Data Analysis and Processing in X-ray Diffraction Studies of Scattering From Myosin Heads in Muscle

Thesis

Data analysis and processing in X-ray diffraction studies of scattering from myosin heads in muscle.

M.C.Prestidge

19 April 1991

Date of submission: 12th June 1991
Date of award: 17th September 1991

Contents

List of Figures

List of Tables

Abstract

The X-ray diffraction pattern of muscle tissue contains sharp reflections from ordered structures, and a region of diffuse scatter from disordered structures. The sharp reflections have been studied in detail over many years, but the diffuse scatter has received much less attention, apart from the pioneering work of J.Lowy and F.R.Poulsen, reviewed in the book 'Fibrous Protein Structure' edited by J.Squire and P.Vibert (Academic Press, 1987).

In this thesis a set of FORTRAN programs were written to analyse diffraction data, with particular emphasis on the diffuse scatter. Film patterns from some synchrotron experiments and from some of J.Lowy and F.R.Poulsen's archived material were analysed, to study the effects of varying osmolarity and/or sarcomere length on the sharp reflections and the diffuse scatter. Conclusions were drawn about the feasibility of using the diffuse scatter as an indication of changes in myosin subfragment 1 (the head of the myosin molecule), and of retrieving scattering data from the archived material which had been taken largely to observe sharp reflections.

It was concluded that although the archived material could give interesting indications about physiological effects, new experiments were needed to confirm these indications, and these new experiments should be designed specifically towards the diffuse scattering data.

In the course of this work, I have considered the possibility of using Guinier plots to derive information on head shapes from data on intact muscle, and the precision with which such information can be obtained. I conclude that the Guinier plots do contain information about the myosin heads in intact muscle, but that the accuracy of the information may not be much better than ±10%, and this may not be sufficient to confirm or deny possible conformation changes in the myosin head. However, I do not consider that the work done in this thesis reached the limit of accuracy, and further experiments may be worthwhile.

Acknowledgements

I would like to thank my supervisors, Jack Lowy and Finn Poulsen, for suggesting and guiding the course of my work, and particularly for contributing the archive X-ray patterns analysed here. Chris Staddon assisted in processing some of the patterns discussed in Chapter 6, and I am grateful for many useful discussions on error analysis. I am also grateful to Gerald Elliott for a great deal of help and encouragement and particularly for advice while I was writing up this thesis.

Chapter 1

Introduction

1.1 The background

The sliding filament model of muscle was published in 1954 in simultaneous papers in Nature by A.F.Huxley and R.Niedergerke [12] and E.J.Hanson and H.E.Huxley [17]. This classical model of muscle shortening postulates that shortening occurs by the sliding of overlapping protein filaments. There are two types of filaments: the thin filaments, composed chiefly of actin, and the thick filaments, composed of myosin. The lengths of these filaments remains constant during contraction (see Figure 1.1). This model has remained virtually unchallenged since its development, and is supported by evidence from light microscopy, electron microscopy and X-ray diffraction. This evidence is described in detail in a number of reviews, for example those by C.Bagshaw [1], A.F.Huxley [14], and R.Cooke [3].

The striation pattern of muscle seen under the light microscope first gave rise to the sliding filament model. Since the inception of the model, attention has been focussed on the mechanism responsible for the sliding of the filaments. An early idea for this mechanism was suggested by A.F.Huxley in 1957 [13]. The idea of independent force generators operating in the region of overlap between actin and myosin was suggested by the shape of the plot of tetanic tension versus sarcomere length, first measured by R.Ramsey and S.Street [34] and later, more precisely, by A.Gordon, A.F.Huxley and F.Julian [8]. Over a certain range of muscle lengths, the tension developed

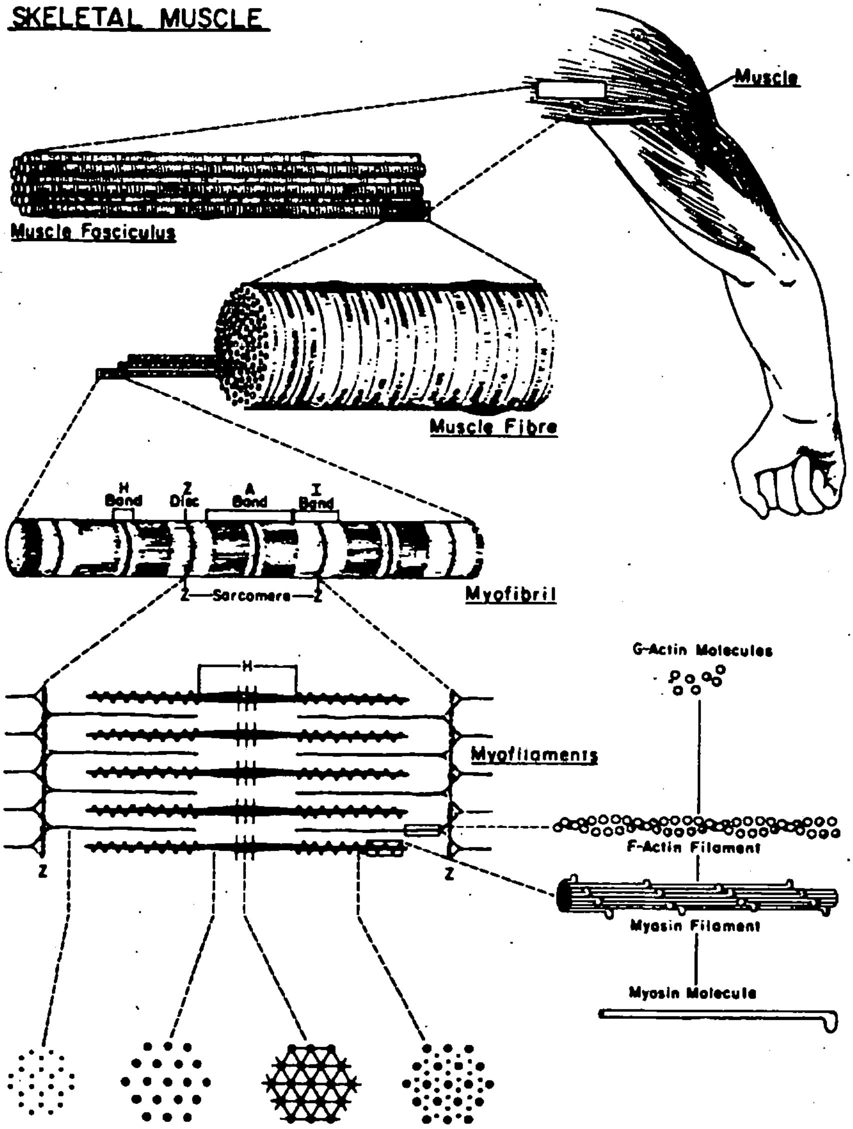


Figure 1.1 The levels of structure in muscle. From C.R.Bagshaw's review.

by the muscle was found to be proportional to the overlap, a relation unlikely to occur if the force generated by each unit were influenced by the number of units acting. More elaborate models involving the cyclic attachment and detachment of crossbridges linking the actin and myosin filaments were put forwards by H.E.Huxley [21] and by A.F.Huxley and R.Simmons [15]. The crossbridges were identified as the S1 portion of the myosin molecule, linked to the backbone of the filament by a flexible chain, the S2 portion, so that the myosin head, S1, could move radially and asimuthally without much axial movement. This would enable the head to attach to actin at a given angle, independent of filament separation (which varies with sarcomere length). Some reaction involving the hydrolysis of ATP would then occur, causing the angle of attachment of S1 to change and generating tension or muscle shortening (see Figure 1.2). On these models, the properties and behavior of the myosin head were of great importance, and were therefore studied by very many diverse methods.

1.2 Studies of S1 in muscle

Movement of crossbridges from myosin to actin occurs in contracting and rigor muscle. This is shown by the changes in the intensities of equatorial X-ray reflections, measured by H.E.Huxley in 1953 [16] and again in 1968 [20]; the intensity of the [1,1] reflection was found to increase relative to the [1,0] when muscle passes into rigor. A similar change was seen in contracting muscle by J.Haselgrove and H.E.Huxley in 1973 [10]. In relaxed muscle a series of layer lines is visible, identified as arising from the myosin crossbridges by G.Elliott [5]. H.E.Huxley, K.Holmes and W.Brown [19] showed in 1965 that these lines become less intense as muscle contracts, while in rigor muscle the myosin lines are reduced in intensity and a pattern of 'decorated' actin lines is visible, indicating that the heads have bound to actin and follow the actin helix repeat. Many detailed studies of the X-ray diffraction pattern have been done, including time-resolved experiments; the only other finding relevant to this thesis is that the crossbridges move out towards the

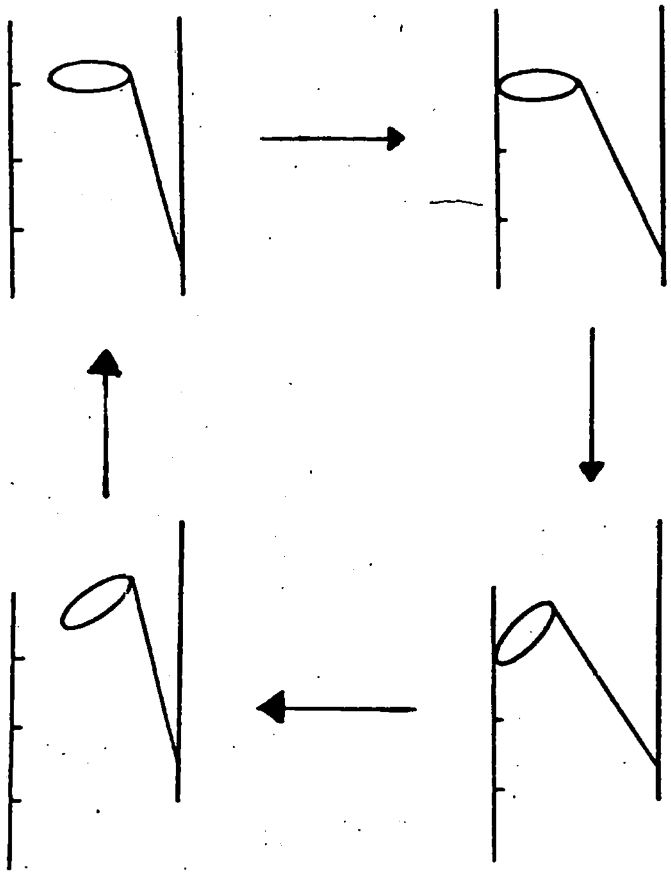


Figure 1.2 A diagram showing the postulated crossbridge cycle. The crossbridge moves out from the myosin filament and attaches to actin. The myosin head goes through an angle change, causing the filaments to slide past each other, and then detaches and returns to its original angle, ready for another cycle.

actin filament ahead of tension rise, by H.E.Huxley in 1975 [22].

Information about the orientations of the myosin heads can also be gained from flourescent probe measurements. The probe is a molecule which emits polarised light at a particular angle. It binds to S1 at a fixed angle, and the orientation of the probe can be measured from the polarization of the emitted light. Paramagnetic probes are used in the same way, their orientation relative to an external magnetic field being measured, and the two methods give similar information. In 1980 D.Thomas and R.Cook [36] measured the orientation of paramagnetic probes on myosin heads in relaxed and rigor glycerinated rabbit psoas muscle. They found that in full-overlap rigor, virtually all the heads had the same well defined orientation, while in relaxed muscle, the heads had random orientation. In stretched rigor muscle, it appeared that those heads unable to interact with actin have random orientation, while those in the region of overlap between the actin and myosin filaments had the same orientation as in full overlap rigor. In 1982 R.Cooke et al. [4] found using the same technique that in isometrically contracting rabbit psoas muscle 80% of the probes have random orientation, as in relaxed muscle, while the remaining 20% had the same well defined orientation as that seen in rigor muscle.

Another method of measuring the myosin head orientation was developed by M.Irving. He argued that crossbridges in muscle contribute to the birefringence in the A-bands, and measured differing degrees of birefringence in different muscle states. Crossbridges in resting muscle seem to be aligned preferentially with the fibre axis, while in rigor the bridges are at an angle of about 50° to the fibre axis [25]. Transients on release of caged ATP give details of the crossbridge behavior as rigor muscle relaxes. These results relate to the work on diffuse scatter from myosin heads.

The decorated actin lines from rigor muscle, described by H.E.Huxley and W.Brown [18], and its characteristic stiffness indicate that a large proportion, if not all, of the bridges are attached to actin in these conditions. The tension responses to rapidly imposed length changes may give information about the details of the crossbridge cycle. A.F.Huxley and

R.Simmons [15] measured the variation in muscle tension over time in tetanically contracting muscle after the imposition of step length changes of various magnitudes and in both directions, and deduced a 12nm working stroke length; they also deduced that the crossbridge contains an elastic element, which is responsible for the instantaneous response to a length change. In 1981, L.Ford et al. [6] measured the variation in the stiffness of contracting muscle with the degree of filament overlap, and concluded that the stiffness can be used to measure the number of attached crossbridges. They found that a quick shortening of 3–4nm brings the tension to zero. This may represent the extension of the elastic element, or the length of the power-generating part of the crossbridge cycle [23].

In summary, a few of the more relevant advances in our knowledge of the contraction of muscle have been described. Much of this and other evidence was summarized and interpreted on a single model by H.E.Huxley and M.Kress in 1985 [23]. However, although much progress has been made, the force generating stage of the crossbridge cycle has not been identified and much remains to be learned about the myosin heads, which have been shown to play such an important part in muscular contraction.

1.3 The diffuse X-ray scatter from myosin heads

In 1983, J.Lowy and F.Poulsen [27] showed for the first time that information about the myosin heads in striated muscle was contained in the region of diffuse scatter in the centre of the diffraction pattern from intact muscle, following on from their measurements of diffuse scatter from smooth muscle in 1982 [26] (described later in this section). Before J.Lowy and F.Poulsen's work, all X-ray studies of intact muscle had concentrated on the sharp reflections arising from well ordered structures.

The theory of X-ray scatter was studied by A.Guinier in the 1930's, and is reviewed in his book [9]. At low angles, the Guinier approximation for low concentration solution scatterers holds, so the Guinier plot—the logarithm of the intensity against the square of the distance from the pattern centre—is linear, and the gradient depends on the radius of gyration (R_g) of the scattering body. The angle to which the Guinier approximation holds depends on the shape and size of the scattering body. The shape of the outer part of the scatter depends on the shape of the scattering body. The Guinier approximation will be discussed in more detail in Chapter 3.

In 1978, K.Kretzchmar, R.Mendelson and M.Morales [30] measured the X-ray scatter from purified S1 in solution. The outer part of the scatter was used to test simple models for S1, by comparing the measured scatter with that calculated from the model. The plot was found to be linear to 360mrad^2, equivalent to a Bragg spacing of about 12nm. The radius of gyration of S1 was found to be 3.24 ± 0.03nm. In 1980, R.Mendelson and K.Kretzchmar [31] performed similar experiments, finding that the R_g of S1 was 3.28±0.06nm, the volume was 151 ± 6nm^3, the surface area was 330 ± 15nm^2 and the maximum chord was 12.0 ± 1.0nm. They tested more complex models for S1 shape, using the outer part of the scatter, and found that a model derived from electron microscopy by J.Seymour and E.O'Brien (quoted as a personal communication) on S1 decorated actin gave the best approximation. This implied that binding to actin did not significantly change the shape of S1, since a model based on bound actin agreed with measurements on S1 in solution.

In relaxed and contracting muscle, some of the myosin heads are disordered and so do not contribute to the sharp reflections. These heads must contribute to the diffuse scatter. In 1982 J.Lowy and F.Poulsen [26] showed that the diffuse scatter in patterns from the anterior byssus retractor muscle of Mytilus edulis (ABRM) was circularly symmetric, and that since the Guinier plot was linear, it arose from a single type of scattering body which obeyed the Guinier approximation. The region examined corresponded to Bragg spacings of about 22nm to 45nm, and the scattering body was identified as a dimer of myosin heads. R_g values of 8–16nm were measured from the Guinier plots. Changes in the disc of diffuse scatter and also in an equatorial streak during contraction were interpreted as showing that myosin heads move out from the thick to the thin filaments on stimulation, ahead of tension rise, and return on relaxation after a delay.

In 1983, J.Lowy and F.Poulsen[27] studied the scatter in striated frog muscle, and showed that the shape of the diffuse scatter in relaxed and non-overlap rigor arose from disordered objects scattering as if in solution, and corresponded with that found for S1 in solution. The scatter in full overlap rigor was decreased in intensity to 30% of that seen in non-overlap,and was compressed in the diagonal directions, indicating that the scattering bodies had a preferred orientation of about 45° to the filament axis. The remaining intensity was thought not to arise from disordered heads but from substitution disorder, as there are more actin sites than myosin heads, so the empty sites act as oriented but spatially disordered 'holes' and contribute to the diffuse scatter. They demonstrated by comparison between the different states that at least 85% of the scatter arises from S1, and argued that the scatter from the soluble proteins is reduced and broadened by interference between particles, so that it is negligable in the region of interest. Two populations of heads were postulated: an ordered population, contributing to the sharp reflections, and a disordered population with a r.m.s. displacement of at least 9nm from the mean positions, contributing to the scatter. The R_g value for the myosin heads was found from the Guinier plot to be 3.1nm, for an angle range corresponding to Bragg spacings 5–14nm, where the Guinier plot is linear.

A final demonstration that the diffuse scatter does arise from S1 came from measurement of the scatter from synthetic myosin filaments prepared by P.Cooke et al.[2], which gave an X-ray diffraction pattern containing a meridional reflection at 14.4nm, which was attributed to the ordered backbone, and a region of diffuse scatter. Measurements of this diffuse scatter by F.Poulsen et al.[33] agreed very well with the solution scatter measured by R.Mendelson et al., except for a contribution on the equator. The scatter was nearly circularly symmetric, but a slight compression in the axial direction indicated a preferred orientation of the myosin heads in this direction. Comparison with taenia coli muscle of guinea pig showed that 75% of the diffuse scatter in this muscle comes from S1. The 14.4nm meridional reflection was used as calibration, as it was argued that the intensity of that reflection is proportional to the number of myosin molecules contributing to it. However, some myosin in taenia coli may be in solution, in which case it will not contribute to the 14.4nm reflection and the calibration will be too low. The value of 75% is therefore a minimum. This paper demonstrated that scatter corresponding to that from S1 in solution is seen in patterns from synthetic threads where only myosin is present. It is therefore reasonable to identify this scatter in the myosin thread patterns as arising from S1 behaving as if in solution, and to suppose that S1 in muscle may also scatter in this way.

In 1987 J.Lowy and F.Poulsen measured the scatter from relaxed and contracting frog striated muscles [28]. They found that the diffuse scatter in relaxed muscle was compressed in the meridional direction, indicating a preferred orientation of the heads parallel to the filament axis. In isometrically contracting muscle there was an increase of 10% in the scatter intensity, indicating an increase of 30% in the numbers of disordered heads, and a compression in the equatorial direction, indicating a preferred orientation perpendicular to the fibre axis. In auxotonically contracting muscle the scatter was circularly symmetric, showing random orientation of the heads. This implies that heads in actively shortening muscle go through a range of orientations, giving an average that is circularly symmetric, and offers some support for the idea that heads cause shortening by a change in orientation while attached to actin. Muscle resting in the 3 minute period between contractions showed random orientation of the myosin heads, differing from normal relaxed muscle, but this finding was later withdrawn in 1990 [29], being attributed to a fault in the method of analysis.

J.Lowy and F.Poulsen also wrote a review of X-ray work in 1987 [35], in which they mentioned new measurements they had made from Guinier plots of contracting ABRM. In isometric contractions of muscle treated with 5-HT, which destroys the muscles ability for tonic contraction, there was a 35% decrease in the number of disordered heads, as compared with relaxed muscle. Without 5-HT, an increase was seen. There was no change in the

R_g, indicating that the separation of the heads in a pair remained constant in contraction.

The scatter from contracting muscle was studied further in 1990, in time-resolved experiments [29]. These confirmed the previous findings, and added the information that the change in the diffuse scatter on stimulation leads the tension development by 20ms; the first meridional layer line intensity decreases on stimulation, leading tension by slightly more than the scatter. No change was seen in quick release, either in the layer line or the scatter—possibly because the resolution was insufficient. These findings were interpreted as confirmation that the scatter contains information about the myosin heads, and that there are two populations of myosin heads in muscle—ordered and disordered.

1.4 The aims of the project

The work described in this thesis derives from J.Lowy and F.Poulsen's work on the diffuse scatter in muscle, described in the previous section. At the same time, some measurements are made of the sharp reflections in the same patterns as are used for the diffuse scatter measurements. The theoretical basis for analysis of the diffuse scatter from muscle will be discussed, and the equipment and data processing capability available at the Oxford Research Unit will be described and evaluated. A method of analysis will be developed, aimed at increasing the precision and objectivity of the data processing, reducing the errors, and improving the speed so that more data can be processed. Having developed the method and the necessary computer software, it will be used to process data from experiments to be done at Daresbury Laboratory with frog striated muscles in various conditions, and also on data from experiments done by J.Lowy and F.Poulsen in 1977 on living frog striated muscles in conditions of varying sarcomere length and/or osmolarity. This data processing will have the dual purposes of testing and refining the processing methods and obtaining information, particularly about the applicability of the Guinier approximation, and about the usefulness of re-analysing old X-ray patterns, taken before the importance of the diffuse scatter was realized.

The precise objectives of this project are thus as follows.

1. To set the analysis of the diffuse scatter of a firm footing, eliminating processing errors as far as possible, and to evaluate the use of the Guinier approximation, applicable to solution scatter at low concentration, for the analysis of scatter from S1 in intact muscle.

2. To increase the rate of processing and thus the throughput of both film and detector data at the Oxford Research Unit, by developing a package for computer processing of diffraction data.

3. To study the diffuse scatter, using the improved analysis methods.

4. To study the sharp reflections using the same programs.

5. In particular, to consider the conditions of data acquisition, to see whether it is sensible to recover information from patterns already on file, but taken before interest was directed at the diffuse scatter from S1.

Chapter 2

The Equipment

The analysis of a pattern is dependent on the equipment for measuring the intensity and processing the data. This chapter describes the equipment available for densitometry of film patterns, and the computers that can be used for processing the data.

2.1 Computers

There is a PDP 11/34 computer available at this Unit for data processing, and also access to the VAX cluster at the main site. At the start of the project there was no link between these machines, but it is now possible to transfer data from one to the other, and thus to use programs on both machines for data treatment. The VAX is very much more powerful than the PDP 11, but can be slower to use if a large number of people are using it, as it operates on a time-share basis. The PDP 11 is connected to a Tektronix graphics screen and a graph plotter, so that writing programs to view and plot the data is straightforward; in addition, the graphics screen has a 'crosswire' facility, so that points on a plot may be selected by the user and their co-ordinates read onto a computer—the PDP 11. Plots can be obtained from the VAX, but since it is a multi-user system, they may not be plotted immediately, which can be inconvenient at times. The PDP 11 is adequate for data processing, though memory limitations occasionally mean that operations cannot be carried out in the most time-efficient manner. In addition, the LKB densitometer (see Section 2.2) could only be connected to the PDP 11. That machine was therefore the most convenient machine for data processing.

2.2 Densitometers

An ideal densitometer will divide a film into a grid of the appropriate resolution, and produce measurements of the average exposure of each grid area. This can be achieved by scanning with a beam of light, and recording the transmitted intensity at each grid point. The intensity may vary rapidly from one grid point to the next, and the densitometer should be able to respond to these changes. The densitometer should also be stable, and not contribute to the measured intensity.

A problem can arise when averaging the intensity of a grid area within which the intensity varies. This is because the transmitted light is th log of the exposure, and the average of a log is not the log af the average. In this thesis, the grid size was small compared with the size of features of the x-ray pattern, so that this problem was ignored.

There are two densitometers available at Foxcombe Hall; an elderly Joyce-Loebl flat plate densitometer, and a more modern LKB. A more sophisticated Joyce-Loebl rotating drum densitometer at Daresbury Laboratory was used for the analysis of the sharp reflections.

The Joyce-Loebl at densitometer Foxcombe Hall produces intensity plots of a line across a film. The film is mounted on a glass plate, which can be translated and rotated in a horizontal plane to adjust the film position. The glass plate is in turn mounted on a base, which can move in a straight line on tracks, without rotating, and to which the glass plate may be clamped when correctly adjusted. Paper for recording the film pattern is put on a platform, which is connected to the base supporting the glass plate by a metal rod, pivoted about one of a number of points on its length. The platform is moved by a motor through a distance equivalent to the length of a sheet of A4 paper, moving the film, with the metal lever, through a distance dependent on the pivot position—thus the distance scale of the intensity tracing can be varied by moving the pivot. A beam of light, whose width can be varied with an adjustable slit, passes through the film and through an optical wedge above the paper, and is then compared with a reference beam. The optical wedge is moved in a direction perpendicular to the movement of the paper, until the intensities of the two beams match. A pen is connected to the optical wedge, so that a tracing of the intensity is drawn on the paper. The intensity scale can be changed by mounting a different optical wedge.

The Joyce-Loebl densitometer responds well to step changes in intensity, and gives an accurate record. Unfortunately, to take a series of parallel scans across a film, it is necessary to adjust the film position by hand after each scan, making scanning an area a laborious and time-consuming process. The sideways adjustment of the film is not sufficiently precise to make scans of sufficient resolution—a few hundred microns—and the mechanism is not up to the wear and tear of taking some hundred scans across each of many films. The output is normally analogue, but there is an analogue to digital converter on the PDP 11, enabling the output of the densitometer to be read onto the computer in digital form. This was attempted by Finn Poulsen and Jack Lowy before the start of this project, but the results were unsatisfactory for the reasons given above.

The LKB densitometer uses a helium-neon laser to produce a spot of $100\mu m$ at the film, which is mounted on a ground glass plate. The laser beam intensity is compared with a reference beam as the film is moved, so that the pattern is scanned from front to back—the y-direction on the densitometer plate. The densitometer can be set up to do 2-dimensional scans, in which case the film is moved to the left after each scan, so that a 2-dimensional picture is built up. The output can go to a printer or to an EIA 232 (RS 232) port, so that the data is very readily transferred to a computer.

The PDP 11 computer was linked to the LKB densitometer via a cable. The two machines are in different buildings, and because of available space and sharing of the densitometer, neither could be moved. The cable therefore runs between the two buildings. The densitometer cannot be controlled by the computer, and except for x-on and x-off there is no 'handshaking'—a densitometer run cannot be started from the computer, and if the computer fails during a run, it cannot be stopped from the computer either. This problem has been overcome with a 'button-pusher', built by our electronics technician, Alan Knight, which recognizes certain characters sent from the computer to the densitometer, and which will press the appropriate buttons on the densitometer to start or abort a run if the densitometer is set up to run and in the correct mode. The characters chosen to operate the 'button-pusher' are unrecognized by the densitometer.

The LKB densitometer specifications imply that the machine will measure sharp step changes in intensity accurately, and thus allow comparison of the layer line intensity in different muscle states. They also imply that the drift is small enough to be neglected, and so the scatter may be accurately measured. The LKB densitometer was compared with the Joyce Loebl, using both film patterns and test plates with step intensity changes (see Figures 2.1, 2.2 and 2.3). Neither of the above specifications was met by the LKB. Step changes in intensity are not followed, giving errors of a factor of two in the area under the layer lines, and the results are not repeatable, so that the densitometer is of no use for this kind of measurement. The Joyce Loebl responds well to step changes; it often overshoots slightly on large steps, but recovers quickly, so that the area is not much affected. The drift of the LKB is small enough to be neglected over the course of a day, provided that the densitometer is allowed an hour to 'warm up' after switching on, but the background intensity is not constant over the scanning area, so that a scan of the densitometer plate must be taken on the same day that the film is scanned, and subtracted from the film data. If these precautions are taken, the LKB gives a good representation of slowly-varying parts of the film pattern—i.e. the scatter. It can do two dimensional scans with a spot size and hence resolution of 100μm, and can send the results to a computer, and so is more useful than the Joyce Loebl, which, though a better machine, can only do one dimensional scans, with the output on paper.

The other densitometer used was a Joyce-Loebl Scandig 3 rotating-drum densitometer at Daresbury Laboratory. The film is mounted on a glass drum, and the beam from a helium-neon laser is shone through the film onto a detector inside the drum. When the film is scanned, the drum is rotated, while the laser beam and detector are moved in a direction parallel to the axis of rotation, thus giving a series of parallel scans through the film. The resulting two-dimensional scan of the film is written to magnetic tape, which can be read onto the VAX at Walton Hall, and thence transferred to the PDP 11. This densitometer responds well to step changes in intensity (see Figure 2.4), and is thus suitable for measurement of the sharp reflections; this measurement cannot be done adequately by either of the densitometers available at Foxcombe Hall.

2.3 Summary

There are two densitometers at Foxcombe Hall, a Joyce Loebl and an LKB. Of these, the Joyce Loebl is the better machine, but is not suitable for 2-dimensional scans. The LKB is inaccurate when measuring rapid changes in intensity, for instance, the areas under the layer lines, but is able to measure the scatter. Therefore the LKB densitometer was used for measuring films. The LKB was connected to the PDP 11 computer, and programs developed for data analysis before the link to the VAX was set up. In addition, writing programs to plot and view the data is simpler on the PDP 11 since it is a single-user machine. The PDP 11 is therefore used for the majority of the data analysis. Subsequently, the PDP 11 was linked to the VAX, so that data can be transferred to the VAX for further processing.

For analysis of the sharp reflections, the Joyce-Loebl rotating-drum densitometer at Daresbury Laboratory was used, and the data transferred to the PDP 11 for processing.

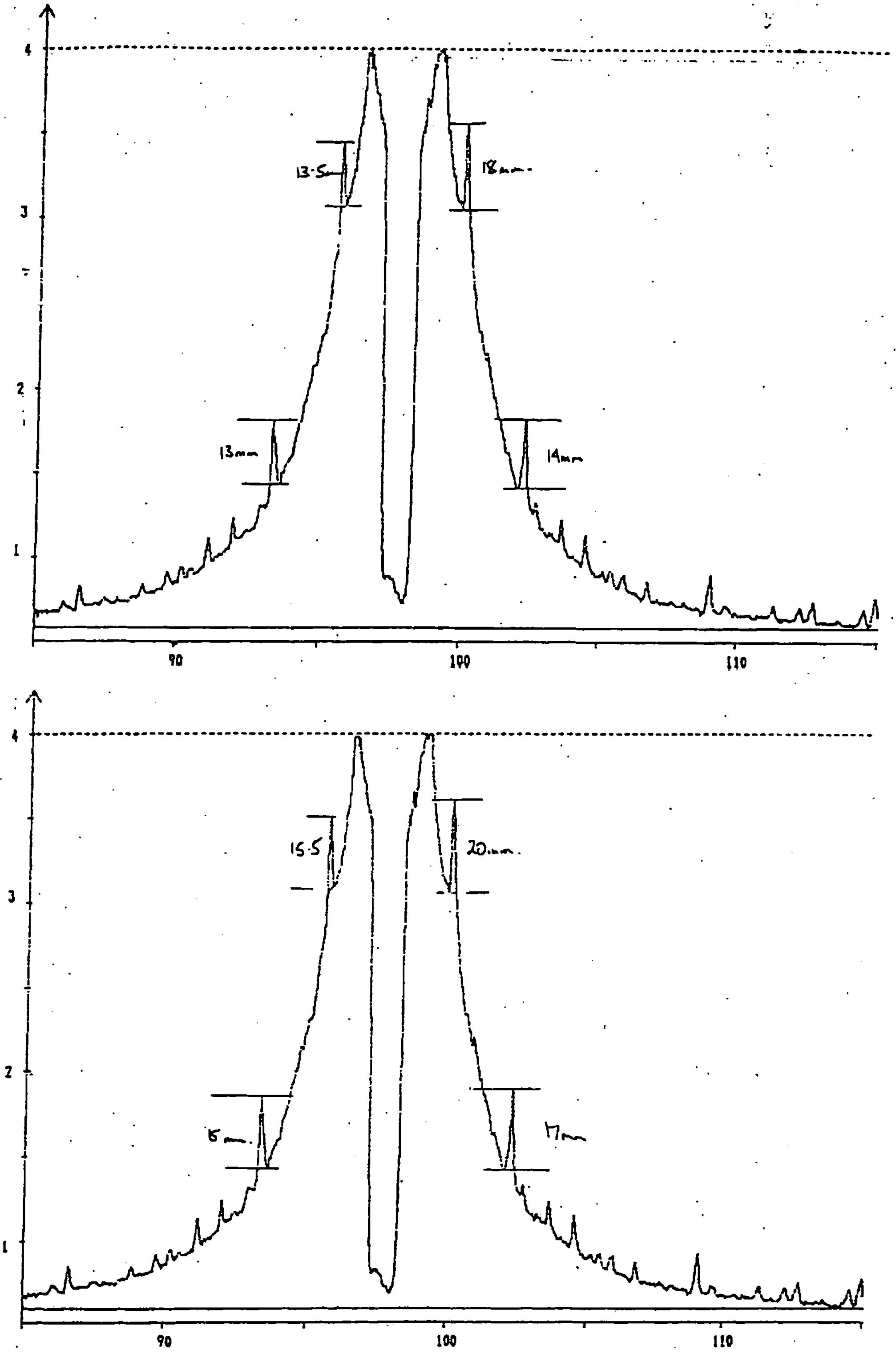
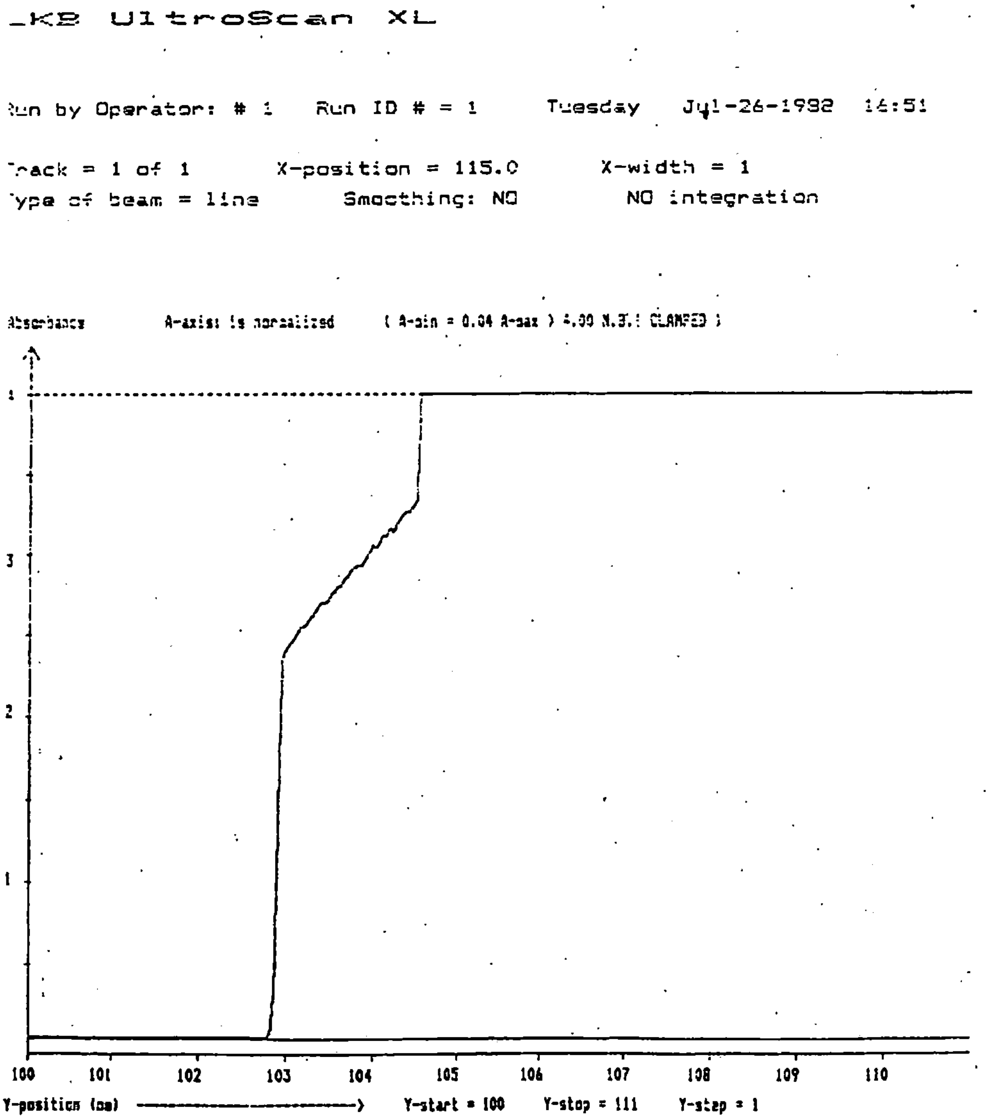


Figure 2.1 Two scans of the same part of the same film, measured by the LKB densitometer. The film was not moved between scans. Note the difference in peak heights between the two scans. The units of the horizontal axis are mm, with Absorbance Units (A.U.) on the vertical.



```
LKB UltroScan XL

Run by Operator: # 1   Run ID # = 1      Tuesday   Jul-26-1992  16:51

Track = 1 of 1        X-position = 115.0       X-width = 1
Type of beam = line       Smoothing: NO          NO integration
```

```
Absorbance          A-axis: is normalized      ( A-min = 0.04 A-max ) 4.00 N.B.: CLAMPED )
```

```
Y-position (mm) ------------------------>    Y-start = 100    Y-stop = 111    Y-step = 1
```

Figure 2.2 A scan of a step intensity change, done on the LKB densitometer. A piece of black paper was used. Note the slow response to the step change. Horizontal axis in mm, vertical in A.U.

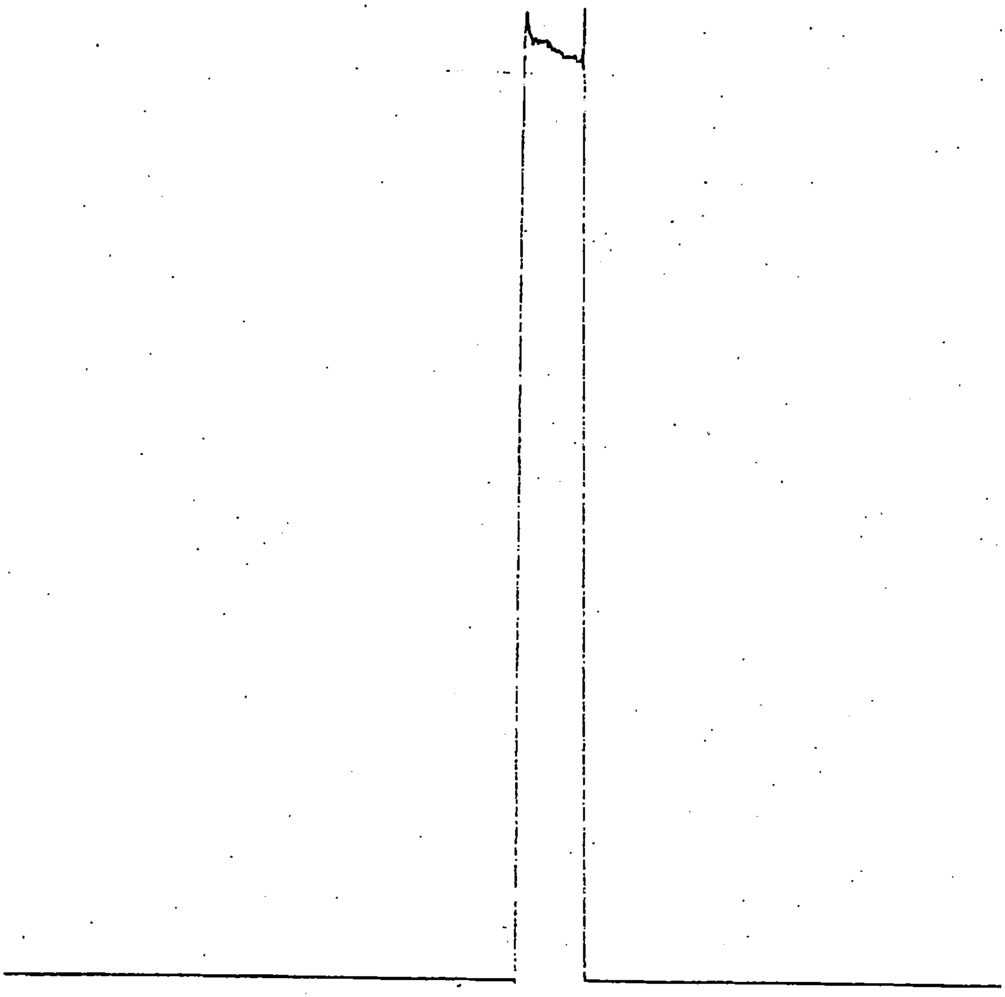


Figure 2.3 A scan of a step intensity change done on the Joyce-Loebl at Foxcombe Hall. A piece of blackened film 6.5mm wide was used. The output of this densitometer has been scanned from right to left, unlike plots from the other two densitometers. Note the rapid response the the step change, with only a small overshoot close to the edge. Absorbance increases up the vertical axis, in arbitrary units, and the horizontal axis measures distance.

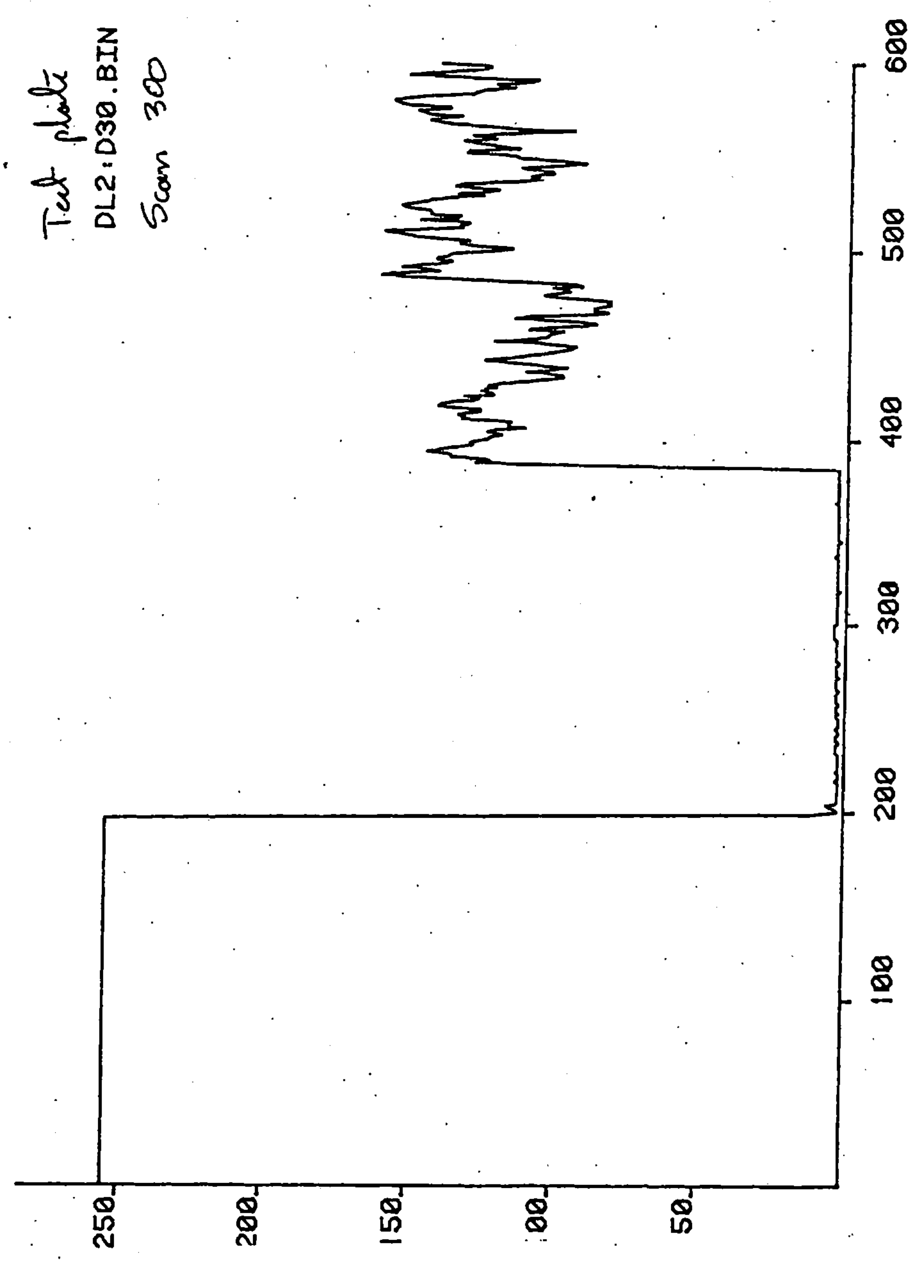


Figure 2.4 A scan of a step intensity change on the Joyce-Loebl at Daresbury Laboratory. On the left was a piece of masking tape; on the right a piece of white paper tape. Each unit on the x-axis represents 50μm, while 255 on the vertical axis represents 3 A.U. Note the rapid response to the step.

Chapter 3

Analysis of the Diffuse Scatter

In this chapter the theory and practice of analysing the diffuse scatter will be discussed. The validity of the Guinier approximation for S1 in muscle will be evaluated, and the analysis methods available at the Oxford Research Unit at the start of this project will be described.

3.1 The validity of the Guinier approximation

Myosin heads in solution at low concentration act as independent scattering bodies. The X-ray diffraction pattern is therefore the sum of intensities of the patterns from each individual head, which is the spherical average of the molecular transform of the myosin head. At small angles, the Guinier approximation holds:

$$\text{Intensity}, i(h) = exp\left(\frac{-h^2 {R_g}^2}{3}\right)$$

Where $h = \frac{4\pi}{\lambda} sin\left(\frac{\theta}{2}\right)$ and R_g is the radius of gyration of the myosin head [9]. Using the approximation $sin(\theta) = \theta$, valid for small θ, and using Bragg's Law to give the distance in reciprocal space $s = \frac{sin(\theta)}{\lambda} = \frac{\theta}{\lambda}$, the above equation can be re-written:

$$i(s) = exp\left(\frac{-4\pi^2 s^2 {R_g}^2}{3}\right)$$

If the data is plotted in the form $\log_{10}(i)$ against s^2 (a Guinier plot), the result will be a straight line over the Guinier region where the approximation holds. The gradient, α, of the Guinier plot is given by:

$$\alpha = \frac{-4\pi^2}{3\ln(10)} {R_g}^2$$

Thus the radius of gyration of the myosin heads can be measured from the gradient α of the Guinier plot of the diffraction intensity in the Guinier region by:

$$R_g = \frac{1}{\pi}\sqrt{\frac{3\ln(10)}{4}}\sqrt{-\alpha}$$

or:

$$R_g - 0.418\sqrt{-\alpha}$$

This measurement was done for S1 in solution by Kretzchmar, Mendelson and Morales [30] (see Section 1.3). The angle range for which the Guinier approximation holds depends on the size and shape of the scattering body; if the approximation is valid, the Guinier plot will be linear. The plot for S1 in solution was found to be linear up to a Bragg spacing of about 12nm. The value of R_g obtained was 3.24 ± 0.03nm.

In muscle, the ordered myosin heads give rise to sharp reflections. However, it is thought that active myosin heads cycle asynchronously, and may be considered as disordered. Thus the sharp reflections in muscle scatter can give little information about active heads. However, the X-ray diffraction pattern of muscle also contains a central region of diffuse scatter. This scatter was studied by Lowy and Poulsen (see Section 1.3), and found to be of the same shape as that from S1 in solution. It was demonstrated that the scatter can be treated as arising from myosin heads obeying the Guinier approximation as if in solution at low concentration [27]. There are several reasons why this scatter may arise from the myosin heads.

1. Myosin is present in smooth muscle though no layer lines are present, so the heads must give rise to diffuse scatter [33].

2. The scatter from muscle is of a similar shape to that of myosin heads in solution [27].

3. Scatter is visible in patterns from synthetic myosin threads, where other proteins that might scatter are not present [33].

4. The shape of the scatter changes when muscle contracts [26].

There therefore seems no doubt that at least some of the scatter in the X-ray pattern of intact muscle arises from the myosin heads. However, there may also be contributions from other components of the muscle, which would mask the effect of the S1 scatter. The intensity of the scatter from the heads will depend on their overall concentration in muscle, since this determines the number of heads in the X-ray beam. The concentration of myosin molecules in frog striated muscle is 120nmol g^{-1} [1]. There are two heads on each molecule, the mass of one head is 120k. Assuming the density of muscle to be approximately the same as water, 1g cm^{-3}, the concentration of myosin heads is $240 \times 10^{-6} \times 120 \times 10^{+3}$g l^{-1} = 29g l^{-1}. This is above the concentration range 7–18gl^{-1} used by Mendelson et al. for solution scattering experiments [31]—i.e. it is sufficient to give a measurable pattern. J.Lowy and F.Poulsen [27] showed by comparing patterns from frog striated muscles in different states that in that muscle at least 85% of the scatter arises from the myosin heads, while the rest is more slowly varying, so that the scatter in the part of the pattern where the heads contribute most can be treated as scatter from myosin heads alone (see Section 1.3). Their argument depends on the assumption that the differences seen in the scatter from rigor muscle at full and no overlap arise from differences in the myosin heads, and not from changes in any other structures. This agrees with the available evidence, and is probably true at least to a first approximation, so the argument will be accepted for this thesis, unless reasons for discarding it arise.

The Guinier theory applies to solution scatter at low concentration, where the heads act as independent scattering bodies. Two conditions must be met for the theory to apply to muscle:

1. The heads must be free to move and to take up any orientation, so that they behave as if in solution (this condition will be discussed further

later);

2. The heads must not interact with each other, so that they behave as independent scatterers.

The first condition can be met, since the heads are attached to the myosin filament by flexible alpha helical chains, the S2 part of the myosin molecule, which probably allow sufficient freedom of movement for the heads to behave as solution scatterers (the structure is described in the review by C.Bagshaw [1]). However, in muscle the scatter is not always circularly symmetric; so that the heads sometimes have a preferred orientation (J.Lowy and F.Poulsen [27]), and in this case the first condition does not apply. However, the second condition may still apply, and then the total intensity is the sum of the intensities of the individual heads. (This contrasts with the case of ordered scatterers, where the intensity is the square of the sum of the amplitudes of the scattered photons.) The solution scatter is thus simply the number of scattering bodies multiplied by the spherical average of the scatter from one body, which is identical to the spherical average of the scatter from all the bodies. This will apply even if there is a preferred orientation, as long as the scatterers are independent, so that the second condition is sufficient in this case.

For no interaction to occur between heads, the relevant concentration in muscle is not the overall concentration of heads, calculated above, but the local concentration between the filaments, which will be higher. This local concentration varies with sarcomere length as the filament separation varies, but an average value may be estimated. In frog striated muscle, the myosin filaments lie on a hexagonal lattice; the separation of adjacent filaments is 35.7nm at body length, and that of next nearest neighbours is 61.8nm [18]. These are the diagonals of a diamond-shaped unit cell, corresponding to the area occupied by one filament, and this area is equal to half the product of the two diagonals, and is $\frac{35.7 \times 61.8}{2} = 1103.13$ nm^2. The spacing of heads on a filament is 14.3nm, and there may be 2 or more pairs of heads around the filament in this length [1]. So the head concentration is:

$$\begin{aligned}
\text{Concentration} &\geq \frac{4}{14.3 \times 1103.13}\ \text{nm}^{-3} \\
&\geq \frac{120000 \times 1.67 \times 10^{-27} \times 4}{14.3 \times 1103.13}\ \text{kg nm}^{-3} \\
&\geq 53\ \text{mg ml}^{-1}
\end{aligned}$$

This value is well above the range of 3–30gl^{-1} recommended for solution scattering experiments by I.Pilz (O.Glatter and O.Kratky [7], p. 242). However, in muscle the heads are not entirely free, but are held to the myosin filament by flexible 'strings', attached at regular intervals. This allows them sufficient freedom of movement to take up any orientation, and so to behave as solution scatterers, but may prevent them from coming close enough for interference between heads. This might explain why the low concentration approximation appears to hold good in muscle although the local concentration is high. The two heads in each pair are clearly not independent, and there must be an effect arising from head interference. These pairs are both bound to S2, and thus to each other, so their maximum separation is limited, and they might represent a single scattering unit which is larger than one head.

The experimental plot of scatter from frog striated muscle shown in Figure 3.1 shows clear linear regions. The effect of the head dimers can be seen in the centre of the pattern; the Guinier approximation holds in this region, but the scattering body is now the head pair, not the single head. The boundary between the inner and outer regions is at 14.3nm Bragg spacing. Mendelson et al. found that Guinier plots from S1 in solution were linear to 12nm [31]. This means that the Guinier approximation does not apply for scattering angles corresponding to spacings larger than this, so the approximation only holds for individual heads in muscle over a region close to 14.3nm—henceforwards referred to as the Guinier point. However, though the gradient may be taken at a point, the Guinier plot should be linear about this point, and especially at slightly larger angles, if the approximation holds.

The disordered myosin head in muscle may have a preferred orientation due to the interactions with the actin and myosin filaments. This orienta-

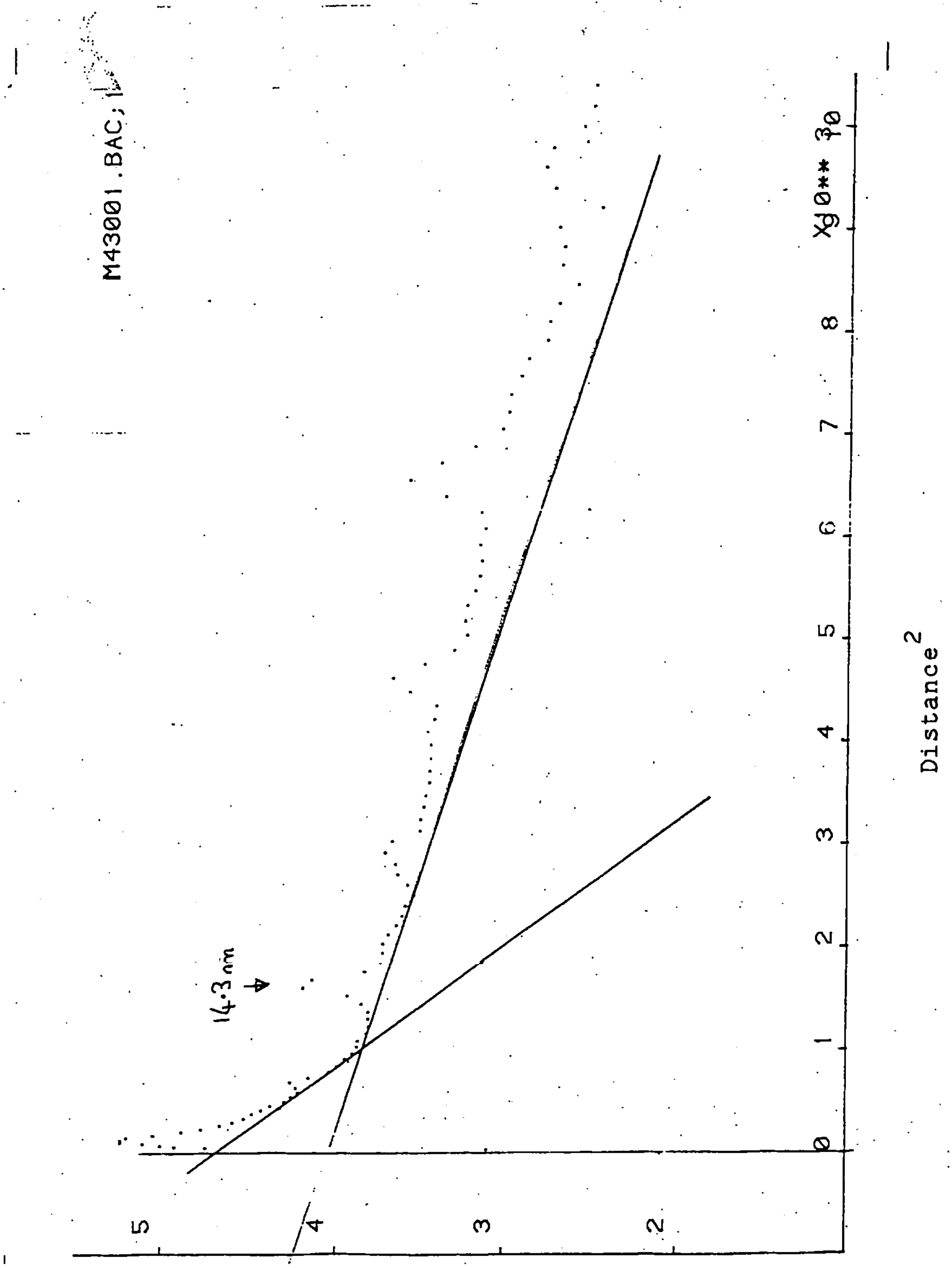


Figure 3.1 A Guinier plot of the meridian of a frog muscle pattern, taken with a 1-dimensional detector at EMBL, Hamburg, showing the inner and outer regions, arising from the myosin head pairs and from the individual heads respectively. By J.Lowy and F.Poulsen (unpublished). The lines were drawn to the curve by J.Lowy and F.Poulsen; there is clearly room for another opinion about their positioning.

tion, if present, would give information about states in the cycle of active heads. If a myosin head lies with its long axis parallel to the filament axis, its contribution to the diffraction pattern will not be circularly symmetric, but will be compressed in the meridional direction. The total scatter is the sum of the contributions of the individual heads, since they scatter independently, so if the heads have a preferred orientation, the scatter from the muscle will be compressed in a direction perpendicular to the average head orientation. This compression can be detected by comparing the meridian and equator with the scatter at 45°, or by examining an isointensity plot of the pattern (a 'contour' plot).

3.2 Data analysis

Two things can be learned about the myosin heads from the scatter in the X-ray diffraction pattern of muscle—the symmetry of the scatter, or the way in which equal intensity lines deviate from a circle, gives information about the orientation of the heads, and the shape of the scatter in a radial direction gives a measure of their radius of gyration, and so of their size. To examine these two aspects of the pattern, contour-type plots and plots of intensity against the distance from the pattern centre are needed. This section outlines the steps involved in analysing films by hand by the method available at the Oxford Research Unit at the start of the project, and evaluates the reliability of this method.

3.2.1 The components of the diffraction pattern

The X-ray diffraction pattern of muscle (see Figure 3.2) has three components:

1. The diffuse scatter, which is the subject of this analysis.

2. Other scatter from muscle, including the sharp reflections and fibre diffraction on the meridian.

3. Contributions from the camera and muscle chamber.

In order to study the diffuse scatter, the pattern must be measured and the sharp reflections and camera contributions removed, leaving only the scatter.

3.2.2 Densitometry of the film

For each film processed, scans were taken parallel to the meridian at 2mm intervals, using the Joyce-Loebl densitometer, over an area extending 12mm either side of the meridian. A distance magnification of 7.5 was used, and beam widths of 0.5mm in the meridional direction and 1mm in the equatorial direction. A smooth line was drawn by hand through each scan, below the peaks (see Figure 3.3).

3.2.3 Data processing

A contour type plot was made from each film, by drawing horizontal lines across each intensity trace, and locating the positions of the intersections of these lines with the intensity traces. These positions were then plotted, and points of equal intensity were joined to give the contour plot(see Figures 3.4 and 3.5).

The points obtained for the contour plots were averaged together, to give a contour plot in one quadrant, and a line was drawn from the centre of this plot at an angle of 45° . The points of intersection of this line with the contours were used to give a plot of intensity against radial distance, which could be compared with the data for S1 in solution. A constant background level may be subtracted, to allow for film fogging, and the intensity values must be scaled to allow for exposure time and beam intensity. Several values for the fogging level and the scale factor were tried in attempts to fit the film patterns to the solution scattering data (see Figures 3.6 and 3.7).

3.2.4 Results

Two films were processed in this way, numbered 1646a and NB16b. These are both exposures of relaxed anterior byssus retractor muscle (ABRM) of

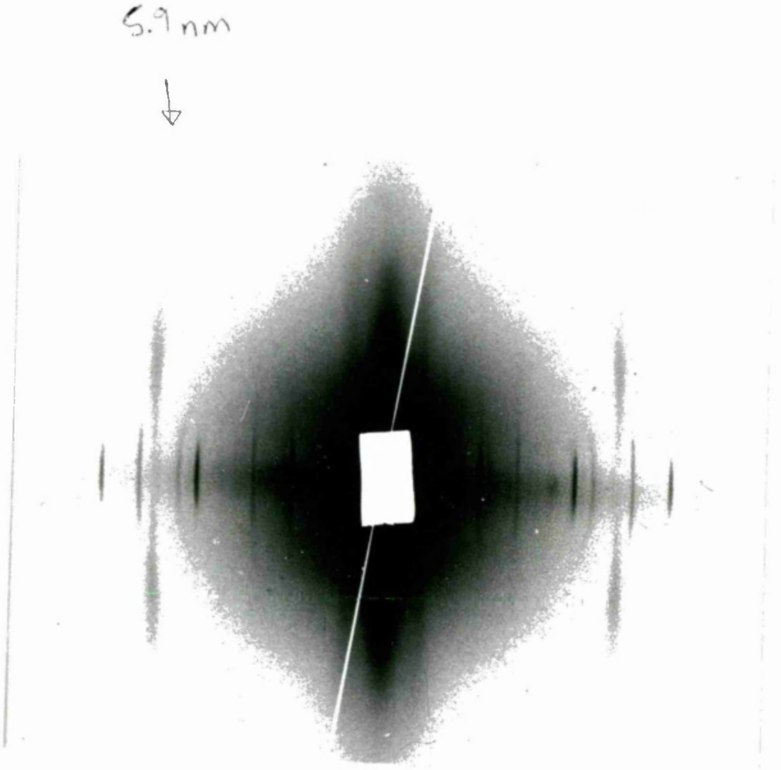


Figure 3.2 A print of an X-ray diffraction pattern of muscle. This is film 1646a, discussed in Section 3.2. The meridian is horizontal.

Mytillus edulis, the edible mussel, taken at EMBL, Hamburg by J. Lowy and F.Poulsen. The symmetry result is not clear—film 1646a (see Figure 3.4) shows a distinct meridional compression, which indicates that the myosin heads have a preferred orientation, more heads being parallel to the muscle axis than perpendicular to it. However, film NB16b (see Figure 3.5) shows circular symmetry, indicating that the heads have random orientation.

Attempts to fit the muscle data to the solution scattering data were not successful for either film (Figures 3.6 and 3.7). This result is also doubtful, both because there was no camera blank subtracted from either film, so that the shape may be distorted, and because, if the heads have a preferred orientation parallel to the fibre axis, as indicated by one film, the data ought to be spherically averaged before the comparison is done. This is because the projection of the heads onto a plane perpendicular to the beam will be larger in this case than if they have a random orientation, giving a narrower and steeper pattern—which is in fact what is seen.

3.3 Conclusions

We have seen that the scatter from S1 in muscle might obey the Guinier approximation even though the heads are bound to the filament backbone and are at a high concentration. That it does so behave cannot be proved without more detailed knowledge of the structures of the proteins present in muscle, and of the forces acting between them, but within our current knowledge such behaviour is conceivable. The empirical demonstration lies in the finding that the Guinier plot of scatter from muscle is linear over the region where the approximation might be expected to hold.

As for the analysis of the scatter, it is clear that the manual processing method described in this chapter is not very satisfactory. The analogue output of the densitometer must be fitted with a smooth line by hand, and measured with a ruler to give an intensity value. This fitting introduces a degree of uncertainty to the data, and this uncertainty is increased when the data points are joined with a hand-drawn line to give a contour plot which is used to make the intensity against radial distance plot. The number of intensity points that can be handled is severely limited, so precision is poor, and the whole process is very slow, so that not many films may be processed. These problems might be largely overcome if the data were read directly onto a computer and processed digitally, as the number of points that could be handled would be vastly increased, and the manual fitting could be replaced by a more objective least-squares method. It would also be relatively easy to plot the data in other forms—for instance, a Guinier plot, from which the radius of gyration of the heads could be measured, and which is impractical at present.

Thus the next step in the project is to write programs to read the data onto a computer, and to analyse it in a more detailed and objective manner, using many more data points to describe the intensity distribution, and least-squares fits rather than hand drawn lines where data must be approximated.

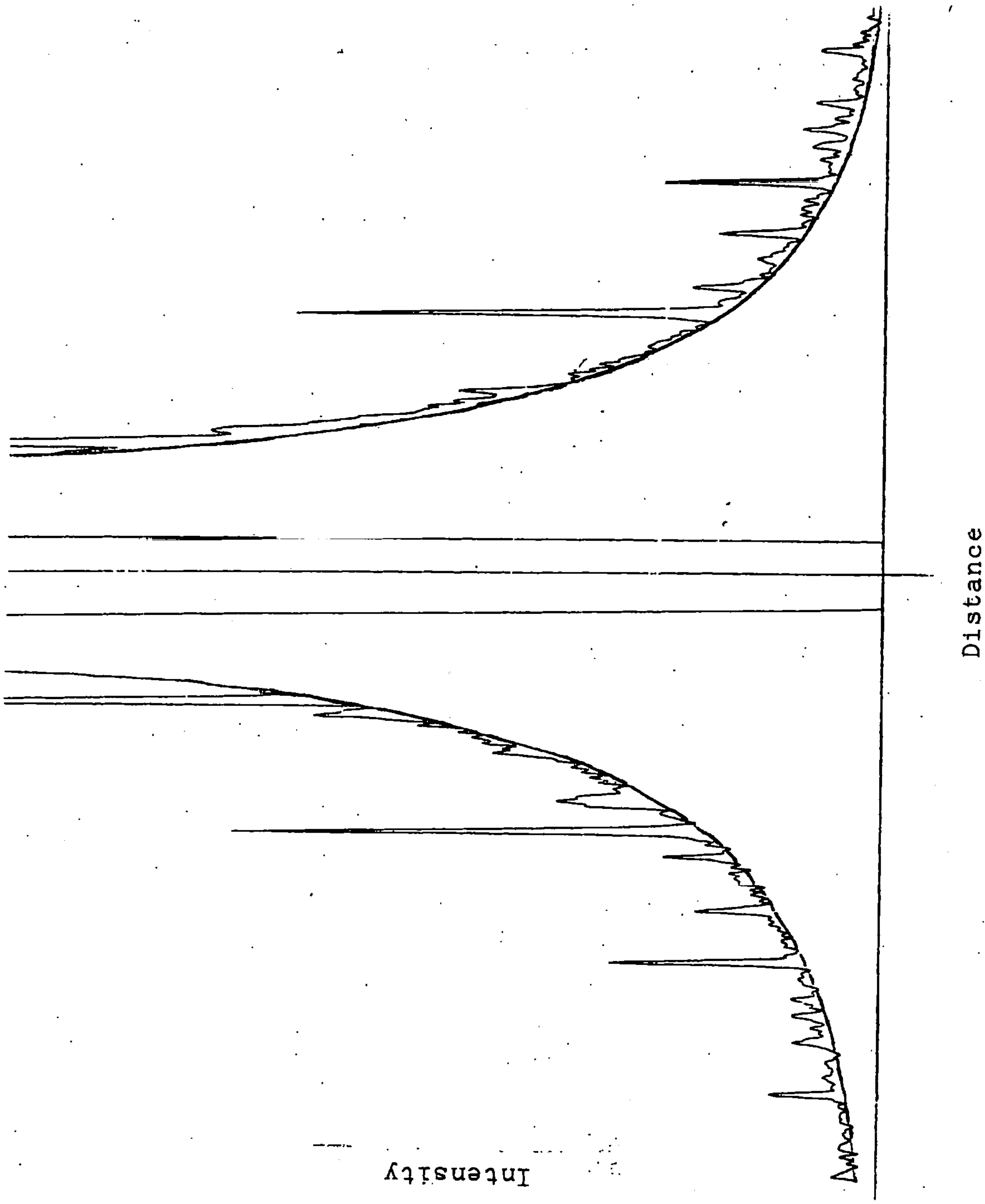


Figure 3.3 A scan through the meridian of film 1646a, with a smooth line drawn beneath the peaks.

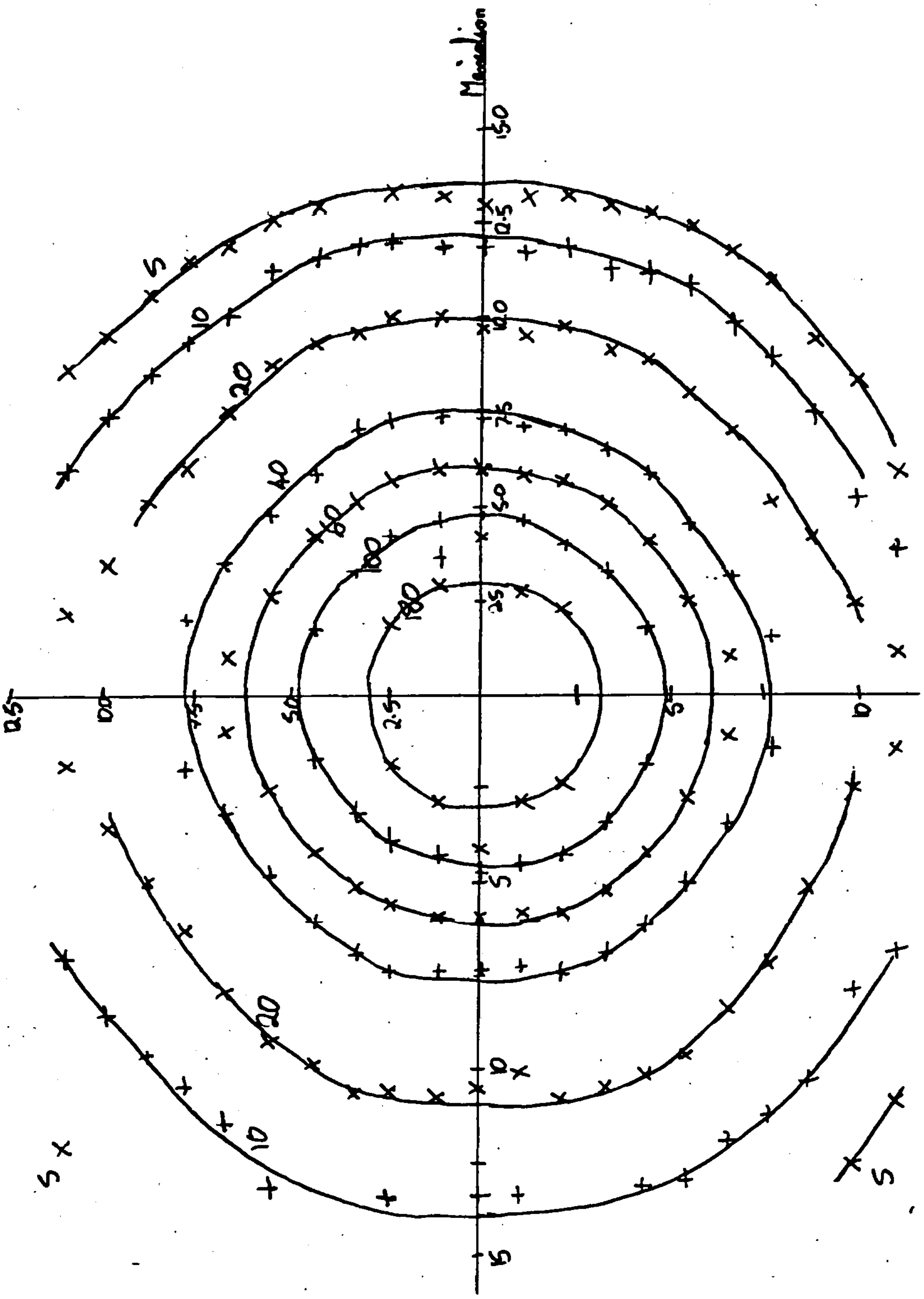


Figure 3.4 A contour plot of the intensity distribution of film 1646a, relaxed ABRM.

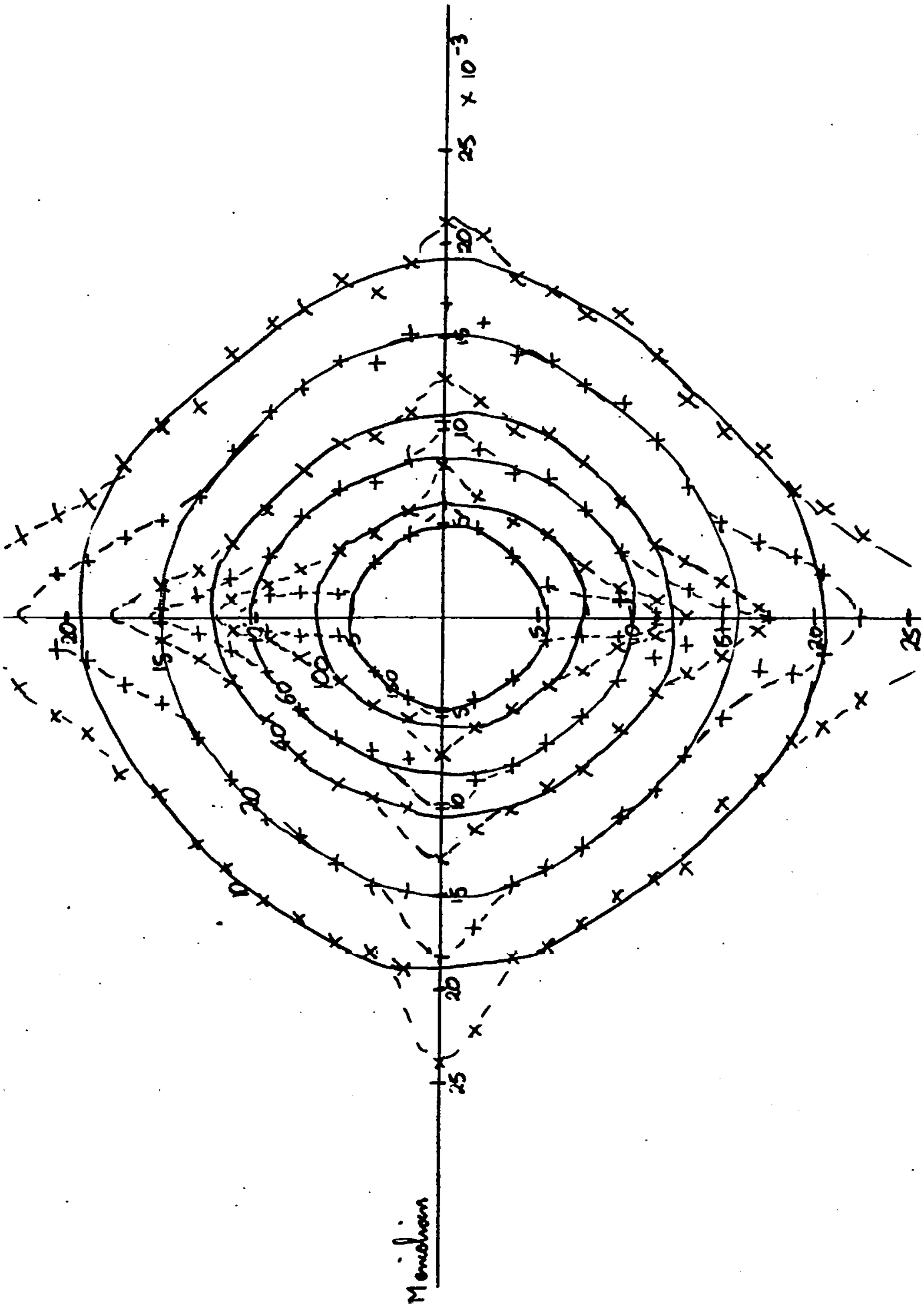


Figure 3.5 A contour plot of the intensity distribution of the film NB16b, relaxed ABRM.

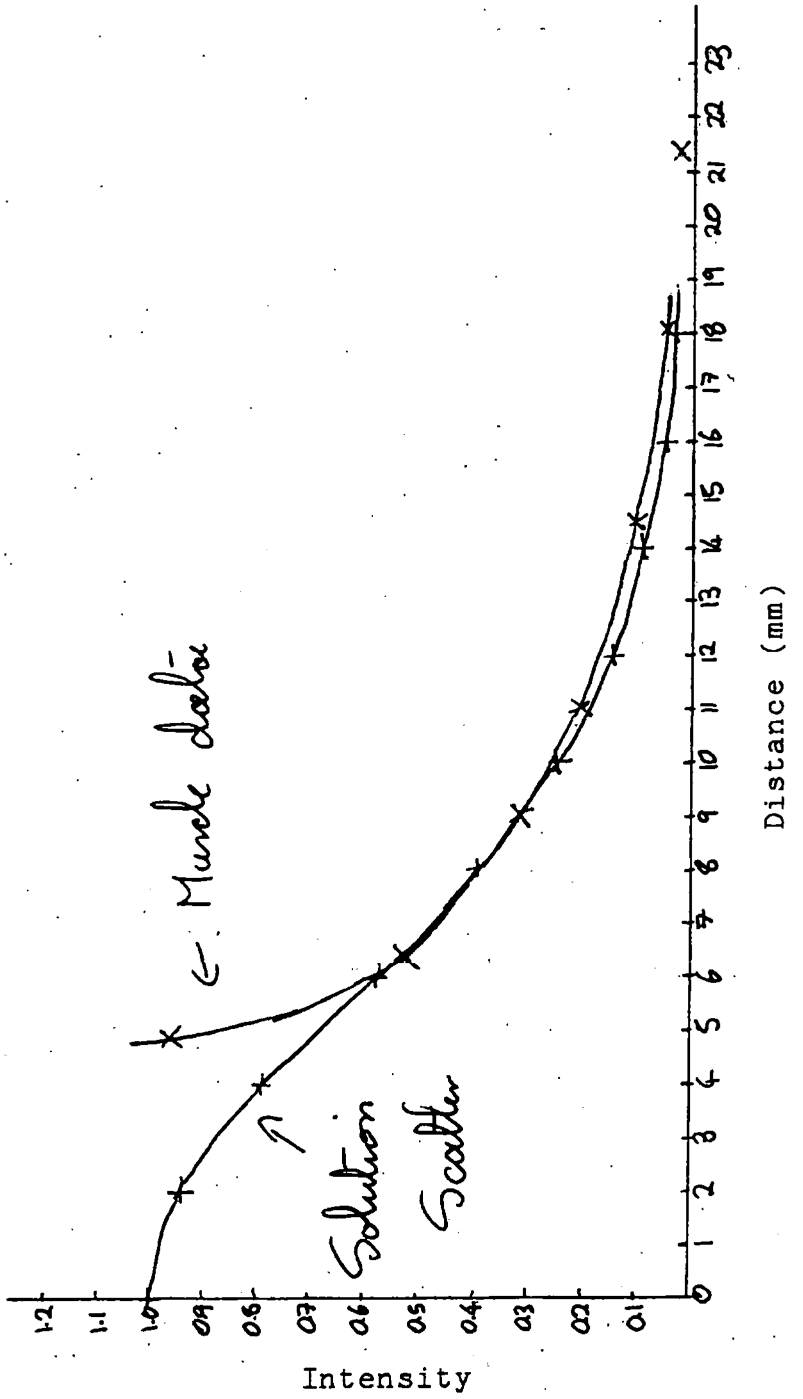


Figure 3.6 Attempts to fit the scatter from film 1646a to the solution scattering data.

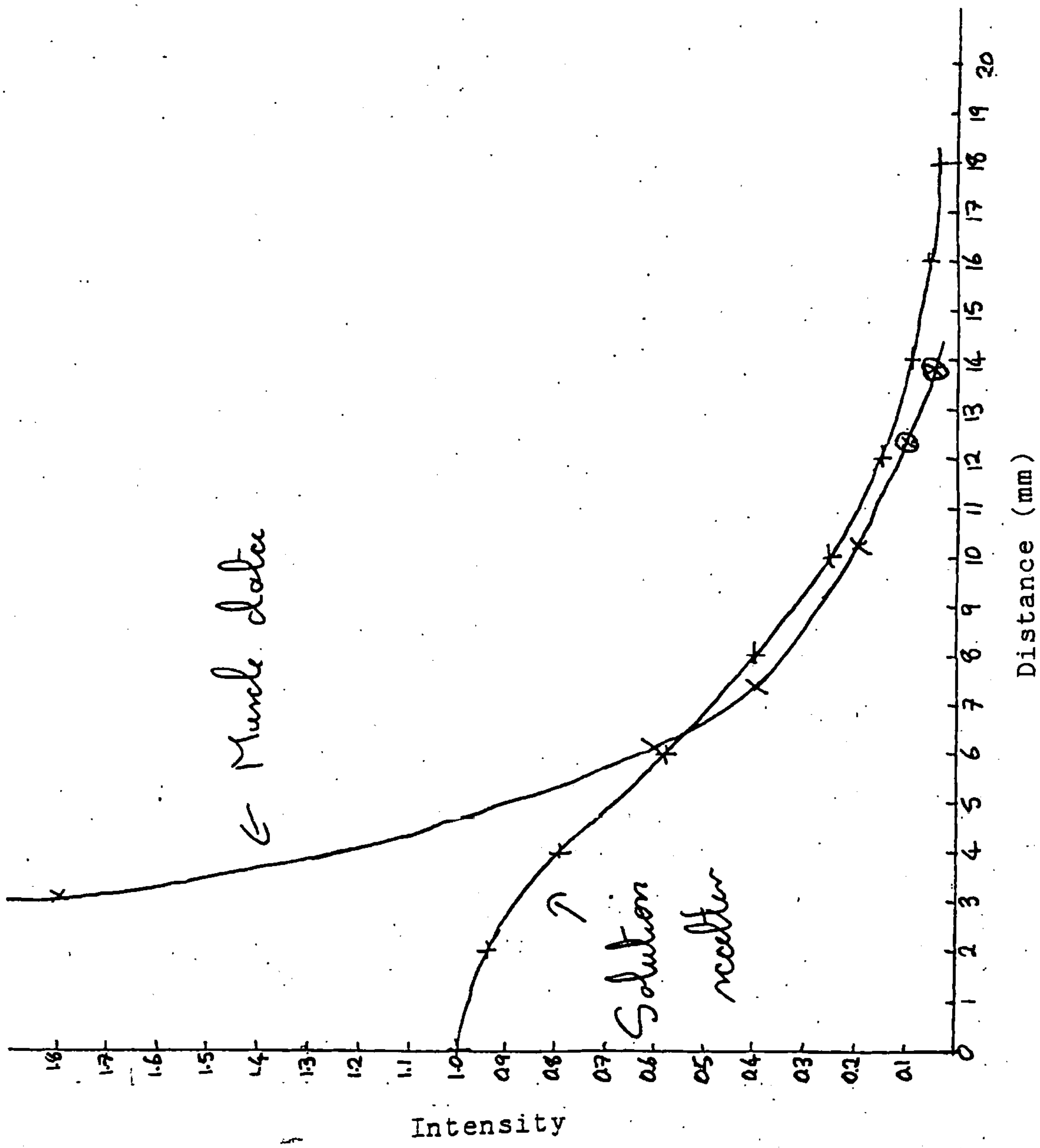


Figure 3.7 Attempts to fit the scatter from film NB16b to the solution scattering data.

Chapter 4

The Computer Analysis of Film Patterns

The next step in the project was to link up the LKB densitometer, which can output data via an EIA 232 (RS 232) port, to the PDP 11 and to write programs to read in and analyse the data. Since these programs were written, certain programs from EMBL, Hamburg, have become available on the OU VAX. These programs were developed for analysis of X-ray data, but with the emphasis on studying the sharp reflections rather than the diffuse scatter. I have not used the Hamburg programs myself, but understand that although they form a more comprehensive and versatile data processing package, the programs described in this thesis have some advantages. The Hamburg programs are used for a wide range of applications to diffraction data, and are thus far more complicated and less 'user-friendly'. The programs developed here, while more limited, are better designed for analysis of the diffuse scatter since they were written with that task in view. In particular, the routine in the Hamburg program set for removing sharp reflections from the data produces a separate file containing the reflection, which has to be subtracted from the original data set as a second operation, and the fit is not displayed superimposed on the original data, though this can be done with another routine in the set (C.Staddon, personal communication). Thus the program LKBDEL to remove sharp reflections from the data is easier to use for that purpose. I understand also (from the same source) that the routine LKBRG to find the radius of gyration of the myosin heads from the diffuse scatter has no direct equivalent in the Hamburg program set, and that it was necessary to fit a tangent to a Guinier plot by hand and to calculate the radius of gyration from the gradient. In any case, the Hamburg programs were not available at the start of the project.

All the programs written for this project were tested carefully to eliminate errors. Plots of the data before and after processing were compared, and, where appropriate, the values of randomly selected data points were checked, to ensure that the correct operation had been performed on the data set.

This chapter describes the development of the set of programs written for the PDP 11, primarily to analyse the diffuse scatter. All the programs described here were developed by myself, unless otherwise stated. Further details of these programs may be found in the Appendix, where they are listed in alphabetical order of program name. The source code listings are also included in the Appendix.

4.1 Reading in the data

The program LKBIN was written to read in runs from the densitometer. It checks for errors as the data is received, and aborts the run if an error is detected. The data is stored in a series of .DAT files, one of which contains the run parameters, or header information, while the others each contain one scan across the film. Each scan from the LKB is sent as 13 lines of header, followed by a number of lines of data. The number of data lines depends on the number of points in the scan, and these values are given in the header, with the maximum and minimum intensities in the scan, the scan number, and various set-up parameters. Each line of data is prefixed with the number of values in the line, and ends with a checksum value. The program LKBIN simply copies the header of the first scan, and then updates the values given as necessary with each subsequent scan, so that the header file gives the number of scans done, and the maximum and minimum intensity values in the run. These values are correct even if the run aborts part way through. The checksum values in each line are matched with the data, and the run is aborted if a discrepancy occurs. The last scan in the run is followed by a Control Z character. .DAT files are ASCII, so that they can be typed out on the screen. The film is placed on the densitometer with the meridian in the y-direction, so that the scans are parallel to the meridian. The data is stored in this way as runs sometimes fail, often because data is lost in transmission, and as a run may take several hours, it is useful to be able to recover part of a failed run rather than repeating the entire run. However, .DAT files are both bulky to store and slow to handle, so the program LKBCVT was written, to convert the data to .BIN file format —a binary direct access file, described in the Appendix, which is economic of storage space and quick to access. The program LKBIN was not changed, as the original file format was better for reading in the data—the convenience of being able to type out the data files on the screen and the added security of not losing the entire data set if a run failed part way through were worth the delay in converting file types after the densitometer run was complete.

4.2 Viewing the data

The program LKBPLT was written to plot out the data either on the graphics screen or on the graph plotter. It will do simple plots of single scans or of a series of scans averaged together into one (see Figure 4.1), or it will do pseudo 3-dimensional plots, where a series of scans are plotted, each offset above and to the right of the previous one (see Figure 4.2). The program can also plot rotated scans, reading the n^{th} point from each scan and plotting the resulting series of points as a scan. The scale is calculated using the header information, to be as large as possible given the maximum intensity value in the file. The data may be multiplied by a scale factor entered by the user, in order to examine the less intense parts of the pattern, in which case intensity values falling outside the plot area are omitted.

The program LKBPLT was modified to read in .BIN as well as .DAT files, and the latest version will only make the simplest plots from .DAT files, as the more complicated plots take too much time, though it is often useful to view a few scans from the data set before conversion, to check the scanning of the film. All subsequent programs read and write .BIN files only.

4.3 Calibrating the data

The distances on the film must be converted to distances in the muscle, expressed in reciprocal nanometres. This can be done either by using the camera length and the X-ray wavelength, or by finding the separation of known peaks in the pattern. The second method is preferable, since the camera length and wavelength may not be accurately known for all films, and since the distance calibration of the densitometer may not be accurate. The program LKBPK was written to locate the positions of two peaks in the pattern using the crosswire facility on the Tektronix screen and to calculate the point separation in reciprocal nanometers from the peak separation, and the position of the pattern centre from the midpoint. The measurement may be repeated, the mean value being used. Errors in aligning the film on the densitometer plate may be seen by comparing the scan centres in scans on either side of the meridian. The calibration and centre values are stored as part of the header information. The program must be run twice, once to examine normal scans and to find the calibration and the centre in the meridional direction, and once to examine rotated scans, and to find the centre in the equatorial direction. In the patterns examined in this thesis, the peaks are sharper in the meridional direction, so the calibration was done on normal not rotated scans.

4.4 Fitting over the sharp reflections

These programs were written in the first place to study the diffuse scatter in the X-ray pattern, and for this purpose the sharp reflections must be removed. The program LKBDEL was written to fit a first or second order polynomial to two regions of the data, chosen by the user, and to replace the points between these regions with intensity values calculated from the fit

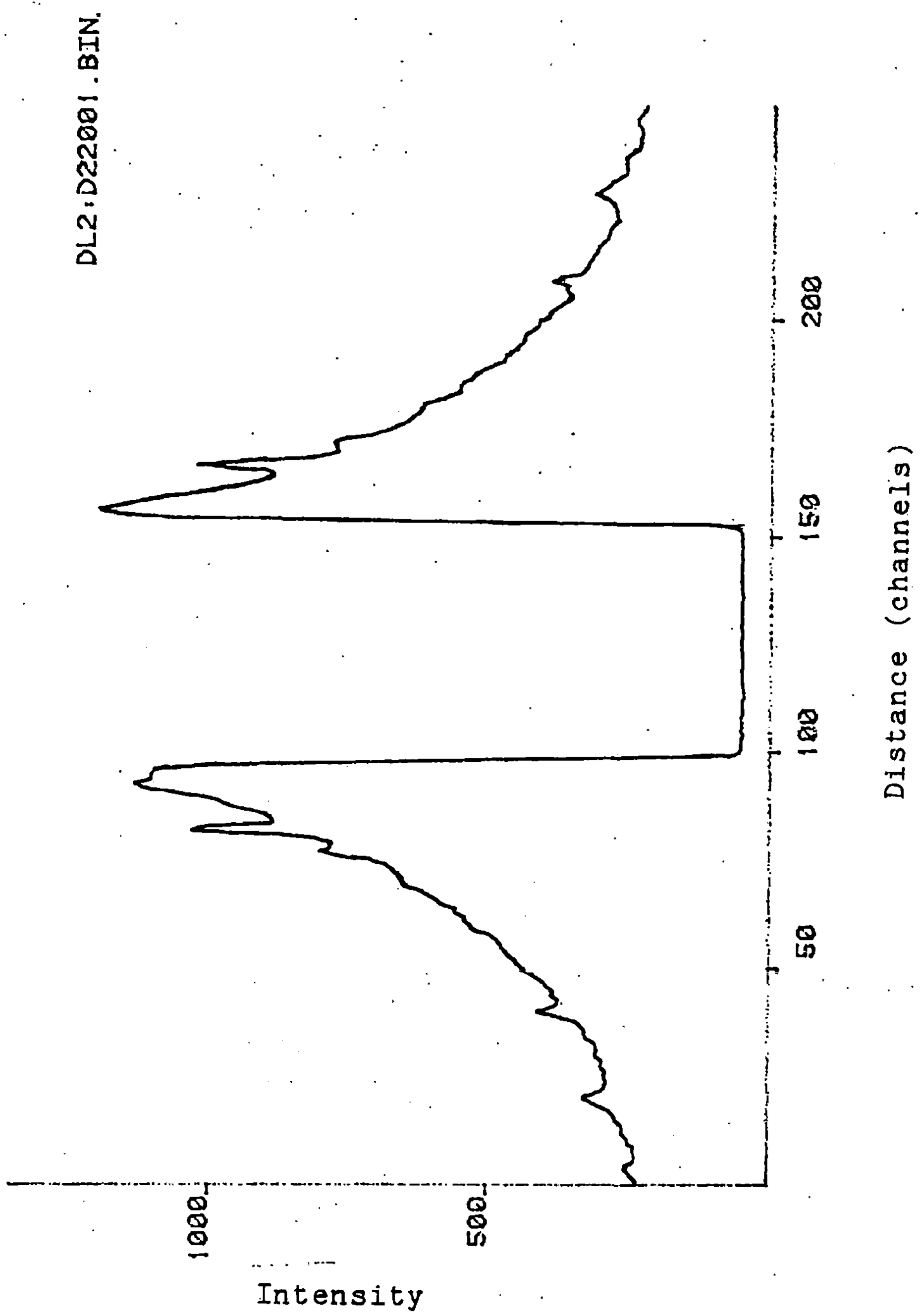


Figure 4.1 A plot of a single scan through a pattern from a heat-treated frog muscle. This is film 22a, from the Daresbury experiments (Chapter 5)

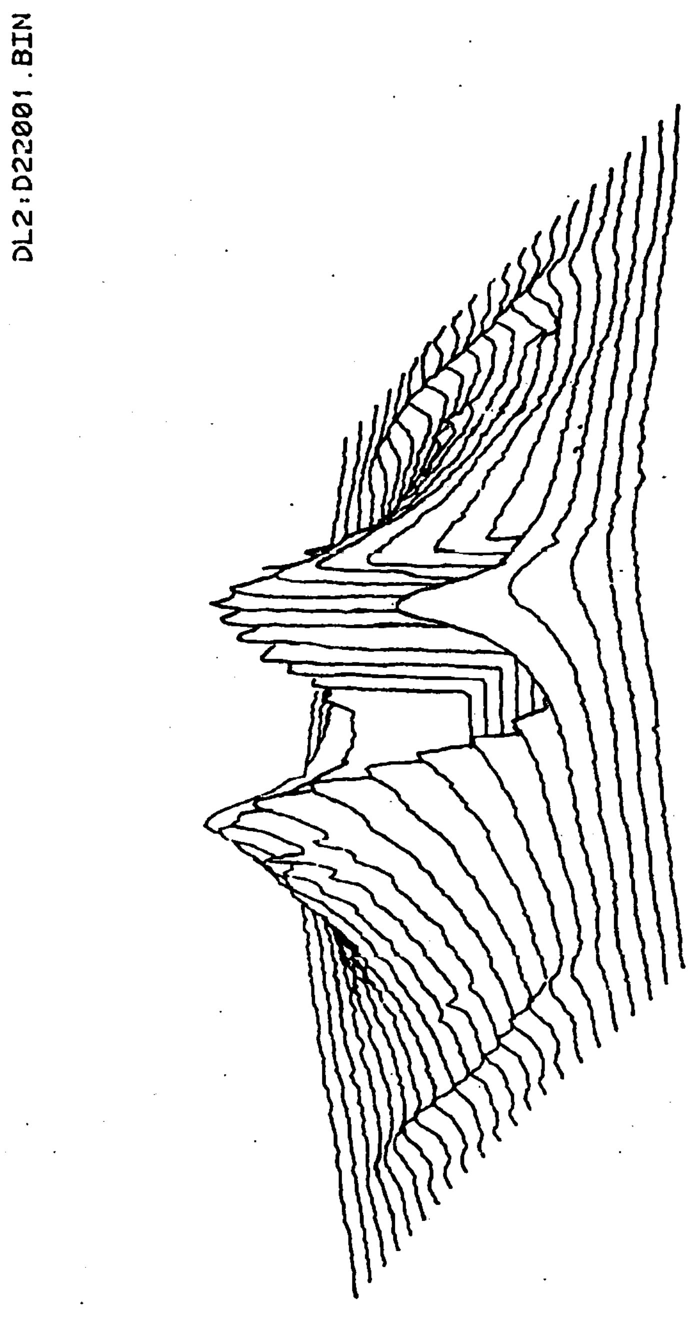


Figure 4.2 A 3-dimensional plot of a frog muscle pattern; this is film 22a, as in Figure 4.1.

parameters. A Numerical Algorithms Group (NAG) library routine is used to do the polynomial fit. The program is used to extrapolate across a peak from the regions on either side, and so to remove the peak (see Figure 4.3). The same regions can be treated in a number of consecutive scans, as peaks usually extend some way in the equatorial direction, and regions can be defined in two non-consecutive scans, and regions in the intervening scans calculated by linear extrapolation between the two, to remove features which run diagonally across the film—for instance, shadows of wires supporting the beam stop. A value for the root mean square residual of the fit over the entire area processed is calculated from the mean square residuals returned by the NAG fitting routine. This enables the user to estimate the uncertainty in the fit. This program changes parts of the input file, and so files should always be copied before using the program. The maximum intensity value in the header is updated if it is increased, but not if it is reduced, as the program does not read all parts of the file and so cannot evaluate the new maximum in the case of a reduction.

The program LKBDEL was compared with the equivalent routine in the Hamburg program set. The two routines are not exactly equivalent; in both routines regions are defined on each side of the part of the data that is to be removed, and these regions are used for the polynomial fit. The chief difference between the routines is that LKBDEL does not replace the data in these regions, but only in the enclosed part, while the Hamburg routine does replace the values of these parts. There may also be a difference in choosing the precise positions of the regions, which may make a slight difference to the fit. This difference will be greater in the case of a second order fit, particularly where the fitted data includes a local maximum or minimum, since the fit is more sensitive to variations in the data in this case. This means that a comparison of the two routines is complicated. It is clear though that the results of the comparison were satisfactory (Figure 4.4).

To study the sharp reflections, the processed file, with the relevant reflections removed, is subtracted from the original data file using the program LKBBG. This leaves the sharp reflections only (Figure 4.5).

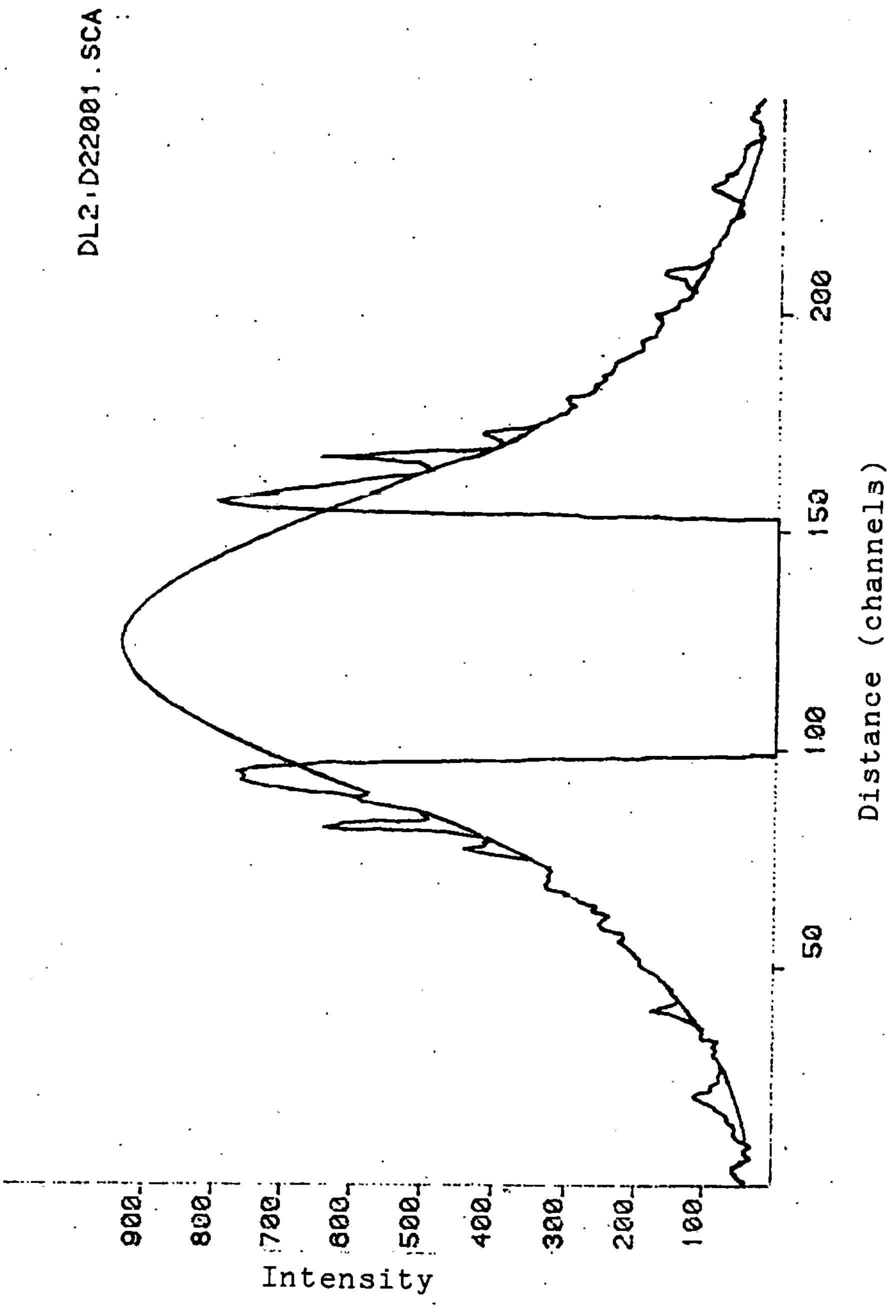


Figure 4.3 A scan through a frog muscle pattern, showing fitted data, with the sharp reflections removed, superimposed on the raw data. This is again film 22a.

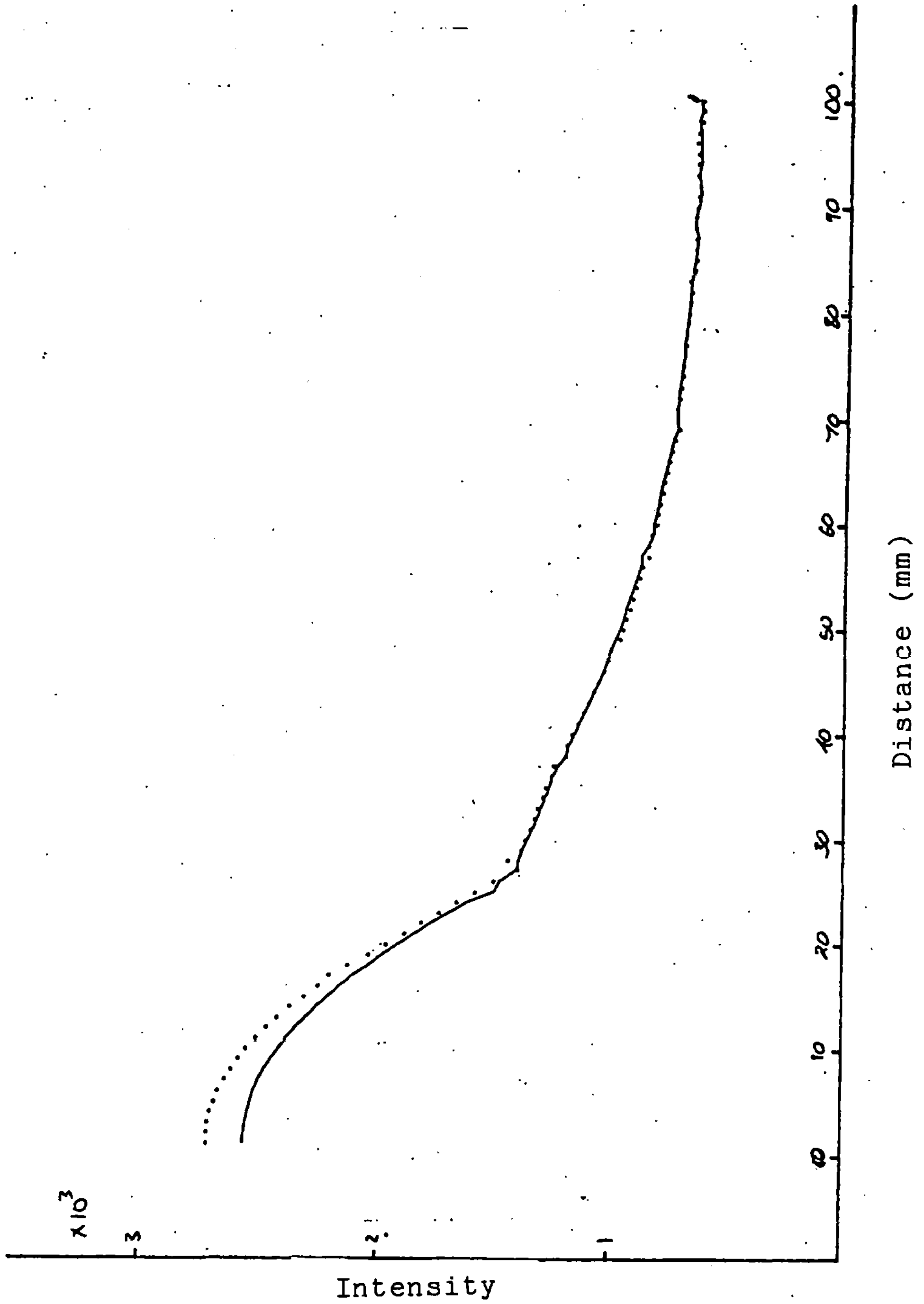


Figure 4.4 A comparison between the program LKBDEL (the dotted line) and the equivalent routine from the Hamburg programs (the solid line). This is a meridional scan from a frog muscle pattern, and all the sharp reflections and the beam stop shadow have been fitted over; a total of 11 regions fitted, including the pattern centre and five peaks on each side, with only 2–4 points between each peak, 6–10 points in each peak, and about 50 points in the centre, so that in fact the greater part of the data here is fitted. The data sets have been averaged into one quadrant after fitting, and the meridional scan from each has been plotted here. Note that a second order fit enclosing a local maximum is very sensitive to the regions of the data that are used.

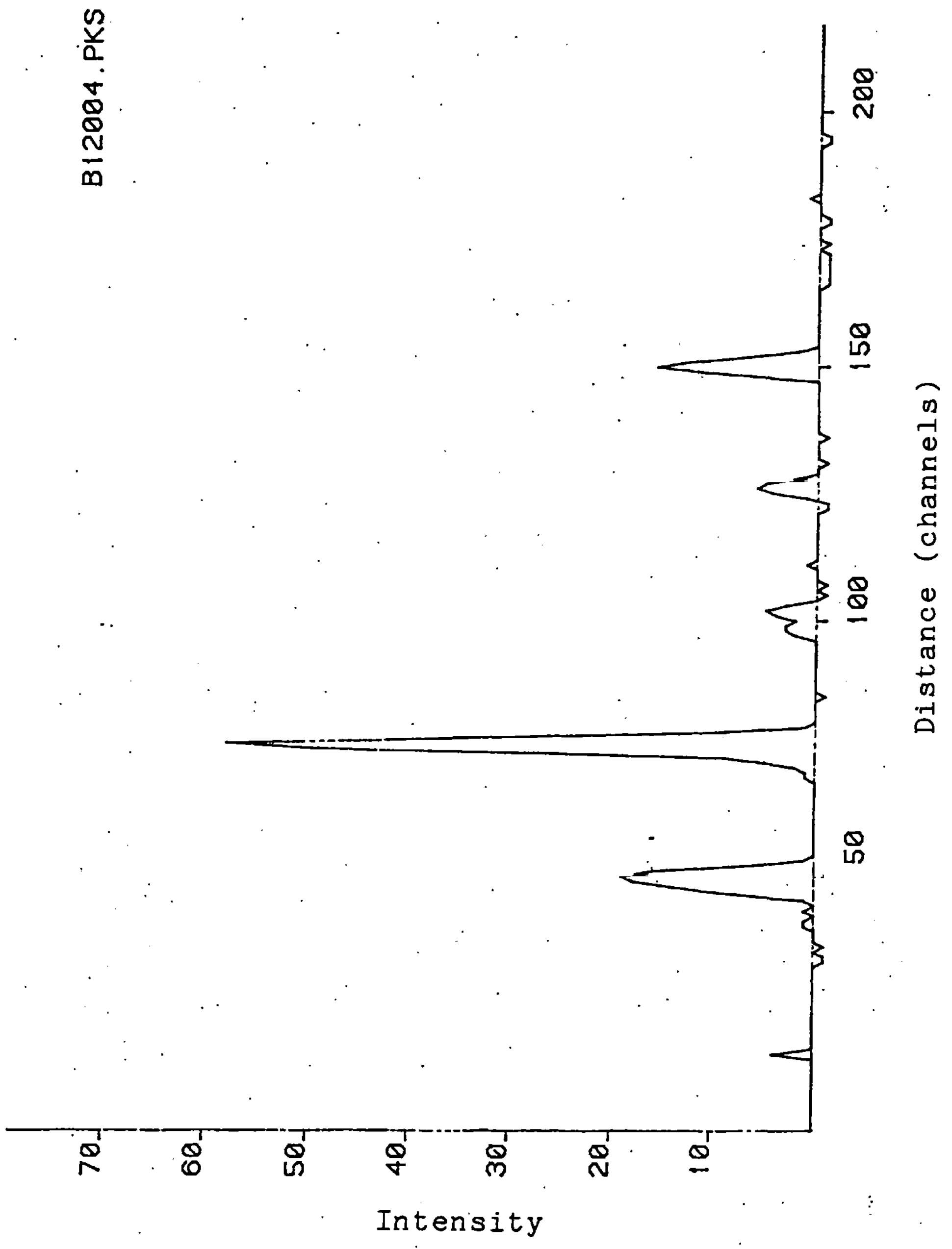


Figure 4.5 A meridional scan of film 1688e, a frog muscle exposure from the archive patterns processed in Chapter 7, showing the sharp reflections, with the scatter removed.

4.5 Manipulating the data set

The program LKBDEL can only treat scans parallel to the meridian, not rotated scans. To fit across features that are narrow in the equatorial direction, the data set must be re-written, so that scans are parallel to the equator. The program LKBREV was written to rotate the data set through 90 degrees, clockwise or anticlockwise, or to reflect the data set in the meridian or equator, or both. The program LKBROT was written a little later, as LKBREV took a very long time to rotate a file. The programs LKBROT and LKBREV will both rotate the data set to give the same output file, but LKBROT will hold a number of scans in memory at a time, and is therefore much more quick and efficient than LKBREV, which only has one scan in memory at a time. Because of memory limitations on the PDP 11, LKBROT can only treat a file with 256 data points or less per scan, and 256 scans or less, while LKBREV can treat much larger files. Data sets of 250 by 250 points are often used, as this gives a good resolution over the area of interest of the film, and these can be treated with LKBROT. These programs do not change the data in any way; they merely re-write it in a different order.

If a peak is near the centre of the film pattern, where the scatter is rapidly varying, it is difficult to get a good fit under the peak. In this case, better results are obtained by fitting to log intensity than to the original intensity data. The program LKBLOG was written to convert a data set to and from log intensity. The log intensity values are multiplied by 1000, so that they can be stored as integers in the same file format without loss of information.

4.6 Averaging the data

To obtain a plot of intensity against distance from the pattern centre, for comparison with the solution scattering data, the pattern must be averaged. First, the program LKBAV was written to average together the four quadrants of the pattern, reducing the noise level in the data. The quadrants can be added or averaged; addition gives a more precise result, as rounding errors are reduced, but it may not be possible to use this option on all data sets since the maximum integer value on the PDP 11 computer is 2^{15}, or 32768. The output data is in the form required for the program SPHAVG, written by Finn Poulsen. This program calculates a circular average of the points, each point being weighted according to its distance from the meridian of the diffraction pattern. This gives a spherical average of the pattern, taking into account the random orientation of the myosin heads in a plane perpendicular to the muscle axis. This random mix of orientations effectively gives a cylindrical average of the pattern from a single myosin head. There is also a version of this program called CIRAVG, which gives a simple circular average of the pattern. This is used for patterns where the cylindrical averaging has not occurred, for instance, camera blanks.

4.7 Background removal and approximation

The intensity variations of X-ray diffraction patterns have been studied in detail by a number of people. At large angles, where the scatter intensity tends to a constant value, the Porod approximation states that the intensity varies inversely as the fourth power of the distance from the pattern centre [7].

The first background removal program was LKBA0, which removes a constant from the intensity values. This is used where a constant intensity level was taken as an approximation to the camera blank, or to subtract the film fogging level from the data.

The program LKBBG was written to subtract a blank from a muscle pattern. The program can perform three functions. First, since the blank was usually scanned over a larger area than the film, to ensure that the film data area is fully covered, the blank data set may need 'trimming' to match the film data set. The program LKBBG can use the film data file header to cut down the blank data file so that both data sets have the same number of points in the meridional and equatorial directions, and the same centre position. When the two files have matching centres and sizes, LKBBG can plot scans from the two files, superimposed on one another. Finally, the blank can be subtracted point for point from the film data, scaled by a value entered by the user—for example, the ratio of the exposure times. It may be necessary to subtract a constant value from the data using LKBA0 to allow for film fogging. Negative intensity values less than -5 are not allowed, so it is sometimes necessary to add a constant to the muscle data with LKBA0 before subtracting the blank, if the blank is to be scaled up, or the resulting data will be zero over a large part of the pattern. The small negative values are allowed to account for noise in the patterns. Then the data must be fitted and extrapolated to a constant value, at a large distance from the pattern centre. This constant intensity value is the background level which should be subtracted. The extrapolation is done using the Porod approximation. The level is found using the program POROD, which fits the outer parts (defined by the user) of the spherically averaged data to the Porod equation:

$$I\alpha\frac{1}{x^4}$$

The data and the fit are displayed on the graphics screen, so that the fit quality may be evaluated. The resulting background level may be subtracted using the program LKBA0 (see Figure 4.6).

There are many films in the Lowy/Poulsen archive for which there are no exposures of the camera scatter. This archive represents a large bank of data taken over more than a decade, with many different experiments done on several different muscle types. To repeat all these experiments with camera blanks would be a very major operation. If the camera contribution could be approximated, these films might be analysed. The camera blank generally consists of three components (see Figure 4.7):

1. Clearly defined streaks or shadows: often a vertical streak (particularly in exposures taken on laboratrory X-ray generators), and shadows of the beamstop and the wires supporting it. These can be removed in the same way as the sharp reflections (see Section 4.4), by fitting a polynomial over them, with the exception of the beamstop shadow, which is not normally in a part of the pattern that is to be studied.

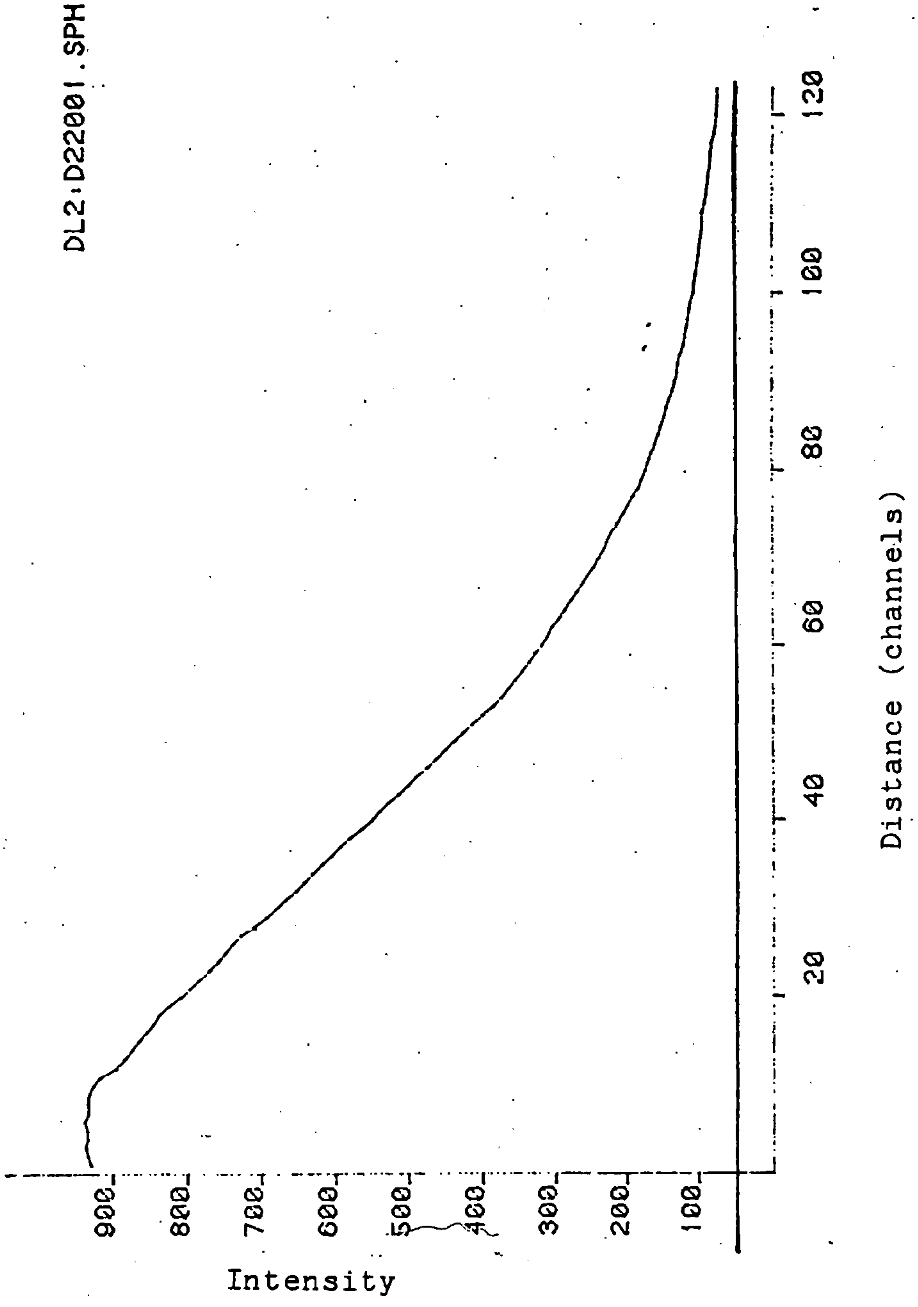


Figure 4.6 A spherically averaged pattern, showing the PoRoD background level; again, this is film 22a from the Daresbury experiments.

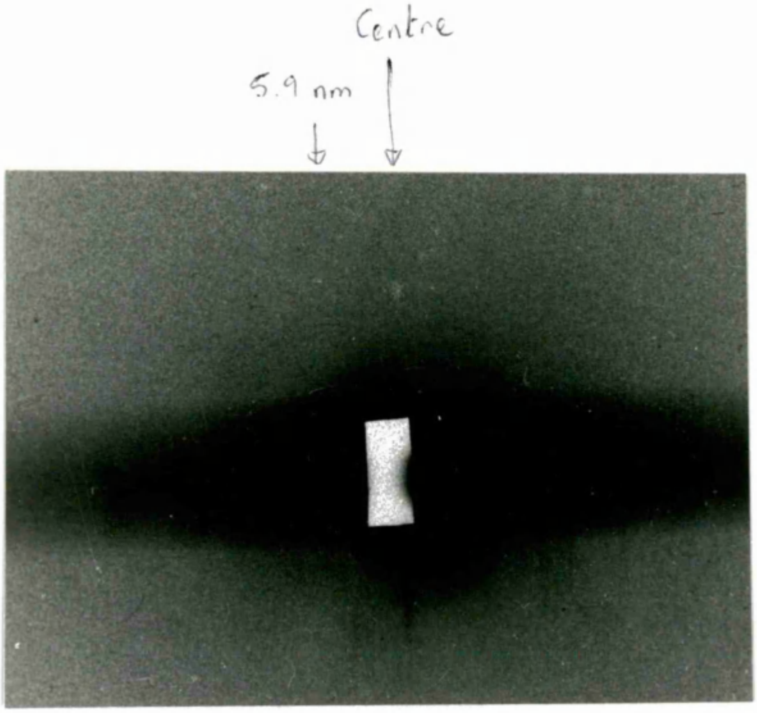


Figure 4.7 A print of an exposure of the camera scatter, from the November Daresbury experiments described in Chapter 5.

2. A constant scattering level, which can be subtracted from the data with the program LKBA0.

3. A region of diffuse scatter near the centre of the pattern, which cannot be distinguished from the muscle scatter. This region is normally of much lower intensity than the muscle scatter, and it may therefore be possible to neglect it (see Figure 6.1), though if a camera blank does not exist it is not possible to know how significant its contribution might have been.

When films for which no camera blank is available are processed, the peaks arising from the camera scatter are removed at the same time as the sharp reflections, and a constant level is subtracted. The value of the constant is important as it will affect the gradient of the Guinier plot (see Section 4.8). Various background levels were tried for patterns where there was no camera blank; the level under the 5.9 nm line is too high, as much of the pattern is reduced to zero if this is used, and the pattern minimum is too low, being the beamstop shadow. The intensity at the edges of the exposed parts of the film was better, but it was difficult to be consistent when comparing a number of patterns. The best value for the background is that given by the program POROD; it is close to the value in the outer parts of the film, and is more consistent from one film to another.

4.8 The radius of gyration

The radius of gyration of the myosin heads, R_g, can be calculated by the equation $R_g = 0.416 \times \sqrt{\alpha}$, where α is the modulus of the gradient of the Gúinier plot—the plot of log intensity against distance squared—in the Guinier region (Section 1.3). This equation holds good for myosin heads in muscle as these behave like myosin heads in solution (see Section 3.1). The program GUIPLT was written to plot out the data as a Guinier plot. The radius of gyration can be found by drawing a straight line through the plot at the Guinier point (see Figure 4.8).

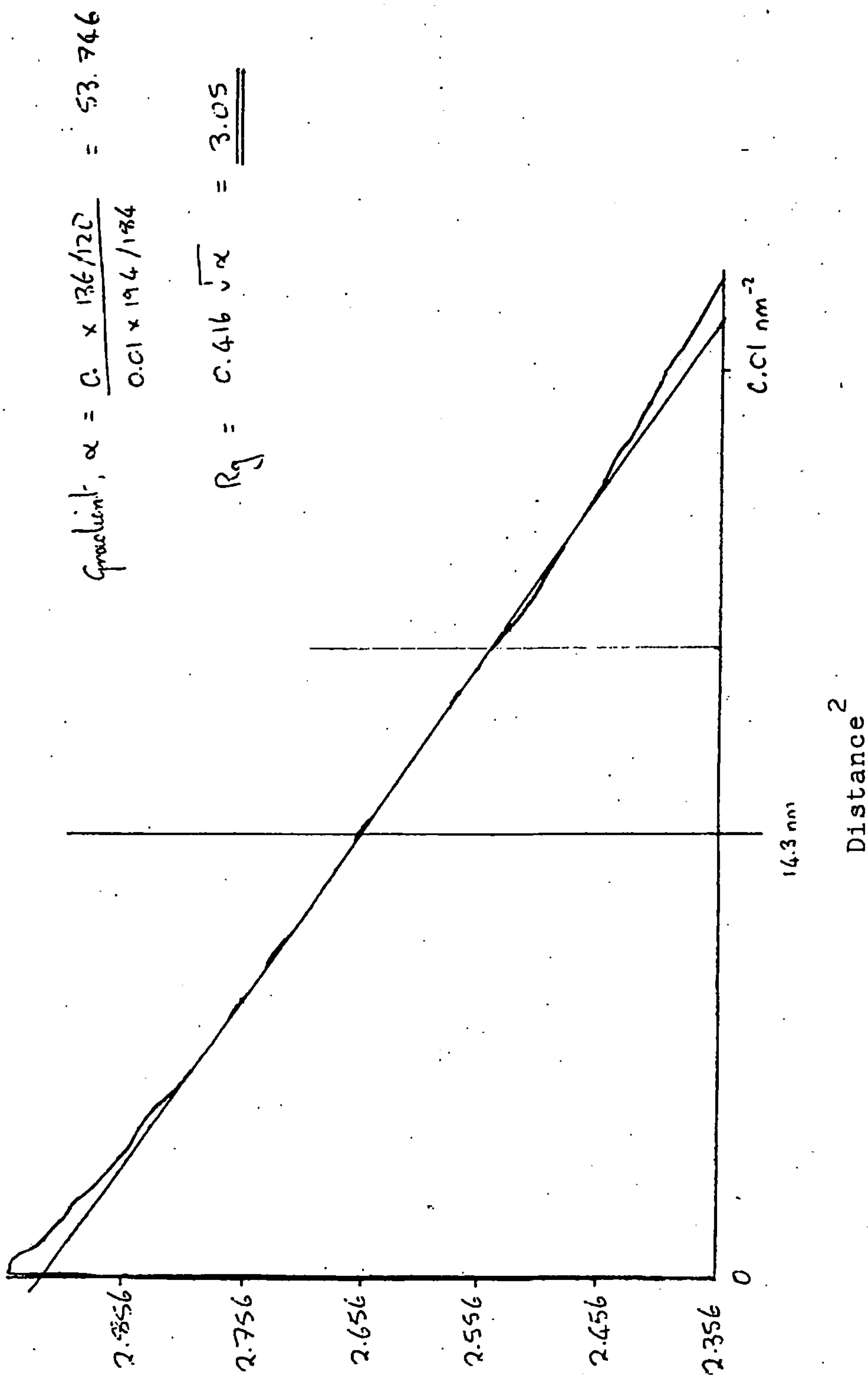


Figure 4.8 A Guinier plot of a spherically averaged pattern, with a line drawn to the curve at the Guinier point.

When the gradient of the curve at the Guinier point was measured by three different people (Finn Poulsen, Chris Staddon and myself), each using copies of the same plots, there was a systematic discrepancy in the positioning of the line through the curve by each person, and hence a difference in the gradients obtained. This problem was overcome with the program LKBRG, which uses a NAG routine to fit a cubic spline to the log intensity data, and uses it to calculate α and hence the radius of gyration of the myosin heads. It was found that LKBRG generally gives a more consistent value of R_g, though if there is an irregularity in the data on the Guinier point, it can sometimes be better to find the gradient of the curve manually. The program LKBRG plots out each stage of the process and so the spline fit quality and the tangent to the curve can be checked, and it is clear whether or not the result is adequate (see Figure 4.9). In particular, it is important to check that the Guinier plot is linear about the Guinier point. The uncertainty in the R_g value was estimated from the discrepancy in the measurements made by different people, and from typical camera blanks, to be about 10% for the manual method, and similar or slightly less for the program LKBRG. The main advantage of the least-squares fit is in the reduction of systematic error due to the 'personal' effect, rather than in a major improvement in precision.

A constant or near constant region of diffuse scatter from the camera will affect the R_g value if not corrected fully. However, exposures taken on the same camera may be compared, since the error in the R_g will be the same for all so that measured changes will be correct even though the absolute values of the R_g's may have a systematic error.

4.9 The symmetry of the data

The symmetry of the data can be assessed by plotting a scan along the meridian superimposed on a scan at 45° to the meridian (see Figure 4.10). However, this is not appropriate if there is a contribution from the camera on the meridian, even if this contribution has been fitted over, as the fit is

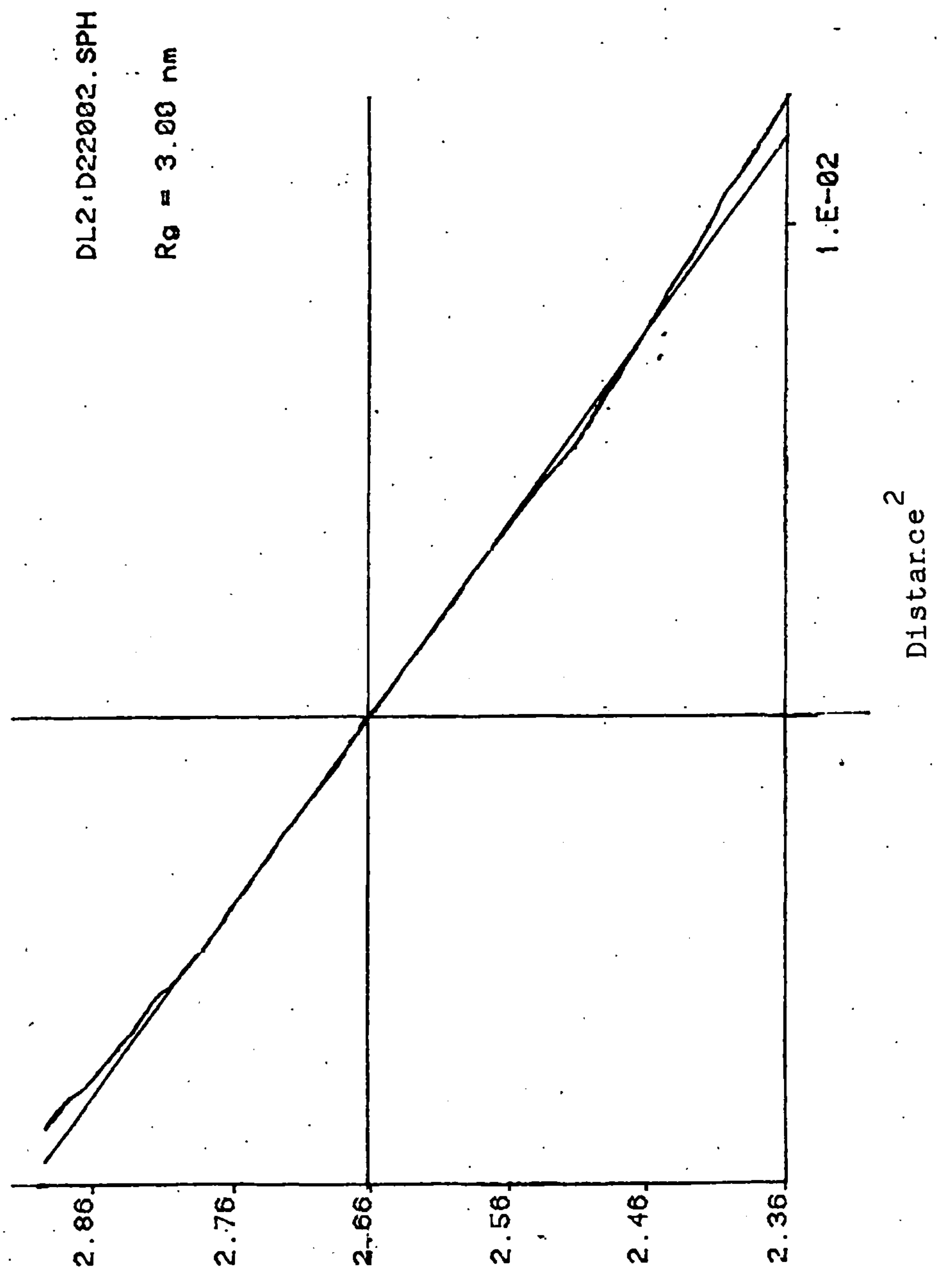


Figure 4.9 The output of the program LKBRG for finding the R_g value from a spherically averaged pattern. Film 22a from the November Daresbury experiments.

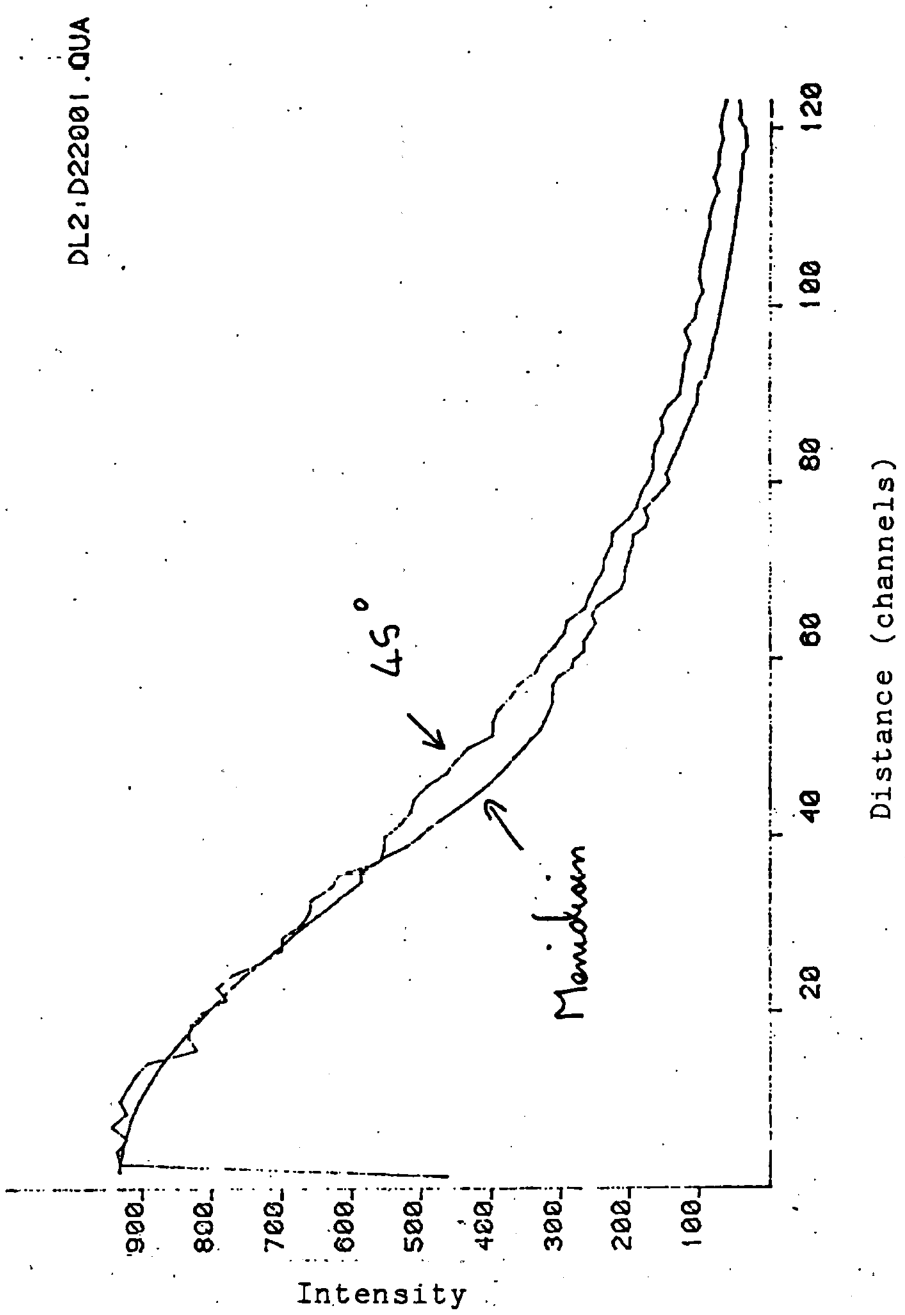


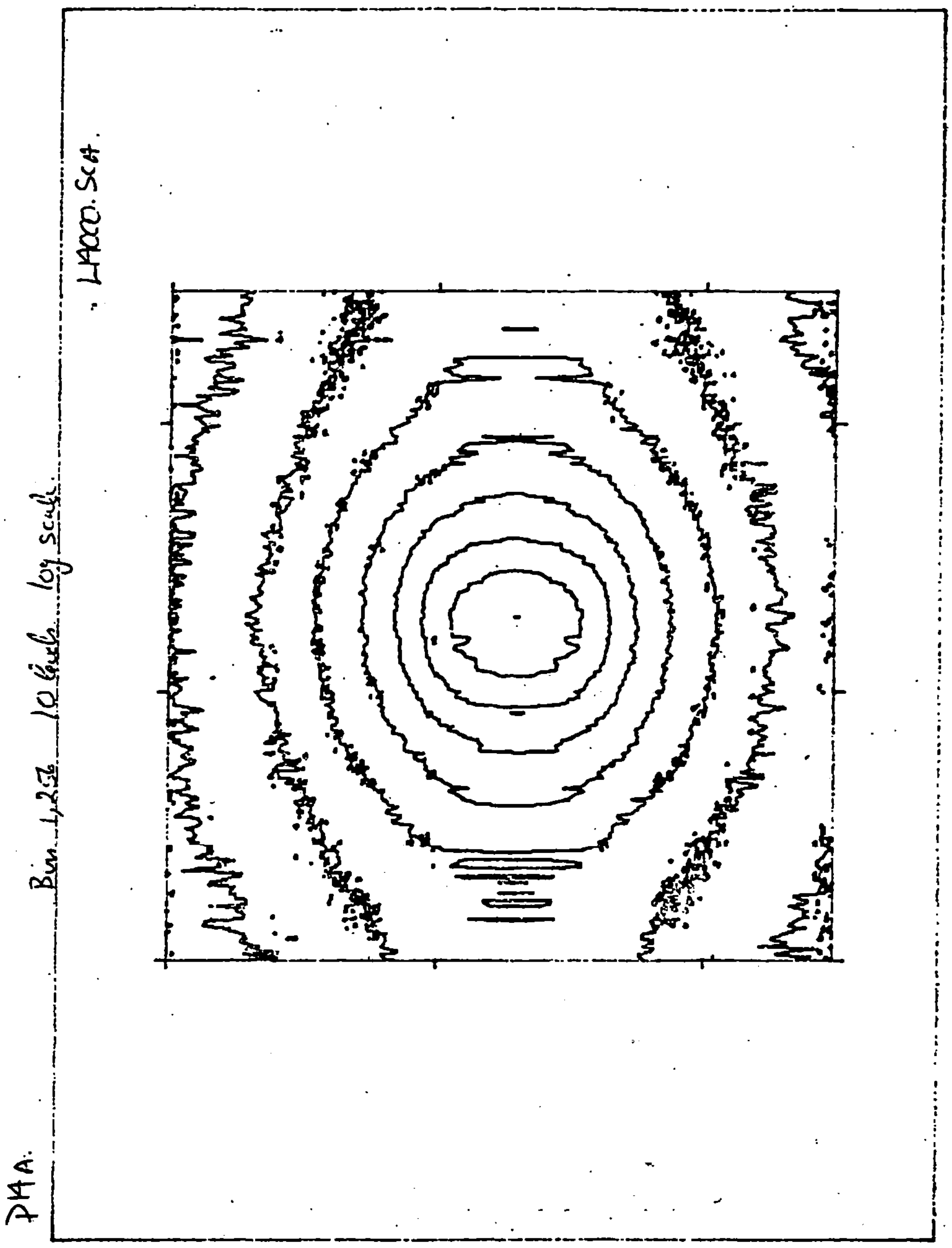
Figure 4.10 The meridional scan of film 22a superimposed on the 45° scan.

not as reliable as the raw data. If there is a 45° streak, the fit will be more reliable, but those quadrants that contain the streak should not be used for the comparison. Thus this method is only really reliable for films where a camera blank is available. For a better assessment, a contour plot might be preferable (see Figure 4.11). A program to do this was not written on the PDP 11, as the EMBL programs had become available on the OU VAX by this time, and these included a contour plot routine. Therefore, data must be transferred to the VAX for symmetry measurements, though much of the initial processing is better done on the PDP 11. However contour plots are difficult to assess, since there is usually fibre scatter on the equator and camera scatter on the meridian, and deviations from circular symmetry are small [28], and so they are of limited value.

The data cannot be transferred to the VAX in binary form, because of differences in the data storage of the two machines. KERMIT, the communications program used, cannot cope with these differences. Therefore, the program LKBCVT, (which converts data from .DAT to .BIN file type), was modified so that it can also convert the data to and from a single-file ASCII format suitable for file transfer, called a .HAM file. The EMBL programs use a binary direct access data file, of a different type from the .BIN files on the PDP 11, with a separate ASCII header file, and a program ASCDA is used for conversion on the VAX.

4.10 Measuring the sharp reflections

Two aspects of the sharp reflections are interesting: the intensity, or area under the peak, and the width. The intensity is measured by summing the individual channels in the peak, in a file where the scatter under the peak has been removed. This can be done with the program INTEG. The user defines the part of the scan to be included using the crosswire facility on the Tektronix screen, and types in the first and last scan. The channels thus defined are then summed and the answer typed out. An estimate of the uncertainty in the intensity can be made using the r.m.s. residuals typed



27/11/88

Figure 4.11 A contour plot of film 14a, rigor frog muscle, from the November Daresbury experiments.

out by the program LKBDEL when the background to the peak is removed, or by fitting over a part of the pattern where there are no peaks with LKBDEL and using the r.m.s. residual of this fit. However, the quality of the fit is best judged by eye.

The peak width at half height can be measured from a plot of a scan through the maximum of the peak, or the program WIDTH can be used. This program operates on a data set where the background has been subtracted. The user locates the peak maximum on a plot on the Tektronix screen, using the crosswire facility, and the half height is marked on the screen. The crosswire is used to measure the peak width at this height. This is quicker and more convenient than using a ruler and a hard-copy plot. A problem arises in that many peaks are compounds of several unresolved or partially resolved peaks. In this case, it is not possible to measure a consistent value for the width, especially since the relative contributions of the individual peaks may vary from one exposure to another as experimental conditions are changed.

4.11 Other programs

There are several other programs for manipulating .BIN files on the PDP 11. These include :

- AMAX, which finds the maximum and minimum intensity values in a file, and updates the header. This can be used after LKBDEL, which may change the maximum or minumum intensity values, but does not always update the header, or on a file transferred from the VAX, which will not have these values in the header.
- HEAD, which allows the user to type out and change some header information values.
- LKBCH, which types out the intensity value of a specified data point.
- LKBRNG, which plots out the intensity as a function of angle, for a ring at a chosen distance from the pattern centre. This can be help-

ful in locating deviations from circular symmetry, though it is not a substitute for a contour plot.

- LKBSM, which smooths a data set by averaging or adding a chosen number of points together, thus reducing the size of the data set.

- MIRROR; this program either takes scans of n points and creates scans of $2n - 1$ points, where the first half of each scan is a mirror image of the second half, and the last n points are the original scan, or the program will do the inverse operation, taking a data set of scans of $2n-1$ points and creating scans of n points, which are the last n points of the input scans. This program can be used to fit over the beamstop region in a quadrant averaged or spherically averaged data set, as the fitting program, LKBDEL can only fit over a region within a scan, not one at the edge of the data.

- UNSA, which recreates a full data set from the spherical or circular average. This can be used with SPHAVG or CIRAVG to produce a circularly symmetric pattern. The programs SPHAVG and CIRAVG are equivalent when the input data set is circularly symmetric, so UNSA is the inverse of both of these programs.

4.12 Summary

A full set of programs for analysis of X-ray diffraction patterns of muscle now exists on the PDP 11, with the exception of a contour plot routine. The PDP 11 has been connected to the VAX, so that data may be transferred between the two machines, and a contour plot routine is available on the VAX. The programs that have been written on the PDP 11 have been tested rigorously. A brief list of these programs follows.

- AMAX updates the maximum intensity value in the file header.

- CIRAVG (by F.Poulsen) performs a circular average.

- GUIPLT makes a Guinier plot of the data.

- HEAD allows the user to change some header values.
- LKBAV averages the quadrants of the data together.
- LKBA0 subtracts a constant intensity from the data.
- LKBBG subtracts one data set from another.
- LKBCH types out the intensity value of a specified data point.
- LKBCVT converts data from one file type to another.
- LKBDEL fits over parts of the data set.
- LKBIN reads in data from the LKB densitometer.
- LKBLOG converts a data set to or from log intensity form.
- LKBPK calibrates a data set.
- LKBPLT plots the data.
- LKBREV rotates a data file of any size, slowly.
- LKBRG finds the R_g value from a data set.
- LKBRNG plots the intensity on a ring about the pattern centre.
- LKBROT rotates a small data set, quickly.
- LKBSM smooths a data set.
- MIRROR appends or removes the mirror image of the data set.
- POROD finds the background intensity using the Porod equation.
- SPHAVG (by F.Poulsen) performs a spherical average.
- UNSA recreates a full data set from the spherical average.

Chapter 5

Experimental Work at the Daresbury Synchrotron Laboratory

Experiments were done using frog semitendinosus muscle on two visits to Daresbury Laboratory, near Warrington, in September and November 1988. The experiments were done on Station 8.2. The wavelength of the X-rays was 0.15nm, and the beam at the sample was about 3mm wide by 1mm high. The muscles were mounted vertically in the beam, so that the best resolution was in the meridional direction, and the beam covered the full width of the muscle. The specimen could be moved horizontally and vertically in the beam by remote control while the beam was on. There were ionization chambers before and after the specimen to measure the beam flux. Three films were used for each exposure, separated by black paper.

5.1 September 1988

The first visit to Daresbury Laboratory was on 26th September 1988. The muscles taken were dorsal semitendinosus muscles of Rana Temporaria. Two muscles were dissected from each frog, of which one was stretched and one kept at body length. One pair was fixed in 2.5% glutaraldehyde for 24 hours, 2 pairs were put into 1mM iodoacetate solution to induce rigor, and 4 pairs were relaxed, in normal Ringer. A one metre camera was used. For a beamstop, a piece of lead was fixed to the film cartridge. This meant that the beamstop could be changed easily, and the best size chosen. However, the first exposures showed intense rings, that proved to be diffraction from the mylar window at the end of the camera. The beamstop was therefore changed for a piece of lead mounted on three wires, just inside the mylar window.

The next exposure was of a live muscle, for 30 minutes. The muscle was not moved during the exposure. After exposure, the muscle was examined under a binocular microscope. An opaque white line was clearly visible across the muscle where it had been in the beam.

Another live muscle was exposed, and was moved 1mm up after 10 minutes. The beam failed after 19 minutes exposure, and remained down for the remainder of the assigned experimental period, so no more exposures were made. The muscle was examined under the binocular microscope for signs of damage by the beam, but none were seen.

5.2 November 1988

The second visit to Daresbury was on the 5th and 6th of November 1989. As before, pairs of dorsal semitendinosus muscles were dissected out, and one muscle of each pair was stretched, while the other was kept at the body length. Two pairs of muscles were heat-treated, by putting them into Ringer at 39°C for 10 minutes. The other muscles—some ten pairs—were put into a 1mM iodoacetate solution to induce rigor. An 80cm camera was used—the specimen to film distance was 91.5cm. The normal beamstop was used, mounted inside the end window of the camera.

Each muscle was exposed for 30 minutes, and was moved 1mm vertically every 6 minutes. This meant that no part of the muscle was exposed for more than 6 minutes, and the whole extent of the muscle cell window was used. The ionization chamber readings were recorded at the start of each exposure and when the muscle was moved. Two muscle chambers were used, so that a muscle could be mounted while the previous muscle was in the beam, and the change-over was as rapid as possible. Twenty-four 30 minute exposures were taken, including blanks of the two muscle chambers containing Ringer's solution only, and one 13 minute exposure, of one of the chambers, containing only Ringer, but screwed down hard so that the mylar windows were stretched out and bulged. The beam then went down for the remainder of the assigned experimental period, and no further data could be obtained.

5.3 Conclusions

The results of the analysis of the films are given in Chapter 6.

There were only two exposures obtained from the September trip, as the mylar scatter obscured the muscle pattern on the other exposures. Of these two, one of the muscles showed clear signs of damage by the X-ray beam after exposure. For these reasons, only one muscle exposure is certainly of use, and no camera blank was taken (as the beam went down before this had been done). However, a certain amount can be learned from this visit:

1. The 30 minute exposure clearly showed the muscle pattern, although the muscle was damaged, and the exposure therefore cannot be used. So 30 minutes is a suitable exposure time for this muscle type on a 1m camera, but the muscle should be moved or replaced several times during this period. This gives an idea of where to start when determining exposure times on future visits.

2. The muscles showed visible damage after 30 minutes, but not after 10 minutes. Thus no part of a muscle should be exposed for more than 10 minutes—preferably less if possible.

3. The mylar window at the end of the camera scatters X-rays strongly if the main beam passes through it, so the beamstop must be inside the end of the camera.

Both exposures from the September trip showed rings as well as the normal muscle pattern, however, as there was no blank exposure taken, it is not known whether these arose from the muscle or from the camera or muscle chamber. Three rings were visible, with diameters of 41.5mm, 62mm and

86mm. The actin 5.9nm line had a spacing of 60mm, so the rings have spacings of 8.5nm, 5.7nm and 4.1nm respectively. These correspond to the second, third and fourth orders of a ring of spacing around 16.8nm.

The November experiments produced more exposures. Many however showed curving of the layer lines, indicating that the orientation of the muscle fibres was poor. This should not matter for the calculation of the radius of gyration of the myosin heads, as the data is spherically averaged, but it may be a sign that the muscles were poor. The beamstop shadow is very large, and may partially obscure the Guinier region; our efforts to decrease the size of the beamstop were frustrated by the failure of the X-ray beam. The rigor muscle exposures did not show the characteristic rigor pattern, although the muscles were stiff as rigor muscles should be. Here again, the beam failure prohibited further experiments.

All the muscle exposures showed rings, of varying intensity but the same diameter. These rings were not present on the exposures of the empty cells. The muscle cells had mylar windows, which may possibly scatter if stretched; it was to test this possibility that the exposure was taken of an empty cell with the mylar stretched. No rings were seen on this exposure. Thus it is clear that the rings arise from the muscles themselves. The diameters of the rings were 35mm, 52mm and 69mm, with the 52mm ring being less intense than the others in those patterns where all three were clearly visible. The actin 5.9nm lines had a separation of 50cm.. So the rings were at spacings 8.4nm, 5.7nm and 4.3nm respectively. These are presumably the second, third and fourth orders of a ring of about 17nm spacing, and the third order appears to be weaker than the other two. This agrees well with the spacing of the rings seen on the September trip from live muscles. These rings may possibly arise from myelin layers in nerve tissue,or other lipid present in the muscle. The spacing of the reflections and the relative weakness of the odd order line agree with R.Worthington's measurements on nerve tissue[37]. The nerves in the samples exposed at Daresbury would be randomly oriented, so the diffraction pattern would consist of rings rather than lines, and these rings were present in rigor, heated and relaxed muscles.

Chapter 6

Radius of Gyration Measurements

The diffuse scatter was studied in various films from experiments in the Lowy/Poulsen archive, and also some films taken in the Daresbury experiments (Chapter 5). The films were measured on the LKB densitometer, using the program LKBIN to read the data onto the PDP 11 computer, and a scan of the densitometer plate was taken on the same day, and subtracted from the data before further processing. The data-sets were then calibrated using the program LKBPK and unwanted parts of the pattern were removed using LKBDEL. The camera blanks were subtracted using LKBBG, or a constant level was removed using LKBAU where the blanks were not available. The data sets were then quadrant averaged with LKBAV and spherically averaged with SPHAVG. Data points within 20° of the equator were excluded in the spherical average to exclude a streak along the equator arising from fibre scatter. Some of the films listed here were processed before the program LKBRG was written, and so their radii of gyration were found from Guinier plots drawn by the program GUIPLT, while others were processed later, and LKBRG was used to find the radii of gyration.

6.1 ABRM patterns

Four films were analysed from experiments on the ABRM (anterior byssus retractor muscle) of Mytillus Edulis (the edible mussel). This analysis was done before the program LKBRG was written. All four muscles had been in cooled natural sea water; three were relaxed, the fourth was isotonically contracting against a weight of 10g. In this analysis, spacings on the films were calibrated using the paramyosin layer lines, as B.Millman and G.Elliott found that these have a constant spacing of 14.4nm, in relaxed, contracting and rigor muscle [32]. There were no exposures of the camera scatter in existance for any of these films, so a constant level was subtracted. The program POROD had not been developed at that time, so the intensity level at 0.22nm^{-1} from the pattern centre was used as an arbitrary standard. These measurements were not repeated using the Porod level; when frog muscle patterns were measured using both levels, the difference was less than the estimated experimental error (see Table 6.2).

The films (all from the Lowy/Poulsen archive) were:

- Film 1646a: Relaxed ABRM in normal sea water. This exposure was taken on a double crystal camera, which produced a fine vertical line. The muscle was set up at 45° to the vertical, so this streak appeared at 45° on the pattern, and did not obscure the meridian. This streak was therefore removed first, after the pattern was calibrated. Next, the actin layer lines were removed, using a second order polynomial fit, as this gave the best results. The meridional lines were then removed, using a first order polynomial, as the pattern was too rapidly varying in this region for a second order polynomial to give a good fit. The R_g value was measured with no background subtracted, and also with the 0.22nm^{-1} level subtracted.

- Film 1445b: also relaxed ABRM in normal sea water, this film was exposed and analysed in the same way as film 1646a above.

- Film 1627a: this was also relaxed ABRM in normal sea water, but was exposed on a mirror-crystal camera. There was no diagonal streak, but a more diffuse streak along the meridian. The layer lines were removed first, in the same way as the above two films, and then the pattern was rotated with LKBROT and the meridional ridge was fitted with a second order polynomial. The equatorial peaks were removed with first order polynomials, and the the pattern rotated back to its initial orientation. Finally, the equator was fitted with a second order fit. The 0.22nm^{-1} intensity level was subtracted from the data set.

- Film 1628a: isotonically contracting ABRM in normal sea water. This was exposed on a mirror-crystal camera, like film 1627a above, and was processed in the same way.

6.2 Frog muscle patterns

Fourteen frog patterns, again from the Lowy/Poulsen archive, from eight muscles in various states, were processed. Each muscle had been exposed twice, on a laboratory GX13 X-ray generator, first relaxed in oxygenated frog ringer as a control, and then either stretched, heated, or in rigor. I used the myosin 14.3nm meridional reflection for calibration, since, though the spacing of this line changes in rigor and possibly in heated muscle, the changes are small—less than 1% [11]. In any case, the treated muscles had been exposed on the same cameras as their controls, and thus the same calibration could be used for both. The analysis of these films was standardized as far as possible. All had been exposed on the same camera type—mirror-crystal cameras—so the same features were present in the camera scatter, namely a diffuse streak along the meridian. None of these films had corresponding camera blanks. For each film, the data was first converted to log intensity form, as this was found to give a better fit over peaks, and the sharp reflections, including contributions from the camera, were removed. Then the log data was rotated, and the meridional streak removed. The data was then rotated back, and the equator fitted over. Finally, the data was converted back to linear intensity data. In most cases, the R_g values were found using the program LKBRG with no background subtracted, with the 0.22nm^{-1} intensity subtracted, and with the POROD level subtracted.

The frog muscles were as follows:

- Muscle 1688. The experiment was done in 1977, with a 36cm camera. Film 1688a was a 17 hour exposure of the relaxed muscle, in oxygenated frog Ringer at 3°C and with a sarcomere length of 2.5μm. Film 1688d was an 18 hour 37 minute exposure of the same muscle, stretched to a sarcomere length of 3.4μm.

- Muscle 1689. This experiment was again done in 1977, on a 36cm camera. The muscle was relaxed, in oxygenated frog Ringer at 3°C. Film 1689a is a 17 hour exposure of the muscle at a sarcomere length of 2.6μm, and film 1689d, a 16 hour 25 minute exposure at 3.8μm.

- Muscle 1690. Also done in 1977, on a 36cm camera, with the muscle in normal Ringer, film 1690a was a 17 hour 6 minute exposure at a sarcomere length of 2.6μm and a temperature of 2°C, and film 1690d was a 20 hour 30 minute exposure, at 3.4μm and $1\frac{1}{2}$°C.

- Muscle 1624. 1977, 36cm camera, film 1624a was a relaxed muscle at 3°C with a sarcomere length of 3.9μm, exposed for 17 hours 5 minutes. For film 1624c, the muscle was put into Ringer at 45°C for 11 minutes, and then returned to Ringer at 8°C, and exposed for 21 hours.

- Muscle 1700. Also in 1977, on a 17cm camera, film 1700c was an 11 hour 20 minute exposure of the relaxed muscle at a sarcomere length of 4μm, and a temperature of 3°C. The muscle was later put into frog Ringer with 1mM iodoacetate, to induce rigor, and exposed for 11 hours 15 minutes at 2.5°C, for film 1700m, the fifth exposure of the experiment.

- Muscle 1697. In 1977, on a 17cm camera, film 1697f is a 15 hour 21 minute exposure of the rigor muscle, in frog Ringer, with 1mM iodoacetate. Only one exposure from this experiment was analysed.

- Muscle ORU17. Done in 1981 on a 36cm camera, film ORU21d was an 18 hour 40 minute exposure of the rigor muscle in frog Ringer, with 1mM iodoacetate at a sarcomere length of 2.3μm. Only one exposure from this experiment was analysed.

- Muscle ORU21. Done in 1981, on a 36cm camera, film ORU21a is an exposure of the relaxed muscle at a sarcomere length of 4.1μm. Film ORU21d is the rigor muscle, at 4.1μm, in frog Ringer with 1mM iodoacetate. The exposure times are not recorded.

6.3 Films from the Daresbury experiments

The September Daresbury experiments produced only two exposures, one of which was useless because of radiation damage to the muscle specimen. There was no camera blank exposure. Therefore the data was not processed.

The second Daresbury trip, in November 1988, was more sucessful, producing a number of muscle patterns, with blank exposures of both the muscle cells used. However, the cell blanks, when examined, proved to have an asymmetric component, in the form of diffuse blobs in two opposing quadrants of the pattern. This scatter was different for each cell, and therefore must arise from the mylar windows of the cells, and not from the camera. The muscle cells are circular, and consist of two parts, one of which contains a frame to which the muscle is tied. The two parts each have a mylar window, between which the muscle sits. The two parts of the cell are fixed together with three screws, and can have any of three relative orientations, and the cell can sit on the camera at any angle. During the experiment, the two halves of the cells were not always fitted together at the same relative orientation, since this was not recognized as being of importance, and the cells were not always mounted in the beam at the same angle. Thus, not only is the orientation of the blanks incorrect, but if the scatter is a sum of scatter from each of the two windows of the cell, the intensity distribution may vary from one exposure to another. However, since the muscle pattern is spherically averaged for the calculation of the radius of gyration of the myosin heads, and since the blank intensity was small compared with the muscle pattern—less than 10% (see Figure 6.1)—the orientation of the blank is not important. Symmetry measurements, however, will be untrustworthy.

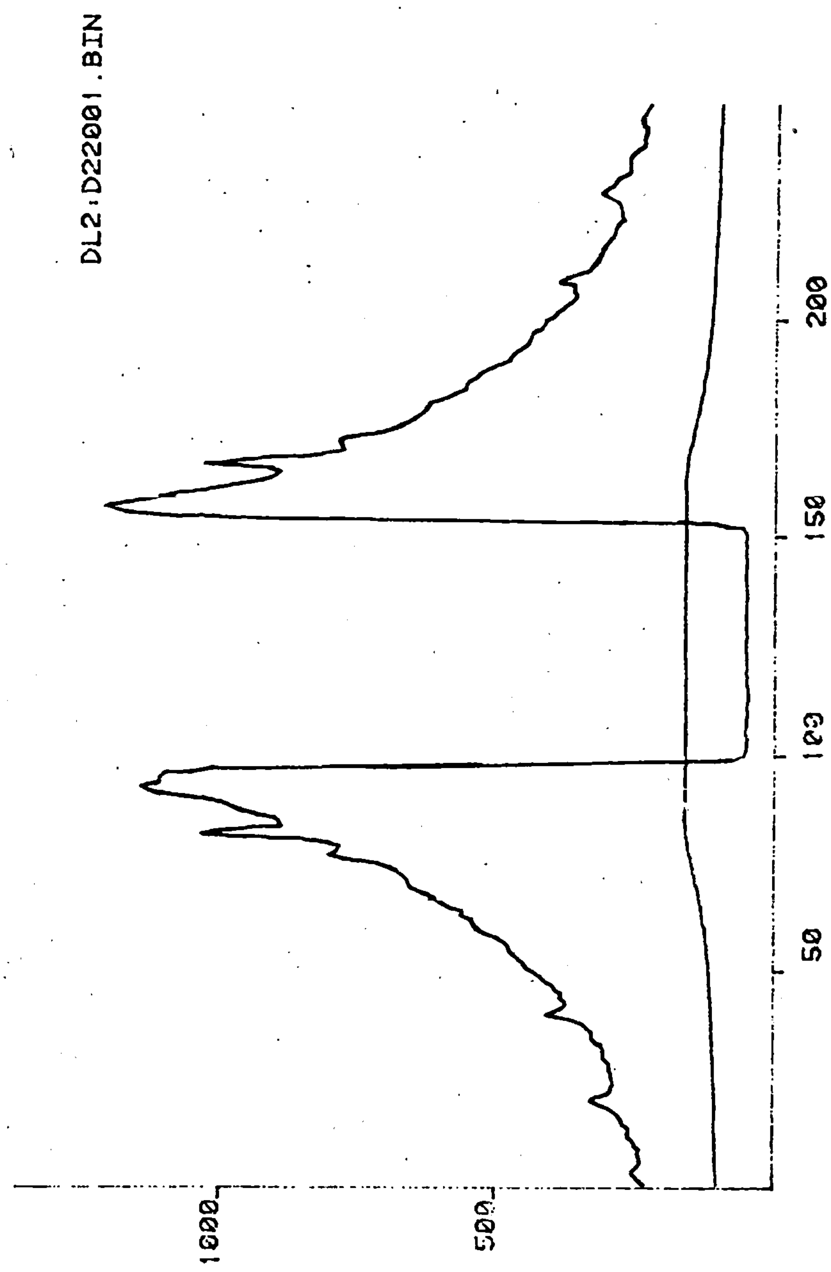


Figure 6.1 A meridional scan through Daresbury film 22a, of heated frog muscle, superimposed on the processed camera blank, showing their relative intensities. Note that the variation in the camera blank is small compared with that in the muscle data.

The camera blanks showed no streaks or other irregularities—they were circularly symmetrical except for the scatter from the mylar windows of the muscle cells, and, of course, the beam stop shadow.Therefore, the cell blanks were scanned over a large area and then circularly averaged, with the beam stop shadow excluded, using the program CIRAVG (a variation of SPHAVG—see Section 4.6). The program MIRROR was used to create a data set equivalent to a single scan with the pattern centre in the centre of the scan, and the backstop shadow was then fitted over. It was decided that, since the data near the edges of the shadow was the average of only a few points, and was therefore imprecise, a second order polynomial, which is very sensitive to those points, should not be used, and a straight line fit was used. The data was then converted back to a half-scan, with the pattern centre at the beginning of the scan, using MIRROR, and the program UNSA was used to re-create a two dimensional data set of size and centre position corresponding to each muscle pattern that was processed. These re-constituted data sets were used as the camera blanks, each being scaled by the ratio of the second ionization chamber readings for the muscle and the blank.

After subtraction of the cell blanks, the muscle patterns were analysed in the same way as the frog patterns in the previous section. The data was converted to log intensity, the layer lines removed and the beamstop fitted over, then it was converted back and averaged. A background intensity found with the program POROD was subtracted to allow for the difference in the film fogging levels of the data and the blank where the blank was scaled before subtraction.

Three patterns were analysed:

- Film 14a: rigor, sarcomere length 2.45μ.
- Film 20a: rigor, sarcomere length 2.5–2.7μ.
- Film 22a: heated, sarcomere length 2.2–2.4μ.

The beam stop was very large (see Section 5.3), so that the shadow extended nearly to the Guinier point in the meridional direction, and beyond it in the equatorial direction (see Figure 6.2). Therefore, the data at the Guinier point in the spherical average is mostly fitted data, and this increased the unreliability. The R_g values found from the three films analysed did not agree with the other frog muscles processed in this chapter (compare Table 6.3, the Daresbury films, with Table 6.2, the frog muscle exposures from the Lowy/Poulsen archives). The rigor muscles from the Daresbury experiments gave a significantly higher R_g value than those from the Lowy/Poulsen archives, which may have been due to denaturing of the S1. All the rigor muscles in the Daresbury experiments were treated in the same way and with the same batch of iodoacetate solution; it is possible that the solution was impure or that the muscles were exposed to it for too long. The heat-treated muscle gave an R_g of 3.0nm, which agrees well with most muscles from the Lowy/Poulsen archive, but is lower than the heated muscle from that source. The heat treatment of the muscle exposed at Daresbury consisted of maintaining the muscle at 39°C for 10 minutes—this was probably too low a temperature to denature the muscle, especially when one considers that mammalian muscle functions normally at this temperature. The experiments could not be repeated since no more beam time was available. In view of these discrepancies, and the unreliability of the data due to the beam stop shadow, the remaining exposures from Daresbury were not processed.

6.4 Results

1. ABRM patterns. See Table 6.1. The mean R_g value for relaxed ABRM is 3.1nm, with a standard deviation of 0.3nm (three muscles). For isotonically contracting ABRM, the R_g value is 3.2nm (one muscle).

2. Frog patterns. See Table 6.2. Using the values calculated after subtraction of the intensity level given by the program POROD, the mean R_g value for relaxed muscle is 2.6nm with a standard deviation of 0.2nm (six muscles). For stretched muscle, the mean R_g is 2.6nm, st. dev. 0.1nm (three muscles), for heated muscle, it is 4.0nm (one muscle)

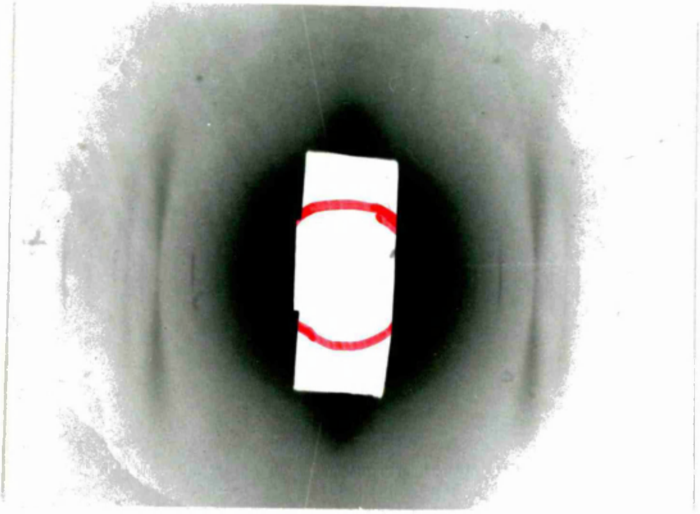


Figure 6.2 A print of film 14a, of rigor frog muscle from the Daresbury experiments, showing the Guinier point position and the beam stop shadow.

Film	Muscle State	R_g value	R_g value
		No background	$0.22nm^{-1}$ bg
1646a	Relaxed	2.6nm	3.3nm
1445b	Relaxed	2.7nm	3.3nm
1627a	Relaxed	—	2.7nm
1628a	Isotonic	—	3.2nm

Table 6.1 ABRM R_g results—the intensity at $0.22nm^{-1}$ was subtracted instead of a camera blank.

Muscle	Treatment	R_g $0.22nm^{-1}$ Bg		R_g POROD Bg		
		Relaxed	Treated	Relaxed	Treated	δR_g
1688	Stretched	2.8nm	2.9nm	2.7nm	2.7nm	0.0nm
1689	Stretched	2.7nm	2.7nm	2.5nm	2.5nm	0.0nm
1690	Stretched	2.9nm	2.7nm	2.7nm	2.5nm	-0.2nm
1624	Heated	—	—	3.2nm	4.0nm	0.8nm
1700	Rigor	2.8nm	2.6nm	2.7nm	2.6nm	-0.1nm
ORU21	Rigor	2.7nm	3.4nm	2.6nm	3.3nm	0.7nm
1697	Rigor	—	—	—	3.0nm	—
ORU17	Rigor	—	—	—	3.1nm	—

Table 6.2 Frog R_g results—a constant intensity was subtracted instead of a camera blank; either the intensity at $0.22nm^{-1}$ or that calculated with the program POROD. The differences in R_g between the different states of the muscle are also given.

and for rigor muscle, 3.0nm, st. dev. 0.3 (four muscles). Comparing the R_g values of the same muscle in different states, the mean difference in R_g when a muscle is stretched is −0.07nm, st. dev. 0.09nm (three muscles), when it is heated, 0.8nm (one muscle) and when a muscle is put into rigor, the mean difference in R_g is 0.3nm, st. dev. 0.4 (two muscles).

3. The Daresbury experiments. See Table 6.3. The R_g results are inconsistent with the other films analysed, but there are plausible reasons for this.

Film	State	R_g value
14a	Rigor	4.2nm
20a	Rigor	3.8nm
22a	Heated	3.0nm

Table 6.3 The Daresbury pattern R_g results—treated camera blanks were subtracted.

6.5 Conclusions

The Daresbury results can be discounted, for reasons which have been discussed. The muscle patterns from the Lowy/Poulsen archive show R_g values of about 3nm, with no significant variation between different states of muscle, except for heated muscle. One pattern from heated muscle was available, and the same muscle in the relaxed state showed an unusually high R_g value, but it appears that the R_g value increases when the muscle is heated. This means that the myosin heads increase in size, which is what one might expect to happen when a protein is denatured.

Chapter 7

Analysis of Specific Experiments from the Lowy/Poulsen Archives

The next step in the project was to use the programs that had been developed to study some complete experiments from the Lowy/Poulsen archives. These experiments date from 1977, and involve exposures of one muscle in varying experimental conditions.

These experiments had been designed to compare the effects of reducing the interaction between the actin and myosin filaments in two different ways; firstly by stretching the muscle to non-overlap, so preventing any interaction between the filaments, and secondly by increasing the osmolarity of the bathing solution, which was intended to increase the ionic strength of the solution between the filaments and to shield the electrostatic interactions between them (J.Lowy, personal communication).

The aim of my analysis was to retrieve as much information as possible from these patterns, both from the sharp reflections and the diffuse scatter. These films were measured on the Joyce Loebl Scandig 3 densitometer at Daresbury Laboratory, allowing measurement of the changes in the myosin sharp reflections as well as in the radius of gyration of the heads (see Chapter 2). The diffuse scatter in some of these films had already been analysed using scans done on the LKB densitometer (Chapter 6). These films were analysed without reference to any previous measurements, and the results for the radius of gyration measurements on these films are thus listed twice in this thesis.

7.1 The experiments

The experiments are listed in Table 7.1. Each used a single frog muscle, which was exposed under various conditions, either the sarcomere length or the osmolarity of the bathing solution or both being changed to vary the actin-myosin interaction.

7.2 The analysis

These films were analysed using the programs written on the PDP 11. They were scanned on the Joyce-Loebl Scandig 3 densitometer at Daresbury Laboratory. Changes in the intensities of the sharp reflections were measured, as well as the radius of gyration of the myosin heads. The original intention was to measure the widths of the meridionals with the program WIDTH, but it was found to be impossible to measure a consistent value since many of the lines were clearly composed of a number of partially resolved reflections.

After scanning, the data was transferred to the PDP 11 for processing.

First, the data set was calibrated and the pattern centre located using the program LKBPK. The alignment of the pattern with the scanning direction was checked at this stage, and the minimum of the 14.3nm line in the equatorial direction was found, giving the width of the meridional myosin peaks. All patterns were found to be aligned on the densitometer with sufficient accuracy, so that the deviation in position over the width of the film was less than the width of the narrowest peak.

Some scans contained peaks that were not visible on the film, and which may have arisen from dust particles settling on the film during densitometry. These peaks were removed before further processing.

Next, the myosin lines were fitted. The four quadrants of the data were added together, reducing noise levels and allowing a better fit to be obtained. The data was then converted to log. form, and the peaks were fitted with a

Film	Exp	Time	Film N°	S.L.	Osm	Temp. °C	Camera length	Type	Date
1688b	1	17:00	2	2.45	—	3	36cm	Il.	17/01
1688e	2	18:37	2	3.38	—				
1688i	3	18:37	3	2.45	—				
1689c	1	17:00	3	2.60	—	3	36cm	Il.	20/01
1689f	2	16:25	3	3.80	—				
1689i	3	23:00	3	—	—				
1689j	4	08:00	1	2.60	0				
1689o	5	13:00	2	2.60	0				
1711b	1	—	2	2.47	225	2	36cm	D.S.	15/03
1711d	2	13:45	1	3.86	459				
1711g	3	07:45	1	3.86	459				
1711k	4	15:45	2	3.86	225				
1714b	1	03:15	2	4.26	225	3	17cm	D.S.	18/03
1714f	2	13:40	3	4.26	225				
1714h	3	—	2	4.26	450				
1714k	4	03:45	2	4.26	225				
1714o	5	—	3	4.26	225				
1714q	6	14:38	2	4.26	225				
1714u	7	08:38	3	4.26	225				
1715b	1	—	2	2.58	225	2	36cm	D.S.	22/03
1715d	2	11:55	2	2.58	450				
1715h	3	15:55	2	2.58	225				
1715j	4	18:15	1	2.58	225				
1716c	1	—	3	2.58	225	2	17cm	D.S.	22/03
1716f	2	11:55	3	2.58	450				
1716i	3	15:50	3	2.58	225				
1716j	4	18:15	1	2.58	225				
1717c	1	20:40	3	2.66	225	2	17cm	D.S.	25/03
1717e	2	22:00	2	2.66	337				

Table 7.1 The experimental conditions. Where the exposure time is not given, the X-ray source failed during the exposure. Each pattern was recorded on three films; the 'first' film is the most heavily exposed, the 'third' the least. The sarcomere length for 1689i was not recorded; the muscle was shortened to a length corresponding to about 2.8μ. Where the osmolarity is not given, the solution is at physiological osmolarity. 1689 j and o were in distilled water. The temperature typically varied by ±1°C during an experiment. Type Il. is iliofibularis muscle, the muscle being cut in half to reduce the thickness. D.S. stands for dorsal semitendinosus.

first order polynomial in all scans using the program LKBDEL. After fitting, the data was converted back to linear intensity form (see Figure 7.1). The resulting data file was used for the R_g measurement, and also was subtracted from the raw data leaving only the sharp reflections.

The R_g measurement was done in the same way as for films scanned on the LKB densitometer; the quadrant-added data between 20° and 90° from the equator was spherically averaged with the program SPHAVG, a background level found with POROD was subtracted, and the R_g value measured with LKBRG (see Figure 7.2).

The intensities of the meridionals and the layer lines were measured with the program INTEG.

The intensity of the 5.9nm actin layer line was used for calibration, since it appears to remain constant [18]. This line is diffuse and of low peak intensity, so that the measurement has a high uncertainty. To reduce this, scans were added together in groups of four, smoothing the quadrant-added data further, and improving the contrast. The resulting data set was converted to log. form, the actin line was fitted over in all scans using LKBDEL, using either a first or a second order polynomial, to obtain the best possible fit; usually a second order polynomial. The data was then converted back to linear form, subtracted from the raw data, and the intensity measured with INTEG.

The intensity values for the myosin reflections were normalized by dividing them by the actin layer line intensity.

An estimate of the error in the actin line integrated intensity can be made by measuring the intensity in scans near the meridian, where the line is not present. This should be less than the minimum theoretical uncertainty, of a $\frac{1}{2}$ unit per channel in the region fitted. The uncertainty in this line is much greater than for the meridionals, which have a higher peak intensity and cover a much smaller area. The myosin layer lines are also diffuse and of low intensity, but they are narrower than the actin line. The actin line measurement limits the precision of all measurements, since it is the calibration factor. The estimated error in the R_g measurement is 10% (see

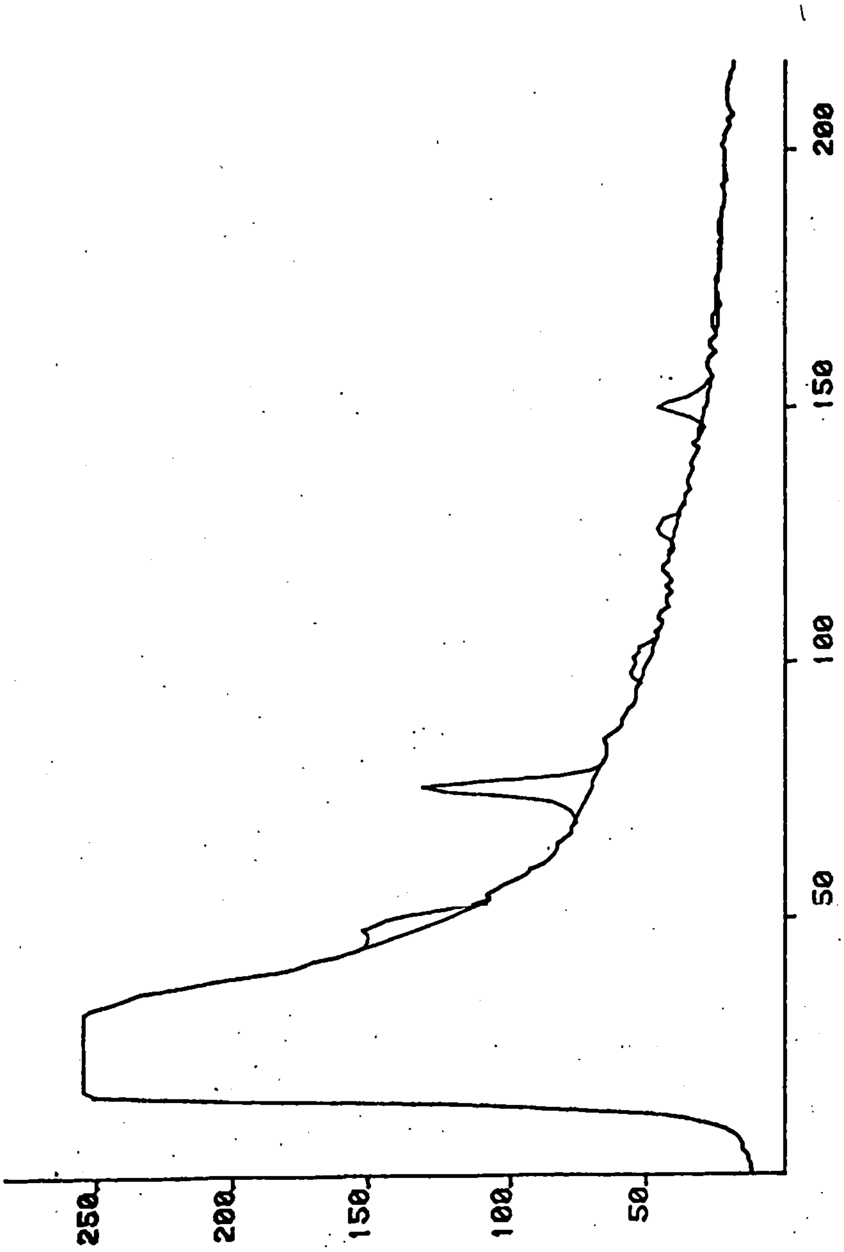


Figure 7.1 A plot through the meridian of film 1688e, showing the fit under the sharp reflections.

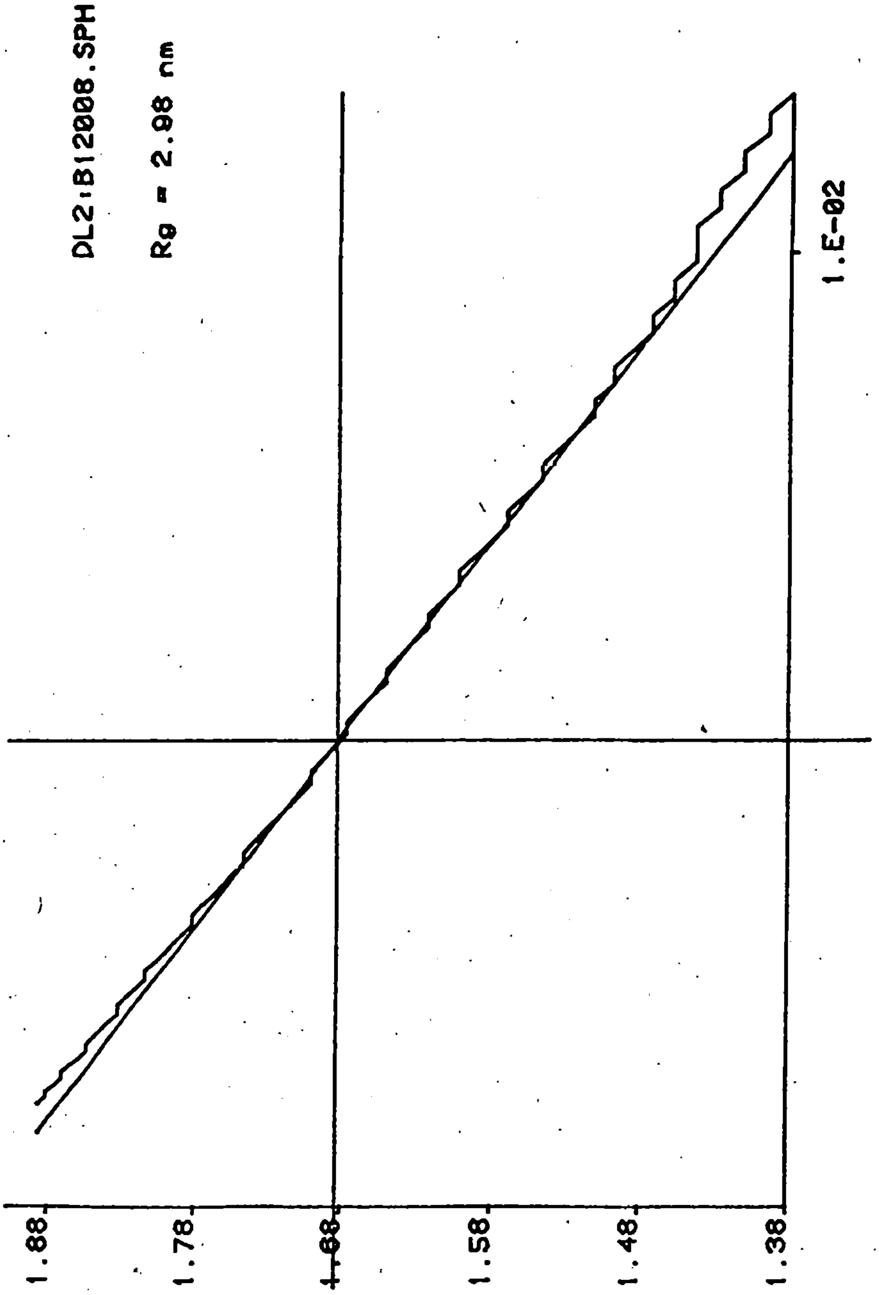


Figure 7.2 The output of the program LKBRG on film 1688e

Section 4.8).

7.3 Results

The results of the measurements of the meridional reflections are given in Table 7.2. The following trends can be seen:

1. Stretching the muscle to non-overlap leads to a reversible drop in intensity of all the lines measured—1688 and 1689 c, f and i. Thus the order of the myosin heads appears to deteriorate in the absence of the actin filaments.

2. An osmolarity increase leads to a reversible increase in the lines at 14.3nm and 7.2nm, and no change or a decrease in the other lines. This is seen in 1714 and 1716; in 1711 the sarcomere length was increased at the same time as the osmolarity increase; in 1715 the expected change is seen in the 7.2nm line, while a larger decrease is seen in the 10.7nm and 8.6nm lines, and a decrease less than the uncertainty in the measurements is seen in the 14.3nm line. An osmolarity increase thus seems to improve the helical order of the myosin heads, increasing the intensity of orders of the 14.3nm reflection, and decreasing the forbidden meridionals. This effect is different to the effect of removing actin by stretching to non-overlap.

3. An increase in intensity of the 14.3nm and 7.2nm lines in subsequent exposures of a muscle in identical experimental conditions is seen in 1711 d and g, and 1714 k and o. In these cases, a change in conditions took place immediately before the first of the two exposures, indicating that after a change in conditions the intensity of these lines may rise gradually over a period of time. The other lines could not be measured in these films.

4. Superimposed on the above changes is a progressive decrease in the intensities of all the lines, either due to increasing exposure to X-rays,

or time elapsed since dissection. This is probably due to decreasing order as muscle damage occurs.

5. Large variations in the peak intensities are seen from one muscle to another, even where the muscle is exposed in the same conditions and where the exposure is the first one.

It should be noted that the above changes might be due to changes in the actin 5.9nm line used for calibration, rather than changes in the myosin lines themselves. Huxley and Brown[18] in their comprehensive study of the X-ray diffraction pattern of frog muscle were not able to detect changes in this reflection in rest, rigor or contraction, however.

The layer line measurements are given in Table 7.3. The changes in these lines are difficult to assess since the uncertainties are very high—in many cases, comparable to the measured intensities—and no conclusions can be drawn from the results.

The R_g values in Table 7.4 show the following trends:

1. There is no significant change with changing sarcomere length (1688 and 1689 c, f and i).

2. An increase in osmolarity causes a decrease in the R_g; this can be seen in 1715, and also in 1711, though in 1711 the change is less than the estimated uncertainty. 1714, 1716, and 1717 all show a decrease in R_g with increased osmolarity, but these values are not trustworthy, as the beamstop shadow obscured part of the Guinier region, so not much importance should be attached to them.

3. Putting the muscle into distilled water causes a marked and progressive increase in R_g, 1689 j and o.

4. The average R_g value for all normal osmolarity exposures taken on a 36cm camera was 3.0nm, standard deviation 0.1nm (11 exposures). This agrees well with previous measurements, and the standard deviation of 0.1nm agrees with the estimated uncertainty of 10% (three standard deviations).

The films 1688 a and d and 1689 a and d were analysed separately after scanning on the LKB densitometer at Foxcombe Hall (see Chapter 6, Table6.2). These films are from the same exposures as 1688 b and e and 1689 c and f, analysed as part of the experiments described in this chapter (three films were used for each exposure, mounted one behind the other). The comparison between these films is given in Table 7.5. The results show a systematic discrepancy, with the data from the LKB giving lower R_g values than that from the Joyce-Loebl. This may reflect a deviation from linearity in one of the densitometers, a difference in the processing, or a difference between the films used. The manufacturer's specifications for both densitometers state that the intensity response is linear up to the maximum; this is difficult to test properly in the laboratory, but is likely to be true (at least for slowly-varying intensity values—see Chapter 2). The LKB data was analysed before that from the Joyce-Loebl, and the experience gained was used to improve the processing method, but this is unlikely to result in a *systematic* difference in the results. As for a difference in the films, in each case a more heavily exposed film was used for the LKB measurement, since the LKB densitometer can measure a greater range of intensities than the Joyce-Loebl, but is less sensitive. These films may have been over-exposed, so that the photographic material did not respond in a linear manner in the centre of the pattern. The result of such over-exposure would be a reduction in the intensity of the central regions of the pattern compared with the outer parts, giving a reduction in the magnitude of the gradient of the Guinier plots, and a lower R_g value. The more heavily exposed of the patterns compared here do give lower R_g values.

7.4 Conclusions

The original aim of these experiments was to compare the effects of reducing the actin-myosin interaction in two ways; by stretching to non-overlap, and by increasing the osmolarity of the bathing solution. The measurements described in this chapter show that these two operations have a different effect on the myosin heads, though the results are not conclusive. Stretching appears to cause a reversible decrease in the order of the myosin heads (Section 7.3, item 1), while an osmolarity increase appears to improve their helical order (Section 7.3, item 2).

The purpose of the analysis described here was to investigate the possibility of recovering data from existing patterns in the Lowy/Poulsen archive. As we have seen, the uncertainties in the measurements are high, so that results are not conclusive. Exposures taken on the shorter, 17cm, camera are of no use for R_g measurements, because of the large beam stop shadow, which also obscures the first and second order meridional reflections from myosin. A statistical analysis might have reduced uncertainties enough to allow useful conclusions to be drawn from the data, but unfortunately, although there are a large number of experiments in the archive, there are few repetitions of similar experiments, so a statistical analysis was not possible. For these reasons, the value of the archive experiments is limited.

Film	SL	Osm	Meridionals:				
	μ		21.5nm	14.3nm	10.7nm	8.6nm	7.2nm
1688b	2.5	—	0.39±.07	1.30±.12	0.20±.04	0.38±.09	0.24±.04
1688e	3.4	—	0.20±.04	0.64±.06	0.06±.03	0.07±.03	0.18±.04
1688i	2.5	—	0.37±.04	0.76±.07	0.16±.02	0.19±.02	0.19±.02
1689c	2.6	—	1.22±.13	2.28±.23	0.49±.06	0.68±.09	0.44±.06
1689f	3.8	—	0.46±.06	0.88±.10	0.20±.03	0.18±.03	0.23±.03
1689i	—	—	0.69±.08	1.08±.12	0.27±.04	0.27±.03	0.31±.04
1689j	2.6	0	0.41±.05	0.87±.10	0.19±.03	0.13±.02	0.18±.03
1689o	2.6	0	—	0.27±.12	0.96±.31	0.50±.17	—
1711b	2.5	225	0.65±.17	1.42±.32	0.38±.12	0.36±.10	0.26±.09
1711d	3.9	459	—	1.31±.11	—	—	0.57±.11
1711g	3.9	459	—	1.83±.17	—	—	0.61±.20
1711k	3.9	225	—	1.09±.09	—	—	0.30±.04
1714b	4.3	225	—	0.66±.04	0.16±.01	0.05±.01	0.27±.02
1714f	4.3	225	—	0.45±.03	0.17±.02	0.10±.01	0.24±.02
1714h	4.3	450	—	0.51±.02	0.19±.02	—	0.30±.02
1714k	4.3	225	—	0.46±.03	0.09±.01	—	0.24±.02
1714o	4.3	225	—	0.55±.04	—	—	0.31±.03
1714q	4.3	225	—	0.38±.02	0.09±.01	0.02±.01	0.14±.01
1714u	4.3	225	—	0.26±.04	—	—	—
1715b	2.6	225	1.18±.21	2.51±.42	0.42±.10	0.36±.10	0.42±.09
1715d	2.6	450	0.24±.02	2.08±.11	0.16±.01	0.19±.02	0.62±.05
1715h	2.6	225	0.57±.08	2.00±.23	0.28±.06	0.41±.08	0.33±.06
1716c	2.6	225	—	0.98±.05	0.17±.01	0.24±.02	0.31±.02
1716f	2.6	450	—	1.60±.06	0.18±.02	0.23±.02	0.61±.03
1716i	2.6	225	—	0.81±.03	0.11±.02	0.18±.02	0.21±.02
1716j	2.6	225	—	0.48±.03	—	—	0.10±.01
1717c	2.7	225	—	—	0.17±.01	0.21±.01	0.21±.01
1717e	2.7	337	—	0.97±.03	0.11±.01	0.26±.01	0.27±.01

Table 7.2 The intensities of the meridional reflections. In patterns taken on a 17cm camera (1714, 1716 and 1717) the 21.5nm line is obscured by the beamstop shadow. Other lines were sometimes too weak to be measured. 1715j was under-exposed, so that the actin 5.9nm calibration line could not be measured.

Film	SL	Osm	Layer Lines:				
	μ		21.5nm	14.3nm	10.7nm	8.6nm	7.2nm
1688b	2.5	—	0.16±.40	0.62±.31	0.25±.28	0.00±.50	-.11±.20
1688e	3.4	—	0.05±.19	0.35±.26	0.23±.26	0.03±.19	0.00±.23
1688i	2.5	—	0.25±.09	0.67±.12	0.27±.09	0.19±.07	0.12±.06
1689c	2.6	—	0.44±.13	1.13±.23	0.55±.17	0.62±.23	0.37±.19
1689f	3.8	—	0.54±.14	0.92±.20	0.46±.14	0.32±.11	0.16±.08
1689i	—	—	0.72±.24	1.13±.23	0.66±.21	0.46±.18	0.22±.12
1689j	2.6	0	0.50±.21	0.96±.25	0.33±.16	0.21±.13	0.09±.11
1689o	2.6	0	—	0.50±.23	0.96±.37	0.48±.21	—
1711b	2.5	225	0.56±.50	0.89±.44	0.42±.45	0.07±.31	0.10±.28
1711d	3.9	459	—	0.32±.05	—	—	0.15±.04
1711g	3.9	459	—	0.32±.08	—	—	0.16±.07
1711k	3.9	225	0.25±.08	0.50±.09	0.45±.10	0.23±.07	0.10±.06
1714b	4.3	225	—	0.25±.06	0.55±.07	0.24±.05	0.12±.03
1714f	4.3	225	—	0.38±.06	0.32±.19	0.18±.05	0.11±.03
1714h	4.3	450	—	0.24±.05	0.18±.07	—	0.09±.05
1714k	4.3	225	—	0.19±.07	0.10±.05	—	0.05±.06
1714o	4.3	225	—	0.29±.10	—	—	0.07±.09
1714q	4.3	225	—	0.18±.06	0.18±.06	0.07±.06	0.05+.06
1714u	4.3	225	—	—	—	—	—
1715b	2.6	225	0.23±.15	0.73±.27	0.47±.20	0.28±.24	0.11±.11
1715d	2.6	450	0.10±.06	0.73±.14	0.45±.11	0.38±.12	0.18±.15
1715h	2.6	225	0.48±.23	1.33±.40	0.76±.38	0.53±.42	0.28±.26
1716c	2.6	225	—	0.66±.07	0.45±.05	0.33±.05	0.17±.05
1716f	2.6	450	—	0.54±.07	0.38±.09	0.36±.09	0.04±.08
1716i	2.6	225	—	0.58±.06	0.30±.15	0.23±.07	0.06±.08
1716j	2.6	225	—	0.33±.06	—	—	0.01±.04
1717c	2.7	225	—	1.10±.07	0.52±.04	0.34±.03	0.28±.04
1717e	2.7	337	—	0.81±.07	0.49±.07	0.37±.07	0.20±.08

Table 7.3 The intensities of the layer lines. The errors are very high compared with the intensity values, because the lines are diffuse and of low intensity.

Film	SL μ	Osm	R_g nm
1688b	2.5	—	3.01
1688e	3.4	—	2.98
1688i	2.5	—	2.80
1689c	2.6	—	2.90
1689f	3.8	—	2.91
1689i	—	—	3.08
1689j	2.6	0	3.45
1689o	2.6	0	3.68
1711b	2.5	225	2.88
1711d	3.9	459	2.73
1711g	3.9	459	2.83
1711k	3.9	225	2.98
1714b	4.3	225	3.0 *
1714f	4.3	225	2.9 *
1714h	4.3	450	2.8 *
1714k	4.3	225	3.2 *
1714o	4.3	225	3.0 *
1714q	4.3	225	3.1 *
1714u	4.3	225	3.6 *
1715b	2.6	225	3.14
1715d	2.6	450	2.58
1715h	2.6	225	3.10
1715j	2.6	225	3.13
1716c	2.6	225	2.6 *
1716f	2.6	450	2.5 *
1716i	2.6	225	3.0 *
1716j	2.6	225	3.2 *
1717c	2.7	225	2.8 *
1717e	2.7	337	2.4 *

Table 7.4 The radii of gyration of the myosin heads. In patterns taken on a 17cm camera (1714, 1716 and 1717), part of the Guinier region is obscured by the beamstop shadow, so the R_g value is inaccurate. These values are marked with an asterisk. Otherwise, the uncertainty is about 10% .

LKB densitometer				Joyce Loebl			
Pattern	Exposure	Film	R_g	Pattern	Exposure	Film	R_g
1688a	1	1	2.7	1688b	1	2	3.0
1688d	2	1	2.7	1688e	2	2	3.0
1689a	1	1	2.5	1689c	1	3	2.9
1689d	2	1	2.5	1689f	2	3	2.9

Table 7.5 Comparison between R_g values measured at different times from scans done on different machines—the LKB densitometer at Foxcombe Hall, and the Joyce-Loebl at Daresbury Laboratory.

Chapter 8

Discussion

In the introduction to this thesis five objectives were set out. The degree to which these objectives have been met will be discussed here, and suggestions made for future work to follow on from this project.

8.1 The success of this project

The first aim was to set the analysis of the diffuse scatter on a firm footing, reducing processing errors, and justifying the use of the Guinier approximation. The use of the Guinier approximation was discussed in Chapter 3, and it was argued that S1 in muscle could behave as if in solution as far as X-ray diffraction is concerned, and that the fact that the Guinier plot is linear demonstrates that this is in fact the case, as does the agreement found by J.Lowy and F.Poulsen between the diffuse scatter from muscle and that from S1 in solution which was measured by R.Mendelson and K.Kretzchmar. The reduction in processing errors was achieved by computer analysis of the patterns, enabling a larger number of data points to be carried through the analysis procecc. Each program was written with a view to minimizing errors, both random and systematic. These programs are described in Chapter 4.

The second objective was to increase the processing rate and thus the throughput of both film and detector data at the Oxford Research Unit, by developing a set of programs for computer analysis of data. To meet this objective, the LKB densitometer was connected to the PDP 11 computer, and a set of programs for data analysis was developed. The use of the computer enabled more complex and detailed data processing than was possible otherwise; in particular, the suite of programs enabled a measurement of the radius of gyration of the myosin heads to be made. The time required for processing each film was greatly reduced by use of the computer. Films densitometered at Daresbury Laboratory can be read onto the computer, and processed in the same way as films read from the LKB, and detector data, when this becomes available, can be treated in exactly the same way as film data.

The third objective was to use the set of programs developed to study the diffuse scatter. Experiments done at Daresbury Laboratory were analysed, with limited success (see Chapter 5). Various films from experiments in the Lowy/Poulsen archive were analysed (Chapter 6), including some complete experiments done in 1977 (Chapter 7), and the results indicate that the radius of gyration of S1 in whole muscle is independent of sarcomere length, at 3.0nm, but shows an inverse relationship to the osmolarity of the bathing solution. The R_g also appears to increase when the muscle is heated. However, the uncertainties in the R_g values were large, the changes seen were generally small, and the number of films from each type of experiment was also small, so that these results are not conclusive. The increase in the R_g value when the muscle was immersed in distilled water *was* significant, as it was much larger than the probable error.

The fourth objective was to study the sharp reflections from archive data, again using the computer programs. The films from the 1977 experiments were used for this purpose (Chapter 7). Again, the uncertainty of the data and the small number of films in each type of experiment meant that the results were not conclusive, but merely indications of likely changes. These results are described in more detail in Chapter 7; to summarize, stretching the muscle seems to cause a reversible deterioration in the order of the myosin heads, increased osmolarity appears to improve their helical arrangement, the intensity of the sharp reflections may increase with time after a change in conditions, the intensity deteriorates over an experiment, and there is a large variation from one muscle to another—at least when using the actin 5.9nm line as calibration. Further experiments need to be done to confirm or modify these observations.

The final objective was to consider the conditions of data aquisition and to see whether it is feasible to recover information from archive patterns taken before interest developed in the diffuse scatter. Although the experiments done at Daresbury Laboratory were unsuccessful insofar as little useful data was acquired, some observations were made which may help future experiments, in particular, the exposure necessary to obtain the diffuse scatter and the maximum exposure that the muscle can survive without significant damage were measured (see Chapter 5). The R_g measurements made on these films are described in Chapter 6; the accuracy was poor, but the value of 3nm for unheated muscle agrees with solution scattering data, and the higher value for heat-treated muscle is reasonable. The conclusions drawn from the 1977 experiments, studying both the diffuse scatter and the sharp reflections, are discussed above. Some suggestive changes were seen, but the measurements were problematic. In the exposures taken on the shorter, 17cm, camera, the large size of the beam stop meant that the Guinier region was partially obscured, as were the first and second myosin layer lines near the centre of the pattern. Thus the R_g value from these films was considered to be so unreliable as to be almost worthless, and changes in the first two layer lines could not be measured at all. In patterns taken on the longer, 36cm, camera, only the first layer line was obscured, and the R_g measurement was unaffected by the beam stop. However, no exposures of the camera scatter were available, so the absolute value of the R_g, and comparisons between exposures from different experiments, are likely to be inaccurate. The measurement of the sharp reflections was confused by the difficulty in measuring the actin 5.9nm layer line, which was used to calibrate the intensity measurements. This line was broad and had a low peak intensity, compared with the myosin layer lines, and its measurement was therefore the limiting factor for the accuracy of the analysis. For all these reasons, the amount of infomation that can be gathered from these experiments in the Lowy/Poulsen archive is limited, and though the results are interesting, they are perhaps more useful as a guide to further experimental work than as a reliable indication of the behaviour of S1 in muscle. This is a disappointing conclusion, considering the amount of data in the archive files, but I believe it to be a realistic one.

8.2 Suggestions for future work

The data processing facility developed in this thesis could be improved in various ways. The programs were written primarily with film data in mind, since that is what was analysed at the time. Detector data can be treated in exactly the same way as the digitised densitometer measurements of a film pattern, but the error analysis could be made more sophisticated, since the error in the photon count is equal to the square root of the count. The errors in the individual data points could be stored in the same way as the data values themselves, and this error file could be updated at each stage in the analysis of the pattern, so that the uncertainty of the final results could be established much more accurately. Detectors available at Daresbury and DESY are continually being developed and improved, and are now much more reliable than at the start of this project.

The analysis of the diffuse scatter was restricted to the measurement of the radius of gyration of the myosin heads, as the symmetry of the scatter is difficult to measure even from a contour plot (see Section 4.9). A small deviation of an isointensity line from a circle is not easy to detect, especially when the line is distorted on the meridian and equator. The deviations from circular symmetry might be more easily detected if the isointensity lines were plotted as the variation in distance from the centre of the pattern against angle (R against θ). The line would then be approximately straight and horizontal, with 'humps' at 90° intervals. A compression of the data in the meridional direction, for example, would show up as a sinusoidal variation in R, so that the undistorted parts of the line, corresponding to data on neither the meridian nor equator, would be slightly sloping. This could be more easily seen than slight deviations from a circle. If the data were stored in (R,θ) form a more accurate fit over the meridian and equator could also be achieved. The sharp reflections are better fitted over in a normal (x,y) data set. At the present time, however, conclusions about the shape of the scatter should be treated with caution.

A problem likely to arise in implementing these improvements on the PDP 11 is the limited memory space available. Several of the programs in the set were affected by memory shortage, especially those for rotating a data set. The number of points that can be handled by the fitting routines is also limited by memory capacity, though the limits did not prove restrictive in the analyses that were done. If the program suite were to be developed significantly, it would be necessary to increase the memory available. In any serious development of my work, it would be best to transfer the programs to the VAX at Walton Hall (see Chapter 2), though it would be necessary to modify the plotting routines.

Suggestions for future experimental work on scatter arise from the analysis of the 1977 experiments. These are as follows.

1. Identical experiments should be carried out on a number of muscles, so that the results could be analysed statistically. In several of the experiments that were studied, both sarcomere length and osmolarity were varied, and this complicated the interpretation of the changes that occurred. It would facilitate matters if a single parameter were used, so that either (a) the sarcomere length were varied, while the muscle was immersed in solution of constant osmolarity, or (b) the osmolarity were varied with the muscle kept at a fixed sarcomere length.

2. The intensity calibration was the limiting factor to the accuracy of the measurement of the sharp reflections in the 1977 experiments. Improvements in the intensity calibration are therefore very important. The X-ray flux should be measured with ionization chambers placed before and after the muscle specimen; the difference in the readings would give the amount of X-rays absorbed by the specimen in the

exposure. The intensity scattered by the specimen holder alone would also be measured in a separate exposure, and the difference would give the intensity absorbed by the specimen, which would be proportional to the volume of muscle in the beam and so to the total intensity scattered. This would give a reliable intensity calibration. The use of a reliable detector with linear characteristics would greatly simplify this, since the total count could be used as the calibration factor.

3. Exposures of the scatter from the camera and sample holder should be taken at intervals, so that the diffuse scatter from S1 can be distinguished from that from the camera. This would enable the radius of gyration of S1 to be measured much more accurately. If film is used, the 'blanks' should be exposed for a similar length of time to the muscle exposures, and using film from the same packet, as otherwise scaling the blank to the muscle pattern whilst allowing for film fogging is difficult. The relative exposure times of muscle and blank would be unimportant if a detector were used.

The precision of the R_g measurements in this thesis was estimated to be about 10%, which agrees with the standard deviation in 11 measurements on archive data (Section 7.3). A reliable exposure of the camera scatter would improve the accuracy, but as there may be contributions from other parts of the muscle than S1 (see Section 1.3), and since the Guinier approximation may not apply so rigorously as for solution scatter, it may never be possible to achieve uncertainty levels as low as those quoted by Mendelson and Kretschmar for their measurements on S1 in solution[31], where the standard deviation in the mean was 0.06nm, giving an uncertainty of 0.18nm or about 6% (three standard deviations).

The precision of the measurements of the sharp reflections (Section 7.3) was more difficult to establish, as there were not enough similar experiments to allow a statistical analysis of the results. The main contribution to the uncertainty of the meridional reflections was the measurement of the actin 5.9nm layer line, which was diffuse and had a low peak intensity. The myosin layer lines were of a similar shape to the actin 5.9nm line, and there was a similar uncertainty in their measurement. The uncertainties here could be greatly reduced by following suggestions 1 and 2 in Section 8.2, to improve the statistics and the intensity calibration. Increasing the exposure would increase the data to noise ratio, and so improve precision, but this must be balanced against the risk of damage to the muscle. With improved data collection, repeated experiments, and a precise calibration, the errors in the meridionals could perhaps be reduced to 1–2%, and those in the layer lines to 5% or even less. However, these uncertainties do not take into account the variations in the intensity of the sharp reflections from one muscle to another, which may confuse the interpretation of the results and make a statistical analysis more complicated. These variations are larger than the uncertainties, and appear to reflect real differences between individual muscles.

8.3 Conclusion

In summary: the analysis of the diffuse scatter in X-ray patterns of whole muscle has been examined, and a method of analysis has been developed, together with the necessary computing tools. Various X-ray patterns have been analysed, and conclusions drawn about the behaviour of the myosin heads, though the reliability of these conclusions was limited by experimental accuracy. It was concluded that new experiments were needed, specifically designed to study the diffuse scatter, and suggestions were made for these experiments, and for expanding and improving the analysis method.

In 1983 J.Lowy and F.Poulsen concluded that the diffuse scatter from intact muscle contained structural information about the myosin heads in both resting and active muscle [27]. The work described in this thesis confirms that statement, though unavoidable limits to the accuracy of the analysis of the diffuse scatter may reduce the detail of the information about S1 that can be obtained by this method. The diffuse scatter represents another source of information about myosin heads in intact, active muscle, that has not yet been fully exploited.

Bibliography

[1] C.R.Bagshaw, 'Muscle Contraction', Chapman and Hall, 1982.

[2] P.H.Cooke, E.M.Bartels, G.F.Elliott and R.A.Hughes, Biophys. J. (1987), 51, pp. 947–957.

[3] R.Cooke, CRC Critical Reviews in Biochemistry, 21, 1, pp.53–118.

[4] R.Cooke, M.S.Crowder and D.D.Thomas, Nature, (1982), 300, pp. 776–778.

[5] G.F.Elliott, Proc. Royal Soc. (1964) B, 160, pp. 467–472.

[6] L.E.Ford, A.F.Huxley and R.M.Simmons, J. Physiol. (1981), 311, pp. 219–249.

[7] O.Glatter and O.Kratky, 'Small Angle X-ray Scattering', Academic Press, 1982.

[8] A.M.Gordon, A.F.Huxley and F.J.Julian, J. Phsiol. (1965), 184, pp. 170–192.

[9] A.Guinier, 'X-ray Diffraction', W.H.Freeman and Company, 1963.

[10] J.C.Haselgrove and H.E.Huxley, J. Mol. Biol. (1973), 77, pp. 549–568.

[11] J.C.Haselgrove, J. Mol. Biol. (1975), 92, pp. 113–143.

[12] A.F.Huxley and R.Niedergerke, Nature (1954) 173, pp. 971–977.

[13] A.F.Huxley, Progress in Biophysics, (1957), 7, pp. 255–318.

[14] A.F.Huxley, 'Reflections on Muscle', Liverpool University Press, 1980.

[15] A.F.Huxley and R.M.Simmons, Nature (1971) **233**, 5321, pp. 533–538.

[16] H.E.Huxley, Proc. Royal Soc. (1953) B, **141**, pp. 59–62.

[17] H.E.Huxley and J.Hanson, Nature (1954) **173**, pp. 978–987.

[18] H.E.Huxley and W.Brown, J. Mol. Biol. (1967) **30**, pp. 383–434.

[19] H.E.Huxley, K.C.Holmes and W.Brown, (1965)

[20] H.E.Huxley, J. Mol. Biol. (1968), **37**, pp. 507–520.

[21] H.E.Huxley, Science (1969) **164**, pp. 1356–1466.

[22] H.E.Huxley, Acta Anatomica Nipponica (1975) **50**, pp. 310–325.

[23] H.E.Huxley and M.Kress, J. Mus. Res. and Cell Mot. (1985) **6**, pp. 153–161.

[24] H.E.Huxley, R.M.Simmons and A.R.Faruqui, Biophys. J. (1989) **55**.

[25] M.Irving, J. Physiol. (1984) **352**, p64.

[26] J.Lowy and F.R.Poulsen, Nature, (1982), **299**, pp. 308–312.

[27] J.Lowy and F.R.Poulsen, Nature, (1983), **303**, 5913, pp. 146–152.

[28] J.Lowy and F.R.Poulsen, J. Mol. Biol. (1987) **194**, pp. 595–600.

[29] J.Lowy and F.R.Poulsen, Biophys. J. (1990) **57**, pp. 977–985.

[30] K.M.Kretzchmar, R.A.Mendelson and M.F.Morales, Biochem. (1978) **17**,12, pp. 2314 2318.

[31] R.A.Mendelson and K.M.Kretzchmar, Biochem. (1980) **19**, pp. 4103–4108.

[32] B.M.Millman and G.F.Elliott, Biophys. J. (1973) **12**, 11, pp. 1405–1414.

[33] F.R.Poulsen, J.Lowy, P.H.Cooke, E.M.Bartels, G.F.Elliott and R.A.Hughes, Biophys. J. (1987) **51**, pp. 959–967.

[34] R.W.Ramsey and S.F.Street, J. Cell & Comp. Physiol. (1940) **15**, pp. 68–70.

[35] J.M.Squire and P.J.Vibart, 'Fibrous Protein Structure', Academic Press, 1987.

[36] D.D.Thomas and R.Cooke, Biophys. J. (1980), **32**, pp. 891–906.

[37] S.Lalitha and C.R.Worthington, J. Mol. Biol. (1975), **96**, pp. 635–639

Data analysis and processing in X-ray diffraction studies of scattering from myosin heads in muscle: Appendix of programs.

M.C.Prestidge

26 April 1991

Contents

Appendix A

A description of the analysis program suite

This chapter descibes in detail the program suite available on the PDP 11 for analysis of X-ray data. The data files are described first, followed by the programs. The programs OTOKO and KOHALA for analysis of X-ray diffraction data, from EMBL, Hamburg, are also available, on the OU VAX at Walton Hall. Documentation for these programs is available from EMBL.

A.1 File Types

There are two types of file used by this set of programs: .DAT files and .BIN files. .DAT files store a data set as a collection of text (ASCII) files, one for each densitometer scan and one for the header information. They have the advantage that if a densitometer run fails part way through, not all the data is lost, but they are slow to read and write, and they take up a lot of disc-space.

A .BIN file is a binary direct access file, with records 256 integers long. The first record contains the header information, and subsequent records contain the scans. Each scan starts on a new record, though scans may be split accross records if each scan contains more than 256 integers.

The header parameters of a .BIN file are :

NY: Integer—the number of channels in each scan.

SCN: Integer—the number of scans in the file.

AMAX: Integer—the maximum intensity value in the file.

AMIN: Integer—the minimum intensity value in the file.

XSTEP: Integer—the separation of channels in units of 40μm.

YSTEP: Integer—the separation of scans in units of 40μm.

A0: Integer—the constant background intensity level, set by the user.

XSCAL: Real—the separation of scans, in reciprocal nanometres.

YSCAL: Real—the separation of channels, in reciprocal nanometres.

X0: Integer—the scan corresponding to the centre of the pattern.

Y0: Integer—the channel corresponding to the centre of the pattern.

There is also a third type of file—the .HAM file. These are used for transferring data to the Vax at Walton Hall. There are two .HAM files for each data set. The first is an ASCII file containing the header information, in the format used by the programs OTOKO and KOHALA from Hamburg. The second file contains the data, with one scan per record. This is also an ASCII file, as binary files cannot be transferred to the Vax, and it must be converted to a binary direct access file on the Vax before the Hamburg programs will operate on it.

A.2 Programs

There are various programs on the PDP/11 at the Oxford Research Unit for manipulating and for viewing the data. Most use .BIN type files, as these are the most convenient. Where the file type used by a program is not stated, you can assume that the program only works on .BIN type files. The programs available are listed in alphabetical order.

A.2.1 AMAX

The program AMAX searches a .BIN file for the maximum and minimum intensities, AMAX and AMIN, and replaces the old values in the header of the file. The new values, together with their positions in the file, are typed out on the screen, as well as the old values from the file header.

A.2.2 GUIPLT

This program reads in the first scan of a .BIN file, and plots out a part of it as a Guinier plot—a plot of log intensity against the square of the distance from the pattern centre. The user selects the first and last channels to be plotted. The y origin and scale are chosen to spread out the plot as much as possible. The x scale is also chosen to spread out the plot, but the x origin is always zero, as this makes for greater clarity. A hard-copy plot is produced. Divisions are marked on the axes, but they are not labelled—this information is displayed on the screen. This program is designed to be used with the sperically averaged output of the program SPHAVG, but it can also be used to plot the meridional scan of a quadrant averaged file, output by LKBAV.

A.2.3 HEAD

The program HEAD is for manipulating the header information of a .BIN file. The values XSCAL, YSCAL, NY, SCN, X0, Y0, XSTEP and YSTEP are typed out on the screen, and may be changed.

A.2.4 INTEG

This program sums the channels in a region defined by the user, who selects the first and last channels in each scan using the Tektronix screen crosswire facility, and who enters the first and last scans. This allows integration of the intensity in a peak.

A.2.5 LKBAV

This program averages the two halves or the four quadrants of a data set into one half or quadrant, placing the pattern centre at scan and/or channel number one. The intensity values may be added, to increase the precision, or they may be averaged, thus keeping the resulting value down. The maximum integer value allowed on the PDP 11 is 2^{15} or 32768. The pattern centre is read from the file header. Where one side of the data set extends further from the centre than the other, that side is trunkated, so that only points present in both sides or all four quadrants are included.

A.2.6 LKBA0

LKBA0 subtracts a constant value from all data points in the set. This value is either A0 from the file header, or if the user wishes, a value that is typed in. Points which would be negative after the subtraction are set to zero. The value that is subtracted is written to the header of the output file as A0.

A.2.7 LKBBG

This program is used to subtract a blank exposure from a muscle pattern. the area of the blank scanned will in general be larger that that of the muscle pattern, to ensure that the full area of the muscle data is covered after the blank is centred; if this condition is not met, an error message will print, and the program will not run. The step sizes used in scanning the muscle pattern and the blank may be different. The program will do three things, any of which may be omitted:

1. The blank data set may be re-written, using the header information from the muscle data file, to produce a third data set with the same number of scans and channels, the same centre position, and the same step sizes, as the muscle pattern.

2. Providing the data sets match, scans from the blank may be plotted on the graphics screen, superimposed on the corresponding scans from

the muscle pattern.

3. The blank may be scaled by a constant factor, and subtracted from the muscle pattern. Intensities below -5 are set to zero—small negative intensities are allowed, as otherwise where the data or blank is noisy, but the average difference is zero, the output data set will have an average positive intensity. For this operation, the blank data set must have the same size,step sizes and centre position as the muscle. The original muscle data is not over-written; a new file is produced.

A.2.8 LKBCH

This program types out the intensity of a channel from a .BIN file. The user types in the scan and channel number, and the program will repeat until a scan number of zero (or a carriage return) is entered.

A.2.9 LKBCVT

This program converts a set of .DAT files to a .BIN file and/or a .HAM file, or alternatively, this program will convert a .BIN file to a .HAM file. A background intensity level, A0 can be included in the .BIN file header, if one is written. This level is not subtracted by LKBCVT, only written to the header.

A.2.10 LKBDEL

The Program LKBDEL is used to remove parts of a data set—e.g. the sharp reflections and the backstop shadow—by fitting a first or second order polynomial to chosen regions either side of the part to be removed, and replacing the required points with intensities calculated from the polynomial coefficients. There are three modes of operation, each using a different subroutine.

1. Subroutine FIT0—a single scan is processed, up to five areas being removed at a time. The scan can be viewed, with the fit superimposed on the original data, before the input file is changed.

2. Subroutine FIT1—several scans are processed, the same regions being used in each. The user chooses the regions in one scan, and then enters the first and last scan numbers. The fit to any of the scans can be viewed before the input file is changed.

3. Subroutine FIT2—several scans are processed; the user chooses the first and last scans, and defines a different region to be removed in each, with different fit regions. The regions used in intermediate scans are calculated by linear extrapolation between the first and last scans. Again, any scan may be viewed before changes are made to the input file.

This program changes its input file, so it is important to copy that file, or data may be lost. There is an option to copy the file at the start of the program, but this is slow—it is better to copy the file using a PIP command before running the program. The copy option in the program is there as a reminder.

Regions are selected first using the crosswire facility on the graphics screen. If after viewing the fit, the user is not satisfied, the channel numbers for the regions can be typed in, or they can be re-entered using the crosswire as before.

The AMAX value in the header is updated if and only if the maximum intensity within a fitted region exceeds the old AMAX value. This means that AMAX is updated if it is increased, but not if it is decreased. The value of AMIN is not changed. These may both be updated using the program AMAX (Section A.2.1). A message is typed on the screen if the AMAX value has been changed.

The program calculates a value equivalent to the standard deviation of the data points about the fit in the region that is used for calculating the fit. The mean square residuals returned by the NAG fitting routine are used to calculate this value, which gives an indication of the error and can be used when comparing areas under the sharp reflections.

A.2.11 LKBIN

This program reads in data from the LKB densitometer, writing out a set of .DAT files.

The data is sent from the densitometer as ASCII characters. For each scan, there is a header, consisting of 21 lines of varying length, each terminated by a carriage-return line-feed pair. The header is followed by several lines of data. Each data line starts with a letter signifying the mode of output—in this case a 'B'—followed by a letter representing the number of characters on the line. Each subsequent pair of characters on the line represent a number, the last number of the line being the checksum. The coding is as follows: if a number is represented by a character of ASCII value x followed by one of ASCII value y, then the number is given by $((x-31)\times 64)+(y-30)$. The scan is terminated by a line consisting of the characters 'STOP'. The last scan is followed by a control Z character.

This program reads in the header, checking the first letter, updates the .DAT file header, then reads in the data, checking the first letter of each line, the line length, and the checksum, and writes out the scan. This is repeated until a control Z is received where the start of the header is expected. There is no handshaking except X-on and X-off, and no adequate input buffer. The baud rate is low enough and the data processing sufficiently fast that data is not lost, and X-on and X-off need not be used except between scans, where there is an appreciable delay by the densitometer in any case. Because of the lack of handshaking, the error checking performed by the program in very important.

There is a machine, designed by Alan Knight, which can be connected to the densitometer. This will intercept characters sent to the densitometer from the computer, and on receipt of the appropriate characters (not recognized by the densitometer) will press one of two buttons on the densitometer. Thus it is possible to start a run from the PDP 11, if the densitometer is in the right mode, and also to abort a run if the PDP 11 fails. This saves several journeys, as the densitometer and PDP 11 are in different buildings.

A.2.12 LKBLOG

This program converts the data values in a .BIN file to their log value, or back to the exponential. The log intensity data is scaled by 1000 to allow storage in integer form. The error introduced by the conversion to log form and back is about 0.2% in an intensity of 1000 units. The log file it used as it often allows a better fit over sharp peaks, particularly near the pattern centre, where the underlying scatter is relatively rapidly varying.

A.2.13 LKBPK

The program LKBPK uses the crosswire facility on the graphics screen to locate peaks in a .BIN file data set, either in normal scans or scans parallel to the equator. This is used to locate the pattern centre and to calibrate the pattern. This information is typed out on the screen and written to the file header as the values X0, Y0, XSCAL and YSCAL (Section A.1).

At the start of the program, after the filename is entered, the user is asked to go to the graphics screen, and to press the 'PAGE' button and then press the space bar several times, until the screen flashes. This is done because, if the screen has just been switched on, or if a key has been pressed by mistake, there may be stray characters in the buffer. In this case, when a number is sent from the screen to the computer, the odd characters are read first, so that an incorrect value is received. Subsequent values are also incorrect, as the last characters of an incorrectly read value will remain in the buffer. The buffer can be cleared by reading a dummy value consisting of space characters. The program clears the graphics screen when it has read in one crosswire position, making it flash. This complicated routine should not, of course, be necessary, but I have not found another way to clear the buffer.

Next, the user enters the type of plot—normal or rotated—and the number of scans to be averaged together for each plot. If more scans are averaged, a noisy data set will be smoothed, and so peaks will be more easily identified, but rotated scans are slow to read in, so it is best to use as few as possible. If the plot type is rotated, only the centre of the pattern will be found, as the peaks are broad in this direction in our patterns, making calibration inaccurate, but if the plot type is normal, the user chooses whether to find the centre and separation (i.e. the calibration), or only the centre.

The user can view any scans, before choosing one to measure. The peak positions are then located, using the crosswire. The positions, centre, and, if appropriate, the separations are typed out. Any number of scans may be measured—the mean values are used, and the standard deviations typed out, at the end of the program. If the user has chosen to calibrate the data set, the peak separation in nanometres will be requested, and the point separations in reciprocal nanometres are typed out. The point separation in the x-direction is found by using the ratio of the x-step and y-step values from the file header. If the centre and scale values are satisfactory, they are written to the file header. The program must be repeated to find the centre position in both the x and y directions.

A.2.14 LKBPLT

This is a plotting program for plotting scans from .BIN or .DAT files, either singly or as a 3-D plot, on the graphics screen or on the graph plotter. Data sets in .BIN files can be rotated, but not those in .DAT files, as this is very slow.

The user first states the file type and then the file name, then chooses a menu option: Single scans on the graph plotter, single scans on the graphics screen, all scans on the plotter, or all scans on the screen. If a .BIN file is used, the user must then choose between normal and rotated plots, and if rotated single scans are chosen, whether the rotation is through 90° or 45°. Also, when .BIN files are used, the number of scans to be averaged together must be entered, for all plot types except 45°—if .DAT files are used, no averaging is done.

When a single scan at 45° is chosen, the points read in are such that each has scan number and channel number one higher than the previous data point, and the centre of the pattern os included. Thus the scan is only truly 45° if the x-step and y-step are equal. If one side of the scan has less than half the number of points that the other has, this side is trunkated to the centre point; otherwise, the centre point of the pattern is made the centre of the scan, and both sides are trunkated or padded out with zeros, so that the scan contains the same number of points as a normal scan from the data set. This means that the 45° scan is plotted out on the same scale as a normal scan. If the scan is to be plotted on the graph plotter, no axes are drawn, so that the plot may be superimposed on a plot of a normal scan.

If a single scan is to be plotted, either unrotated or rotated through 90°, the program requests a scan number. The appropriate scan, or scans if averaged, is read in and plotted. Another scan number is then requested, and this repeats until a zero is received, when the program returns to the menu.

For a 3-D plot, all scans are read in, averaged into groups. These averaged scans are plotted, each offset a little upwards and to the left of the previous one, thus giving a pseudo 3-D plot. When the plot is complete, the menu is displayed again. Rotated 3-D plots are very slow; don't ask for one unless you mean it!

A.2.15 LKBREV

This program reverses the order of the data points in the x and/or y directions, i.e. reflects the data set in the x and/or y axes, or rotates the data set through 90° clockwise or anticlockwise. The rotation is very slow, but it will work on a file of any size. The program LKBROT (Section A.2.18) will rotate a file 20–30 times as quickly, but can only operate on a file of 256 points or fewer in each direction.

The questions asked by the program are as follows:

- Enter input filename:
- Enter output filename:
- Reverse x?
- Reverse y?

And, if the last two questions are both answered in the negative :

- Rotate file ?

If 'yes':

- Clockwise?

The program then re-writes the file, according to the answers received, and makes any necessary changes to the header information.

A.2.16 LKBRG

The program LKBRG plots the first scan of a .BIN file—either a spherically averaged data set or the meridian of a quadrant averaged data set—and fits a cubic spline to a part of the scan chosen by the user. The NAG library routine E02BAF is used to do the least-squares spline fit. Normal or log intensity data may be used. The gradient of the curve at the Guinier point is found from the coefficients of the spline, and this is used to calculate the gradient of the Guinier plot at this point, and thus the R_g value. The data and fit are plotted on the graphics screen at each stage, so that the user can see the fit quality, and a hard-copy Guinier plot of the data, with the tangent at the Guinier point drawn in, is produced on the graph plotter at the end of the program. Note that the program finds the gradient of a tangent to the curve; if the Guinier theory holds the plot should be linear about this point (see Chapter 3). This must be checked by eye using the output plot.

This program probably does not produce a more accurate value of R_g, but it does remove some of the systematic error inevitable when the person doing the fit expects a particular value.

A.2.17 LKBRNG

This program plots the intensity values of a ring of points about the centre of the pattern, at a radius chosen by the user. The plot starts on the meridian, above the pattern centre as the film is ositioned on the densitometer scan plate, and goes clockwise round the pattern. The angles 1°–180° and 360°–181° can be plotted on the same plot for comparison. This program is useful for detecting an asymmetric component in the scatter, which may be too small to show up in normal plots.

A.2.18 LKBROT

This program rotates a data set through 90°, clockwise or anticlockwise. Unlike the program LKBREV (Section A.2.15), which will also rotate a data set, LKBROT can only operate on a file with no more than 256 scans or channels, but it is faster by a factor of 20–40. Most of our data sets are 250 by 249 points, so this program will operate on them. The header information is updated.

A.2.19 LKBSM

Sometimes it is necessary to reduce the size of a data set; this program will do that, by averaging or adding points in the x and y directions. Thus each point in the output file is the average or sum of a square of points in the input file. The user chooses the numbers of points to average in each direction, and whether to add or average them.

A.2.20 MIRROR

This program appends the data in a .BIN file to its mirror image, thus creating a file with scans of $2 \cdot NY - 1$ channels from a file with scans of NY channels, where the scans of the output file are symmetric about the centre channel, which corresponds to the first channel of the scan in the input file. The program will also perform the inverse operation, trunkating the beginning of each scan to the centre, giving a file with scans of $(NY+1) \div 2$ channels from a file with scans of NY channels. This is used to fit over the equator of a data set, or the meridian of a rotated data set.

A.2.21 POROD

Because the R_g value is found from the gradient of a log plot, a constant intensity added to the data, e.g. film fogging, will influence the value. This program will give an estimate of the constant background level using the Porod equation (Reference), valid in the wings of the diffraction pattern, $I \propto x^{-4}$. The program fits an equation of the form $I = \alpha \cdot x^{-4} + \beta$ to the data, using a NAG library routine,and the value β is given as the background. Plots are drawn, showing the quality of the fit and the level of the background, so that the user can choose the best range of the data set. As the Porod equation is only valid in the wings of the curve, only the end of the spherically averaged scan should be used—the best value comes from a compromise between using data as far from the centre as possible, and using enough points for an accurate polynomial fit.

A.2.22 SPHAVG

This program was written by Finn Poulsen, to spherically average a data set. Each point is weighted according to the distance from the meridian, to allow for the cylindrical averaging that occurs owing to the random orientation of the myosin heads about the fibre axis. Thus a true spherical average of the data set is obtained. The input file must be a single quadrant of the pattern, with the origin at scan and channel number one. A rectangle at the pattern centre may be excluded from the average, to allow for a backstop shadow, and segments on the meridian and equator may also be excluded, by specifying an angle range. The output file is a .BIN file containing only one scan.

A version CIRAVG also exists, which performs a simple circular average on the data set. This is for use with patterns where the cylindrical averaging has not taken place, e.g. a camera or cell blank. The programs SPHAVG and CIRAVG are equivalent when the input file is circularly symmetric.

A.2.23 UNSA

The program UNSA recreates a 2-dimensional data set from the spherical average. The user chooses the centre and dimensions of the output data set (NY, SCN, X0 and Y0). Points which fall outside the range of the spherically averaged input data are set to zero. Header values in the output file are the same as those in the input file or those input by the user, as appropriate, except that AMAX and AMIN are updated. This program is used to create a circularly symmetric camera blank for subtraction from the muscle pattern, in cases where the orientation of the muscle cell was not constant (see Chapter 6).

A.2.24 WIDTH

This program allows the user to measure the peak width at half height, using the crosswire facility on the Tektronix screen.

Appendix B

Program source code listings and flow diagrams

The source code of the programs is listed here. The function of each routine is summarized briefly—for a more detailed description of the programs see Appendix A.

B.1 Main Programs

This is a list of the main programs. The subroutines called by these programs are given in Section B.2, except for a few subroutines which are used only by one program, and which are appended to that program. The routines are listed in alphabetical order of program name.

B.1.1 AMAX

The program AMAX finds the maximum and minimum intensity values in a .BIN file and updates the header information.

This routine calls the subroutines OLDNAM and BININ.

```
C       AMAX.FOR - UPDATES THE AMAX,AMIN VALUES IN A .BIN HEADER.
C
        LOGICAL*1 FNAM(15)
        INTEGER A(2000),NA,NY,SCN,AMAX,AMIN,XSTEP,YSTEP,AO,XO,YO
        1,XAMAX,YAMAX,XAMIN,YAMIN
        REAL XSCAL,YSCAL
        DATA FNAM/15*0/
        DATA NA/2000/
C
        TYPE 10                         ! ENTER FNAM
        CALL OLDNAM(FNAM,98)
        OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128)
        READ (98'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,AO,XSCAL,YSCAL,XO,YO
        TYPE *,'OLD AMAX,AMIN - ',AMAX,AMIN
        CALL BININ(1,NY,NA,A,98)
        AMAX=A(1)
        AMIN=A(1)
        DO 100 I=1,SCN
        CALL BININ(I,NY,NA,A,98)
        DO 110 J=1,NY
        IF (AMAX.GE.A(J)) GOTO 120
        AMAX=A(J)
        XAMAX=I
        YAMAX=J
120     IF (AMIN.LE.A(J)) GOTO 110
        AMIN=A(J)
        XAMIN=I
        YAMIN=J
110     CONTINUE
```

```
100     CONTINUE
        WRITE(98'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,Y0
        CLOSE(UNIT=98)
        TYPE *,'NEW AMAX,AMIN - ',AMAX,AMIN
        TYPE *,'X(AMAX),Y(AMAX) - ',XAMAX,YAMAX
        TYPE *,'X(AMIN),Y(AMIN) - ',XAMIN,YAMIN
        TYPE *,'END OF RUN'
        CALL EXIT
10      FORMAT(' ENTER FILENAME : ',$)
        END
```

B.1.2 GUIPLT

The program GUIPLT plots a .BIN file containing a single scan as a Guinier plot.

This routine calls the subroutines BININ, OLDNAM and RSIZE, and the graphics macro P12M.

```
C       GUINIER PLOT OF OUTPUT OF SPHAVG
        LOGICAL*1 FNAM(15),TEMP
        INTEGER A(500),NY,SCN,AMAX,AMIN,XSTEP,YSTEP
        1,A0,X0,Y0,XT,YT,OLDX,OLDY,X1,X2,N
        REAL XSCAL,YSCAL,X(500),Y(500),XMAX,YMAX,YMIN,DX,DY,B,C
        DATA FNAM/15*0/
        DATA OLDX,OLDY/0,0/
        TYPE 10                 ! ENTER FILENAME
        CALL OLDNAM(FNAM,98)
        OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128,READONLY)
        READ(98'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,Y0
        CALL BININ(1,NY,500,A,98)
        CLOSE(UNIT=98)
        TYPE 30                 ! ENTER FIRST,LAST CHANNEL
        ACCEPT *,X1,X2
        IF (X1.LT.2) X1=2
        IF (X2.GT.NY) X2=NY
        IF (X1.GE.X2) GOTO 1000
        IF (XSCAL.NE.0.) GOTO 200
        TYPE 40                 ! ENTER XSCAL
        ACCEPT *,XSCAL
200      DO 110 I=X1,X2
         X(I)=(FLOAT(I-1)*XSCAL)**2
         Y(I)=0
         IF (A(I).GT.0) Y(I)=ALOG10(FLOAT(A(I)))
         IF (I.EQ.X1) YMAX=Y(I)
         IF (Y(I).GT.YMAX) YMAX=Y(I)
         IF (I.EQ.X1) YMIN=Y(I)
        IF (Y(I).LT.YMIN) YMIN=Y(I)
110     CONTINUE
C
        DO 100 I=1,X1-1
        X(I)=(FLOAT(I-1)*XSCAL)**2
        Y(I)=YMIN
100     CONTINUE
C
      XMAX=X(X2)
      IF (XMAX.EQ.0..OR.(YMAX-YMIN).EQ.0.) GOTO 1100
      DX=3450/XMAX
      DY=2450/(YMAX-YMIN)
      CALL P12M(250,250,OLDX,OLDY,.FALSE.)
      CALL P12M(250,2700,OLDX,OLDY,.TRUE.)
      CALL P12M(250,250,OLDX,OLDY,.FALSE.)
      CALL P12M(3700,250,OLDX,OLDY,.TRUE.)
      CALL RSIZE(XMAX,B,N)
        DO 300 I=1,INT(B)
        XT=250+INT(FLOAT(I)*(10.**N)*DX)
        CALL P12M(XT,250,OLDX,OLDY,.FALSE.)
        CALL P12M(XT,230,OLDX,OLDY,.TRUE.)
300     CONTINUE
      TYPE *,'XDIV =',10.**N
      C=YMAX-YMIN
      CALL RSIZE(C,B,N)
        DO 310 I=1,INT(B)
        YT=250+INT(FLOAT(I)*(10.**N)*DY)
        CALL P12M(230,YT,OLDX,OLDY,.FALSE.)
        CALL P12M(250,YT,OLDX,OLDY,.TRUE.)
310     CONTINUE
      TYPE *,'Y-ORIGIN =',YMIN
      TYPE *,'YDIV =',10.**N
      XT=250+INT(X(1)*DX)
      YT=250+INT((Y(1)-YMIN)*DY)
      CALL P12M(XT,YT,OLDX,OLDY,.FALSE.)
        DO 120 I=2,X2
```

```
         XT=250+INT(X(I)*DX)
         YT=250+INT((Y(I)-YMIN)*DY)
         CALL P12M(XT,YT,OLDX,OLDY,.TRUE.)
120      CONTINUE
      GOTO 2000
C
1000  TYPE *,'X1 NOT LESS THAN X2'
      GOTO 2000
1100  TYPE *,'XMAX OR YMAX IS ZERO'
      TYPE *,'XMAX,YMAX',XMAX,YMAX
2000  TYPE *,'END OF RUN'
      CALL EXIT
10    FORMAT(' ENTER FILENAME : ',$)
20    FORMAT(14A)
30    FORMAT(' ENTER FIRST, LAST CHANNEL (INTEGERS) : ',$)
40    FORMAT(' ENTER XSCAL (REAL) : ',$)
      END
```

B.1.3 HEAD

The program HEAD allows one to change some of the header values in a .BIN file.

This routine does not call any subroutines.

```
C       CHANGES CENTRE,SCALE IN .BIN HEADER
        LOGICAL*1 FNAM(15),TEMP
        INTEGER NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,X0,Y0
        REAL XSCAL,YSCAL
        DATA FNAM/15*0/
        TYPE 10            ! ENTER FILENAME
        ACCEPT 20,(FNAM(I),I=1,14)
        OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128)
        READ(98'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,Y0
        TYPE *,'XSCAL=',XSCAL,'YSCAL=',YSCAL
        TYPE 50            ! CHANGE ?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') GOTO 100
        TYPE 30            ! ENTER NEW XSCAL,YSCAL
        ACCEPT *,XSCAL,YSCAL
100     TYPE *,'NY=',NY,'SCN=',SCN
        TYPE 50            ! CHANGE?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') GOTO 105
        TYPE 35            ! ENTER NEW NY,SCN
        ACCEPT *,NY,SCN
105     TYPE *,'X0=',X0,'Y0=',Y0
        TYPE 50            ! CHANGE ?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') GOTO 110
        TYPE 40            ! ENTER NEW X0,Y0
        ACCEPT *,X0,Y0
110     TYPE *,'XSTEP = ',XSTEP,'  YSTEP = ',YSTEP
        TYPE 50            ! CHANGE ?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') GOTO 120
        TYPE 60           ! ENTER NEW XSTEP,YSTEP
        ACCEPT *,XSTEP,YSTEP
120     WRITE (98'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,Y0
        CLOSE(UNIT=98)
        CALL EXIT
10      FORMAT(' ENTER FILENAME : ',$)
20      FORMAT(A:13A)
30      FORMAT(' ENTER NEW XSCAL,YSCAL (REAL) : ',$)
35      FORMAT(' ENTER NEW NY,SCN (INTEGER) : ',$)
40      FORMAT(' ENTER NEW X0,Y0 (INTEGER) : ',$)
50      FORMAT(' CHANGE THESE (CR=Y) ? ',$)
60      FORMAT(' ENTER NEW XSTEP,YSTEP (INTEGER) : ',$)
        END
```

B.1.4 LKBAV

The program LKBAV averages the four quadrants of a data set into one or two quadrants. The intensity values may be added or averaged—addition reduces noise levels, but the maximum intensity may then be more than the maximum integer value possible on the PDP 11.

This routine calls the subroutines CLRSCR, OLDNAM and NEWNAM.

```
C       LKBAV
C       AVERAGES QUADRANTS TOGETHER
        LOGICAL*1 FNAM(15),FNAM1(15),TEMP,AV(2),ADD
        INTEGER A(2000),B(2000),NY,SCN,AMAX,AMIN,XSTEP,YSTEP
        1,AO,XO,YO,XO1,YO1,XN,YN,REC,NS
        REAL XSCAL,YSCAL
        DATA AV/2*.TRUE./
        DATA ADD/.FALSE./
        DATA FNAM,FNAM1/15*0,15*0/
        CALL CLRSCR
C       ***********************************************************
        TYPE 10,'Y','Y'                    ! AVERAGE +Y AND -Y ?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') AV(2)=.FALSE.
        TYPE 10,'X','X'                    ! AVERAGE +X AND -X ?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') AV(1)=.FALSE.
        IF (AV(1).EQ..FALSE..AND.AV(2).EQ..FALSE.) GOTO 2000
        TYPE 30,' INPUT'                   ! ENTER INPUT FILENAME
        CALL OLDNAM(FNAM,98)
        TYPE 30,'OUTPUT'                   ! ENTER OUTPUT FILENAME
        CALL NEWNAM(FNAM1,98)
C       ***********************************************************
        OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128,READONLY,ERR=1100)
        OPEN(UNIT=99,NAME=FNAM1,TYPE='NEW',ACCESS='DIRECT'
        1,RECORDSIZE=128,ERR=1100)
        READ(98'1,ERR=1200) NY,SCN,AMAX,AMIN,XSTEP,YSTEP
        1,AO,XSCAL,YSCAL,XO,YO
        IF (XO.EQ.O.AND.YO.EQ.O) GOTG 1300
        TYPE *,'AMAX = ',AMAX
        TYPE 40                     !AVERAGE OR ADD DATA?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') ADD=.TRUE.
C
C       CALCULATING HEADER PARAMETERS
        IF (ADD.EQ..TRUE..AND.AV(1).EQ..TRUE.) AMAX=AMAX*2
        IF (ADD.EQ..TRUE..AND.AV(2).EQ..TRUE.) AMAX=AMAX*2
        XN=XO-1
        IF (XN.GT.SCN-XO) XN=SCN-XO
        IF (AV(1).EQ..FALSE.) XN=SCN
        YN=YO
        IF (YN.GT.NY-YO) YN=NY-YO
        IF (AV(2).EQ..FALSE.) YN=NY
        YO1=YO
        XO1=XO
        IF (AV(2).EQ..TRUE.) YO1=1
        IF (AV(1).EQ..TRUE.) XO1=1
        WRITE (99'1,ERR=1200) YN,XN,AMAX,AMIN,XSTEP,YSTEP
        1,AO,XSCAL,YSCAL,XO1,YO1
        IF (AV(2).EQ..FALSE.) YO=0
        IF (AV(1).EQ..FALSE.) XO=0
C       **********************************************************
          DO 300 NS=1,XN
          IF (AV(1).EQ..TRUE.) GOTO 340
C
C       READING IN SCAN - NO AVERAGING IN X
          CALL BININ(NS,NY,2000,B,98)
          GOTO 380
C
C       READING IN SCANS - AVERAGING IN X
340       CALL BININ(XO-NS,NY,2000,A,98)
          CALL BININ(XO+NS-1,NY,2000,B,98)
            DO 360 I=1,NY
            A(I)=A(I)+B(I)
360         CONTINUE
C
          IF (AV(2).EQ..TRUE.) GOTO 380
C       CALCULATING OUTPUT SCAN - NO AVERAGING IN Y
            DO 390 I=1,YN
            B(I)=A(I)
390         CONTINUE
          GOTO 210
C
C       CALCULATING OUTPUT SCAN - AVERAGING IN Y
380       B(1)=A(Y0)*2
            DO 200 I=2,YN
            B(I)=A(Y0+I-1)+A(Y0-I+1)
200         CONTINUE
C
210     IF (ADD.EQ..TRUE.) GOTO 230
C
C       DIVIDING OUTPUT SCAN FOR AVERAGE
          A(1)=1
          IF (AV(1).EQ..TRUE.) A(1)=A(1)*2
          IF (AV(2).EQ..TRUE.) A(1)=A(1)*2
            DO 220 I=1,YN
            B(I)=INT(0.5+(FLOAT(B(I))/FLOAT(A(1))))
220         CONTINUE
C
C       WRITING OUTPUT SCAN
230       CALL BINOUT(NS,YN,2000,B,99)
300       CONTINUE
C
        CLOSE(UNIT=98,ERR=1100)
        CLOSE(UNIT=99,ERR=1100)
        TYPE *,'AVERAGING COMPLETE'
        GOTO 2000
1100    TYPE *,'ERROR OPENING/CLOSING FILE'
        GOTO 2000
```

```
1200    TYPE *,'ERROR READING/WRITING FILE'
        GOTO 1900
1300    TYPE *,'XO AND YO BOTH ZERO'
1900    CLOSE(UNIT=98,ERR=1100)
        CLOSE(UNIT=99,ERR=1100)
2000    TYPE *,'END OF RUN'
        CALL EXIT
10      FORMAT(' AVERAGE +',A,' AND -',A,' (CR=Y) ? ',$)
20      FORMAT(A:13A)
30      FORMAT(' ENTER ',A6,' FILENAME : ',$)
40      FORMAT(' AVERAGE DATA (CR=Y, N=ADD DATA) ? ',$)
        END
```

B.1.5 LKBAO

The program LKBAO subtracts a constant from the intensity values in a .BIN file.

This routine calls the subroutines BININ, BINOUT, NEWNAM and OLDNAM.

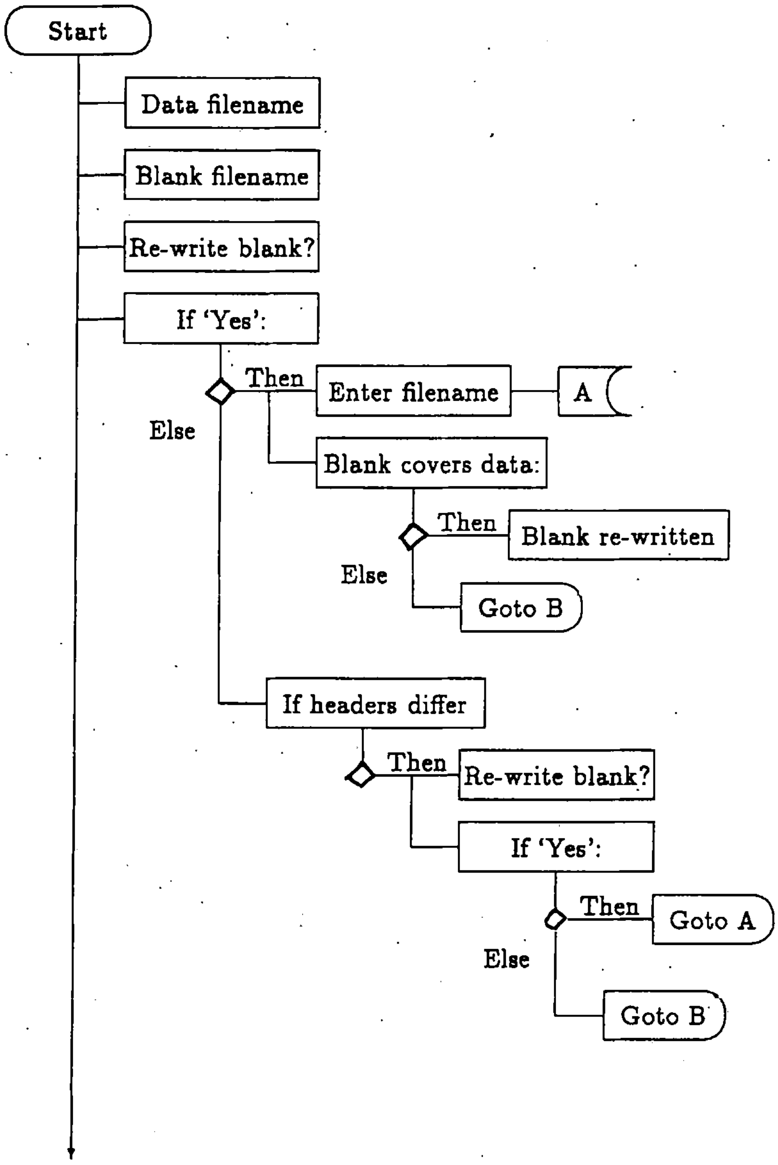
```
C       PROGRAM LKBAO
C       SUBTRACTS CONSTANT BACKGROUND INTENSITY
        INTEGER A(2000),NY,SCN,AMAX,AMIN,XSTEP,YSTEP,AO,XO,YO,NS
        1,NR,REC
        REAL XSCAL,YSCAL
        LOGICAL*1 FNAM(15),FNAM1(15),TEMP
        DATA FNAM,/15*0,15*0/
        TYPE 10,' IN'            ! ENTER INPUT FILENAME
        CALL OLDNAM(FNAM,98)
        TYPE 10,'OUT'            ! ENTER OUTPUT FILENAME
        CALL NEWNAM(FNAM1,98)
        OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128,READONLY)
        OPEN(UNIT=99,NAME=FNAM1,TYPE='NEW',ACCESS='DIRECT'
        1,RECORDSIZE=128)
        READ (98'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,AO
        1,XSCAL,YSCAL,XO,YO
        NR=INT(FLOAT(NY-1)/256)+1
        TYPE 30,AO                       ! AO O.K. ?
        ACCEPT 40,TEMP
        IF (TEMP.NE.'N') GOTO 100
        TYPE 50                  ! ENTER NEW LEVEL
        ACCEPT *,AO
100     AMAX=AMAX-AO
        AMIN=AMIN-AO
        IF (AMIN.LT.0) AMIN=0
        WRITE(99'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,AO
        1,XSCAL,YSCAL,XO,YO
        TYPE *,'SUBTRACTING BACKGROUND . . . .'
```

```
          DO 200 NS=1,SCN
          CALL BININ(NS,NY,2000,A,98)
            DO 210 I=1,NY
            A(I)=A(I)-A0
            IF (A(I).LT.0) A(I)=0
 210        CONTINUE
          CALL BINOUT(NS,NY,2000,A,99)
 200      CONTINUE
        CLOSE(UNIT=98)
        CLOSE(UNIT=99)
        TYPE *,'END OF RUN'
        CALL EXIT
10      FORMAT(' ENTER ',A3,'PUT FILE NAME : ',$)
20      FORMAT(14A)
30      FORMAT(' BACKGROUND LEVEL IS ',I5,/,' IS THIS OK (CR=Y) ? ',$)
40      FORMAT(A)
50      FORMAT(' ENTER NEW A0 : ',$)
        END
```

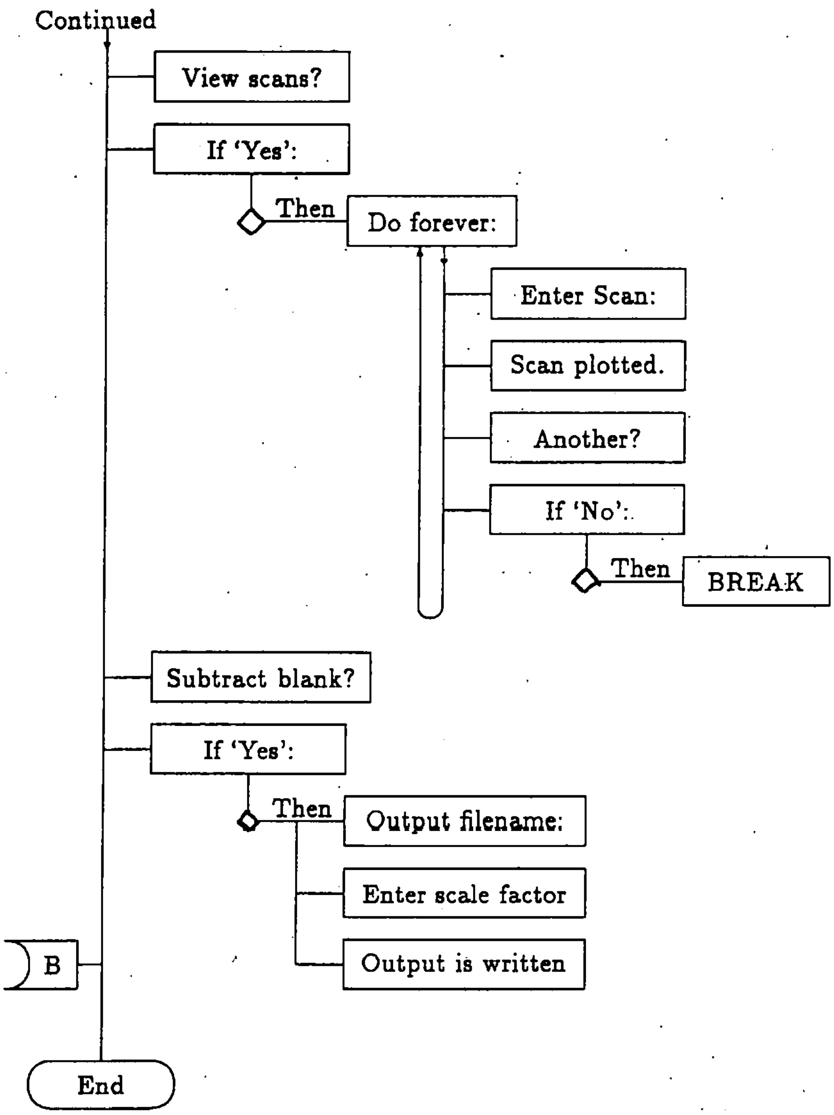
B.1.6 LKBBG

The program LKBBG subtracts a scaled data set (the camera blank) from another data set (the muscle data).

This routine calls the subroutines BININ, BINOUT, BITPLT, NEWNAM, OLDNAM and TKPLT.



P.T.O.


Continued
View scans?
If 'Yes':
Then
Do forever:
Enter Scan:
Scan plotted.
Another?
If 'No':
Then
BREAK
Subtract blank?
If 'Yes':
Then
Output filename:
Enter scale factor
Output is written
B
End

```
C       RE-WRITES A FILE TO HAVE THE SAME NY,SCN,X0,Y0,XSTEP,YSTEP AS
C       A CHOSEN DATA FILE.  SCANS FROM BOTH FILES CAN BE VIEWED
C       SUPERIMPOSED ON THE TEKTRONIX SCREEN.  THIS FILE MAY THEN BE
C       MULTIPLIED BY A CONSTANT FACTOR AND SUBTRACTED FROM
C       THE DATA FILE.
C
C       UNIT 97 IS USED FOR THE DATA FILE - INPUT ONLY.
C       UNIT 98 IS FIRST USED FOR THE ORIGINAL BLANK FILE AND THEN FOR
C       THE RE-FORMATTED BLANK FILE - INPUT ONLY.
C       UNIT 99 IS USED FOR OUTPUT FILES - FIRST THE REFORMATTED BLANK
C       FILE AND THEN THE SUBTRACTED FILE.
C
        LOGICAL*1 FNAM(15),FNAM1(15),FNAM2(15),TEMP
        INTEGER A(1000),NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,X0,Y0
        INTEGER B1(1000),B2(1000),NY1,SCN1,AMAX1,AMIN1,ATMAX
        1,XSTEP1,YSTEP1,X01,Y01,NS
        REAL XSCAL,YSCAL,XT,YT,DX,DY,XDIV,YDIV,F
        DATA FNAM,FNAM1,FNAM2/15*0,15*0,15*0/
        TYPE 10,'  DATA'                ! ENTER DATA FILE NAME
        CALL OLDNAM(FNAM,98)
        TYPE 10,' BLANK'                ! ENTER BLANK FILE NAME
        CALL OLDNAM(FNAM1,98)
        TYPE 80                         ! RE-WRITE BLANK?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') GOTO 1100
        TYPE 10,'OUTPUT'                ! ENTER OUTPUT FILE NAME
        CALL NEWNAM(FNAM2,98)
        OPEN(UNIT=97,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128,READONLY)
        OPEN(UNIT=98,NAME=FNAM1,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128,READONLY)
        READ (98'1) NY1,SCN1,AMAX1,AMIN1,XSTEP1,YSTEP1,A0
        1,XSCAL,YSCAL,X01,Y01
        READ (97'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,Y0
120     IF (Y01*YSTEP1.LT.Y0*YSTEP) GOTO 1000
```

```
        IF (X01*XSTEP1.LT.X0*XSTEP) GOTO 1000
        IF ((NY1-Y01)*YSTEP1.LT.(NY-Y0)*YSTEP) GOTO 1000
        IF ((SCN1-X01)*XSTEP1.LT.(SCN-X0)*XSTEP) GOTO 1000
        TYPE *,'THIS WILL TAKE ME A LITTLE WHILE . . . .'
        OPEN(UNIT=99,NAME=FNAM2,TYPE='NEW',ACCESS='DIRECT'
        1,RECORDSIZE=128)
        WRITE (99'1) NY,SCN,AMAX1,AMIN1,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,Y0
          DO 100 I=1,SCN
          XT=FLOAT((I-X0)*XSTEP)/FLOAT(XSTEP1)+FLOAT(X01)
          DX=XT-FLOAT(INT(XT))
          CALL BININ(INT(XT),NY1,1000,B1,98)
          CALL BININ(INT(XT+1),NY1,1000,B2,98)
            DO 110 J=1,NY
            YT=FLOAT((J-Y0)*YSTEP)/FLOAT(YSTEP1)+FLOAT(Y01)
            DY=YT-FLOAT(INT(YT))
            A(J)=B1(INT(YT))*(1-DX)*(1-DY)+B1(INT(YT)+1)*(1-DX)*DY
            A(J)=A(J)+B2(INT(YT))*DX*(1-DY)+B2(INT(YT)+1)*DX*DY
110         CONTINUE
          CALL BINOUT(I,NY,1000,A,99)
100       CONTINUE
        CLOSE(UNIT=98)          ! ORIG. BLANK FILE
        CLOSE(UNIT=99)          ! NEW BLANK FILE
        OPEN(UNIT=98,NAME=FNAM2,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128,READONLY)
C
200     TYPE 30                 ! VIEW SCANS ?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') GOTO 300
        ATMAX=AMAX1
        IF (ATMAX.LT.AMAX) ATMAX=AMAX
210     TYPE 60,SCN             ! ENTER SCAN NUMBER
        CALL GINT(NS)
        CALL BININ(NS,NY,1000,B1,97)
        CALL BININ(NS,NY,1000,B2,98)
        CALL TKPLT(NY,1000,B1,AMAX,AMIN,XDIV,YDIV)
        CALL BITPLT(NY,1000,B2,ATMAX,AMIN,1,NY)
```

```
      TYPE 50                     ! VIEW ANOTHER ?
      ACCEPT 20,TEMP
      IF (TEMP.NE.'N') GOTO 210
C
300   TYPE 40                     ! SUBTRACT BLANK FROM DATA ?
      ACCEPT 20,TEMP
      IF (TEMP.EQ.'N') GOTO 400
      TYPE 10,'OUTPUT'            ! ENTER FILE NAME
      CALL NEWNAM(FNAM1,99)
      OPEN(UNIT=99,NAME=FNAM1,TYPE='NEW',ACCESS='DIRECT'
      1,RECORDSIZE=128)
      TYPE *,'BLANK AMAX,AMIN = ',AMAX1,AMIN1
      TYPE *,'DATA AMAX,AMIN  = ',AMAX,AMIN
      TYPE 70            ! ENTER SCALE FACTOR
      CALL GREAL(F)
      TYPE *,'HANG ON A SEC. . . . .'
      WRITE(99'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,Y0
        DO 310 NS=1,SCN
        CALL BININ(NS,NY,1000,B1,97)
        CALL BININ(NS,NY,1000,B2,98)
        IF (NS.EQ.1) AMAX=B1(1)-INT(F*FLOAT(B2(1)))
        IF (NS.EQ.1) AMIN=AMAX
          DO 320 I=1,NY
          A(I)=B1(I)-INT((F*FLOAT(B2(I)))+0.5)
          IF (A(I).LT.-5) A(I)=0
          IF (AMAX.LT.A(I)) AMAX=A(I)
          IF (AMIN.GT.A(I)) AMIN=A(I)
320       CONTINUE
        CALL BINOUT(NS,NY,1000,A,99)
310     CONTINUE
      WRITE(99'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,Y0
      CLOSE(UNIT=99)
C
400   CLOSE(UNIT=97)
      CLOSE(UNIT=98)
      GOTO 2000
```

```
1000    TYPE *,'BLANK DOES NOT COVER DATA AREA'
        GOTO 2000
1100    OPEN(UNIT=97,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128,READONLY)
        OPEN(UNIT=98,NAME=FNAM1,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128,READONLY)
        READ (98'1) NY1,SCN1,AMAX1,AMIN1,XSTEP1,YSTEP1,A0
        1,XSCAL,YSCAL,X01,Y01
        READ (97'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,Y0
        IF (NY1.NE.NY.OR.SCN1.NE.SCN) GOTO 1200
        IF (XSTEP1.NE.XSTEP.OR.YSTEP1.NE.YSTEP) GOTO 1200
        GOTO 200
1200    TYPE *,'FILE HEADERS NOT COMPATABLE - '
        TYPE 80                     ! RE-WRITE BLANK ?
        ACCEPT 20,TEMP
        IF (TEMP.NE.'N') GOTO 120
2000    TYPE *,'END OF RUN'
        CALL EXIT
10      FORMAT(' ENTER ',A6,' FILE NAME (14A) : ',$)
20      FORMAT(A:13A)
30      FORMAT(' VIEW SCANS (CR=Y) ? ',$)
40      FORMAT(' SUBTRACT BLANK (CR=Y) ? ',$)
50      FORMAT(' VIEW ANOTHER (CR=Y) ? ',$)
60      FORMAT(' ENTER SCAN NUMBER (MAX=',I5,') : ',$)
70      FORMAT(' ENTER FACTOR FOR SCALING BACKGROUND (REAL) : ',$)
80      FORMAT(' RE-WRITE BLANK WITH NEW STEP,CENTRE (CR=Y) ? ',$)
        END
```

B.1.7 LKBCH

The program LKBCH types out the intensity value of a data point from a .BIN file.

This routine calls the subroutines BININ, GINT and OLDNAM.

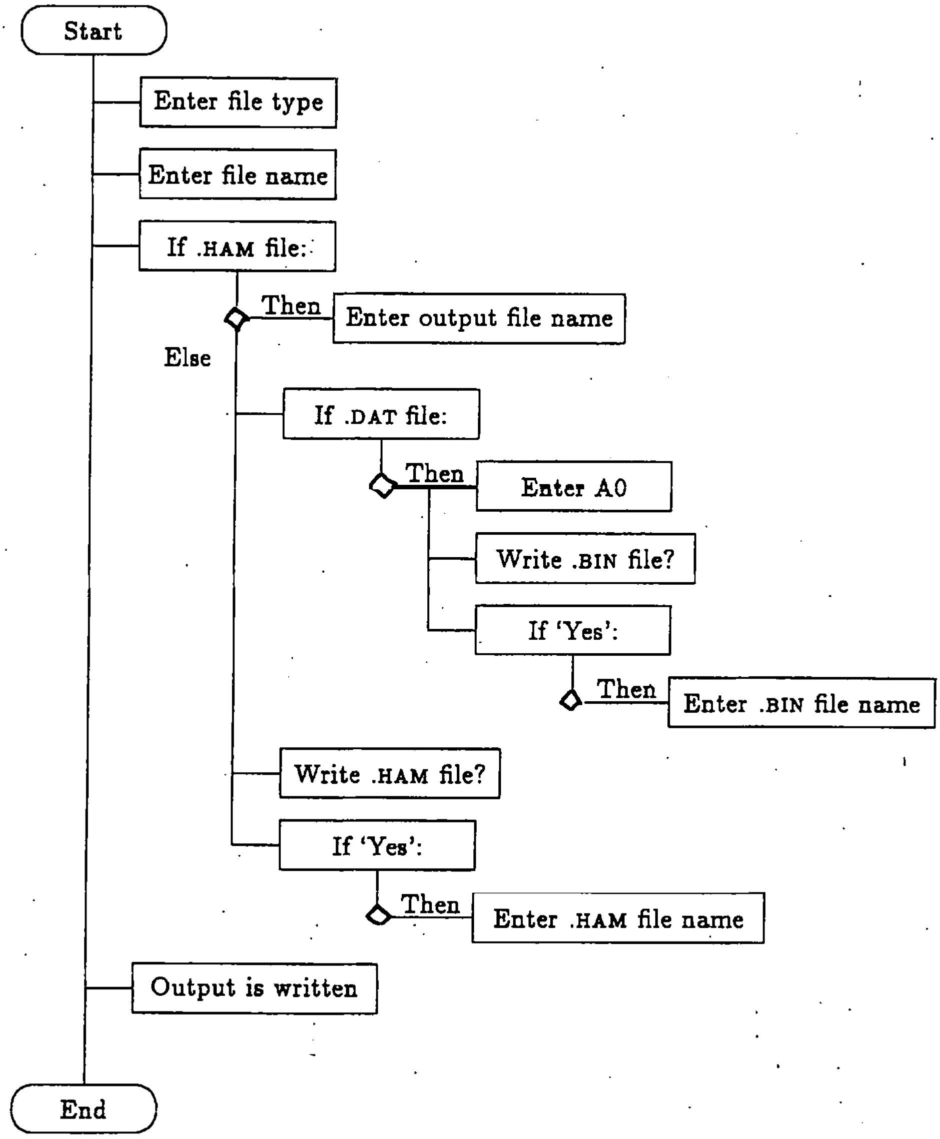
```
C       TYPES OUT THE INTENSITY OF A CHOSEN CHANNEL
        LOGICAL*1 FNAM(15)
        INTEGER NY,SCN,A(2000),NA,NS,N1,N2
        DATA NA/2000/
        DATA FNAM/15*0/
C
        TYPE 10                         ! ENTER FILENAME
        CALL OLDNAM(FNAM,98)
        OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128)
        READ(98'1) NY,SCN
        TYPE *,'NY,SCN = ',NY,SCN
C
100     TYPE 20                         ! ENTER SCAN NUMBER
        CALL GINT(N1)
        IF (N1.EQ.0) GOTO 2000
        IF (N1.GT.SCN) GOTO 1000
        IF (N1.NE.NS) CALL BININ(N1,NY,NA,A,98)
        NS=N1
110     TYPE 30                         ! ENTER CHANNEL
        CALL GINT(N2)
        IF (N2.EQ.0.OR.N2.GT.NY) GOTO 1100
        TYPE *,A(N2)
        GOTO 100
C
1000    TYPE *,'NUMBER OUT OF RANGE - RE-ENTER'
        GOTO 100
1100    TYPE *,'NUMBER OUT OF RANGE - RE-ENTER'
        GOTO 110
C
```

```
2000     TYPE *,'END OF RUN'
         CLOSE(UNIT=98)
         CALL EXIT
C
10       FORMAT(' ENTER FILE NAME : ',$)
20       FORMAT(' ENTER SCAN NUMBER (0 FOR END OF RUN) : ',$)
30       FORMAT(' ENTER CHANNEL NUMBER : ',$)
         END
```

B.1.8 LKBCVT

The program LKBCVT converts a set of .DAT files to a .BIN file and/or a pair of .HAM files, or a .BIN file to a pair of .HAM files, or a pair of .HAM files to a .BIN file.

This routine calls the subroutines BININ, BINOUT, GINT, HEADER, NEWNAM and OLDNAM, as well as the subroutines DATIN and HAMOUT included in this file.



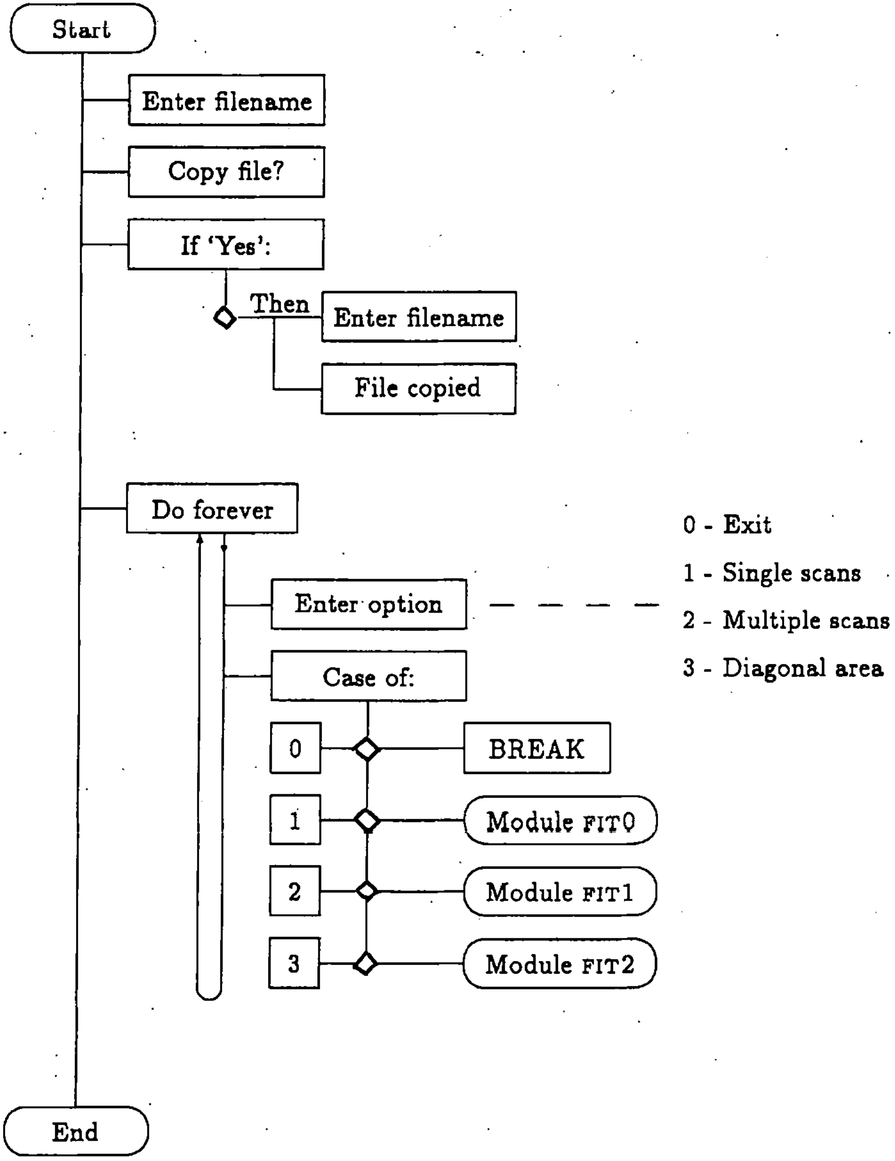
```
C       LKBCVT
C       CONVERTS .DAT FILES TO A .BIN FILE AND/OR .HAM FILES,
C       A .BIN FILE TO .HAM FILES, OR A .HAM FILE TO A .BIN FILE.
C
        INTEGER NY,A(2000),SCN,AMAX,AMIN,XSTEP,YSTEP,REC,NR,A0
        LOGICAL*1 FNAM(15),FNAM1(15),FNAM2(15),INFIL,OUTFIL(2),TEMP
        DATA FNAM/14*'0',0/
        DATA FNAM1,FNAM2/15*0,15*0/
        DATA INFIL,OUTFIL/1,1,1/
C
        TYPE 40                           ! ENTER INPUT FILETYPE
        ACCEPT 10,TEMP
        IF (TEMP.EQ.'B') INFIL=2
        IF (TEMP.EQ.'H') INFIL=3
        IF (INFIL.EQ.2) GOTO 100
        IF (INFIL.EQ.3) GOTO 300
C
        CALL HEADER(FNAM,AMAX,AMIN,SCN,XSTEP,YSTEP)
        CALL DATIN(FNAM,1,NY,A)
        TYPE 30                           ! ENTER BACKGROUND LEVEL
        CALL GINT(A0)
        TYPE 60,'.BIN'                    ! WRITE .BIN FILE ?
        ACCEPT 10,TEMP
        IF (TEMP.EQ.'N') OUTFIL(1)=0
        IF (TEMP.EQ.'N') GOTO 110
        TYPE 20,'OUTPUT','.BIN'           ! ENTER OUTPUT .BIN FILENAME
        CALL NEWNAM(FNAM1,98)
        OPEN(UNIT=98,NAME=FNAM1,TYPE='NEW',ACCESS='DIRECT'
        1,RECORDSIZE=128)
        GOTO 110
C
100     OUTFIL(1)=0
        TYPE 20,'INPUT ','.BIN'           ! ENTER INPUT FILENAME
        CALL OLDNAM(FNAM1,98)
        OPEN(UNIT=98,NAME=FNAM1,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128)
        READ (98'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0
C
110     TYPE 60,'.HAM'                    ! WRITE .HAM FILE ?
        ACCEPT 10,TEMP
        IF (TEMP.EQ.'N') OUTFIL(2)=0
        IF (OUTFIL(1).EQ.0.AND.OUTFIL(2).EQ.0) GOTO 2000
        IF (TEMP.EQ.'N') GOTO 120
        TYPE 20,'OUTPUT','.HAM'           ! ENTER OUTPUT .HAM FILENAME
        CALL NEWNAM(FNAM2,99)
        FNAM2(10)='0'
        OPEN(UNIT=99,NAME=FNAM2,TYPE='NEW')
        FNAM2(10)='1'
        WRITE(99,50) NY,SCN,0,0,0,0,0,0,0,0,(FNAM2(I),I=5,10)
        CLOSE(UNIT=99)
        GOTO 120
C
300     TYPE 20,'HEADER','.HAM'           ! ENTER INPUT HEADER FILE
        CALL OLDNAM(FNAM,97)
        OPEN(UNIT=97,NAME=FNAM,TYPE='OLD')
        READ (97,80) NY,SCN
        CLOSE(UNIT=97)
        TYPE 20,' DATA ','.HAM'           ! ENTER INPUT DATA FILE
        CALL OLDNAM(FNAM,97)
        OPEN(UNIT=97,NAME=FNAM,TYPE='OLD')
        OUTFIL(2)=0
        TYPE 20,'OUTPUT','.BIN'           ! ENTER OUTPUT FILENAME
        CALL NEWNAM(FNAM1,98)
        OPEN(UNIT=98,NAME=FNAM1,TYPE='NEW',ACCESS='DIRECT'
        1,RECORDSIZE=128)
        AMAX=0
        AMIN=0
        XSTEP=1
        YXTEP=1
        A0=0
C
```

```
120       DO 200 K=1,SCN
          NS=K
          TYPE *,NS
          IF (INFIL.EQ.1) CALL DATIN(FNAM,NS,NY,A)
          IF (INFIL.EQ.2) CALL BININ(NS,NY,2000,A,98)
          IF (INFIL.EQ.3) READ(97,90) (A(I),I=1,NY)
          IF (NS.NE.1) GOTO 210
          IF (OUTFIL(1).EQ.1) WRITE(98'1) NY,SCN,AMAX,AMIN
        1,XSTEP,YSTEP,AO
          IF (OUTFIL(2).EQ.1)         OPEN(UNIT=99,NAME=FNAM2,TYPE='NEW')
          IF (OUTFIL(2).EQ.1) WRITE(99,70)
          TYPE *,'AMAX=',AMAX
          TYPE *,'AMIN=',AMIN
          TYPE *,'SCN=',SCN
          TYPE *,'NY=',NY
210       IF (OUTFIL(1).EQ.1) CALL BINOUT(NS,NY,2000,A,98)
          IF (OUTFIL(2).EQ.1) CALL HAMOUT(NS,NY,A)
 200      CONTINUE
        IF (INFIL.EQ.2.OR.OUTFIL(1).EQ.1) CLOSE(UNIT=98)
        IF (OUTFIL(2).EQ.1) CLOSE(UNIT=99)
2000    TYPE *,'END OF RUN'
        CALL EXIT
10      FORMAT(A:13A)
20      FORMAT(' ENTER 14 CHARACTER ',A6,' FILENAME
        1 (',A4,' FILE) : ',$)
30      FORMAT(' ENTER BACKGROUND LEVEL :',$)
40      FORMAT(' ENTER FILETYPE FOR READ - '
        1,' D=.DAT,B=.BIN,H=.HAM (CR=D) : ',$)
50      FORMAT('+ ',/,X,' ',/,X,10I8,/,X,6A,'.RAW;1',X,/)
60      FORMAT(' WRITE OUT A ',A4,' FILE (CR=Y) ? ',$)
70      FORMAT('+ ')
80      FORMAT(/,/,2I8)
90      FORMAT(16I5)
        END
C
        SUBROUTINE DATIN(FNAM,NS,NY,A)
        LOGICAL*1 FNAM(15)
        INTEGER A(2000),NS,NY,A1
        CALL FILNAM(FNAM,NS)
        OPEN(UNIT=97,NAME=FNAM,TYPE='OLD')
        READ(97,10) NY,A1,(A(I),I=1,NY-1)
        CLOSE(UNIT=97)
        NY=NY-1
        RETURN
10      FORMAT(I5)
        END
C
        SUBROUTINE HAMOUT(NS,NY,A)
        INTEGER NS,NY,A(2000)
        WRITE(99,10) (A(I),I=1,NY)
        RETURN
10      FORMAT(X,16I5)
        END
```

B.1.9 LKBDEL

The program LKBDEL removes peaks in the data by fitting over them. Scans may be processed individually or in groups. A root mean square residual for the fit is calculated from the mean square residuals returned by the NAG routines used for the fitting, and this value is typed out on the screen as a guide to the uncertanty in the fit.

This routine calls the subroutines CLRSCR, FIT0, FIT1, FIT2, NEWNAM and OLDNAM and the graphics macros RDCW, TEKOUT and TEKWIP. The routines FIT, FTPLT and PLT which are plotting routines used solely by this program, are included in this file.



```
C       DELM.FOR
C       MAIN PROGRAM CORE FOR LKBDEL
C       REMOVES PEAKS FROM A .BIN FILE
C
        INTEGER A(512),AMAX,AMIN,AO,XSTEP,YSTEP
        1,REC,SCN,NY,X,Y,XO,YO,NA
```

```
      REAL XSCAL,YSCAL
      LOGICAL*1 FNAM(15),FNAM1(15),TEMP
      DATA NA/512/
      DATA FNAM,FNAM1/15*0,15*0/
      CALL CLRSCR
      TYPE 10,' INPUT'                    ! ENTER FILENAME
      CALL OLDNAM(FNAM1,98)
      TYPE *,'THIS PROGRAM WILL DELETE PARTS OF THIS FILE - '
      TYPE 30           ! COPY?
      ACCEPT 20,TEMP
C
        DO 100 I=1,15
        FNAM(I)=FNAM1(I)
100     CONTINUE
C
      IF (TEMP.EQ.'N') GOTO 130
      TYPE 10,'OUTPUT'                    ! ENTER FILENAME
      CALL NEWNAM(FNAM,98)
      TYPE *,'COPYING TO NEW FILE . . . . .'
      OPEN (UNIT=99,NAME=FNAM1,TYPE='OLD',ACCESS='DIRECT'
      1,RECORDSIZE=128,ERR=1000,READONLY)
      READ(99'1,ERR=1100) NY,SCN,AMAX,AMIN
      1,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,Y0
      REC=(SCN*INT(FLOAT(NY-1)/256)+1)+1
      OPEN (UNIT=98,NAME=FNAM,TYPE='NEW',ACCESS='DIRECT'
      1,RECORDSIZE=128,ERR=1000)
      WRITE (98'1,ERR=1100) NY,SCN,AMAX,AMIN
      1,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,Y0
C
        DO 120 I=2,REC
        READ(99'I,ERR=1100) (A(J),J=1,256)
        WRITE(98'I,ERR=1100) (A(J),J=1,256)
120     CONTINUE
C
      CLOSE (UNIT=99)
      CLOSE (UNIT=98)
```

```
        GOTO 140
C
130     OPEN (UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128)
        READ(98'1,ERR=1100) NY,SCN,AMAX,AMIN
        1,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,Y0
        CLOSE(UNIT=98)
C
140     IF (NY.GT.NA) GOTO 1400
        TYPE *,'AMAX -',AMAX,'       AMIN -',AMIN
        TYPE 40                 !CR TO CONTINUE
        ACCEPT 20
        CALL CLRSCR
        TYPE *,'PRESS "PAGE" ON TEKTRONIX, THEN SPACE BAR SEVERAL'
        TYPE *,'TIMES UNTIL SCREEN FLASHES'
        CALL RDCW(X,Y)
        CALL TEKWIP
C
150     CALL CLRSCR
        TYPE 50                 ! MENU
        ACCEPT 20,TEMP
        IF (TEMP.EQ.' ') GOTO 2000
        IF (TEMP.LT.49.OR.TEMP.GT.51) GOTO 1300
        CALL CLRSCR
        IF (TEMP.EQ.'1') CALL FIT0(FNAM,NY,NA,A,SCN,AMAX,AMIN)
        IF (TEMP.EQ.'2') CALL FIT1(FNAM,NY,NA,A,SCN,AMAX,AMIN)
        IF (TEMP.EQ.'3') CALL FIT2(FNAM,NY,NA,A,SCN,AMAX,AMIN)
        GOTO 150
C       ********************************************************
1000    TYPE *,'ERROR OPENING/CLOSING FILE - EXITING ROUTINE'
        GOTO 2000
1100    TYPE *,'ERROR READING/WRITING FILE - EXITING ROUTINE'
        CLOSE(UNIT=98)
        GOTO 2000
1300    TYPE *,'INVALID OPTION - PLEASE TRY AGAIN'
        GOTO 150
```

```
1400    TYPE *,'ARRAY OVERFLOW '
        GOTO 2000
C       *******************************************************
2000    TYPE *,'END OF RUN'
        CALL TEKOUT(31)
        CALL EXIT
C       *******************************************************
10      FORMAT(' ENTER ',A6,' FILE NAME : ',$)
20      FORMAT(A:13A)
30      FORMAT(' DO YOU WANT TO COPY IT (CR=Y) ?',$)
40      FORMAT(' ENTER <CR> TO CONTINUE',$)
50      FORMAT(/,/,X,'  1. FIT OVER PEAKS IN SINGLE SCANS'
        1  ,/,X,'       2. FIT OVER PEAKS IN MULTIPLE SCANS'
        2  ,/,X,'       3. FIT OVER DIAGONAL AREA'
        3  ,/,X,'       <CR> FOR END OF RUN'
        4  ,/,/,/,X,' ENTER OPTION : ',$)
        END
C       DELSUB.FOR
C       SUBROUTINES FOR DELETIONS
C
C       PLT - READS IN AND PLOTS A SCAN
C       FTPLT - READS IN AND PLOTS A SCAN, FITTING OVER AREAS
C       FIT - FITS OVER AN AREA IN A SCAN USING E02ADF
C
C       FTPLT CALLS FIT
CΩ      SUBROUTINE PLT(FNAM,NS,NY,SCN,NA,A,AMAX,AMIN,NAV)
        INTEGER REC,NS,NY,SCN,NA,A(NA),AMAX,AMIN,NAV,XDIV,YDIV
        LOGICAL*1 FNAM(15)
        OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128,ERR=1000)
        CALL AVIN(NS,NY,SCN,NA,A,NAV,'N',98)
        CLOSE(UNIT=98)
        CALL TKPLT(NY,NA,A,AMAX,AMIN,XDIV,YDIV)
        RETURN
1000    TYPE *,'ERROR OPENING FILE IN PLT - EXITING ROUTINE'
        GOTO 2000
```

```
2000    CALL EXIT
        END
C       ********************************************************
        SUBROUTINE FTPLT(FNAM,NS,NY,NA,A,AMAX,AMIN,AMAX1,NAV,N
        1,XT1,XT2,FT1,FT2,KPLUS1,IFAIL)
        INTEGER REC,NS,NY,NA,A(NA),AMAX,AMIN,NAV,XDIV,YDIV
        1,N,XT1(4),XT2(4),FT1(4),FT2(4),IFAIL,AMAX1,M
        REAL B(500),ERR
        LOGICAL*1 FNAM(15)
        OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128,ERR=1000)
          DO 100 I=1,NY
          B(I)=0.0
100       CONTINUE
C
          DO 210 J=NS-NAV,NS+NAV
          CALL BININ(J,NY,NA,A,98)
            DO 200 I=1,N
            CALL FIT(XT1(I),XT2(I),FT1(I),FT2(I),NY,NA,A,AMAX1
        1,KPLUS1,ERR,M,IFAIL)
            IF (IFAIL.NE.0) TYPE *,'ERROR IN ROUTINE "FIT"'
200         CONTINUE
C
            DO 220 K=1,NY
            B(K)=B(K)+FLOAT(A(K))
220         CONTINUE
210       CONTINUE
        AMAX1=AMAX
          DO 110 I=1,NY
          A(I)=INT(B(I)/FLOAT((2*NAV)+1))
          IF (AMAX1.LT.A(I)) AMAX1=A(I)
110       CONTINUE
        CLOSE(UNIT=98)
        CALL TKPLT(NY,NA,A,AMAX1,AMIN,XDIV,YDIV)
        IF (AMAX1.NE.AMAX) TYPE *,'******* SCALE DIFFERENT *******'
        RETURN
```

```
1000    TYPE *,'ERROR OPENING FILE IN FTPLT - EXITING ROUTINE'
        GOTO 2000
2000    CALL EXIT
        END
C       *******************************************************
        SUBROUTINE FIT(XT1,XT2,FT1,FT2,NY,NA,A,AMAX,KPLUS1,ERR,M,IFAIL)
C       FITS ACCROSS AN AREA BETWEEN XT1 AND XT2.
C       FIT USES THE NAG LIBRARY ROUTINE E02ADF.
C
        INTEGER XT1,XT2,NY,NA,A(NA),AMAX,FT1,FT2
        INTEGER M,KPLUS1,NROWS,IFAIL
        REAL ERR
        DOUBLE PRECISION X(70),Y(70),W(70),WORK1(3,70),WORK2(2,3)
        1,C(3,3),S(3),XT,AT
        DATA NROWS/3/
        DATA W/70*1D1/
        IFAIL=0
        M=0
C
          DO 120 J=FT1,XT1-1
          IF (A(J).LT.0) GOTO 120
          M=M+1
          IF (M.GT.70) GOTO 1000
          X(M)=FLOAT((2*J)-FT2-FT1)/FLOAT(FT2-FT1)
          Y(M)=FLOAT(A(J))
120       CONTINUE
C
          DO 125 J=XT2+1,FT2
          IF (A(J).LT.0) GOTO 125
          M=M+1
          IF (M.GT.70) GOTO 1000
          X(M)=FLOAT((2*J)-FT2-FT1)/FLOAT(FT2-FT1)
          Y(M)=FLOAT(A(J))
125       CONTINUE
C
        IF (M.LT.KPLUS1) GOTO 1100
        CALL E02ADF(M,KPLUS1,NROWS,X,Y,W,WORK1,WORK2,C,S,IFAIL)
        IF (IFAIL.NE.0) GOTO 2000
        ERR=SNGL(S(KPLUS1))*FLOAT(M)
C
          DO 130 J=XT1,XT2
          XT=FLOAT((2*J)-FT2-FT1)/FLOAT(FT2-FT1)
          AT=(C(KPLUS1,1)/2)+(C(KPLUS1,2)*XT)
          IF (KPLUS1.EQ.3) AT=AT+(C(KPLUS1,3)*((2*XT*XT)-1))
          A(J)=INT(AT)
          IF (A(J).GT.AMAX) AMAX=A(J)
130       CONTINUE
C
        GOTO 2000
1000    TYPE *,'ARRAY SIZE EXCEEDED'
        IFAIL=1
        GOTO 2000
1100    TYPE *,'NOT ENOUGH POINTS FOR FIT'
        IFAIL=1
2000    RETURN
        END
```

B.1.10 LKBIN

The program LKBIN reads in a run from the LKB densitometer.

This routine calls the subroutines BELL, CLRSCR, CONV, FILNAM, GETNAM and LKBSTS, and the graphics macros TPIN and TPOUT.

```
C       LKBIN
C       READS DENSITOMETER SCANS TO A FILE
        LOGICAL*1 B,C(2),FNAM(15),TEMP,HOUT(13)
        INTEGER NC,NS,A(2000),AT(100),NY,NL,IERR,CTOT,ATMP,NTMP
        INTEGER YB,YE,YS,AM(2),H(21,13),HT(21,13)
C       B,C,D ARE USED FOR INTEGER TO CHARACTER CONVERSION
C       H IS OUTPUT HEADER, HT IS TEMPORARY HEADER FOR INPUT
C       FNAM IS FILENAME FOR OUTPUT. TEMP IS USED FOR TEMPORARY
C       RESPONSES FROM KEYBOARD ETC.. NC IS NO. OF CHARS. IN
C       CURRENT LINE. NS IS CURRENT SCAN NUMBER. A IS DECODED
C       DATA ARRAY. AT IS CODED DATA ARRAY FOR CURRENT LINE.
C       NY IS NUMBER OF Y-POINTS EXPECTED IN SCAN. NL IS NUMBER
C       OF LINES OF DATA. IERR IS AN ERROR COUNT. CTOT IS NUMBER
C       OF Y-POINTS IN SCAN - USED AS CHECK. ATMP IS USED FOR
C       CHECKSUM CALC. AND AS NO. OF POINTS WRITTEN OUT.
C       YB,YE,YS,AM(=AMAX,AMIN) ARE VALUES IN HEADER
        DATA FNAM/'D','L','0',':',3*' ',3*'0','.','D','A';'T',0/
        DATA NS/0/
C       **************************************************
        CALL TPOUT(19)                 ! CONTROL S
        CALL CLRSCR
        TYPE *,' ROUTINE TO READ DATA FROM DENSITOMETER'
        CALL LKBSTS
C       **************************************************
C       RUN IDENTIFIER
        TYPE *,' PLEASE ENTER OUTPUT FILE SPECIFICATIONS'
140     CALL GETNAM(FNAM)
        IF (IERR.GT.0) GOTO 400
        TYPE 5
        ACCEPT 30,TEMP
        IF (TEMP.NE.'N') CALL TPOUT(63)
        TYPE 10                         ! LKB READY
        ACCEPT 20
C       *****************************************************
C       READ HEADER
        CALL CLRSCR
        NS=0
300     CALL TPOUT(17)                  ! CONTROL Q
        NS=NS+1
        CTOT=0
        TYPE *,'SCAN ',NS
        TYPE *,'HEADER'
        CALL TPOUT(17)                  ! CONTROL Q
        DO 100 I=1,21
        DO 110 J=1,13
        CALL TPIN(HT(I,J))
        IF (HT(I,J).EQ.26) GOTO 500     ! CONTROL Z
        IF (HT(1,1).NE.66) GOTO 1700    ! B
        IF (HT(I,J).EQ.13) GOTO 100     ! CR
 110    CONTINUE
 100    CONTINUE
C       *****************************************************
C       READ DATA LINE BY LINE
        NL=0
120     NL=NL+1
        TYPE *,'LINE ',NL
        CALL TPIN(AT(1))
        IF (AT(1).EQ.83) GOTO 200       ! 'S'
        IF (AT(1).NE.65) GOTO 1000      ! 'A'
        CALL TPIN(AT(2))
        NC=AT(2)-32
        DO 130 I=3,NC+1
        CALL TPIN(AT(I))
        IF (AT(I).EQ.13.AND.I.NE.(NC+1)) GOTO 1100      ! CR
 130    CONTINUE
        IF (AT(NC+1).NE.13) GOTO 1100
        IF ((CTOT+(NC/2)-1).GT.2000) GOTO 1800
```

```
      ATMP=0
      DO 150 I=1,(NC-4)/2
      A(I+CTOT)=((AT((2*I)+1)-32)*64)+(AT((2*I)+2)-32)
      ATMP=ATMP+A(I+CTOT)
 150  CONTINUE
      A(CTOT+(NC/2)-1)=((AT(NC-1)-32)*16)+(AT(NC)-32)
      ATMP=ATMP+A(CTOT+(NC/2)-1)
      IF (MOD(ATMP,256).NE.0) GOTO 1400       ! CHECKSUM
      ATMP=(NC-4)/2
      CTOT=CTOT+ATMP
      GOTO 120
C     **************************************************
C     <CR>,S RECEIVED
200   CALL TPIN(AT(2))
      CALL TPIN(AT(3))
      CALL TPIN(AT(4))
      CALL TPOUT(19)                  ! CONTROL S
      CALL TPIN(AT(5))
      IF (AT(2).NE.84) GOTO 1600      ! T
      IF (AT(3).NE.79) GOTO 1600      ! O
      IF (AT(4).NE.80) GOTO 1600      ! P
      IF (AT(5).NE.13) GOTO 1600      ! <CR>
C     **************************************************
C     SCAN COMPLETE
C     CALCULATE AMAX,AMIN,ETC
      IF (NS.GT.1) GOTO 160
      DO 170 I=1,21
      DO 180 J=1,13
      H(I,J)=HT(I,J)
180   CONTINUE
170   CONTINUE
      CALL CONV(H,17,YB)
      CALL CONV(H,18,YE)
      CALL CONV(H,19,YS)
      NY=INT(FLOAT(YE-YB)*25/FLOAT(YS))
      GOTO 400
 160    CALL CONV(H,16,NTMP)
        CALL CONV(HT,16,ATMP)
        IF (NTMP.GT.ATMP) GOTO 1900
        DO 165 I=1,13
        H(16,I)=HT(16,I)
  165   CONTINUE
        DO 260 J=10,11
        ATMP=J
        CALL CONV(H,ATMP,AM(1))
        CALL CONV(HT,ATMP,AM(2))
        IF (J.EQ.10.AND.AM(1).LE.AM(2)) GOTO 260
        IF (J.EQ.11.AND.AM(1).GE.AM(2)) GOTO 260
        DO 250 I=3,13
        H(J,I)=HT(J,I)
 250    CONTINUE
 260    CONTINUE
C       **************************************************
400     CALL FILNAM(FNAM,NS)
        OPEN(UNIT=99,NAME=FNAM,TYPE='NEW',ERR=1200)
        IERR=10
        WRITE (99,95,ERR=1300) CTOT,(A(I),I=1,CTOT)
        CLOSE(UNIT=99)
        IF (CTOT.NE.NY.AND.NS.EQ.1) GOTO 1500
        GOTO 300
C       **************************************************
C       RUN COMPLETE
500     TYPE *,'^Z RECEIVED'
530     CALL FILNAM(FNAM,0)
        OPEN(UNIT=99,NAME=FNAM,TYPE='NEW',ERR=1210)
          DO 510 I=1,21
          J=0
520       J=J+1
          IF (H(I,J).EQ.13) GOTO 540
          TEMP=H(I,J)
          ENCODE(1,30,HOUT(J)) TEMP
          IF (J.EQ.13) GOTO 540
        GOTO 520
540       DO 560 K=J,13
          HOUT(K)=' '
560       CONTINUE
        WRITE(99,80,ERR=1300) (HOUT(K),K=1,13)
510     CONTINUE
      CLOSE(UNIT=99)
      CALL CLRSCR
      TYPE *,'NO. OF SCANS =',NS-1
      GOTO 2000
C     ****************************************************
C             ERROR MESSAGES
C     ****************************************************
1000  TYPE *,' NOT ABSOLUTE BINARY - EXITING ROUTINE'
      TYPE *,NS,NL
      TYPE *,AT(1)
      CALL BELL
      CALL TPOUT(27)
      GOTO 2000
C     ****************************************************
1100  TYPE *,' UNEXPECTED CARRIAGE RETURN'
      TYPE *,' EXITING ROUTINE'
      CALL BELL
      CALL TPOUT(27)
      GOTO 2000
C     ****************************************************
1200  IERR=IERR+1
      IF (IERR.GT.2) GOTO 1210
      TYPE *,' ERROR OPENING FILE - PLEASE TRY AGAIN'
      TYPE 50,(FNAM(I),I=1,15)
      GOTO 140
 1210 TYPE *,' ERROR OPENING FILE - EXITING ROUTINE'
      CALL BELL
      CALL TPOUT(27)
      GOTO 2000
C     ****************************************************
```

```
1300    TYPE *,' ERROR WRITING TO FILE - EXITING ROUTINE'
        CLOSE(UNIT=99)
        CALL BELL
        CALL TPOUT(27)
        GOTO 2000
C       ****************************************************
1400    TYPE *,' CHECKSUM ERROR'
        CALL BELL
        CALL TPOUT(27)
        GOTO 2000
C       ****************************************************
1500    TYPE *,' UNEXPECTED NUMBER OF DATA POINTS IN SCAN :'
        TYPE *,'NO. RECEIVED=',CTOT,'   NO. EXPECTED=',NY
        TYPE *,'YB',YB,'   YE',YE,'   YS',YS
        GOTO 300
C       ****************************************************
1600    TYPE *,' UNEXPECTED CHARACTERS - EXITING ROUTINE'
        CALL BELL
        CALL TPOUT(27)
        GOTO 2000
C       ****************************************************
1700    TYPE *,' FIRST CHAR. OF HEADER NOT "B"'
        TYPE *,'  CHARACTERS MISSED - EXITING ROUTINE'
        CALL BELL
        CALL TPOUT(27)
        GOTO 2000
C       ****************************************************
1800    TYPE *,'ARRAY OVERFLOW - EXITING ROUTINE'
        CALL BELL
        CALL TPOUT(27)
        GOTO 2000
C       ****************************************************
1900    TYPE *,'END OF FIRST FIELD'
        CALL TPOUT(27)
        GOTO 530
C       ****************************************************
```

```
2000    TYPE *,' END OF RUN'
        CALL EXIT
5       FORMAT(' START RUN (Y/N, CR=Y) ?',$)
10      FORMAT(' ENTER CR WHEN LKB READY:',$)
20      FORMAT(X)
30      FORMAT(A:2A:A)
35      FORMAT(' ENTER DEVICE (CR FOR DL0:) : ',$)
40      FORMAT(' ENTER 3-CHARACTER RUN IDENTIFIER : ',$)
50      FORMAT(' FILENAME IS ',15A)
60      FORMAT(' IS THIS OK (Y/N: CR=Y) ?',$)
80      FORMAT(X,13A)
90      FORMAT(I2)
95      FORMAT(I5)
        END
```

B.1.11 LKBLOG

The program LKBLOG converts a data set to or from log intensity form.

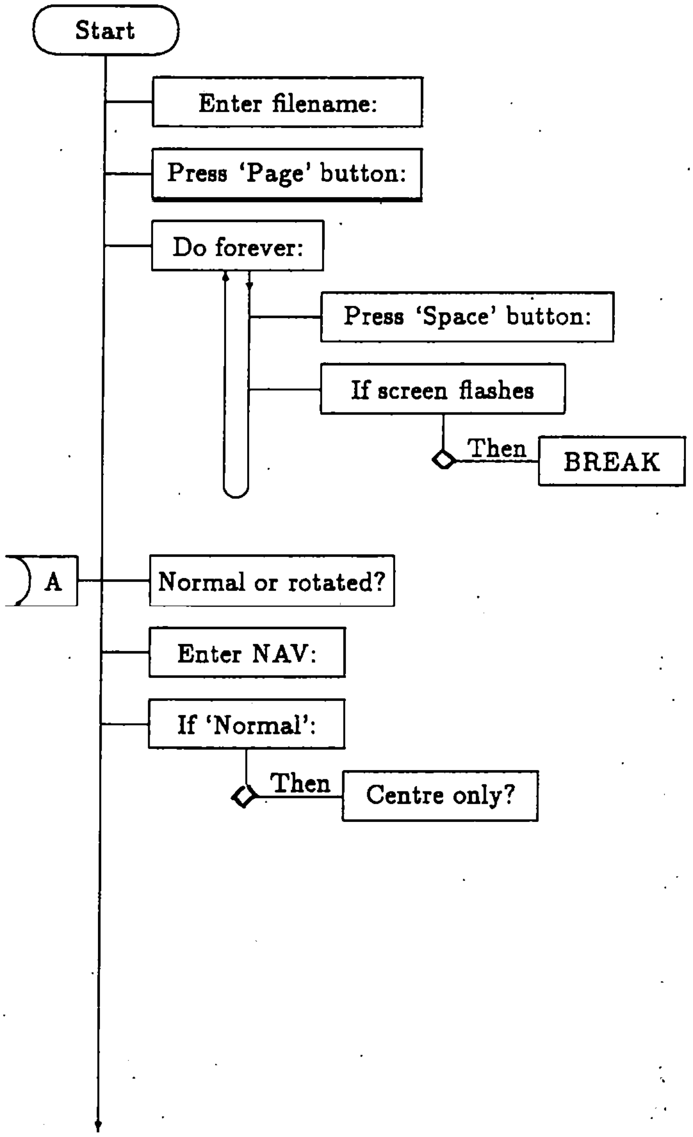
This routine calls the subroutines BININ, BINOUT, NEWNAM and OLDNAM.

```
C       LKBLOG.FOR.
C       CONVERTS TO AND FROM LOG INTENSITY DATA - LOG DATA IS
C       1000*LOG(A) TO MAINTAIN RESOLUTION WHILE STILL USING INTEGERS.
C
        LOGICAL*1 FNAM(15),FNAM1(15),TEMP,SET
        INTEGER A(2000),B(2000),NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,X0,Y0
        REAL XSCAL,YSCAL
        TYPE 10,' INPUT'                ! ENTER INPUT FILENAME
        CALL OLDNAM(FNAM,98)
        TYPE 10,'OUTPUT'                ! ENTER OUTPUT FILENAME
        CALL NEWNAM(FNAM1,99)
120     TYPE 30                         ! TO OR FROM LOG DATA?
        ACCEPT 20,SET
        IF (SET.NE.'1'.AND.SET.NE.'2') GOTO 120
        OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128,READONLY)
        OPEN(UNIT=99,NAME=FNAM1,TYPE='NEW',ACCESS='DIRECT'
        1,RECORDSIZE=128)
        READ (98'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,Y0
        IF (SET.EQ.'2') GOTO 100
        IF (AMAX.LE.0) GOTO 1000
        IF (AMIN.LT.0) AMIN=-1
        AMAX=INT((1000.*ALOG(FLOAT(AMAX)))+0.5)
        IF (AMIN.GT.0) AMIN=INT((1000.*ALOG(FLOAT(AMIN)))+0.5)
        GOTO 110
100     AMAX=INT((EXP(FLOAT(AMAX)/1000.))+0.5)
        AMIN=INT((EXP(FLOAT(AMIN)/1000.))+0.5)
110     WRITE (99'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,Y0
          DO 200 I=1,SCN
          CALL BININ(I,NY,2000,A,98)
          IF (SET.EQ.'2') GOTO 220
            DO 210 J=1,NY
            B(J)=-1
            IF (A(J).LT.1) GOTO 210
            B(J)=INT((1000.*ALOG(FLOAT(A(J))))+0.5)
210         CONTINUE
          GOTO 240
220         DO 230 J=1,NY
            B(J)=INT((EXP(FLOAT(A(J))/1000.))+0.5)
230         CONTINUE
240       CALL BINOUT(I,NY,2000,B,99)
200       CONTINUE
        CLOSE(UNIT=98)
        CLOSE(UNIT=99)
        GOTO 2000
C
1000    TYPE *,'FILE AMAX IS ZERO OR NEGATIVE'
C
2000    TYPE *,'END OF RUN'
        CALL EXIT
C
10      FORMAT(' ENTER ',A6,' FILENAME (14A) : ',$)
20      FORMAT(A)
30      FORMAT(/,'  1 - CONVERT TO LOG INTENSITY',/,'  2 - CONVERT
        1 FROM LOG INTENSITY',/,' ENTER OPTION : ',$)
        END
```

B.1.12 LKBPK

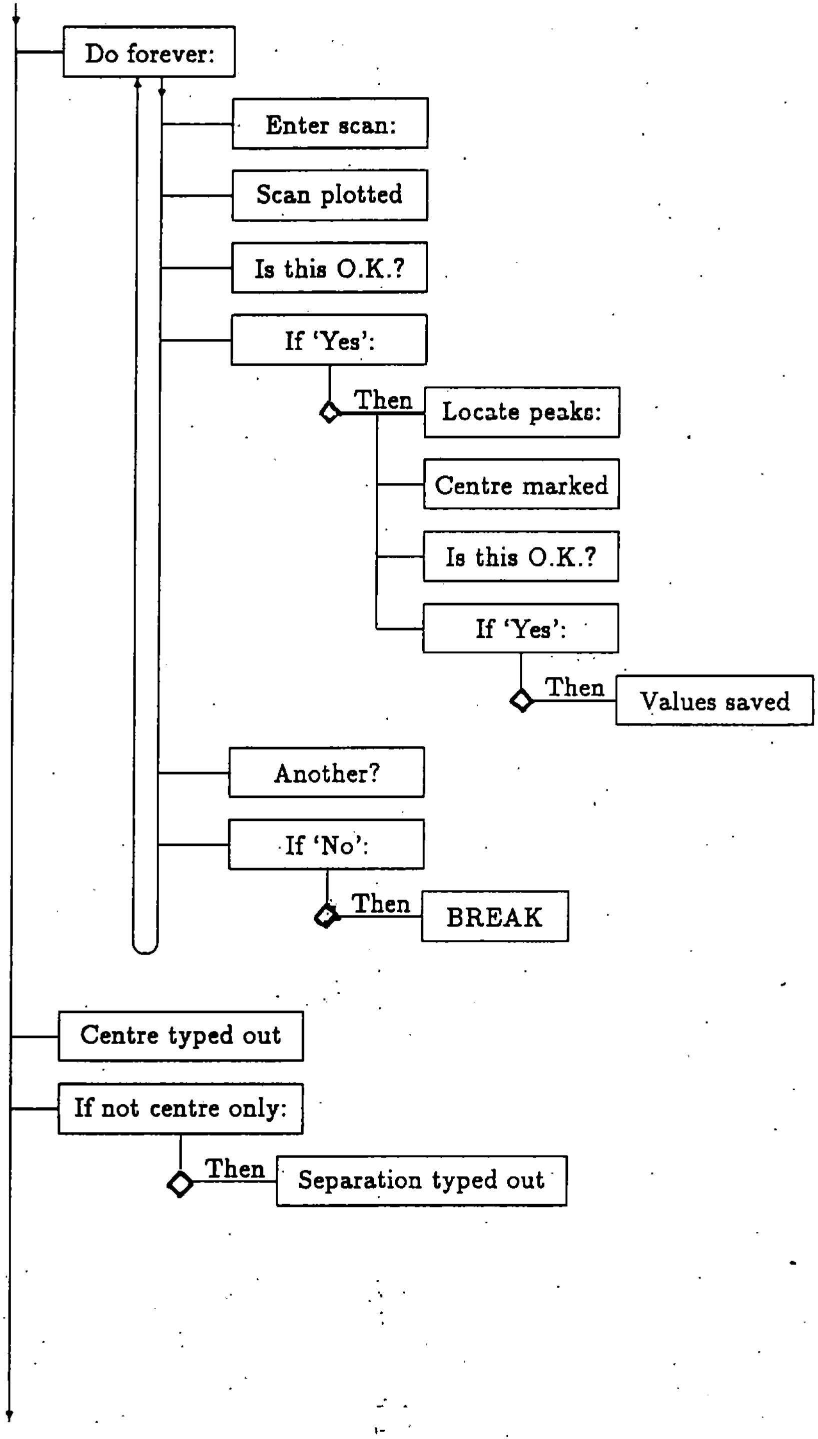
The program LKBPK locates the centre and calibrates a data set.

This routine calls the subroutines AVIN, BELL, CLRSCR, GINT, GREAL, OLDNAM, TKPLT and POSN, and the graphics macros DRAWX, MOVEX, RDCW, TEKOUT and TEKWIP.

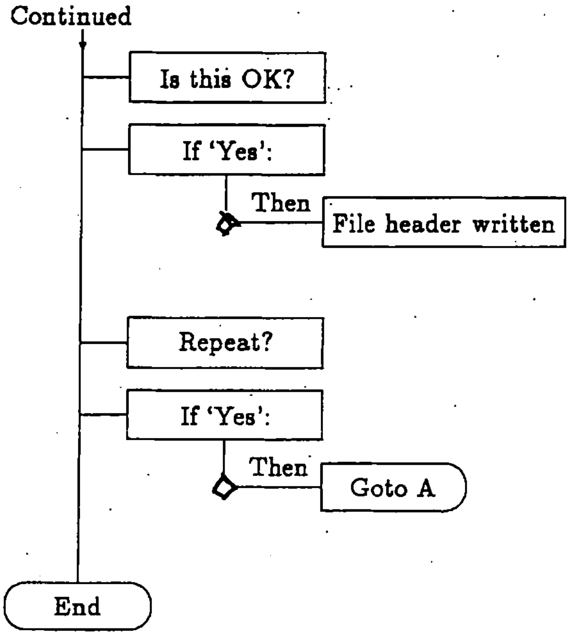


P.T.O.

Continued



P.T.O.



```
C       PROGRAM LKBPK
C       LOCATES PEAKS USING TEKTRONIX SCREEN AND FINDS
C       THE PEAK SEPARATION AND PATTERN CENTRE
        LOGICAL*1 FNAM(15),TEMP,TYPE(2)
        INTEGER A(2000),AMAX,AMIN,SCN,NS,NY,NA,NSC,A0
        INTEGER XMAX,YMAX,ZMAX,MARG,ASIZE,XDIV,YDIV
        INTEGER X1,Y1,X2,Y2,XT1,XT2
        INTEGER XSEP,X0,Y0,Z0,DZ(50),ZCEN(50),N
        INTEGER XSTEP,YSTEP,B(256),C,NAV
        REAL SDX0,SDXS,XSCAL,YSCAL,PEAK
        DATA ASIZE/2000/
        DATA XMAX,YMAX/1000,750/
        DATA FNAM/15*0/
        DATA TYPE/'N','C'/
        DATA MARG/30/
        ZMAX=XMAX
        CALL CLRSCR
        TYPE 70                           ! ENTER FILE NAME
        CALL OLDNAM(FNAM,98)
        OPEN (UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128,ERR=1000,READONLY)
        READ(98'1,ERR=1100) NY,SCN,AMAX,AMIN
        1,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,Y0
        CLOSE(UNIT=98)
        NR=INT(FLOAT(NY-1)/256)+1
        TYPE *,'AMAX -',AMAX,'       AMIN -',AMIN
        TYPE 30                           !CR TO CONTINUE
        ACCEPT 40
        CALL CLRSCR
        TYPE *,'PRESS "PAGE" ON TEKTRONIX, THEN SPACE BAR SEVERAL'
        TYPE *,'TIMES UNTIL SCREEN FLASHES'
        CALL RDCW(X1,Y1)
        CALL TEKWIP
400     CALL CLRSCR
        TYPE(2)='C'
        N=0
        Z0=0
        XSEP=0
        NA=NY
        NSC=SCN
        TYPE 80                           ! ENTER PLOT TYPE (N OR R)
        ACCEPT 40,TYPE(1)
        IF (TYPE(1).NE.'R') TYPE(1)='N'
        TYPE 15                           ! ENTER NAV
        CALL GINT(NAV)
        IF (TYPE(1).EQ.'R') GOTO 100
        TYPE 90                           ! SEPN AND/OR CENTRE
        ACCEPT 40,TYPE(2)
        IF (TYPE(2).NE.'C') TYPE(2)='S'
        GOTO 160
100     NA=SCN
        NSC=NY
160     TYPE 10,NSC-NAV                   ! ENTER SCAN NUMBER
```

```
         CALL GINT(NS)
         IF (NS.GT.NSC-NAV) GOTO 1200
         IF (NS.LT.1+NAV) GOTO 1200
         OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
         1,RECORDSIZE=128,ERR=1000)
         CALL AVIN(NS,NY,SCN,ASIZE,A,NAV,TYPE(1),98)
         CLOSE(UNIT=98)
         CALL TKPLT(NA,ASIZE,A,AMAX,AMIN,XDIV,YDIV)
         TYPE 60                         ! OK?
         ACCEPT 40,TEMP
         IF (TEMP.EQ.'N') GOTO 120
         CALL POSN(XT1,XT2,X1,X2,MARG,NA,ZMAX)
         CALL MOVEX((X1+X2)/2,0)
         CALL DRAWX((X1+X2)/2,750)
         IF (TYPE(2).EQ.'S') TYPE *,'SEPARATION =',(XT2-XT1)
         TYPE *,'CENTRE IS AT',((XT1+XT2)/2)
         TYPE 60                         ! OK?
         ACCEPT 40,TEMP
         IF (TEMP.EQ.'N') GOTO 120
         N=N+1
         DZ(N)=XT2-XT1
         ZCEN(N)=(XT1+XT2)/2
120      TYPE 50                         ! ANOTHER?
         ACCEPT 40,TEMP
         IF (TEMP.NE.'N') GOTO 160
         IF (N.EQ.0) GOTO 500
         DO 130 I=1,N
         Z0=Z0+ZCEN(I)
         XSEP=XSEP+DZ(I)
 130     CONTINUE
         XSEP=XSEP/N
         Z0=Z0/N
         DO 140 I=1,N
         SDX0=SDX0+(FLOAT(ZCEN(I)-Z0)**2)
         SDXS=SDXS+(FLOAT(DZ(I)-XSEP)**2)
 140     CONTINUE
```

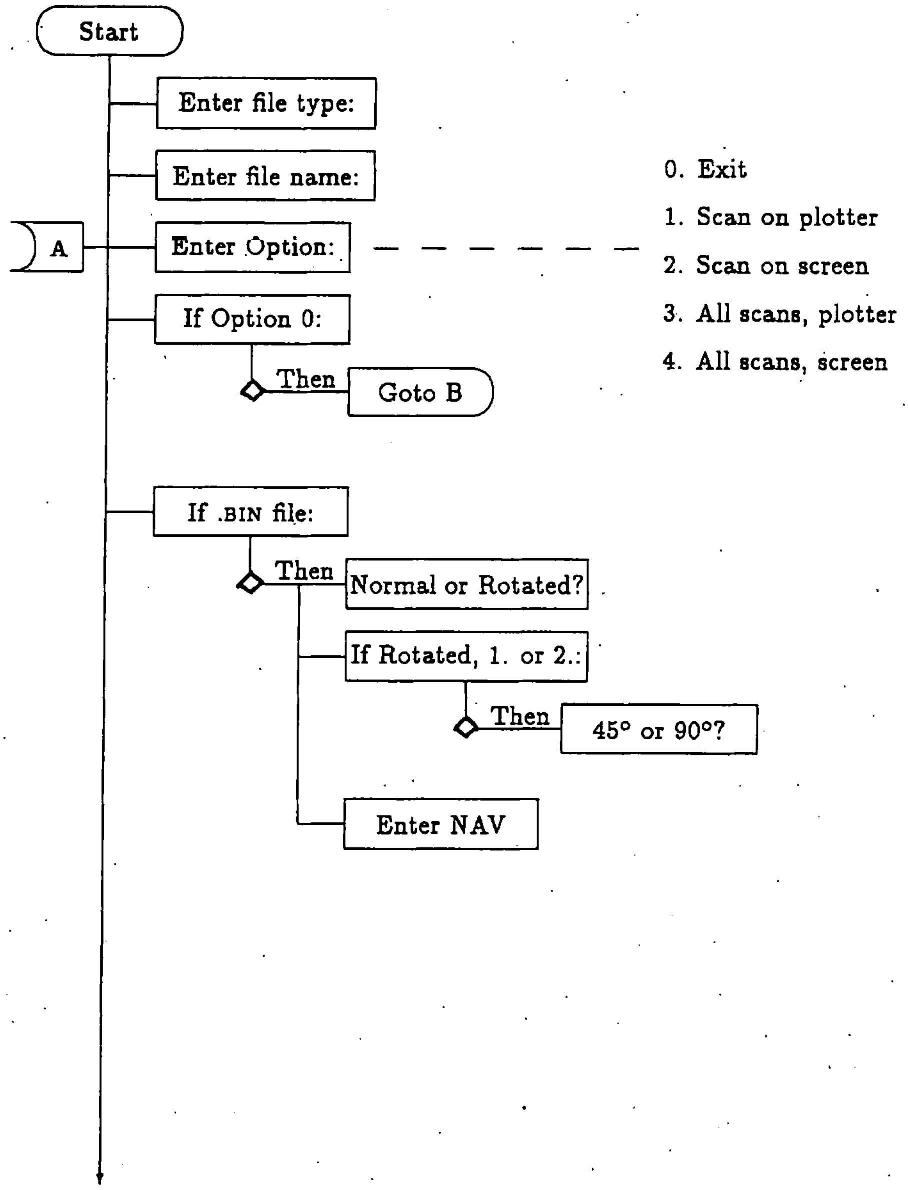
```
      SDXO=SQRT(SDXO)
      SDXS=SQRT(SDXS)
      IF (TYPE(2).EQ.'C') GOTO 300
      TYPE *,'SEPARATION =',XSEP
      IF (N.GT.2) TYPE *,'STD. DEV.',SDXS
300   TYPE *,'CENTRE IS AT',ZO
      IF (N.GT.2) TYPE *,'STD. DEV.',SDXO
      TYPE 60                         ! OK ?
      ACCEPT 40,TEMP
      IF (TEMP.EQ.'N') GOTO 500
      IF (TYPE(1).EQ.'R') XO=ZO
      IF (TYPE(1).NE.'R') YO=ZO
      IF (TYPE(2).EQ.'C') GOTO 310
I     IF (XSEP.LE.0) GOTO 1300
      TYPE 92                         ! ENTER PEAK SEPN.
      CALL GREAL(PEAK)
      YSCAL=2/(FLOAT(XSEP)*PEAK)
      XSCAL=YSCAL*XSTEP/YSTEP
      TYPE *,'X-SCALE :',XSCAL,'      Y-SCALE :',YSCAL
      TYPE *,' (UNITS - SEPN. OF POINTS IN RECIPROCAL NM)'
      TYPE 60                         ! OK ?
      ACCEPT 40,TEMP
      IF (TEMP.EQ.'N') GOTO 500
310   OPEN (UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
      1,RECORDSIZE=128,ERR=1000)
      WRITE(98'1,ERR=1100) NY,SCN,AMAX,AMIN
      1,XSTEP,YSTEP,AO,XSCAL,YSCAL,XO,YO
      CLOSE(UNIT=98)
500   TYPE 95                         ! REPEAT ?
      ACCEPT 40,TEMP
      IF (TEMP.NE.'N') GOTO 400
      GOTO 2000
C     *******************************************************
1000  TYPE *,'ERROR OPENING FILE - EXITING ROUTINE'
      GOTO 2000
1100  TYPE *,'ERROR READING FILE - EXITING ROUTINE'
```

```
      CLOSE(UNIT=98)
      GOTO 2000
1200  TYPE *,'INVALID SCAN NUMBER !'
      CALL BELL
      GOTO 160
1300  TYPE *,'SEPARATION = 0 - SAVING CENTRE ONLY'
      CALL BELL
      GOTO 310
C     ********************************************************
2000  TYPE *,'END OF RUN'
      CALL TEKOUT(31)
      CALL EXIT
C     ********************************************************
10    FORMAT(' ENTER SCAN NUMBER (MAX.=',I4,') :',$)
15    FORMAT(' PLOT N SCANS EACH SIDE
      1 OF CHOSEN ONE - ENTER N (I5) : ',$)
30    FORMAT(' ENTER <CR> TO CONTINUE',$)
40    FORMAT(A:13A)
50    FORMAT(' ANOTHER (CR=Y) ?',$)
60    FORMAT(' IS THIS OK (CR=Y) ?',$)
70    FORMAT(' ENTER FILE NAME :',$)
80    FORMAT(' N - NORMAL, R - ROTATED',/,' ENTER PLOT TYPE
      1 (CR=N) :',$)
90    FORMAT(' S - FIND SEPN. AND CENTRE, C - FIND CENTRE ONLY',/
      1,' ENTER YOUR CHOICE (CR=S) :',$)
92    FORMAT(' ENTER PEAK SEPARATION FOR CALIBRATION'
      1,/,' (PARAMYOSIN=14.4 NM) :',$)
95    FORMAT(' REPEAT (Y/N, CR=Y) ?',$)
      END
```

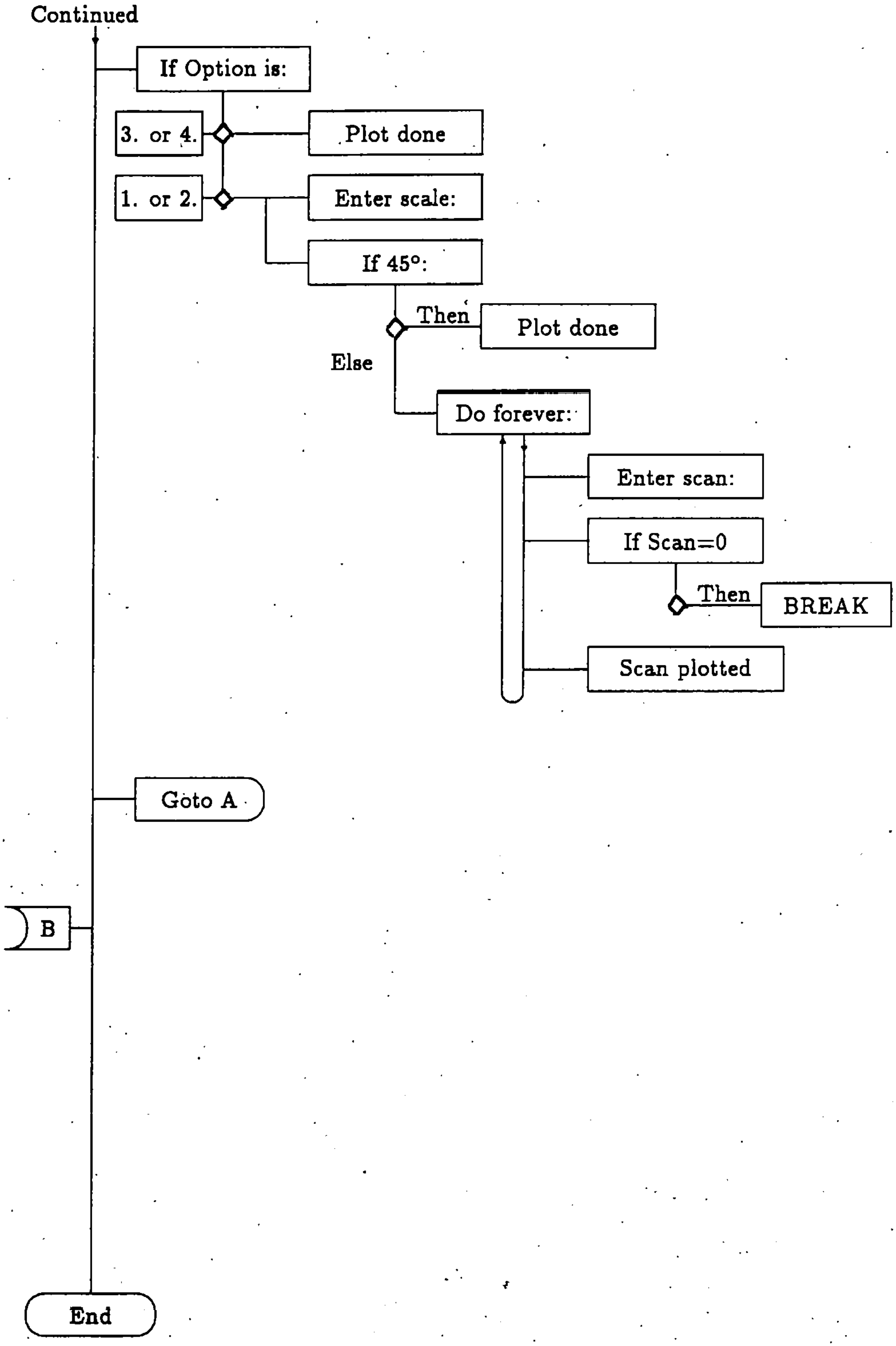
B.1.13 LKBPLT

The program LKBPLT makes various plots from data files.

This routine calls the subroutines AVIN, BITGRP, CLRSCR, FILNAM, GINT, GREAL, GRPLT, GR2PLT, HEADER, OLDNAM, TKPLT and TK2PLT.



P.T.O.


Continued
If Option is:
3. or 4.
Plot done
1. or 2.
Enter scale:
If 45°:
Then
Plot done
Else
Do forever:
Enter scan:
If Scan=0
Then
BREAK
Scan plotted
Goto A
B
End

```
C       LKBPLT
C       PLOTS LKB OUTPUT
C
        LOGICAL*1 FNAM(15),TEMP,TYPE(2),PLOT
        INTEGER NS,SCN,XSTEP,YSTEP,A0,X0,Y0,STEP,XDIV,YDIV,X1,X2,Y1,Y2
        INTEGER AMAX,AMAX1,AMIN,NY,A(2000),AM(4000),NA,NSC,THET,ASIZE
        INTEGER NR,REC,C,B(256),IERR,IFAIL,NAV
        REAL XSCAL,YSCAL,CTMP
        COMMON /PLT/A,AM
        DATA ASIZE/2000/
        DATA FNAM/'D','L','0',':',6*'0','.','D','A','T',0/
        DATA TYPE/'B','N'/
        DATA NY/0/
        CALL CLRSCR
        TYPE *,'B=BINARY DIRECT ACCESS,D=.DAT'
        TYPE 30                    ! ENTER FILE TYPE
        ACCEPT 20,TYPE(1)
        IF (TYPE(1).NE.'D') GOTO 100
        CALL HEADER(FNAM,AMAX,AMIN,SCN,XSTEP,YSTEP)
        GOTO 110
100     TYPE(1)='B'
        TYPE 90                    ! ENTER FILE NAME
        CALL OLDNAM(FNAM,98)
        OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128)
        READ(98'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0
        1,XSCAL,YSCAL,X0,Y0
        CLOSE(UNIT=98)
        NR=INT(FLOAT(NY-1)/256)+1
110     TYPE *,'AMAX=',AMAX,'      AMIN=',AMIN
C       ****************************************************
C       MENU
400     CALL CLRSCR
        NA=NY
        NSC=SCN
        NAV=0
        THET=0
        TYPE *,'TYPE OF PLOT:'
        TYPE 10                 ! MENU
        ACCEPT 20,PLOT
        IF (PLOT.LT.48.OR.PLOT.GT.52) GOTO 410
        IF (TYPE(1).NE.'B') GOTO 425
        TYPE *,'N=NORMAL,R=ROTATED'
        TYPE 40                 ! ENTER PLOT TYPE
        ACCEPT 20,TYPE(2)
        IF (TYPE(2).NE.'R') GOTO 430
430     IF (TYPE(2).NE.'R') TYPE(2)='N'
        IF (TYPE(2).NE.'R') GOTO 420
        NA=SCN
        NSC=NY
        TYPE 95                 ! ENTER ANGLE
        CALL GINT(THET)
        IF (THET.EQ.45) GOTO 700
420     TYPE 60                 ! ENTER NAV
        CALL GINT(NAV)
425     IF (PLOT.EQ.'1') GOTO 300
        IF (PLOT.EQ.'2') GOTO 300
        IF (PLOT.EQ.'3') GOTO 500
        IF (PLOT.EQ.'4') GOTO 510
410     IF (PLOT.EQ.' ') GOTO 2000
        TYPE *,'INVALID CHOICE'
        GOTO 400
C       *****************************************************
C       SELECT AND READ IN SCAN TO BE PLOTTED
300     CALL CLRSCR
        TYPE 75                                 ! ENTER INTENSITY SCALE
FACTOR
        CALL GREAL(CTMP)
        IF (CTMP.EQ.0.) CTMP=1.0
        AMAX1=INT(FLOAT(AMAX)/CTMP)
305     TYPE *,'ENTER SCAN NUMBER (0 FOR END OF RUN) :'
```

```
      TYPE *,'MAX. SCAN NO.=',NSC-NAV
310   CALL GINT(NS)
      IF (NS.EQ.0) GOTO 400
      IF (NS.GT.NSC-NAV.OR.NS.LT.NAV+1) GOTO 1300
      IF (TYPE(1).NE.'B') GOTO 340
      OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
      1,RECORDSIZE=128,ERR=1100)
      CALL AVIN(NS,NY,SCN,ASIZE,A,NAV,TYPE(2),98)
      CLOSE(UNIT=98)
      GOTO 350
340   CALL FILNAM(FNAM,NS)
      OPEN(UNIT=99,NAME=FNAM,TYPE='OLD',ERR=1100)
      READ(99,70,ERR=1200) NY,(A(I),I=1,NY)
      CLOSE(UNIT=99)
      NA=NY
      GOTO 350
700   CALL CLRSCR
      TYPE 75                           ! ENTER INTENSITY SCALE FACTOR
      CALL GREAL(CTMP)
      IF (CTMP.EQ.0.) CTMP=1.0
      AMAX1=INT(FLOAT(AMAX)/CTMP)
      TYPE *,'CALCULATING PLOT . . . . . .'
      NSC=INT(FLOAT(NY*YSTEP)/(FLOAT(XSTEP**2+YSTEP**2)**0.5))
      X1=1.
      IF (X0.GT.Y0) X1=X0-Y0+1
      NA=SCN-X1+1
      IF (NY-Y0.LT.SCN-X0) NA=NY-Y0+X0-X1+1
      X2=0
      IF (2*(X0-X1).GE.NA+X1-X0-1) GOTO 760    ! X0 AT L.H.S.
      NA=NA+X1-X0
      X1=X0
      Y1=Y0
      IF (NA.LT.NSC) GOTO 755
      NA=NSC
      GOTO 740
755   DO 776 I=NA+1,NSC
```

```
        A(I)=0
776     CONTINUE
        GOTO 740
760     IF (X0-X1.LE.2*(NA+X1-X0-1)) GOTO 770   ! X0 AT R.H.S.
        NA=X0-X1+1
        X1=X0-NSC
        Y1=Y0-X0+X1
        IF (NA.LT.NSC) GOTO 765
        NA=NSC
        GOTO 740
765     DO 766 I=1,NSC-NA
        A(I)=0
766     CONTINUE
        X2=NSC-NA+1
        X1=X1+X2
        Y1=Y1+X2
        GOTO 740
770     X2=INT(0.5+FLOAT(NSC)/2)                ! X0 IN CENTRE
        IF (X0-X2.GT.X1) NA=NA+X1-X0+X2
        IF (X0-X2.GT.X1) X1=X0-X2
        X2=X1-X0+X2
        Y1=Y0-X0+X1
        IF (NA.GT.NSC-X2) NA=NSC-X2
        GOTO 740
C
740     X1=X1-1
        Y1=Y1-1
        OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128)
          DO 750 I=1,NA
          X1=X1+1
          Y1=Y1+1
          C=INT(FLOAT(Y1-1)/256)
          REC=2+((X1-1)*NR)+C
          READ(98'REC) (B(J),J=1,256)
          C=Y1+1-(256*C)
          A(I+X2)=B(C)
750       CONTINUE
        CLOSE(UNIT=98)
        NA=NSC
        IF (PLOT.EQ.'2') GOTO 320
        TYPE 80                     ! CR WHEN PLOTTER READY
        ACCEPT 20
        CALL BITGRP(NA,ASIZE,A,AMAX1,AMIN,1,NA)
        GOTO 400
350     IF (PLOT.EQ.'2') GOTO 320
C       ******************************************************
C       CALL GRPLT
        TYPE 80                     ! CR WHEN PLOTTER READY
        ACCEPT 20
        CALL GRPLT(FNAM,NA,ASIZE,A,AMAX1,AMIN)
        IF (THET.EQ.45) GOTO 400
        GOTO 300
C       ******************************************************
C       CALL TKPLT
320     CALL TKPLT(NA,ASIZE,A,AMAX1,AMIN,XDIV,YDIV)
        TYPE *,'XDIV =',XDIV,'       YDIV =',YDIV
        IF (THET.EQ.45) GOTO 400
        GOTO 300
C       ******************************************************
500     CALL CLRSCR
        TYPE 80                     ! CR WHEN PLOTTER READY
        ACCEPT 20
        CALL GR2PLT(FNAM,AMAX,AMIN,SCN,NAV,TYPE,IFAIL)
        IF (IFAIL.NE.0) GOTO 2000
        GOTO 400
510     CALL CLRSCR
        CALL TK2PLT(FNAM,AMAX,AMIN,SCN,NAV,TYPE,IFAIL)
        IF (IFAIL.NE.0) GOTO 2000
        GOTO 400
C       ******************************************************
1100    IERR=IERR+1
```

```
        TYPE *,'ERROR OPENING FILE'
        TYPE 20,(FNAM(I),I=1,14)
        IF (IERR.GT.5) GOTO 1110
        TYPE *,'ENTER NEW SCAN NUMBER'
        GOTO 310
1110    TYPE *,'EXITING ROUTINE'
        GOTO 2000
C       *****************************************************
1200    TYPE *,'ERROR READING FILE'
        TYPE *,'EXITING ROUTINE'
        GOTO 2000
C       *****************************************************
1300    TYPE *,'SCAN NUMBER IS OUT OF RANGE - '
        TYPE *,NAV+1,' TO ',SCN-NAV
        TYPE *,'PLEASE REPEAT'
        GOTO 305
C       *****************************************************
2000    TYPE *,'END OF RUN'
        CALL EXIT
10      FORMAT(/,/,X,'  1 - SINGLE SCANS ON GRAPH PLOTTER'
        1  ,/,X,'       2 - SINGLE SCANS ON TEKTRONIX SCREEN'
        2  ,/,X,'       3 - ALL SCANS ON GRAPH PLOTTER'
        3  ,/,X,'       4 - ALL SCANS ON TEKTRONIX SCREEN'
        4  ,/,X,'       <CR> - EXIT PROGRAM'
        5  ,/,/,/,X,'ENTER OPTION:',$)
20      FORMAT(A:2A:A:10A)
30      FORMAT(' ENTER FILE TYPE (CR=B) :',$)
40      FORMAT(' ENTER PLOT TYPE (CR=N) :',$)
60      FORMAT(' AVERAGE N SCANS EACH SIDE OF THAT SELECTED -',/
        1,' ENTER N (I5) : ',$)
70      FORMAT(I5)
75      FORMAT(' ENTER INTENSITY SCALE FACTOR (CR=1.) : ',$)
80      FORMAT(' ENTER <CR> WHEN PLOTTER READY',$)
90      FORMAT(' ENTER FILE NAME :',$)
95      FORMAT(' ENTER ANGLE OF ROTATION - 45 OR 90 : ',$)
        END
```

B.1.14 LKBREV

The program LKBREV reverses the x and/or y axes of a data set, or rotates the data set about the origin.

This routine calls the subroutines CLRSCR, NEWNAM and OLDNAM.

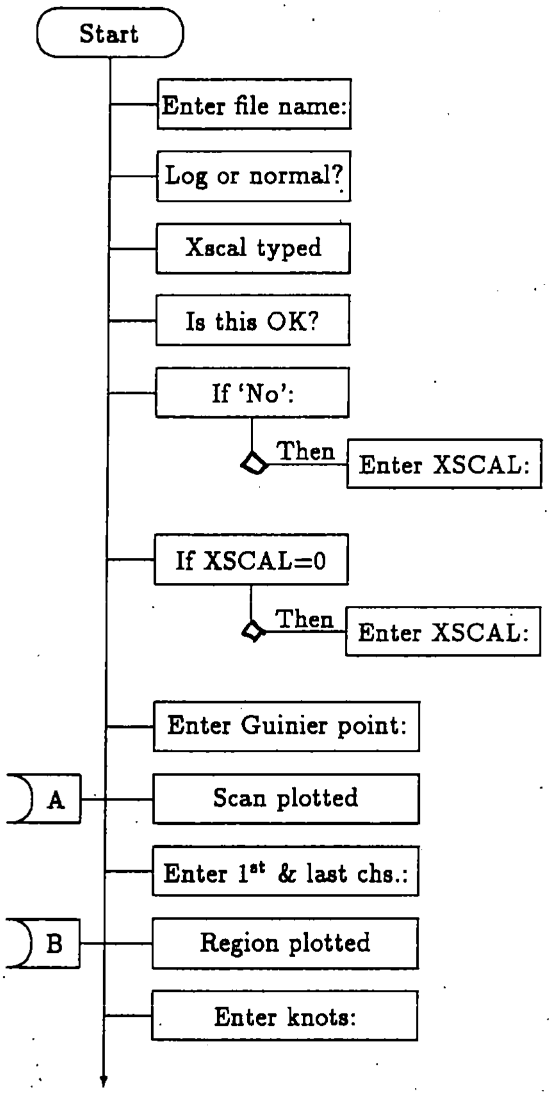
```
C       LKBREV
C       REVERSES X AND/OR Y
C       SUBROUTINES ARE IN LKBLIB
C
        LOGICAL*1 FNAM1(15),FNAM2(15),TEMP,REV(2)
        REAL XSCAL,YSCAL
        INTEGER A(2000),B(2000),NY,SCN,AMAX,AMIN,XSTEP,YSTEP
        1,A0,X0,Y0,X01,Y01,NR,NR1,REC,NS,C
        DATA FNAM1,FNAM2/15*0,15*0/
        DATA REV/2*.FALSE./
        CALL CLRSCR
C       ************ FILENAMES ******************************
100     TYPE 10,' INPUT'                  ! ENTER FILENAME
        CALL OLDNAM(FNAM1,98)
        TYPE 10,'OUTPUT'                  ! ENTER FILENAME
        CALL NEWNAM(FNAM2,98)
        TYPE 30                   ! REVERSE X ?
        ACCEPT 20,TEMP
        IF (TEMP.NE.'N') REV(1)=.TRUE.
        TYPE 40                   ! REVERSE Y ?
        ACCEPT 20,TEMP
        IF (TEMP.NE.'N') REV(2)=.TRUE.
        IF (REV(1).EQ..TRUE..OR.REV(2).EQ..TRUE.) GOTO 120
        TYPE 50                   ! ROTATE THROUGH 90 DEGREES?
        ACCEPT 20,TEMP
        IF (TEMP.NE.'N') GOTO 300
        GOTO 2000
C       ************ HEADER *********************************
120     OPEN(UNIT=98,NAME=FNAM1,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128,READONLY,ERR=1100)
        OPEN(UNIT=99,NAME=FNAM2,TYPE='NEW',ACCESS='DIRECT'
        1,RECORDSIZE=128,ERR=1100)
        READ(98'1,ERR=1200) NY,SCN,AMAX,AMIN,XSTEP,YSTEP
        1,A0,XSCAL,YSCAL,X0,Y0
        IF (REV(1).EQ..TRUE.) X0=SCN-X0+1
        IF (REV(2).EQ..TRUE.) Y0=NY-Y0+1
        IF (X0.EQ.SCN+1) X0=0
        IF (Y0.EQ.NY+1) Y0=0
        WRITE(99'1,ERR=1200) NY,SCN,AMAX,AMIN,XSTEP,YSTEP
        1,A0,XSCAL,YSCAL,X0,Y0
        NR=INT(FLOAT(NY-1)/256)+1
C       ************ REVERSING X ***************************
C
          DO 200 NS=1,SCN
C
            DO 210 I=1,NR
            IF (REV(1).EQ..TRUE.) REC=((SCN-NS)*NR)+I+1
            IF (REV(1).EQ..FALSE.) REC=((NS-1)*NR)+I+1
            READ(98'REC,ERR=1200) (A(J),J=((I-1)*256)+1,I*256)
            IF (REV(2).EQ..TRUE.) GOTO 210
            REC=((NS-1)*NR)+I+1
            WRITE(99'REC,ERR=1200) (A(J),J=((I-1)*256)+1,I*256)
 210        CONTINUE
C
          IF (REV(2).EQ..FALSE.) GOTO 200
C         ************ REVERSING Y ***************************
C
            DO 230 I=1,NY
            B(I)=A(NY-I+1)
 230        CONTINUE
C
            DO 240 I=1,NR
            REC=((NS-1)*NR)+I+1
            WRITE(99'REC,ERR=1200) (B(J),J=((I-1)*256)+1,I*256)
240         CONTINUE
C
200       CONTINUE
```

```
C
        CLOSE(UNIT=98,ERR=1100)
        CLOSE(UNIT=99,ERR=1100)
        GOTO 2000
C       ************ ROTATE ********************************
300     REV(1)=.TRUE.
        TYPE 60                 ! CLOCKWISE ?
        ACCEPT 20,TEMP
        IF (TEMP.NE.'N') REV(1)=.FALSE.
        OPEN(UNIT=98,NAME=FNAM1,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128,READONLY)
        OPEN(UNIT=99,NAME=FNAM2,TYPE='NEW',ACCESS='DIRECT'
        1,RECORDSIZE=128)
        READ(98'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,XSCAL,YSCAL
        1,X0,Y0
        X01=NY-Y0+1
        Y01=X0
        IF (REV(1).EQ..TRUE.) X01=Y0
        IF (REV(1).EQ..TRUE.) Y01=SCN-X0+1
        WRITE(99'1) SCN,NY,AMAX,AMIN,YSTEP,XSTEP,A0,YSCAL,XSCAL
        1,X01,Y01
        NR=INT(FLOAT(NY-1)/256)+1
        NR1=INT(FLOAT(SCN-1)/256)+1
C
          DO 320 I=1,NY
          C=NY-I+1-(256*INT(FLOAT(NY-I)/256))
          IF (REV(1).EQ..TRUE.) C=I-(256*INT(FLOAT(I-1)/256))
            DO 310 NS=1,SCN
            REC=((NS-1)*NR)+2+INT(FLOAT(NY-I)/256)
            IF (REV(1).EQ..TRUE.)
        1   REC=(SCN-NS)*NR+2+INT(FLOAT(I-1)/256)
            READ(98'REC) (B(J),J=1,256)
            A(NS)=B(C)
310         CONTINUE
          REC=((I-1)*NR1)+1
            DO 330 J=1,NR1
            WRITE(99'REC+J) (A(K),K=(J-1)*256+1,J*256)
330         CONTINUE
320       CONTINUE
C
        CLOSE(UNIT=98)
        CLOSE(UNIT=99)
        GOTO 2000
C       ************ ERROR MESSAGES ************************
1000    TYPE *,'FILENAMES MUST BE DIFFERENT !'
        GOTO 100
1100    TYPE *,'ERROR OPENING/CLOSING FILE'
        GOTO 2000
1200    TYPE *,'ERROR READING/WRITING FILE'
2000    TYPE *,'END OF RUN'
        CALL EXIT
C       ************ FORMAT STATEMENTS ********************
10      FORMAT(' ENTER FILE ',A6,' NAME : ',$)
20      FORMAT(A:13A)
30      FORMAT(' REVERSE X - I.E. REVERSE ORDER OF SCANS
        1 (CR=Y) ? ',$)
40      FORMAT(' REVERSE Y - I.E. CHANGE ORDER OF DATA IN
        1 SCANS ? ',$)
50      FORMAT(' ROTATE THROUGH 90 DEGREES ? ',$)
60      FORMAT(' ROTATE CLOCKWISE (CR=Y) ? ',$)
        END
```

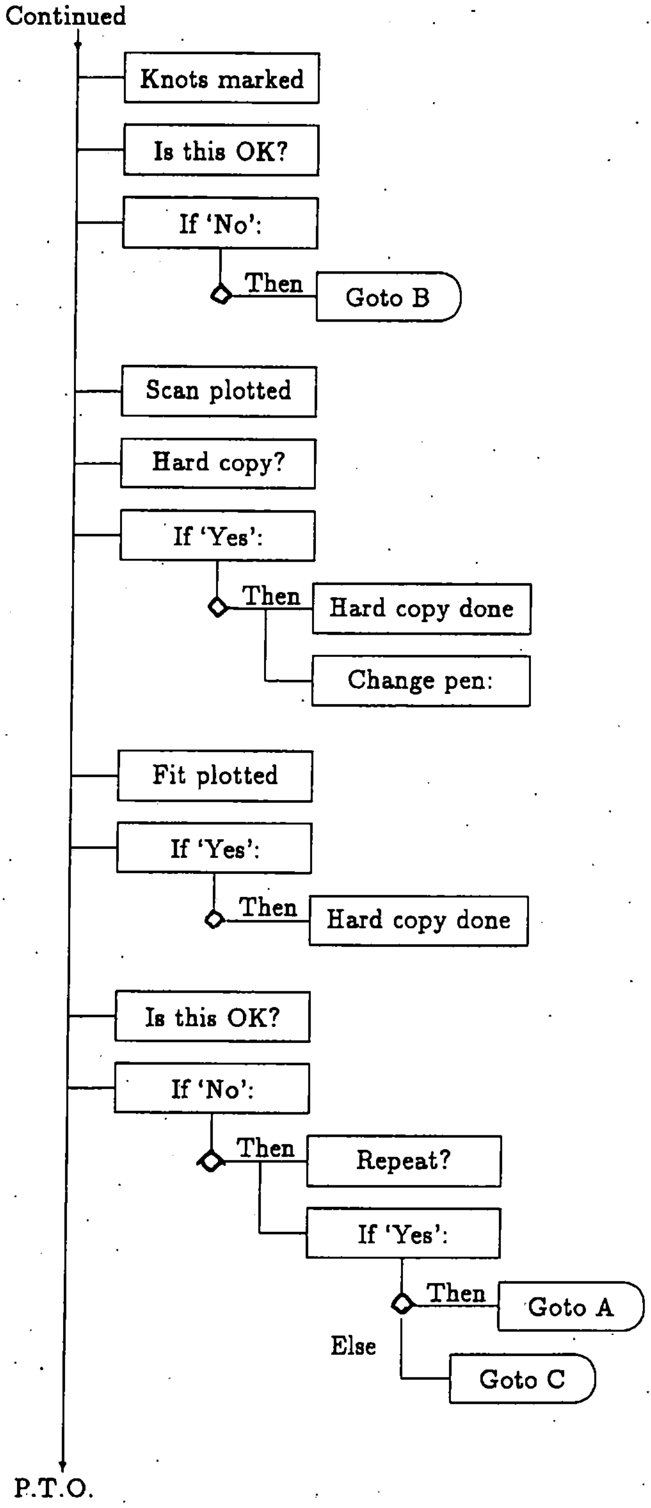
B.1.15 LKBRG

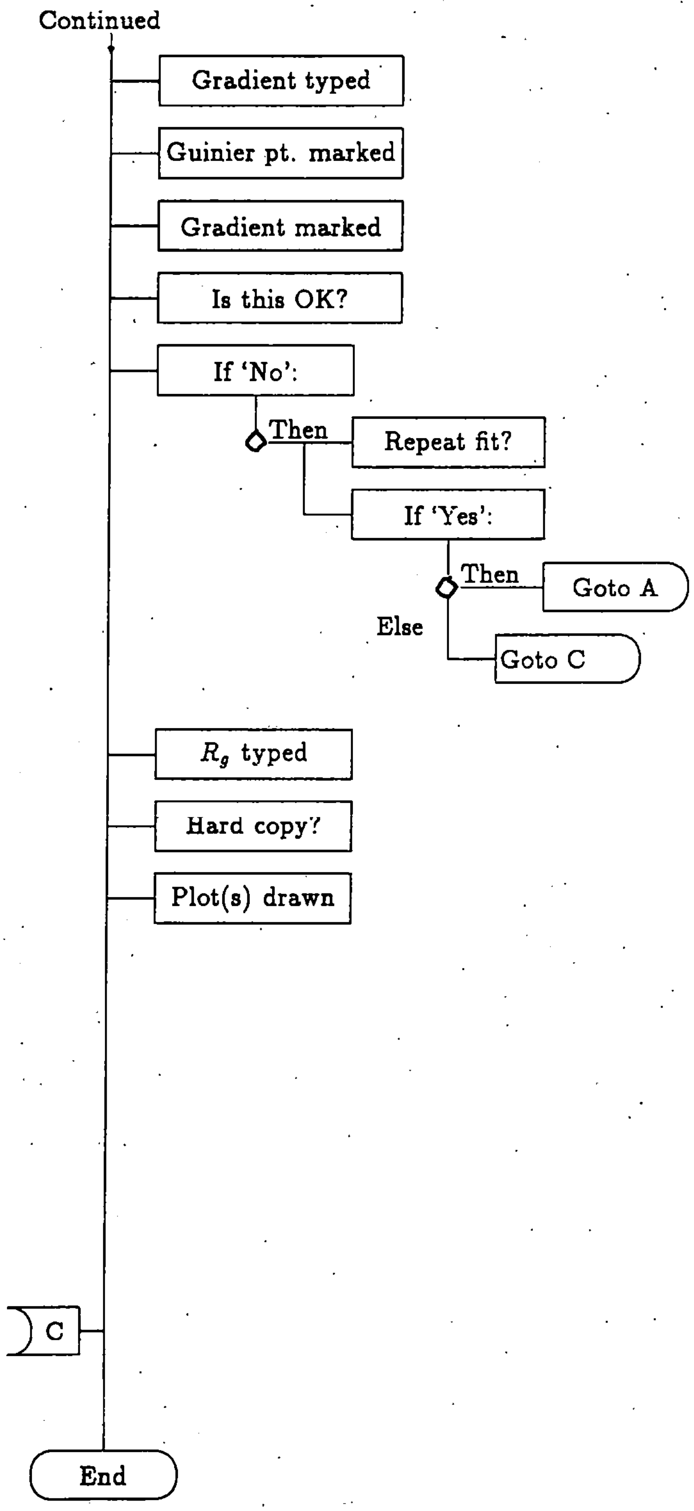
The program LKBRG finds the R_g value from a spherically averaged data set.

This routine calls the subroutines BININ, BITGRP, BITPLT, CHRPLT, GINT, GREAL, GRPLT, OLDNAM, SPLIN and TKPLT, the NAG library routine E02BAF, and the graphics macros DRAWX, MOVEX and P12M.



P.T.O.


Continued
Knots marked
Is this OK?
If 'No':
Then
Goto B
Scan plotted
Hard copy?
If 'Yes':
Then
Hard copy done
Change pen:
Fit plotted
If 'Yes':
Then
Hard copy done
Is this OK?
If 'No':
Then
Repeat?
If 'Yes':
Then
Goto A
Else
Goto C
P.T.O.


Continued
Gradient typed
Guinier pt. marked
Gradient marked
Is this OK?
If 'No':
Then
Repeat fit?
If 'Yes':
Then
Goto A
Else
Goto C
R_g typed
Hard copy?
Plot(s) drawn
C
End

```
C     LKBRG.FOR - CALCULATES THE GRADIENT OF THE GUINIER PLOT
C     AND THE RG VALUE, BY FITTING A SPLINE TO A(X) OR LOG(A)
C     DATA.
C
      LOGICAL*1 FNAM(15),TEMP,TYPE,XSTR(6),YSTR(4)
      INTEGER NA,A(501),NY,NY1,SCN,AMAX,AMIN,XSTEP,YSTEP
      1,A0,X0,Y0,XDIV,YDIV,XT,YT,NX,CH1,CH2,CHT,OLDX,OLDY
      2,N
      REAL XSCAL,YSCAL,RG,CTMP,B,XMAX,YMAX
      INTEGER NCAPM1,NKMAX
      DOUBLE PRECISION XG,YG,DY
C
C     VARIABLES FOR E02BAF :
      INTEGER M,NCAP7,IFAIL
      DOUBLE PRECISION C(17),K(17),W(200),WORK1(200),WORK2(4,17)
      1,X(200),Y(200),SS
C
      DATA FNAM/15*0/                 ! FILENAME
      DATA TYPE/'L'/                  ! DEFAULT - LOG INTENSITY
      DATA OLDX,OLDY/0,0/             ! PLOTTER COORDINATES
      DATA NA,NX/501,200/             ! INTENSITY ARRAY SIZE
      DATA NKMAX/10/                  ! MAX. NO. OF INT. KNOTS
      DATA IFAIL/0/                   ! ERROR FLAG
      DATA XG/6.99301D-2/             ! GUINIER REGION
C
C     READ IN HEADER AND (FIRST) SCAN :
      TYPE 10                         ! ENTER FILENAME
      CALL OLDNAM(FNAM,98)
      TYPE 20                         ! LOG INTENSITY ?
      ACCEPT 30,TEMP
      IF (TEMP.EQ.'N') TYPE='A'
      OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
      1,RECORDSIZE=128)
      READ(98'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,Y0
      IF (NY.GE.NA) GOTO 1100
```

```
        TYPE 40,XSCAL                      ! XSCAL O.K. ?
        ACCEPT 30,TEMP
        IF (TEMP.NE.'N') GOTO 105
        TYPE 50                            ! ENTER XSCAL
        CALL GREAL(XSCAL)
105     IF (XSCAL.EQ.0.0) GOTO 1300
C       SELECT GUINIER REGION :
        TYPE 520                           ! GUINIER REGION ?
        CALL GREAL(CTMP)
        IF (CTMP.NE.0.) XG=1/DBLE(CTMP)
        XG=XG/DBLE(XSCAL)                  ! GUINIER REGION IN CH. NOS.
        TYPE *,'GUINIER POINT IS ',XG
        TYPE 55                            ! CR WHEN TEK. SCREEN READY
        ACCEPT 30,TEMP
100     CALL BININ(1,NY,NA,A,98)
        CLOSE(UNIT=98)
C
C       CHOOSE FIRST,LAST CHANNEL AND VERIFY :
        CALL TKPLT(NY,NA,A,AMAX,AMIN,XDIV,YDIV)
300     TYPE 510,'FIRST',NY                ! ENTER FIRST CHANNEL
        CALL GINT(CH1)
        IF (CH1.LT.1.OR.CH1.GT.NY) GOTO 1400
310     TYPE 510,' LAST',NY                ! ENTER LAST CHANNEL
        CALL GINT(CH2)
        IF (CH2.LT.1.OR.CH2.GT.NY) GOTO 1410
        IF (CH2-CH1+1.GT.NX) GOTO 1500
        IF (SNGL(XG).LT.FLOAT(CH1).OR.XG.GT.FLOAT(CH2)) GOTO 1600
        XT=30+INT(970.*FLOAT(CH1)/FLOAT(NY))
        CALL MOVEX(XT,0)
        CALL DRAWX(XT,750)
        XT=30+INT(970.*FLOAT(CH2)/FLOAT(NY))
        CALL MOVEX(XT,0)
        CALL DRAWX(XT,750)
        TYPE 80                   ! O.K. ?
        ACCEPT 30,TEMP
        IF (TEMP.EQ.'N') GOTO 300
```

```
C
C     SET UP NEW ARRAY A :
      NY1=CH2-CH1+1
      XG=XG-DBLE(FLOAT(CH1))+1D0
      AMAX=A(CH1)
        DO 320 I=1,NY1
        A(I)=A(I+CH1-1)
        IF (A(I).GT.AMAX) AMAX=A(I)
320     CONTINUE
C
C     CHOOSE KNOTS :
110   CALL TKPLT(NY1,NA,A,AMAX,AMIN,XDIV,YDIV)
      TYPE 60,NKMAX                         ! HOW MANY KNOTS ?
120   CALL GINT(NCAPM1)
      IF (NCAPM1.GT.NKMAX.OR.NCAPM1.LT.0) GOTO 1000
      IF (NCAPM1.EQ.0) GOTO 135
        DO 130 I=1,NCAPM1
        K(I+4)=DBLE(FLOAT(NY1)*FLOAT(I)/FLOAT(NCAPM1+1))
        XT=30+INT(970.*SNGL(K(I+4))/FLOAT(NY1))
        CALL MOVEX(XT,0)
        CALL DRAWX(XT,750)
130     CONTINUE
135   TYPE 80                               ! O.K. ?
      ACCEPT 30,TEMP
      IF (TEMP.EQ.'N') GOTO 110
C
C     SET UP ARGUMENTS FOR E02BAF :
      NCAP7=NCAPM1+8
      M=NY1
        DO 150 I=1,NY1
        X(I)=DBLE(FLOAT(I))
        Y(I)=DBLE(FLOAT(A(I)))
        W(I)=1D0
150     CONTINUE
C
C     ASSESS FIT QUALITY :
        CALL TKPLT(NY1,NA,A,AMAX,AMIN,XDIV,YDIV)
        TYPE 530                        ! HARD COPY ?
        ACCEPT 30,TEMP
        IF (TEMP.NE.'N') CALL GRPLT(FNAM,NY1,NA,A,AMAX,AMIN)
        IF (TEMP.NE.'N') TYPE 550       ! CHANGE PEN
        CALL E02BAF(M,NCAP7,X,Y,W,K,WORK1,WORK2,C,SS,IFAIL)
        IF (IFAIL.NE.0) GOTO 1200
        SS=SS/DBLE(FLOAT(M))
        TYPE *,'MEAN SQUARE RESIDUAL = ',SS
        DO 200 I=1,NY1
        CALL SPLIN(NCAP7,C,K,X(I),Y(I),DY)
        A(I)=INT(SNGL(Y(I)))
200     CONTINUE
        CALL BITPLT(NY1,NA,A,AMAX,AMIN,1,NY1)
        IF (TEMP.NE.'N') ACCEPT 30
        IF (TEMP.NE.'N') CALL BITGRP(NY1,NA,A,AMAX,AMIN,1,NY1)
        TYPE 80                         ! O.K. ?
        ACCEPT 30,TEMP
        IF (TEMP.NE.'N') GOTO 210
        TYPE 90                         ! REPEAT FIT ?
        ACCEPT 30,TEMP
        IF (TEMP.EQ.'N') GOTO 2000
        XG=XG+DBLE(FLOAT(CH1-1))
        OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128)
        GOTO 100
C
C       FIND GRADIENT AT GUINIER REGION AND VERIFY :
210     CALL SPLIN(NCAP7,C,K,XG,YG,DY)
        TYPE *,'GRADIENT = ',DY
        XT=30+INT(970.*SNGL(XG)/FLOAT(NY1))
        CALL MOVEX(XT,0)
        CALL DRAWX(XT,750)
        YT=30+INT(690.*SNGL(YG)/FLOAT(AMAX))
        CALL MOVEX(0,YT)
        CALL DRAWX(1000,YT)
```

```
C
        CHT=1
        IF (((1D0-XG)*DY)+YG.GT.DBLE(FLOAT(AMAX)))
        1   CHT=INT(SNGL(XG+((DBLE(FLOAT(AMAX))-YG)/DY)))
        XT=30+INT(970.*FLOAT(CHT)/FLOAT(NY1))
        YT=30+INT(690.*SNGL(YG-(DY*(XG-DBLE(FLOAT(CHT)))))
        1 /FLOAT(AMAX))
        CALL MOVEX(XT,YT)
        CHT=NY1
        IF ((XG-DBLE(FLOAT(CHT)))*DY.GT.YG) CHT=INT(SNGL(XG-(YG/DY)))
        XT=30+INT(970.*FLOAT(CHT)/FLOAT(NY1))
        YT=30+INT(690.*SNGL(YG-(DY*(XG-DBLE(FLOAT(CHT)))))
        1 /FLOAT(AMAX))
        CALL DRAWX(XT,YT)
C
        TYPE 80                             ! O.K. ?
        ACCEPT 30,TEMP
        IF (TEMP.NE.'N') GOTO 220
        TYPE 90                             ! REPEAT FIT ?
        ACCEPT 30,TEMP
        IF (TEMP.EQ.'N') GOTO 2000
        OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128)
        GOTO 100
220     XG=XG+DBLE(FLOAT(CH1-1))
        DY=DY/(XG*2D0)
        IF (DY.LT.0.) DY=-DY
        IF (TYPE.EQ.'A') DY=1000*DY/YG
        RG=0.416*SNGL((4.342945D-1*DY/1D3)**0.5)/XSCAL
        TYPE 95,RG
C
C       PLOT GUINIER SHOWING GRADIENT :
        CALL TEKWIP
C       REPLACE ARRAY A WITH ORIGINAL DATA :
        OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128,READONLY)
```

```
      CALL BININ(1,NY,NA,A,98)
      CLOSE(UNIT=98)
        DO 330 I=1,NY1
        A(I)=A(I+CH1-1)
330     CONTINUE
C     CONVERT TO LOG(A), AND FIND AMIN :
        DO 400 I=1,NY1
        IF (TYPE.EQ.'L') GOTO 410
        IF (A(I).GT.0) A(I)=INT(1000.*ALOG(FLOAT(A(I))))
        IF (I.EQ.1) AMAX=A(I)
        IF (A(I).GT.AMAX) AMAX=A(I)
410     IF (I.EQ.1) AMIN=A(I)
        IF (A(I).LT.AMIN) AMIN=A(I)
400     CONTINUE
C     DRAW AXES :
      TYPE 540                           ! HARD COPY ?
      ACCEPT 30,TEMP
      CALL MOVEX(30,30)
      CALL DRAWX(30,750)
      CALL MOVEX(30,30)
      CALL DRAWX(1000,30)
      IF (TEMP.EQ.'N') GOTO 450
      CALL P12M(250,250,OLDX,OLDY,.FALSE.)
      CALL P12M(250,2700,OLDX,OLDY,.TRUE.)
      CALL P12M(250,250,OLDX,OLDY,.FALSE.)
      CALL P12M(3700,250,OLDX,OLDY,.TRUE.)
      XMAX=(FLOAT(CH2)*XSCAL)**2
      CALL RSIZE(XMAX,B,N)
        DO 460 I=1,INT(B)
        XT=250+INT(FLOAT(I)*(10.**N)*3450./XMAX)
        CALL P12M(XT,250,OLDX,OLDY,.FALSE.)
        CALL P12M(XT,230,OLDX,OLDY,.TRUE.)
        ENCODE(6,560,XSTR) (FLOAT(I)*(10.**N))
        CALL CHRPLT(XT-100,100,OLDX,OLDY,6,XSTR)
460     CONTINUE
      YMAX=FLOAT(AMAX-AMIN)*4.34294E-4
      CALL RSIZE(YMAX,B,N)
        DO 470 I=1,INT(B)+1
        YT=250+INT(FLOAT(I-1)*(10.**N)*2450./YMAX)
        CALL P12M(250,YT,OLDX,OLDY,.FALSE.)
        CALL P12M(230,YT,OLDX,OLDY,.TRUE.)
        ENCODE(4,570,YSTR) (4.34294E-4*FLOAT(AMIN))
      1   +(FLOAT(I-1)*(10.**N))
        CALL CHRPLT(0,YT,OLDX,OLDY,4,YSTR)
470     CONTINUE
C     PLOT GUINIER :
450   XT=30+INT(FLOAT(CH1)**2*970./FLOAT(CH2)**2)
      YT=30+INT(FLOAT(A(1)-AMIN)*690./(AMAX-AMIN))
      CALL MOVEX(XT,YT)
      IF (TEMP.EQ.'N') GOTO 425
      XT=250+INT(FLOAT(XT-30)*345./97.)
      YT=250+INT(FLOAT(YT-30)*245./72.)
      CALL P12M(XT,YT,OLDX,OLDY,.FALSE.)
425     DO 420 I=2,NY1
        XT=30+INT(FLOAT(CH1+I-1)**2*970./FLOAT(CH2)**2)
        YT=30+INT(FLOAT(A(I)-AMIN)*690./(AMAX-AMIN))
        CALL DRAWX(XT,YT)
        IF (TEMP.EQ.'N') GOTO 420
        XT=250+INT(FLOAT(XT-30)*345./97.)
        YT=250+INT(FLOAT(YT-30)*245./72.)
        CALL P12M(XT,YT,OLDX,OLDY,.TRUE.)
420     CONTINUE
C     DRAW TANGENT :
      XG=XG*XG
      IF (TYPE.EQ.'A') YG=1D3*DBLE(ALOG(SNGL(YG)))
      YG=YG-DBLE(FLOAT(AMIN))
      XT=30+INT(970.*SNGL(XG)/FLOAT(CH2)**2)
      YT=30+INT(690.*SNGL(YG)/FLOAT(AMAX-AMIN))
      CALL MOVEX(XT,0)
      CALL DRAWX(XT,750)
      CALL MOVEX(0,YT)
      CALL DRAWX(1000,YT)
```

```
        IF (TEMP.EQ.'N') GOTO 430
        XT=250+INT(FLOAT(XT-30)*345./97.)
        YT=250+INT(FLOAT(YT-30)*245./72.)
        CALL P12M(XT,0,OLDX,OLDY,.FALSE.)
        CALL P12M(XT,2700,OLDX,OLDY,.TRUE.)
        CALL P12M(0,YT,OLDX,OLDY,.FALSE.)
        CALL P12M(3700,YT,OLDX,OLDY,.TRUE.)
430     CHT=1
        IF (((XG-1D0)*DY)+YG.GT.DBLE(FLOAT(AMAX-AMIN)))
        1   CHT=INT(SNGL(XG+((YG-DBLE(FLOAT(AMAX-AMIN)))/DY)))
        XT=30+INT(970.*FLOAT(CHT)/FLOAT(CH2)**2)
        YT=30+INT(690.*SNGL(YG+DY*(XG-DBLE(FLOAT(CHT))))
        1/FLOAT(AMAX-AMIN))
        CALL MOVEX(XT,YT)
        IF (TEMP.EQ.'N') GOTO 440
        XT=250+INT(FLOAT(XT-30)*345./97.)
        YT=250+INT(FLOAT(YT-30)*245./72.)
        CALL P12M(XT,YT,OLDX,OLDY,.FALSE.)
440     CHT=CH2**2
        IF ((DBLE(FLOAT(CHT))-XG)*DY.GT.YG) CHT=INT(SNGL(XG+(YG/DY)))
        XT=30+INT(970.*FLOAT(CHT)/FLOAT(CH2)**2)
        YT=30+INT(690.*SNGL(YG+DY*(XG-DBLE(FLOAT(CHT))))
        1/FLOAT(AMAX-AMIN))
        CALL DRAWX(XT,YT)
        IF (TEMP.EQ.'N') GOTO 2000
        XT=250+INT(FLOAT(XT-30)*345./97.)
        YT=250+INT(FLOAT(YT-30)*245./72.)
        CALL P12M(XT,YT,OLDX,OLDY,.TRUE.)
        CALL CHRPLT(3200,2450,OLDX,OLDY,15,FNAM)
        ENCODE(4,570,YSTR) RG
        FNAM(1)=82                  ! R
        FNAM(2)=103                 ! SMALL G
        FNAM(3)=32                  ! SPACE
        FNAM(4)=61                  ! =
        FNAM(5)=32                  ! SPACE
        FNAM(6)=YSTR(1)
```

```
      FNAM(7)=YSTR(2)
      FNAM(8)=YSTR(3)
      FNAM(9)=YSTR(4)
      FNAM(10)=32             ! SPACE
      FNAM(11)=110            ! SMALL N
      FNAM(12)=109            ! SMALL M
      CALL CHRPLT(3200,2200,OLDX,OLDY,12,FNAM)
C
C     END OF RUN :
      GOTO 2000
C
C     ERROR CONDITIONS :
1000  TYPE 70                         ! TOO MANY - RE-ENTER
      GOTO 120
1100  TYPE *,'ARRAY OVERFLOW - EXITING ROUTINE'
      GOTO 2000
1200  TYPE *,'ERROR IN NAG ROUTINE E02BAF - EXITING ROUTINE'
      GOTO 2000
1300  TYPE 500                        ! RE-ENTER XSCAL
      CALL GREAL(XSCAL)
      GOTO 105
1400  TYPE *,'CHANNEL NUMBER OUT OF RANGE  - RE-ENTER'
      GOTO 300
1410  TYPE *,'CHANNEL NUMBER OUT OF RANGE - RE-ENTER'
      GOTO 310
1500  TYPE *,'RANGE TOO LARGE FOR ARRAYS - PLEASE RE-ENTER'
      GOTO 300
1600  TYPE *,'GUINIER REGION NOT INCLUDED - RE-ENTER CHANNELS'
      TYPE *,'XG = ',XG
      GOTO 300
C
2000  CALL TEKOUT(31)
      TYPE *,'END OF RUN'
      CALL EXIT
C
C     FORMAT STATEMENTS :
```

```
10        FORMAT(' ENTER FILENAME : ',$)
20        FORMAT(' IS THIS LOG INTENSITY (CR=Y) ? ',$)
30        FORMAT(A)
40        FORMAT(' POINT SEPARATION = ',F10.8,' RECIP. NM :',/
          1,'IS THIS O.K. (CR=Y) ? ',$)
50        FORMAT(' ENTER NEW XSCAL : ',$)
55        FORMAT(' ENTER CR WHEN TEK. SCREEN READY : ',$)
60        FORMAT(' HOW MANY INTERNAL KNOTS (0-',I2,') ? ',$)
70        FORMAT(' INVALID NUMBER - PLEASE RE-ENTER (0-20) : ',$)
80        FORMAT(' IS THIS O.K. (CR=Y) ? ',$)
90        FORMAT(' REPEAT FIT ? ',$)
95        FORMAT(' RADIUS OF GYRATION IS ',F6.3,' NM')
500       FORMAT(' XSCAL IS ZERO - PLEASE RE-ENTER : ',$)
510       FORMAT(' ENTER ',A5,' CHANNEL FOR FIT (MAX.=',I3,') : ',$)
520       FORMAT(' ENTER GUINIER REGION (NM) - (CR FOR 14.3) : ',$)
530       FORMAT(' HARD COPY (CR=Y) ? ',$)
540       FORMAT(' HARD COPY OF GUINIER (CR=Y) ? ',$)
550       FORMAT(' CHANGE PEN COLOUR ON PLOTTER AND ENTER CR : ')
560       FORMAT(1PE6.0)
570       FORMAT(F4.2)
          END
```

B.1.16 LKBRNG

The program LKBRNG plots the intensity of a ring about the centre of the pattern.

This routine calls the subroutines GINT, GRPLT, OLDNAM, RING and TKPLT.

```
C       LKBRNG.FOR
C       PLOTS INTENSITY IN A RING ABOUT THE PATTERN CENTRE.
C
        LOGICAL*1 FNAM(15),TEMP
        INTEGER B(360),NB,BMAX,BMIN,NY,SCN,AMAX,AMIN,XSTEP,YSTEP
        1,A0,X0,Y0,R,XDIV,YDIV
        REAL XSCAL,YSCAL
        DATA FNAM/15*0/
        DATA NB/360/
C
        TYPE 10                             ! ENTER FILENAME
        CALL OLDNAM(FNAM,98)
        OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128)
        READ(98'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0
        1,XSCAL,YSCAL,X0,Y0
        CLOSE(UNIT=98)
        TYPE 20                             ! ENTER RADIUS (CHANNELS)
        CALL GINT(R)
        CALL RING(NB,B,BMAX,BMIN,R,FNAM,NY,SCN,X0,Y0)
        TYPE *,'BMAX = ',BMAX
        IF (BMAX.LT.10) BMAX=10
        CALL TKPLT(360,NB,B,BMAX,BMIN,XDIV,YDIV)
        TYPE 30                             ! HARD COPY ?
        ACCEPT 40,TEMP
        IF (TEMP.NE.'N') CALL GRPLT(FNAM,360,NB,B,BMAX,BMIN)
        TYPE 50                             ! SUBTRACT HALVES ?
        ACCEPT 40,TEMP
        IF (TEMP.EQ.'N') GOTO 2000
        BMIN=BMAX
        BMAX=0
          DO 100 I=1,180
          B(I)=B(I)-B(360-I+1)
          IF (BMIN.GT.B(I)) BMIN=B(I)
          IF (BMAX.LT.B(I)) BMAX=B(I)
100       CONTINUE
C
          DO 110 I=1,180
          B(I)=B(I)-BMIN
110       CONTINUE
        BMAX=BMAX-BMIN
        BMIN=0
        CALL TKPLT(180,NB,B,BMAX,BMIN,XDIV,YDIV)
        TYPE 30                           ! HARD COPY ?
        ACCEPT 40,TEMP
        IF (TEMP.NE.'N') CALL GRPLT(FNAM,180,NB,B,BMAX,BMIN)
2000    TYPE *,'END OF RUN'
        CALL EXIT
10      FORMAT(' ENTER FILENAME : ',$)
20      FORMAT(' ENTER RADIUS (CHANNELS) : ',$)
30      FORMAT(' HARD COPY (Y/N) ? ',$)
40      FORMAT(A)
50      FORMAT(' PLOT DIFFERENCE BETWEEN HALVES (Y/N) ? ',$)
        END
```

B.1.17 LKBROT

The program LKBROT rotates a data set through 90° about the origin.

This routine calls the subroutines NEWNAM and OLDNAM.

```
C     ROTATES A DATA SET THROUGH 90 DEGREES, CLOCKWISE OR
C     ANTICLOCKWISE
      LOGICAL*1 FNAM1(15),FNAM2(15),TEMP,REV
      INTEGER A(256,32),NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,X0,Y0
      1,X01,Y01,N,B(256),C
      REAL XSCAL,YSCAL
      DATA FNAM1,FNAM2/15*0,15*0/
      REV=.TRUE.
      TYPE 10,' INPUT'                ! ENTER FILENAME
      CALL OLDNAM(FNAM1,98)
      TYPE 10,'OUTPUT'                ! ENTER FILENAME
      CALL NEWNAM(FNAM2,99)
      TYPE 30                         ! ROTATE CLOCKWISE ?
      ACCEPT 20,TEMP
      IF (TEMP.NE.'N') REV=.FALSE.
      OPEN(UNIT=98,NAME=FNAM1,TYPE='OLD',ACCESS='DIRECT'
      1,RECORDSIZE=128,READONLY)
      OPEN(UNIT=99,NAME=FNAM2,TYPE='NEW',ACCESS='DIRECT'
      1,RECORDSIZE=128)
      READ(98'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,Y0
      IF (NY.GT.256.OR.SCN.GT.256) GOTO 1000
      X01=NY-Y0+1
      Y01=X0
      IF (REV.EQ..TRUE.) X01=Y0
      IF (REV.EQ..TRUE.) Y01=SCN-X0+1
      WRITE(99'1) SCN,NY,AMAX,AMIN,YSTEP,XSTEP,A0
      1,YSCAL,XSCAL,X01,Y01
      DO 200 N=1,8
        DO 100 I=1,SCN
        READ (98'I+1)(B(J),J=1,256)
        IF (REV.EQ..TRUE.) C=((N-1)*32)
        IF (REV.EQ..FALSE.) C=((8-N)*32)+NY-256
```

```
           DO 120 J=1,32
           IF (J+C.LT.1) GOTO 120
           IF (REV.EQ..TRUE.) A(I,J)=B(J+C)
           IF (REV.EQ..FALSE.) A(I,32-J+1)=B(J+C)
120        CONTINUE
100      CONTINUE
C
         DO 110 I=1,32
         C=I+(32*(N-1))+1
         IF (C.GT.NY+1) GOTO 110
         IF (REV.EQ..FALSE.) WRITE(99'C) (A(J,I),J=1,SCN)
         IF (REV.EQ..TRUE.) WRITE(99'C) (A(J,I),J=SCN,1,-1)
110      CONTINUE
200    CONTINUE
       GOTO 2000
1000   TYPE *,'ARRAY SIZE EXCEEDED'
2000   CLOSE(UNIT=98)
       CLOSE(UNIT=99)
       TYPE *,'END OR RUN'
       CALL EXIT
10     FORMAT(' ENTER ',A6,' FILENAME : ',$)
20     FORMAT(A:13A)
30     FORMAT(' ROTATE CLOCKWISE (CR=Y) ? ',$)
       END
```

B.1.18 LKBSM

The program LKBSM smooths a data set by averaging or adding points together.

This routine calls the subroutines CLRSCR, GETNAM, NEWNAM and OLDNAM.

```
C       LKBSM
C       SMOOTHS A .BIN FILE
        LOGICAL*1 FNAM(15),FNAM1(15),TEMP,ADD
        INTEGER A(1000),B(1000),NY,SCN,AMAX,AMIN,XSTEP,YSTEP,AO
        1,XO,YO,NS,N(2),NT,NY1,SCN1,NS1
        REAL XSCAL,YSCAL
        DATA ADD/.FALSE./
        DATA N/2*1/
        DATA FNAM/3*0,':',6*'0','.',4*0/
        CALL CLRSCR
        TYPE 10,' IN'                   ! INPUT FILENAME
        CALL OLDNAM(FNAM,98)
110     TYPE 10,'OUT'                   ! OUTPUT FILENAME
        CALL NEWNAM(FNAM1,98)
        TYPE 30,'Y'                     ! SMOOTH Y ?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') GOTO 120
        TYPE 40
        ACCEPT *,N(2)
        IF (N(2).EQ.0) N(2)=1
120     TYPE 30,'X'                     ! SMOOTH X ?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') GOTO 130
        TYPE 40
        ACCEPT *,N(1)
        IF (N(1).EQ.0) N(1)=1
130     IF (N(1).EQ.1.AND.N(2).EQ.1) GOTO 2000
        OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128,READONLY,ERR=1100)
        OPEN(UNIT=99,NAME=FNAM1,TYPE='NEW',ACCESS='DIRECT'
        1,RECORDSIZE=128,ERR=1100)
        READ(98'1,ERR=1200) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,AO
```

```
      1,XSCAL,YSCAL,X0,Y0
      TYPE *,'AMAX = ',AMAX
      TYPE 50                               !AVERAGE POINTS?
      ACCEPT 20,TEMP
      IF (TEMP.EQ.'N') ADD=.TRUE.
C
C     CALCULATING HEADER
      IF (ADD.EQ..TRUE.) AMAX=AMAX*N(1)*N(2)
      SCN1=INT(FLOAT(SCN)/FLOAT(N(1)))
      NY1=INT(FLOAT(NY)/FLOAT(N(2)))
      XSTEP=XSTEP*N(1)
      YSTEP=YSTEP+N(2)
      XSCAL=XSCAL*FLOAT(N(1))
      YSCAL=YSCAL*FLOAT(N(2))
      X0=INT(FLOAT(X0)/FLOAT(N(1)))
      Y0=INT(FLOAT(Y0)/FLOAT(N(2)))
      WRITE(99'1,ERR=1200) NY1,SCN1,AMAX,AMIN,XSTEP,YSTEP,A0
      1,XSCAL,YSCAL,X0,Y0
C     ********************************************************
       DO 300 NS1=1,SCN1
        DO 200 I=1,NY1
        A(I)=0
 200    CONTINUE
C
        DO 210 I=1,N(1)
        NS=((NS1-1)*N(1))+I
        CALL BININ(NS,NY,1000,B,98)
         DO 230 J=1,NY1
          DO 240 K=1,N(2)
          NT=((J-1)*N(2))+K
          A(J)=A(J)+B(NT)
240       CONTINUE
230      CONTINUE
210     CONTINUE
       IF (ADD.EQ..TRUE.) GOTO 250
        DO 260 J=1,NY1
```

```
            A(J)=INT(0.5+FLOAT(A(J))/FLOAT(N(1)*N(2)))
260         CONTINUE
C
250       CALL BINOUT(NS1,NY1,1000,A,99)
300       CONTINUE
        CLOSE(UNIT=98,ERR=1100)
        CLOSE(UNIT=99,ERR=1100)
        GOTO 2000
C       ***********************************************************
1100    TYPE *,'ERROR OPENING/CLOSING FILE !'
        GOTO 2000
1200    TYPE *,'ERROR READING/WRITING FILE !'
        CLOSE(UNIT=98)
        CLOSE(UNIT=99)
        GOTO 2000
C       ***********************************************************
2000    TYPE *,'END OF RUN'
        CALL EXIT
10      FORMAT(' ENTER ',A3,'PUT FILENAME : ',$)
20      FORMAT(A)
30      FORMAT(' SMOOTH IN ',A,'-DIRECTION (CR=Y) ? ',$)
40      FORMAT(' ENTER NUMBER OF POINTS TO AVERAGE OVER : ',$)
50      FORMAT(' AVERAGE POINTS (CR=Y, N=ADD POINTS) ? ',$)
        END
```

B.1.19 MIRROR

The program MIRROR takes the mirror image of each scan in a .BIN file, and appends to it the original scan, or performs the inverse operation.

This routine calls the subroutines BININ, BINOUT, NEWNAM and OLDNAM.

```
C       MIRROR.FOR
C       ADDS A MIRROR IMAGE TO A DATA SET TO GIVE A SYMMETRIC
C       SCAN, OR SUBTRACTS HALF OF AN ALREADY-TREATED DATA SET.
C
        LOGICAL*1 FNAM(15),FNAM1(15),TEMP
        INTEGER NY,SCN,AMAX,AMIN,XSTEP,YSTEP,AO,XO,YO,NA
        1,A(750),NB,B(750),NS,NY1
        REAL XSCAL,YSCAL
        DATA FNAM,FNAM1/15*0,15*0/
        DATA NA,NB/750,750/
C
        TYPE 10                         ! ENTER INPUT FILENAME
        CALL OLDNAM(FNAM,98)
        OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128)
        READ(98'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,AO,XSCAL,YSCAL,XO,YO
        IF (NY.LE.NA) GOTO 100
        TYPE *,'ARRAY OVERFLOW'
        CLOSE(UNIT=98)
        GOTO 2000
100     TYPE 20                         ! ENTER OUTPUT FILENAME
        CALL NEWNAM(FNAM1,99)
        TYPE 30                         ! ADD MIRROR IMAGE ?
        ACCEPT 40,TEMP
        IF (TEMP.EQ.'N') GOTO 1000      ! REMOVE HALF OF DATA
        IF ((2*NY)-1.LE.NB) GOTO 200
        TYPE *,'ARRAY OVERFLOW'
        GOTO 2000
C
200     TYPE *,'ADDING MIRROR IMAGE TO DATA SET . . . .'
        OPEN(UNIT=99,NAME=FNAM1,TYPE='NEW',ACCESS='DIRECT'
        1,RECORDSIZE=128)
        NY1=(2*NY)-1
        WRITE(99'1) NY1,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,NY
          DO 230 J=1,SCN
          NS=J
          CALL BININ(NS,NY,NA,A,98)
            DO 210 I=1,NY-1
            B(I)=A(NY-I+1)
210         CONTINUE
C
            DO 220 I=1,NY
            B(NY+I-1)=A(I)
220         CONTINUE
          CALL BINOUT(NS,NY1,NB,B,99)
230       CONTINUE
        CLOSE(UNIT=98)
        CLOSE(UNIT=99)
        GOTO 2000
C
1000    TYPE 50                         ! REMOVING HALF - SURE ?
        ACCEPT 40,TEMP
        IF (TEMP.EQ.'N') GOTO 2000      ! END
        TYPE *,'O.K. - REMOVING HALF OF DATA SET . . . .'
        NY1=INT(FLOAT(NY+1)/2)
        OPEN(UNIT=99,NAME=FNAM1,TYPE='NEW',ACCESS='DIRECT'
        1,RECORDSIZE=128)
        WRITE(99'1) NY1,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,1
          DO 1200 J=1,SCN
          NS=J
          CALL BININ(NS,NY,NA,A,98)
            DO 1100 I=1,NY1
            B(I)=A(I+NY1-1)
1100        CONTINUE
          CALL BINOUT(NS,NY1,NB,B,99)
1200      CONTINUE
        CLOSE(UNIT=98)
        CLOSE(UNIT=99)
C
2000    TYPE *,'END OF RUN'
        CALL EXIT
10      FORMAT(' ENTER INPUT FILENAME : ',$)
20      FORMAT(' ENTER OUTPUT FILENAME : ',$)
30      FORMAT(' ADD MIRROR IMAGE (Y/N, CR=Y) ? ',$)
40      FORMAT(A)
50      FORMAT(' REMOVING HALF OF DATA - ARE YOU SURE (CR=Y) ? ',$)
        END
```

B.1.20 POROD

The program POROD uses $I\alpha\frac{1}{x^4}$, valid in the wings of the diffraction pattern, to estimate a constant background level for a data set.

This routine calls the subroutines BININ, BITGRP, BITPLT, CHEB1, GINT, GRPLT, OLDNAM and TKPLT, the graphics macros DRAWX, MOVEX, P12M and TEKOUT, and the NAG library routine E02ADF.

```
C       PROGRAM POROT.FOR, TO FIND A CONSTANT BACKGROUND INTENSITY
C       LEVEL BY USING THE POROT EQUATION AND FITTING A STRAIGHT
C       LINE TO A PLOT OF INTENSITY AGAINST 1/(X**4)
C
        LOGICAL*1 FNAM(15),TEMP
        INTEGER NA,A(501),NY,SCN,AMAX,AMIN,AMAX1,AMIN1,XSTEP,YSTEP,A0
        1,NS,XDIV,YDIV,AT,OLDX,OLDY,CH1,CH2
        REAL XSCAL,YSCAL,XMAX,XMIN,XT,YT,ALPHA
C
        INTEGER M,KPLUS1,NROWS,IFAIL
        DOUBLE PRECISION X(200),Y(200),W(200),WORK1(3,200),WORK2(2,2)
        1,C(2,2),S(2)
C
        DATA NA/501/
        DATA KPLUS1,NROWS/2,2/
        DATA W/200*1D0/
        DATA OLDX,OLDY/0,0/
C
C
        TYPE 10                         ! ENTER INPUT FILENAME
        CALL OLDNAM(FNAM,98)
        OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128)
        READ(98'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,XSCAL,YSCAL
        TYPE *,'NY = ',NY,'      SCN = ',SCN
        NS=1
        IF (SCN.EQ.1) GOTO 110
100     TYPE 20                         ! ENTER SCAN NUMBER
        CALL GINT(NS)
        IF (NS.GT.0.AND.NS.LE.SCN) GOTO 110
        TYPE *,'SCAN NUMBER OUT OF RANGE !'
        GOTO 100
110     CALL BININ(NS,NY,NA,A,98)
        CALL TKPLT(NY,NA,A,AMAX,AMIN,XDIV,YDIV)
        TYPE *,'XDIV = ',XDIV,' YDIV = ',YDIV
120     TYPE 30,'FIRST'                 ! ENTER FIRST CHANNEL
        CALL GINT(CH1)
        IF (CH1.GE.1.AND.CH1.LT.NY) GOTO 130
        TYPE *,'CHANNEL NUMBER OUT OF RANGE !'
        GOTO 120
130     AT=30+INT(970.*FLOAT(CH1)/FLOAT(NY))
        CALL MOVEX(AT,0)
        CALL DRAWX(AT,750)
        TYPE 60                         ! OK ?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') GOTO 120
140     TYPE 30,' LAST'                 ! ENTER LAST CHANNEL
        CALL GINT(CH2)
        IF (CH2.GT.CH1.AND.CH2.LE.NY) GOTO 150
        TYPE *,'CHANNEL NUMBER OUT OF RANGE !'
        GOTO 140
150     IF (CH2-CH1+1.LE.200) GOTO 160
        TYPE *,'RANGE TOO LARGE FOR ARRAYS - RE-ENTER'
        GOTO 120
160     AT=30+INT(970.*FLOAT(CH2)/FLOAT(NY))
        CALL MOVEX(AT,0)
        CALL DRAWX(AT,750)
        TYPE 60                         ! OK ?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') GOTO 140
        M=CH2-CH1+1
        AMAX1=A(CH1)
        AMIN1=A(CH1)
          DO 170 I=1,M
          X(I)=1D0/DBLE(FLOAT(CH2-I+1))**4
        Y(I)=DBLE(FLOAT(A(CH2-I+1)))
        A(I)=A(I+CH1-1)
        IF (AMAX1.LT.A(I)) AMAX1=A(I)
        IF (AMIN1.GT.A(I)) AMIN1=A(I)
170     CONTINUE
      XMAX=SNGL(X(M))
      XMIN=SNGL(X(1))
      CALL E02ADF(M,KPLUS1,NROWS,X,Y,W,WORK1,WORK2,C,S,IFAIL)
      TYPE *,'XMAX=',XMAX,'  XMIN=',XMIN
      ALPHA=SNGL(C(2,1)/2)-SNGL(C(2,2))*(XMAX+XMIN)/(XMAX-XMIN)
      TYPE *,'BACKGROUND LEVEL = ',ALPHA
      TYPE *,'R.M.S. RESIDUAL = ',S(2)
      TYPE 40                           ! CR FOR PLOT OF FIT
      ACCEPT 20,TEMP
      CALL TKPLT(M,NA,A,AMAX1,AMIN1,XDIV,YDIV)
      TYPE 70                           ! HARD COPY ?
      ACCEPT 20,TEMP
      IF (TEMP.NE.'N') CALL GRPLT(FNAM,M,NA,A,AMAX1,AMIN1)
      S(1)=C(2,1)
      S(2)=C(2,2)
        DO 200 I=1,M
        A(I)=0
        XT=SNGL(X(M-I+1))
        CALL CHEB1(2,S,XT,XMAX,XMIN,YT,IFAIL)
        A(I)=INT(YT)
200     CONTINUE
      CALL BITPLT(M,NA,A,AMAX1,AMIN1,1,M)
      IF (TEMP.EQ.'N') GOTO 210
      TYPE 80                           ! CHANGE PLOTTER PEN
      ACCEPT 20
      CALL BITGRP(M,NA,A,AMAX1,AMIN1,1,M)
C
210   TYPE 50                           ! CR FOR PLOT OF DATA
      ACCEPT 20,TEMP
      CALL BININ(NS,NY,NA,A,98)
      CLOSE(UNIT=98)
```

```
        CALL TKPLT(NY,NA,A,AMAX,AMIN,XDIV,YDIV)
        AT=30+INT(690.*ALPHA/FLOAT(AMAX))
        CALL MOVEX(0,AT)
        CALL DRAWX(1000,AT)
        CALL TEKOUT(31)
        TYPE 70                           ! HARD COPY ?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') GOTO 2000
        CALL GRPLT(FNAM,NY,NA,A,AMAX,AMIN)
        TYPE 80                           ! CHANGE PLOTTER PEN
        ACCEPT 20
        AT=250+INT(2200.*ALPHA/FLOAT(AMAX))
        CALL P12M(0,AT,OLDX,OLDY,.FALSE.)
        CALL P12M(3700,AT,OLDX,OLDY,.TRUE.)
C
C
2000    TYPE *,'END OF RUN'
        CALL EXIT
C
10      FORMAT(' ENTER INPUT FILE NAME : ',$)
20      FORMAT(A)
30      FORMAT(' ENTER ',A5,' CHANNEL OF RANGE : ',$)
40      FORMAT(' ENTER <CR> FOR PLOT OF FIT :',$)
50      FORMAT(' ENTER <CR> FOR PLOT OF ORIGINAL DATA :',$)
60      FORMAT(' IS THIS OK (CR=Y) ? ',$)
70      FORMAT(' HARD COPY (CR=Y) ? ',$)
80      FORMAT(' CHANGE PLOTTER PEN AND ENTER <CR> : ',$)
C
        END
```

B.1.21 UNSA

The program UNSA recreates a 2-dimensional data set from the spherical average.

This routine calls the subroutines BININ, BINOUT, GINT, NEWNAM and OLDNAM.

```
C      UNSA.FOR
C      PROGRAM TO RECREATE A FULL DATA SET FROM
C      THE SPHERICAL AVERAGE.
C
       LOGICAL*1 FNAM(15),TEMP
       INTEGER NY,SCN,AMAX,AMIN,XSTEP,YSTEP,AO,XO,YO,NS,BX,BY
       1,NA,A(1000),NB,B(500),NY1,YT
       REAL XSCAL,YSCAL,R,DR
       DATA FNAM/15*0/
       DATA NA,NB/1000,500/
C
       TYPE 10                 ! ENTER INPUT FILENAME
       CALL OLDNAM(FNAM,98)
       OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
       1,RECORDSIZE=128)
       READ(98'1) NY1,SCN,AMAX,AMIN,XSTEP,YSTEP
       1,AO,XSCAL,YSCAL,XO,YO
       CALL BININ(1,NY1,NB,B,98)
       CLOSE(UNIT=98)
       TYPE 20                 ! ENTER OUTPUT FILENAME
       CALL NEWNAM(FNAM,98)
       TYPE 30                 ! ENTER NY
       CALL GINT(NY)
       TYPE 40                 ! ENTER SCN
       CALL GINT(SCN)
       TYPE 50                 ! ENTER XO
       CALL GINT(XO)
       TYPE 60                 ! ENTER YO
       CALL GINT(YO)
       OPEN(UNIT=98,NAME=FNAM,TYPE='NEW',ACCESS='DIRECT'
       1,RECORDSIZE=128)
       WRITE(98'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP
```

```
        1,A0,XSCAL,YSCAL,X0,Y0
        AMIN-AMAX
        AMAX=0
          DO 100 NS=1,SCN
            DO 110 I=1,NY
            R=1+(FLOAT(NS-X0)**2+FLOAT(I-Y0)**2)**0.5
            IF (R.GT.FLOAT(NY1)) GOTO 120
            YT=INT(R)
            DR=R-FLOAT(YT)
            A(I)=INT(0.5+(FLOAT(B(YT))*(1-DR))+(FLOAT(B(YT+1))*DR))
            GOTO 130
120         A(I)=0
130         IF (AMAX.LT.A(I)) AMAX=A(I)
            IF (AMIN.GT.A(I)) AMIN=A(I)
110         CONTINUE
          CALL BINOUT(NS,NY,NA,A,98)
100       CONTINUE
        WRITE(98'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP
        1,A0,XSCAL,YSCAL,X0,Y0
        CLOSE(UNIT=98)
        TYPE *,'END OF RUN'
        CALL EXIT
10      FORMAT(' ENTER INPUT FILENAME : ',$)
20      FORMAT(' ENTER OUTPUT FILENAME : ',$)
30      FORMAT(' ENTER NUMBER OF CHANNELS : ',$)
40      FORMAT(' ENTER NUMBER OF LINES : ',$)
50      FORMAT(' ENTER CENTRAL LINE, X0 : ',$)
60      FORMAT(' ENTER CENTRAL CHANNEL, Y0 : ',$)
        END
```

B.2 The Subroutines

The subroutines, which may be called by more than one program, are listed here, in alphabetical order of subroutine name.

B.2.1 AVIN

The routine AVIN reads in a number of scans from a .BIN file, averages them together, and passes the result to the calling routine.

This routine calls the routine BININ.

```
      SUBROUTINE AVIN(NS,NY,SCN,NA,A,N,ROT,IUNIT)
C     READS IN SCANS NS-N TO NS+N, NORMAL OR ROTATED, AND
C     RETURNS THE AVERAGE.
C
      INTEGER NS,NY,SCN,NA,A(NA),N,IUNIT
      LOGICAL*1 ROT
      REAL B(1000)
      IF (ROT.EQ.'R') GOTO 200
C
        DO 100 I=1,NY
        B(I)=0.0
100     CONTINUE
C
        DO 110 I=NS-N,NS+N
        IF (I.GT.SCN) GOTO 110
        CALL BININ(I,NY,NA,A,IUNIT)
          DO 120 J=1,NY
          B(J)=B(J)+FLOAT(A(J))
120       CONTINUE
110     CONTINUE
C
        DO 130 I=1,NY
        A(I)=INT(B(I)/(2*FLOAT(N)+1))
130     CONTINUE
      GOTO 300
C
200     DO 210 I=1,SCN
        B(I)=0.0
        CALL BININ(I,NY,NA,A,IUNIT)
          DO 220 J=NS-N,NS+N
          B(I)=B(I)+FLOAT(A(J))
```

```
220         CONTINUE
210       CONTINUE
C
          DO 230 I=1,SCN
          A(I)=INT(B(I)/(2*FLOAT(N)+1))
230       CONTINUE
C
300     RETURN
        END
```

B.2.2 BELL

This routine rings the bell on the terminal.

```
        SUBROUTINE BELL
C       RINGS BELL
        LOGICAL*1 A,B
        A=7
        ENCODE(1,10,B) A
        TYPE 20,B
        RETURN
10      FORMAT(A)
20      FORMAT(X,A)
        END
```

B.2.3 BININ

The routine BININ reads in scan number NS into array A from a .BIN file open on unit IUNIT.

```
        SUBROUTINE BININ(NS,NY,NA,A,IUNIT)
        INTEGER NS,NY,NA,A(NA),NR,REC,IUNIT
        NR=INT(FLOAT(NY-1)/256)+1
        IF (NR*256.GT.NA) GOTO 1000
          DO 100 I=1,NR
          REC=((NS-1)*NR)+I+1
          READ(IUNIT'REC) (A(J),J=((I-1)*256)+1,I*256)
100       CONTINUE
        GOTO 2000
1000    TYPE *,'ARRAY OVERFLOW IN "BININ"'
2000    RETURN
        END
```

B.2.4 BINOUT

This routine writes out scan number NS in array A to a .BIN file open on unit IUNIT.

```
      SUBROUTINE BINOUT(NS,NY,NA,A,IUNIT)
      INTEGER NS,NY,NA,A(NA),NR,REC,IUNIT
      NR=INT(FLOAT(NY-1)/256)+1
      IF (NR*256.GT.NA) GOTO 1000
        DO 100 I=1,NR
        REC=((NS-1)*NR)+I+1
        WRITE(IUNIT'REC) (A(J),J=((I-1)*256)+1,I*256)
100     CONTINUE
      GOTO 2000
1000  TYPE *,'ARRAY OVERFLOW IN "BINOUT"'
2000  RETURN
      END
```

B.2.5 BITGRP

This routine plots out the section of the scan in array A between points XT1 and XT2, on the graph plotter.

This routine calls the routine FNC, and the graphics macro P12M.

```
      SUBROUTINE BITGRP(NY,NA,A,AMAX,AMIN,XT1,XT2)
C     PLOTS SECTION OF A BETWEEN X1 AND X2 ON SAME SCALE AS GRPLT
C     ON THE GRAPH PLOTTER.
      LOGICAL*1 PEN
      INTEGER NY,NA,A(NA),AMAX,AMIN,X1,X2,XT1,XT2,XT,AT,FNC
      1,OLDX,OLDY,XMAX,YMAX,MARG
      REAL DX,DA
      DATA XMAX,YMAX,MARG/3700,2700,250/
      DATA OLDX,OLDY/0,0/
      PEN=.FALSE.
      X1=XT1-1
      IF (X1.LT.1) X1=1
      X2=XT2+1
      IF (X2.GT.NY) X2=NY
      DX=FLOAT(XMAX-MARG)/FLOAT(NY)
      DA=FLOAT(YMAX-(2*MARG))/FLOAT(AMAX)
C
        DO 100 I=X1,X2
        IF (A(I).LE.AMAX) GOTO 110
        PEN=.FALSE.
        GOTO 100
110     XT=FNC(MARG,DX,I)
        AT=FNC(MARG,DA,A(I))
        CALL P12M(XT,AT,OLDX,OLDY,PEN)
        PEN=.TRUE.
100     CONTINUE
      RETURN
      END
```

B.2.6 BITPLT

This routine plots the section of the scan in array A between the points XT1 and XT2, on the Textronix graphics screen.

This routine calls the function FNC and the graphics macros DRAWX, MOVEX and TEKOUT.

```
      SUBROUTINE BITPLT(NY,NA,A,AMAX,AMIN,XT1,XT2)
C     PLOTS SECTION OF A BETWEEN X1 AND X2 ON SAME SCALE AS TKPLT
      INTEGER NY,NA,A(NA),AMAX,AMIN,X1,X2,XT1,XT2,XT,AT,FNC
      REAL DX,DA
      X1=XT1-1
      IF (X1.LT.1) X1=1
      X2=XT2+1
      IF (X2.GT.NY) X2=NY
      DX=970./FLOAT(NY)
      DA=690./FLOAT(AMAX)
      AT=FNC(30,DA,A(X1))
      XT=FNC(30,DX,X1)
      CALL MOVEX(XT,AT)
      DO 100 I=X1+1,X2
      XT=FNC(30,DX,I)
      AT=FNC(30,DA,A(I))
      CALL DRAWX(XT,AT)
100   CONTINUE
      CALL TEKOUT(31)
      RETURN
      END
```

B.2.7 CHEB1

This routine calculates the value of the Chebychev polynomial at X, from the polynomial coefficients.

```
      SUBROUTINE CHEB1(NA,A,X,XMAX,XMIN,Y,IERR)
C       CALCULATES Y FROM THE CHEBYCHEV POLYNOMIAL OF ORDER
C     (NA-1), COEFFICIENTS A.
C       NA MUST BE GREATER THAN 1 AND NOT MORE THAN 10.
C       X MUST BE IN THE RANGE XMIN TO XMAX.
C       IERR IS AN ERROR FLAG - IT IS ZERO IF NO ERRORS HAVE
C     BEEN DETECTED.
C       T(N) IS THE CHEBYCHEV POLYNOMIAL OF ORDER (N-1)
C
      INTEGER NA,IERR
      DOUBLE PRECISION A(NA),T(10)
      REAL X1,X,XMAX,XMIN,Y
      IERR=0
      IF (NA.LT.2.OR.NA.GT.10) GOTO 1000
      X1=((2*X)-XMAX-XMIN)/(XMAX-XMIN)
      IF (X1.LT.-1.OR.X1.GT.1) GOTO 1100
      T(1)=1D0
      T(2)=DBLE(X1)
      Y=0.5*A(1)+T(2)*A(2)
      IF (NA.LT.3) GOTO 2000
      DO 100 I=3,NA
      T(I)=2*X1*T(I-1)-T(I-2)
      Y=Y+A(I)*T(I)
 100  CONTINUE
      GOTO 2000
1000  TYPE *,'DEGREE LESS THAN 1 OR GREATER THAN 9'
      IERR=1
      GOTO 2000
1100  TYPE *,'X OUT OF RANGE'
      IERR=1
2000  RETURN
```

```
END
```

B.2.8 CHRPLT

This routine writes a string of characters onto the graph plotter at a specified position.

This routine calls the graphics macros P12M and TPOUT.

```
        SUBROUTINE CHRPLT(X,Y,OLDX,OLDY,A,F)
C       WRITES CHARACTERS IN F TO PLOTTER AT X,Y
C       OLDX,OLDY ARE VARIABLES CONTAINING PREVIOUS X,Y
C       NEW X,Y POSITION IS RETURNED IN OLDX,OLDY
C       VARIABLES X,Y ARE UNCHANGED, AS ARE A AND F(A)
C       REQUIRES LIBRARIES PLOLIB AND GRAFX TO LINK
        INTEGER X,Y,OLDX,OLDY,A
        LOGICAL*1 F(A)
        CALL P12M(X,Y,OLDX,OLDY,.FALSE.)
        CALL TPOUT(31)
        DO 1 I=1,A
        CALL TPOUT(F(I))
 1      CONTINUE
        RETURN
        END
```

B.2.9 CLRSCR

This routine clears the screen of the terminal.

```
      SUBROUTINE CLRSCR
C     CLEARS SCREEN
      LOGICAL*1 A,CLR
      A=31
      ENCODE (1,5,CLR) A
      TYPE 10,CLR
      RETURN
 5    FORMAT (A)
10    FORMAT (X,A)
      END
```

B.2.10 CONV

This routine converts characters 3–6 of line J of a .DAT header file, contained in the array H, into an integer, AM.

```
        SUBROUTINE CONV(H,J,AM)
C       CONVERTS CHARS. 3-6 OF LINE J TO INTEGER AM
        LOGICAL*1 D(4)
        INTEGER J,AM,H(21,13)
        D(1)='X'
        D(2)='X'
        D(3)='X'
        D(4)='X'
        DO 10 I=1,4
        IF (H(J,I+2).EQ.13) GOTO 20
        D(I)=H(J,I+2)
 10     CONTINUE
20      IF (D(4).NE.'X') GOTO 30
        D(4)=D(3)
        D(3)=D(2)
        D(2)=D(1)
        D(1)=48
        GOTO 20
30      DECODE(4,50,D) AM
        RETURN
50      FORMAT(I4)
        END
```

B.2.11 FILNAM

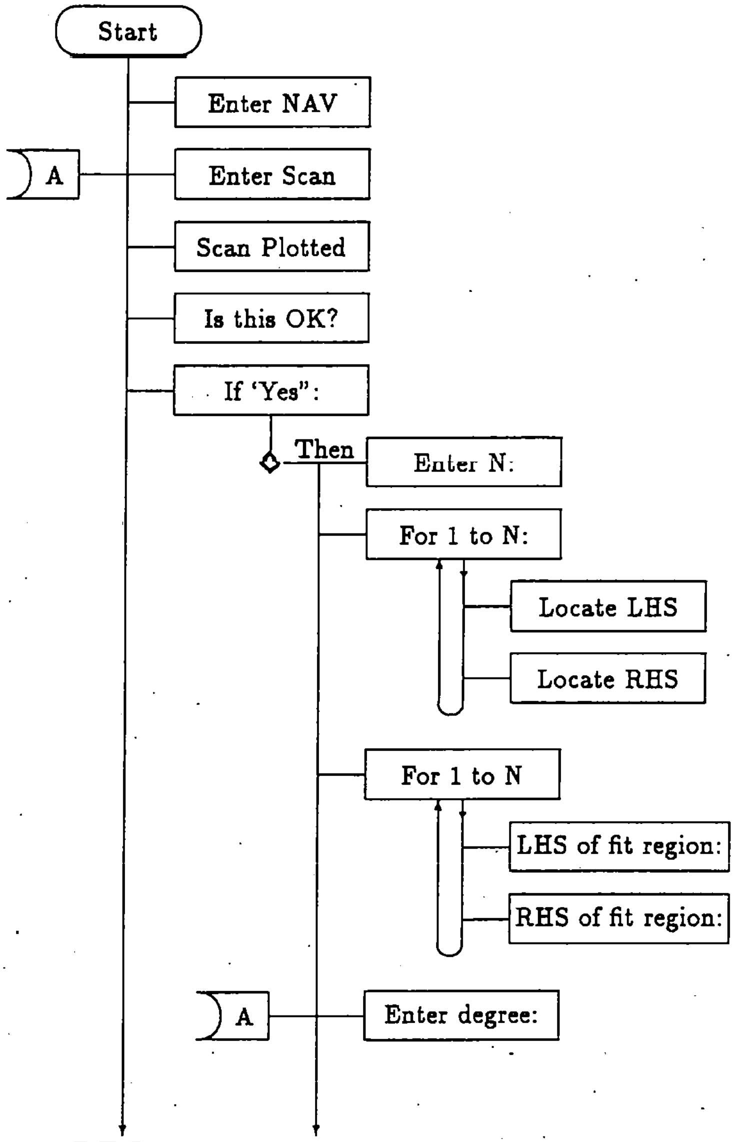
This routine replaces characters 8–10 of a file name with the digits of an integer NS, with leading zeros if necessary. It is used for the naming of .DAT files.

```
        SUBROUTINE FILNAM(FNAM,NS)
        LOGICAL*1 FNAM(15),TEMP(3)
        INTEGER NS
        ENCODE(3,10,TEMP) NS
        DO 20 I=1,3
        IF (TEMP(I).EQ.' ') TEMP(I)='0'
        FNAM(I+7)=TEMP(I)
 20     CONTINUE
        RETURN
10      FORMAT(I3)
        END
```

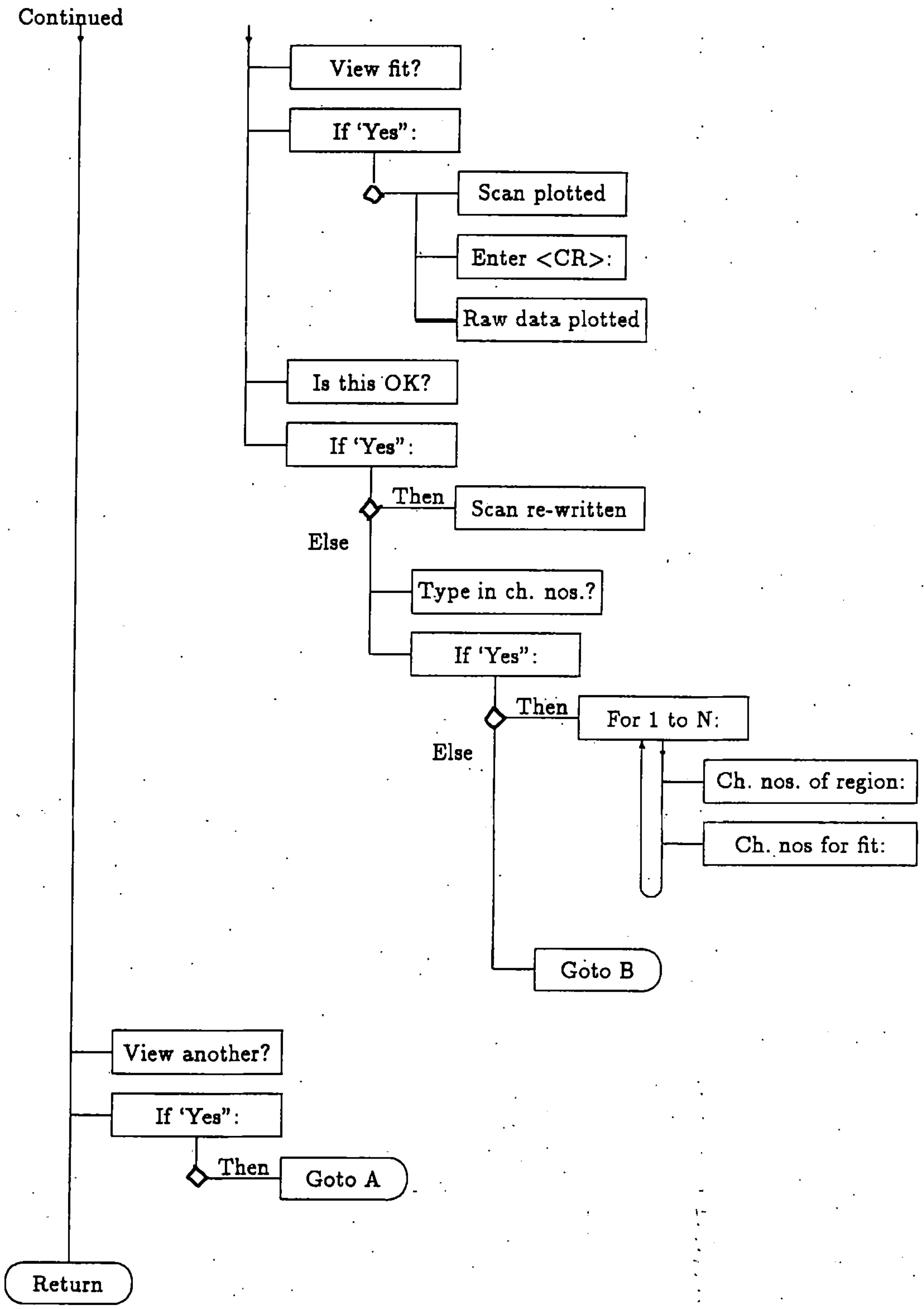
B.2.12 FITO

This routine fits a polynomial over parts of a single scan. It is part of the program LKBDEL (Section B.1.9).

This routine calls the routines AVIN, BELL, BININ, BINOUT, BITPLT, GINT and POSN, the routines FIT, FTPLT and PLT listed with the program LKBDEL and the graphics macros DRAWX and MOVEX.



P.T.O.


Continued
View fit?
If 'Yes":
Scan plotted
Enter <CR>:
Raw data plotted
Is this OK?
If 'Yes":
Then
Scan re-written
Else
Type in ch. nos.?
If 'Yes":
Then
For 1 to N:
Else
Ch. nos. of region:
Ch. nos for fit:
Goto B
View another?
If 'Yes":
Then
Goto A
Return

```
C       SINGLE SCAN FITTING

C
        SUBROUTINE FITO(FNAM,NY,NA,A,SCN,AMAX,AMIN)
        LOGICAL*1 TEMP,FNAM(15)
        INTEGER NY,SCN,AMAX,AMIN,NA,A(NA),NS,REC,XMAX,MARG,KPLUS1,NAV
        1,XT1(4),XT2(4),X1(4),X2(4),FT1(4),FT2(4),F1(4),F2(4),N,IFAIL
        2,NTMP,AMAX1,M(4),XSTEP,YSTEP,A0,X0,Y0
        REAL ERR(4),XSCAL,YSCAL
        DATA XMAX,MARG/1000,30/
C
        TYPE 5                  ! ENTER NAV
        CALL GINT(NAV)
100     TYPE 10,SCN-NAV         ! ENTER SCAN NUMBER
        CALL GINT(NS)
        IF (NS.LT.1+NAV.OR.NS.GT.SCN-NAV) GOTO 1200
        CALL PLT(FNAM,NS,NY,SCN,NA,A,AMAX,AMIN,NAV)
        TYPE 30                 ! OK ?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') GOTO 160
110     TYPE 40                 ! ENTER NUMBER OF AREAS
        CALL GINT(N)
        IF (N.LT.1.OR.N.GT.4) GOTO 1300
C
          DO 120 K=1,N
130       CALL POSN(XT1(K),XT2(K),X1(K),X2(K),MARG,NY,XMAX)
          IF (XT1(K).EQ.0.OR.XT2(K).EQ.0) GOTO 2000
          IF (XT2(K).GT.XT1(K)) GOTO 120
          CALL BELL
          TYPE *,' SECOND CH. NO. MUST BE > FIRST - RE-ENTER BOTH'
          GOTO 130
 120      CONTINUE
C
          DO 200 K=1,N
230       TYPE *,'ENTER FIT RANGES :'
          CALL POSN(FT1(K),FT2(K),F1(K),F2(K),MARG,NY,XMAX)
          IF (FT1(K).LT.XT1(K).AND.FT2(K).GT.XT2(K)) GOTO 240
          CALL BELL
          TYPE *,'FIT RANGE DOES NOT INCLUDE PEAK'
          GOTO 230
240       IF ((FT2(K)-FT1(K)+XT1(K)-XT2(K)).LE.80) GOTO 200
          CALL BELL
          TYPE *,'FIT RANGE TOO LARGE'
          GOTO 230
200       CONTINUE
C
300     TYPE 90                 ! ENTER DEGREE
        ACCEPT 20,TEMP
        KPLUS1=2
        IF (TEMP.EQ.'2') KPLUS1=3
        TYPE 80                 ! VIEW FIT?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') GOTO 220
        CALL FTPLT(FNAM,NS,NY,NA,A,AMAX,AMIN,AMAX1,NAV,N
        1,XT1,XT2,FT1,FT2,KPLUS1,IFAIL)
        IF (IFAIL.NE.0) GOTO 1400
C
          DO 210 K=1,N
          CALL MOVEX(X1(K),0)
          CALL DRAWX(X1(K),750)
          CALL MOVEX(X2(K),0)
          CALL DRAWX(X2(K),750)
 210      CONTINUE
C
        TYPE 95                 ! ENTER CR FOR ORIG DATA
        ACCEPT 20
        OPEN (UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128,ERR=1000)
        CALL AVIN(NS,NY,SCN,NA,A,NAV,98)
        CLOSE(UNIT=98)
          DO 250 K=1,N
          CALL BITPLT(NY,NA,A,AMAX1,AMIN,XT1(K),XT2(K))
```

```
250       CONTINUE
220      TYPE 430                 ! FIT OVER AREA?
         ACCEPT 20,TEMP
         IF (TEMP.NE.'Y') GOTO 1500
         TYPE *,'RE-WRITING SCAN . . . . .'
         OPEN (UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
         1,RECORDSIZE=128,ERR=1000)
         CALL BININ(NS,NY,NA,A,98)
         IF (AMAX1.LT.AMAX) AMAX1=AMAX
           DO 140 K=1,N
           CALL FIT(XT1(K),XT2(K),FT1(K),FT2(K),NY,NA,A,AMAX1,KPLUS1
         1,ERR,M,IFAIL)
           IF (IFAIL.NE.0) TYPE *,'ERROR IN ROUTINE "FIT"'
 140       CONTINUE
         CALL BINOUT(NS,NY,NA,A,98)
         IF (AMAX1.LE.AMAX) GOTO 215
         READ(98'1) NY,SCN,AMAX,AMIN,XSTEP,YSTEP,A0,XSCAL,YSCAL,X0,Y0
         WRITE(98'1) NY,SCN,AMAX1,AMIN,XSTEP,YSTEP
         1,A0,XSCAL,YSCAL,X0,Y0
         TYPE *,'AMAX UPDATED'
215      CLOSE (UNIT=98,ERR=1000)
         TYPE *,'ERRORS IN FIT :'
           DO 145 K=1,N
           ERR(K)=SQRT(ERR(K))/(M(K)-1)
           TYPE *,'PEAK ',K,' ERROR = ',ERR(K)
145        CONTINUE
160      TYPE 50          ! ANOTHER SCAN?
         ACCEPT 20,TEMP
         IF (TEMP.NE.'N') GOTO 100
         GOTO 2000
1000     TYPE *,'ERROR OPENING/CLOSING FILE - EXITING ROUTINE'
         GOTO 2000
1100     TYPE *,'ERROR READING/WRITING FILE - EXITING ROUTINE'
         CLOSE(UNIT=98)
         GOTO 2000
1200     TYPE *,'INVALID SCAN NUMBER !'
```

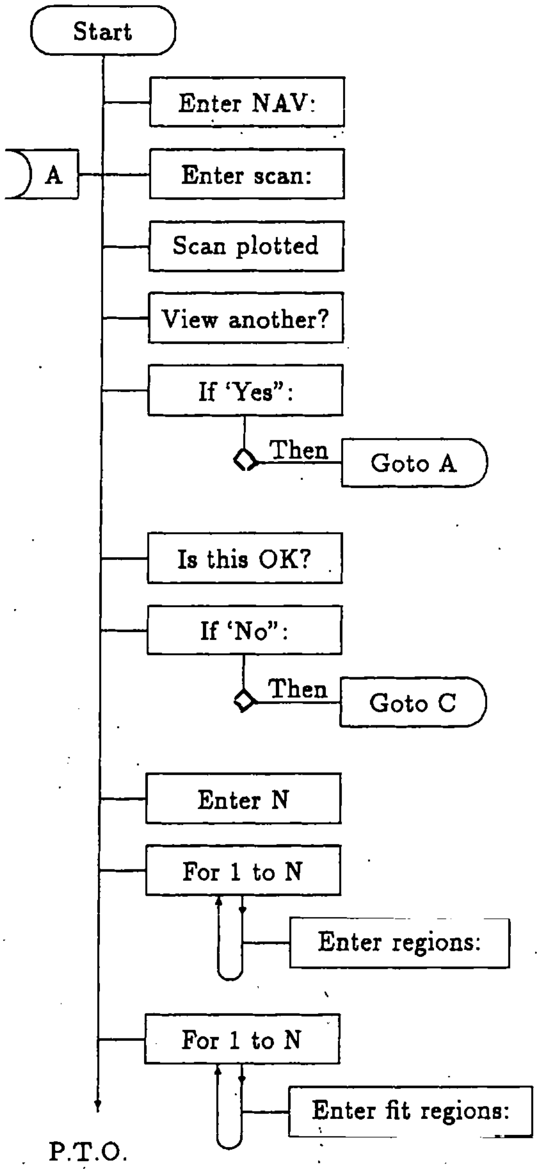
```
        GOTO 100
1300    TYPE *,'INVALID NUMBER'
        GOTO 110
1400    TYPE *,'ERROR IN ROUTINE FTPLT'
        GOTO 2000
1500    TYPE 400                         ! TYPE IN CH. NOS.
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') GOTO 160
          DO 1510 I=1,N
          TYPE 410,'L.H.S.',I
          CALL GINT(NTMP)
          IF (NTMP.NE.0) XT1(I)=NTMP
          TYPE 410,'R.H.S.',I
          CALL GINT(NTMP)
          IF (NTMP.NE.0) XT2(I)=NTMP
          TYPE 420,'L.H.S.',I
          CALL GINT(NTMP)
          IF (NTMP.NE.0) FT1(I)=NTMP
          TYPE 420,'R.H.S.',I
          CALL GINT(NTMP)
          IF (NTMP.NE.0) FT2(I)=NTMP
          X1(I)=MARG+INT(FLOAT(XT1(I))*FLOAT(XMAX-MARG)/FLOAT(NY))
          X2(I)=MARG+INT(FLOAT(XT2(I))*FLOAT(XMAX-MARG)/FLOAT(NY))
          F1(I)=MARG+INT(FLOAT(FT1(I))*FLOAT(XMAX-MARG)/FLOAT(NY))
          F2(I)=MARG+INT(FLOAT(FT2(I))*FLOAT(XMAX-MARG)/FLOAT(NY))
1510      CONTINUE
        GOTO 300
C       *******************************************************
2000    CALL TEKOUT(31)
        RETURN
C       *******************************************************
5       FORMAT(' AVERAGE N SCANS EACH SIDE OF THAT CHOSEN - '
        1,/,' ENTER N (I5) : ',$)
10      FORMAT(' ENTER SCAN NUMBER (MAX.=',I4,') : ',$)
20      FORMAT(A)
30      FORMAT(' IS THIS OK (CR=Y) ? ',$)
```

```
40      FORMAT(’ ENTER NUMBER OF AREAS TO REMOVE (1-4) : ’,$)
50      FORMAT(’ ANOTHER SCAN (CR=Y) ? ’,$)
80      FORMAT(’ VIEW FITTED SCAN (CR=Y) ? ’,$)
90      FORMAT(’ ENTER DEGREE (1 OR 2, CR=1) : ’,$)
95      FORMAT(’ ENTER <CR> FOR ORIGINAL DATA :’,$)
400     FORMAT(’ TYPE IN CHANNEL NUMBERS FOR RANGES (CR=Y) ? ’,$)
410     FORMAT(’ ENTER ’,A6,’ CHANNEL FOR PEAK ’,I1,’ (CR
        1 FOR ORIG.) : ’,$)
420     FORMAT(’ ENTER ’,A6,’ OF FIT RANGE FOR PEAK ’,I1,’ : ’,$)
430     FORMAT(’ FIT OVER THIS AREA (CR=N) ? ’,$)
        END
```

B.2.13 FIT1

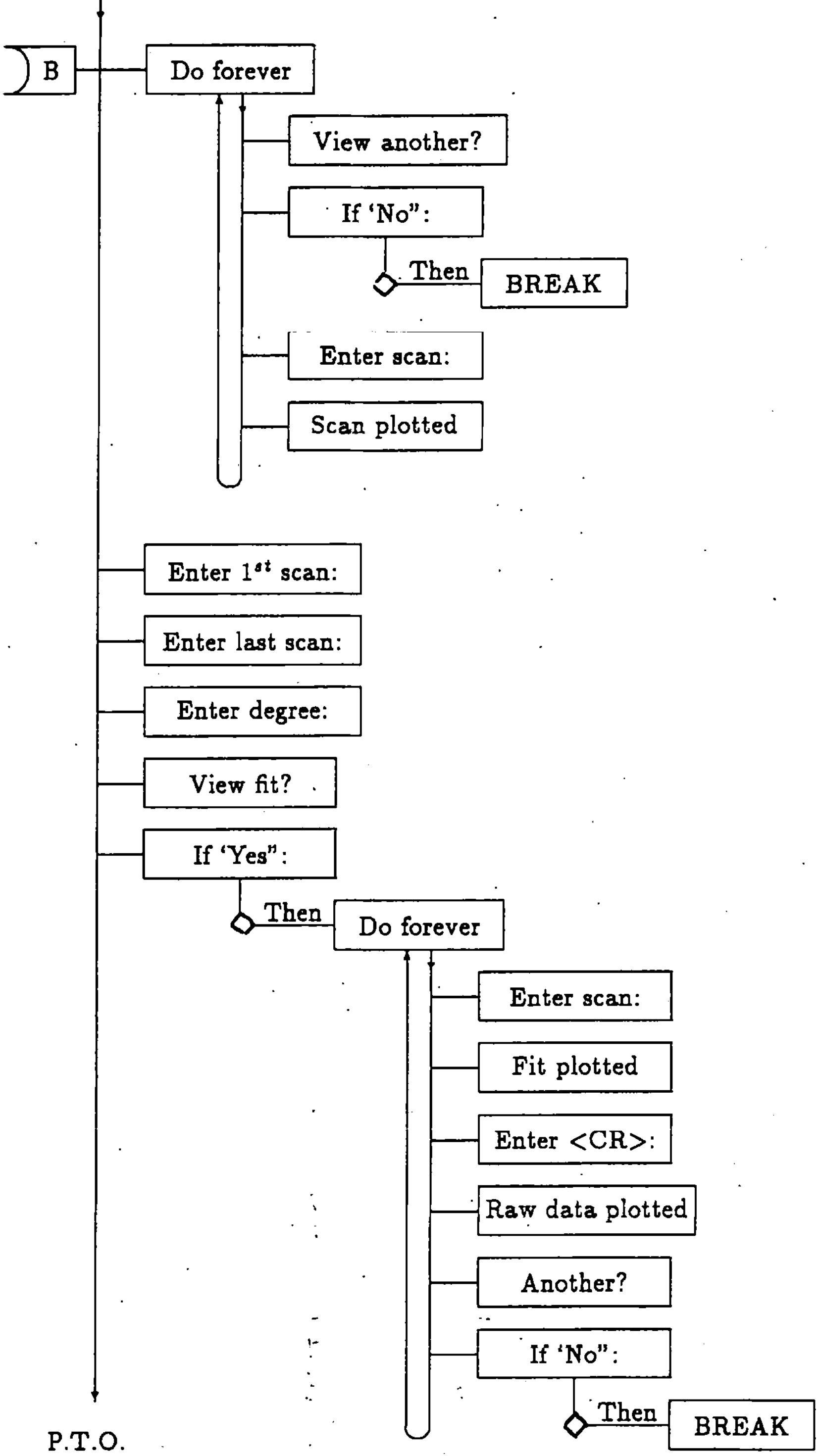
This routine fits a polynomial over the same parts of a series of scans. It is part of the program LKBDEL (Section B.1.9).

This routine calls the routines AVIN, BELL, BININ, BINOUT, BITPLT, GINT and POSN, the routines FIT, FTPLT and PLT listed with the program LKBDEL and the graphics macros DRAWX, MOVEX and TEKOUT.

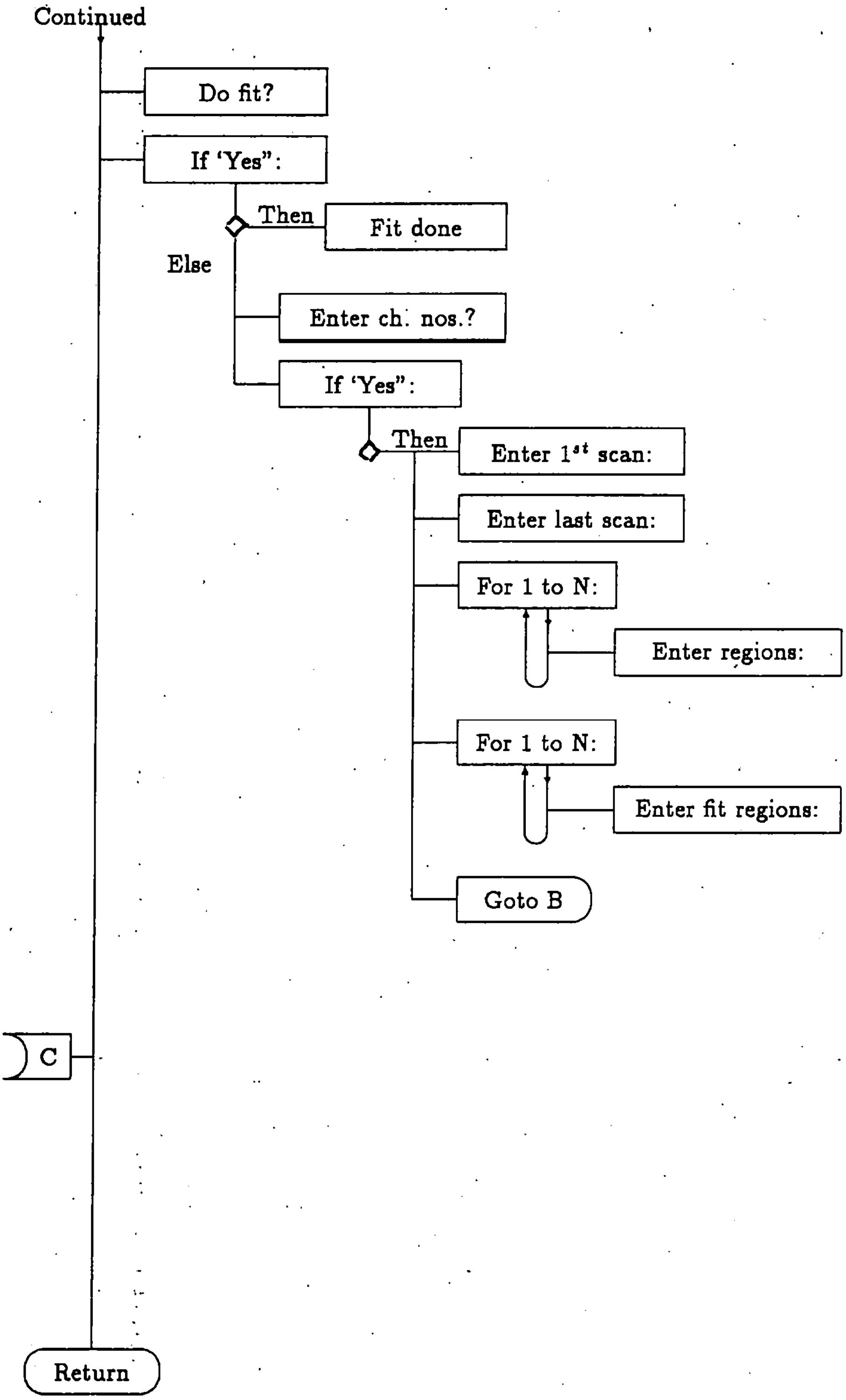


P.T.O.

Continued



P.T.O.


Continued
Do fit?
If 'Yes":
Then
Fit done
Else
Enter ch. nos.?
If 'Yes":
Then
Enter 1st scan:
Enter last scan:
For 1 to N:
Enter regions:
For 1 to N:
Enter fit regions:
Goto B
C
Return

```
C       MULTIPLE SCAN FITTING

C
        SUBROUTINE FIT1(FNAM,NY,NA,A,SCN,AMAX,AMIN)
        LOGICAL*1 TEMP,FNAM(15)
        INTEGER NY,SCN,AMAX,AMIN,NA,A(NA),NS,S1,S2,REC
        1,XMAX,MARG,KPLUS1,NAV,XT1(4),XT2(4),X1(4),X2(4)
        2,FT1(4),FT2(4),F1(4),F2(4),N,IFAIL,NTMP,AMAX1,M(4),MT
        REAL ERR,SDEV(4)
        DATA XMAX,MARG/1000,30/
        S1=0
        S2=0
        N=0
        TYPE 5                    ! AVERAGE 2N+1 SCANS - ENTER N
        CALL GINT(NAV)
C
100     TYPE 10,SCN-NAV           ! ENTER SCAN NUMBER
        CALL GINT(NS)
        IF (NS.LT.1+NAV.OR.NS.GT.SCN-NAV) GOTO 1200
        CALL PLT(FNAM,NS,NY,SCN,NA,A,AMAX,AMIN,NAV)
        TYPE 30                   ! VIEW ANOTHER ?
        ACCEPT 20,TEMP
        IF (TEMP.NE.'N') GOTO 100
        TYPE 40                   ! OK ?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') GOTO 220
110     TYPE 50                   ! ENTER NUMBER OF AREAS
        CALL GINT(N)
        IF (N.LT.1.OR.N.GT.4) GOTO 1300
C
          DO 120 K=1,N
170       CALL POSN(XT1(K),XT2(K),X1(K),X2(K),MARG,NY,XMAX)
          IF (XT1(K).LT.0.OR.XT2(K).LT.0) GOTO 2000
          IF (XT2(K).GT.XT1(K)) GOTO 120
          CALL BELL
          TYPE *,'SECOND CH. NO. MUST BE > FIRST - RE:ENTER'
          GOTO 170
 120      CONTINUE
C
          DO 300 K=1,N
230       TYPE *,'ENTER FIT RANGES :'
          CALL POSN(FT1(K),FT2(K),F1(K),F2(K),MARG,NY,XMAX)
          IF (FT1(K).LT.XT1(K).AND.FT2(K).GT.XT2(K)) GOTO 240
          CALL BELL
          TYPE *,'FIT RANGE DOES NOT INCLUDE PEAK'
          GOTO 230
240       IF ((FT2(K)-FT1(K)+XT1(K)-XT2(K)).LE.80) GOTO 300
          CALL BELL
          TYPE *,'FIT RANGE TOO LARGE'
          GOTO 230
300       CONTINUE
C
130     TYPE 30                    ! VIEW ANOTHER?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') GOTO 150
        TYPE 10,SCN-NAV            ! ENTER SCAN NUMBER
        CALL GINT(NS)
        IF (NS.LT.1+NAV.OR.NS.GT.SCN-NAV) GOTO 1210
        CALL PLT(FNAM,NS,NY,SCN,NA,A,AMAX,AMIN,NAV)
C
          DO 140 K=1,N
          CALL MOVEX(X1(K),0)
          CALL DRAWX(X1(K),750)
          CALL MOVEX(X2(K),0)
          CALL DRAWX(X2(K),750)
 140      CONTINUE
C
        GOTO 130
150     IF (S1.NE.0) GOTO 160
        TYPE 60,'FIRST',SCN                  ! ENTER FIRST SCAN
        CALL GINT(S1)
        IF (S1.EQ.0) GOTO 220
        IF (S1.LT.1.OR.S1.GT.SCN) GOTO 1600
        GOTO 130
160     TYPE 60,'LAST',SCN               ! ENTER LAST SCAN
        CALL GINT(S2)
        IF (S2.EQ.0) GOTO 220
        IF (S2.LT.1.OR.S2.GT.SCN) GOTO 1610
        TYPE 95                          ! ENTER DEGREE
        ACCEPT 20,TEMP
        KPLUS1=2
        IF (TEMP.EQ.'2') KPLUS1=3
        TYPE 80                          ! VIEW FIT ?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') GOTO 330
310     TYPE 10,SCN-NAV
        CALL GINT(NS)
        IF (NS.LT.1+NAV.OR.NS.GT.SCN-NAV) GOTO 1220
        IF (NS.LT.S1.OR.NS.GT.S2) CALL PLT(FNAM,NS,NY,SCN,NA,A
        1,AMAX,AMIN,NAV)
        IF (NS.GE.S1.AND.NS.LE.S2) CALL FTPLT(FNAM,NS,NY,NA,A
        1,AMAX,AMIN,AMAX1,NAV,N,XT1,XT2,FT1,FT2,KPLUS1,IFAIL)
        IF (IFAIL.NE.0) GOTO 1400
C
          DO 320 K=1,N
          CALL MOVEX(X1(K),0)
          CALL DRAWX(X1(K),750)
          CALL MOVEX(X2(K),0)
          CALL DRAWX(X2(K),750)
 320      CONTINUE
C
        TYPE 97                  ! ENTER CR FOR ORIG. DATA
        ACCEPT 20
        OPEN (UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
        1,RECORDSIZE=128,ERR=1000)
        CALL AVIN(NS,NY,SCN,NA,A,NAV,'N',98)
        CLOSE(UNIT=98)
```

```
       DO 340 K=1,N
       CALL BITPLT(NY,NA,A,AMAX1,AMIN,XT1(K),XT2(K))
340    CONTINUE
      TYPE 30              !VIEW ANOTHER?
      ACCEPT 20,TEMP
      IF (TEMP.NE.'N') GOTO 310
330   TYPE 90              ! FIT OVER AREAS ?
      ACCEPT 20,TEMP
      IF (TEMP.EQ.'N') GOTO 1500
      TYPE *,'RE-WRITING SCANS . . . . .'
      OPEN (UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
      1,RECORDSIZE=128,ERR=1000)
      IF (AMAX1.LT.AMAX) AMAX1=AMAX
       DO 335 K=1,N
       SDEV(K)=0
       M(K)=0
335    CONTINUE
C
       DO 210 NS=S1,S2
       CALL BININ(NS,NY,NA,A,98)
        DO 190 K=1,N
        CALL FIT(XT1(K),XT2(K),FT1(K),FT2(K),NY,NA,A,AMAX1
      1 ,KPLUS1,ERR,MT,IFAIL)
        IF (IFAIL.NE.0) TYPE *,'ERROR IN ROUTINE "FIT"'
        M(K)=M(K)+MT
        SDEV(K)=SDEV(K)+ERR
 190    CONTINUE
       CALL BINOUT(NS,NY,NA,A,98)
 210  CONTINUE
C
       DO 215 K=1,N
       SDEV(K)=SQRT(SDEV(K))/(M(K)-1)
215    CONTINUE
C
      IF (AMAX1.GT.AMAX) WRITE(98'1) NY,SCN,AMAX1,AMIN,XSTEP,YSTEP
      1,A0,XSCAL,YSCAL,X0,Y0
```

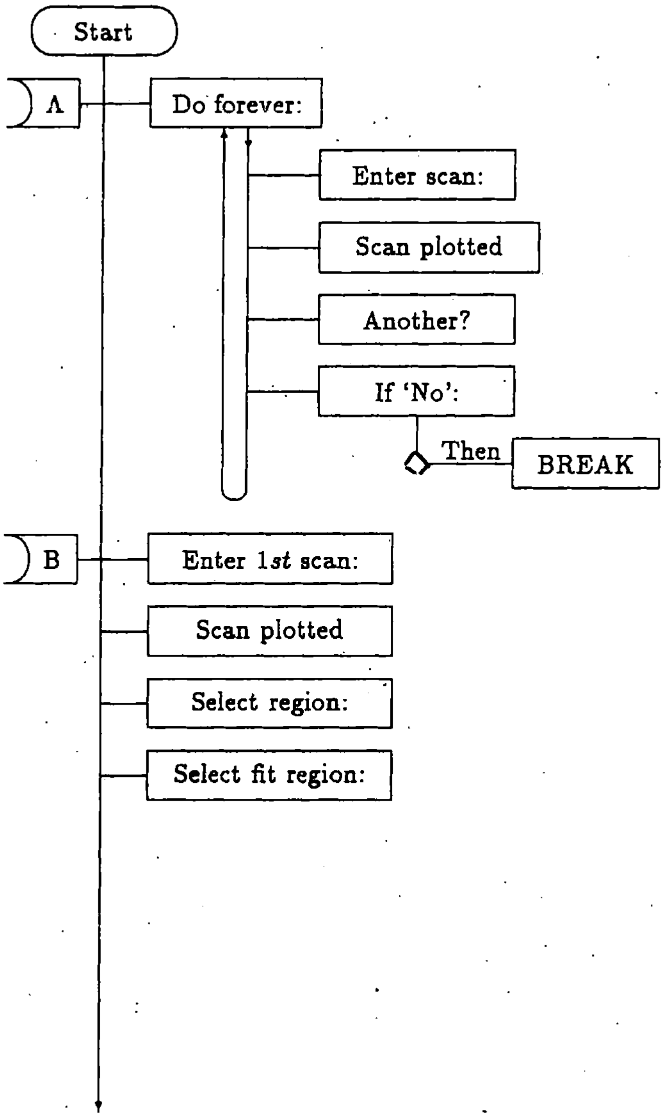
```
        IF (AMAX1.GT.AMAX) TYPE *,'AMAX UPDATED'
        CLOSE (UNIT=98,ERR=1000)
        TYPE *,'ERRORS IN FIT :'
          DO 225 K=1,N
          TYPE *,'PEAK ',K,' : ',SDEV(K)
225       CONTINUE
220     IF (N.EQ.0) GOTO 2000
        S1=0
        S2=0
        TYPE 70           ! ANOTHER RANGE?
        ACCEPT 20,TEMP
        IF (TEMP.NE.'N') GOTO 130
        GOTO 2000
1000    TYPE *,'ERROR OPENING/CLOSING FILE - EXITING ROUTINE'
        GOTO 2000
1100    TYPE *,'ERROR READING/WRITING FILE - EXITING ROUTINE'
        CLOSE(UNIT=98)
        GOTO 2000
1200    TYPE *,'INVALID SCAN NUMBER !'
        TYPE *,1+NAV,' < NS < ',SCN-NAV
        GOTO 100
1210    TYPE *,'INVALID SCAN NUMBER !'
        TYPE *,1+NAV,' < NS < ',SCN-NAV
        GOTO 130
1220    TYPE *,'INVALID SCAN NUMBER !'
        TYPE *,1+NAV,' < NS < ',SCN-NAV
        GOTO 310
1300    TYPE *,'INVALID NUMBER'
        GOTO 110
1400    TYPE *,'ERROR IN ROUTINE FTPLT'
        GOTO 2000
1500    TYPE 400                        ! TYPE IN CH. NOS. FOR FIT ?
        ACCEPT 20,TEMP
        IF (TEMP.EQ.'N') GOTO 220
        S1=0
          DO 1510 I=1,N
          TYPE 410,’L.H.S.’,I
          CALL GINT(NTMP)
          IF (NTMP.NE.0) XT1(I)=NTMP
          TYPE 410,’R.H.S.’,I
          CALL GINT(NTMP)
          IF (NTMP.NE.0) XT2(I)=NTMP
          TYPE 420,’L.H.S.’,I
          CALL GINT(NTMP)
          IF (NTMP.NE.0) FT1(I)=NTMP
          TYPE 420,’R.H.S.’,I
          CALL GINT(NTMP)
          IF (NTMP.NE.0) FT2(I)=NTMP
          X1(I)=MARG+INT(FLOAT(XT1(I))*FLOAT(XMAX-MARG)/FLOAT(NY))
          X2(I)=MARG+INT(FLOAT(XT2(I))*FLOAT(XMAX-MARG)/FLOAT(NY))
          F1(I)=MARG+INT(FLOAT(FT1(I))*FLOAT(XMAX-MARG)/FLOAT(NY))
          F2(I)=MARG+INT(FLOAT(FT2(I))*FLOAT(XMAX-MARG)/FLOAT(NY))
1510      CONTINUE
        GOTO 130
1600    TYPE *,’INVALID SCAN NUMBER - PLEASE REPEAT’
        S1=0
        GOTO 150
1610    TYPE *,’INVALID SCAN NUMBER - PLEASE REPEAT’
        GOTO 160
C       ******************************************************
2000    CALL TEKOUT(31)
        RETURN
C       ******************************************************
5       FORMAT(’ AVERAGE N SCANS EACH SIDE OF THAT SELECTED - ’
        1,/,’ ENTER N (I5) : ’,$)
10      FORMAT(’ ENTER SCAN NUMBER (MAX.=’,I4,’) : ’,$)
20      FORMAT(A)
30      FORMAT(’ VIEW ANOTHER SCAN (CR=Y) ? ’,$)
40      FORMAT(’ IS THIS OK (CR=Y) ? ’,$)
50      FORMAT(’ ENTER NUMBER OF AREAS TO REMOVE (1-4) : ’,$)
60      FORMAT(’ ENTER ’,A5,’ SCAN (0=NO SCANS, MAX.=’,I4,’) : ’,$)
70      FORMAT(’ ANOTHER RANGE OF SCANS (SAME AREAS) (CR=Y) ? ’,$)
```

```
80      FORMAT(' VIEW FITTED SCANS (CR=Y) ? ',$)
90      FORMAT(' FIT OVER THESE PARTS (CR=Y) ?',$)
95      FORMAT(' ENTER DEGREE (1 OR 2, CR=1) : ',$)
97      FORMAT(' ENTER <CR> FOR ORIGINAL DATA :',$)
400     FORMAT(' TYPE IN CHANNEL NUMBERS FOR RANGES (CR=Y) ? ',$)
410     FORMAT(' ENTER ',A6,' CHANNEL FOR PEAK ',I1,' (CR
        1 FOR ORIG.) : ',$)
420     FORMAT(' ENTER ',A6,' OF FIT RANGE FOR PEAK ',I1,' : ',$)
        END
```

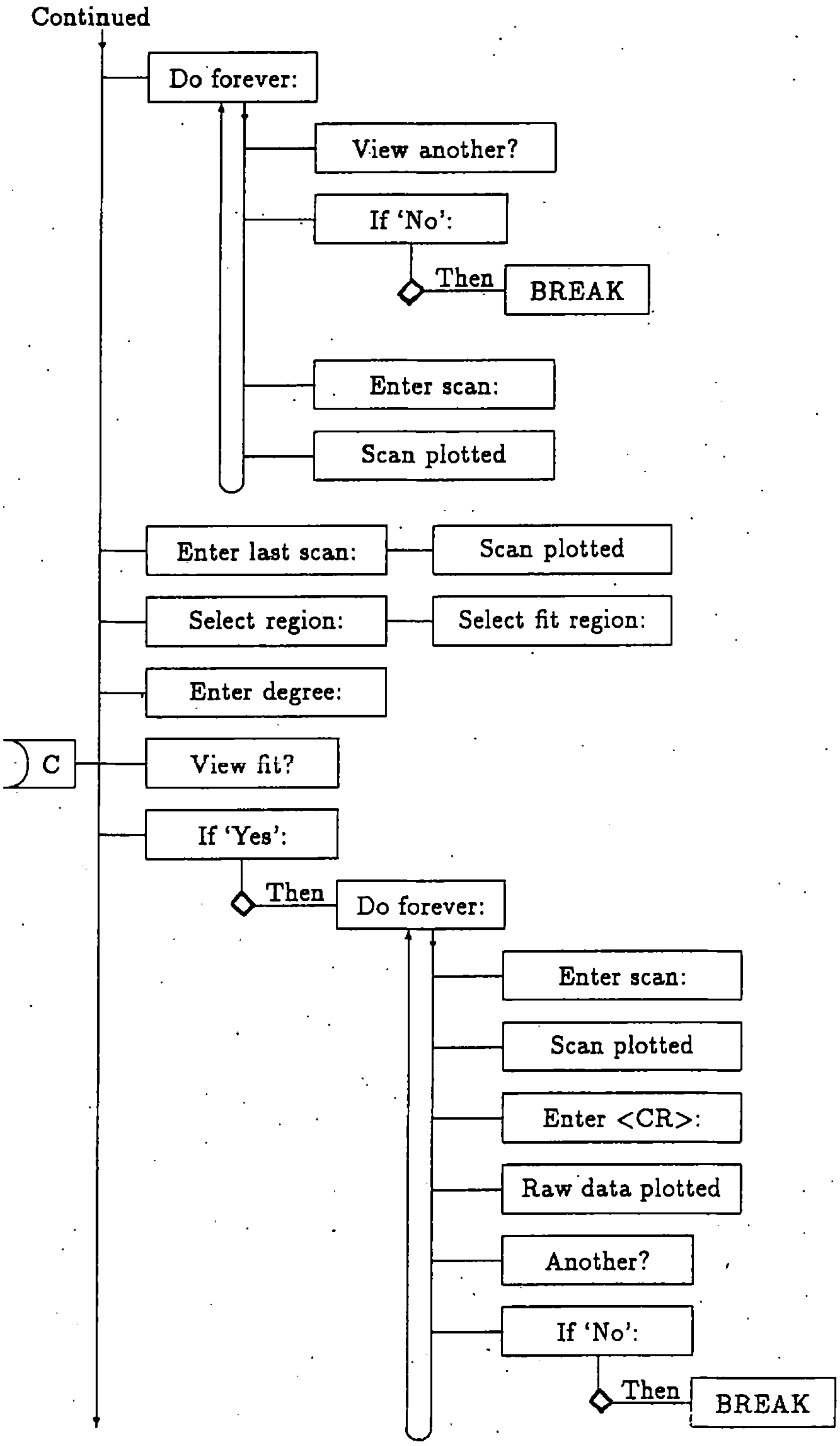
B.2.14 FIT2

This routine fits a polynomial over a diagonal area in a film pattern. It is part of the program LKBDEL (Section B.1.9).

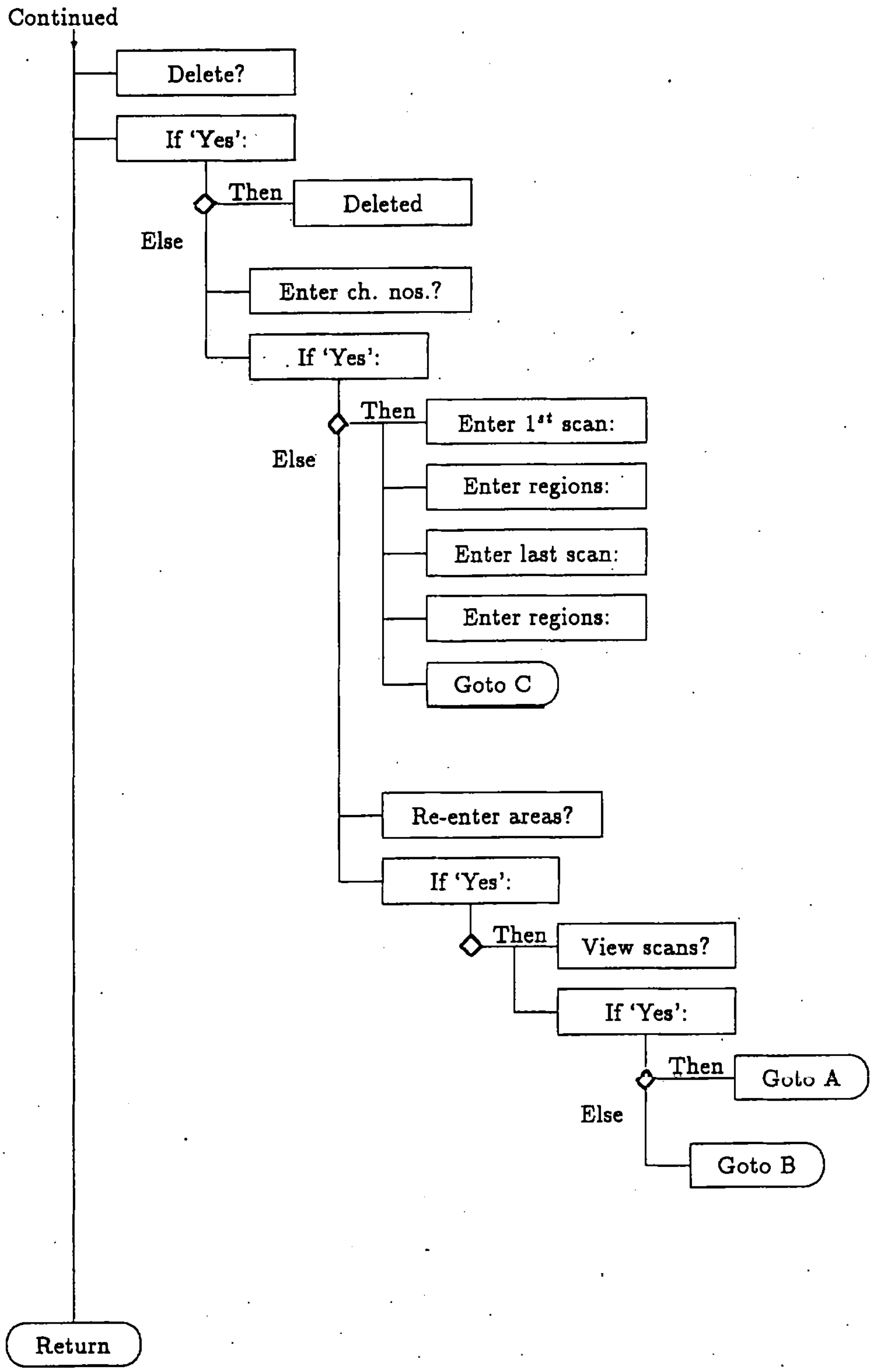
This routine calls the routines BELL, BININ, BINOUT, BITPLT, GINT and POSN, the routines FIT, FTPLT and PLT listed with the program LKBDEL and the graphics macros DRAWX, MOVEX and TEKOUT.



P.T.O.


Continued
Do forever:
View another?
If 'No':
Then
BREAK
Enter scan:
Scan plotted
Enter last scan:
Scan plotted
Select region:
Select fit region:
Enter degree:
C
View fit?
If 'Yes':
Then
Do forever:
Enter scan:
Scan plotted
Enter <CR>:
Raw data plotted
Another?
If 'No':
Then
BREAK

P.T.O.


Continued
Delete?
If 'Yes':
Then
Deleted
Else
Enter ch. nos.?
If 'Yes':
Then
Enter 1st scan:
Enter regions:
Enter last scan:
Enter regions:
Goto C
Else
Re-enter areas?
If 'Yes':
Then
View scans?
If 'Yes':
Then
Goto A
Else
Goto B
Return

```
C DIAGONAL AREA FITTING
C
SUBROUTINE FIT2(FNAM,NY,NA,A,SCN,AMAX,AMIN)
LOGICAL*1 TEMP,FNAM(15)
INTEGER IFAIL,KPLUS1
INTEGER NY,SCN,AMAX,AMIN,NA,A(NA),NS,S1,S2,REC,MARG,XMAX
INTEGER XT1(4),XT2(4),X1(3),X2(3),FT1(4),FT2(4),F1(3),F2(3)
1,NTMP,AMAX1,M,MT
REAL ERR,SDEV
DATA XMAX,MARG/1000,30/
S1=0
S2=0
C
100 TYPE 10,SCN ! ENTER SCAN NUMBER
CALL GINT(NS)
IF (NS.GT.SCN.OR.NS.LT.1) GOTO 1200
CALL PLT(FNAM,NS,NY,SCN,NA,A,AMAX,AMIN,0)
110 TYPE 30 ! VIEW ANOTHER?
ACCEPT 20,TEMP
IF (TEMP.NE.'N') GOTO 100
120 TYPE 50,'FIRST',SCN
CALL GINT(S1)
IF (S1.EQ.0) GOTO 230
IF (S1.LT.1.OR.S1.GT.SCN) GOTO 1210
IF (S1.NE.NS) CALL PLT(FNAM,S1,NY,SCN,NA,A,AMAX,AMIN,0)
NS=S1
TYPE 40 ! OK?
ACCEPT 20,TEMP
IF (TEMP.EQ.'N') GOTO 110
300 CALL POSN(XT1(2),XT2(2),X1(2),X2(2),MARG,NY,XMAX)
IF (XT1(2).LE.0.OR.XT2(2).LE.0) GOTO 2000
IF (XT1(2).GT.XT2(2).OR.XT2(2).GT.NY) GOTO 1400
250 TYPE *,'ENTER RANGE FOR FIT :'
310 CALL POSN(FT1(2),FT2(2),F1(2),F2(2),MARG,NY,XMAX)
IF (FT1(2).LE.0.OR.FT2(2).LE.0) GOTO 2000
IF (FT2(2).GT.NY) GOTO 1500
IF (FT1(2).LT.XT1(2).AND.FT2(2).GT.XT2(2)) GOTO 240
CALL BELL
TYPE *,'FIT RANGE DOES NOT INCLUDE PEAK'
GOTO 250
240 IF ((FT2(2)-FT1(2)+XT1(2)-XT2(2)).LE.80) GOTO 130
CALL BELL
TYPE *,'FIT RANGE TOO LARGE'
GOTO 250
130 TYPE 30 ! VIEW ANOTHER?
ACCEPT 20,TEMP
IF (TEMP.EQ.'N') GOTO 150
140 TYPE 10,SCN ! ENTER SCAN NUMBER
CALL GINT(NS)
IF (NS.GT.SCN.OR.NS.LT.1) GOTO 1220
CALL PLT(FNAM,NS,NY,SCN,NA,A,AMAX,AMIN,0)
GOTO 130
150 TYPE 50,' LAST',SCN
CALL GINT(S2)
IF (S2.EQ.0) GOTO 230
IF (S2.LT.1.OR.S2.GT.SCN) GOTO 1230
IF (S2.NE.NS) CALL PLT(FNAM,S2,NY,SCN,NA,A,AMAX,AMIN,0)
NS=S2
TYPE 40 ! OK?
ACCEPT 20,TEMP
IF (TEMP.EQ.'N') GOTO 130
320 CALL POSN(XT1(3),XT2(3),X1(3),X2(3),MARG,NY,XMAX)
IF (XT1(3).LE.0.OR.XT2(3).LE.0) GOTO 2000
IF (XT1(3).GT.XT2(3).OR.XT2(3).GT.NY) GOTO 1410
270 TYPE *,'ENTER RANGE FOR FIT :'
330 CALL POSN(FT1(3),FT2(3),F1(3),F2(3),MARG,NY,XMAX)
IF (FT1(3).LE.0.OR.FT2(3).LE.0) GOTO 2000
IF (FT2(3).GT.NY) GOTO 1510
IF (FT1(3).LT.XT1(3).AND.FT2(3).GT.XT2(3)) GOTO 260
CALL BELL
TYPE *,'FIT RANGE DOES NOT INCLUDE PEAK'
```

```
GOTO 270
260 IF ((FT2(2)-FT1(2)+XT1(2)-XT2(2)).LE.80) GOTO 160
CALL BELL
TYPE *,'FIT RANGE TOO LARGE'
GOTO 270
160 TYPE 90 ! ENTER DEGREE
ACCEPT 20,TEMP
KPLUS1=2
IF (TEMP.EQ.'2') KPLUS1=3
TYPE 80 ! VIEW FIT?
ACCEPT 20,TEMP
IF (TEMP.EQ.'N') GOTO 180
170 TYPE 10,SCN
CALL GINT(NS)
IF (NS.LT.1.OR.NS.GT.SCN) GOTO 1240
IF (NS.LT.S1.OR.NS.GT.S2) CALL PLT(FNAM,NS,NY,SCN,NA,A
1,AMAX,AMIN,0)
IF (NS.LT.S1.OR.NS.GT.S2) GOTO 165
XT1(1)=XT1(2)+INT(0.5+(FLOAT(XT1(3)-XT1(2))*FLOAT(NS-S1)/
1FLOAT(S2-S1)))
XT2(1)=XT2(2)+INT(0.5+(FLOAT(XT2(3)-XT2(2))*FLOAT(NS-S1)/
1FLOAT(S2-S1)))
FT1(1)=FT1(2)+INT(0.5+(FLOAT(FT1(3)-FT1(2))*FLOAT(NS-S1)/
1FLOAT(S2-S1)))
FT2(1)=FT2(2)+INT(0.5+(FLOAT(FT2(3)-FT2(2))*FLOAT(NS-S1)/
1FLOAT(S2-S1)))
X1(1)=MARG+INT(FLOAT(XT1(1))*FLOAT(XMAX-MARG)/FLOAT(NY))
X2(1)=MARG+INT(FLOAT(XT2(1))*FLOAT(XMAX-MARG)/FLOAT(NY))
F1(1)=MARG+INT(FLOAT(FT1(1))*FLOAT(XMAX-MARG)/FLOAT(NY))
F2(1)=MARG+INT(FLOAT(FT2(1))*FLOAT(XMAX-MARG)/FLOAT(NY))
CALL FTPLT(FNAM,NS,NY,NA,A,AMAX,AMIN,AMAX1,0,1
1,XT1,XT2,FT1,FT2,KPLUS1,IFAIL)
IF (IFAIL.NE.0) GOTO 1300
CALL MOVEX(X1(1),0)
CALL DRAWX(X1(1),750)
CALL MOVEX(X2(1),0)
CALL DRAWX(X2(1),750)
CALL MOVEX(F1(1),0)
CALL DRAWX(F1(1),750)
CALL MOVEX(F2(1),0)
CALL DRAWX(F2(1),750)
TYPE 95 ! ENTER <CR> FOR ORIGINAL
ACCEPT 20,TEMP
OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
1,RECORDSIZE=128,ERR=1000)
CALL BININ(NS,NY,NA,A,98)
CLOSE(UNIT=98)
CALL BITPLT(NY,NA,A,AMAX1,AMIN,XT1,XT2)
165 TYPE 30 ! VIEW ANOTHER?
ACCEPT 20,TEMP
IF (TEMP.NE.'N') GOTO 170
180 TYPE 70 ! DELETE AREA?
ACCEPT 20,TEMP
IF (TEMP.EQ.'N') GOTO 1600
OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
1,RECORDSIZE=128,ERR=1000)
IF (AMAX1.LT.AMAX) AMAX1=AMAX
C
SDEV=0
M=0
  DO 220 NS=S1,S2
  CALL BININ(NS,NY,NA,A,98)
  XT1(1)=XT1(2)+((XT1(3)-XT1(2))*(NS-S1)/(S2-S1))
  XT2(1)=XT2(2)+((XT2(3)-XT2(2))*(NS-S1)/(S2-S1))
  FT1(1)=FT1(2)+((FT1(3)-FT1(2))*(NS-S1)/(S2-S1))
  FT2(1)=FT2(2)+((FT2(3)-FT2(2))*(NS-S1)/(S2-S1))
  CALL FIT(XT1(1),XT2(1),FT1(1),FT2(1),NY,NA,A,AMAX1
1,KPLUS1,ERR,MT,IFAIL)
  M=M+MT
  SDEV=SDEV+ERR
  CALL BINOUT(NS,NY,NA,A,98)
 220   CONTINUE
```

```
SDEV=SQRT(SDEV)/(M-1)
C
IF (AMAX1.GT.AMAX) WRITE (98'1) NY,SCN,AMAX1,AMIN,XSTEP,YSTEP
1,A0,XSCAL,YSCAL,X0,Y0
IF (AMAX1.GT.AMAX) TYPE *,'AMAX UPDATED'
CLOSE(UNIT=98,ERR=1000)
TYPE *,'ERROR IN FIT = ',SDEV
S1=0
S2=0
GOTO 2000
230 S1=0
S2=0
NS=0
TYPE 60 ! RE-ENTER AREA?
ACCEPT 20,TEMP
IF (TEMP.NE.'N') GOTO 110
GOTO 2000
C *********************************************************
1000 TYPE *,'ERROR OPENING/CLOSING FILE - EXITING ROUTINE'
GOTO 2000
1100 TYPE *,'ERROR READING/WRITING FILE - EXITING ROUTINE'
CLOSE(UNIT=98)
GOTO 2000
1200 TYPE *,'INVALID SCAN NUMBER !'
GOTO 100
1210 TYPE *,'INVALID SCAN NUMBER !'
GOTO 120
1220 TYPE *,'INVALID SCAN NUMBER !'
GOTO 140
1230 TYPE *,'INVALID SCAN NUMBER !'
GOTO 150
1240 TYPE *,'INVALID SCAN NUMBER !'
GOTO 170
1300 TYPE *,'ERROR IN ROUTINE FTPLT'
GOTO 2000
1400 TYPE *,'INVALID POSITIONS - L.H.S.>R.H.S. OR R.H.S.> NY'
```

```
TYPE *,'PLEASE RE-ENTER'
CALL BELL
GOTO 300
1410 TYPE *,'INVALID POSITIONS - L.H.S.>R.H.S. OR R.H.S.> NY'
TYPE *,'PLEASE RE-ENTER'
CALL BELL
GOTO 320
1500 TYPE *,'INVALID NUMBERS - R.H.S.> NY - PLEASE RE-ENTER'
CALL BELL
GOTO 310
1510 TYPE *,'INVALID NUMBERS - R.H.S.> NY - PLEASE RE-ENTER'
CALL BELL
GOTO 330
1600 TYPE 75 ! TYPE IN CH. NOS. FOR FIT ?
ACCEPT 20,TEMP
IF (TEMP.EQ.'N') GOTO 230
TYPE 55,'FIRST' ! ENTER SCAN NUMBER
CALL GINT(NTMP)
IF (NTMP.NE.0) S1=NTMP
TYPE 85,'L.H.S.'
CALL GINT(NTMP)
IF (NTMP.NE.0) XT1(2)=NTMP
TYPE 85,'R.H.S.'
CALL GINT(NTMP)
IF (NTMP.NE.0) XT2(2)=NTMP
TYPE 86,'L.H.S.'
CALL GINT(NTMP)
IF (NTMP.NE.0) FT1(2)=NTMP
TYPE 86,'R.H.S.'
CALL GINT(NTMP)
IF (NTMP.NE.0) FT2(2)=NTMP
X1(2)=MARG+INT(FLOAT(XT1(2))*FLOAT(XMAX-MARG)/FLOAT(NY))
X2(2)=MARG+INT(FLOAT(XT2(2))*FLOAT(XMAX-MARG)/FLOAT(NY))
F1(2)=MARG+INT(FLOAT(FT1(2))*FLOAT(XMAX-MARG)/FLOAT(NY))
F2(2)=MARG+INT(FLOAT(FT2(2))*FLOAT(XMAX-MARG)/FLOAT(NY))
TYPE 55,'LAST' ! ENTER SCAN NUMBER
```

```
CALL GINT(NTMP)
IF (NTMP.NE.0) S2=NTMP
TYPE 85,'L.H.S.'
CALL GINT(NTMP)
IF (NTMP.NE.0) XT1(3)=NTMP
TYPE 85,'R.H.S.'
CALL GINT(NTMP)
IF (NTMP.NE.0) XT2(3)=NTMP
TYPE 86,'L.H.S.'
CALL GINT(NTMP)
IF (NTMP.NE.0) FT1(3)=NTMP
TYPE 86,'R.H.S.'
CALL GINT(NTMP)
IF (NTMP.NE.0) FT2(3)=NTMP
X1(3)=MARG+INT(FLOAT(XT1(3))*FLOAT(XMAX-MARG)/FLOAT(NY))
X2(3)=MARG+INT(FLOAT(XT2(3))*FLOAT(XMAX-MARG)/FLOAT(NY))
F1(3)=MARG+INT(FLOAT(FT1(3))*FLOAT(XMAX-MARG)/FLOAT(NY))
F2(3)=MARG+INT(FLOAT(FT2(3))*FLOAT(XMAX-MARG)/FLOAT(NY))
TYPE 80 ! VIEW FITTED SCANS ?
ACCEPT 20,TEMP
IF (TEMP.NE.'N') GOTO 170
GOTO 180
C *********************************************************
2000 CALL TEKOUT(31)
RETURN
C *********************************************************
10 FORMAT(' ENTER SCAN NUMBER (MAX.=',I4,') : ',$)
20 FORMAT(A)
30 FORMAT(' VIEW ANOTHER SCAN (CR=Y) ? ',$)
40 FORMAT(' IS THIS OK (CR=Y) ? ',$)
50 FORMAT(' ENTER ',A5,' SCAN (0=NO SCANS, MAX.=',I4,') : ',$)
55 FORMAT(' ENTER ',A5,' SCCAN (CR FOR PREVIOUS VALUE) : ',$)
60 FORMAT(' RE-ENTER AREA USING CROSSWIRES (CR=Y) ? ',$)
70 FORMAT(' FIT OVER THIS AREA (CR=Y) ? ',$)
75 FORMAT(' TYPE IN CHANNEL NUMBERS FOR RANGES (CR=Y) ? ',$)
80 FORMAT(' VIEW FITTED SCANS (CR=Y) ? ',$)
```

```
85 FORMAT(' ENTER ',A6,' CHANNEL : ',$)
86 FORMAT(' ENTER ',A6,' OF FIT RANGE : ',$)
90 FORMAT(' ENTER DEGREE (1 OR 2, CR=1) : ',$)
95 FORMAT(' ENTER <CR> TO SEE ORIGINAL DATA : ',$)
END
```

B.2.15 FNC

This routine is the function $FNC = M + INT(D \times X)$.

```
FUNCTION FNC(M,D,X)
INTEGER FNC,M,X
REAL D
FNC=M+INT(D*(FLOAT(X)))
RETURN
END
```

B.2.16 GETNAM

This routine gets a .DAT file name that the user types in, checking that it is in the correct form.

```
      SUBROUTINE GETNAM(FNAM)
      LOGICAL*1 FNAM(15),TEMP,DEV(4)
100   FNAM(1)='D'
      FNAM(2)='L'
      FNAM(3)='0'
      FNAM(4)=':'
      FNAM(11)='.'
      FNAM(12)='D'
      FNAM(13)='A'
      FNAM(14)='T'
      FNAM(15)=0
      TYPE 10                              ! DEVICE
      ACCEPT 20,(DEV(I),I=1,4)
      IF (DEV(1).EQ.' ') GOTO 120
      DO 110 I=1,4
      FNAM(I)=DEV(I)
 110  CONTINUE
120   TYPE 30                              ! FILENAME
      ACCEPT 20,(FNAM(I),I=5,7)
      TYPE 60
      ACCEPT 20,(DEV(I),I=1,3)
      IF (DEV(1).EQ.' ') GOTO 130
      FNAM(12)=DEV(1)
      FNAM(13)=DEV(2)
      FNAM(14)=DEV(3)
130   TYPE 40,(FNAM(I),I=1,15)
      TYPE 50
      ACCEPT 20,TEMP
      IF (TEMP.EQ.'N') GOTO 100
      RETURN
10    FORMAT(' ENTER DEVICE (CR FOR DL0:) : ',$)
20    FORMAT(A:2A:A)
```

```
30      FORMAT(' ENTER 3-CHARACTER RUN IDENTIFIER : ',$)
40      FORMAT(' FILENAME IS ',15A)
50      FORMAT(' IS THIS OK (Y/N: CR=Y) ?',$)
60      FORMAT(' ENTER FILE TYPE (CR FOR "DAT") :',$)
        END
```

B.2.17 GINT

This routine accepts an integer from the keyboard.

This routine calls the routine BELL.

```
        SUBROUTINE GINT(N)
C       ACCEPTS A FIVE-DIGIT INTEGER, N.
        INTEGER N,S
        LOGICAL*1 A(5),B(7)
100     ACCEPT 10,(B(I),I=1,7)
        S=1
        IF (B(1).EQ.'-') S=-1
        IF (S.EQ.1) N=0
        IF (S.EQ.-1.OR.B(1).EQ.'+') N=1
        IF (B(6+N).NE.' ') GOTO 1100
        DO 105 I=1,5
        A(I)=B(I+N)
105     CONTINUE
        N=0
110     N=N+1
        IF (A(5).NE.' ') GOTO 130
          DO 120 I=1,4
          A(6-I)=A(5-I)
120       CONTINUE
        A(1)='0'
        IF (N.LT.5) GOTO 110
130       DO 140 I=1,5
          IF ((A(I).LT.48.OR.A(I).GT.57).AND.A(I).NE.' ') GOTO 1000
140       CONTINUE
        DECODE(5,20,A,ERR=1000) N
        N=N*S
        GOTO 2000
C
1000    CALL BELL
        TYPE 30                 ! NOT INTEGER - RE-ENTER
        GOTO 100
1100    CALL BELL
```

```
        TYPE 40                  ! MORE THAN 5 DIGIIS - RE-ENTER
        GOTO 100
2000    RETURN
10      FORMAT(7A)
20      FORMAT(I5)
30      FORMAT(' NOT INTEGER - RE-ENTER : ',$)
40      FORMAT(' ONLY FIVE DIGITS ALLOWED - RE-ENTER : ',$)
        END
```

B.2.18 GREAL

The routine GREAL accepts a real number from the keyboard.

This routine calls the routine BELL.

```
        SUBROUTINE GREAL(X)
C       ACCEPTS A REAL NUMBER : F12.0
        REAL X
        INTEGER N,S
        LOGICAL*1 A(12),B(13)
        EQUIVALENCE (A(1),B(1))
100     DO 110 I=1,13
        B(I)=' '
110     CONTINUE
        ACCEPT 10,(B(I),I=1,13)
        IF (B(13).NE.' ') GOTO 1100
120     IF (A(12).NE.' ') GOTO 140
        DO 130 I=1,11
        A(13-I)=A(12-I)
130     CONTINUE
        A(1)='0'
        GOTO 120
140     DECODE(12,20,A,ERR=1000) X
        GOTO 2000
C
1000    TYPE 50,(A(I),I=1,12)
        TYPE 30                 ! INPUT CONVERSION ERROR - RE-ENTER
        GOTO 100
1100    TYPE 40                 ! MORE THAN 12 DIGITS - RE-ENTER
        GOTO 100
2000    RETURN
10      FORMAT(13A)
20      FORMAT(F12.0)
30      FORMAT(' INPUT CONVERSION ERROR - RE-ENTER : ',$)
40      FORMAT(' ONLY 12 DIGITS ALLOWED (INC. ".") - RE-ENTER : ',$)
50      FORMAT(X,13A)
        END
```

B.2.19 GRPLT

This routine plots the scan in array A on the graph plotter.

This routine calls the routines CHRPLT and SIZE, and the graphics macro P12M.

```
      SUBROUTINE GRPLT(FNAM,NY,NA,A,AMAX,AMIN)
      INTEGER NY,NA,A(NA),AMAX,AMIN,OLDX,OLDY,XT,AT
      INTEGER XMAX,YMAX,XDIV,YDIV,N,M
      REAL DX,DA,B
      LOGICAL*1 PEN,FNAM(15),TEMP(4)
      DATA OLDX,OLDY/0,0/
      DATA XMAX,YMAX/3700,2700/
      MARG=250
      IF (AMAX.LE.5) GOTO 1000
      DX=FLOAT(XMAX-MARG)/FLOAT(NY)
      DA=FLOAT(YMAX-(2*MARG))/FLOAT(AMAX)
      PEN=.FALSE.
      CALL P12M(MARG,MARG,OLDX,OLDY,.FALSE.)
      CALL P12M(XMAX,MARG,OLDX,OLDY,.TRUE.)
C     X-AXIS DRAWN
      CALL SIZE(NY,B,N)
      IF (B.LE.5.) XDIV=(10**N)/2
      IF (B.LE.2.) XDIV=(10**N)/5
      IF (B.GT.5.) XDIV=10**N
        DO 20 I=1,INT(FLOAT(NY)/FLOAT(XDIV))
        XT=MARG+INT(FLOAT(I*XDIV)*DX)
        CALL P12M(XT,(MARG-30),OLDX,OLDY,.FALSE.)
        CALL P12M(XT,MARG,OLDX,OLDY,.TRUE.)
        M=I*XDIV
        ENCODE(4,100,TEMP) M
        CALL CHRPLT((XT-70),(MARG-150),OLDX,OLDY,4,TEMP)
 20     CONTINUE
C     **********************************************************
      CALL P12M(MARG,MARG,OLDX,OLDY,.FALSE.)
      CALL P12M(MARG,YMAX,OLDX,OLDY,.TRUE.)
C     Y-AXIS DRAWN
      CALL SIZE(AMAX,B,N)
      IF (B.LE.5.) YDIV=(10**N)/2
      IF (B.GT.5.) YDIV=10**N
        DO 30 I=1,INT(FLOAT(AMAX)/FLOAT(YDIV))
        AT=MARG+INT(FLOAT(I*YDIV)*DA)
        CALL P12M((MARG-30),AT,OLDX,OLDY,.FALSE.)
        CALL P12M(MARG,AT,OLDX,OLDY,.TRUE.)
        M=I*YDIV
        ENCODE(4,100,TEMP) M
        CALL CHRPLT(0,AT,OLDX,OLDY,4,TEMP)
 30     CONTINUE
C     *********************************************************
        DO 10 I=1,NY
        IF (A(I).LE.AMAX) GOTO 40
        PEN=.FALSE.
        GOTO 10
40      AT=MARG+INT(DA*FLOAT(A(I)))
        XT=MARG+INT(DX*FLOAT(I))
        CALL P12M(XT,AT,OLDX,OLDY,PEN)
        PEN=.TRUE.
 10     CONTINUE
      CALL CHRPLT((XMAX-500),(YMAX-MARG),OLDX,OLDY,15,FNAM)
      GOTO 2000
1000  TYPE *,'AMAX TOO SMALL - EXITING ROUTINE GRPLT'
2000  RETURN
100   FORMAT(I4)
      END
```

B.2.20 GR2PLT

This routine plots the data set in the .BIN file FNAM as a 3-dimensional plot on the graph plotter.

This routine calls the routines AVIN, CHRPLT and FILNAM, and the graphics macro P12M.

```
      SUBROUTINE GR2PLT(FNAM,AMAX,AMIN,SCN,NAV,TYPE,IERR)
      INTEGER NY,A(2000),AM(4000),AMAX,AMIN,OLDX,OLDY,XT,AT,NAV
      INTEGER NS,SCN,REC,NA,NSC,B(200),C,IERR
      INTEGER XMAX,YMAX,MARG
      REAL DX,DA,SEP
      LOGICAL*1 PEN,FNAM(15),TYPE(2)
      COMMON /PLT/A,AM
      DATA XMAX,YMAX/4000,2500/
      DATA OLDX,OLDY/0,0/
      IERR=0
        DO 100 I=1,4000
        AM(I)=0
 100    CONTINUE
      PEN=.FALSE.
      IF (TYPE(1).NE.'B') GOTO 110
      OPEN (UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
      1,RECORDSIZE=128)
      READ (98'1) NY,SCN,AMAX,AMIN
      NR=INT(FLOAT(NY-1)/256)+1
C     *******************************************************
110   NA=NY
      NSC=SCN
      IF (TYPE(2).EQ.'R') NA=SCN
      IF (TYPE(2).EQ.'R') NSC=NY
      MARG=YMAX/21
      SEP=FLOAT(MARG*19)/FLOAT(NSC*2)
      DX=FLOAT(XMAX-MARG)-(SEP*FLOAT(NSC))
      IF (TYPE(2).EQ.'R') DX=DX/FLOAT(NA)
      DA=(FLOAT(YMAX-(2*MARG))-(SEP*FLOAT(NSC)))/FLOAT(AMAX)
C     *******************************************************
```

```
          DO 300 NS=1+NAV,NSC-NAV,(2+NAV)+1
          IF (NS.GT.NSC-NAV) GOTO 300
          IF (TYPE(1).EQ.'B') GOTO 120
          CALL FILNAM(FNAM,NS)
          OPEN(UNIT=99,NAME=FNAM,TYPE='OLD',ERR=1100)
          READ(99,10,ERR=1200) NY,(A(I),I=1,NY)
          CLOSE(UNIT=99)
          NA=NY
          GOTO 140
120       CALL AVIN(NS,NY,SCN,2000,A,NAV,TYPE(2),98)
140       IF (NS.EQ.1+NAV) DX=DX/FLOAT(NA)
C         *****************************************************
            DO 130 I=1,NA
            AT=MARG+INT(FLOAT(A(I))*DA)+INT(SEP*FLOAT(NS))
            XT=MARG+INT(FLOAT(I)*DX)+INT(FLOAT(NSC-NS)*SEP)
            IF (I.EQ.1.OR.I.EQ.2) GOTO 330
            IF (PEN.NE..FALSE..OR.AT.LE.AM(XT)) GOTO 320
            IF (OLDY.GE.AM(XT)) GOTO 330
            CALL P12M(XT,AM(XT),OLDX,OLDY,.FALSE.)
            GOTO 330
320         IF (PEN.NE..TRUE..OR.AT.GE.AM(XT)) GOTO 330
            IF (OLDY.LE.AM(XT)) GOTO 330
            CALL P12M(XT,AM(XT),OLDX,OLDY,.TRUE.)
330         IF (AT.LT.AM(XT)) PEN=.FALSE.
            IF (AT.GT.AM(XT)) PEN=.TRUE.
            IF (I.EQ.1) PEN=.FALSE.
            CALL P12M(XT,AT,OLDX,OLDY,PEN)
            IF (AT.LE.AM(XT)) GOTO 130
              DO 310 J=(XT-INT(DX/2)-1),(XT+INT(DX/2)+1)
              IF (AM(J).LT.AT) AM(J)=AT
 310          CONTINUE
 130        CONTINUE
 300      CONTINUE
        IF (TYPE(1).EQ.'B') CLOSE(UNIT=98)
        IF (TYPE(1).NE.'B') CALL FILNAM(FNAM,0)
        CALL CHRPLT(XMAX-500,YMAX-MARG,OLDX,OLDY,15,FNAM)
        IF (TYPE(2).NE.'R') GOTO 2000
        CALL CHRPLT(XMAX-500,YMAX-(MARG*2),OLDX,OLDY,7,'ROTATED')
        GOTO 2000
C       ******************************************************
1100    TYPE *,'ERROR OPENING FILE'
        TYPE *,'EXITING ROUTINE'
        IERR=1
        GOTO 2000
C       ******************************************************
1200    TYPE *,'ERROR READING FILE'
        TYPE +,'EXITING ROUTINE'
        IERR=1
        GOTO 2000
C       ******************************************************
2000    RETURN
10      FORMAT(I5)
        END
```

B.2.21 HEADER

This routine reads in a .DAT header file.

This routine calls the routines CONV, FILNAM and GETNAM.

```
        SUBROUTINE HEADER(FNAM,AMAX,AMIN,SCN,XSTEP,YSTEP)
C       READS AMAX, AMIN AND NUMBER OF SCANS FROM HEADER
        LOGICAL*1 FNAM(15),HT(21,13)
        INTEGER H(21,13),SCN
        INTEGER AMAX,AMIN,XSTEP,YSTEP
C       READ HEADER
        TYPE *,'ENTER INPUT FILE SPECIFICATIONS:'
200     CALL GETNAM(FNAM)
        CALL FILNAM(FNAM,0)
        OPEN(UNIT=99,NAME=FNAM,TYPE='OLD',ERR=1000)
        DO 100 I=1,21
        READ (99,10) (HT(I,J),J=1,13)
 100    CONTINUE
        CLOSE(UNIT=99)
        DO 120 I=1,21
        DO 110 J=1,13
        H(I,J)=HT(I,J)
        IF (HT(I,J).EQ.' ') H(I,J)=13
 110    CONTINUE
 120    CONTINUE
C       FIND AMAX,AMIN AND NO. OF SCANS
        CALL CONV(H,10,AMIN)
        CALL CONV(H,11,AMAX)
        CALL CONV(H,16,SCN)
        CALL CONV(H,15,XSTEP)
        CALL CONV(H,19,YSTEP)
        RETURN
1000    TYPE *,'ERROR OPENING FILE - PLEASE RE-ENTER FILENAME'
        GOTO 200
10      FORMAT(13A)
        END
```

B.2.22 LKBSTS

This routine inquires the status of the LKB densitometer.

This routine calls the graphics macros TPIN and TPOUT.

```
        SUBROUTINE LKBSTS
C       TYPES STATUS OF LKB ON SCREEN
C       INCIDENTALLY, CLEARS "BUFFER" IF FULL!
        INTEGER A,TIM,H,M,S(2),T(2),DT,IERR
        IERR=0
        CALL TPOUT(36)
        CALL GTIM(TIM)
        CALL CVTTIM(TIM,H,M,S(1),T(1))
10      CALL GTIM(TIM)
        CALL CVTTIM(TIM,H,M,S(2),T(2))
        IF (S(2).EQ.S(1)) DT=T(2)-T(1)
        IF (S(2).NE.S(1)) DT=50+T(2)-T(1)
        IF (DT.LT.20) GOTO 10
20      CALL TPIN(A)
        TYPE *,'STATUS OF LKB :'
        IF (A.NE.48) GOTO 30
        TYPE *,'LOG IN MENU'
        GOTO 120
30      IF (A.NE.49) GOTO 40
        TYPE *,'SET UP MENU'
        GOTO 120
40      IF (A.NE.50) GOTO 50
        TYPE *,'MAIN MENU'
        GOTO 120
50      IF (A.NE.51) GOTO 60
        TYPE *,'PLOT MENU'
        GOTO 120
60      IF (A.NE.55) GOTO 70
        TYPE *,'ESCAPE'
        GOTO 120
70      IF (A.NE.56) GOTO 80
        TYPE *,'DEFINE TRACK'
        GOTO 120
80      IF (A.NE.72) GOTO 90
        TYPE *,'HOLD'
        GOTO 120
90      IF (A.NE.80) GOTO 100
        TYPE *,'SENDING DATA'
        GOTO 120
100     IF (A.NE.82) GOTO 110
        TYPE *,'RUNNING'
        GOTO 120
110     IERR=IERR+1
        IF (IERR.LT.2) GOTO 20
120     CONTINUE
        RETURN
        END
```

B.2.23 NEWNAM

This routine accepts the name of a new file, and checksthat it is a valid file name and that no file of that name already exists.

This routine calls the routine BELL.

```
        SUBROUTINE NEWNAM(FNAM,IUNIT)
C       ACCEPTS THE NAME OF A NEW FILE, AND CHECKS THAT IT IS VALID
        LOGICAL*1 FNAM(15)
        INTEGER IUNIT
100     ACCEPT 10,(FNAM(I),I=1,14)
        OPEN(UNIT=IUNIT,NAME=FNAM,TYPE='OLD',ERR=110)
        CLOSE(UNIT=IUNIT)
        GOTO 1000
110     OPEN(UNIT=IUNIT,NAME=FNAM,TYPE='SCRATCH',ERR=1100)
        CLOSE(UNIT=IUNIT)
        GOTO 2000
1000    CALL BELL
        TYPE 20                 ! FILE EXISTS - OVERWRITE ?
        ACCEPT 10,TEMP
        IF (TEMP.EQ.'N') GOTO 100
        GOTO 2000
1100    CALL BELL
        TYPE 30                 ! INVALID FILENAME - RE-ENTER
        GOTO 100
2000    RETURN
10      FORMAT(A:13A)
20      FORMAT(' FILE EXISTS - OVERWRITE (CR=Y) ? ',$)
30      FORMAT(' INVALID FILENAME - RE-ENTER (14A) : ',$)
        END
```

B.2.24 OLDNAM

The routine OLDNAM accepts the name of an existing file from the user, and verifies that the filename is valid and that the file exists.

This routine calls the routine BELL.

```
        SUBROUTINE OLDNAM(FNAM,IUNIT)
C       ACCEPTS THE NAME OF AN EXISTING FILE
        LOGICAL*1 FNAM(15)
        INTEGER IUNIT
100     ACCEPT 10,(FNAM(I),I=1,14)
        OPEN(UNIT=IUNIT,NAME=FNAM,TYPE='OLD',ERR=1000)
        CLOSE(UNIT=IUNIT)
        GOTO 2000
1000    CALL BELL
        TYPE 20         ! ERROR - RE-ENTER
        GOTO 100
2000    RETURN
10      FORMAT(14A)
20      FORMAT(' ERROR OPENING FILE - PLEASE RE-ENJER NAME : ',$)
        END
```

B.2.25 POSN

This routine reads two crosswire positions from the Tektronix screen.

This routine calls the graphics macros DRAWX, MOVEX and RDCW.

```
SUBROUTINE POSN(XT1,XT2,X1,X2,MARG,NY,XMAX)
INTEGER XT1,XT2,X1,X2,MARG,NY,XMAX
TYPE *,'MOVE CROSS-WIRE TO LEFT-HAND SIDE OF PEAK AND PRESS'
TYPE *,'SPACE BAR, THEN DO THE SAME FOR THE RIGHT-HAND SIDE'
CALL RDCW(X1,Y1)
XT1=INT(0.5+FLOAT(X1-MARG)*FLOAT(NY)/FLOAT(XMAX-MARG))
X1=MARG+INT((FLOAT(XT1)*FLOAT(XMAX-MARG))/FLOAT(NY))
CALL MOVEX(X1,0)
CALL DRAWX(X1,750)
TYPE *,'POSITION =',XT1
CALL RDCW(X2,Y2)
XT2=INT(0.5+FLOAT(X2-MARG)*FLOAT(NY)/FLOAT(XMAX-MARG))
X2=MARG+INT((FLOAT(XT2)*FLOAT(XMAX-MARG))/FLOAT(NY))
CALL MOVEX(X2,0)
CALL DRAWX(X2,750)
TYPE *,'POSITION =',XT2
RETURN
END
```

B.2.26 RING

The routine RING returns the intensity of a ring radius R in the data set contained in the file FNAM. It is part of the program LKBRNG.

This routine calls the routine BININ.

```
      SUBROUTINE RING(NB,B,BMAX,BMIN,R,FNAM,NY,SCN,X0,Y0)
C     RETURNS INTENSITY VALUES B OF A RING RADIUS R ABOUT X0,Y0
C
      LOGICAL*1 FNAM(15)
      INTEGER NB,B(NB),BMAX,BMIN,R,NY,SCN,X0,Y0,XT,YT,NA,A(1000)
      REAL THET,X,Y,DX,DY
C
      DATA NA/1000/
C
      OPEN(UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
      1,RECORDSIZE=128)
      BMAX=0
      BMIN=0
        DO 100 I=1,360
        THET=FLOAT(I)
        X=FLOAT(X0)+(FLOAT(R)*SIN(THET*3.1415927/180))
        IF (X.LT.1.0.OR.X.GT.FLOAT(SCN)) GOTO 110
        Y=FLOAT(Y0)+(FLOAT(R)*COS(THET*3.1415927/180.))
        IF (Y.LT.1.0.OR.Y.GT.FLOAT(NY)) GOTO 110
        B(I)=0
        XT=INT(X)
        DX=1.0-X+FLOAT(XT)
        YT=INT(Y)
        DY=Y-FLOAT(YT)
          DO 130 J=1,2
          CALL BININ(XT,NY,NA,A,98)
          B(I)=B(I)+INT(FLOAT(A(YT))*DX*(1.0-DY))
          B(I)=B(I)+INT(FLOAT(A(YT+1))*DX*DY)
          XT=XT+1
          DX=1.0-DX
130       CONTINUE
```

```
        IF (I.NE.1) GOTO 120
        BMAX=B(I)
        BMIN=B(I)
120     IF (BMAX.LT.B(I)) BMAX=B(I)
        IF (BMIN.GT.B(I)) BMIN=B(I)
        GOTO 100
110     B(I)=0
100     CONTINUE
      CLOSE(UNIT=98)
      RETURN
      END
```

B.2.27 RSIZE

The routine RSIZE converts the real number A into the form $A = B \times 10^N$, returning the values B and N.

```
      SUBROUTINE RSIZE(A,B,N)
C     RETURNS SIZE OF A  (A=B*10^N,1<=B<10)
      INTEGER N
      REAL A,B
      B=FLOAT(A)
      N=0
10    IF (B.LT.10.) GOTO 20
      N=N+1
      B=B/10
      GOTO 10
20    IF (B.GT.1) GOTO 2000
      N=N-1
      B=B*10
      GOTO 20
2000  RETURN
      END
```

B.2.28 SIZE

This routine converts the integer A into the form $A = B \times 10^N$, and returns the values B and N.

```
      SUBROUTINE SIZE(A,B,N)
C     RETURNS SIZE OF A (A=B*10^N,1<=B<10)
      INTEGER A,N
      REAL B
      B=FLOAT(A)
      N=0
10    IF (B.LT.10.) GOTO 20
      N=N+1
      B=B/10
      GOTO 10
20    RETURN
      END
```

B.2.29 SPLIN

The routine SPLIN returns $y(x)$ and $\frac{dy}{dx}$ given x and the coefficients of the spline y.

```
C       SPLIN.FOR
C       ROUTINE TO CALCULATE Y(X) FROM THE B-SPLINE REPRESENTATION.
C
        SUBROUTINE SPLIN(NCAP7,C,K,X,Y,DY)
        INTEGER J,NCAP7,NCAPM1,NCAP
        DOUBLE PRECISION X,Y,DY,C(NCAP7),K(NCAP7)
        DOUBLE PRECISION D4,D5,D6,D7,D8,D9,E2,E3,E4,E5,N1,N2,N3
        1,DN1,DN2,DN3,K1,K2,K3,K4,K5,K6,SP(4),DSP(4)
C
        J=0
        NCAP=NCAP7-7
        NCAPM1=NCAP-1
100     IF (X.LT.K(J+4) .OR. J.GT.NCAPM1) GO TO 110
        J=J+1
        GO TO 100
C
110     K1 = K(J+1)
        K2 = K(J+2)
        K3 = K(J+3)
        K4 = K(J+4)
        K5 = K(J+5)
        K6 = K(J+6)
        D4 = 1.0D0/(K4-K1)
        D5 = 1.0D0/(K5-K2)
        D6 = 1.0D0/(K6-K3)
        D7 = 1.0D0/(K4-K2)
        D8 = 1.0D0/(K5-K3)
        D9 = 1.0D0/(K4-K3)
C
        E5=K5-X
        E4=K4-X
        E3=X-K3
        E2=X-K2
```

```
      N1=D9
      N2=E3*N1*D8
      N1=E4*N1*D7
      N3=E3*N2*D6
      DN3=2D0*D6*D8*D9*E3
      N2=(E2*N1+E5*N2)*D5
      DN2=D9*D5*((D7*(E4-E2))+(D8*(E5-E3)))
      DN1=-2D0*N1*D4
      N1=E4*N1*D4
      SP(4)=E3*N3
      SP(3)=E2*N2 + (K6-X)*N3
      SP(2)=(X-K1)*N1 + E5*N2
      SP(1)=E4*N1
      DSP(1)=-3D0*N1
      DSP(2)=N1-N2+((X-K1)*DN1)+((K5-X)*DN2)
      DSP(3)=N2-N3+((X-K2)*DN2)+((K6-X)*DN3)
      DSP(4)=3D0*N3
C
      Y=0D0
      DY=0D0
        DO 120 I=1,4
        Y=Y+(C(I+J-1)*SP(I))
        DY=DY+(C(I+J-1)*DSP(I))
120     CONTINUE
C
      RETURN
      END
```

B.2.30 TKPLT

The routine TKPLT plots out the scan in array A on the tektronix screen.

This routine calls the routine SIZE, and the graphics macros DRAWX, MOVEX, TEKOUT and TEKWIP.

```
      SUBROUTINE TKPLT(NY,NA,A,AMAX,AMIN,XDIV,YDIV)
      INTEGER NY,NA,A(NA),AMAX,AMIN,XT,AT
      INTEGER XMAX,YMAX
      INTEGER N,XDIV,YDIV
      REAL DX,DA,B
      LOGICAL*1 PEN
      DATA XMAX,YMAX/1000,750/
      MARG=30
      IF (AMAX.LE.5) AMAX=20
      DX=FLOAT(XMAX-MARG)/FLOAT(NY)
      DA=FLOAT(YMAX-(2*MARG))/FLOAT(AMAX)
      PEN=.FALSE.
      CALL TEKWIP
      CALL MOVEX(MARG,MARG)
      CALL DRAWX(XMAX,MARG)
C     X-AXIS DRAWN
      CALL SIZE(NY,B,N)
      IF (B.LE.5.) XDIV=(10**N)/2
      IF (B.LE.2.) XDIV=(10**N)/5
      IF (B.GT.5.) XDIV=10**N
        DO 20 I=1,INT(FLOAT(NY)/FLOAT(XDIV))
        XT=MARG+INT(FLOAT(I*XDIV)*DX)
        CALL MOVEX(XT,(MARG-5))
        CALL DRAWX(XT,MARG)
 20     CONTINUE
C     ************************************************************
      CALL MOVEX(MARG,MARG)
      CALL DRAWX(MARG,YMAX)
C     Y-AXIS DRAWN
      CALL SIZE(AMAX,B,N)
      IF (B.LE.5.) YDIV=(10**N)/2
      IF (B.GT.5.) YDIV=10**N
        DO 30 I=1,INT(FLOAT(AMAX)/FLOAT(YDIV))
        AT=MARG+INT(FLOAT(I*YDIV)*DA)
        CALL MOVEX((MARG-5),AT)
        CALL DRAWX(MARG,AT)
 30     CONTINUE
C     ********************************************************
        DO 10 I=1,NY
        IF (A(I).LE.AMAX) GOTO 40
        PEN=.FALSE.
        GOTO 10
40      AT=MARG+INT(DA*FLOAT(A(I)))
        IF (AT.GE.0) GOTO 50
        PEN=.FALSE.
        GOTO 10
50      XT=MARG+INT(DX*FLOAT(I))
        IF (PEN.EQ..TRUE.) CALL DRAWX(XT,AT)
        IF (PEN.EQ..FALSE.) CALL MOVEX(XT,AT)
        PEN=.TRUE.
 10     CONTINUE
      CALL TEKOUT(31)
      RETURN
      END
```

B.2.31 TK2PLT

The routine TK2PLT plolts out the data set in the file FNAM as a 3-dimensional plot on the graphics screen.

This routine calls the routines AVIN and FILNAM, and the graphics macros DRAWX, MOVEX, TEKOUT and TEKWIP.

```
      SUBROUTINE TK2PLT(FNAM,AMAX,AMIN,SCN,NAV,TYPE,IERR)
      INTEGER NY,A(2000),AM(4000),AMAX,AMIN,OLDX,OLDY,XT,AT,NAV
      INTEGER NS,SCN,REC,NA,NSC,B(200),C,IERR
      INTEGER XMAX,YMAX,MARG
      REAL DX,DA,SEP
      LOGICAL*1 PEN,FNAM(15),TYPE(2)
      COMMON /PLT/A,AM
      DATA XMAX,YMAX/1000,750/
      DATA OLDX,OLDY/0,0/
      IERR=0
        DO 100 I=1,4000
        AM(I)=0
 100    CONTINUE
      IF (TYPE(1).NE.'B') GOTO 110
      OPEN (UNIT=98,NAME=FNAM,TYPE='OLD',ACCESS='DIRECT'
      1,RECORDSIZE=128)
      READ (98'1) NY,SCN,AMAX,AMIN
      NR=INT(FLOAT(NY-1)/256)+1
C     ******************************************************
110   NA=NY
      NSC=SCN
      IF (TYPE(2).EQ.'R') NA=SCN
      IF (TYPE(2).EQ.'R') NSC=NY
      MARG=YMAX/20
      SEP=FLOAT(MARG*19)/FLOAT(NSC*2)
      DX=FLOAT(XMAX-MARG)-(SEP*FLOAT(NSC))
      IF (TYPE(2).EQ.'R') DX=DX/FLOAT(NA)
      DA=(FLOAT(YMAX-MARG)-(SEP*FLOAT(NSC)))/FLOAT(AMAX)
C     ******************************************************
      CALL TEKWIP
```

```
C        ******************************************************
          DO 300 NS=1+NAV,NSC-NAV,(2*NAV)+1
          IF (NS.GT.NSC-NAV) GOTO 300
          IF (TYPE(1).EQ.'B') GOTO 120
          CALL FILNAM(FNAM,NS)
          OPEN(UNIT=99,NAME=FNAM,TYPE='OLD',ERR=1100)
          READ(99,10,ERR=1200) NY,(A(I),I=1,NY)
          CLOSE(UNIT=99)
          NA=NY
          GOTO 140
120       CALL AVIN(NS,NY,SCN,2000,A,NAV,TYPE(2),98)
140       IF (NS.EQ.1+NAV) DX=DX/FLOAT(NA)
C         ******************************************************
            DO 130 I=1,NA
            AT=MARG+INT(FLOAT(A(I))*DA)+INT(SEP*FLOAT(NS))
            XT=MARG+INT(DX*FLOAT(I))+INT(FLOAT(NSC-NS)*SEP)
            IF (I.EQ.1.OR.I.EQ.2) GOTO 330
            IF (PEN.NE..FALSE..OR.AT.LE.AM(XT)) GOTO 320
            IF (OLDY.GE.AM(XT)) GOTO 330
            CALL MOVEX(XT,AM(XT))
            GOTO 330
320         IF (PEN.NE..TRUE..OR.AT.GE.AM(XT)) GOTO 330
            IF (OLDY.LE.AM(XT)) GOTO 330
            CALL DRAWX(XT,AM(XT))
330         IF (AT.LT.AM(XT)) PEN=.FALSE.
            IF (AT.GT.AM(XT)) PEN=.TRUE.
            IF (I.EQ.1) PEN=.FALSE.
            IF (PEN.EQ..TRUE.) CALL DRAWX(XT,AT)
            IF (PEN.EQ..FALSE.) CALL MOVEX(XT,AT)
            OLDY=AT
            OLDX=XT
            IF (AT.LE.AM(XT)) GOTO 130
              DO 310 J=(XT-INT(DX/2)-1),(XT+INT(DX/2)+1)
              IF (AM(J).LT.AT) AM(J)=AT
310           CONTINUE
130         CONTINUE
```

```
 300      CONTINUE
        IF.(TYPE(1).EQ.'B') CLOSE(UNIT=98)
        CALL TEKOUT(31)
        GOTO 2000
C       ******************************************************
1100    TYPE *,'ERROR OPENING FILE'
        TYPE *,'EXITING ROUTINE'
        IERR=1
        GOTO 2000
C       ******************************************************
1200    TYPE *,'ERROR READING FILE'
        TYPE *,'EXITING RQUTINE'
        IERR=1
        GOTO 2000
C       ******************************************************
2000    RETURN
10      FORMAT(I5)
        END
```