FAC units at X-RAC oscilloscopes; inset right: X-RAC contour map of phthalocyanine [taken from Fig. 105, Fig. 116 and Fig. 72 of the Pepinsky (1952e) computing monograph; courtesy of Penn State University.]

Science's Nautical Almanac Office (NAO) in 1925; followed by appointment as its Superintendent in 1931 (Massey, 1953; Croarken, 1999). Comrie was the first person to apply a Hollerith general-use computer for scientific computation in 1928 (Comrie, 1932). This was *'the summation of harmonic terms, or, in other words, Fourier synthesis. In this way the principal terms in the motion of the Moon from 1935 to 2000 were computed.'* (Comrie, 1946.) That 'good enough' is often the natural enemy of 'better' could perhaps describe the style of resistance Comrie encountered to his introduction of new computing methods within the NAO. These difficulties *'were accentuated by the fact that the absence of rapid change in the work of the [Nautical Almanac] Office had become almost traditional, the Nautical Almanac having largely retained the*

*general form introduced in 1834.'* (Massey, 1953.) Scientific Computing Services Ltd was formed by Comrie in 1936 after his dismissal from the Nautical Almanac Office by the British Admiralty (Croarken, 1999). The primary documentation of the civil service investigation leading to his dismissal, *'provide both fascinating and melancholy reading. Comrie's often-expressed complaints that civil service regulations were petty and restrictive are given some justification by these documents.'* (Croarken, 1999.) *'For example, Comrie felt the embargo on the employment of married women was unnecessarily wasteful in terms of invested training.'* (Croarken, 1999.) However, *'it is clear that Comrie was inept at "playing the game". Unfortunately, Comrie played his own game, which was why he was sacked.'* (Croarken, 1999.) The commercial computing service that resulted as a consequence of Comrie's dismissal had significant influence on the development of crystallographic computing.

Comrie's American equivalent, Wallace J. Eckert, was a Professor of Astronomy at New York's Columbia University from 1926 to 1970 and was influenced by Comrie's successful use of commercial tabulators for scientific computation (Comrie, 1946). Eckert was closely associated with, and supported by, the IBM Corporation and its President, Thomas J. Watson (da Cruz, 2006a). From 1945 to 1966, Eckert was also IBM Director of Pure Science, as well as being Director of the Watson Scientific Computing Laboratory at Columbia University (da Cruz, 2006a). Eckert *'was one of the first to apply punched-card machines to the solution of complex scientific problems. Perhaps more significantly, he was the first to automate the process when, in 1933–34, he interconnected various IBM calculators and tabulators with control circuits and devices of his design to solve differential equations.'* (da Cruz, 2006a.) *'Eckert – first, foremost, and always an astronomer – claimed little credit for his innovations in computing which, to him, were only detours required to get his real work done'* (da Cruz, 2004). *'Eckert directed the construction of a number of innovative computers for performing astronomical calculations, including the [IBM] Selective Sequence Electronic Calculator [SSEC, 1949] and the [IBM] Naval Ordnance Research Calculator [NORC, 1954],'* (see Fig. 2c) *"which for many years was the most powerful computer in the world. The accuracy of Eckert's calculations of the Moon's orbit was so good that in 1965 he was able to correctly show that there was a concentration of mass near the lunar surface."* (da Cruz, 2006a). The influence of Comrie and Eckert was to guide crystallographers away from custom analogue devices to that of general-use commercial computers in the form of Hollerith tabulators.

**2.2.1. 1939: Arnold Beevers and British crystallographers versus Leslie John Comrie.** As stated by Cruickshank (1949), *'it was not until about thirteen years after W. H. Bragg, in 1915, had shown that the electron density in a crystal could be represented by a Fourier series of structure factors that the experimental technique was sufficiently advanced to permit of the application of the method, except in simple one-dimensional cases.'* With leading crystallographers in the late 1930's eager to break away from human computing methods, the



**Figure 2**

The two main pioneers of scientific computation using general-use electronic computers: (a) Leslie John Comrie (1893–1950) (courtesy of The Royal Society and ©The Godfrey Argent Studio); (b) Wallace J. Eckert (courtesy of the Astronomical Applications Department of the United States Naval Observatory); and (c) a photograph containing Wallace J. Eckert (1902–1971) courtesy of Herbert R. J. Grosch: *"It's the dedication of the NORC [IBM Naval Ordnance Research Calculator] (1954, at 612 W. 115 Street [Columbia University, New York City]). Sent me because I was not able to attend. Characters are [left to right] Eckert, Fackenthal, Oppie [J. Robert Oppenheimer], [IBM's Thomas John] Watson, Pegram (signing guest book). Johnnie von [Neumann], [Isidor Isaac] Rabi. The event is the dedication ceremony on 115th, but the signing is somewhere in the Men's Faculty Club a few blocks away."*

application of general-use computers to crystallographic computing could most likely be said to have been initiated in 1939 *via* the failed attempt of Leslie John Comrie to introduce the potential of Hollerith tabulators to British crystallographers (Beevers, 1939). The IBM Hollerith tabulator, originally invented in 1890 by the American born Herman Hollerith for government census and business applications, had its first above-mentioned scientific application on Fourier synthesis performed by Comrie in 1928 (Comrie, 1932). Comrie stated that, due to economic, and other, reasons, there was more of a disposition to use electronic computation machines in the US than Europe (Comrie, 1946). Later, as the use of IBM-style Hollerith tabulators were becoming accepted into mainstream crystallography (see *Acta Cryst.* Volume 2, Part 6, December 1949), Cox & Jeffrey (1949) also allude to the requirement of commercial British tabulators to handle the non-metric local currency of pounds, shillings and pence as affecting their applicability for advanced scientific computation. Owing to the expense of owning or leasing the Hollerith, use would require centralized or commercial computing centres of the time; such as provided by Comrie's London-based Scientific Computing Services Ltd.

The initial response by the leading British crystallographers (see Fig. 3) was to continue advocating the development of affordable specialized home-made laboratory-based analogue crystallographic computational apparatus (Beevers, 1939). This is not to suggest the British crystallographers did not understand the computation implications of the Hollerith. As articulated by Beevers (1939), it was more their abhorrence of the financial cost and that the calculations would be taken out of their hands: *'Now the selection and addition of strips of numbers is precisely the process which is accomplished in*



**Figure 3**

Photograph from the 1950 conference on computing methods and the phase problem in X-ray analysis at Pennsylvania State College, which includes some of the crystallographers mentioned within the 1939 British paper of Beevers. Left to right: J. M. Bijvoet, J. F. Schouten, H. Lipson, E. G. Cox, C. H. MacGillavry, M. F. Perutz, C. W. Bunn, C. A. Beevers, R. Pepinsky, J. M. Robertson and E. Grison with 'Tiddles'. (Plate 13 from the American Crystallographic Association monograph of McLachlan & Glusker, 1983.)

*systems such as the Hollerith series of calculating machines . . . The writer had an opportunity (provided by Dr L. J. Comrie, of Scientific Computing Service Ltd.) of a thorough investigation of the application of the Hollerith system to Fourier summations. Unfortunately the cost of the adding machine provides a serious obstacle. A central bureau, to which the data would be sent for computation, would be required . . . The simplest synthesis would require several days for the results to come to hand, and it is a great disadvantage for the computation to be outside the direct control of the investigator.'* The emphasis of the British crystallographers was to get the job done conveniently and affordably within their own laboratories by whatever means current, or which they thought could be developed in-house. This is not an isolated crystallographic attitude of the time, or of other times. Of the ambitious American X-RAC crystallographic computing machine of the late 1940's, Pepinsky states that it *'was built entirely by students at Auburn [Alabama, USA]. Even the power and output transformers were wound by these youngsters'* (Pepinsky, 1952*f*). Back to the 1939 British paper, Beevers states that *'It is therefore of importance to develop the fastest possible method of making the one-dimensional summation, and to render this method available to all crystal-structure laboratories.'* I. Fankuchen expressed a similar comment that it is *'desirable that any devices used for the rapid summation of such series should be of a nature which will make possible their use in individual laboratories.'* For example, Cox included that *'I have now under construction a machine designed primarily for the calculation of structure factors.'*

To this, Comrie replied, *"I have always advocated, as a policy, that the capabilities of existing commercial machines should be fully exploited before the laborious and costly task of designing and making special apparatus is embarked upon. In this case the nature of the problem points to the use of punch-card machines, which have been successfully applied to Fourier synthesis in the summation of the harmonic terms in the moon's motion".* Comrie was *"not convinced by the [crystallographers'] curt dismissal of these [Hollerith] machines. First, the cost of using them is not likely to exceed the cost of designing and constructing the machine described [by Beevers], if due allowance be made for the time of the designer, and all overhead costs. Secondly, the centralizing of the computations is not necessarily an ultimate disadvantage. It means that one machine only is required, and that can be maintained and run with a higher efficiency factor than could be secured with scattered installations working part time only. The investigator's job is to plan and execute his observations, and to interpret the results; he should not be called on to do an amount of arithmetic comparable with his observations, because manipulative skill and numerical skill are not necessarily associated."* and *'In conclusion, I am surprised that crystallographers do not prefer an existing machine now to waiting several years for a machine that is still in the early stages of design.'* (Beevers, 1939.)

While there is a footnote from Beevers (1939) seeing the need for both styles of lab-based and centralized computing machinery, the first home-built Beevers analogue machine for the 'rapid summation of Fourier series' was described in 1942

(Macewan & Beevers, 1942), and could be wired up in a few weeks by a worker. Bunn (1946*b*) describes that “*Evans and Peiser (1942) [at the University of Cambridge, UK] have a machine (made largely of ‘Meccano’ parts) which evaluates  $f \cos$  (or  $\sin$ )  $2\pi(hx + ky)$* ” for the computation of structure factors. The 1942 paper records time savings of 49 min using the analogue machine compared to 240 min using manual numerical methods. The Meccano-based structure-factor computer is also described as being less fatiguing and less error prone to use than manual methods (Evans & Peiser, 1942). While restricted access to construction materials within wartime Britain was an incentive to use Meccano, Evans & Peiser (1942) noted the machine was ‘*robustly constructed, and it has been found possible to realise a surprisingly high degree of accuracy, adequate for the normal requirements of crystal-structure analysis.*’ The paper thankfully acknowledges Meccano Ltd, and especially ‘*Mr. F. Riley BSc, who made available to us the parts used in the machine.*’

**2.2.2. 1941: under the influence of Columbia University’s Wallace J. Eckert, the first published use of Hollerith tabulators by American crystallographers at Caltech.** The experiences of Comrie are in contrast to that of his American

equivalent, Wallace J. Eckert. Eckert encouraged the use of IBM Hollerith computers for crystallographic applications *via* the laboratory of Linus Pauling at the California Institute of Technology (Lu *et al.*, 1941; Shaffer *et al.*, 1946; Eckert, 1947; Donohue & Schomaker, 1949; Hughes, 1952*a*). This influence was *via* Eckert being a reviewer on a funding proposal to build a mechanical computer for summing Fourier series as submitted by Linus Pauling to a Committee on Scientific Aids to Learning (Hughes, 1952*a*; Shaffer *et al.*, 1946). Eckert instead recommended that existing IBM Hollerith computers would provide an effective computing solution (Hughes, 1952*a*). The funding body stated they would provide money to use the IBM equipment if Pauling agreed (Hughes, 1952*a*). The acceptance by Pauling resulted in the first crystallographic paper to appear in print using IBM Hollerith computers, on the structure of melamine as published in 1941 by Edward W. Hughes (1941). Where Beevers–Lipson strips required several days of calculation, the early 1940’s Hollerith took 5 to 7 h, and gave higher density of calculated points with assured accuracy (Shaffer *et al.*, 1946). In a letter to Roy Stephens of IBM dated 23 January 1940, Linus Pauling thanked IBM for providing the computers free of rental charge, and concluded that ‘*I wish to extend my sincere thanks to you for assisting our researches in this way, and to congratulate the International Business Machines Corporation on its interest in the problems of pure science.*’ (Pauling, 1940.)

The delay in detailed publication of the Caltech Hollerith method until the Shaffer *et al.* paper of 1946, and presentations by Hughes, which occurred at the 1946 ASXRED X-ray and electron diffraction conference (Hughes, 1952*b*; Armstrong, 1946), might have been due to wartime work requirements during World War II. Hughes (1952*a*) mentions the Caltech-based Hollerith equipment was performing calculations to assist the aircraft industries of Southern California during the war. A summary of an 8 November 1943 letter received by Pauling at Caltech from William Astbury (1898–1961), based at the University of Leeds, UK, notes his request to know if the Hollerith method has been published, and Astbury ‘*states that he would like a copy, and if necessary he can put through an official request through his government as they are also doing war-related work.*’ (Pauling, 1943.) The entry in the Pauling Archive for 12 February 1945 notes: ‘*Letter from William Astbury to LP [Linus Pauling] RE: Once again asks about the use of Hollerith machines, requesting a sample pack of 11 cards from the Cal Tech machine to see if they will be compatible with those at his university in England. Also asks for information on how they deal with the method of least squares as used by Hughes.*’ (Pauling, 1945.) After publication of the 1946 Shaffer *et al.* paper, reprint request cards extant at Caltech include those from the USA, one from Canada and one from The Netherlands (Henling, 2007). A letter requesting updated information on the Hollerith method, written by Martin Buerger of MIT to Verner Schomaker at Caltech and dated 27 November 1946, also states ‘*As one who is not familiar with the I B M machines, the complexity of the whole scheme is appalling.*’ (Henling, 2007.) (See Fig. 4.)



**Figure 4**

A 1947 or prior photograph (cropped) of IBM (Hollerith) operators at the Caltech Computing Center machine room performing crystallographic computation using the IBM card punch systems as described in Shaffer *et al.* (1946) and briefly summarized in Swift (1947). Photograph courtesy of Larry Henling and Dick Marsh, Caltech, USA: “*The structure model on the table appears to be a Leybold phenakite (label missing). The drawing on the wall is probably an incorrect model for serine. The machines are most likely IBM numeric duplicating key punch, 10043, Type 012 (which replaced the IBM electric key punch Type 011 used for the Shaffer paper); IBM horizontal sorter, 55645, Type 080 Model 1; and IBM electric accounting machine (or alphabetic tabulator) with direct subtraction and 88 type bars fitted with 7 extra ‘X-distributors’ for a total of twelve, 10978, Type 405.*” The key punch operator at the bottom left is long-time Caltech employee, Lillian Casler. The names of the other two operators could not be identified.

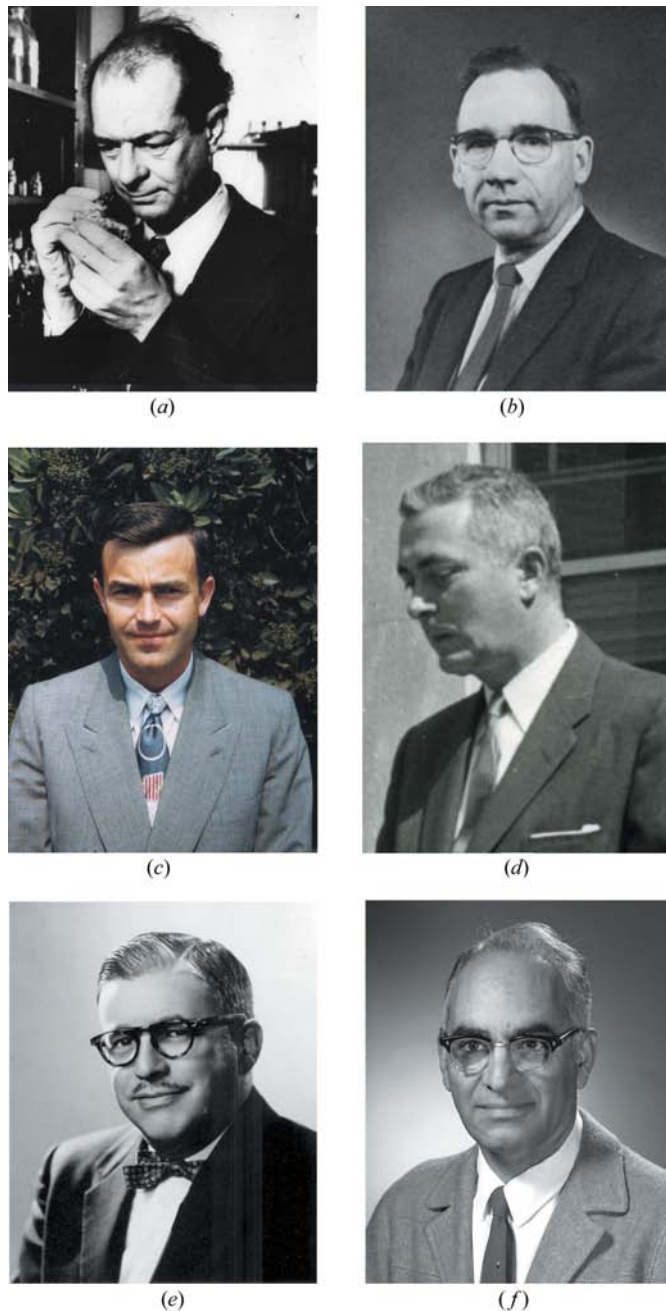


The 1941 paper by Hughes had another first for its time: Hughes mentions that the optimization of crystallographic parameters seemed ‘to call for the use of the method of least squares’ and in consequence the paper included use of a ‘new method for refining [crystallographic] parameters, based upon least squares’ (Hughes, 1941). Hughes (1952b) states that it was actually a 1940 structure on dicyandiamide (Hughes, 1940) which first used least squares, but that he was not confident enough in the method to admit it, and stated in the 1940 paper that parameters were refined by systematized trial and error. Hollerith-based least-squares calculations took a few hours where manual methods required one to two days work with an adding machine (Shaffer *et al.*, 1946). It took time for least squares to become known and appreciated by the crystallographic community. The 1946 edition of the *Chemical Crystallography* monograph of Bunn (1946a) states only as a footnote that a “numerical ‘least squares’ method of adjusting parameters is described by Hughes (1941)”. Fig. 5 shows photographs of some of the scientists who were at Caltech in the 1940’s, and who either developed or used these Hollerith-based methods.

**2.2.3. 1945: British use of Hollerith tabulator for the crystal structure of penicillin.** In mid 1940’s wartime Britain, and ‘owing to the extreme importance of penicillin as a military weapon’ (Clarke *et al.*, 1949), Dorothy Crowfoot-Hodgkin’s crystallographic study of penicillin made use of Hollerith-based punch-card computing methods for Fourier synthesis (Crowfoot *et al.*, 1949a). While acknowledging the prior American work with Holleriths, the method is described as being similar in principle but different in detail from the Caltech group in America. With funding from the Medical Research Council, this method was developed by Comrie’s Scientific Computing Services Ltd in London and used a Pierce Alpha calculating machine on loan from H. M. Treasury and Stationery Office (Crowfoot *et al.*, 1949b). A biography of Dorothy Hodgkin states the computer used was in Cirencester, being “an American Hollerith machine used principally for tracking ship’s cargoes” (Ferry, 1998b) and convoy planning (Dodson, 2002). ‘The penicillin program was run at night, when the machine would otherwise have been idle’ (Ferry, 1998b) (see Fig. 6). Detailed public information on the penicillin results were delayed because ‘all of the information secured during the period of active collaboration bore a high security classification. By January 1, 1946, all participants had been released from their obligations to hold the information in secrecy, but they agreed to refrain from any independent publication that might, in the opinion of the Editorial Board, interfere with the plans for the monograph.’ (Clarke *et al.*, 1949.)

Within the Summarized Proceedings of Conference on X-ray Analysis – London, 1945 (Parker *et al.*, 1945), it states “Mr G. B. Hey presented a paper entitled ‘The Calculations Involved in the X-ray Analysis of Crystals’, by Dr L. J. Comrie and himself. He said that he approached the problem as a mathematician, not as a crystallographer, and that he came from an organization which could deal with calculations when crystallographers found them too tedious. “ Dr Comrie and he

hoped to publish in a few months’ time a paper on the best methods of performing Fourier summations.” ‘In the discussion



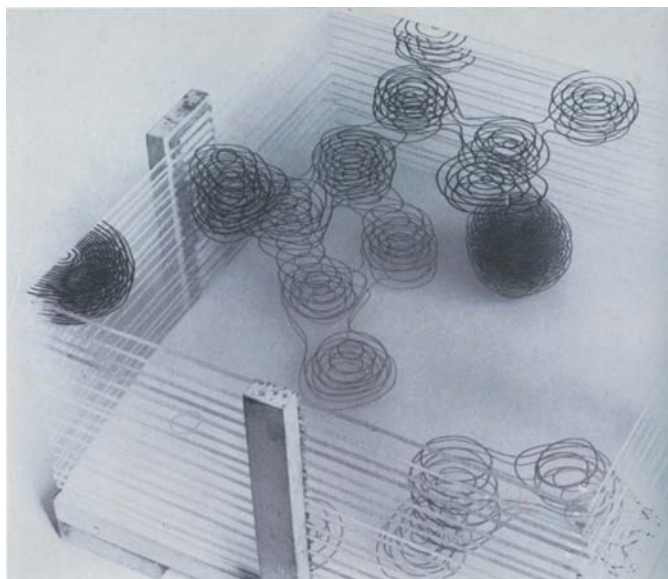
**Figure 5**

Some of the scientists at Caltech in the early 1940’s involved in the development and first applications of IBM Hollerith computers to crystallography: (a) Linus Pauling (1901–1994); (b) Edward Wesley Hughes (1904–1987); (c) Verner Schomaker (1914–1997); (d) Philip Anderson Shaffer Jr (1916–1974); (e) Paul-Antoine Giguère (1910–1987). (f) Henri A. Levy (1913–2003) [Figure (a) courtesy of the Archives, California Institute of Technology, (b) from the American Crystallographic Association monograph of McLachlan & Glusker, 1983; (c) courtesy of Dick Marsh, Caltech, USA; (d) courtesy of Becker Medical Library, Washington University School of Medicine, St Louis, USA; (e) courtesy of Département de Chimie, Université Laval, Québec, Canada; (f) courtesy of Oak Ridge National Laboratory.] Two other Caltech crystallographers of the time but not pictured here were Lu Jiaxi (Chia-Si Lu) (1915–2001) and Robert Brainard Corey (1897–1971).

Dr Dorothy Crowfoot (Mrs Hodgkin) said that nowadays the X-ray analysis of a complex crystal structure fell into two stages: the determination of its essential structure, and the refinement of parameters. The first was the job of the crystallographer; the second could be left to professional computers.” (Parker *et al.*, 1945.) The *Summarized Proceedings of Conference on X-ray Analysis – London 1947* (Kellar & Douglas, 1948) describes talks by Hodgkin and Bunn on their methods for determining the structure of penicillin, but no mention is made of Hollerith computers or Comrie. Hodgson *et al.* (1949) cite the Comrie crystallographic computing method as being unpublished, and Comrie, impaired by a stroke after 1948, and with deteriorating health, died in 1950 (Massey, 1953). *Acta Cryst.* describes further interaction of Comrie with crystallographers such as Abrahams *et al.* (1949), and their work on naphthalene (see Fig. 7).

### 2.3. Late 1940's to early 1960's

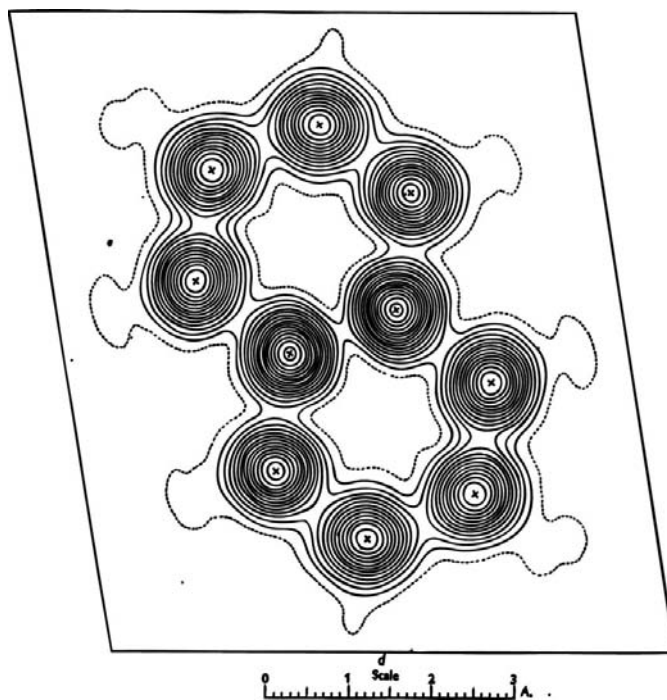
The lead developer of Fortran, IBM's John Backus, recollected that programming in the early 1950's 'was really fun. Much of its pleasure resulted from the absurd difficulties that "automatic calculators" created for their would-be users and the challenge this presented. The programmer had to be a resourceful inventor to adapt his program to the idiosyncrasies of the computer.' (Backus, 1980a) and that programming was



**Figure 6**

From the 1949 penicillin monograph: photograph of electron-density contours drawn on sheets of perspex showing the thiazolidine and  $\beta$ -lactam of potassium benzylpenicillin (Crowfoot *et al.*, 1949a). Owing to urgency created by the 'extreme importance of penicillin as a military weapon' in WWII, determining the crystal structure of penicillin involved the first crystallographic use in Britain of a Hollerith computer; with Dorothy Hodgkin-Crowfoot as lead crystallographer and with Leslie J. Comrie's Scientific Computing Services Ltd developing the computing methodology. As noted in the penicillin monograph: 'Under the terms of the contract for the publication of the *Chemistry of Penicillin* the publisher has agreed to waive its rights under the copyright after five years from the date of publication. Thereafter this volume will be in the public domain and dedicated to the public.'

also 'a black art, a private arcane matter involving only a programmer, a problem [and] a computer.' (Backus, 1980b). Early generations of general-use computers were also expensive and not readily accessible, which helps explain why crystallographers were slow to the full use of them. Rollett, in his article on the DEUCE computer mentions 'it was not uncommon to hear of large three-dimensional calculations which had been waiting for periods of up to two years for lack of suitable computing equipment', and, due to costs of computer usage at that time, only an economically justified subset of crystallographic software programs was written (Rollett, 1961). Obtaining access to computers could be a battle, though the following imagery of the early 1960's is not of the metaphorical wielding of swords: 'Crystallographers are not the elect among contemporary scientists. Unless they are associated with large industrial or federally supported laboratories in America, they must beg time on large-scale machines where they can find it' (Pepinsky *et al.*, 1961a). Stanley Nyburg describes the late 1940's where the Cox group in Leeds, UK, used by night a Hollerith at the West Riding Electricity Board in Yorkshire (Nyburg, 2004); made feasible by application of the more user friendly Hollerith-based methods developed within the Leeds crystallography group (Cox *et al.*, 1947a,b). Dick van der Helm describes using an IBM 604 computer at night in the mid 1950's, the IBM was the main computer of an Amsterdam bank (van der Helm, 2004); ZEBRA computers being installed at some Dutch universities in 1958 (Bijvoet, 1962). In a 1953 *Acta Cryst.* paper on the crystallographic computing potential of Manchester University Electronic



**Figure 7**

Electron-density contour map through the plane of the naphthalene molecule, as taken from Abrahams *et al.* (1949), where the numerical work was carried out by Comrie's Scientific Computing Service Ltd using Hollerith computers.



Digital Computer (Mark II)/MUEC, Ahmed & Cruickshank (1953) describe Hollerith punch-card crystallographic calculations taking 63 h, whereas the equivalent calculations using this latest generation computer and programs of 1953 took 73 min, a 50:1 improvement.

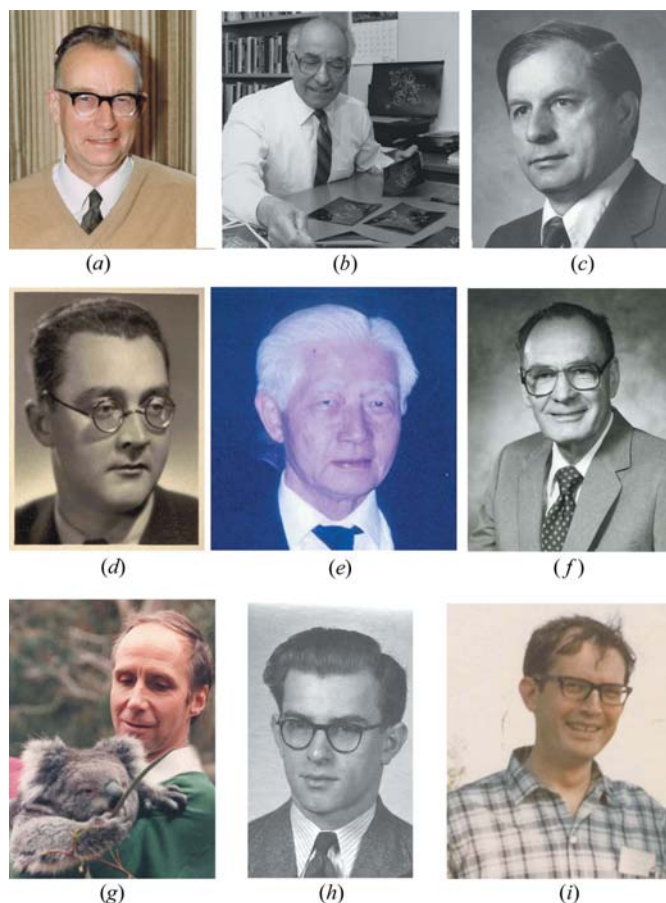
Despite such advances, owing to accessibility and expense of general-use computers, the period of the late 1940's to the 1960's still saw the side-by-side use of manual methods (Beevers–Lipson Strips; Robertson strips; Patterson and Tunell strips and stencils), specialized crystallographic analogue computers (X-RAC, RUFUS, ODFAC, Hagg and Laurent, Hoppe and Pannke, UTAH), and all-purpose general-use digital computers (MUEC, Bull Gamma, FERUT, ZEBRA, IBM-407, IBM-650, IBM-704, IBM-709, SEAC, SWAC, PEGASUS, MUSE/ATLAS, DEUCE, MERCURY, Whirlwind, EDSAC II). The summary of the 1946 Conference on X-ray Analysis in London (Institute of Physics, 1947) lists a variety of computational techniques, including ideas based on 'physical similitude', and where one described by Harker involved a plan to use a 3 cm radar transmitter to simulate an X-ray beam. A detailed description of many manual and analogue computing technologies of the time, with photographs and diagrams, is within chapter 8 of the crystallography monograph of McLachlan (1957). At the digital computer end of the computing spectrum, a 1955 *Acta Cryst.* article by Friedlander *et al.* (1955) details a least-squares program on an IBM 701 where each refinement took just over 10 min, and quoted the corresponding hourly rental cost for using this commercial facility as USD 300 per hour, and thus 'inexpensive'. Hughes (1952c) detailed methods at Caltech by which crystallographic computing costs could be kept at a minimum, while the institute could still cover the rental cost of the IBM equipment; as well as modifications such that '*there would be no hand calculations at all*'.

#### 2.4. Early 1960's: smaller, 'more affordable', general-use electronic computers

In the 1950 conference proceedings, Fred Ordway of the National Bureau of Standards in Washington, DC (which at the time had the SEAC computer), stated in his talk on high-speed digital computers, '*Crystallographers are naturally interested in the possibility of delegating their onerous computations to a Giant Brain*' and that digital computers '*widespread use for crystallographic problems should come, therefore, with the passage of time and the relaxation of military urgencies.*' (Ordway, 1952a.) The developing history of smaller-computer hardware for accessible laboratory-based crystallographic computing is one where 'affordable' has been approaching asymptotically as being a synonym for 'cheap'. As of the early 1960's, these more-affordable computers could be located conveniently near or in crystallography laboratories. Crystallographic programs required writing in machine language or symbolic language due to small core memory (Shiono, 1970; van der Helm, 2004). 'More affordable' being in the eye of the beholder and should not be considered a synonym of 'cheap' at this point: a basic IBM 1620 computer

of 1959 vintage would be the modern equivalent of an over one million US dollar capital purchase – around USD 1600 a month to rent or about USD 85000 to purchase outright (Spicer, 2005). What could make do for a single American crystallography laboratory, such as in A. Lindo Patterson's laboratory (van der Helm, 2004), might have to make do for an entire country elsewhere. In 1964, an IBM 1620 computer was the first general-purpose computer for scientific and engineering use in Iceland; and was only purchased courtesy of a surprise donation from the Icelandic Development Bank in celebration of the bank's 10th anniversary (Magnússon, 2003). Of United Kingdom based university and research council scientific computing resources, the 1965 'Flowers Report' records its concerns about the lack of parity (normalized to GDP) in computing power compared to the USA (Flowers *et al.*, 1966).

Much of the state of crystallographic computing history can be gleaned from back issues of *Acta Cryst.*, and from the



**Figure 8**

A small subset of computational crystallographers: (a) John S. Rollett; (b) Farid Ramadan Ahmed; (c) Durward William John Cruickshank; (d) Vladimir Vand; (e) Ryonosuke Shiono; (f) William R. Busing; (g) Howard Flack; (h) Sidney Cyril Abrahams and (i) Walter Clark Hamilton. [(a) courtesy of Michael J. Field, Oxford University Computing Laboratory, UK; (b) Photo Credit: Tom Devecseri, NRC Institute for Biological Sciences; (c) courtesy of the Cruickshank family; (d) courtesy of the Czech Astronomical Society (CAS); (e) courtesy of R. Shiono; (f) courtesy of Oak Ridge National Laboratory; (g) courtesy of H. Flack, (h), (i) courtesy of S. C. Abrahams.]



monographs produced from the crystallographic computing schools of the 1950's to 1990's. A short overview of the IUCr computing schools from 1969 to 1996 is provided by Stewart & Bourne (1996). Fig. 8 shows photographs of a very small subset of computational crystallographers. However, much crystallographic computing information is also in the form of 'myth' (personal experiences; unpublished information; distilled wisdom within the brains of life-long crystallographers but not in print!). Such information would benefit from being published *via* appropriate forums such as the IUCr journals, the IUCr Commission on Crystallographic Computing Newsletter or other publications. '*Records of what has been considered and what has been accomplished are essential to the proper development of this or any other field.*' (Pepinsky, 1952b.)

### 2.5. Late 1980's to mid 1990's: small-molecule and powder crystallography using personal computers

At a time where Apple IIe personal computers had 64 kB of memory, and with the 640 kB conventional memory limit of the Intel 8088 CPU and early Microsoft DOS operating system, it is unlikely that anyone in the early 1980's would have anticipated just how powerful single-user personal computers would become. Or that mainstream world-wide computer use would be driven not by requirements of high-end number crunching, true multitasking/multi-user capabilities, but by: (i) desires for WYSIWYG (what you see is what you get) menu-based word-processing packages, spreadsheets programs and drawing programs, and (ii) the want for the young (and young at heart) to play computer video games of ever increasing graphical and interactive complexity. While Jones (1977) provided an overview of micro, mini, midi and maxi computers in *Acta Cryst.*, and Stewart & Bourne (1996) described the first reference to 'micro-processors' during a 1978 crystallographic computing conference in The Netherlands, the routes by which various crystallography laboratories would proceed to use personal computers for computation could be many and various. The following is but one.

The history of digital computers is generally that of more power, faster speed, larger memory and more storage capacity. However, personal convenience can create a strong incentive to tolerate lesser computing power, where '*a small computer in a crystallographic laboratory is a very useful tool and it provides better over-all efficiency in many calculations, than a big computer elsewhere.*' (Shiono, 1970.) Comrie in 1939 articulated the benefits of central computing facilities (Beavers, 1939), but the theoretical benefits could be in conflict with the reality. Computer centres could be quirky, with restrictive and varying rules on disk quotas, compute job queuing and compute job priority. There is generally much love declared by 1970's and 1980's era crystallographers for the VAX VMS computer systems of the Digital Equipment Corporation (DEC). Though unfortunately in the late 1980's the yearly service contract of an old VMS system could be more expensive than the hardware cost for a brand new and more computationally powerful UNIX-based server. To

crystallographers enamoured by VMS, switching to a UNIX server could in hindsight be found a retrograde step. Compared to the features of VMS, this was often due to the cryptic and unforgiving UNIX command-line user interface, lack of a versioning file system, and the disliked 'vi' text editor. With maintenance costs, the UNIX servers might only have a 30 to 90 day equipment warranty; after which a standard, and often expensive, service contract would then apply. In contrast, PCs, at the fraction of the cost of a UNIX server, would routinely have a minimum one year equipment warranty going up to three years; and be upgradeable by owners using the equivalent of commonly available generic 'no-name' parts. This, on top of usual central computer facility irritations, could create a reservoir of sentiment receptive to personal computers, even if the PCs had significant limitations. In practice, crystallographic programs could run faster on a single-user PC than when run on a popular multi-user server. The common PC operating system, Microsoft DOS, was simple and did not require systems management training or experience to install or maintain.

The increasing use of minicomputers and personal computers would have initial consequences for crystallographic software. Of the programming systems of the early 1950's, Backus (1980c) alludes to their success (or lack of) which could '*depend on the number of machines they would run on*'. Huml (1985b) noted that the popularity of high-efficiency user-oriented minipackages, like *SHELX*, were '*growing with increasing quality of low cost minicomputers*' but that other crystallographic systems were restricted to working on main-frame style systems. As stated from the 1987 crystallographic computing meeting, '*since good optimizing FORTRAN-77 compilers are now available for MS-DOS/PCDOS based personal computers, SHELX can be readily adapted to micro-computers*' and also notes that pre-compiled PC versions of *SHELX* had already been distributed (Robinson & Sheldrick, 1988). Facility computer servers may still have been required until a suite of either suitable PC binaries of crystallographic programs could be obtained, or a PC Fortran compiler matched to the requirements of the crystallographic source code. On average, and depending on laboratory requirements, by the late 1980's to mid 1990's, it became possible to acquire enough of a toolkit of freeware and commercial crystallographic programs for PCs to be a going concern for small-molecule single-crystal or powder diffraction analysis. With respect to the strong desires of crystallographers declared in the Beavers paper of 1939; with personal computers, crystallographers and users of crystallography could now obtain an affordable, effective, in-lab (and at-home) crystallographic computing apparatus under their full control.

## 3. Crystallographic software development

### 3.1. Crystallography's software gene pool

Crystallography's historical gene pool of software and methods is extremely large, so attempting a linear world-wide factual history could be tedious, long and potentially

unenlightening. Even by the early 1960's, the number of available crystallographic software packages numbered in at least the hundreds. Shoemaker mentions a list being developed by the American Crystallographic Association of software programs, which already had over 200 programs (Shoemaker, 1961). This developed into the *IUCr World List of Crystallographic Computer Programs*: the first edition listed 577 programs (described in International Union of Crystallography, 1962b), the second edition listed 533 programs (described in International Union of Crystallography, 1965), and the third edition in 1973 (International Union of Crystallography, 1973). An overview from Huml (1985a) describes the state of crystallographic software in the mid 1980's.

The following attempts to average a mass of material to some summary observations and history. While the following interpretations could be highly simplified and flawed in various ways (the author was not yet born when many of these events occurred), it can be worthwhile for crystallographers to look into the past for analogous situations, even if the details were different. Similar to examining the crystal structure databases, many current and future computing situations may have had similarities to past occurrences. Operating systems, programming languages, programming paradigms have come, gone and evolved. A plethora of crystallographic software has come, gone and evolved. Issues can be very situational: a strategy that was successful for one might have failed for another. Thus generalizations should be viewed with caution.

### 3.2. Recurring themes in crystallographic software development and usage

Irrespective of the era in which crystallographic software was developed and its level of automation, there seem to be recurring themes at the user end of crystallographic computing methods and software. Using the Beevers paper of 1939 as a window into the aims of practising crystallographers, these include: development time for algorithms; implementation in software; its acceptance by other crystallographers; cost; convenience and availability. Seeking a balance of expense, expertise, time and results with respect to established *versus* new crystallographic computing methods might not be clear-cut, except in hindsight. Much of the development or acceptance of computing methods and software have related to the desire of crystallographers (i) to have control over their analysis and (ii) to have affordable hardware and software tools considered most effective for the desired analysis. Prior to personal computers, both these desires have often been in conflict with the ability and cost of the computing technology to achieve this.

### 3.3. The crystallographic computing community, the sharing of ideas and development of direct methods

The international spirit of the early crystallographic computing community can be seen *via* the conference proceedings of the first conference on crystallographic computing held at The Pennsylvania State College, USA, in April 1950 (Pepinsky, 1952a); mainly based around Ray

Pepinsky's X-RAC analogue crystallographic computing apparatus. With current and future crystallographic luminaries in attendance, Pepinsky described that solving the major computing problems would '*require many minds. Our aim here is to share what we know – to cross-pollinate our minds.*' (Pepinsky, 1952c.) The international travel for attending this first conference was apparently: '*not easy. I draw to your attention the report, incredible as it may seem as we view him now, that in braving the rigors of the Atlantic, Henry Lipson lost seven pounds, hard-won from the Austerity program. Among other matters, we must strive to return these to him.*' (Pepinsky, 1952c.) IUCr journals and the crystallographic computing conferences were important in sharing and discussing crystallographic computing developments, many of which are still in use today.

If there is a crystallographic problem that most readily reflects a requirement for many minds, it is the development of direct methods and related algorithms for the solving of crystal structures. The computing school monographs and IUCr journals display a massive breadth and depth of participation. The impact of this research was demonstrated by a 1985 Nobel Prize in Chemistry for Herbert Hauptman and Jerome Karle, due to automated direct methods software having made structure solution generally accessible to non-experts in the form of *MULTAN* (Main *et al.*, 1980), *SIMPEL* (Schenk, 1988), *SHELX-76* (Sheldrick, 1976) and *SIR* (Cascarano *et al.*, 1982) (see Fig. 9). In his Nobel Prize speech, Jerome Karle (1985) noted that some '*names that have been associated with the preparation and dissemination of computer programs for various aspects of automated, direct structure determination are Ahmed, Andrianov, Beurskens, Germain, Gilmore, Hall, Main, Schenk, Sheldrick, Stewart, Viterbo and Woolfson.*' Woolfson (1987) describes the history of direct methods up to the late 1980's, and research in direct methods is still active, as evidenced by releases of powerful new and updated programs, *ShelXD* (Schneider & Sheldrick, 2002), *SnB* (Rappleye *et al.*, 2002), *SIR/IL MILIONE* (Burla *et al.*, 2007).

### 3.4. Creating software for a tangible scientific purpose

Being on the cutting edge of computing technology as an end in itself can be a highly expensive, all consuming and a potentially fruitless or underachieving process. In the late 1940's, this was on the mind of the US Office of Naval Research (ONR). Despite the ONR believing it obvious that digital computers were the future, this concern ended up benefiting the funding of Ray Pepinsky's X-RAC analogue crystallographic computer: '*We were particularly interested in the fact that he [Pepinsky] was a man with a problem. I should interpolate remarks to the effect that the exciting new developments in digital computing have seized the imagination of a lot of engineers, and we fear that some development is going off into the direction of gadgeteering, that people are losing sight of the problems for which the machines are being built and concentrating on making bigger and better machines.*' (Rees, 1952.)

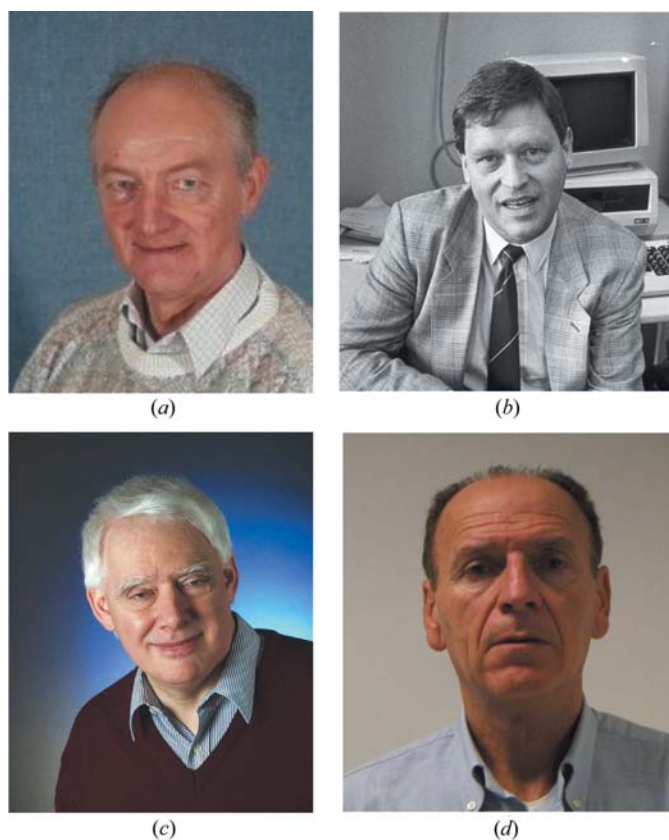
Large and small crystallographic software projects should be careful of not falling into the type of zero-output trap described by historian Alfred Cobban: *'The weakness of much social thought, it seems to me, is that it is so largely concerned with packing its bag (or even with working out a general theory about the way in which a bag should be packed) for a journey which is never taken.'* (Cobban, 1964.) A crystallographic software project or algorithmic idea, to be scientifically useful, should have the aim of it being applied to at least one new scientific journey. Where its intended journey to act upon a current scientific problem is of secondary consideration, it can potentially become a fruitless development. The history of successful crystallographic computing has been where the ends (e.g. successfully solving a crystal structure) justified the means.

### 3.5. Programming languages and crystallographic software

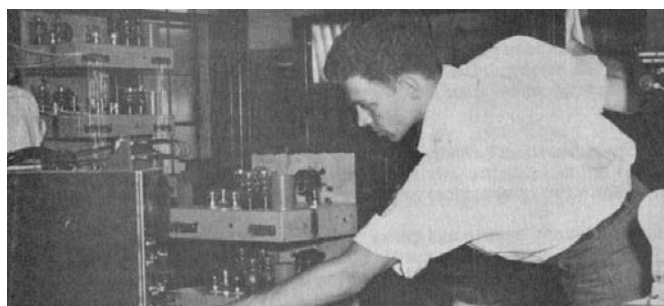
If the aim is to create a software program for the exclusive use of its author, with modern computing, it is near irrelevant as to what programming language the crystallographer uses, as long as it can express the underlying science in the way the author wishes it to be done. With distribution of program binaries, programming language or coding style is irrelevant to users of most crystallographic software. For many styles of

crystallographic problem, even a slow language written with slow code can still perform the desired function for the program author. A caveat on this is that program developers may have to consider, whether desired or not, that their initial versions may end up being expendable as more experience is gained. A first version of an idea might be written in a language like Basic to prove the concept, then perhaps a sequel version written in a faster programming language. Tricks and methods for writing computationally efficient crystallographic code are still relevant where runtime speed is important. The crystallographic computing monographs of the 1960's to 1980's elaborate on some principles found effective in optimal crystallographic coding (e.g. Sheldrick, 1985), including the manual optimization of the assembler within important inner loops of compiled programs (see Rollett, 1970*a* and Shiono, 1970).

**3.5.1. Fortran.** The Fortran programming language was developed by IBM in the mid 1950's to support numerical and scientific computing (Backus *et al.*, 1956, 1957). A crystallographer on the development team was David Sayre, of Sayre equation fame (Sayre, 1952) (see Fig. 10). With machine language of the time, *'programming and debugging accounted for as much as three-quarters of the cost of operating a computer; and obviously as computers got cheaper, this situation would get worse.'* (Backus, 1981*a*.) Thus the need to create an 'automatic programming' system. As also stated by the main originator of Fortran, John Backus, there was an emphasis on program efficiency: *'To this day I believe that our emphasis on object program efficiency rather than on language design was basically correct. I believe that had we failed to produce efficient programs, the widespread use of languages like FORTRAN would have been seriously delayed.'* (Backus, 1981*b*.) This necessity for compiled program efficiency was also indicated by programmer attitudes of the 1950's: *'just as freewheeling westerners developed a chauvinistic pride in their frontiership and a corresponding conservatism, so many programmers of the freewheeling 1950's began to regard themselves as members of a priesthood guarding skills and mysteries far too complex for ordinary mortals. ... The priesthood wanted and got simple mechanical aids for the clerical drudgery which burdened them, but they regarded with*



**Figure 9**  
A small subset of automated direct-methods software developers: (a) Peter Main; (b) Henk Schenk; (c) George Sheldrick and (d) Carmelo Giacobazzo. [Photographs courtesy of the crystallographers in this figure and (b) ©Ronald Sweering, Amsterdam.]



**Figure 10**  
David Sayre testing the first X-RAC power regulators in Auburn, Alabama. [Cropped photo taken from Fig. 109 of the 1952 computing monograph. Courtesy of Penn State University.]

hostility and derision more ambitious plans to make programming accessible to a larger population. To them, it was obviously a foolish and arrogant dream to imagine that any mechanical process could possibly perform the mysterious feats of invention required to write an efficient program.' (Backus, 1980d.) The indifference and disdain for the development of automatic programming as implemented in Fortran only relented after it was shown to work: 'with a few other exceptions, our plans and efforts were regarded with a mixture of indifference and scorn until the final checking out of the compiler, at which time some other groups became more interested.' (Backus, 1980e.)

The first reference to Fortran in *Acta Cryst.* is a 1957 report on a Conference on the Use of the IBM 704 Computer for Crystal-Structure Analysis. Besides descriptions of several crystallographic programs written in Fortran, a Fortran seminar was presented by one of the members of the Fortran development team, David Sayre: 'The fourth session was opened with a FORTRAN seminar. D. Sayre (IBM) explained the principle of FORTRAN (formula translation). It is an artificial language which closely resembles the mathematician's language, but which can be read by the IBM 704 and translated first into a symbolic assembly program (SAP) language and then into the binary machine language. It can be learned in about one or two days, whereas the SAP language requires several weeks of training. The programming effort is a fraction of that spent in ordinary programming and the resultant machine operation is about 90% as efficient. In the discussion novices were sternly warned that the FORTRAN Programmer's Reference Manual must be read with extreme care.' (Vand, 1958.) Backus (1980f) notes that, due to the work by the Fortran development team on optimization techniques, Fortran coded programs could 'compete with and often exceed the efficiency of hand-coded ones'.

As stated by Robinson & Sheldrick (1988) of *SHELX* using Fortran, 'good cases could be made for writing in for example C or PASCAL. FORTRAN is didactically and aesthetically so abysmal that it was by no means an automatic choice', however, 'the overriding argument for FORTRAN is that the "FORTRAN-77" standard has now been very widely implemented, and is the only de facto machine independent language.' Some Fortran programs have had a long productive lifetime. An example still in use is Allan Zalkin's *FORDAP* Fourier synthesis program (Zalkin, 1962). *SHELX-76* is over 30 years old [refer to a copy within Armel Le Bail's Crystallography Source Code Museum (Le Bail, 2002)], but it compiles seamlessly using a freeware GNU G77 Fortran compiler and happily runs on the same Windows-XP based laptop used to write this article. Though replaced by newer *SHELX* releases, *SHELX76* can still be found performing useful crystallographic service in recent literature.

The idea of crystallographic Fortran being a paradise of easily compliable machine-independent programs should be regarded with caution. Ryonosuke Shiono (1970) mentioned 'contrary to popular misconception that a program written in FORTRAN can be universally used in different installations, more often than not it is compiler-dependent and considerable

effort has to be made before it can be compiled for another computer of the same type.' Such concerns were an impetus for two very different approaches. The first was where the *XTAL* suite moved to the *RATMAC* pre-processor language. 'The importance and the inadequacy of Fortran has given birth to a new breed of languages, the Fortran preprocessors. Preprocessor languages have structured features similar to languages such as Pascal, but differ from these in that they are not translated (i.e. compiled) directly into machine code. Rather, the preprocessor language is "preprocessed" into another language, such as Fortran, and this in turn is compiled in the normal way.' (Hall et al., 1980.) The other is stated by Robinson & Sheldrick (1988) that 'programs which adhere strictly to the FORTRAN-77 standard are so transportable that there is little advantage in distributing them in preprocessor code such as *RATMAC*.' For a crystallographically oriented overview of modern (as opposed to archaic) Fortran, readers are referred to Juan Rodríguez-Carvajal's article, *The Once and Everliving FORTRAN: Why Fortran still goes onward and upward while many of its 'replacement' languages have already died* (Rodríguez-Carvajal, 2004).

### 3.5.2. Longevity and usefulness of programming languages:

**Fortran versus ALGOL.** As a consequence of an international meeting on automatic computing held in Germany in 1955, ALGOL, the 'international algorithmic language', was designed by a committee of European and American computer scientists in the late 1950's, with an aim of creating a universal programming language (de Beer, 2006a). Fortran of the time was not well regarded as suitable for building upon. 'Looking at the situation today [1981], one may ask why FORTRAN was not chosen since it has now become more widely accepted than any other language. Today FORTRAN is the property of the computing world, but in 1957 it was an IBM creation and closely tied to use of IBM hardware. For these reasons, FORTRAN was unacceptable as a universal language.' (Perlis, 1981.) The 1960 release of the ALGOL 60 standard (see Naur et al., 1963) attracted, then and after, much admiration amongst computer scientists: 'The more I ponder the principles of language design, and the techniques which put them into practice, the more is my amazement and admiration of ALGOL 60. Here is a language so far ahead of its time, that it was not only an improvement on its predecessors, but also on nearly all its successors.' (Hoare, 1973.) ALGOL is first mentioned in *Acta Cryst.* in 1962, within a Commission on Crystallographic Computing section subtitled *Recommendations on the Publication of Computing Methods and Programs in Crystallographic Journals* (International Union of Crystallography, 1962a). 'The use of general reference languages such as ALGOL gives promise of relieving crystallographers of the recurring need to reprogram computations for new machines. Programmers should, therefore, be encouraged to make available algorithms of their computing procedures so that experience can be gained of the effectiveness with which they can be converted into machine language. In the absence of this experience, no general recommendations could be made in regard to the publication of such algorithms in crystallographic journals.'



**Table 1**

Program types listed in the three editions of the IUCr *World List of Crystallographic Computer Programs*, showing the numbers of Fortran, ALGOL and other programs.

The third edition (1973) discouraged machine-specific entries. Numbers of Fortran programs in the first and second editions can be under-determined due to implementations being labelled as IBM specific.

| World List version | Fortran | ALGOL | Other language | Machine specific/or other | Total |
|--------------------|---------|-------|----------------|---------------------------|-------|
| 1962               | 5       | 27    |                | 545                       | 577   |
| 1965               | 82      | 66    |                | 405                       | 553   |
| 1973               | 189     | 34    | 15             | 5                         | 243   |

**Table 2**

Number of IUCr journal articles within which the words Fortran and ALGOL appear.

Year ranges of editions match that of the IUCr *World List of Crystallographic Computer Programs* and ranges beyond this of roughly a decade.

| Years     | Fortran | ALGOL |
|-----------|---------|-------|
| 1948–1962 | 5       | 3     |
| 1962–1965 | 55      | 13    |
| 1965–1973 | 707     | 77    |
| 1974–1980 | 416     | 39    |
| 1980–1990 | 438     | 16    |
| 1991–2000 | 475     | 2     |
| 2001–2007 | 196     | 0     |

**Table 3**

Numbers of papers within the Computing School monographs that refer to their discussed systems as Fortran, ALGOL, Autocode, Machine Code and Analogue Computers.

Total numbers of papers in the monograph are listed; differences are due to papers on topics not related to a particular computer or program system.

| Year  | Published | Fortran | ALGOL | Auto-code | Machine code | Analogue computer | Total papers |
|-------|-----------|---------|-------|-----------|--------------|-------------------|--------------|
| 1950  | 1952      | n/a     | n/a   | n/a       | 3            | 5                 | 20           |
| 1960  | 1961      | 4       | 0     | 2         | 14           | 2                 | 29           |
| 1969  | 1970      | 16      | 2     | 0         | 1            | 0                 | 50           |
| 1984† | 1985      |         |       |           |              |                   |              |

† In the monograph resulting from the 1984 school, software is commonly mentioned without reference to programming language used; or if mentioned, the language is Fortran.

In contrast with Fortran having defined input–output (I/O), one of the problems of the ALGOL 60 standard was a ‘*lack of input–output procedures, which were needed in practical situations*’ and ‘*enforced implementers to add extensions to the language*’ (de Beer, 2006b). Rollett (1970a), while lauding ALGOL’s design as a programming language, describes practical experiences of using ALGOL for crystallography. Compiled ALGOL code was slow compared to machine-coded programs; required inner-loops to be coded in machine language to obtain acceptable speed; and the ‘*Algol system has rather slow input and output, so that it takes more central processor time (not elapsed time) to output a Fourier map than it does to compute and machine scan it for peaks, which is absurd.*’ Continuing with de Beer’s history of ALGOL, ‘*In America, however, the market demanded a working implementation of a good language, not an academic promise for a*

*better language. During the development of ALGOL 60 IBM became the dominant player in the computer industry. FORTRAN matured into a usable and efficient language and was ported to many machines. As a result, FORTRAN became the de facto standard programming language for scientific computing.*’ (de Beer, 2006a.) The discussion appended to the Rollett paper (Rollett, 1970b) indicates that the theory of ALGOL being a universal machine-independent programming language did not seem to work in practice for the crystallographers who tried it.

Ahmed: *One of the ideas for standardizing the data files was to be able to exchange programs among the different centres. How effective has this been?*

Rollett: *Not very, mainly because the operating systems were different, and this prevented us from using the KDF9 programs that were available in Glasgow when we got our computer.*

Bassi: *I tried to do the same thing in France 4 or 5 years ago but it did not work either.*

In the mid to late 1960’s, all was not going well with further development of the ALGOL language (Hoare, 1981): ‘*At last, in December 1968, in a mood of black depression, I attended the meeting in Munich at which our long-gestated monster was to come to birth and receive the name ALGOL 68.*’ While Stewart (1970) states that ‘*FORTRAN and ALGOL are the languages most used at this time*’ in crystallography, the statistics of Tables 1, 2 and 3 indicate ALGOL’s falling out of use in crystallography compared to Fortran. The last edition of the Association for Computing Machinery’s *ALGOL Bulletin* is listed as being Issue 52, August 1988.

### 3.6. Open source software

Much early crystallographic software was effectively open source, including a custom and practice for local modification and redistribution. Indication of the informal methods of early software distribution and comparison can be gleaned from *Acta Cryst.* (e.g. Jeffrey & Shiono, 1959). Crystallographic software for early IBMs, such as Busing and Levy’s least-squares program for the IBM 704, could be distributed via SHARE (Boyko *et al.*, 1959). Formed in 1955, SHARE was, and still is, a volunteer association run by IBM customers (SHARE, 2007). Backus (1980c) describes the general programming ethos of the ‘freewheeling Fifties’: ‘*Ideas flowed freely along with the liquor at innumerable meetings, as well as in sober private discussions and informally distributed papers. An idea was the property of anyone who could use it.*’ As an indication of the custom and practice of building upon other people’s coding, in the Conference on the Use of the IBM 704 Computer for Crystal-Structure Analysis, ‘*the second session was opened by an introductory paper ‘The least-squares program NY XRI’, presented by D. Sayre (IBM). This program was one of the first ever written for the IBM 704.*’ In the next presentation, ‘*Modifications of Sayre’s IBM 704 machine least-squares program NY XRI for refinement of crystal structures, V. Vand & R. Pepinsky (Penn State) described modifications and extensions to D. Sayre’s program.*’ (Vand, 1958.) Busing & Levy (1961) clarify that Sayre’s NY XRI was

“superseded by Hatch’s NY XR2 and by Vand and Pepinsky’s extensively modified version, PS XR3.” In *Acta Cryst.* articles of the 1960’s, it is not uncommon to find text of the ilk: ‘least-squares refinement using program of crystallographer(s)-X, as modified by crystallographer(s)-Y’ (e.g. Johnson *et al.*, 1968) or of ‘local variations’ (Ibers & Smith, 1964). Standardization in computers, operating systems and use of Internet for fast download of updated software reduces much of the practical incentive for creating local modifications.

The immediate advantage of software authorship is the ability to tackle difficult crystallographic problems outside the default range of other software. Details on software should be published in a peer-reviewed journal such as the IUCr’s *Journal of Applied Crystallography*. This is so that key points and ideas implemented in the software are not lost from view over time. Literature citations for software are considered highly valuable in many circumstances and, where software is distributed, publication also allows the software to be cited easily and consistently. Another advantage of software distribution is that the author can by default become the centre of a web of ideas being passed back by users. This can provide a faster mechanism for the author to create extremely robust software, which can then be applied to the author’s advanced research.

The modern advantage of open-source software for academics, using a GNU GPL or related license, is to ensure authors the freedom to continue working on their software, irrespective of changes in employment. As per the report of Ledford (2007) in the journal *Nature*, it is currently considered a “buyers’ market for academic labour”, spawning a new generation of gypsy scientists leading a peripatetic lifestyle on temporary contracts. Therefore, computational crystallographers may need a method to ensure they can build on their software, and not have it orphaned from them if they move jobs. The most common open-source licence is the GNU GPL (General Public License). If the software author wishes more flexibility with respect to possible commercialization or linking with other code, a BSD-style licence or one of a variety of permissive free software licences can be considered. Building upon existing open-source software is a method to ensure any consequent work is also open source.

### 3.6.1. Community-based development and maintenance of software using Internet-based Concurrent Versions System (CVS).

“MY NAME IS OZYMANDIAS, KING OF KINGS:  
LOOK ON MY WORKS, YE MIGHTY, AND  
DESPAIR!”

*Nothing beside remains: round the decay  
Of that colossal wreck, boundless and bare,  
The lone and level sands stretch far away.*

So ends Shelley’s poem, *Ozymandias*. The OZY header was added to the *Xtal* suite by “founding father Jim Stewart. Jim had a realistic view of the longevity of computers and software, and he thought that Shelley’s poem ‘Ozymandias’ reflected the fragility of massive programming efforts, especially in the 60’s and 70’s.” (Hall *et al.*, 2000.) However, with Internet-based source-code development methods, the problems associated

with the pre-Internet 60’s, 70’s and 80’s may no longer be relevant in making collaborative programming efforts so fragile.

The use of an Internet-based Concurrent Versions System (CVS) provides an effective mechanism for collaborative software projects with world-wide participation. The most important requirement of a successful CVS-based project is a maintainer to oversee the system. This role can be distributed or reassigned as required. Decisions on features or changes can be discussed on a restricted or public mailing list. Patches to the code base found to be unsuitable can be backed out of the system. Via an Internet-based CVS system, programs can develop with participation by interested people as interest takes them. Websites like <http://www.SourceForge.net> offer collaborative web-based facilities including bug tracking, support requests, patches, feature requests, download of binaries, source code and the CVS-based download of the source code. For a modern open software project to have any hope for a continuing non-local developer community, something like a network-accessible CVS system is required. An initial challenge can be to introduce code writers to a CVS type of development system, and encourage participation. Where active development of a CVS-based project is lacking, its presence (such as on <http://www.SourceForge.net>) keeps it accessible and visible: a project cannot regain the interest of the community if it is hidden from view.

### 3.7. Invention, rediscovery, repackaging or reinvention of algorithms and methods

‘We are like dwarfs on the shoulders of giants, so that we can see more than they, and things at a greater distance, not by virtue of any sharpness of sight on our part, or any physical distinction, but because we are carried high and raised up by their giant size.’ (Bernard of Chartres, in John of Salisbury’s *Metalogicon* [1159].)

‘A dwarf standing on the shoulders of a giant may see farther than a giant himself.’ (Didacus Stella, in Robert Burton’s *The Anatomy of Melancholy* [1621].)

‘A dwarf on a giant’s shoulders sees farther of the two.’ (George Herbert, *Jacula Prudentum* [1651].)

‘If I have seen a little further it is by standing on the shoulders of Giants.’ (Sir Isaac Newton, letter to Robert Hooke [5 February 1676].)

A method or algorithm can be invented but once, or rediscovered or reinvented multiple times before it catches on within the community. Of early crystallography, Pepinsky (1952d) noted that: ‘The X-ray analytical methods have not been in the forefront of scientific attention. For this reason, and perhaps more for the reason that we have not sought assistance from other sciences as energetically as we might have, we have found ourselves stumbling upon analytical methods which were well known in other disciplines.’ For convenient manual Fourier summation, A. Lindo Patterson & George Tunell (1942) describe that, in hindsight, they found their strips and stencils method was ‘essentially due to L. Hermann (1890) who made use of it in the analysis of voice records.’ Of analogue

computers for Fourier analysis, Miller (1916) describes a 32-element harmonic synthesizer (an analogue computer for mechanically deriving or projecting the Fourier equation of a curve) applicable to a variety of analyses including acoustics. Miller also includes a historical background (as of 1916) back to the first harmonic synthesizer of Lord Kelvin (Thomson, 1878), which was used for tide prediction. As previously mentioned, the first application of general-use computers for Fourier summation was by Comrie in 1928; but it was over a decade until crystallographers started to make use of this.

There is significant interest in the new structure-solving charge flipping algorithm of Oszlányi & Sütő (2004, 2005). Palatinus describes the use of charge flipping for solving incommensurately modulated structures (Palatinus, 2004) as incorporated in the *SuperFlip* software (Palatinus & Chapuis, 2007). Coelho (2007a) describes the benefits of combining the charge flipping algorithm with the tangent formula of Karle & Hauptman (1956) for solving difficult commensurate structures as implemented in the *TOPAS* software (Coelho, 2007b). Baerlocher *et al.* (2007) describe use of the charge flipping algorithm, as enhanced for application to powder diffraction data, for solving difficult zeolite structures. The 1952 computing monograph includes a comment by Fred Ordway of the US Government's National Bureau of Standards (NBS) that '*a procedure involving successive Fourier inversions, with elimination of negative excursions of the electron density function at each step, has been coded but not yet tried*' (Ordway, 1952b). Limited access to scarce computing resources and the financial computing cost of trying such an algorithm in the early 1950's could have been a significant disincentive. Though as commented by David Watkin, '*he [Ordway] almost discovered charge flipping!!*' (Watkin, 2007.)

A potentially useful method or algorithm may fail to catch on in crystallography for one or more of the following reasons:

- (i) developed outside the crystallographic arena and thus unknown or ignored;
- (ii) not considered applicable to other situations where in hindsight it would have been highly beneficial;
- (iii) reported in a journal not read by the crystallographic computing community;
- (iv) reported in obscure language or not the main topic of the paper;
- (v) unmatched to the computational power of the time;
- (vi) requiring an in-depth knowledge for implementation not apparent from the paper;
- (vii) solved a situation that at the time was not considered as needing solving;
- (viii) needed to be combined with other algorithms before it was effective;
- (ix) not checked whether other areas of crystallography could benefit;
- (x) area at the time was saturated with algorithms and methods and thus little interest in new algorithms;
- (xi) lacked an available prototype implementation to prove the worth of the method to the community;
- (xii) software that implemented the algorithm was unavailable to the community;

- (xiii) not promoted by anyone as being an effective method;
- (xiv) only considered useful by most of its potential community once the algorithm was packaged *via* a suitable program interface (*i.e.* a near-black-box automated program);
- (xv) in hindsight, was 20 years ahead of its time, and thus ignored for 20 years.

### 3.8. Crystallographic source code libraries

For accelerated building of new and modified programs, a number of crystallographic source code libraries are current or in development. The progenitor of these styles of software library initiatives is the Fortran 77 based Cambridge Crystallography Subroutine Library (CCSL) of Matthewman *et al.* (1982). The <http://www.SourceForge.net>-based Computational Crystallography Toolbox (*cctbx*) of Grosse-Kunstleve *et al.* (2002) is a C++/Python based open-source project aimed at high-throughput macromolecular structure determination. The Fortran 95 equivalent to *cctbx* is the general crystallography, powder diffraction and neutron scattering focused Crystallographic Fortran Modules Library (CrysFML) of Rodriguez-Carvajal & Gonzalez-Platas (2005). DANSE (Distributed Data Analysis for Neutron Scattering Experiments) is a project aimed at providing a comprehensive range of interlinked neutron scattering analysis tools *via* a standard framework. The Crystallography Source Code Museum of Armel Le Bail (2002) has been found useful by software authors for browsing through old programs and seeing past implementations. Where software is no longer being maintained, authors should feel encouraged to deposit their code so that new generations of programmers can benefit from their ideas. The algorithms, ideas and science within the code are normally far more important than the style of coding, spaghetti or otherwise, or the programming language.

**3.8.1. Age concern.** Much small-molecule crystallographic software is of ageing design where program authors are either retired or close to retirement. There are concerns within the small-molecule community of a loss of much expert crystallographic software knowledge when these codes are no longer maintained. The UK EPSRC funded 'Age Concern' project (Howard, 2005; Watkin, 2006) is aimed at preventing loss of small-molecule single-crystal-based knowledge. An intent of Age Concern is to develop a knowledge base to rewrite useful ideas within existing software into a source-code library that is compatible with the *cctbx* source-code project. Crystallographers at or near retirement should also consider publishing their distilled wisdom, which is normally of the type that can only be obtained from a full lifetime of scientific and crystallographic experience.

### 3.9. Crystallographic software

**3.9.1. Features of successful crystallographic software.** When creating crystallographic software, it is worth considering what made for prior successful crystallographic software. The following list of features and principles are suggested:

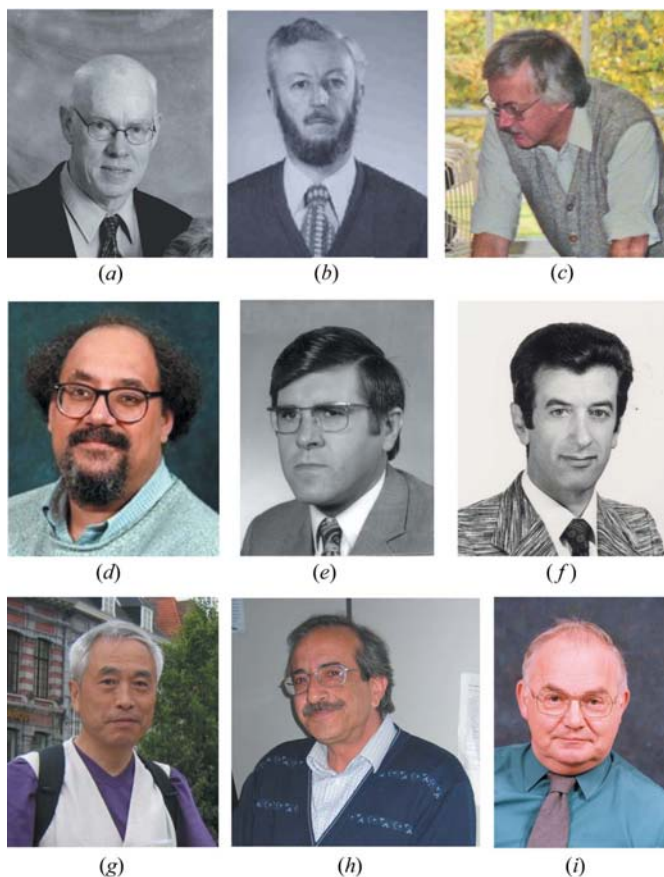
- (i) respects the user's time as being extremely valuable:

- (a) easy and quick to install, requiring minimum (preferably zero) source-code modification or system configuration,
- (b) easy to copy/re-install,
- (c) easy to use,
- (d) core functionality very streamlined and effective to use,
- (e) intelligent use of default values akin to software being an expert system,
- (f) absence of alienating features (bugs fall into this category),
- (g) useful scientist friendly features,
- (h) contains time saving features,
- (i) standard custom-and-practice for program operation;
- (ii) matched to readily available computing power:
  - (a) algorithms matched to the speed of affordable computers,
  - (b) program works on computer with its commonly installed operating system;
- (iii) meets a need of the community that other available programs don't;
- (iv) program is pragmatic:
  - (a) takes the users' priorities above any others, but
  - (b) knows where users (or their data) might perform poorly and anticipates this;
- (v) easily taught by one user to another;
- (vi) easily obtainable:
  - (a) for academics, students and not-for-profit use, software readily obtained at preferably zero or voluntary financial cost,
  - (b) prior to the Internet: easy for users to re-distribute/with Internet: easy to download.

**3.9.2. User interfaces for crystallographic software.** As stated by Robinson & Sheldrick (1988), 'from the crystallographer's point of view, the ease of communicating with a program is probably the most important factor in determining whether he will use it.' If a program is without rival and has functionality important to its potential user base, then the program authors are pretty much free to create any style of interface they believe appropriate. A program can reach a critical mass of users whereby the interface can become a standard within the field of interest, such as Ton Spek's *PLATON* with its range of structure validation and analysis features (Spek, 2003). However, program design running contrary to norms in common computer interface and operation can have consequences for ease of user acceptance. Traditionally in crystallography, batch (file-in, file-out) mode programs were custom and practice for most crystallographic computation. An exemplar example of file-in, file-out operation is the philosophy behind *SHELX*: 'When *SHELX* was first written, communication with *FORTRAN* programs usually took the form of fixed format input on punched cards, and the idea of free format input with extensive use of "default values" (for parameters not specified by the user) was fairly novel' and 'A simple criterion is that an experienced user should not normally need to look at the manual. . . . It is essential to keep the input *SIMPLE*, and to reduce the number of files required or generated by the program to an *ABSOLUTE MINIMUM*.' (Robinson & Sheldrick, 1988.) Another style of program control for single-crystal software was based on interactive

interface designs, such as within *NRC-VAX* (Gabe *et al.*, 1985), *CAOS* (Cerrini & Spagna, 1977; Camalli & Spagna, 1994) and *Crystals* (Betteridge *et al.*, 2003) (see Fig. 11). Gabe (1988) noted that batch systems were developed more for the convenience of the machines than the human operators; and also describes the abilities of interactive software for crystallographic analysis on complex structures and complex crystallographic problems.

For a new generation of computer users raised on Windows-style operating systems (OS), the expectation is that of exploratory interactive interfaces providing instant graphical feedback in the form of Graphical User Interfaces (GUI). Until the mid to late 1990's, it was difficult to near impossible to write a reliable operating-system-independent graphical program system and such programs tended to only run on a single type of operating system. Currently, a variety of toolkits exists in a variety of programming and scripting languages that make practical the writing of OS-independent graphical programs. Implementing GUIs over established software still allows the design option of maintaining back-up access to the traditional modes of program operation, thus not alienating a



**Figure 11**  
A small subset of single-crystal suite developers: (a) Jim Stewart; (b) Eric Gabe; (c) David Watkin; (d) Louis Farrugia; (e) Anthony Spek; (f) Riccardo Spagna; (g) Kenji Okada, (h) Gianluca Cascarano and (i) Kenneth Muir. [Photographs courtesy of the crystallographers in this figure, except (a) courtesy of Marcia Colquhoun, American Crystallographic Association; and (b) courtesy of National Research Council of Canada.]

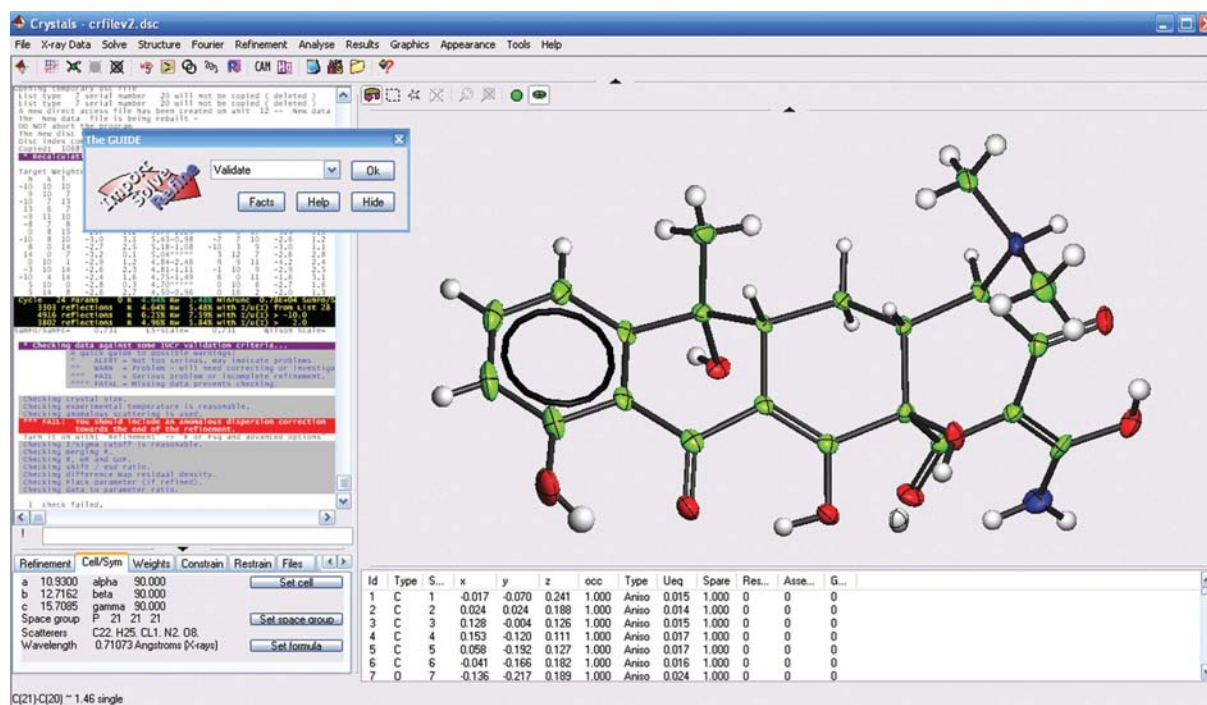


long-standing user base. User-effective examples of programs that have done this include *Rietica Rietveld* software (Hunter & Howard, 2000); *EXPGUI* (Toby, 2001) for the *GSAS* Rietveld suite (Larson & Von Dreele, 2004) and the exceptional Windows interface of the *Crystals* software optimized for single-crystal analysis (Betteridge *et al.*, 2003) (see Fig. 12).

Crystallography suites allow users to avoid the continuous tedium and drudgery of interconverting data and results in the formats of various individual programs under use. Comprehensive crystallographic suites of various shapes and sizes have a history dating back to the early 1960's, where two suites mentioned in the 1965 *IUCr World List of Computer Programs* are the Atlas–Imperial College Crystallographic System (UK) developed mainly by M. G. B. Drew and the *X-RAY63* system (USA) of J. M. Stewart (Stewart, 1963; NASA, 1964). The early NRC software of Ahmed *et al.* (1966) was also based on the benefits of suite-based crystallographic development. The *WinGX* for MS-Windows single-crystal suite by Louis Farrugia (1999) started as a Windows version of the *GX* suite of Mallinson & Muir (1985), which itself was influenced by the *Xtal* suite paper of Hall *et al.* (1980). To the *GX* authors, the *Xtal* suite paper provided 'an excellent account of the possible strategies for crystallographic program systems.' (Mallinson & Muir, 1985.) *WinGX* links to a wide variety of crystallographic programs ranging from absorption correction, structure solution, structure refinement through to structure validation, thus allowing a wide range of routine and non-routine exploratory single-crystal analyses *via* a standard operating system GUI interface.

**3.9.3. Marketing and promotion of software.** Freeware crystallographic software programs, and the methods within, do not distribute or market themselves. Why a software program or method catches on relatively slowly or quickly in the community may involve more than published literature, but also a number of informal interactions pushing the software into view. Carroll Johnson's *ORTEP* structure plotting software, with Mike Burnett now being its developer (Johnson, 1965, 1976; Burnett & Johnson, 1996), had such high impact in the crystallographic community that *ORTEP* became a common synonym for thermal parameters. Carroll Johnson mentions that there was much interest generated in the software 'with *ORTEP* drawings for the stereo slides I made for crystal structure papers at the 1965 ACA meeting in Gatlinburg TN. We had stereo projectors and Polaroid viewing glasses there and with ORNL the host lab for the meeting, I offered to make one free stereo slide for any participant who would send me coordinate data at least a month before the meeting and almost everyone there took me up on the offer. That was a quite successful debugging (and to my surprise also marketing) exercise for *ORTEP*.' (Johnson, 2007.) The same was done for an international biophysics conference held in Vienna in 1966, which included the structures of myoglobin, vitamin B12 coenzyme and poly-L-alanine.

The effective marketing of a new method may end up being done by someone other than its inventor. In the case of the Rietveld method for structure refinement of powder diffraction data, *via* Alan Hewat, then at Harwell, UK. The Rietveld 'method was first reported at the seventh Congress of the IUCr



**Figure 12**

A screen image of *Crystals* for Windows showing part of its exploratory interactive interface with 'organometallic and organic chemist-friendly' real-time graphical feedback of the structure refinement.

in Moscow in 1966. The response was slight, or, rather, non-existent, and it was not until the full implementation of the method was published (Rietveld, 1969a), that reactions came.' (Rietveld, 1993.) It was first implemented in 1969 in ALGOL by Hugo Rietveld at the Pettin reactor in The Netherlands (Rietveld, 1969b); but the ALGOL implementation had low interest in the crystallographic community. The method was rewritten in Fortran by Rietveld in 1970. Alan Hewat, visiting Pettin in 1971, obtained a copy of the Fortran code and modified it when back at his workplace of the Harwell reactor, including addition of anisotropic temperature factors refinement (Hewat, 1973). This code was made highly visible *via* the Harwell neutron facility having a wide neutron user base, and resulted in all Rietveld code in common use until at least the 1980's being derived from this Rietveld-Hewat code. The only known exception is a Rietveld program written by John Taylor and G. W. Cox of the Australian Atomic Energy Commission at Lucas Heights (Taylor & Wilson, 1973), an adaptation of the *ORFLS* single-crystal code of Busing *et al.* (1962). Some of the Rietveld programs originally derived from the Harwell code are still in use and under active development.

### 3.10. Distribution of freeware crystallography software

**3.10.1. Pre-Internet software distribution.** Pre-Internet software distribution *via* formal methods could be expensive and time-consuming due to effort in duplication, media costs and postage. As a possible formal solution to these difficulties, the *SHELX* distribution policy strikes what looks a particularly effective balance, based around a request for a voluntary payment from academics who are in a position to contribute to support overall distribution costs (Robinson & Sheldrick, 1988).

The benefits of informal user redistribution of freeware crystallographic software, if program authors wanted it or not, was that users of software did the work in publicizing, marketing, training and distributing the software. This could be particularly effective when scientific usage of single-user personal computers was expanding, and crystallographic software could be easily copied onto floppy disk with little effort. This would not only save the software author on expensive distribution costs and time, but provide increasing cascades of new users, with (for academic software) all the practical benefit still going to the author in the form of resulting literature citations. An example ranging from the 1960's to the present is that of the *FORDAP* Fourier synthesis software of Zalkin (1962). As *FORDAP* is still being used and archived, and with modification summaries often visible in the Fortran source code header, it is possible to trace some of *FORDAP*'s distribution routes. Of the citation classic *ORFLS* software of Busing *et al.* (1962), which had been cited 3035 times from 1962 to 1982 (Busing, 1982), Busing noted that while they had distributed 395 copies of the program on request, 'there has been much indirect propagation' (Busing, 1982). Where software authors did not wish to see local modifications of their programs, the software could either be distributed as binaries or written in a way

that made local source-code modification unnecessary or difficult.

**3.10.2. Internet-based distribution of software.** At the 1987 crystallographic computing conference, Philip Bourne described the then current state of international computer networks for crystallographers (Bourne, 1988). Software distribution *via* these networks could be problematic due to slow data transfer rates and transfer time outs. Some academics or researchers might only find out their institute use was chargeable at high rates after being presented with the bill. The Mosaic web browser was released in 1993, which popularized the World Wide Web, and the Netscape Navigator web browser was released in 1994. Unix-based computers were favoured initially for Internet use, as it took time for usable TCP-IP network drivers to be released for commercial operating systems running on personal computers. The report of the 7th IUCr Congress and General Assembly, Seattle, 8-17 August 1996, contains a summary of the Internet workshop organized by Howard Flack and included the observation that the Web/W3/Information Superhighway was being increasingly used for crystallographic software distribution. A resulting *Crystallographer's Guide to Internet Tools and Resources* was released as a print and web-accessible resource (Flack, 1996).

When using the modern Internet for software distribution, the level of cost and time associated with pre-Internet manual distribution almost disappears. Announcements to a mailing list or newsgroup can alert users to obtain newer versions; as can update checks written into the software. The time the author saves using Internet distribution compared with manual distribution can be more than given back *via* e-mail user support. An effective method of handling user queries can be to direct users to an e-mail discussion list, where users can help each other, and any directions the software authors provide are visible to all and archived for future referral. Laboratory-based researchers or home-hobbyists alike can, with similar ease, create and make available their ideas as implemented in software to the world. Freeware or commercial software, as binaries or source code, can be easily downloaded with various levels of control for the software author, either completely open, download *via* ftp or website password, or methods to ensure the software is registered before it can be run. If desiring maximum levels of program uptake, it is important that ease of download, installation, starting functionality and core functionality are as streamlined as possible.

## 4. Crystallographic teaching

'On two occasions I have been asked, -"Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?" ... I am not able rightly to apprehend the kind of confusion of ideas that could provoke such a question.' (Charles Babbage, 1864.)

Crystallographic algorithms and software development requires an understanding of the fundamental mathematics and physics of crystallography at a level commensurate with the difficulty of the problems being studied. Since before *Acta Cryst.* was published, there is evidence that experienced crystallographers have been concerned by the effects of lack of available crystallographic teaching. In response to Harry Carlisle's January 1943 Cambridge presentation on the structure of cholesterol iodide, W. L. Bragg allegedly reported that he had "written in a recent review ... that the younger generation of crystallographers hadn't the initiative to tackle really difficult problems and now he would have to write and withdraw it all." (Ferry, 1998a.) The Summarized Proceedings of the Conference on X-ray Analysis – London 1947 (Kellar & Douglas, 1948) describes a variety of requirements, methods, ideas and conclusions within a series of presentations relating to 'What Industry requires of its X-ray crystallographers, and some methods adopted for training them' and 'training of X-ray Crystallographers in Polytechnics and Universities'. As to the teaching methods of e.g. Cox at the University of Leeds, UK, the summary describes that 'to avoid the danger of narrow-mindedness, colloquia, visits to other workers, etc., were considered essential, but most important of all was the maintenance of the proper atmosphere in the laboratory.' Although, in the forward to the 1952 translation of Guinier's book *X-ray Crystallographic Technology*, Kathleen Lonsdale stated 'it is still difficult for a thorough training in the subject to be obtained except in a few places, either in Great Britain or the U.S.A.' (Guinier, 1952). In response to a book review comment by Ray Pepinsky (1953), Lonsdale (1953) clarified that, without quality professional crystallographic education, 'I believe that our work will be pedestrian and our subject regarded simply as a highly specialized form of technology instead of as an important science with many technological applications.' Pepinsky's reply (1954) stated that students should be instilled with the skills for the 'development of competence to learn for themselves'. Glusker (1998) provides a historical overview and advice on the matter of crystallographic teaching with respect to broad brush versus pinpoint detail technique, and the importance of mentors. Turowska-Tyrk (2004) describes a Polish crystallographic course lasting the academic year while Blake *et al.* (2002) describe the history of a British biennial intensive course for teaching crystallography. Using Internet-based technologies, Chapuis & Schoeni (2005) introduce the development of a web-based interactive environment for crystallographic teaching and Orlov *et al.* (2006) show examples of educational possibilities offered by mobile phones. However, Schenk (2004) describes issues that are becoming more common by the downsizing of crystallographic teaching in universities.

In future, the only practical and effective process of crystallographic self-teaching might be *via* modifying or writing crystallographic software. Where teaching is lacking, users of crystallographic tools can be limited to performing only routine analysis as defined within existing crystallographic software. But by developing their own programs, software and algorithm developers are under no such limitation.

## 5. The future

'From a glance at developments since 1955, no one should be eager to predict what a third Conference on computing methods in crystallography in 1970, will involve – other than to say that what we report now will in 1970 appear as ancient as Egypt.' (Pepinsky *et al.*, 1961a.)

The primary requirement for the future of crystallographic software development seems clear cut: it will need human crystallographic expertise. The wide applicability of crystallographic methods involves many who would not regard themselves as crystallographers but as users of crystallographic techniques. Most of these researchers are reliant on crystallographic software tools written by people with expert crystallographic knowledge. A healthy future for the users of crystallography is a wide genetic diversity of algorithms and software enabling varieties of methods for tackling current and future problems. For discovering and implementing this computational science, the challenge is to ensure a critical mass of expertise into the future, and ensure it is not drowned out by the routine pressing of buttons.

## 6. Conclusions

The history of crystallographic computing and software development is that of continual escape from the Sisyphean dungeon: taking the non-routine and making it as routine as possible, which then reveals new scientific challenges and hurdles. The desires of the original computational crystallographers are expected to remain similar for the present and future; that of overcoming crystallographic computational bottlenecks by any method found practical. The act of operating much modern crystallographic software is one where the user has delegated much of the crystallographic scientific thought to the program authors. But the hardware, software and algorithms do not make themselves. Resolution of non-routine and difficult crystallographic problems requires the input of crystallographic expertise commensurate to the problem at hand. The experience gained in solving complex crystallographic problems can then be made more routine by inserting acquired knowledge into software. It is crystallographic software that constitutes one of the main expert knowledge bases within the field of crystallography. The wider and deeper the knowledge base of algorithms and software, the healthier it is for the field of crystallography. The difficult challenge of current and future crystallographic computing is not that of the specific scientific challenges, but maintaining a continuing critical mass of human expertise with which these computational challenges can be tackled.

The author thanks the referees of the Nexus article (published within this issue of *Acta Cryst.*) and editor Henk Schenk for recommending that the topic of this paper should be undertaken, David Watkin for his encouragement to closely examine the early Computing School monographs, Michael Dacombe and Brian McMahon at the IUCr in Chester for supplying the 1962 first and 1965 second editions

of the IUCr *World List of Crystallographic Computer Programs*, and Michael Dacombe for identifying the crystallographers in Fig. 3. The work of the NRC CISTI library service for their ability to obtain articles and books that were way outside the range of standard library and electronic journal download. Alistair McIvor of the NRC at Chalk River for providing a tutorial on the use of oversampling and ‘Gaussian blurring’ for removing Moiré effects in the scanned images from the penicillin and 1952 computing monographs. The people who implemented the digitization of all the IUCr journals back to 1948; it is a wonderful resource. Larry Henling at Caltech for his scanning of some photos, and high-level detective skills and ability to track down and provide leads for the photographs for use in Figs. 3, 4, 5 and 7. Larry Henling again for searching and providing archival information of Caltech and Linus Pauling of the 1940’s. Philip Skroska of the Washington University School of Medicine for photographs and biographical information on P. A. Shaffer Jr. Frank da Cruz of Columbia University and computer pioneer Herbert R. J. Grosch for clarifications on the biographies of Comrie and Eckert. Larry McEwan, machinist at NRC, Chalk River, for discussions on the capabilities of using Meccano for scientific instrumentation. The comments and guidance of the three anonymous reviewers were appreciated; including one who stated the original draft synopsis and abstract did not do justice to the full scope of the paper, and so took the liberty of slightly recasting them. This updated synopsis and abstract were used without further modification. Thanks also go to IUCr Senior Technical Editor Sue Barnes, as recommended by two of the referees, for remedial action on some ‘unusual grammatical constructions’ and other miscellaneous improvements.

## References

- Abrahams, S. C., Robertson, J. M. & White, J. G. (1949). *Acta Cryst.* **2**, 238–244.
- Ahmed, F. R. & Cruickshank, D. W. J. (1953). *Acta Cryst.* **6**, 765–769.
- Ahmed, F. R., Hall, S. R., Pippy, M. E. & Huber, C. P. (1966). *NRC Crystallographic Programs for the IBM 360 System, World List of Crystallographic Computer Programs*, 2nd ed., Appendix, p. 52.
- Armstrong, E. J. (1946). *Phys. Rev.* **70**, 575.
- Babbage, C. (1864). *Passages from the Life of a Philosopher*, p. 67. London: Longman, Green, Longman, Roberts and Green.
- Backus, J. (1980a). *A History of Computing in the Twentieth Century*, edited by N. Metropolis, J. Howlett & G.-C. Rota, p. 125. New York: Academic Press.
- Backus, J. (1980b). *A History of Computing in the Twentieth Century*, edited by N. Metropolis, J. Howlett & G.-C. Rota, p. 126. New York: Academic Press.
- Backus, J. (1980c). *A History of Computing in the Twentieth Century*, edited by N. Metropolis, J. Howlett & G.-C. Rota, p. 127. New York: Academic Press.
- Backus, J. (1980d). *A History of Computing in the Twentieth Century*, edited by N. Metropolis, J. Howlett & G.-C. Rota, pp. 127–128. New York: Academic Press.
- Backus, J. (1980e). *A History of Computing in the Twentieth Century*, edited by N. Metropolis, J. Howlett & G.-C. Rota, p. 131. New York: Academic Press.
- Backus, J. (1980f). *A History of Computing in the Twentieth Century*, edited by N. Metropolis, J. Howlett & G.-C. Rota, p. 132. New York: Academic Press.
- Backus, J. (1981a). *History of Programming Languages*, edited by R. L. Wexelblat, p. 27. New York: Academic Press.
- Backus, J. (1981b). *History of Programming Languages*, edited by R. L. Wexelblat, p. 29. New York: Academic Press.
- Backus, J. W., Beeber, R. J., Best, S., Goldberg, R., Haibt, L. M., Herrick, H. L., Nelson, R. A., Sayre, D., Sheridan, P. B., Stern, H. J., Ziller, I., Hughes, R. A. & Nutt, R. (1957). *The FORTRAN Automatic Coding System*, pp. 187–198. Proceedings Western Joint Computer Conference, Los Angeles, CA, USA.
- Backus, J. W., Beeber, R. J., Best, S., Goldberg, R., Herrick, H. L., Hughes, R. A., Mitchell, L. B., Nelson, R. A., Nutt, R., Sayre, D., Sheridan, P. B., Stern, H. & Ziller, I. (1956). *The FORTRAN Automatic Coding System for the IBM 704 EDPM: Programmer’s Reference Manual*. New York: International Business Machines Corporation.
- Baerlocher, Ch., Gramm, F., Massüger, L., McCusker, L. B., He, Z., Hövöller, S. & Zou, X. (2007). *Science*, **315**, 1113–1116.
- Beer, H. T. de (2006a). *The History of the ALGOL Effort*, p. 11. Masters thesis, Eindhoven University of Technology, The Netherlands.
- Beer, H. T. de (2006b). *The History of the ALGOL Effort*, p. 67. Masters thesis, Eindhoven University of Technology, The Netherlands.
- Beevers, C. A. (1939). *Proc. Phys. Soc.* **51**, 660–667.
- Betteridge, P. W., Carruthers, J. R., Cooper, R. I., Prout, K. & Watkin, D. J. (2003). *J. Appl. Cryst.* **36**, 1487.
- Bijvoet, J. M. (1962). *Fifty Years of X-ray Diffraction*, edited by P. P. Ewald, ch. 21, p. 476. Reprinted in pdf format for the IUCr XVIII Congress, Glasgow, Scotland, <http://www.iucr.org/iucr-top/publ/50YearsOfXrayDiffraction/>.
- Blake, A. J., Clegg, W., Howard, J. A. K., Main, P., Parsons, S., Watkin, D. J. & Wilson C. (2002). *Z. Kristallogr.* **217**, 427–428.
- Bourne, P. E. (1988). *Crystallographic Computing 4: Techniques and New Technologies*, edited by N. W. Isaacs & M. R. Taylor, pp. 426–439. Oxford University Press.
- Boyko, E. R., Busing, W. R., Fitzwater, D. R., Van der Helm, D., Jeffrey, G. A., Jensen, L. H., Merritt, L. L., Shiono, R. & Williams, D. E. (1959). *Acta Cryst.* **12**, 350–351.
- Bromley, A. G. (1990). *Computing Before Computers*, edited by W. Aspray, pp. 156–159. Ames: Iowa State University Press.
- Bunn, C. W. (1946a). *Chemical Crystallography*, p. 268. Oxford University Press.
- Bunn, C. W. (1946b). *Chemical Crystallographic*, p. 270. Oxford University Press.
- Burla, M. C., Caliandro, R., Camalli, M., Carrozzini, B., Cascarano, G. L., De Caro, L., Giacovazzo, C., Polidori, G., Siliqi, D. & Spagna, R. (2007). *J. Appl. Cryst.* **40**, 609–613.
- Burnett, M. N. & Johnson, C. K. (1996). *ORTEP-III: Oak Ridge Thermal Ellipsoid Plot Program for Crystal Structure Illustrations*, Oak Ridge National Laboratory Report ORNL-6895. Oak Ridge National Laboratory, Oak Ridge, TN, USA.
- Busing, W. R. (1982). *Curr. Contents*, **13**, 20.
- Busing, W. R. & Levy, H. A. (1961). *Computer Methods and the Phase Problem in X-ray Crystal Analysis*, edited by R. Pepinsky, J. M. Robertson & J. C. Speakman, p. 146. New York: Pergamon Press.
- Busing, W. R., Martin, K. O. & Levy, H. A. (1962). *ORFLS, a Fortran Crystallographic Least-Squares Program*, Report ORNL-TM-305. Oak Ridge National Laboratory, Oak Ridge, TN, USA.
- Camalli, M. & Spagna, R. (1994). *J. Appl. Cryst.* **27**, 861–862.
- Cascarano, G., Giacovazzo, C., Polidori, G., Spagna, R. & Viterbo, D. (1982). *Acta Cryst.* **A38**, 663–670.
- Cerrini, S. & Spagna, R. (1977). *Crystallographic Software for a Minicomputer, IV*, European Crystallography Meeting, Oxford, UK, Abstract A-212.



- Chapuis, G. & Schoeni, N. (2005). *Acta Cryst.* **A61**, C121–C122.
- Clarke, H. T., Johnson, J. R. & Robinson, R. (1949). *The Chemistry of Penicillin*, edited by H. T. Clarke, J. R. Johnson & R. Robinson, p. vii. Princeton University Press.
- Cobban, A. (1964). *The Social Interpretation of the French Revolution*, p. 23. Cambridge University Press.
- Coelho, A. A. (2007a). *Acta Cryst.* **A63**, 400–406.
- Coelho, A. A. (2007b). *TOPAS-Academic V4.1-beta*, <http://members.optusnet.com.au/alancoelho/>.
- Comrie, L. J. (1932). *Mon. Not. R. Astron. Soc.* **92**, 694–707.
- Comrie, L. J. (1946). *Math. Tables Other Aids Comput. (MTAC)*, **2**, 149–159.
- Cox, E. G., Gross, L. & Jeffrey, G. A. (1947a). *Nature (London)*, **156**, 433–434.
- Cox, E. G., Gross, L. & Jeffrey, G. A. (1947b). *Proc. Leeds Philos. Soc.* **5**, 1–13.
- Cox, E. G. & Jeffrey, G. A. (1949). *Acta Cryst.* **2**, 341–343.
- Croarken, M. (1999). *IEEE Ann. Hist. Comput.* **21**, 70–71.
- Crowfoot, D., Bunn, C. W., Rogers-Low, B. W. & Turner-Jones, A. (1949a). *The Chemistry of Penicillin*, edited by H. T. Clarke, J. R. Johnson & R. Robinson, pp. 310–367. Princeton University Press.
- Crowfoot, D., Bunn, C. W., Rogers-Low, B. W. & Turner-Jones, A. (1949b). *The Chemistry of Penicillin*, edited by H. T. Clarke, J. R. Johnson & R. Robinson, p. 355. Princeton University Press.
- Cruickshank, D. W. J. (1949). *Acta Cryst.* **2**, 65–82.
- Cruz, F. da (2004). *Wallace Eckert*, Columbia 250th Anniversary Profile. New York: Columbia University, <http://www.columbia.edu/acis/history/eckert.html>.
- Cruz, F. da (2006a). *L. J. Comrie*, Columbia University Computing History. New York: Watson Laboratory, Columbia University, <http://www.columbia.edu/acis/history/comrie.html>.
- Cruz, F. da (2006b). *Professor Wallace J. Eckert*, Columbia University Computing History. New York: Watson Laboratory, Columbia University, <http://www.columbia.edu/acis/history/comrie.html>.
- Dodson, G. (2002). *Biogr. Mem. Fellows R. Soc. London*, **48**, 179–219.
- Donohue, J. & Schomaker, V. (1949). *Acta Cryst.* **2**, 344–347.
- Eckert, W. J. (1947). *J. Chem. Educ.* **24**, 54–57.
- Evans, R. C. & Peiser, H. S. (1942). *Proc. Phys. Soc.* **54**, 457–462.
- Farrugia, L. J. (1999). *J. Appl. Cryst.* **32**, 837–838.
- Ferry, G. (1998a). *Dorothy Hodgkin: a Life*, p. 189. London: Granta Books.
- Ferry, G. (1998b). *Dorothy Hodgkin: A Life*, p. 212. London: Granta Books.
- Flack, H. (1996). Editor (*IUCr*) *Crystallographer's Guide to Internet Tools and Resources*, <http://www.iucr.org/cww-top/int.index.html>.
- Flowers, B. H., Black, G., Churchouse, R. F., Collinge, B., Roberts, K. V., Seaton, M. J., Laver, F. J. M., Cohen, A. V. & Tanner, D. W. (1966). *A Report of a Joint Working Group on Computers for Research*, Council for Scientific Policy and the University Grants Committee, Cmnd 2883. London: Her Majesty's Stationery Office.
- Freeth, T., Bitsakis, Y., Moussas, X., Seiradakis, J. H., Tselikas, A., Mangou, H., Zafeiropoulou, M., Hadland, R., Bate, D., Ramsey, A., Allen, M., Crawley, A., Hockley, P., Malzbender, T., Gelb, D. *et al.* (2006). *Nature (London)*, **444**, 587–591.
- Friedlander, P. H., Love, W. & Sayre, D. (1955). *Acta Cryst.* **8**, 732.
- Gabe, E. (1988). *Crystallographic Computing 4: Techniques and New Technologies*, edited by N. W. Isaacs & M. R. Taylor, pp. 381–392. Oxford University Press.
- Gabe, E., Lee, F. L. & Le Page, Y. (1985). *Crystallographic Computing 3: Data Collection, Structure Determination, Proteins, and Databases*, edited by G. M. Sheldrick, C. Kruger & R. Goddard, pp. 167–174. Oxford University Press.
- Glusker, J. P. (1998). *Acta Cryst.* **A54**, 707–715.
- Grosch, H. R. J. (2003). *COMPUTER: Bit Slices From a Life*, ch. 7, Web edition, <http://www.columbia.edu/acis/history/computer.html>.
- Grosse-Kunstleve, R. W., Sauter, N. K., Moriarty, N. W. & Adams, P. D. (2002). *J. Appl. Cryst.* **35**, 126–136.
- Guinier, A. (1952). *X-ray Crystallographic Technology*, translated by T. L. Tippell and edited by K. Lonsdale, p. xiii. London: Hilger and Watts.
- Hall, S. R., du Boulay, D. J. & Olthof-Hazekamp, R. (2000). Editors. *Xtal3.7 System*. University of Western Australia, <http://Xtal.SourceForge.net/>.
- Hall, S. R., Stewart, J. M. & Munn, R. J. (1980). *Acta Cryst.* **A36**, 979–989.
- Helm, D. van der (2004). *IUCr Commission on Crystallographic Computing Newsletter*, No. 4, pp. 92–99, <http://www.iucr.org/iucr-top/comm/ccom/newsletters/>.
- Henling, L. (2007). Personal communication.
- Hermann, L. (1890). *Archiv gesammte Physiol.* **47**, 44–53.
- Hewat, A. W. (1973). *The Rietveld Computer Program for the Profile Analysis of Neutron Diffraction Powder Patterns, Modified for Anisotropic Thermal Vibration*, Atomic Energy Research Establishment, Report No. R7350, Harwell, UK.
- Hoare, C. A. R. (1973). *Hints of Programming Language Design*, Computer Science Department Report No. CS-403, STAN-CS-73-403, p. 27, Stanford University, CA, USA.
- Hoare, C. A. R. (1981). *Commun. ACM*, **24**, 75–83.
- Hodgson, M. L., Clews, C. J. B. & Cochran, W. (1949). *Acta Cryst.* **2**, 113–116.
- Howard, J. A. K. (2005). *Age Concern: Crystallographic Software for the Future*, EP/C536274/1, Engineering and Physical Sciences Research Council, UK.
- Hughes, E. W. (1940). *J. Am. Chem. Soc.* **62**, 1258–1267.
- Hughes, E. W. (1941). *J. Am. Chem. Soc.* **63**, 1737–1752.
- Hughes, E. W. (1952a). *Computing Methods and the Phase Problem in X-ray Crystal Analysis*, edited by R. Pepinsky, p. 141. Department of Physics, The Pennsylvania State College, USA.
- Hughes, E. W. (1952b). *Computing Methods and the Phase Problem in X-ray Crystal Analysis*, edited by R. Pepinsky, p. 145. Department of Physics, The Pennsylvania State College, USA.
- Hughes, E. W. (1952c). *Computing Methods and the Phase Problem in X-ray Crystal Analysis*, edited by R. Pepinsky, pp. 141–147. Department of Physics, The Pennsylvania State College.
- Huml, K. (1985a). *Crystallographic Computing 3: Data Collection, Structure Determination, Proteins, and Databases*, edited by G. M. Sheldrick, C. Kruger & R. Goddard, pp. 131–145. Oxford University Press.
- Huml, K. (1985b). *Crystallographic Computing 3: Data Collection, Structure Determination, Proteins, and Databases*, edited by G. M. Sheldrick, C. Kruger & R. Goddard, p. 140. Oxford University Press.
- Hunter, B. A. & Howard, C. J. (2000). *LHPM: a Computer Program for Rietveld Analysis of X-ray and Neutron Powder Diffraction Patterns*, Australian Nuclear Science and Technology Organization, Lucas Heights, Sydney, Australia.
- Ibers, J. A. & Smith, G. W. (1964). *Acta Cryst.* **17**, 190–197.
- Institute of Physics (1947). *J. Sci. Instrum.* **24**, 20.
- International Union of Crystallography (1962a). *Acta Cryst.* **15**, 515–516.
- International Union of Crystallography (1962b). *Acta Cryst.* **15**, 1190.
- International Union of Crystallography (1965). *Acta Cryst.* **18**, 981–982.
- International Union of Crystallography (1973). *J. Appl. Cryst.* **6**, 309–346.
- Jeffrey, G. A. & Shiono, R. (1959). *Acta Cryst.* **12**, 819.
- Johnson, C. K. (1965). *ORTEP: a Fortran Thermal Ellipsoid Plot Program for Crystal Structure Illustrations*, Oak Ridge National Laboratory Report ORNL-3794. Oak Ridge National Laboratory, Oak Ridge, TN, USA.
- Johnson, C. K. (1976). *ORTEP-II: a Fortran Thermal-Ellipsoid Plot Program for Crystal Structure Illustrations*, Oak Ridge National

- Laboratory Report ORNL-5138. Oak Ridge National Laboratory, Oak Ridge, TN, USA.
- Johnson, C. K. (2007). Personal communication.
- Johnson, Q., Wood, D. H., Smith, G. S. & Ray, A. E. (1968). *Acta Cryst.* **B24**, 274–276.
- Jones, P. D. (1977). *Acta Cryst.* **A33**, 20–24.
- Karle, J. (1985). In *Nobel Lectures, Chemistry 1981–1990*, edited by T. Frängsmyr & B. G. Malmström. Singapore: World Scientific.
- Karle, J. & Hauptman, H. (1956). *Acta Cryst.* **9**, 635–651.
- Kellar, J. N. & Douglas, A. M. B. (1948). *J. Sci. Instrum.* **25**, 90–95.
- Larson, A. C. & Von Dreele, R. B. (2004). *GSAS. General Structure Analysis System Manual*, LAUR, pp. 86–748. Los Alamos National Laboratory, Los Alamos, NM, USA.
- Le Bail, A. (2002). *Crystallography Source Code Museum*, <http://www.cristal.org/museum/>.
- Ledford, H. (2007). *Nature (London)*, **445**, 678–679.
- Lipson, H. & Beevers, C. A. (1936). *Proc. Phys. Soc. London*, **48**, 772–780.
- Lonsdale, K. (1953). *Acta Cryst.* **6**, 874–875.
- Lu, C.-S., Hughes, E. W. & Giguère, P. A. (1941). *J. Am. Chem. Soc.* **63**, 1507–1513.
- Macewan, D. & Beevers, C. A. (1942). *J. Sci. Instrum.* **19**, 150–156.
- McLachlan, D. Jr (1957). *X-ray Crystal Structure*, ch 8. New York: McGraw-Hill.
- McLachlan, D. Jr & Glusker, J. P. (1983). Editors. *Crystallography in North America*, p. 191. New York: American Crystallographic Association.
- McLintock, A. H. (1966). Editor. *COMRIE, Leslie John, F.R.S., In An Encyclopaedia of New Zealand*. New Zealand Ministry for Culture and Heritage.
- Magnússon, M. (2003). *The Advent of the First General Purpose Computer in Iceland and its Impact on Science and Engineering*. History of Nordic Computing, Trondheim, Norway.
- Main, P., Fiske, S. J., Hull, S. E., Lessinger, L., Germain, G., Declercq, J.-P. & Woolfson, M. M. (1980). *MULTAN. A System of Computer Programs for the Automatic Solution of Crystal Structures from X-ray Diffraction Data*. Universities of York, England, and Louvain, Belgium.
- Mallinson, P. R. & Muir, K. W. (1985). *J. Appl. Cryst.* **18**, 51–53.
- Massey, H. S. W. (1953). *Obit. Not. Fellows R. Soc.* **8**, 97–107.
- Matthewman, J. C., Thompson, P. & Brown, P. J. (1982). *J. Appl. Cryst.* **15**, 167–173.
- Miller, D. C. (1916). *J. Franklin Inst.* **181**, 51–81.
- NASA (1964). *X-ray63. Program System for X-ray Crystallography for the IBM 709-7090-7094*, National Aeronautics and Space Administration Technical Report, Report No. NASA-CR-56472; Document ID: 19640012512, <http://ntrs.nasa.gov/>.
- Naur, P., Backus, J. W., Bauer, F. L., Green, J., Katz, C., McCarthy, J., Naur, P., Perlis, A. J., Rutishauser, H., Samuelson, K., Vauquois, B., Wegstein, J. H., van Wijngaarden, A. & Woodger, M. (1963). *Num. Math.* **4**, 420–453.
- Niggli, A. (1961). *Computer Methods and the Phase Problem in X-ray Crystal Analysis*, p. 14, edited by R. Pepinsky, J. M. Robertson & J. C. Speakman. New York: Pergamon Press.
- Nyburg, S. C. (2004). *IUCr Commission on Crystallographic Computing Newsletter*, No. 3, p. 41, <http://www.iucr.org/iucr-top/comm/ccom/newsletters/>.
- Ordway, F. (1952a). *Computing Methods and the Phase Problem in X-ray Crystal Analysis*, edited by R. Pepinsky, p. 148. Department of Physics, The Pennsylvania State College, USA.
- Ordway, F. (1952b). *Computing Methods and the Phase Problem in X-ray Crystal Analysis*, edited by R. Pepinsky, p. 161. Department of Physics, The Pennsylvania State College, USA.
- Orlov, I., Schoeni, N. & Chapuis, G. (2006). *J. Appl. Cryst.* **39**, 595–597.
- Oszlányi, G. & Sütő, A. (2004). *Acta Cryst.* **A60**, 134–141.
- Oszlányi, G. & Sütő, A. (2005). *Acta Cryst.* **A61**, 147–152.
- Palatinus, L. (2004). *Acta Cryst.* **A60**, 604–610.
- Palatinus, L. & Chapuis, G. (2007). *J. Appl. Cryst.* **40**, 786–790.
- Parker, A. M. B., Stokes, A. R. & Wilson, A. J. C. (1945). *J. Sci. Instrum.* **22**, 131–138.
- Patterson, A. L. & Tunell, G. (1942). *Am. Mineral.* **27**, 655–678.
- Pauling, L. (1940). 26 January 1940, *It's in the Blood! A Documentary History of Linus Pauling, Hemoglobin and Sickle Cell Anemia*. Special Collections, The Valley Library, Oregon State University, USA.
- Pauling, L. (1943). 8 November 1943, *It's in the Blood! A Documentary History of Linus Pauling, Hemoglobin and Sickle Cell Anemia*. Special Collections, The Valley Library, Oregon State University, USA.
- Pauling, L. (1945). 12 February 1945, *It's in the Blood! A Documentary History of Linus Pauling, Hemoglobin and Sickle Cell Anemia*. Special Collections, The Valley Library, Oregon State University, USA.
- Pepinsky, R. (1947). *J. Appl. Phys.* **18**, 601–604.
- Pepinsky, R. (1952a). Editor. *Computing Methods and the Phase Problem in X-ray Crystal Analysis*. Department of Physics, The Pennsylvania State College, USA.
- Pepinsky, R. (1952b). *Computing Methods and the Phase Problem in X-ray Crystal Analysis*, edited by R. Pepinsky, Foreword. Department of Physics, The Pennsylvania State College, USA.
- Pepinsky, R. (1952c). *Computing Methods and the Phase Problem in X-ray Crystal Analysis*, edited by R. Pepinsky, p. 2. Department of Physics, The Pennsylvania State College, USA.
- Pepinsky, R. (1952d). *Computing Methods and the Phase Problem in X-ray Crystal Analysis*, edited by R. Pepinsky, p. 8. Department of Physics, The Pennsylvania State College, USA.
- Pepinsky, R. (1952e). *Computing Methods and the Phase Problem in X-ray Crystal Analysis*, edited by R. Pepinsky, p. 167. Department of Physics, The Pennsylvania State College, USA.
- Pepinsky, R. (1952f). *Computing Methods and the Phase Problem in X-ray Crystal Analysis*, edited by R. Pepinsky, p. 276. Department of Physics, The Pennsylvania State College, USA.
- Pepinsky, R. (1953). *Acta Cryst.* **6**, 751–752.
- Pepinsky, R. (1954). *Acta Cryst.* **7**, 520–521.
- Pepinsky, R., Van den Hende, J. & Vand, V. (1961a). *Computer Methods and the Phase Problem in X-ray Crystal Analysis*, p. 155, edited by R. Pepinsky, J. M. Robertson & J. C. Speakman. New York: Pergamon Press.
- Pepinsky, R., Van den Hende, J. & Vand, V. (1961b). *Computer Methods and the Phase Problem in X-ray Crystal Analysis*, p. 156, edited by R. Pepinsky, J. M. Robertson & J. C. Speakman. New York: Pergamon Press.
- Perlis, A. J. (1981). *History of Programming Languages*, edited by R. L. Wexelblat, p. 77. New York: Academic Press.
- Rappleye, J., Innus, M., Weeks, C. M. & Miller, R. (2002). *J. Appl. Cryst.* **35**, 374–376.
- Rees, M. (1952). *Computing Methods and the Phase Problem in X-ray Crystal Analysis*, edited by R. Pepinsky, p. 1. Department of Physics, The Pennsylvania State College.
- Rietveld, H. M. (1969a). *J. Appl. Cryst.* **2**, 65–71.
- Rietveld, H. M. (1969b). *An ALGOL Program for the Refinement of Nuclear and Magnetic Structures by the Profile Method*, Research Report RCN-104. Reactor Centrum Nederland.
- Rietveld, H. M. (1993). *The Rietveld Method*, edited by R. A. Young, pp. 39–42. Oxford University Press.
- Robinson, W. & Sheldrick, G. M. (1988). *Crystallographic Computing 4: Techniques and New Technologies*, edited by N. W. Isaacs & M. R. Taylor, pp. 366–377. Oxford University Press/IUCr.
- Rodríguez-Carvajal, J. (2004). *IUCr Commission on Crystallographic Computing Newsletter*, No. 3, pp. 32–40, <http://www.iucr.org/iucr-top/comm/ccom/newsletters/>.
- Rodríguez-Carvajal, J. & Gonzalez-Platas, J. (2005). *Acta Cryst.* **A61**, C22.

- Rollett, J. S. (1961). *Computer Methods and the Phase Problem in X-ray Crystal Analysis*, edited by R. Pepinsky, J. M. Robertson & J. C. Speakman, p. 88. New York: Pergamon Press.
- Rollett, J. S. (1970a). *Crystallographic Computing*, edited by F. R. Ahmed, p. 304. Copenhagen: Munksgaard.
- Rollett, J. S. (1970b). *Crystallographic Computing*, edited by F. R. Ahmed, p. 306. Copenhagen: Munksgaard.
- Santander, M. (1992). *Am. J. Phys.* **60**, 782–787.
- Sayre, D. (1952). *Acta Cryst.* **5**, 60–65.
- Schenk, H. (1988). *Crystallographic Computing 4: Techniques and New Technologies*, edited by N. W. Isaacs & M. R. Taylor, pp. 31–43. Oxford University Press/IUCr.
- Schenk, H. (2004). *Acta Cryst.* **A60**, s105.
- Schneider, T. R. & Sheldrick, G. M. (2002). *Acta Cryst.* **D58**, 1772–1779.
- Shaffer, P. A. Jr, Schomaker, P. & Pauling, L. (1946). *J. Chem. Phys.* **14**, 648–658.
- SHARE (2007). <http://www.share.org/>.
- Sheldrick, G. M. (1976). *SHELX76. Program for Crystal Structure Determination*. University of Cambridge, England.
- Sheldrick, G. M. (1985). *Crystallographic Computing 3: Data Collection, Structure Determination, Proteins, and Databases*, edited by G. M. Sheldrick, C. Kruger & R. Goddard, pp. 175–183. Oxford University Press.
- Shiono, R. (1970). *Crystallographic Computing*, edited by F. R. Ahmed, pp. 312–315. Copenhagen: Munksgaard.
- Shoemaker, D. P. (1961). *Computer Methods and the Phase Problem in X-ray Crystal Analysis*, edited by R. Pepinsky, J. M. Robertson & J. C. Speakman, p. 300. New York: Pergamon Press.
- Spek, A. L. (2003). *J. Appl. Cryst.* **36**, 7–13.
- Spicer, D. (2005). *IEEE Ann. Hist. Comput.* **27**, 3, 33–43.
- Stewart, J. M. (1963). *X-RAY63. Program System for X-ray Crystallography*. University of Maryland, College Park, Maryland, USA.
- Stewart, J. M. (1970). *Crystallographic Computing*, edited by F. R. Ahmed, p. 298. Copenhagen: Munksgaard.
- Stewart, J. M. & Bourne, P. E. (1996). International Union of Crystallography Computing Schools, IUCr Commission on Crystallographic Computing, <http://www.iucr.org/iucr-top/comm/ccom/School96/jms/jms.html>.
- Swift, E. H. (1947). *J. Chem. Educ.* **24**, 574–583.
- Taylor, J. C. & Wilson, P. W. (1973). *Acta Cryst.* **B29**, 1073–1076.
- Toby, B. H. (2001). *J. Appl. Cryst.* **34**, 210–213.
- Thomson, W. (1878). *Proc. R. Soc. London*, **27**, 371–373.
- Turowska-Tyrk, I. (2004). *Acta Cryst.* **A60**, s107.
- Vand, V. (1958). *Acta Cryst.* **11**, 311–314.
- Watkin, D. J. (2006). *Age Concern: Crystallographic Software for the Future*. Report EP/C536282/1, Engineering and Physical Sciences Research Council, UK.
- Watkin, D. (2007). Personal communication.
- Woolfson, M. M. (1987). *Acta Cryst.* **A43**, 593–612.
- Zalkin, A. (1962). *FORDAP*. University of California, Berkeley, CA, USA.