# *X-ray double crystal characterisation of epitaxial layers*

Martin John Hill

# X-RAY DOUBLE CRYSTAL CHARACTERISATION OF EPITAXIAL LAYERS

MARTIN JOHN HILL, B.Sc.

Thesis submitted to the University of Durham in Candidature for the Degree of Doctor of Philosophy (November, 1985).

# ABSTRACT

Double crystal x-ray diffractometry is a well established method for the measurement of the lattice parameter difference between expitaxial layers and substrates. The diffracted intensity profile versus angle, rocking curves, are highly sensitive to such variations giving rise to complex peak shapes. Consequently, computer simulation is required to enable complete interpretation of the measured data.

A detailed description of a computer simulation technique suitable for calculating rocking curves from arbitary III-V structures, based on the Takagi-Taupin equations for dynamical diffraction from a non uniform crystal, is presented.

Radiation from synchrotron and laboratory sources has been used to measure rocking curves from single uniform and single graded layers of (Ga,In)(As,P) on (001) InP substrates and artificial superlattices of both (Ga,AL)As on (001) GaAs and (Ga,In)As on (001) InP. Excellent agreement has been obtained between computed and experimental curves for all types of structure, enabling the layer thicknesses and compositions to be determined to within 0.1μm and 10 ppm respectively. For single layers less than 0.5 μm thick highly asymmetric reflections are shown to give greatly increased diffracted intensities from the layer, enabling more accurate interpretation. There will always be doubt as to the validity of the lattice parameter profile deduced for a graded layer from a single curve. Rocking curves at various wavelengths, using synchotron radiation, have been used to confirm the profile determined previously using a single measurement from a laboratory source. For superlattices, the dynamical theory approach permits

satellite peaks, allowing the individual layer thicknesses and compositions to be determined in addition to the repeat period. Further, this dynamical approach is particularly suitable for calculating the complete rocking curve where thick confining layers are present and is also directly applicable to multiple layers with varying layer thicknesses.

# ACKNOWLEDGEMENTS

Financial support from the Science and Engineering Research Council and British Telecommunications plc through a CASE award is gratefully acknowledged. I would also like to thank the SERC for making available the facilities of the S.R.S., Daresbury Laboratory.

It is my pleasure to acknowledge the help and assistance from the following people during the course of this work.

Firstly, I would like to thank Professor A.W. Wolfendale and Professor B.H. Bransden for making available the facilities of the Department of Physics at the University of Durham.

I am grateful to Dr. B.K. Tanner for his supervision of this project and his continued enthusiasm. Thanks are also due to Mrs. M.A.G. Halliwell of British Telecom Research Laboratories for initiating this project. I am also grateful to her colleagues; Dr. M.H. Lyons for his thoughtful suggestions, Dr. M.R. Taylor and Dr. M. Hockly for the transmission electron microscopy results; Mr. D.A. Andrews for the samples provided.

I also thank the technical staff of the Department of Physics, especially Mr. D. Josling, Mr. T. Jackson, Mr. P. Armstrong and Mr. G. Teasdale for their help and advice.

I am particularly grateful to Mr. S.J. Barnett for his help in performing many of the experiments at the S.R.S., to Dr. D.Y. Parpia and Dr. M. Surowiec for their useful discussions.

Finally, I would like to thank my parents for their assistance, encouragement and support.

# LIST OF PUBLICATIONS

HALLIWELL, M.A.G., LYONS, M.H. and HILL, M.J. (1984) J. Cryst. Growth, **68**, 523-531.

HILL, M.J., TANNER, B.K. and HALLIWELL, M.A.G. (1984) Proc. Mater. Res. Soc. November Meet. Boston, USA. Layered Structures, Epitaxy and Interfaces II, Vol **37**, 53-58.

HILL, M.J., TANNER, B.K., HALLIWELL, M.A.G. and LYONS, M.H. (1985) J. Appl. Cryst. **18** (in press).

TANNER, B.K., BARNETT, S.J. and HILL, M.J. (1985) Microscopy of Semiconductor Materials, Oxford 1985 (in press).

TANNER, B.K. and HILL, M.J. (1985) Proc. Denver Conf. Applications of X-ray Analysis, Denver, USA. To be published in: Advances in X-ray Analysis Vol. 29 (1986).

# CONTENTS

# CHAPTER 1

# INTRODUCTION

## 1.1 III-V Optoelectronic Devices and Superlattices

### 1.1.1 Optoelectronic Devices

Currently the majority of fibre optic based telecommunications systems being installed are based on graded index fibre, operating with gallium arsenide lasers or light emitting diodes (l.e.d.s) as the light sources. Operation is at the near infra red, between 0.8 and 0.9 μm, with silicon based detectors. Speeds between 8 and 140 Mbit/s are possible over lengths up to about 8 km without repeaters. Recently improved manufacture has lead to fibres with superior attenuation characteristics further into the infra red, in the 1.2 to 1.6 μm region. Therefore, new materials are required to produce the optoelectronic devices operating at these wavelengths, with the most promising being based on gallium indium arsenide (GaInAs) and gallium indium arsenide phosphide (GaInAsP), epitaxially grown on indium phosphide (InP) substrates. Utilising graded index fibre, systems operating at 1.3 μm are capable of speeds between 8 and 140 Mbit/s with lengths of 10 to 20 km between repeaters.

More recently, attention has been focussed on monomode fibres, in which the central core is so small (5 - 10 μm) that only one mode of propagation is possible. This eliminates the modal spread that limits the length between repeaters of graded index, multimode fibre. Close to 1.3 μm

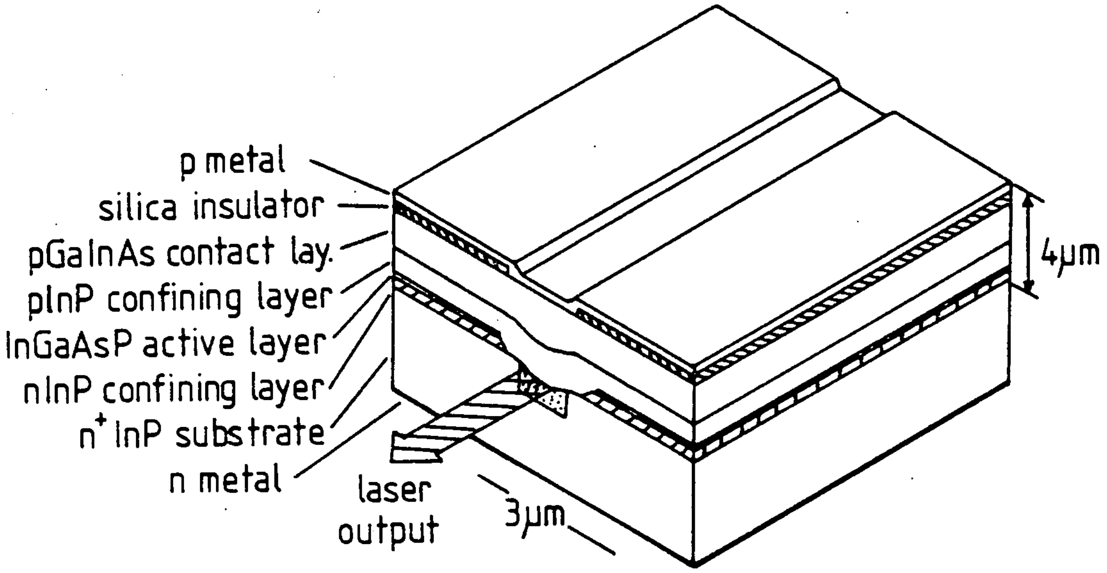


Fig. 1.1 Schematic diagram of a (Ga,In)(As,P) channel substrate buried heterostructure laser (c.s.b.h.)

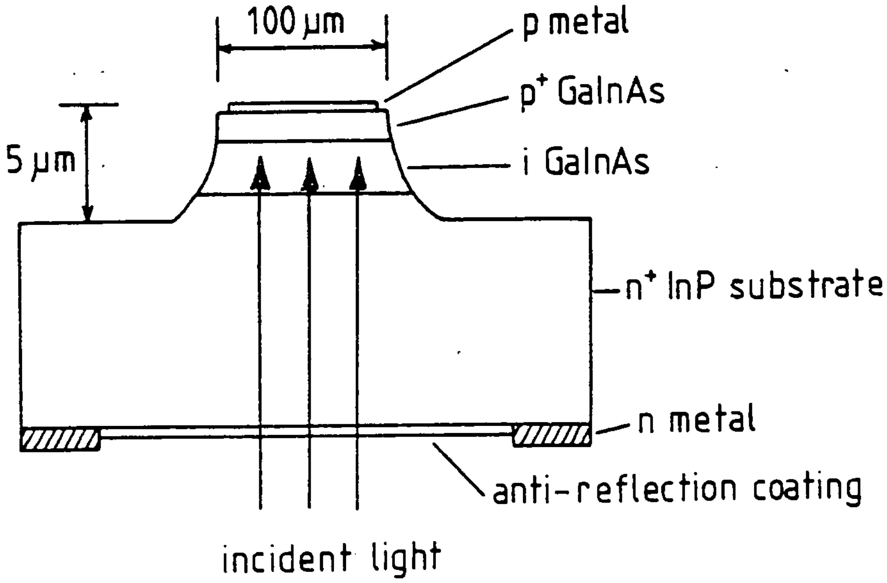


Fig. 1.3 Schematic diagram of a GaInAs p.i.n. detector as used in fibre optic systems operating in the range 1.3-1.55 µm.

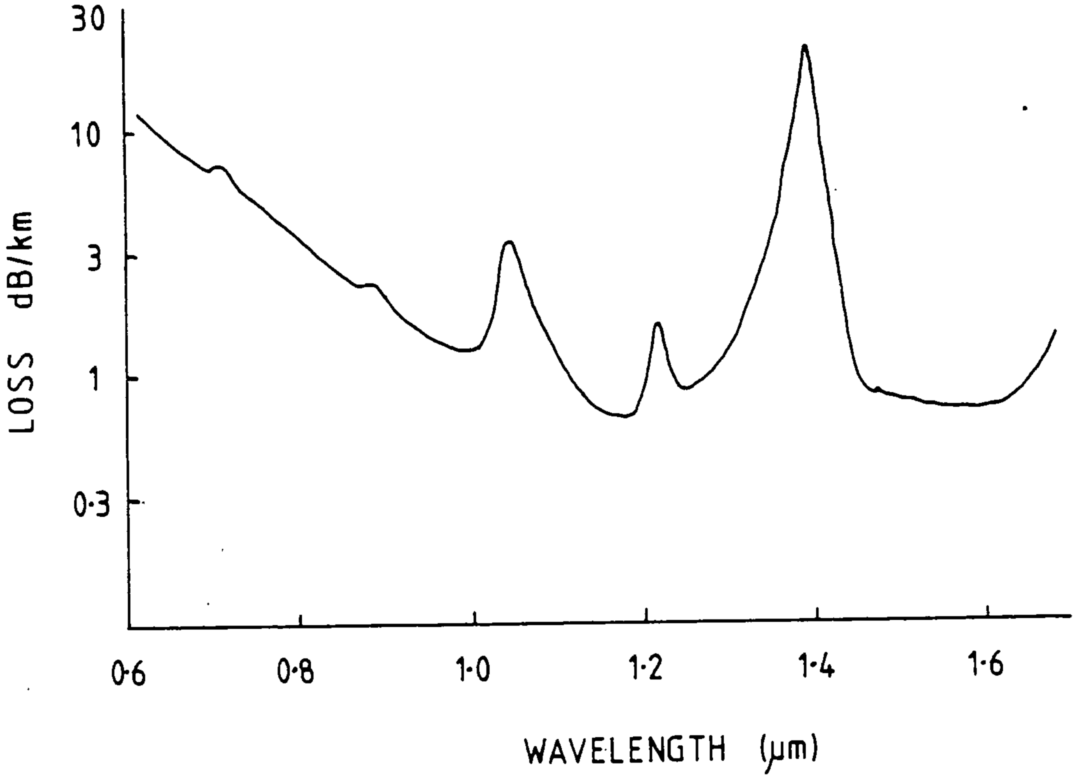


Fig. 1.2 Attenuation of silica based optical fibre as a function of wavelength in the infra red region.

the fibre chromatic dispersion is a minimum, so that even with a source of non zero spectral width there is no first order limitation on bandwidth. Systems have been demonstrated at 140 Mbit/s over lengths of 64 km and at 650 Mbit/s over 32 km without repeaters. Launching the light into such a small region of fibre is a considerable problem, hence the need for laser light sources as opposed to l.e.d.s.. The double heterostructure laser, and its close derivatives, have been developed to provide such sources. The schematic diagram of a channel substrate buried heterostructure laser (c.s.b.h.) is shown in fig. 1.1.

The attenuation of silica based optical fibres reaches an absolute minimum of about 0.15 dB/km at 1.55 μm, but chromatic dispersion is present at this wavelength (Ainslie et al, 1979). Fig. 1.2 shows this loss as a function of wavelength. Therefore, in order to develop longer lengths of fibre between repeaters the chromatic dispersion must be reduced. This can be achieved by both moving the zero dispersion point in the fibre towards 1.55 μm and limiting the spectral width of the laser sources.

Clearly, to make full use of the low attenuation of the fibre the highest possible sensitivity is required at the detector. Silicon avalanche photodiodes (a.p.d.s) are the standard at 0.85 μm, but they are insensitive in the 1.2 - 1.6 μm range. Germanium a.p.d.s have been developed but have signal to noise ratios inferior to the silicon a.p.d.s. At present the major effort is in developing GaInAs type non avalanching detectors (p.i.n. type) and GaInAsP based avalanching detectors. The schematic diagram of a GaInAs p.i.n. detector is shown in fig. 1.3. Light is incident through the transparent InP substrate and is almost all absorbed in the i-GaInAs layer, with nearly 100% quantum efficiency.

### 1.1.2 Superlattices

Artificial semiconductor superlattices are composed of a periodic sequence of thin (50 - 200 Å) layers of alternating composition (eg AlAs/GaAs, GaAlAs/GaAs, GaInAs/InP etc). Interest in such structures originates from the work of Esaki and Tsu (1970) to produce a one dimensional periodic potential by a periodic variation of either impurities or alloy composition in semiconductors, with the period shorter than the electron mean free path. They first considered the compositional superlattice and predicted that they should exhibit unusual transport phenomena. These peculiarities originate from the splitting of the conduction and valence bands into narrow mini or sub bands. This splitting is a consequence of the reduction of the original Brillouin zone due to the strongly increased (superlattice) periodicity. Also in the early 1970's Dohler (1972a,b) analysed in great detail the electronic properties of doping superlattices. In addition to the phenomena characteristic of compositional superlattices, a number of new peculiarities which are specific to doping superlattices were predicted. These unique features arise from the unusual indirect energy gap in real space of the layered material.

The first artificial superlattices were reported by Esaki, Chang and Tsu (1971), where they were obtained by periodic variation of the phosphorous content in layers of $GaAs_{1-x}P_x$ grown by vapour phase epitaxy. The first superlattices grown by molecular beam epitaxy were reported by Chang et al (1973). This prototype structure consisted of several hundred alternating layers of GaAs and $Ga_{1-x}Al_xAs$, grown on a GaAs substrate. Periods ranging from 50 to 200 Å were grown with various compositions of GaAlAs. Early experiments on these structures were used to investigate transport anomalies (Esaki and Chang, 1974) and to demonstrate the quantum mechanical particle in a box behaviour of electrons (Dingle, 1975). The introduction of modulation doping in GaAlAs/GaAs superlattices (Dingle et al, 1978) subsequently modified the established square well concept, and it extended the underlying physics to totally new fundamental phenomena including the quantised Hall effect (von Klitzing et al,1981), near zero resistance state (Tsui et al, 1982a) and electron localization in a two dimensional electron gas in strong magnetic fields (Tsui et al, 1982b; Ebert et al, 1982).

The so called type II compositional superlattice was conceived by Sai-Halasz et al (1977) and exemplified by the material combination of $Ga_{1-x}In_xAs$ and $GaAs_ySb_{1-y}$ (Sakaki et al,1977). The type II superlattice differs from its type I counterpart by the sign of the band edge discontinuity between the two components of the superlattice. While in the type I superlattice the conduction and valence band edge discontinuities have opposite signs, the type II system is characterised by a band edge modulation which has the same sign for the conduction and valence bands. In the case of small x and y values (and in the pure binary system), the bottom of the conduction band of GaInAs becomes even lower in energy than the top of the valence band of GaAsSb. This coexistence of conduction and valence band states in the respective constituent layers can lead to an electron transfer from the GaAsSb to the GaInAs layers, if the layers are sufficiently thin, resulting in a semiconductor-semimetal transition (Sai-Halasz et al 1978 ).

Significant improvements in the spatial control of dopant incorporation during molecular beam epitaxial growth was required before the first doping superlattices could be realised (Ploog et al, 1981a). Many of the electronic peculiarities of GaAs superlattices predicted by Dohler (1972a,b) have since been demonstrated by detailed experiments on the tunability of photoluminescence (Dohler et al,1981 ; Jung et al, 1982), tunability of electroluminescence (Kunzel et al, 1982 ) and on two dimensional carrier confinement and tunability of subband spacing (Dohler et al, 1981 ; Zeller et al, 1982; Maan et al, 1983). These measurements provided the experimental evidence that GaAs doping superlattices indeed formed a new class of semiconductor materials. The electronic properties of a doping superlattice are no longer fixed material parameters but tunable quantities.

Recently, improvements in growth techniques have also enabled the growth of superlattices from constituent materials with large differences in lattice parameter (Matthews and Blakeslee, 1977). Such 'strained layer' superlattices (SLS's) were theoretically predicted to have electrical properties that depended on the state of strain of the individual layers as well as on the composition modulation (Osbourn, 1982). These superlattices are grown by alternately depositing lattice mismatched materials so that strain is necessary to form a coherent interface, but where the layers are below the critical thickness for dislocation generation. A buffer layer of graded composition is also grown between the substrate and the first layer which tends to force any grown-in dislocations towards the edge of the sample, thus enabling relatively low quality substrates to be used. The metallo organic chemical vapour deposition (MOCVD) growth technique is often used to grow such structures. The intentionally built-in strain and freedom to combine lattice mismatched semiconductors has spurred interest in these structures. GaAsP/GaAs SLS's have recently been shown to possess an independently variable band gap and lattice constant (Osbourn et al, 1982; Biefield et al, 1983), and have also exhibited stimulated emission (Ludowise et al, 1983). Quantum size effects observed as optical transitions in both photoluminescence and excitation spectroscopy have also been observed (Gourley and Biefield, 1984).

## 1.2 Growth Techniques for III-V Epitaxy

At present there are four widely used epitaxial growth techniques, namely liquid techniques (liquid phase epitaxy), vapour transport techniques (vapour phase epitaxy, metallo organic chemical vapour deposition) and molecular transport techniques (molecular beam epitaxy). These four techniques vary considerably in operation and complexity, as well as being suited to growing layers for different applications.

### 1.2.1 Liquid phase epitaxy (LPE)

LPE is particularly suited to growing relatively thick layers (2 - 10 μm) of high crystalline quality. A two piece graphite sliding boat is used to hold the substrate while the liquified material for the layer is held in the top, sliding, part. This is heated in a furnace to the required temperature (about 660 ℃ for GaInAs layers) and the molten material slowly slid across the surface of the substrate. A step or ramp cooling technique is used to promote crystal growth, while the furnace is filled with $PH_3$ and $H_2$, the $PH_3$ preventing substrate degradation through loss of phosphorous.

It has been reported (Stringfellow, 1972) that lattice matched layers of GaInP were grown preferentially to lattice mismatched layers, due to the higher free energy associated with a lattice mismatched layer. However, Hseih et al (1977) were unable to find any evidence of this effect for the growth of GaInAs on InP substrates. In particular, the lattice parameter of the layer was found to be a sensitive function of the growth solution composition, the supercooling time and the time of contact between the solution and substrate.

### 1.2.2 Vapour phase epitaxy (VPE)

Vapour phase epitaxy has been used extensively to grow ternary layers of (Ga,In)As, although it is also suited to the growth of quaternary and higher alloys. Binary compounds of Ga and In are transported in the vapour phase over the heated substrate along with As gas, leading to growth at the substrate surface according to the equilibrium conditions. Ga and In are transported as metal chlorides by passing hydrogen over the heated metal source, according to

$$GaAs(s) + HCl(g) \rightarrow GaCl(g) + \tfrac{1}{4}As_4(g) + \tfrac{1}{2}H_2(g) \qquad (1.1)$$

$$InAs(s) + HCl(g) \rightarrow InCl(g) + \tfrac{1}{4}As_4(g) + \tfrac{1}{2}H_2(g) \qquad (1.2)$$

The Ga and In are usually combined in a mixed metal alloy source. The ternary composition is then a function of the ratio of halide pressures and the ratio of equilibrium constants, which is a function of temperature only. Arsenic is introduced into the system by the decomposition of arsenic trichloride,

$$AsCl_3 + \frac{3}{2}H_2 \rightarrow \frac{1}{4}As_4 + 3HCl \qquad (1.3)$$

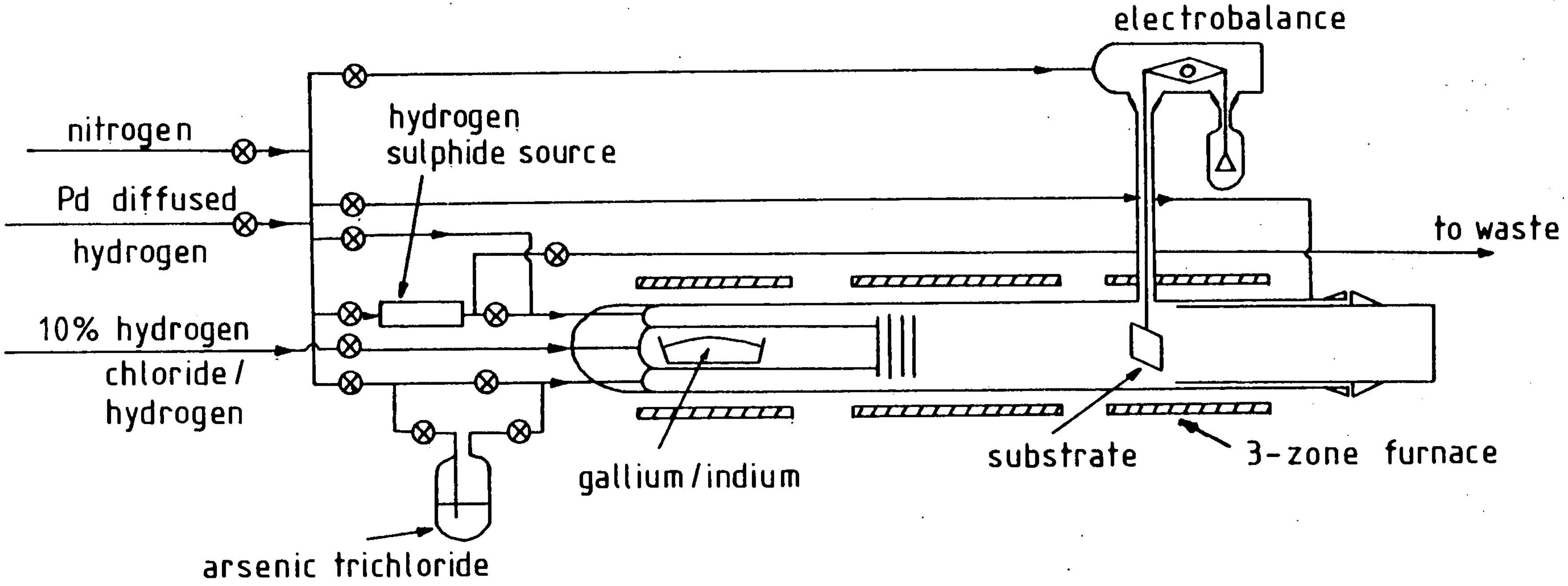


Fig. 1.4 A typical vapour phase epitaxial growth apparatus as used for the growth of GaInAs layers on InP substrates.

Since the equilibrium constants for the interaction of In and Ga with HCl are different Ga is transported preferentially. As the required ratio of Ga to In in the metal source is only about 6% the Ga concentration is reduced as a function of growth time, leading to an increasingly In rich layer (Chaterjee et al,1982). This effect can be minimised by using an overlarge melt. Hydrogen sulphide is used as a dopant and a diagram of a typical growth apparatus is shown in fig. 1.4. Growth temperatures are typically 630 - 660 ℃ giving growth rates in the range 1.5 to 2.0 µm/hour, the amount of material grown can be continuously monitored with an electrobalance.

### 1.2.3 Metallo organic chemical vapour deposition (MOCVD)

MOCVD is a similar technique to VPE but the gases used are of different composition. Growth of GaInAs takes place by introducing metered amounts of TEIn (triethylindium), TEGa (triethylgallium) and $AsH_3$ into a quartz reaction chamber containing the RF heated substrate, placed on a susceptor surface. The hot susceptor has a catalytic effect on the decomposition of the gaseous products, and growth primarily occurs at this surface. The growth rate is proportional to the flow rate of the group III species, but is independent of temperature and also of the partial pressure of $AsH_3$. Growth occurs according to the following scheme

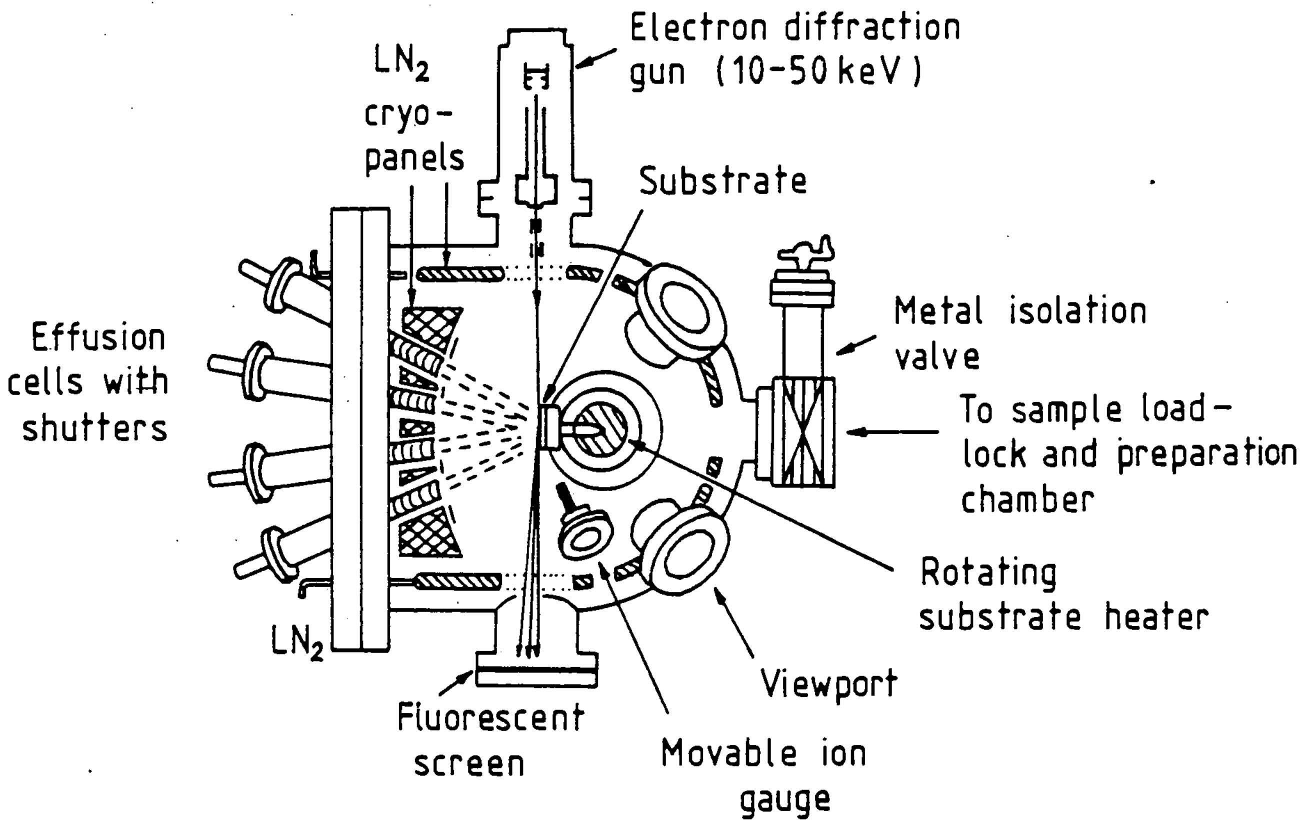

Electron diffraction gun (10-50 keV)
$LN_2$ cryo-panels
Substrate
Metal isolation valve
Effusion cells with shutters
To sample load-lock and preparation chamber
Rotating substrate heater
$LN_2$
Viewport
Fluorescent screen
Movable ion gauge

Fig. 1.5 A typical MBE growth apparatus as used for the growth of (Ga,Al)As layers on GaAs substrates.

$$x * Ga(C_2H_5)_3 + (1-x) * In(C_2H_5)_3 + AsH_3 \rightarrow Ga_xIn_{1-x}As + 3C_2H_6 \qquad (1.4)$$

Typically growth temperatures are ∿ 550 ℃ and with the usual gas flow rates produces growth at the order of 300 Å/min. The growth temperature is less than that used in VPE systems which leads to an improvement in the sharpness of the interfaces due to reduced interdiffusion.

Many of the recently developed systems have computer controlled mass flow valves to control the growth rate, thus enabling the growth of superlattices, as an alternative to the MBE technique. In particular MOCVD is better suited to the growth of the (Ga,In)As alloys than is MBE.

### 1.2.4 Molecular beam epitaxy (MBE)

Molecular beam epitaxy is the most flexible technique available for the growth of artificial superlattices, particularly those based on (Ga,Al)As. This method is based on the reaction of thermal beams of atoms or molecules directed onto a heated substrate under ultra high vacuum conditions. This technique enables control of composition, thickness and doping level down to the atomic scale. Detailed reviews have been presented by Cho and Arthur (1975), Ploog (1980,1981,1982), Chang (1980a) and Foxon and Joyce (1981). Since the growth is a kinetic and not an equilibrium process abrupt changes in composition can be achieved by stopping the beam of the species required. A typical growth apparatus is shown in fig. 1.5. The effusion cells are individually shuttered, operated by vacuum feedthroughs, and may additionally be computer controlled for the accurate growth of repeat layer sequences. Low growth rates of about 1 μm/hour are usual with growth temperatures of 530 - 630 ℃ for GaAs and 630 - 670 ℃ for GaAlAs. The use of $LN_2$ cooled cryopanels for pumping of the condensible residual gas species is required for the growth of high quality GaAlAs (Jung et al, 1983 ).

Since the growth takes place inside a UHV system it is ideal for in situ surface studies such as Auger electron spectroscopy, reflection high energy electron diffraction (RHEED) and secondary ion mass spectroscopy (SIMS), although the addition of these techniques adds significantly to the cost of the system.

## 1.3 Lattice Mismatch and Tetragonal Distortion of the Epitaxial Layer

Since ternary alloys of III-V elements are only lattice matched to binary III-V substrates at one composition any deviation from this matched value will lead to misfit stress occurring during growth. Clearly, any lattice parameter measurements performed will be on this final structure and it is, therefore, important to be able to relate these values to the initial compositions. This can only be performed if we understand how the lattice is deformed during growth. Since the stresses associated with the interface and the epilayer are directly related to the generation and propagation of dislocations we must include the effect of misfit dislocations in any prediction of the misfit stress. Ternary alloys are clearly restricted in their compositional range if large values of misfit stress are not to occur, thus making the alteration of the band gap rather difficult. For this reson the extra degree of freedom provided by quaternary alloys has greatly increased their popularity. The original composition of the quaternary cannot be determined from a lattice parameter measurement alone, the measurement of the band gap is usually used to provide the additional parameters required.

Misfit stress has been quantitatively described without the inclusion of the effects of misfit dislocations by a large number of workers (Jesser and Kuhlman-Wilsdorf, 1978; Hornstra and Bartels, 1978; Bartels and Nijman, 1978; Matsui et al, 1979). Their predictions have been compared with experimental observations for dislocation free samples and found to be in good agreement.

The stress distribution inside the heteroepitaxial layer has been used to determine the radius of curvature of the crystal by a number of authors. A simple beam bending theory for single layers has been developed by Timoshenko (1925) based on the theory of a bimetallic strip. However, inconsistencies in the theory have been reported for it is not internally consistent when the two phase composite strip is reduced to a single phase strip. The bending moment for such a two phase composite strip has been derived properly

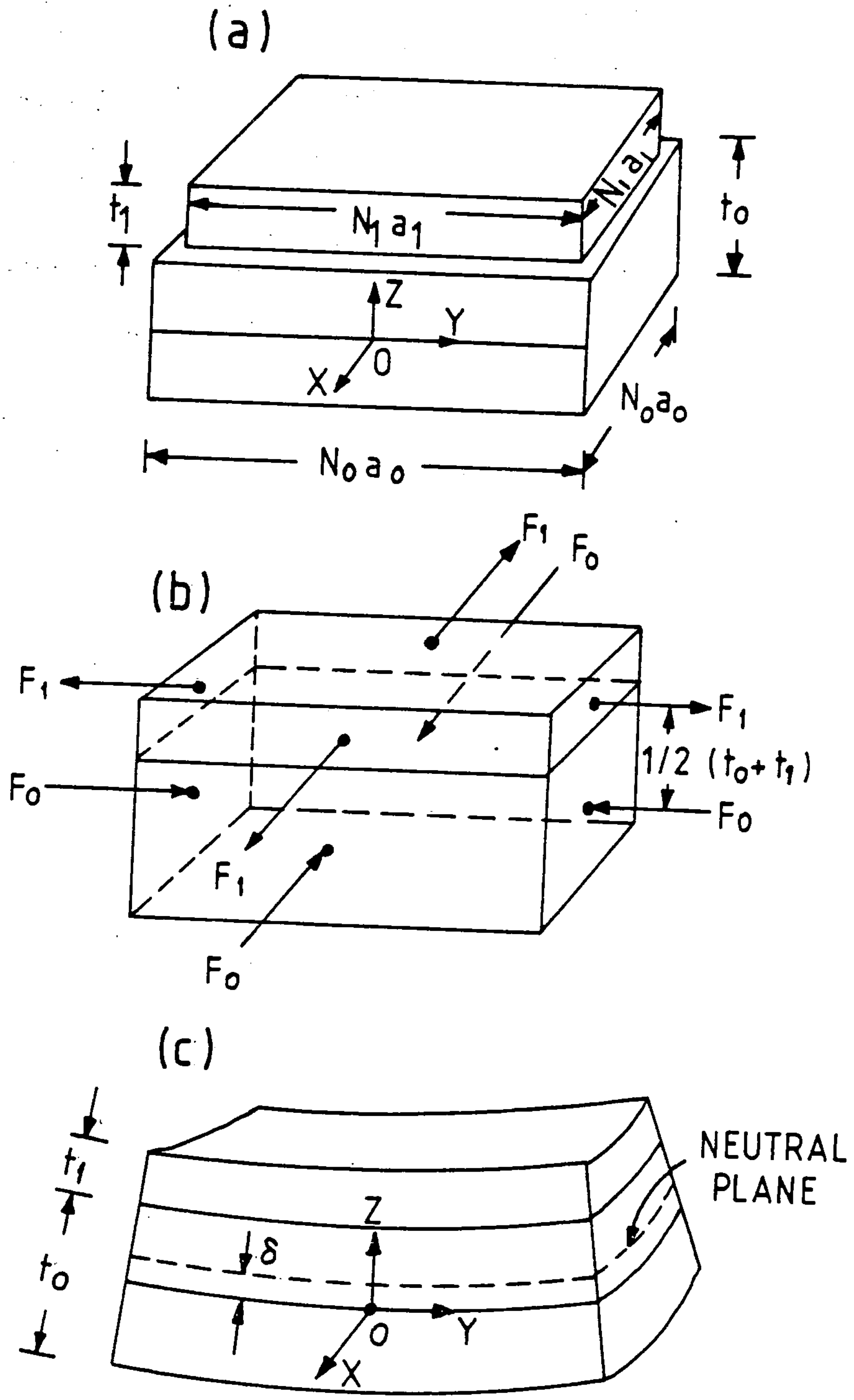


Fig. 1.6 Formation of a two layer composite (a) two single crystal plates with lattice constants $a_1$ and $a_0$ number of atoms along the edge $N_1$ and $N_0$, and thicknesses $t_1$ and $t_0$ respectively; (b) Plate 1 is stretched and plate 0 is compressed to match the macroscopic dimensions, the two plates are then bonded to form a composite; (c) the composite bends towards the side with shorter lattice constant after the removal of the external stresses.

by Davidenkov (1961). Chu et al (1985) have since used this approach to analyse in detail this situation with the inclusion of the effect of interfacial misfit dislocations. The misfit stress in arbitrary multiple layer systems wihout the effect of misfit dislocations has been determined by Vilms and Kerps (1982) and Olsen and Ettenberg (1977), while Chu et al (1985) have extended their approach to cover these structures.

### 1.3.1 Misfit stress in a single epitaxial layer-substrate composite

Following Chu et al (1985) we consider the effect of bonding a thin plate of isotropic single crystal of size $N_1a_1$ x $N_1a_1$ x $t_1$ onto a substrate of size $N_0a_0$ x $N_0a_0$ x $t_0$ ($N_i$ is the number of unit cells along an edge, $a_i$ the lattice parameter and $t_i$ the thickness of the plate), as shown in fig. 1.6. For a coherent interface, with no misfit dislocations, $N_1$ must become equal to $N_0$ during epitaxy. We can initially assume that $a_1$ is less than $a_0$, without any loss of generality, which gives $l_1=N_0a_1$ less than $l_0=N_0a_0$. The bonding process is then achieved by applying equal and opposite forces, F, to stretch plate 1 and compress plate 0, uniformly in the lateral direction, to the final dimension $l_f$ x $l_f$. The two plates can then be bonded together with perfect alignment of the atomic planes. At the moment the two plates are bonded the composite experiences an applied bending moment $F(t_0+t_1)/2$, which is counterbalanced by the moment resulting from the internal elastic stress. Finally the applied forces are removed and the moments from the elastic stress bend the composite in the direction shown in fig. 1.6. This bending relaxes some of the stress with the final radius of curvature being determined by the final state of the internal stress.

In order to include the effect of a partially coherent interface (ie one containing misfit dislocations) we introduce an effective lattice constant. If we consider the bonding of two plates where $a_1$ is less than $a_0$, $N_1$ greater than $N_0$ and $l_1=N_1a_1$ smaller than $l_0=N_0a_0$ when the macroscopic dimensions are matched an extra number of atomic planes $N_1-N_0$ exist in plate 1 and are terminated at the interface. These then become the interfacial misfit dislocations. If the Burgers vector along the interface is $b_I$ the total amount of mismatch taken up by the misfit dislocations in plate 1 is $(N_1-N_0)b_I$. This corresponds to $(N_1-N_0)b_I/N_0$ per bonded atom. An effective lattice parameter $a_1'$ can then be defined as

$$a_1' = a_1 + \frac{N_1-N_0}{N_0} b_I \qquad (1.5)$$

If we let $\rho$ be the linear density of misfit dislocations given by $\rho=(N_1-N_0)/N_0a_0$ the effective lattice becomes

$$a_1' = a_1 + a_0\rho b_I \qquad (1.6)$$

### 1.3.2 Stress distribution

The balance of forces acting on the sample gives

$$F_1 + F_0 = 0 \tag{1.7}$$

while balancing the moments acting on the sample gives

$$F_1 \frac{t_0+t_1}{2} + F_0 \frac{t_0}{2} = M_0 + M_1 \tag{1.8}$$

where the forces are taken to act at the centre plane of each layer.

$M_0$ and $M_1$ are given by

$$M_0 = \frac{E_0}{1-\nu} \frac{1}{R} \int_{-t_0/2}^{t_0/2} (z-\delta)^2 l_f dz \tag{1.9}$$

$$M_1 = \frac{E_1}{1-\nu} \frac{1}{R} \int_{t_0/2}^{t_0/2+t_1} (z-\delta)^2 l_f dz \tag{1.10}$$

where R is the radius of curvature, $\nu$ is Poisson's ratio (which is assumed equal for the two plates), and $\delta$ is the shift of the neutral axis from $t_0/2$, as in fig. 1.6. As given by Davidenkov (1961)

$$\delta = \frac{E_0}{E_1} \frac{t_1}{t_0} \frac{(1 + t_1/t_0)}{(1 + E_0 t_1/E_1 t_0)} \tag{1.11}$$

Thus,
$$M_0 = \frac{E_0 l_f}{(1-\nu)R} \left( \frac{t_0^3}{12} + t_0 \delta^2 \right) = \frac{E_0 I_0}{(1-\nu)R} \tag{1.12}$$

$$M_1 = \frac{E_1 l_f}{(1-\nu)R} \left[ \frac{t_1^3}{12} + t_1 \left( \delta - \frac{t_1+t_0}{2} \right)^2 \right] = \frac{E_1 I_1}{(1-\nu)R} \tag{1.13}$$

where $I_i$, i=0 and 1, is defined as the moment of inertia associated with each plate of the composite. We note that for the case $E_1/E_0 \approx 1$ and $t_1/t_0 << 1$, and keeping only the lowest order terms in $t_1/t_0$ these become

$$\delta = \frac{t_1}{2} \tag{1.14}$$

$$I_0 = \frac{l_f t_0^3}{12} \tag{1.15}$$

$$I_1 = l_f \left( \frac{t_1^3}{12} + \frac{t_1 t_0^2}{4} \right) \approx l_f \frac{t_1 t_0^2}{4} \tag{1.16}$$

Now the strain in any layer is given by the sum of the strain due to the force $F_i$ and the strain due to the bending, $(z-\frac{1}{2}t_i)/R$, where z is the distance in the ith layer from the bottom of the ith layer.

Clearly the atomic separation in each layer must match at the interface for coherency, thus

$$a_1[\ 1 + \frac{F_1}{l_f t_1}\frac{(1-\nu)}{E_1} - \frac{t_1}{2R}\ ] + \rho b_I a_0 = a_0[\ 1 + \frac{F_0}{l_f t_0}\frac{(1-\nu)}{E_0} + \frac{t_0}{2R}\ ] \tag{1.17}$$

Now, since from force equilibrium we have $F_1 = -F_0$ this becomes

$$a_1 - \frac{a_1 F_0 (1-\nu)}{l_f t_1 E_1} - \frac{a_1 t_1}{2R} + \rho b_I a_0 = a_0 + \frac{a_0 F_0 (1-\nu)}{l_f t_0 E_0} + \frac{a_0 t_0}{2R} \tag{1.18}$$

ie $$\frac{-F_0(1-\nu)}{l_f t_1 E_1}\,\frac{a_1}{a_0}\,[\,1 + \frac{a_0 E_1 t_1}{a_1 E_0 t_0}\,] = \frac{a_0 - a_1}{a_0} - \rho b_I + \frac{t_0}{2R}\; 1 + \frac{a_1 t_1}{a_0 t_0} \qquad (1.19)$$

ie $$F_0 = -\,\frac{a_0}{a_1}\,\frac{l_f E_1 t_1}{1-\nu}\,\frac{1}{1 + a_0/a_1\;E_1/E_0\;t_1/t_0}$$

$$\times\left[\frac{a_0 - a_1}{a_0} - \rho b_I + \frac{t_0}{2R}\left(1 + \frac{a_1 t_1}{a_0 t_0}\right)\right] \qquad (1.20)$$

The stress at any point in each layer is given by

$$\sigma^i_{xx} = \frac{F_i}{l_f t_i} + \frac{E_i}{(1-\nu)}\,\frac{(z_i - t_i/2)}{R} \qquad (1.21)$$

where $z_i$, $i=0$ and 1, is measured from the bottom of the ith layer.

We then obtain

$$\sigma^0_{xx} = -\,\frac{E_1}{(1-\nu)}\left[\frac{a_0}{a_1}\,\frac{t_1}{t_0}\,\frac{1}{1 + a_0/a_1\;E_1/E_0\;t_1/t_0}\right.$$

$$\times\left\{\frac{a_0 - a_1}{a_0} + \frac{t_0}{2R}\left(1 + \frac{a_1 t_1}{a_0 t_0}\right) - \rho b_I\right\}$$

$$\left. + \;\frac{E_0}{E_1}\,\frac{(z_0 - t_0/2)}{R}\right] \qquad (1.22)$$

and

$$\sigma'_{xx} = -\frac{E_1}{(1-\nu)}\left[\frac{a_0}{a_1}\;\frac{1}{1 + a_0/a_1\; E_1/E_0\; t_1/t_0} \times\left\{\frac{a_0-a_1}{a_0} + \frac{t_0}{2R}\left(1 + \frac{a_1t_1}{a_0t_0}\right) - \rho b_I\right\} + \frac{(z_0 - t_0/2)}{R}\right] \tag{1.23}$$

We note that the above equations differ slightly from those given by Chu et al (1985) which are dimensionally incorrect.

Clearly the other stress components will be

$$\sigma^i_{yy} = \sigma^i_{xx} \tag{1.24}$$

$$\sigma^i_{zz} = \sigma^i_{xy} = \sigma^i_{xz} = \sigma^i_{yz} = 0 \tag{1.25}$$

for i=0 and 1.

The radius of curvature is given by

$$\frac{1}{R} = \frac{2E_1t_1t_0(1+t_1/t_0)a_0/a_1}{A}\left\{\frac{a_0-a_1}{a_0} - \rho b_I\right\} \tag{1.26}$$

where

$$A = 4\left(1 + \frac{a_0}{a_1}\frac{E_1}{E_0}\frac{t_1}{t_0}\right)\left(\frac{E_0I_0}{l_f} + \frac{E_1I_1}{l_f}\right) - \frac{a_0}{a_1}E_1t_1t_0(t_1 + t_0)\left(1 + \frac{a_1}{a_0}\frac{t_1}{t_0}\right) \tag{1.27}$$

### 1.3.3 Parallel mismatch and interfacial dislocation density

Since the parallel mismatch, $(\Delta a/a)_{11}$, is defined as the lattice mismatch in the plane parallel to the (001) interface, a coherent interface will have $(\Delta a/a)_{11}=0$. The parallel mismatch can only be non zero in the presence of misfit dislocations. Now $a_1=a_1+a_0\rho b_I$, so if we stress the lattice such that $a_1$ and $a_0$ match the final value $a_f$ the lattice constant $a_1$ is changed to $a_{1f}$ and we have

$$a_f = a_{1f} + a_f \rho b_I \qquad (1.28)$$

Therefore,
$$\left(\frac{\Delta a}{a}\right)_{11} = \frac{a_f - a_{1f}}{a_f} = \rho b_I \qquad (1.29)$$

We note that with no misfit dislocations $a_1$ and $a_0$ must be strained to the same final value.

### 1.3.4 Tetragonal distortion

Tetragonal distortion occurs because the lattice is free to expand in the z direction, ie perpendicular to the interface. Since $\sigma^{o}_{xx}$ and $\sigma^{\prime}_{xx}$ are opposite in sign the cubic lattices will distort in the opposite sense.

The lattice constants perpendicular to the interface are given by

$$a_1^{\perp} = a_1\left(1 - 2\nu\,\frac{\sigma^{\prime}_{xx}}{E_1}\right) \qquad (1.30)$$

and

$$a_0^{\perp} = a_0\left(1 - 2\nu\,\frac{\sigma^{o}_{xx}}{E_0}\right) \qquad (1.31)$$

where the second terms are $e_{zz}$ due to the Poisson effect. Therefore,

$$\left(\frac{\Delta a}{a}\right)_{\perp} = \frac{a_0^{\perp} - a_1^{\perp}}{a_0} \tag{1.32}$$

and substituting the value for $\sigma_{xx}$ and $\sigma_{xx}$ we obtain

$$\left(\frac{\Delta a}{a}\right)_{\perp} = \left(\frac{\Delta a}{a}\right)_{r} \frac{1 + \nu}{1 - \nu} - \frac{2\nu}{1-\nu} \rho b_I \tag{1.33}$$

Replacing $\rho b_I$ by the parallel mismatch we obtain

$$\left(\frac{\Delta a}{a}\right)_{r} = \frac{1 - \nu}{1 + \nu}\left(\frac{\Delta a}{a}\right)_{\perp} + \frac{2\nu}{1+\nu} \left(\frac{\Delta a}{a}\right)_{\parallel} \tag{1.34}$$

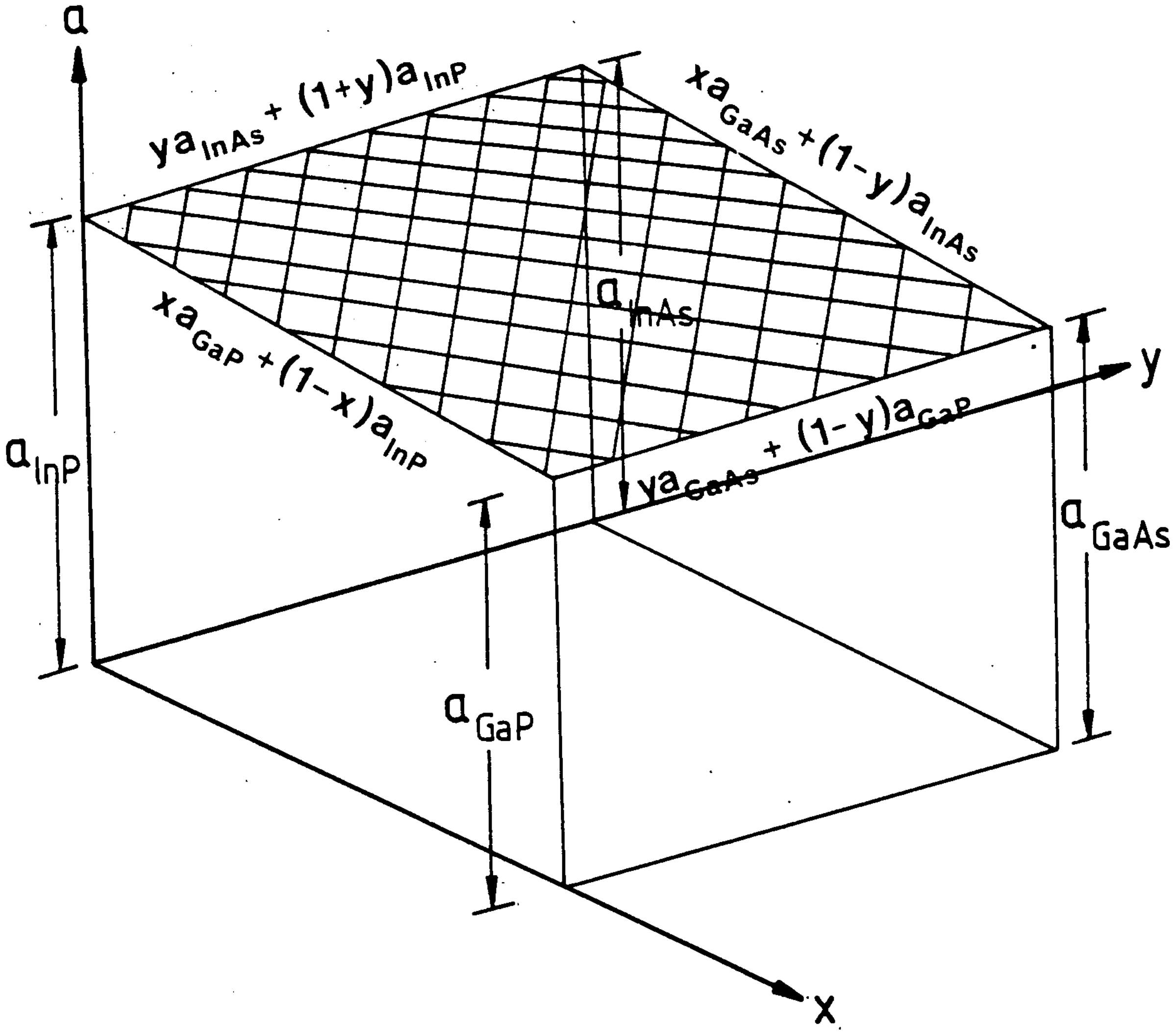
We note that from equation (1.27) that 1/R = 0 if the parallel mismatch is equal to the perpendicular mismatch, ie for a completely incoherent interface. Therefore, coherently matched layers must produce a structure that has a non infinite radius of curvature unless the relaxed lattice mismatch is zero.

This procedure can be extended to multiple layer structures giving, after some work,

$$\left(\frac{\Delta a_i}{a}\right)_{\perp} = \frac{1 + \nu}{1 - \nu} \left(\frac{\Delta a_i}{a}\right)_{r} - \frac{2\nu}{1-\nu} \rho_i b_{Ii} \tag{1.35}$$

for i=1,N where N is the number of layers.

We note that we have assumed isotropic elasticity in deriving the above equations and also that the layer/substrate interface is of the type (001). The reader is referred to the paper of Honstra and Bartels (1978) for results when epitaxy is not on a (001) plane, although



a
$ya_{InAs} + (1+y)a_{InP}$
$xa_{GaAs} + (1-y)a_{InAs}$
$a_{InAs}$
$xa_{GaP} + (1-x)a_{InP}$
$a_{InP}$
y
$ya_{GaAs} + (1-y)a_{GaP}$
$a_{GaAs}$
$a_{GaP}$
x

Fig. 1.7 Surface representing the lattice parameter of $Ga_xIn_{1-x}AsyP_{1-y}$ as a function of x and y.

partial coherency of the interface is not included.

As already mentioned, the relaxed lattice mismatch $(\Delta a/a)_r$ can be directly related to the composition for ternary alloys, using Vegard's law, but not for quaternary or higher alloys. For quaternaries another, independent, measurement relating x and y is required.

## 1.4 Vegard's Law and Band Gaps

Once the relaxed lattice parameter of the layer has been determined the composition of the layer can be found. For ternary alloys the composition is given by Vegard's law, since the lattice parameter is a linear function of the composition. For example for $Ga_xIn_{1-x}As$ we have

$$a(Ga_xIn_{1-x}As) = (1-x)a(InAs) + xa(GaAs) \qquad (1.36)$$

where a(alloy) is the relaxed lattice parameter of the alloy. Clearly, similar equations exist for other ternaries. Thus x can easily be determined.

For quaternaries the variation in lattice parameter is slightly more complicated. Taking $Ga_xIn_{1-x}As_yP_{1-y}$ as an example, the lattice parameter is given by the surface shown in fig. 1.7. Considering the change in lattice parameter parallel to the x and y axes we obtain

$$\frac{\partial a}{\partial x} = ya(GaAs) + (1-y)a(GaP) - ya(InAs) - (1-y)a(InP) \qquad (1.37)$$

and

$$\frac{\partial a}{\partial y} = xa(GaAs) + (1-x)a(InAs) - xa(GaP) - (1-x)a(InP) \quad (1.38)$$

With the limit that at x=0, y=0 a=a(InP) we obtain

$$a(Ga_xIn_{1-x}As_yP_{1-y}) = xya(GaAs) + x(1-y)a(GaP) + (1-x)ya(InAs) + (1-x)(1-y)a(InP) \quad (1.39)$$

Clearly, we cannot determine x and y from this single equation. Empirical relationships between the composition and band gap have been determined by Nahory et al (1978). The variations found to best fit the experimental data are shown in Table 1.1 and the values of the lattice parameters for some of the common III-V binary alloys are shown in Table 1.2.

We note that the equation for the quaternary could be found in a similar manner to the lattice parameter by integrating the partial derivatives found from the ternary variations, with the required equation being that of a surface similar to that in fig. 1.7. This relation would then be

$$Eg(x,y) = 1.35 + 0.668x - 1.17y + 0.758x^2 - 0.18y^2 - 0.069xy - 0.322x^2y + 0.03xy^2 \text{ eV} \quad (1.40)$$

However, Nahory et al (1978) found that the relation given in Table 1.1 gave a better fit to the experimental data.

TABLE 1.1

**Band gaps of some of the common III-V ternary and quaternary compounds.**

| Alloy | Band gap (eV) |
|---|---|
| $In_{1-x}Ga_xAs$ | $Eg(x) = 0.36 + 0.629x + 0.436x^2$ |
| $InAs_yP_{1-y}$ | $Eg(y) = 1.35 - 1.17y + 0.18y^2$ |
| $In_{1-x}Ga_xP$ | $Eg(x) = 1.35 + 0.668x + 0.758x^2$ |
| $GaAs_{1-y}P_y$ | $Eg(y) = 2.77 - 1.56y + 0.21y^2$ |
| $In_{1-x}Ga_xAs_yP_{1-y}$ | $Eg(x,y) = 1.35 - 0.72y + 0.12y^2$ |

TABLE 1.2

**Lattice parameters of some of the more common binary III-V alloys.**

| Alloy | Lattice Parameter (Å) |
|---|---|
| GaAs | 5.6535 |
| InP | 5.8688 |
| InAs | 6.0590 |
| GaP | 5.4512 |

## 1.5 Characterisation of Epitaxial Layers

The main requirements of any characterisation technique are the determination of the layer composition and the layer thickness. If more than one layer is present the individual thicknesses and compositions should also be determined. Also, if a continuous variation in composition is present in a layer this needs to be determined. The crystalline quality (surface morphology and dislocation density) of the layer is also important. A number of techniques are available including transmission electron microscopy (TEM), Auger electron spectroscopy (AES), Rutherford backscattering (RBS), secondary ion mass spectrometry (SIMS), reflection high energy electron diffraction (RHEED) plus many varied X-ray and electrical methods. In particular, the first few techniques require extensive sample preparation and extremely expensive apparatus.

Of these techniques TEM is the most widely used, but since electrons are absorbed by very small amounts of material the samples need to be considerably thinned. For layered structures these thin sections need to be taken perpendicular to the layer interfaces. Since only a small region of the sample can be studied some doubt exists as to it being representative of the bulk sample. Also since some of the strain present in the bulk sample may be relieved during the thinning process errors may be present in any lattice parameter determination. TEM is particularly useful for studying the repeat periodicity and layer thicknesses of superlattices, see for example Petroff et al (1984).

Clearly, since the lattice parameter of a mismatched layer is different from that of the substrate X-ray diffraction should be able to detect this difference through the change in Bragg angle for a given reflection. Advantages of X-ray techniques are that little or no sample preparation is required and that it is non destructive. However, due to the magnitude of the mismatch usually found in III-V materials being of the order of parts in $10^3$ (often less) difficulties arise with single crystal techniques which are limited by the angular width of the reflection, which is determined by the collimation and hence beam divergence being used. Nevertheless, single crystal techniques have been used, but are limited to detecting mismatches down to the order of $5x10^{-4}$ (see for example Isherwood, Brown and Halliwell, 1981; Chang et al, 1979; Chang, 1980b). It has been well known since the 1920's that the double crystal spectrometer can overcome these problems and that the sensitivity to lattice mismatch will then be limited by the dynamical width of the reflections themselves (the order of 10 seconds of arc for the majority of reflections from III-V materials).

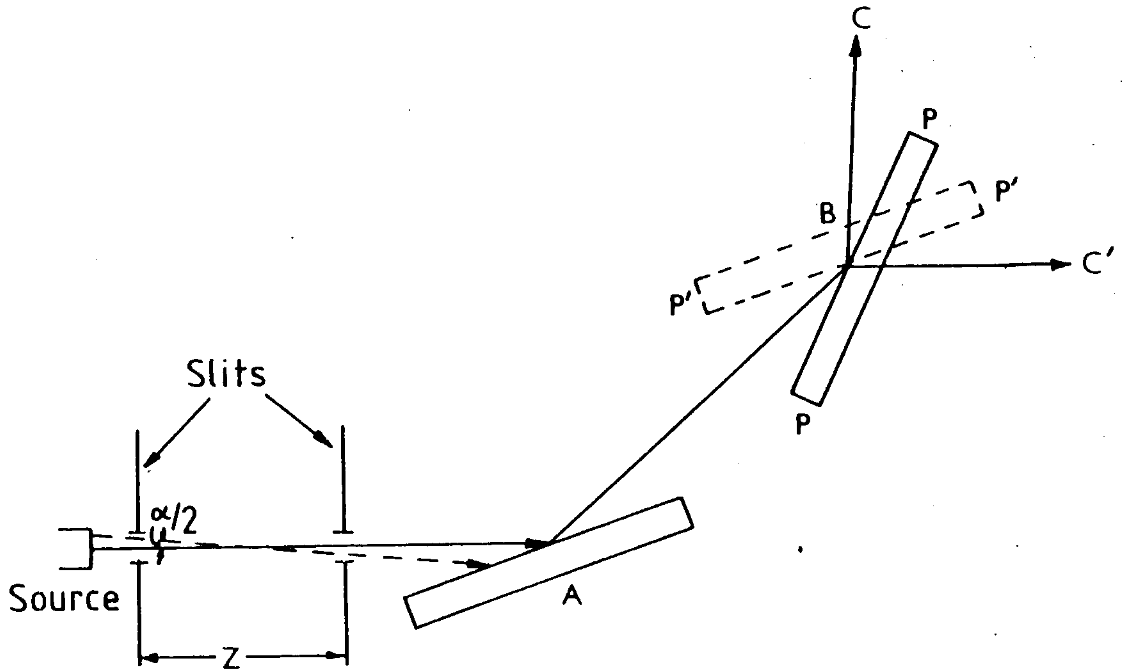


Fig. 2.1 The two [illegible] possible settings of a double crystal spectrometer. The beam incident on crystal A is characterised by two angles of divergence: α - horizontal and ϕ - vertical.

# CHAPTER 2

# DOUBLE CRYSTAL DIFFRACTOMETRY

## 2.1 General Theory of the Double Crystal Camera

The double crystal spectrometer (or diffractometer) has the property of removing the effect of the X-ray linewidth and beam divergence on the reflection profile of a crystal. This effect has been well known since the 1920's, and the reader is referred to the papers of Allison and Williams (1930), Allison (1932) Compton and Allison (1935) and du Mond (1937). Double crystal diffractometers have been in existence since the early 1930's (Compton, 1931; du Mond and Marlow, 1937), but only recently has their use become more widespread with the development of highly perfect semiconductor crystals.

The designs of the double crystal spectrometer, for which the theory was developed, are based on combining two crystals with successive Bragg reflection as shown in fig. 2.1. The X-ray beam passes through two parallel horizontal and vertical slits before being incident on the monochromator crystal A. As a result it is characterised by two values of divergence: $\alpha$ - horizontal divergence and $\varphi$ - vertical divergence. The maximum values of these angles are then given by

$$\alpha_m = c/z, \ \varphi_m = h/z \qquad (2.1)$$

where c is the width and h the height of the slits, with z

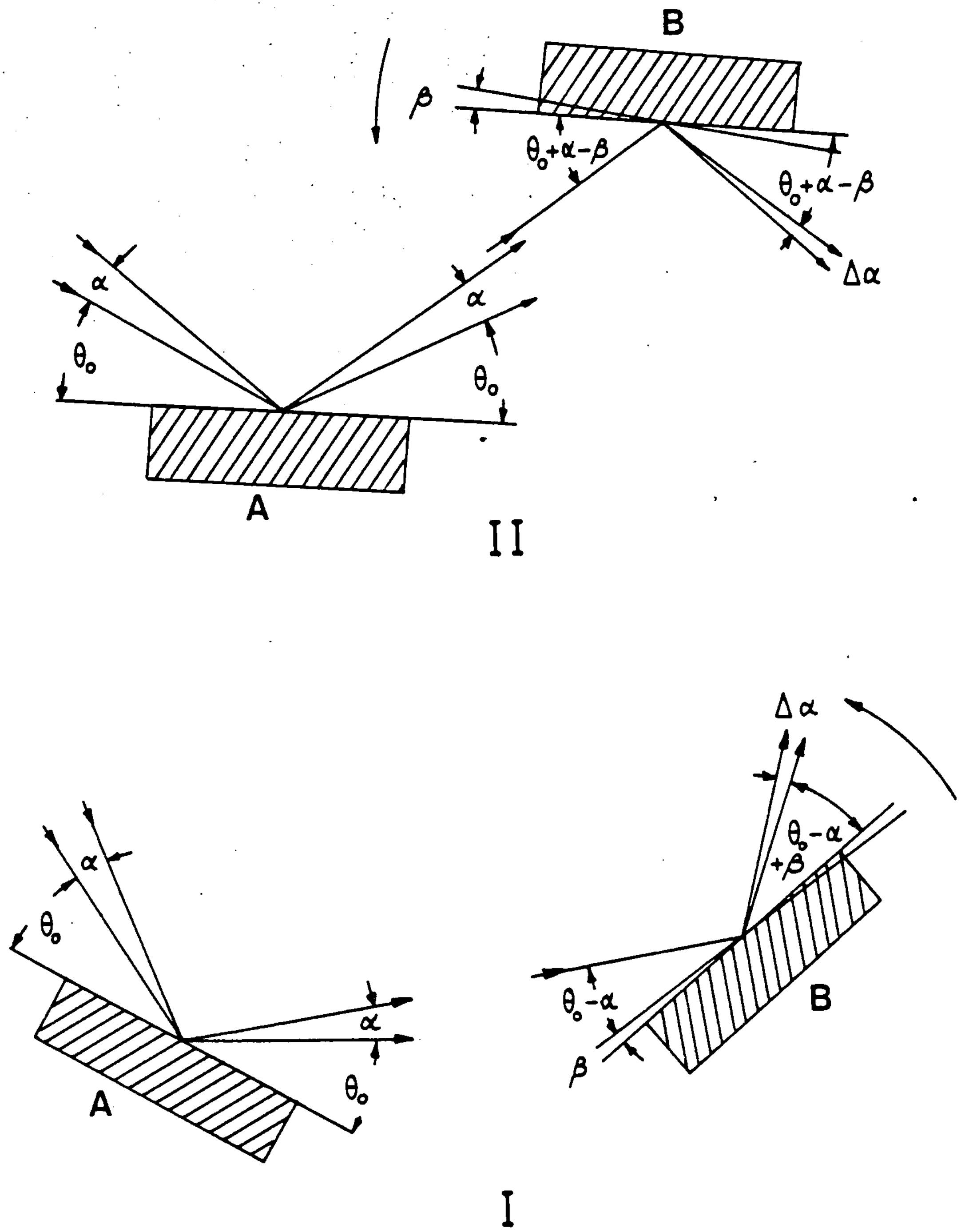


Fig. 2.2 The x-ray paths and angles for the two possible settings of the double crystal spectrometer.

the distance between them. Clearly there are two fundamentally possible settings of the spectrometer as shown in fig. 2.2, with the second crystal B in either the PP or P'P' positions.

The first crystal A is aligned so that a central ray in the incident beam makes an angle $\theta_0 = \theta + \eta_0$ with the reflecting planes, corresponding to the centre of the total reflection region. We note that the value of $\eta_0$ is dependent on the refractive index of the crystal and its value is given in the next chapter. Consequently the maximum of the reflection profile does not correspond exactly with the kinematic Bragg angle θ. Crystal B is then set up in an arbitrary initial position, close to the position of reflection. Rotating crystal B about the vertical axis and recording the reflected intensity as a function of angle gives the rocking curve. Therefore, we need to obtain the relationship between this curve and the true dynamical diffraction curve from crystal B alone. We first consider the values of the angles formed by the various rays in the incident beam with crystals A and B at various wavelengths.

Following Compton and Allison (1935) and Pinsker (1978) the deviation of an arbitrary ray in the incident beam from the central ray is (θ is the glancing angle and $n_A$ the order of the reflection)

$$\alpha - \frac{1}{2}\varphi^2 \tan\theta(\lambda_0, n_A) - (\lambda-\lambda_0)\left(\frac{d\theta}{d\lambda}\right)'_0 \qquad (2.2)$$

where the central ray is characterised by the parameters $\theta_0$,

$\lambda_0$, $\alpha=0$ and $\varphi=0$. The second term corresponds to the deviation due to vertical divergence and the third due to non monochromaticity. The last term takes this form after assuming that within the spectral width of the incident beam the reflection angles change only slightly.

For the second crystal the deviation from the central ray is given by

$$\pm \beta \mp \alpha - \frac{1}{2}\varphi^2 \tan\theta(\lambda_0, n_B) - (\lambda-\lambda_0)\left(\frac{d\theta}{d\lambda}\right)_0^2 \tag{2.3}$$

The upper signs correspond to position I and the lower to position II.

We must now account for the intensity distribution within the original beam in relation to the divergence and wavelength. We assign functions $G(\alpha,\varphi)$, $J(\lambda-\lambda_0)$ to represent these distributions. The functions are normalised so that the intensity within an interval $(d\alpha, d\varphi, d\lambda)$ is determined by multiplying by the interval.

The total power reflected by the second crystal is then

$$\begin{aligned} P(\beta) = & \int_{-\phi_m}^{\phi_m} \int_{\lambda_{min}}^{\lambda_{max}} \int_{-\alpha_m}^{\alpha_m} G(\alpha,\varphi)\, J(\lambda-\lambda_0) \\ & \times C_A\left[\alpha - \frac{1}{2}\varphi^2 \tan\theta_1 - (\lambda-\lambda_0)\left(\frac{d\theta}{d\lambda}\right)_0^1\right] \\ & \times C_B\left[\pm\beta \mp \alpha - \frac{1}{2}\varphi^2 \tan\theta_2 - (\lambda-\lambda_0)\left(\frac{d\theta}{d\lambda}\right)_0^2\right] d\alpha d\varphi d\lambda \end{aligned} \tag{2.4}$$

where $C_A$ and $C_B$ are expressions corresponding to the reflection curves of crystals A and B. The rocking curve is then given by $P(\beta)$ over the range of angles $\beta$ required.

We can study the general properties of this equation by considering $C_A$ and $C_B$ to be non zero only when their arguments are zero (although this is unphysical it is a reasonable approximation). Therefore,

$$\alpha - \frac{1}{2}\varphi^2 \tan\theta(\lambda_0, n_A) - (\lambda-\lambda_0)\left(\frac{d\theta}{d\lambda}\right)_0^1 = 0 \qquad (2.5)$$

and

$$\pm\beta \mp \alpha - \frac{1}{2}\varphi^2 \tan\theta(\lambda_0, n_B) - (\lambda-\lambda_0)\left(\frac{d\theta}{d\lambda}\right)_0^2 = 0 \qquad (2.6)$$

Eliminating $\alpha$ gives

$$\beta - \frac{\varphi^2}{2}\left[\tan\theta(\lambda_0, n_A) \pm \tan\theta(\lambda_0, n_B)\right]$$

$$- (\lambda-\lambda_0)\left[\left(\frac{d\theta}{d\lambda}\right)_0^1 \pm \left(\frac{d\theta}{d\lambda}\right)_0^2\right] = 0 \qquad (2.7)$$

Now defining D as

$$D = \left(\frac{d\theta}{d\lambda}\right)_0^1 \pm \left(\frac{d\theta}{d\lambda}\right)_0^2 \qquad (2.8)$$

which with the Bragg equation gives

$$D = \frac{n_A}{2d\cos\theta(\lambda_0, \eta_A)} \pm \frac{n_B}{2d\cos\theta(\lambda_0, \eta_B)} \qquad (2.9)$$

ie

$$\beta = \frac{1}{2}D\lambda_0\varphi^2 + D(\lambda-\lambda_0) \qquad (2.10)$$

Now the dispersion in the double crystal arrangment is defined as $\frac{d\beta}{d\lambda}$, which gives

$$\text{dispersion} = \frac{d\beta}{d\lambda} = D \qquad (2.11)$$

Hence, for the type II arrangement where two identical reflections are used the dispersion is zero.

We will now consider the (+-) type II arrangement. For two identical crystals we have $C_A=C_B=C$ and $\theta(\lambda_0,n_A) = \theta(\lambda_0,n_B) = \theta$, giving

$$P(\beta) = \int_{-\phi_m}^{\phi_m} \int_{\lambda_{min}}^{\lambda_{max}} \int_{-\alpha_m}^{\alpha_m} G(\alpha,\varphi)\ J(\lambda-\lambda_0) \qquad (2.12)$$

$$\times\ C\ [\alpha - \frac{1}{2}\varphi^2 \tan\theta - (\lambda-\lambda_0)\left(\frac{d\theta}{d\lambda}\right)_0^1]$$

$$\times\ C\ [-\beta + \alpha - \frac{1}{2}\varphi^2 \tan\theta - (\lambda-\lambda_0)\left(\frac{d\theta}{d\lambda}\right)_0^2]\ d\alpha d\varphi d\lambda$$

For nearly perfect crystals C is only non-zero over a range of a few seconds of arc ($10^{-5}$ radians), when we are within the dynamical diffraction width. The function $G(\alpha,\varphi)$ can be written as

$$G(\alpha,\varphi) = G_1(\alpha)G_2(\varphi) \qquad (2.13)$$

Although $G_i$ are different from zero over the range of a few minutes of arc, when the term $\frac{1}{2}\varphi^2\tan\theta$ is considered the effective region of variation of $G_2$ is about the same as function C. It can, however, be shown that the shape of the resultant reflection curve is independent of the vertical divergence of the beam incident on the first crystal in the parallel arrangement (see the next section). For each of the monochromatic components of the incident beam the effective range of the argument is very small as, approximately,

$$\alpha_e \approx (\lambda-\lambda_0)\left(\frac{d\theta}{d\lambda}\right)_o \qquad (2.14)$$

This estimate corresponds to the crystal A transforming the incident beam into a set of parallel beams for the monochromatic components.

The effective range of the wavelengths taking part in the formation of the maximum inside the reflection curve is estimated at

$$\lambda_0 \pm \left(\frac{d\lambda}{d\theta}\right)_o \alpha_m \qquad (2.15)$$

Typically this amounts to a range that considerably exceeds the half width of the spectral lines.

The function P(β) is different from zero over a very small range of the arguments. Thus in the scheme (+n,-n) the half width of the reflection curve is comparable to the half width of the dynamical diffraction peak from one crystal. This value is typically exceeded by a factor of 1.3.

Comparing the narrow range of the functions C, which are taken into account during integration, and the wide ranges of the variables α, φ and λ (the functions $G_1$, $G_2$ and J change gradually over seconds of arc) we have the expression for the reflection curve as

$$P(\beta) = k \int_{-\infty}^{\infty} C(\alpha)\, C(\alpha-\beta)\, d\alpha \qquad (2.16)$$

where k represents the functions $G_1$, $G_2$ and J. Conversion between this curve and the dynamical curve of the crystal has been shown to be not possible by Laue (1931).

We also note that, from inspection of this equation, that the (+n,-n) rocking curve is symmetric about $\alpha=0$ even if the curves C are not symmetrical.

For non polarised radiation, as produced in the characteristic lines of conventional X-ray sources, we must add the contribution from $\sigma$ and $\pi$ polarisations. Thus

$$P(\beta) = k\int_{-\infty}^{\infty}[\ C_\sigma(\alpha)C_\sigma(\alpha-\beta) + C_\pi(\alpha)C_\pi(\alpha-\beta)\ ]\ d\alpha \qquad (2.17)$$

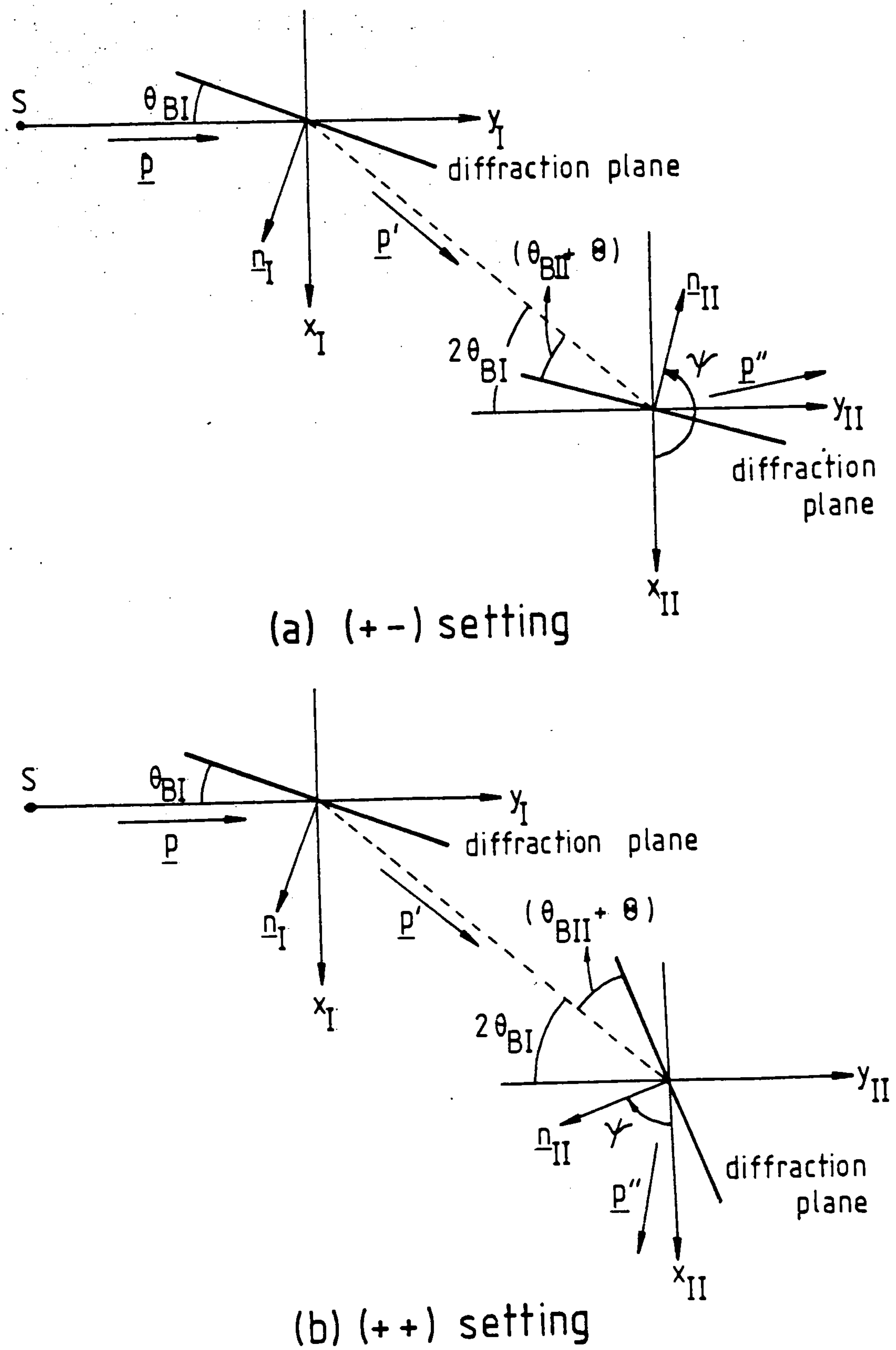
gives the rocking curve for unpolarised radiation.

Often the reflecting power is required and this is defined as

$$R(\beta) = \frac{\int_{-\infty}^{\infty}[C_\sigma(\alpha)C_\sigma(\alpha-\beta) + C_\pi(\alpha)C_\pi(\alpha-\beta)]\ d\alpha}{\int_{-\infty}^{\infty}C_\sigma(\alpha)\ d\alpha \ + \ \int_{-\infty}^{\infty}C_\pi(\alpha)\ d\alpha} \qquad (2.18)$$

## 2.2 The effect of vertical divergence on the rocking curve

The effect of vertical divergence needs to be considered where there is a non zero angle between the plane of the normal of the second crystal and that of the first crystal, ie the second crystal is tilted with respect to the first crystal. This effect has been studied in detail by Schwarzschild (1928), Jager (1965,1966) and more recently by



(a) (+-) setting

(b) (++) setting

Fig. 2.3 The direction vectors and angles of the x-ray beams as required for the effect of vertical divergence.

Yoshimura (1984). The reader is referred to these papers for a complete description, while a summary of the results are presented here.

Following Yoshimura (1984), we can write the diffraction conditions as

$$\underline{p}\cdot\underline{n}_I = -\sin\theta_I;\ \underline{p}'\cdot\underline{n}_I = \sin\theta_I';\tag{2.19}$$

$$[\underline{p}x\underline{n}_I]/\cos\theta_I = [\underline{p}'x\underline{n}_I]/\cos\theta_I'\tag{2.20}$$

for the first crystal, and

$$\underline{p}'\cdot\underline{n}_{II} = -\sin\theta_{II};\ \underline{p}''\cdot\underline{n}_{II} = \sin\theta_{II}';\tag{2.21}$$

$$[\underline{p}'x\underline{n}_{II}]/\cos\theta_{II} = [\underline{p}''x\underline{n}_{II}]/\cos\theta_{II}'\tag{2.22}$$

for the second crystal. The angles $\theta_i$ are given by

$$\theta_I = \theta_{BI} + \eta_I;\ \theta_I' = \theta_{BI} + \eta_I'\tag{2.23}$$

$$\theta_{II} = \theta_{BII} + \eta_{II};\ \theta_{II}' = \theta_{BII} + \eta_{II}'\tag{2.24}$$

where

$$\eta_I = (\Delta\lambda/\lambda)\tan\theta_{BI} + (1-1/b_I)u_I + |1/b_I|^{\frac{1}{2}}v_I W_I\tag{2.25}$$

$$\eta_I' = (\Delta\lambda/\lambda)\tan\theta_{BI} + (1-b_I)u_I + |b_I|^{\frac{1}{2}}v_I W_I\tag{2.26}$$

$$\eta_{II} = (\Delta\lambda/\lambda)\tan\theta_{BII} + (1-1/b_{II})u_{II} + |1/b_{II}|^{\frac{1}{2}}v_{II}W_{II}\tag{2.27}$$

$$\eta_{II}' = (\Delta\lambda/\lambda)\tan\theta_{BII} + (1-b_{II})u_{II} + |b_{II}|^{\frac{1}{2}}v_{II}W_{II}\tag{2.28}$$

The direction of the vectors and crystals are as shown in fig. 2.3. $\underline{p}$, $\underline{p}'$ and $\underline{p}''$ represent the directions of the X-ray beams, $\underline{n}_I$ and $\underline{n}_{II}$ unit normals to the diffracting planes and $\theta_I$, $\theta_I'$, $\theta_{II}$, $\theta_{II}'$ are the angles the X-ray beams form with the diffraction planes for the two crystals. $b_I$ and $b_{II}$ are, as usual, the asymmetry factors, $\gamma_0/\gamma_h$. $\theta_{BI}$ and $\theta_{BII}$ are the kinematical Bragg angles, $u_I$ and $u_{II}$ the deviation due to the refraction effect and $v_I$, $v_{II}$ the half widths of the diffraction peaks. $W_I$ and $W_{II}$ represent the deviation from the exact Bragg angle, ie $-1<W_i<1$ corresponds to the total reflection range. The second crystal is tilted by an angle $\rho$ with respect to the first crystal.

Yoshimura (1984) has shown that the relationship between the angular position of the second crystal $\theta$, such that a beam from the first crystal with parameter $W_I$ is diffracted by the second crystal with parameter $W_{II}$ in the parallel (+-) setting, is

$$\begin{aligned} z_I/f = {} & \sin(\theta_B-\alpha_I)\cos\theta_B\{\rho[(\theta+\Delta\theta_u)\cos\theta_B\sin\alpha_I \\ & - (\eta+\Delta\theta_u)\sin\theta_B\cos\alpha_I] \pm [(\theta+\Delta\theta_u)\sin\theta_B\cos\alpha_I \\ & - (\eta+\Delta\theta_B)\cos\theta_B\sin\alpha_I] \\ & \times [(\theta+\Delta\theta_u)^2+\rho^2-(\eta+\Delta\theta_u)^2]^{\frac{1}{2}}\} \\ & \times \{[(\theta+\Delta\theta_u)\sin\theta_B\cos\alpha_I-(\eta+\Delta\theta_u)\cos\theta_B\sin\alpha_I]^2 \\ & + \rho^2\sin(\theta_B+\alpha_I)\sin(\theta_B-\alpha_I)\}^{-1} \qquad (2.29) \end{aligned}$$

Where the vertical coordinate of the incident beam on the first crystal is $z_I$ and the distance to the X-ray source is

f. $\alpha_I$ is the angle between the diffraction plane and the surface of the first crystal. The ± sign is taken according to $\alpha_I+\theta_B>0$ or $\alpha_I+\theta_B<0$. $\theta_B$ is taken as $\theta_B=\theta_{BI}=\theta_{BII}$ while $\eta$ is defined as

$$\begin{aligned}\eta &= \eta_{II} - \eta_I' + \tfrac{1}{2}\rho^2\tan\theta_B \\ &= [(1-1/b_{II})u_{II} - (1-b_I)u_I] \\ &\quad + (|1/b_{II}|^{\frac{1}{2}}v_{II}W_{II} - |b_I|^{\frac{1}{2}}v_IW_I) \\ &\quad + \Delta\lambda/\lambda_0\,\Delta\theta_u/\cos^2\theta_B + \tfrac{1}{2}\rho^2\tan\theta_B \qquad (2.30)\end{aligned}$$

where $\Delta\theta_u = \theta_{BII} - \theta_{BI}$ accounts for lattice deformation in the second crystal. For the non parallel (+-) setting and the (++) setting the above equation can be replaced by

$$\begin{aligned}z_I/f = \{ &-\rho\cos\theta_{BI} \pm (\cos\theta_{BI}\cos\theta_{BII})^{\frac{1}{2}} \\ &\times [2\sin(\theta_{BII} \mp \theta_{BI})(\theta-\eta) + \rho^2(\cos\theta_{BI}/\cos\theta_{BII})]^{\frac{1}{2}} \} \\ &\times [\sin(\theta_{BII} \mp \theta_{BI})]^{-1} \qquad (2.31)\end{aligned}$$

The upper sign is for the (+-) setting and the lower for the (++) setting. $\eta$ given by

$$\begin{aligned}\eta = &[(1-1/b_{II})u_{II} \mp (1-b_I)u_I] \qquad (2.32) \\ &+ (|1/b_{II}|^{\frac{1}{2}}v_{II}W_{II} \mp |b_I|^{\frac{1}{2}}v_IW_I) \\ &+ \Delta\lambda/\lambda_0(\tan\theta_{BII} \mp \tan\theta_{BI}) + \tfrac{1}{2}\rho^2\tan\theta_{BII}\end{aligned}$$

(signs as before).

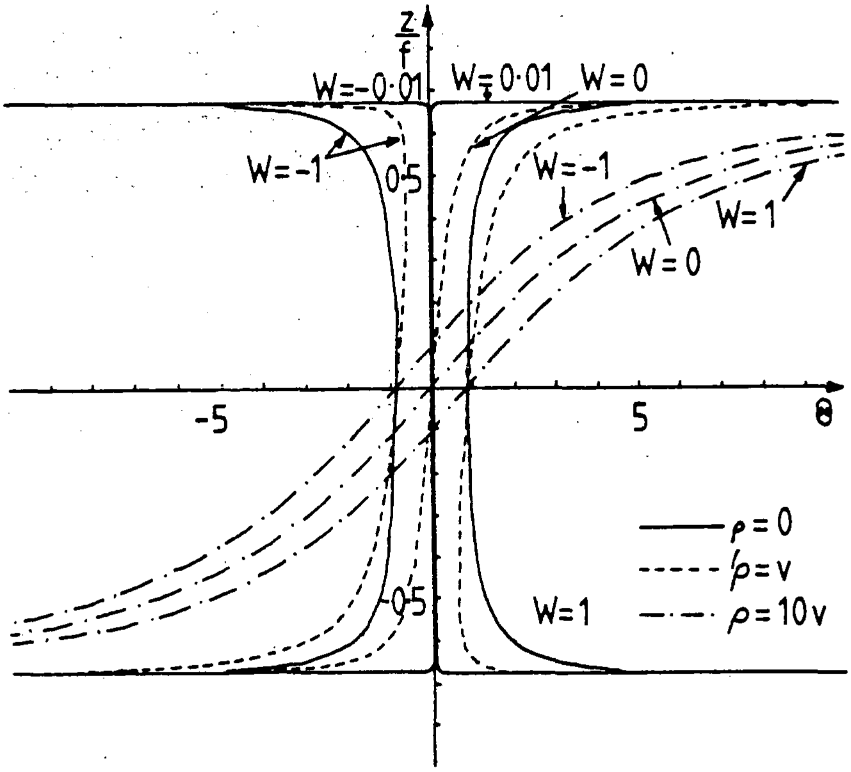


Fig. 2.4 The change in crystal angle (Θ) required to diffract a beam with value W as a function of z/f, the position of the incident beam on the first crystal, for a parallel (+-) setting with two silicon (333) reflections and CuK$\alpha_1$ radiation, for varying values of ρ the tilt angle. v = 1·55".

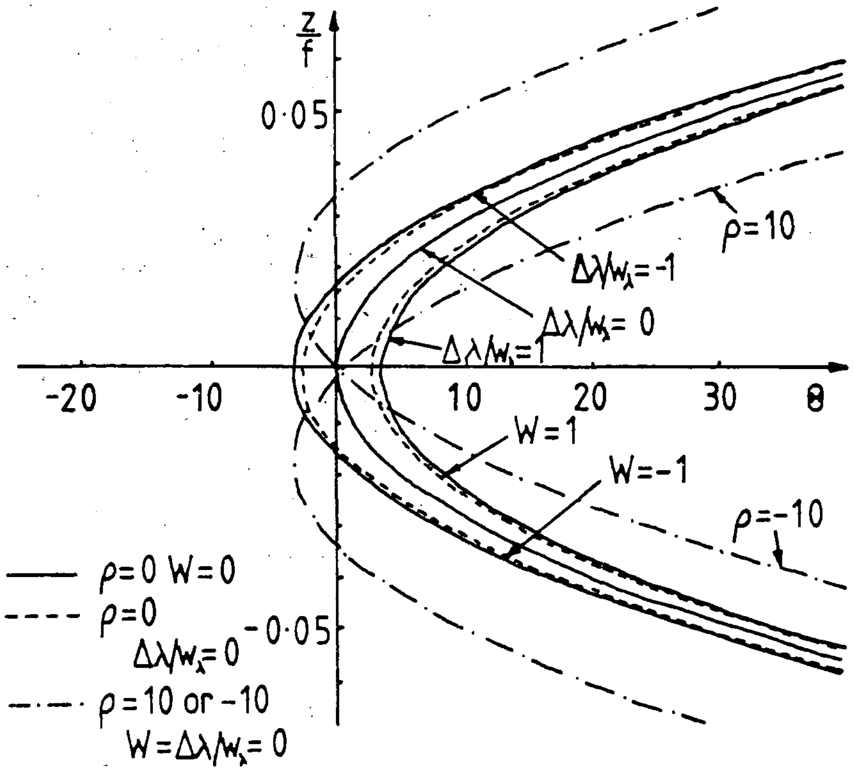


Fig. 2.5 Curve of z/f vs (Θ) for a non parallel (+-) setting with si (111) and si (220) reflections. CuK$\alpha_1$ radiation with the abscissa scaled in units of 1.55 arc secs.

The curve of $z/f$ vs $\theta$ is shown in fig. 2.4 for a parallel (+-) setting with two Si (333) reflections. The rocking curve at a particular height $z/f$ is given by traversing the curve along a line with $y=z/f$ parallel to the $\theta$ axis. We note that for no tilt, ie $\rho=0$, the situation corresponds to the condition where the vertically divergent beam can be diffracted over its entire length with the same intensity. The curve for a non parallel (+-) arrangement, with Si (111) and Si (220) reflections, is shown in fig. 2.5. As can be seen the wavelength and parameter W vary rapidly with $z/f$ along the line $\theta=0$, while the distribution of high diffracted intensity is concentrated over a narrow vertical range.

From equation (2.29), substituting $z/f=0$, we can find the value of $\theta$ at which the $z/f$ curve crosses the $z/f=0$ line as a function of $\rho$, the tilt angle. We obtain

$$\begin{aligned} \theta_0 &= \eta_0 \qquad\qquad (2.33) \\ &= [(1-1/b_{II})u_{II} - (1-b_I)u_I] + wW \\ &\quad + \Delta\lambda/\lambda_0 \, \Delta\theta_u/\cos^2\theta_B + \tfrac{1}{2}\rho^2 \tan\theta_B, \end{aligned}$$

where w is the half width of the integrated reflection curve obtained by integrating the reflection curves over the height of the X-ray beam incident on the first crystal. W is analogous to $W_I$ and $W_{II}$ for this integrated curve. Equation (2.33) is plotted in fig. 2.6. This curve has been used to aid the rapid alignment of the double crystal arrangement to

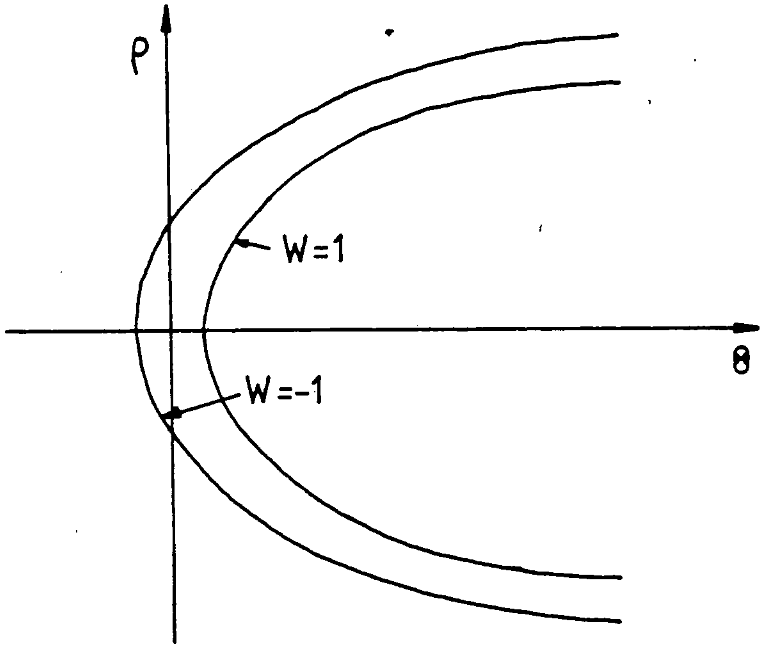


Fig. 2.6 Change in angle Θ required to give the centre of Bragg reflection tilt angle p.

the position $\rho=0$ (see for example Fewster, 1985).

The half width of the rocking curve measured by rotating the second crystal is clearly the half width of this integrated curve, ie w. Yoshimura (1985) has shown that this can be related to the half width at $\rho=0$ by

$$w = \frac{w(z/f=0)}{[1 - (z/f)^2/\cos^2\theta_B]^{\frac{1}{2}}} \tag{2.34}$$

if we assume Gaussian forms for the two reflection curves and that $z/f \ll 1$.

Ignoring the refraction and dynamical effects Schwarzschild (1928) showed that the half width of this curve due to purely geometrical effects could be related to the tilt angle, $\rho$, by

$$w = 2\rho z/f \qquad \text{for } \rho > z/f[\tan\theta_{BI} \pm \tan\theta_{BII}] \tag{2.35}$$

and

$$w = Mz^2/f^2 + \rho z/f + \rho^2/4M \qquad \text{for } \rho < 2Mz/f \tag{2.36}$$

where $M=\frac{1}{2}(\tan\theta_{BII} \pm \tan\theta_{BII})$ with the upper sign for the (++) setting and the lower for the (+-) setting. We note that for the parallel (+-) setting M=0 so that the geometrical half width varies linearly with tilt angle.

## 2.3 Double Crystal X-ray Diffraction Characterisation of Epitaxial and Ion Implanted Layers and Superlattices

Both epitaxial layers and ion implanted layers are well suited for the application of double crystal X-ray diffractometry. In both systems a highly perfect substrate material is present with an added distortion due to either the growth of an epitaxial layer with a slightly different lattice parameter ($\Delta d/d < 10^{-4}$) or ion implantation into the substrate distorting the substrate lattice (again $\Delta d/d < 10^{-4}$). Generally single crystal X-ray diffraction techniques do not have the required resolution to detect such small lattice parameter variations, although several techniques have been used for the measurement of mismatch in heteroepitaxial layers (Chang, Patel, Nannichi and de Prince, 1979; Chang, 1980b, Isherwood, Brown and Halliwell, 1981). The depth penetration of X-rays into typical semiconductor materials and for the usual reflections is the order of 10 μm, which is highly suitable as the majority of epitaxial layers are less than 10 μm.

To date there has been a considerable difference in the interpretation of rocking curves recorded from ion implanted systems and heteroepitaxial layered systems. The interpretation of rocking curves from epitaxial layered systems has been limited to measuring the perpendicular and parallel mismatch from the angular separation of the diffraction peaks, using both surface symmetric and asymmetric reflections. Little use has been made of theoretical simulations to aid the interpretation and to determine more parameters such as layer thicknesses and composition variations with depth. This is not the case, however, for diffusion and ion implanted layers where extensive use of theoretical simulations has enabled the strain distribution as a function of depth to be accurately determined. Clearly, much can be gained by applying these simulations to the interpretation of rocking curves from heteroepitaxial systems.

Much use has been made of X-ray diffraction techniques to analyse the structure of both naturally occurring and artificial superlattices. Naturally occurring superlattices have long been known to produce additional satellite reflections in the region of the main diffraction peaks (Dehlinger, 1927), and similar observations have been made for artificial superlattices. Utilising single crystal techniques the position of these satellites can be measured and used to determine the superlattice period. However, the zeroth order satellite peak cannot be resolved from the main substrate diffraction peak from such measurements, and these peaks are essential in the determination of the composition of the superlattice. Consequently double crystal diffraction has also been used to study these structures and the use of both the kinematical and dynamical diffraction theories for simulation of the rocking curves enables complete characterisation of the superlattice.

### 2.3.1 Diffusion and ion implanted layers

A wide variety of ion and diffusion implanted layers have been studied using double crystal diffractometry. Rocking curves from such structures exhibit highly asymmetrical peaks with additional fine structure and oscillations up to 500-1000 arc seconds away from the main diffraction peak. The strain distribution and hence doping as a function of depth can be obtained from the rocking curve by comparison with theoretically calculated rocking curves for varying distributions. The distribution that gives a theoretical curve that best fits the experimental curve can be taken as the actual distribution.

Alpha irradiated (111) silicon was studied by Burgeat and Colella (1969). 333 rocking curves using MoK$\alpha_1$ radiation were shown to exhibit a secondary peak on the low angle side of the main peak, displaced by 5-10 arc seconds and between 20% and 50% of the intensity of the main peak. Theoretical curves were calculated using the Takagi-Taupin equations for diffraction from a distorted crystal. The irradiated region of the crystal was divided into a number of laminae of constant lattice parameter, in which the Takagi-Taupin equations could be solved analytically to give the complex amplitude ratio of the diffracted and incident beams at the top of the laminae in relation to that at the bottom of the laminae. By matching this amplitude ratio at the interface of each laminae the reflectivity at the surface of the crystal can be calculated, using the reflectivity for an infinitely thick crystal at the bottom of the first laminae. Using a computed profile for the number of displaced Si atoms as a function of depth the theoretical rocking curves were found to agree fairly well with the experimenatal ones. However, no attempt was made to calculate curves with a slightly different strain distribution.

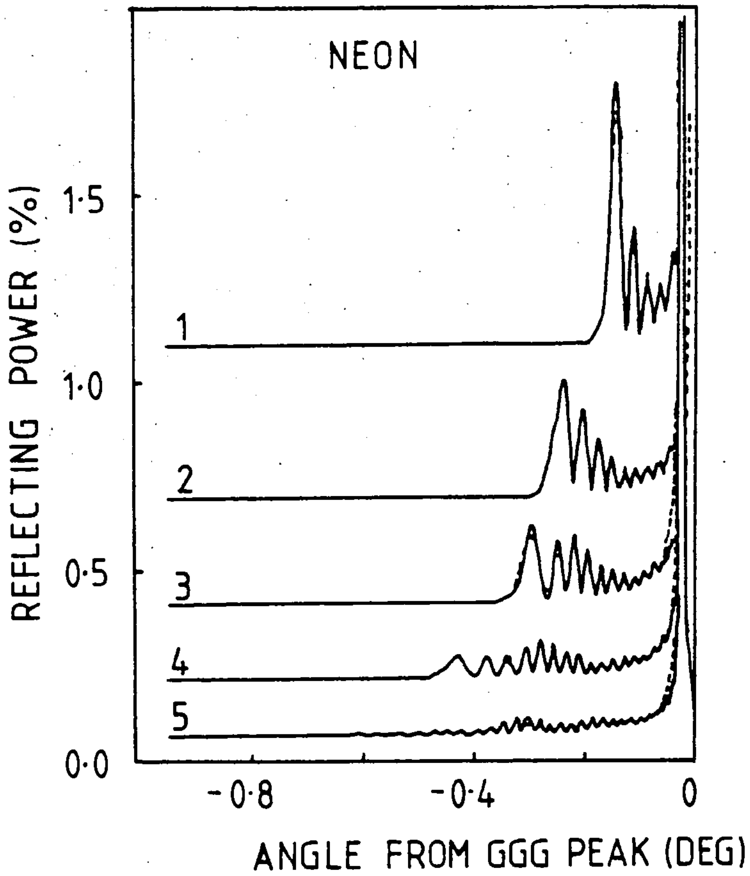
Fukuhara and Takano (1977a,1977b) have similarly applied the dynamical theory approach to the simulation of rocking curves from boron, phosphorous and germanium diffusion doped silicon. Rocking curves from the boron and phosphorous doped silicon showed a broad tail on the high angle side of the peak, suggesting a decrease in the lattice parameter, while the germanium doped samples exhibited considerable detail on the low angle side. By calculating rocking curves for varying strain profiles, profiles were obtained that gave rocking curves that fitted the experimental curves extremely well. Far better agreement was obtained than that obtained by Burgeat and Colella. The deduced strain profiles monotonically decreased from the surface down to 10-15 µm and were found to agree well with those obtained from resistivity measurements.

Ion implantation into silicon gives a much higher strain than that obtained by diffusion and over a much smaller region of the crystal. Consequently rocking curves exhibit non zero diffracted intensity up to the order of 1000 seconds from the main peak. Larson and Barhorst (1980) studied laser annealed boron implanted silicon. Oscillations in the tail of the rocking curves, on the low angle side of the substrate peak, were observed with intensities about 1% of that of the main peak. A similar approach to that of Burgeat and Colella and Fukuhara and Takano was used to calculate rocking curves for various strain distributions. However, the inclusion of strain was slightly different in that a continuous distribution was used and the Takagi-Taupin equations solved numerically. The formalism of Klar and Rustichelli (1973) was used in these calculations. Excellent agreement between the rocking curves was found with the theoretical curve being highly sensitive to the strain profile. Although the magnitude of the strain differed to that obtained by SIMS measurements the shape agreed extremely well.

The effect of increasing Si implantation doses on rocking curves from GaAs, Si and Ge crystals has been studied by Speriosu et al (1982). For small doses Bragg case Pendellosung fringes were observed but the amplitude of these oscillations decreased rapidly as the dose was increased. This was expected as the Pendellosung fringes only occur in thin, perfect crystals and the increased dosage introduces large amounts of damage in the crystal. The kinematical theory of X-ray diffraction was used to calculate rocking curves, using an approach based on the previous work of Speriosu (1981). In a similar manner to the dynamical calculations of Fukuhara and Takano, the strained region was divided into a number of laminae of constant strain. Dynamical interactions among the beams from each laminae were ignored. The total diffracted amplitude is then given by the sum of the coherently interfering amplitudes from each laminae, adjusted for phase lags and normal absorption. The theoretical curves were found to agree extremely well with the experimental ones, corresponding to Gaussian strain distributions with widths of typically 5000 A and maxima of up to 0.7%. Clearly, with strains as large as these the kinematical approach is perfectly vaild.

Various types of ion implantation into magnetic bubble garnets have also been studied using double crystal diffractometry. Ion implantation into magnetic films is particularly useful for the suppression of 'hard' bubbles and the control of bubble wall states. Additionally many of the techniques for achieving high packing densities require ion implantation (Voegli et al, 1974). Komenou et al (1978) have utilised the period of the Bragg case Pendellosung oscillations to determine the thickness of the strained region. Similar results to those obtained by Speriosu et al (1982) for silicon were obtained. By removing the implanted region using a step etching technique and measuring the rocking curve at each stage the thickness predicted by the period of the oscillations was found to be directly proportional to the amount of material removed. This confirmed the applicabilty of the technique even when the layer is not highly perfect.

More recently Speriosu and Wilts (1983) have made a detailed analysis of $He^+$, $Ne^+$ and $H_2^+$ implantation into magnetic bubble garnets using both rocking curves and ferromagnetic resonance spectra (FMR). A wide range of doses were studied for all three species of ions with rocking



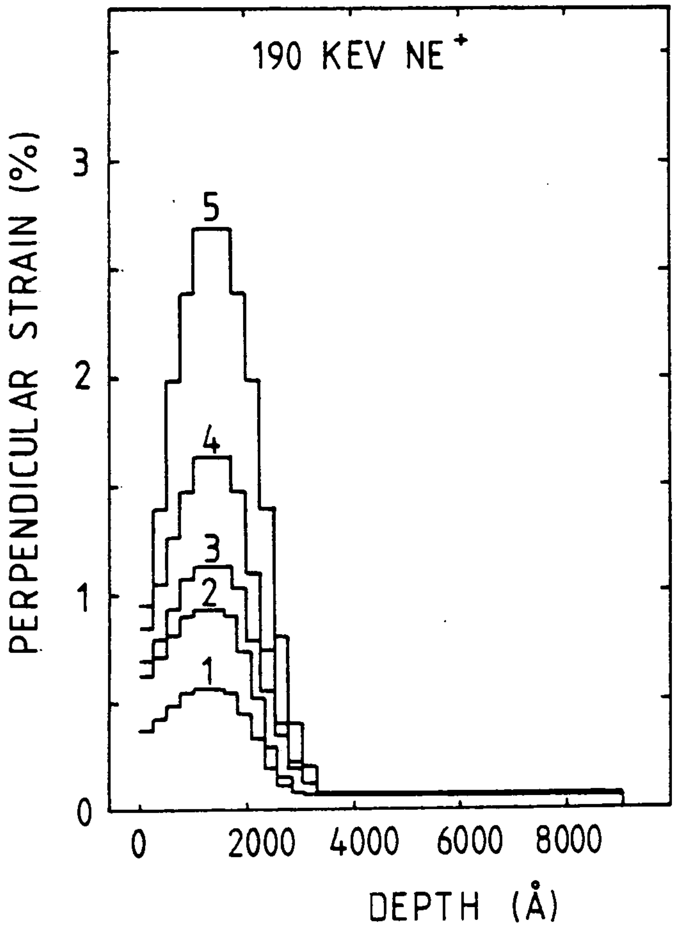


Fig. 2.7 Typical theoretical and experimental rocking curves for a Ne+ implanted magnetic bubble garnet sample and the strain distributions determined.

curves similar in appearance to those obtained by Komenou et al (1978) found. The complete rocking curve was calculated using the kinematical theory in order to determine the strain distribution. Gaussian like strain distributions were again found with widths of the order of 5000 A and maximum strains up to 3%. A typical theoretical and experimental curve are shown in fig. 2.7, along with the strain distribution determined. Agreement with the strain distribution from FMR was fairly good at low doses but considerable differences were found at higher doses.

Takeuchi et al (1983) have also studied ion implantation into bubble garnets but have used the dynamical diffraction theory to calculate the rocking curves. The approach used was that of Fukuhara and Takano. Comparison was made between the kinematical approach of Speriosu et al (1979), and considerable differences in the calculated curves were found. It appeared that some care needed to be taken in the choice of the depth included in the kinematical calculations before the dynamical intensity for the substrate is used. Comparison between the calculated and theoretical curves was again excellent.

### 2.3.2 Heteroepitaxial layers

Double crystal rocking curves from mismatched single layers of uniform composition are often rather more easy to interpret than those from ion or diffusion implanted layers since the epitaxial layers are usually more perfect than the implanted regions, giving rise to narrow, well resolved peaks. The strains encountered are at least an order of magnitude smaller than those encountered in ion implanted crystals. For these types of layers measurement of the perpendicular and parallel mismatches and hence the composition is straightforward. However, this is not the case with multiple and graded composition layers.

Since GaAlAs is well matched to GaAs over a wide composition range and it is not particularly well suited for the manufacture of optoelectronic devices for fibre optic systems little work has been published on characterisation by double crystal diffractometry. Bartels and Nijman (1978) have studied the strain of $Ga_{1-x}Al_xAs$ layers grown on (001), (110), (111) and (113) oriented GaAs substrates. Double heterostructures consisting of 3.2 µm n-$Ga_{0.71}Al_{0.29}As$ / 0.3 µm p-GaAs / 2.5 µm p-$Ga_{0.68}Al_{0.32}As$ / 1.5 µm p-GaAs were studied with the perpendicular and parallel mismatches measured by using the (115) [illegible] reflections. Bragg case Pendellosung fringes were also observed between the substrate and layer peaks.

Ternary compounds of (Ga,In)As grown on (001) InP substrates are rather more important than GaAlAs compounds due to their use in fibre optic systems. Since GaInAs is only well matched to InP over a small range of compositions, and good quality layers can only be grown when the lattice mismatch is <=800ppm, the growth of high quality layers is not straightforward. Of the growth techniques currently available Goetz et al (1983) have compared layers grown by

LPE, VPE and MOCVD. Some 30 samples were studied using double crystal diffractometry to compare the mismatches and layer qualities. Lattice mismatch was measured perpendicular to the interface using the 115 reflection and the layer quality estimated from the half width of the layer peak. In all cases the parallel mismatch was found to be at least an order of magnitude smaller than the perpendicular mismatch suggesting good coherency at the interface. LPE grown layers were found to be better matched to the substrate than those grown by both VPE and MOCVD and also had the narrowest peaks. Both the VPE and MOCVD grown layers gave peaks considerably broader than the theoretical width, suggesting a change in composition with depth. However, no attempt was made to measure this variation.

Similar results have been reported by Kawamura and Okamoto (1979) and by Macrander et al (1984) for MBE grown layers. Again the parallel mismatches were found to be at least an order of magnitude smaller than the perpendicular ones.

Using an automated scanning double crystal camera, with which rocking curves can be recorded automatically at a number of points over the sample, Halliwell et al (1983) have shown that variations in layer perfection occur for MBE grown ternary layers, with a noticeable decrease in layer perfection towards the edge of the sample. This was related to a fall in growth temperature at the edges of the sample, due to a variable thermal contact afforded by the indium solder used to attach the sample to the holder.

Quaternary compounds of (Ga,In)(As,P) offer perhaps the best possibilities for producing high quality, variable band gap layers. LPE has been the favoured growth technique with many X-ray rocking curve studies of single layers being reported. Matsui et al (1979) and Burgeat et al (1981) have both shown that the composition of the layer varies as a function of the layer thickness. Matsui et al were able to show a continuous variation while Burgeat et al assertained a stepwise variation. Both used a similar technique of step etching the sample and measuring the rocking curve at each stage. Some doubt exists as to both of these interpretations as no theoretical calculations were used for comparison. Matsui et al took only the angular separation of the maximum of the layer peak as a measure of the composition while Burgeat et al used large etch steps and associated each peak removed from the curve with the layer removed. Matsui et al showed that Auger electron spectroscopy also revealed a continuous linear variation in composition with depth.

Such variations were also reported by Tashima et al (1981) using the 006 reflection in conjunction with a modified powder goniometer. However, no effect could be observed from the thin region near to the interface expected to have a rapidly varying composition (Rezek et al, 1980).

Double heterostructure layers of GaInAsP/InP, grown by LPE, have been similarly studied by Oe et al (1978). Since the layers studied were only approximately 0.4 μm thick no composition variations were observed. Similar results were also obtained for 1.5 μm GaInAsP / 8 μm GaInP / InP layers by Bartels (1983) using a multiple crystal spectrometer with a four reflection Ge monochromator. Using the dynamical theory to calculate the peak heights expected from such a structure the layer thicknesses were found to agree well with those expected from the growth conditions.

Halliwell et al (1983) were also able to show an increase in the layer thickness perpendicular to the push direction for LPE grown quaternary layers. Layer quality remained good over the majority of the sample and the layer thicknesses were calculated to increase by about 50% towards the edges.

In none of the above work has the complete rocking curve been calculated and compared with the experimental curves. Clearly if the rocking curve from a layer with an arbitrary composition profile was calculated, comparison with experimental curves could be used to aid interpretation in a similar manner to that used for ion and diffusion implanted layers.

Using a similar approach to that of Burgeat and Colella (1969), Halliwell, Juler and Norman (1983) were able to obtain a fairly good agreement between theoretical and experimental curves from linearly graded single layers of GaInAs. The composition profile was also measured with electron probe X-ray microanalysis on a bevelled section, and agreement between the two methods was fairly good.

Discrepancies between the theoretical and experimental curves was thought to be due to a region of different composition near to the interface, a region that the microprobe cannot accurately measure due to the proximity of the substrate. This is in agreement with the observations of Rezek et al (1980).

Very thin heteroepitaxial layers have not been extensively studied by double crystal diffractometry, primarily because the diffracted intensity from such layers is very low for the normally used reflections. Instead other X-ray techniques such as low angle interference (Segmuller, 1973) have been used. However, the use of highly asymmetric reflections, in which either the diffracted or incident beams forms a low angle with the surface can overcome this problem (Tanner, Barnett and Hill, 1985).

### 2.3.3 Superlattices

X-ray diffraction studies of artificial superlattices were first reported by Segmuller and Blakeslee (1973) where alternating layers of $GaAs_{0.5}P_{0.5}$ and GaAs formed the superlattice. Vapour phase transport was used as the growth technique. Measurements were recorded with a modified single crystal camera with radiation monochromated by a double reflecting channel cut Ge crystal. Satellite peaks were observed near to the 004 GaAs reflection. A model based on a Fourier decomposition of the composition variation was used to calculate the satellite intensities in conjunction with the kinematical diffraction theory. This model was very similar to that of Dehlinger (1927) for the study of diffraction from cold worked metals, although a second harmonic term was included by Segmuller and Blakeslee. Similar theories have been presented by Kochendorfer (1939) and Daniel and Lipson (1943, 1944), where a sinusoidal variation in lattice parameter and structure factor was considered. Hargreaves (1951) studied the case for a square wave model of the superlattice, but his results were the same as those obtained by the sinusoidal models due to the approximations used. A sinusoidal model was also used by De Fontaine (1966) for such one dimensional variations.

Using this approach Segmuller and Blakeslee were able to calculate satellite intensity ratios for different strain and composition distributions. Reasonable agreement with the first order satellite peaks was found, although not for the second order satellites which were only observed for one sample. It was also found that this model gave better agreement than a step model, suggesting that some degree of interdiffusion had occurred at the interfaces.

This approach was repeated by Segmuller, Krishna and Esaki (1977) for GaAs/AlAs superlattices grown by MBE, where a greater number of high order satellites were observable. Low angle X-ray scattering interference has also been used to determine the ratio of thicknesses of the layers forming the repeat period (Chang et al, 1976). Calculated superlattice structure factors were found to agree fairly well with those calculated from the satellite intensities assuming a perfect stepwise composition variation. The effect of interdiffusion between the layers and compositional non uniformity along the layers was also discussed. Both of these factors have the same effect of reducing the satellite intensities. Since the zeroth order peak, corresponding to the reflection from a layer with the average composition of the superlattice, could not be resolved from the substrate reflection their separation could not be used to determine the composition.

This kinematical approach to calculating satellite intensities has also been used by Fleming et al (1980) and Kevarec et al (1984). By calculating the change in the Fourier components of the strain modulation Fleming et al were able to measure the effect of interdiffusion as a function of annealing time for GaAs/AlAs superlattices, again grown by MBE. Up to eight harmonics were included in the Fourier series, although after long annealing times only the first two could be determined. The diffusion coefficients obtained were found to be in good agreement with those obtained from optical measurements (Dingle, 1977) and electron microscopy (Petroff, 1977), but were smaller than those obtained by Auger electron spectroscopy (Chang and Koma, 1976).

$Ga_{1-x}Al_xAs/GaAs$ superlattices were studied by Kevarec et al using both a double crystal technique and a modified powder goniometer to observe the high order, weak satellites. Clearly, for these superlattices the value x must also be determined and the separation of the zeroth order and substrate peaks can be used for this. Finally a step wise calculation of the structure factor was used to calculate the satellite intensities.

More recently Speriosu and Vreeland (1984) have applied their model of X-ray diffraction from ion implanted layers to the calculation of diffracted intensities from MBE grown GaAlAs/GaAs and MOCVD grown AlSb/GaSb superlattices. A step wise model was used with the inclusion of both perpendicular and parallel mismatches. Double crystal rocking curves of various reflections were recorded with FeK$\alpha_1$ radiation. Good agreement was found with the calculated curves, particularly for the GaAlAs/GaAs samples.

The dynamical diffraction theory has not been extensively used in the calculation of diffracted intensities from superlattices, due to the individual layer thicknesses being very much less than the extinction distances. However, recently Vardanyan, Manoukyan and Petrosyan (1985) have pointed out that, as the total thickness of the superlattice is the order of an extinction distance, dynamical interactions between the beams from the layers should not be ignored. They have used the well known matrix method to calculate the overall diffracted intensities from a perfect superlattice.

# CHAPTER 3

# CALCULATING ROCKING CURVES

As we have seen, it is necessary to calculate rocking curves in order to extract as much information as possible as to the composition and thickness of any epitaxial layers. An approach based on dynamical diffraction theory will be used to calculate the single crystal reflection profiles for the crystals, the rocking curve is then the convolution of the reflection curves of the first and second crystals, as already seen.

## 3.1 Dynamical Diffraction Theory

There are many excellent reviews of dynamical diffraction theory available in the literature, the reader is referred to these for a complete description (Zachariasen, 1945; James, 1948; von Laue, 1960; Batterman and Cole, 1964; Pinsker, 1978).

A solution is required to Maxwell's equations for a medium with a periodic, complex dielectric constant. The fundamental equation linking the incident and diffracted waves within the crystal is obtained,

$$\alpha_0\alpha_h = \frac{k^2}{4} C^2 \chi_h \chi_{\bar{h}} \qquad (3.1)$$

where

$$\alpha_0 = \frac{k}{2} (\underline{K}_0 \cdot \underline{K}_0 - k^2 (1+\chi_0)) \qquad (3.2)$$

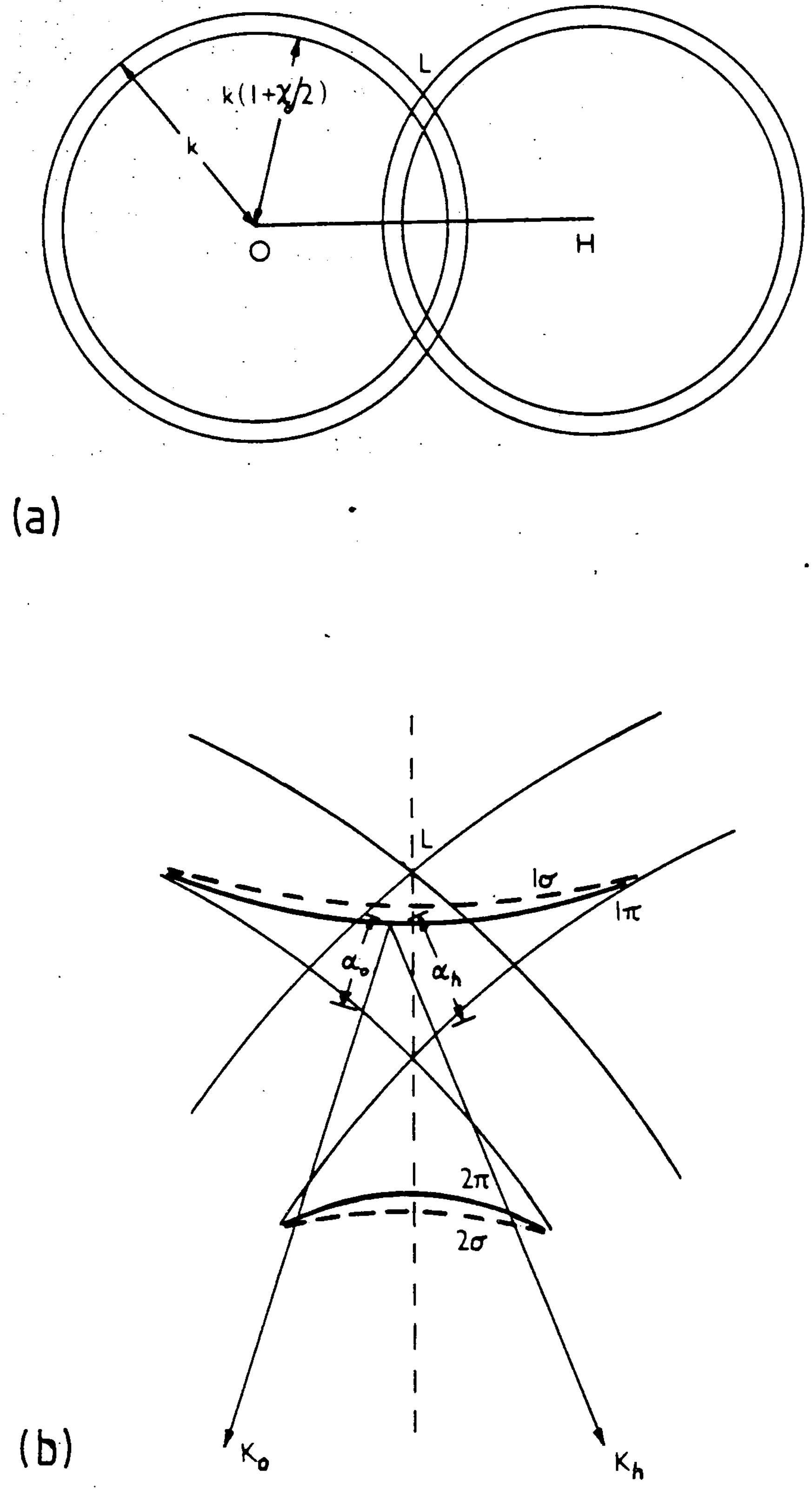


Fig. 3.1 (a) Laue construction for the diffraction of an incident beam of wavevector K.

(b) The dispersion surface construction showing the four possible branches. Arcs AA and BB are part of circles of radii K and A'A' and B'B' are of radii $k(1+X_0/2)$.

and $$\alpha_h = \frac{k}{2} (\underline{K}_h \cdot \underline{K}_h - k^2(1+\chi_0)). \qquad (3.3)$$

C is the polarisation factor (=1 for $\sigma$ and =cos2$\theta_B$ for $\pi$ polarisations). As usual $\underline{K}_0$, $\underline{K}_h$ are the incident and diffracted wavevectors in the crystal and $\chi_0$, $\chi_h$, $\chi_{\bar{h}}$ the complex susceptibilities. This equation can be usefully represented geometrically by the dispersion surface construction. Considering the Laue construction shown in fig. 3.1, surfaces on which the tails of the possible wavevectors $\underline{K}_0$ and $\underline{K}_h$ lie can be drawn, asymptotic to spheres of radius $k(1+\chi_0/2)$ and k drawn about the reciprocal lattice points 0 and H. $\alpha_0$ and $\alpha_h$ are then the perpendicular distances between the dispersion surface branches and these spheres. As this region is very small compared to the radius of the spheres the spheres can be approximated by planes, in which case the equation of the dispersion surface (3.1) becomes a hyperboloid of revolution. Generally there are four dispersion surface branches, two each for the two polarisation states.

Since any wave propagating within the crystal must have wavevectors lying on the dispersion surface we can determine both the amplitude and wavevector from the tie points excited on the surface. Hence this construction is extremely useful in visualising the physical processes occurring within the crystal.

The tie points excited on the dispersion surface are determined by the boundary conditions. Clearly these conditions are extremely important as we must be able to determine the wave amplitudes outside the crystal, since it is only these that we can measure. The required condition is that the tangential components of the plane waves across the boundary must be equal. Hence the wavevectors inside the crystal can only differ from those outside by a vector perpendicular to the boundary. This condition can be used to determine the tie points excited on the dispersion surface.

For Bragg reflection no tie points can be excited in the central region of the reflection profile, thus no propagated wave can exist inside the crystal. Instead a highly damped, evanescent wave exists and the incident beam is totally externally reflected. The Bragg reflection profile, with zero absorption, will then have a flat topped central region. The actual position of the centre of this profile does not correspond to the kinematical Bragg angle due to the refractive effect. The corrected Bragg angle is given by

$$\theta_B = \theta_0 + \chi_0(1 - \gamma_h/\gamma_0)/2\sin2\theta_0 \qquad (3.4)$$

where $\theta_0$ is the kinematical Bragg angle and $\gamma_h$, $\gamma_0$ the usual asymmetry factors. We note that only for symmetric Laue case diffraction does $\theta_B = \theta_0$. The full width at half maximum of the reflection peak is given by

$$\Delta\theta_{\frac{1}{2}} = \frac{2C(\chi_h\chi_{\bar{h}})^{\frac{1}{2}}}{\sin2\theta_B}\sqrt{\frac{|\gamma_h|}{\gamma_0}} \qquad (3.5)$$

For Bragg geometry, with no crystal absorption, the reflected intensity ratio is given by (Pinsker, 1978)

$$R = \frac{1 - \cos 2A(y^2 - 1)^{\frac{1}{2}}}{\cosh 2v - \cos 2A(y^2 - 1)^{\frac{1}{2}}} \tag{3.6}$$

where

$$y = \frac{\beta}{2C(\chi_h \chi_{\bar{h}})^{\frac{1}{2}}(\gamma_h/\gamma_0)^{\frac{1}{2}}} \tag{3.7}$$

and

$$\beta = 2\Delta\theta \sin 2\theta - \chi_0(1 + |\gamma_h|/\gamma_0). \tag{3.8}$$

A is given by

$$A = \frac{\pi C k t(\chi_h \chi_{\bar{h}})^{\frac{1}{2}}}{(\gamma_0 |\gamma_h|)^{\frac{1}{2}}} \tag{3.9}$$

and v by y = sinhv.

We note that equation (3.6) describes subsidiary maxima of Bragg case Pendellosung in the lateral regions of the reflection peak. The expression vanishes to zero provided that

$$2A(y^2 - 1)^{\frac{1}{2}} = 2\pi m, \text{ m an integer,} \tag{3.10}$$

giving an angular interval of

$$\Delta y \approx \frac{\pi}{A} = \frac{\lambda(\gamma_0 |\gamma_h|)^{\frac{1}{2}}}{Ct(\chi_h \chi_{\bar{h}})^{\frac{1}{2}}} \tag{3.11}$$

This interval corresponds to that for Laue case Pendellosung, and has been used in the determination of crystal thickness as previously described. We note that the length $\lambda(\gamma_0|\gamma_h|)^{\frac{1}{2}}/C(\chi_h\chi_{\bar{h}})^{\frac{1}{2}}$ is known as the extinction distance, and corresponds to the separation of the maxima of the wavefield excited within the crystal due to the interference of the two waves propagating in the crystal. This length also gives an indication of the thickness of the crystal sampled by the X-ray beam during Bragg reflection, and is the order of 10 μm for the majority of reflections for III-V materials. If absorption is included the Bragg case Pendellosung oscillations become rapidly damped as the thickness of the crystal is increased, hence these oscillations are only observable for thin crystals.

Clearly we could use this dynamical approach to calculate the rocking curve from a layered crystal, matching the wavevectors out of one layer into the next at each interface. For multiple layers, or graded layers divided into a number of laminae, this approach would be extremely laborious. A much simpler approach, which is well suited to computation, is to use the differential equations of Takagi and Taupin (Takagi, 1962, 1969; Taupin, 1964). However, it is rather difficult to visualise the physical processes occurring within the crystal with this approach, hence it is useful to keep the dispersion surface approach in mind.

## 3.2 The Takagi-Taupin Equations

The derivation of Taupin (1964), which has also been presented by Pinsker (1978), is followed here. The reader is referred to these works for complete details.

The electric displacement vector in a vacuum can be described by the expression

$$\underline{D} = \underline{D}_0(\underline{r})\exp[i(\omega_0 t - 2\pi\Phi_0(\underline{r})] \qquad (3.12)$$

where $\omega_0$ is the cyclical frequency ($\omega_0=2\pi\nu_0$). This expression descibes an arbitrary wave. For a plane wave $\Phi_0(r) = \underline{K}_0\underline{r}$, $|\underline{K}_0|=1/\lambda$. In the general case $\underline{D}$ satisfies the wave equation in a vacuum

$$\nabla^2\underline{D} + \frac{\omega^2{}_0}{c^2}\underline{D} = 0 \qquad (3.13)$$

Substituting (3.12) into (3.13) gives

$$\nabla^2\underline{D}_0 - i4\pi\sum_i \frac{\partial\Phi_0}{\partial x_i}\frac{\partial\underline{D}_0}{\partial x_i} - i2\pi\underline{D}_0\nabla^2\Phi_0 - 4\pi^2\sum_j \frac{\partial\Phi_0}{\partial x_j}{}^2\underline{D}_0$$

$$+ \frac{4\pi^2}{\lambda^2}\underline{D}_0 = 0 \qquad (3.14)$$

Since $\underline{D}_0$ and $\Phi_0$ are real functions of the coordinates separation into real and imaginary parts is straightforward.

Considering waves for which the radii of curvature of the equiphase surfaces are much larger than the wavelength $\lambda$; for example, $\lambda/R<<1$ can be assumed for a spherical wave thus

$$|\mathrm{grad}\ \Phi_0| \approx \lambda^{-1}[\ 1 + O(\lambda^2/R^2)\ ] \qquad (3.15)$$

where $O(\lambda^2/R^2)$ represents negligibly small terms of higher orders of $\lambda/R$. For a nearly plane incident wave we have

$$\mathrm{grad}\ \Phi_0 = \underline{K}_0 + \Delta\underline{K}_0 \qquad (3.16)$$

and $\underline{K}_0 \Delta \underline{K}_0 = 0$, $|\Delta \underline{K}_0| << |\underline{K}_0|$. From (3.15) we then have

$$|\underline{K}_0| = \lambda^{-1}, \quad |\Delta \underline{K}_0| = R^{-1} \tag{3.17}$$

corresponding to an incident wave packet with an angular width of the order of 1/R.

The function $\underline{D}_0$ is retained for the phase when describing the wave inside the crystal. The amplitude $\underline{D}_0$ becomes a complex function of the coordinates, which is dependent on the difference in the paths of propagation in the crystal and the vacuum. In the case of a perfect crystal we have a single pseudo periodic wave corresponding to the right hand side of

$$\begin{aligned} &\sum_i D_0^{(i)} \exp 2\pi i[\nu t - (K_0^{(i)} . \underline{r})] \\ &\quad = \exp 2\pi i[\nu t - (K_0 r)] \sum_i D_0^{(i)} \exp(\pm\pi i \Delta K z) \\ \text{and} \quad &\sum_i D_h^{(i)} \exp 2\pi i[\nu t - (K_h^{(i)} \underline{r})] \\ &\quad = \exp 2\pi i[\nu t - (\underline{K}_h \underline{r})] \sum_i D_h^{(i)} \exp(\pm\pi i \Delta K z) \end{aligned} \tag{3.18}$$

for each state of polarisation. Hence the boundary condition

$$D_0^{(1)} + D_0^{(2)} = D_0^{(a)} \tag{3.19}$$

amounts to regarding $\underline{D}_0$ as a function of the coordinates on transition from the vacuum to the crystal.

The wavefield in the crystal can be described through the use of a variable amplitude Bloch function,

$$\underline{D} = \sum_m \underline{D}_m \exp \{i[\omega t - 2\pi(\underline{K}_m . \underline{r})]\}. \tag{3.20}$$

For a perfect crystal and plane waves the following values remain constant

$$\underline{K}_m = \underline{K}_0 + \underline{h}_m,\ \underline{K}_0 = \text{grad } \Phi_0. \qquad (3.21)$$

However, generally $\underline{K}_0$ and $\underline{h}_m$ are functions of the coordinates. For each point in the crystal the vector $\underline{h}_m$ can be determined. Additionally,

$$|\underline{h}_m(\underline{r})| = d^{-1}{}_m(\underline{r}). \qquad (3.22)$$

Hence, moving from the plane $n_m$ to the plane $n_m+1$ of the same system, we have $\Delta n_m = 1 = \underline{h}_m\underline{dr}$, where $\underline{dr}$ is the change in the radius $\underline{r}$ in moving from the point $\underline{r}$ on the $n_m$-th plane to the point $\underline{r}+\underline{dr}$ on the $n_m$+1-th plane.

If $n_m$ is regarded as a continuous function of the coordinates, which takes integral values on each plane m, we find

$$\underline{h}_m = \text{grad } n_m \qquad (3.23)$$

$$\underline{D} = \sum_m \underline{D}_m\ [i(\omega_0 t - 2\pi\varphi_m)] \qquad (3.24)$$

where
$$\varphi_m(\underline{r}) = \varphi_0(\underline{r}) + n_m(\underline{r}). \qquad (3.25)$$

The continuity of $n_m$ breaks down in crystals containing dislocations. However, it can be shown that the results of the theory outlined can be used in the general case of an arbitrary distribution of $n_m$ if the distortion, $\partial u/\partial x$, is small (ie $\partial u/\partial x << 1$).

The dielectric constant remains a periodic function of the coordinates, but $\varepsilon(\underline{r})$ can be approximately assumed to be an exponential function of $n_m$ in each system of reflecting planes since $\partial u/\partial x << 1$.

$$\chi(\underline{r}) = \sum_m \chi_m \exp[-2\pi i n_m(\underline{r})] \qquad (3.26)$$

$$\chi_m \equiv \chi_{hm} = -\frac{e^2}{mc^2}\frac{\lambda^2}{\pi V} F_{hm} \qquad (3.27)$$

and for the general case of an absorbing crystal

$$\chi_0 = \chi_{0r} + i\chi_{0i};\ \chi h = \chi_{hr} + i\chi_{hi}. \qquad (3.28)$$

The variable representing the angular departure from the Bragg condition is taken as

$$\alpha_m = 2\Delta\theta_m \sin 2\theta_m \qquad (3.29)$$

which is derived from the approximate value of the expressions

$$\alpha_m = K_0^{-2}[h_m^2 + 2(\underline{K}_0 . \underline{h}_m)] = \lambda^2(1/d_m^2 - 2\sin\theta_m/\lambda d_m) \qquad (3.30)$$

where both $d_m$ and $\sin\theta_m$ are coordinate dependent.

The wave equation inside the crystal is derived from Maxwell's equations, which are

$$\text{curl } \underline{E} = c^{-1}\ \partial\underline{H}/\partial t, \quad \text{curl } \underline{H} = c^{-1}(\partial\underline{E}/\partial t + 4\pi\underline{J})$$

$$(3.31)$$

$$\text{div } \underline{E} = 4\pi\rho; \quad \text{div } \underline{H} = 0.$$

Taking the curl of both sides we obtain

$$\text{curl curl}\underline{E} = c^{-1} \text{ curl } \partial\underline{H}/\partial t = c^{-1} \partial/\partial t \text{ curl } \underline{H} \qquad (3.32)$$

but
$$\text{curl } \underline{H} = c^{-1} \partial\underline{D}/\partial t \qquad (3.33)$$

and from $\underline{D} = \varepsilon\underline{E} = (1 + \chi)\underline{E}$, $\underline{E} \approx (1-\chi)\underline{D}$ we obtain

$$\text{curl curl } (1-\chi)\underline{D} = 4\pi^2/\lambda^2 \ \underline{D}. \qquad (3.34)$$

In order to calculate the left hand side of equation (3.34) the value of $(1-\chi)\underline{D}$ is rewritten using the solution of the wave equation and the Fourier expansion of the polarisability. We can write $n_K + n_L = n_{K+L}$ by analogy with the reciprocal vectors in the perfect crystal ie $h_K + h_L = h_{K+L}$. Thus

$$\begin{aligned}\underline{E} = (1-\chi)\underline{D} &= \exp(i\omega t) \Big[\sum_m \underline{D}_m \exp(-i2\pi\varphi_m) \\ &\quad -\sum_m \sum_h \chi_m \underline{D}_h \exp[-i2\pi(n_m+\varphi_h)]\Big] \\ &= \exp\,(i\omega_0 t) \sum_m \underline{Q}_m \exp(-i2\pi\varphi_m) \qquad (3.35)\end{aligned}$$

$$\underline{Q}_m = \underline{D}_m - \sum_h \chi_{m-h} \underline{D}_h \qquad (3.36)$$

Omitting the factor $\exp(i\omega_0 t)$ and using the following relations

$$\underline{K}_0 = \text{grad } \varphi_0, \ \underline{h}_m = \text{grad } n_m, \ \underline{K}_m = \text{grad } \varphi_m. \qquad (3.37)$$

and
$$(\text{curl curl } \underline{E})_i = -\frac{\partial^2 E_i}{\partial x_k^2} + \frac{\partial^2 E_k}{\partial x_i \partial x_k} \qquad (3.38)$$

where i and k take the value 1,2 and 3, we obtain

$$\frac{\partial}{\partial x_k}(1-\chi)\, D_{mi} = \exp\,(-i2\pi\varphi_m) \ \frac{\partial Q_{mi}}{\partial x_k} - i2\pi K_{mk} Q_{mi} \qquad (3.39)$$

Taking a second differential gives, after using

$$\frac{\partial}{\partial x_i} K_{mk} = \frac{\partial^2 \varphi_m}{\partial x_i \partial x_k} = \frac{\partial}{\partial x_k} K_{mi} \tag{3.40}$$

$$\begin{aligned} \text{curl curl } (1-\chi)\underline{D} = \sum_m \exp(-i2\pi\varphi_m) \{ & 4\pi^2 \, [K_m^2 \underline{Q}_m - (\underline{K}_m \underline{Q}_m)\underline{K}_m] \\ & + i4\pi \underline{K}_m \frac{\partial \underline{Q}_m}{\partial r} - i2\pi \text{grad } (\underline{K}_m \underline{Q}_m) \\ & - i2\pi \underline{K}_m \text{ div } \underline{Q}_m + i2\pi \underline{Q}_m \nabla^2 \varphi_m \\ & + \text{grad div } \underline{Q}_m \} = A \end{aligned} \tag{3.41}$$

With the Bloch solution we obtain

$$4\pi^2/\lambda^2 \sum_m \underline{D}_m \exp(-i2\pi\varphi_m) = A. \tag{3.42}$$

This equation has an infinite number of unknowns. If it is assumed that the preexponential factors with number m only change over effective distances that are much larger than the wavelength each of the waves referring to different waves m can be equated. Thus

$$\begin{aligned} & 4\pi^2 \, [K_m^2 Q_m^2 - K_0^2 (\underline{D}_m \underline{Q}_m) - (\underline{Q}_m \underline{K}_m)^2] + i2\pi \, \underline{K}_m \text{ grad } Q_m^2 \\ & + i2\pi \, \lfloor Q_m^2 \nabla^2 \varphi_m - \underline{Q}_m \text{grad}(\underline{K}_m \underline{Q}_m) - (\underline{K}_m \underline{Q}_m)\text{div } \underline{Q}_m] \\ & + \underline{Q}_m \text{grad div } \underline{Q}_m - \underline{Q}_m \nabla^2 \underline{Q}_m = 0 \end{aligned} \tag{3.43}$$

replacing $K_m^2$ by $\alpha_m$ in (3.43) we obtain

$$\begin{aligned} & 4\pi^2 \, [K_0^2 \alpha_m Q_m^2 - K_0^2 \sum \chi_{m-h} \underline{D}_h \underline{Q}_m - (\underline{Q}_m \underline{K}_m)^2] \\ & + i2\pi \, [\underline{K}_m \text{grad } Q_m^2 + Q_m^2 \nabla^2 \underline{Q}_m - \underline{Q}_m \text{grad } (\underline{K}_m \underline{Q}_m)\rfloor \\ & - (\underline{K}_m \underline{Q}_m)\text{div } \underline{Q}_m + \underline{Q}_m \text{ grad div } \underline{Q}_m - \underline{Q}_m \nabla^2 \underline{Q}_m = 0 \end{aligned} \tag{3.44}$$

This system contains terms which differ considerably in their order of magnitude. Terms small comapared to $Q_m^2/\lambda^2$ =

1 are now eliminated. We then obtain (see Pinsker (1978) for a discussion of the magnitude of each term)

$$\alpha_m \underline{D}_m - \sum_h \chi_{m-h} \underline{D}_h \cos X_{mh} + i\lambda^2/\pi\ (\underline{K}_m \mathrm{grad})\underline{D}_m = 0 \qquad (3.45)$$

where $\cos X_{mh}$ is the polarisation factor. The calculation of the solution for X-ray propagation in the perfect crystal is then the solution of this system in first order partial derivatives. The variable $\alpha_m$ is calculated to allow for local deformations at any point in the crystal.

For the two beam case we need only consider m=0 and m=h. Let $\underline{s}_0$ and $\underline{s}_h$ be unit vectors in the direction of the incident and diffracted beams respectively, thus

$$\underline{s}_0 = \lambda\ \underline{K}_0, \quad \underline{s}_h = \lambda\ \underline{K}_h \qquad (3.46)$$

and for any point on the reflection plane

$$\underline{r} = s_0 \underline{s}_0 + s_h \underline{s}_h \qquad (3.47)$$

and the system (3.35) reduces to

$$\frac{i\lambda}{\pi} \frac{\partial D_0}{\partial s_0} = \chi_0 D_0 + C\chi_{\bar{h}} D_h$$

$$\frac{i\lambda}{\pi} \frac{\partial D_h}{\partial s_h} = (\chi_0 - \alpha_h) D_h + C\chi_h D_0 . \qquad (3.48)$$

$$C = \cos X_{0h}$$

Clearly, we can include the polarisation factor C in the values $\chi_h$ and $\chi_{\bar{h}}$ by adding the symbols $\sigma$ and $\pi$, ie

$$\chi_h^{\sigma} = \chi_h^{\pi} \, (|\cos 2\theta|)^{-1} \qquad (3.49)$$

and

$$\chi_{\bar{h}}^{\sigma} = \chi_{\bar{h}}^{\pi} \, (|\cos 2\theta|)^{-1}.$$

## 3.3 Solution of the Takagi-Taupin Equations and Calculation of the Rocking Curve

The Takagi-Taupin equations (3.48) can now be used to calculate the reflectivity of a given crystal in the Bragg case. The amplitude ratio of the incident and diffracted beams is required. We will assume an incident plane wave, and that variations in diffracted intensity will be a function of depth only.

If $\gamma_0$ and $\gamma_h$ are the direction cosines of the incident and diffracted beams with respect to the surface, ie

$$z = s_0\gamma_0 + s_h\gamma_h, \qquad (3.50)$$

then (3.48) becomes

$$\frac{i\lambda}{\pi}\gamma_0\frac{\partial D_0}{\partial z} = \chi_0 D_0 + C\chi_{\bar{h}}D_h$$
$$\frac{i\lambda}{\pi}\gamma_h\frac{\partial D_h}{\partial z} = (\chi_0 - \alpha_h)D_h + C\chi_h D_0 \qquad (3.51)$$

Now the complex reflection coefficient, X, is defined as

$$X = \sqrt{\frac{|\gamma_h|}{\gamma_0}}\,\frac{D_h}{D_0} \qquad (3.52)$$

where the $\gamma_h$, $\gamma_0$ terms account for any beam expansion or compression. Differentiating (3.52) we obtain

$$\frac{dX}{dz} = \sqrt{\frac{|\gamma_h|}{\gamma_0}}\left\{\frac{1}{D_0}\frac{\partial D_h}{\partial z} - \frac{D_h}{D_0^2}\frac{\partial D_0}{\partial z}\right\} \qquad (3.53)$$

hence, substituting into (3.48) we arrive at

$$\frac{dX}{dz} = \sqrt{\frac{|\gamma_h|}{\gamma_0}}\left\{C\frac{\chi_h}{\gamma_0}X^2 + \left(\frac{\chi_0}{\gamma_0} + \frac{\chi_0}{|\gamma_h|} - \frac{\alpha_h}{|\gamma_h|}\right)X + \frac{C\chi_h}{|\gamma_h|}\right\}\frac{i\pi}{\lambda} \qquad (3.54)$$

where $|\gamma_h| = -\gamma_h$ has been assumed for Bragg reflection. The expression is similar to that given by Burgeat and Collela (1967), but includes the factors $\gamma_h$ and $\gamma_0$ as required for non surface symmetric reflections. The susceptibilities $\chi_h$ and $\chi_{\bar{h}}$ are also included and are not assumed to be equal.

Clearly, in a crystal/layer structure containing compositional variations $\alpha_h$ will be a function of the depth, z, below the surface. Solution of equation (3.54) would then need to be performed numerically. However, if we divide the crystal into a number of laminae of constant composition (these laminae will correspond to individual layers if the layers are of uniform composition), in which $\alpha_h(z)$ is taken to be constant, equation (3.54) can be solved analytically for each laminae. The complex amplitude ratios are then matched at each boundary in order to obtain the reflectivity at the surface.

Let us rewrite equation (3.54) as

$$\frac{dX}{dz} = iD(AX^2 + 2BX + E) \qquad (3.55)$$

where $D = \frac{\pi}{\lambda}\sqrt{\frac{|\gamma_h|}{\gamma_0}}$, $A = C\frac{\chi_{\bar{h}}}{\gamma_0}$, $B = \frac{1}{2}\left(\frac{\chi_0}{\gamma_0} + \frac{\chi_0}{|\gamma_h|} - \frac{\alpha_h}{|\gamma_h|}\right)$, $E = C\frac{\chi_h}{|\gamma_h|}$.

Then

$$\frac{dX}{dz} = iDA\left[\left(X + \frac{B}{A}\right)^2 - \frac{B^2}{A^2} + \frac{E}{A}\right] \qquad (3.56)$$

With the substitution $X = -\frac{B}{A} + \frac{\sqrt{EA - B^2}}{A}\tan Y$,

the RHS of (3.56) becomes

$$i\,DA\,\frac{(EA - B^2)}{A^2}\,(1 + \tan^2 Y)$$

and the LHS becomes $\frac{\sqrt{EA - B^2}}{A}(1 + \tan^2 Y)\frac{dY}{dz}$

ie $$\frac{dY}{dz} = iD\sqrt{EA - B^2} \qquad (3.57)$$

We can assume that the reflectivity is known at a depth W, ie X(W)=K, hence we obtain

$$\int_{Y(W)}^{Y(z)} dY = \int_{W}^{Z} iD\sqrt{EA - B^2}\, dz$$

ie $$Y(z) = iD\sqrt{EA - B^2}\,(z-W) + \tan^{-1}\left(\frac{AK + B}{\sqrt{EA - B^2}}\right)$$

ie $$X = \frac{1}{A}\left[-B\sqrt{EA - B^2} + B(AK + B)\tan(iD\sqrt{EA - B^2}\,(z-W)) + (EA - B^2)\tan(iD\sqrt{EA - B^2}\,(z-W)) + (AK + B)\sqrt{EA - B^2}\right] \times \left[\sqrt{EA - B^2} - (AK + B)\tan(iD\sqrt{EA - B^2}\,(z-W))\right]^{-1}$$

ie $$X = \frac{K\sqrt{EA - B^2} + (E + BK)\tan(iD\sqrt{EA - B^2}\,(z-W))}{\sqrt{EA - B^2} - (AK + B)\tan(iD\sqrt{EA - B^2}\,(z-W))} \qquad (3.58)$$

Now for an infinitely thick crystal $K \to 0$ as $z-W \to \infty$. Since $\tan(a+ib) \to i$ as $b \to \infty$ and $\tan(a+ib) \to -i$ as $b \to -\infty$ we can write (3.58) as

$$X = -\frac{B + \sqrt{B^2 - EA} \times \text{sign}(\text{Imaginary}\sqrt{B^2 - EA})}{A} \qquad (3.59)$$

provided that $B^2$-EA is not wholly real.

Now for a centrosymmetric crystal E=A, and writing B/A = $\eta$ we obtain

$$X = \eta \pm \sqrt{\eta^2 - 1} \qquad (3.60)$$

which is equivalent to the result derived for an infinitely thick crystal by Darwin (see for example Batterman and Cole, 1964).

The reflectivity which represents the Bragg reflection profile is then the modulus squared of this complex reflectivity, ie $C=|X|^2$.

In order to calculate the diffracted intensity from a layered crystal we use the equation for the reflectivity from an infintely thick crystal for the reflectivity at the top of the substrate. This then provides the necessary boundary condition at the bottom of the first layer. Using equation (3.58) we can then obtain the reflectivity at the top of this layer. This process is then repeated until the reflectivity is obtained at the surface of the crystal, over the range of angles $\alpha$ required.

Once the reflectivity is obtained we can calculate the rocking curve using the convolution equation (2.17), which gives the total power reflected by the second crystal at an angle $\beta$. The rocking curve is calculated by performing this integration for the range of angles $\beta$ required. $C_{A,B}$ are the single crystal reflectivities for the first and second crystals repectively. For the first crystal the reflectivity for an infinite crystal is used.

## 3.4 The Effect of Sample Curvature on the Rocking Curve

As already seen, the growth of a mismatched epitaxial layer on a substrate produces overall sample curvature. This curvature will, clearly, affect the shape of the rocking curve if the misorientation introduced is of similar order to the width of the rocking curve. There are two ways in which the misorientation can be introduced.

If we consider the incident beam passing through a

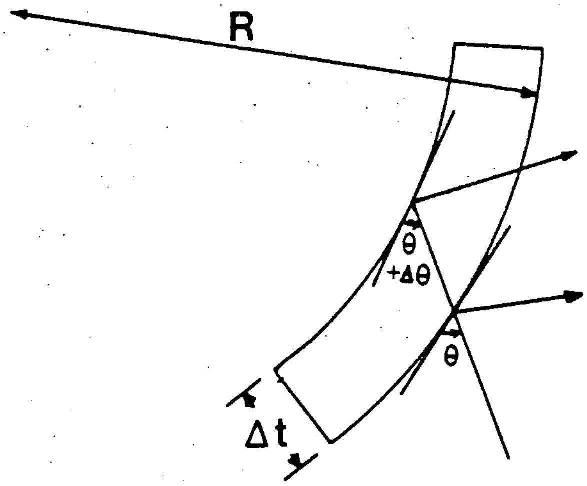


Fig. 3.2 Change in glancing angle Θ for an incident x-ray beam and a curved crystal of thickness Δt. R is the radius of curvature of the crystal.

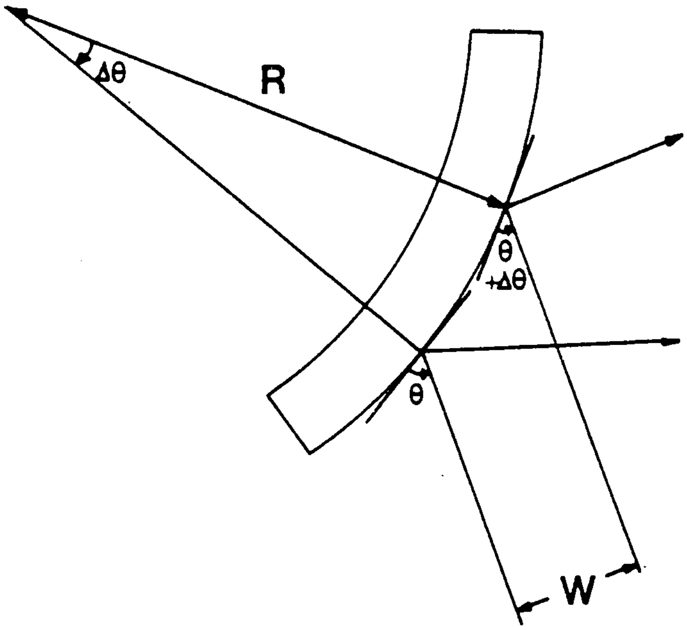


Fig. 3.4 Change in Bragg angle along an incident beam of width W for a crystal of radius of curvature R.

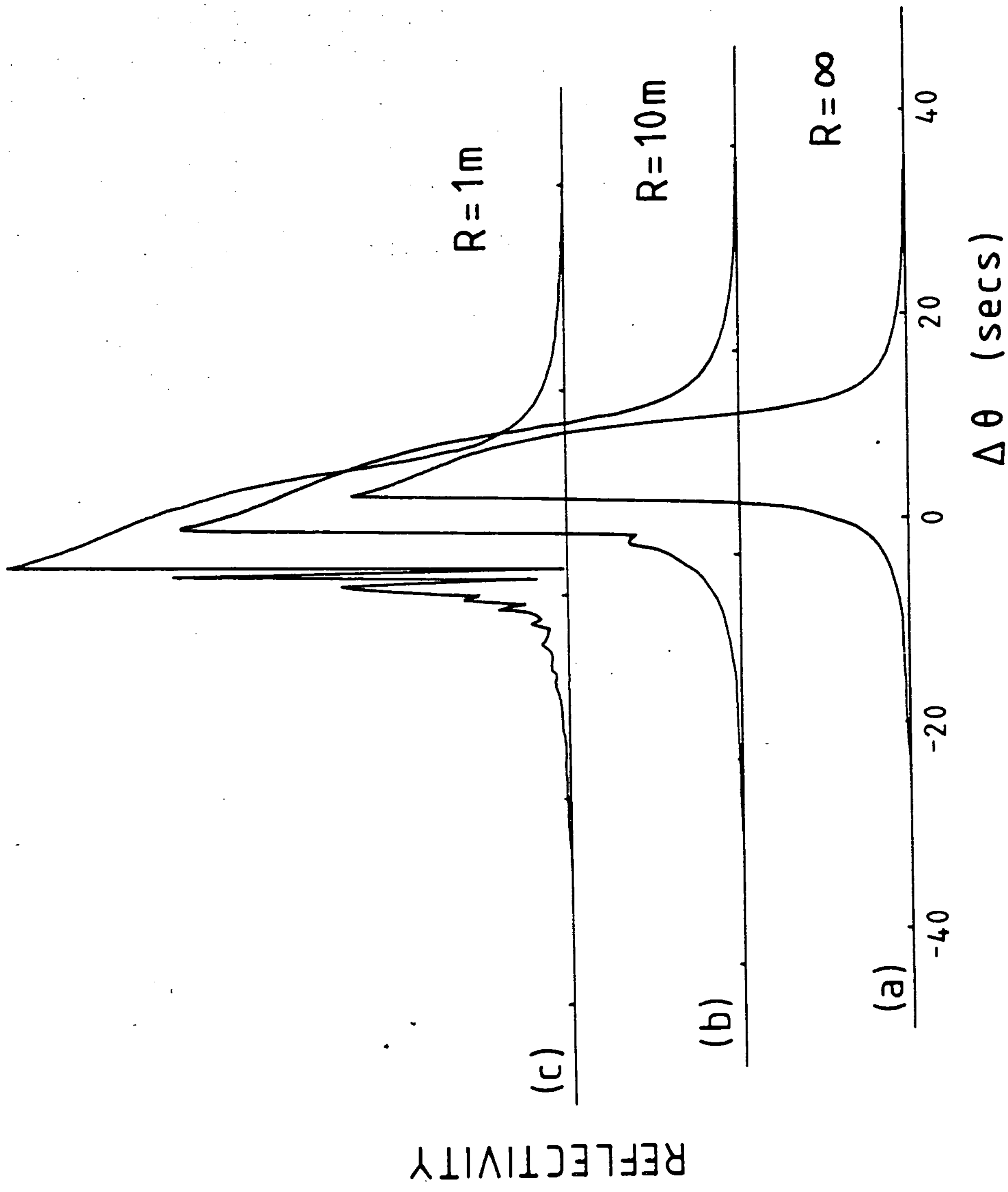


Fig. 3.3 004 CuK$\alpha_1$ single crystal reflection curves for an InP crystal with (a) a flat crystal (b) a curved crystal with radius of curvature 10m and (c) a 1m radius of curvature. The effect assumed is that of the change in angle as a function of depth.

curved sample, the angle between the beam and the reflection planes will alter as a function of depth, as shown in fig. 3.2. This misorientation is given by

$$\Delta\theta = \Delta t \cot\theta \;/\; R \qquad (3.61)$$

where R is the radius of curvature of the sample. We can calculate the reflection profile from such a crystal by dividing it into a number of laminae in which the Bragg angle is taken to be constant. Dividing a 1mm thick crystal into laminae of 1 µm thickness 004 CuK$\alpha_1$ reflection curves for a flat crystal, a 10m curved crystal and a 1m curved crystal are shown in fig. 3.3. We note that the reflection profile is both broadened and, for the highly curved crystal, Bragg case Pendellosung fringes are visible. However, since the majority of crystals have curvatures greater than 1m this effect is not too important. Generally this effect is not so important as that reported for neutron diffraction (Klar and Rustichelli, 1973), due to the stronger absorption of X-rays by the crystal.

For X-ray diffraction the most important effect is due to the change in Bragg angle along the width of incident beam, as shown in fig. 3.4. For a particular angle at one extreme of the beam there will be an intensity contribution to the overall curve from all angles up to that subtended at the other extreme of the beam. Approximately, we find

$$\Delta\theta = W \;/\; (R \sin\theta) \qquad (3.62)$$

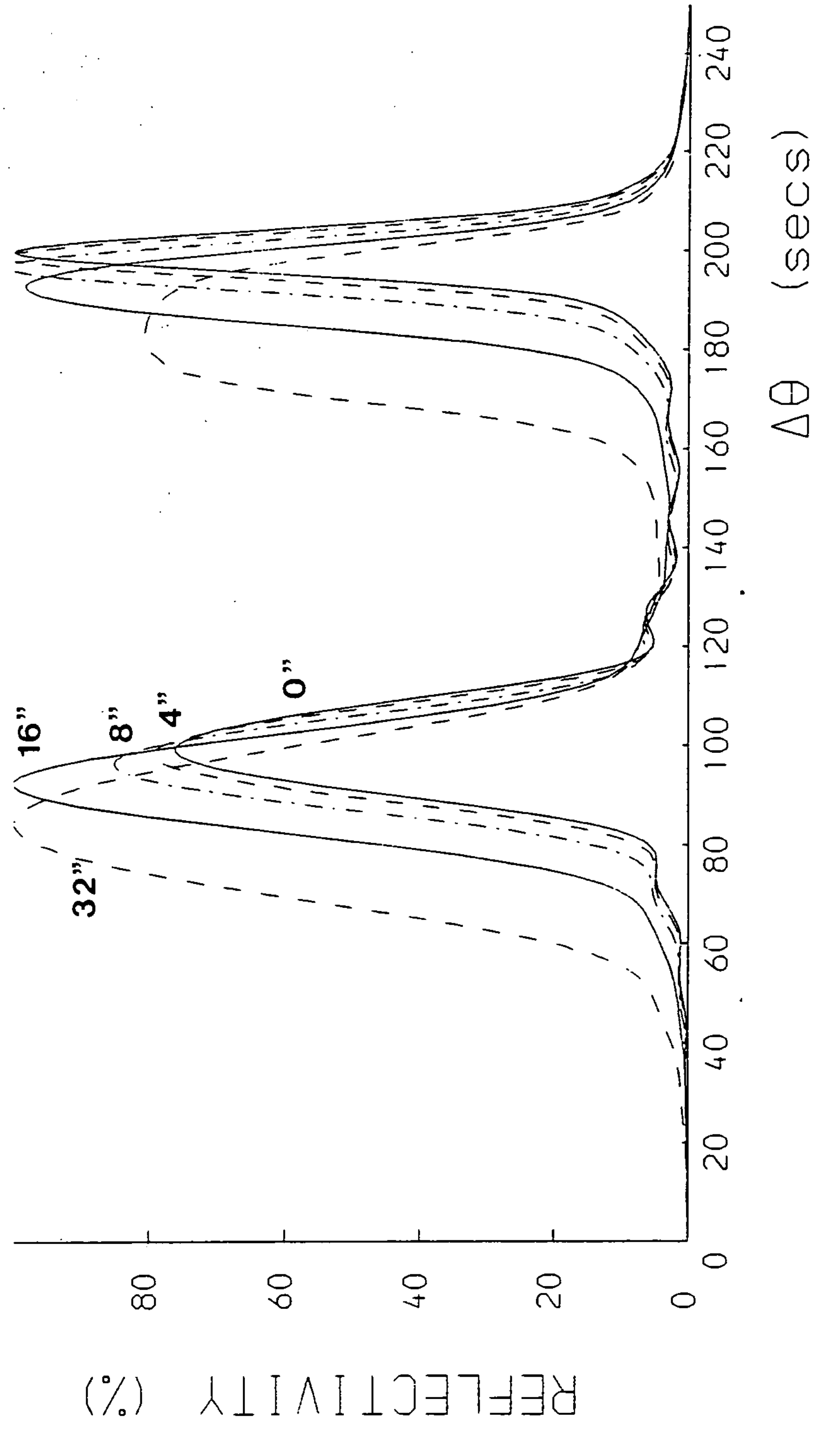


Fig. 3.5 The effect of sample curvature, assuming the effect due to beam width, on the 004 CuK$\alpha_1$ rocking curve from a sample with a 400 ppm mismatched 1μm thick GaInAs layer.

where W is the width of the incident beam. For a 004 reflection with CuK$\alpha_1$ radiation from a 10m radius of curvature InP crystal this change in angle is approximately 20 arc secs for a 0.5mm wide beam, clearly a large angle compared to the 9 second theoretical half width of the reflection. Fig. 3.5 shows the effect of decreasing the radius of curvature on the rocking curve for such a crystal plus a 400 ppm mismatched 1 µm thick GaInAs layer. Not only are the peaks broadened but the ratio of peak heights alters due to the difference in half widths of the peaks. The values of peak half widths and the ratio of peak heights are also given in Table 3.1.

**TABLE 3.1**

**Values of the substrate and layer peak half widths and ratio of peak heights for an increasing sample curvature on a sample with a 1 µm, 400 ppm mismatch GaInAs layer. The 004 reflection is used with CuK$\alpha_1$ radiation.**

| $\Delta\theta$ due to sample curvature (secs) | $\Delta\theta_{\frac{1}{2}}$ of the substrate peak (secs) (±0.4) | $\Delta\theta_{\frac{1}{2}}$ of the layer peak (secs) (±0.4) | Ratio heights of substrate:layer peaks (±0.01) |
|---|---|---|---|
| 0 | 11.0 | 17.6 | 0.76 |
| 4 | 12.2 | 21.1 | 0.79 |
| 8 | 13.0 | 22.2 | 0.85 |
| 16 | 18.4 | 23.7 | 1.01 |
| 32 | 33.0 | 35.2 | 1.24 |

## 3.5 Computation of Rocking Curves

Since the susceptibilities $\chi_0$, $\chi_h$ and $\chi_{\bar{h}}$ and the reflectivity X are complex, Fortran was chosen as the programming language due to the ease of handling complex arithmetic. Unfortunately, Fortran is not well structured so Pascal was chosen to programme the interactive sections. Since Pascal also provides enumerated variables the accessing of the analytical approximation and dispersion correction data, required to calculate the structure factors for each material, was also simplified. The Fortran programme was written to calculate the reflectivities and convolution using the structure factors, Bragg angles, lattice parameters and layer thicknesses generated by the Pascal programme. This data was passed between the programmes in a data file. This technique is particularly useful as the Fortran programme requires the largest amount of CPU time and can be left to run in 'Batch' mode where large numbers of layers are present.

### 3.5.1 The Pascal Programme

This programme handles all the interactive sections of the overall calculation, enabling the layer parameters to be entered and easily edited. Once all the reflection and layer parameters have been entered the structure factors are calculated using the analytical approximation to provide the scattering factors (International Tables, 1974). The dispersion corrections are calculated at the required wavelength using a polynomial approximation to the tabulated values (International Tables).

Operation of the programme is outlined by the flow

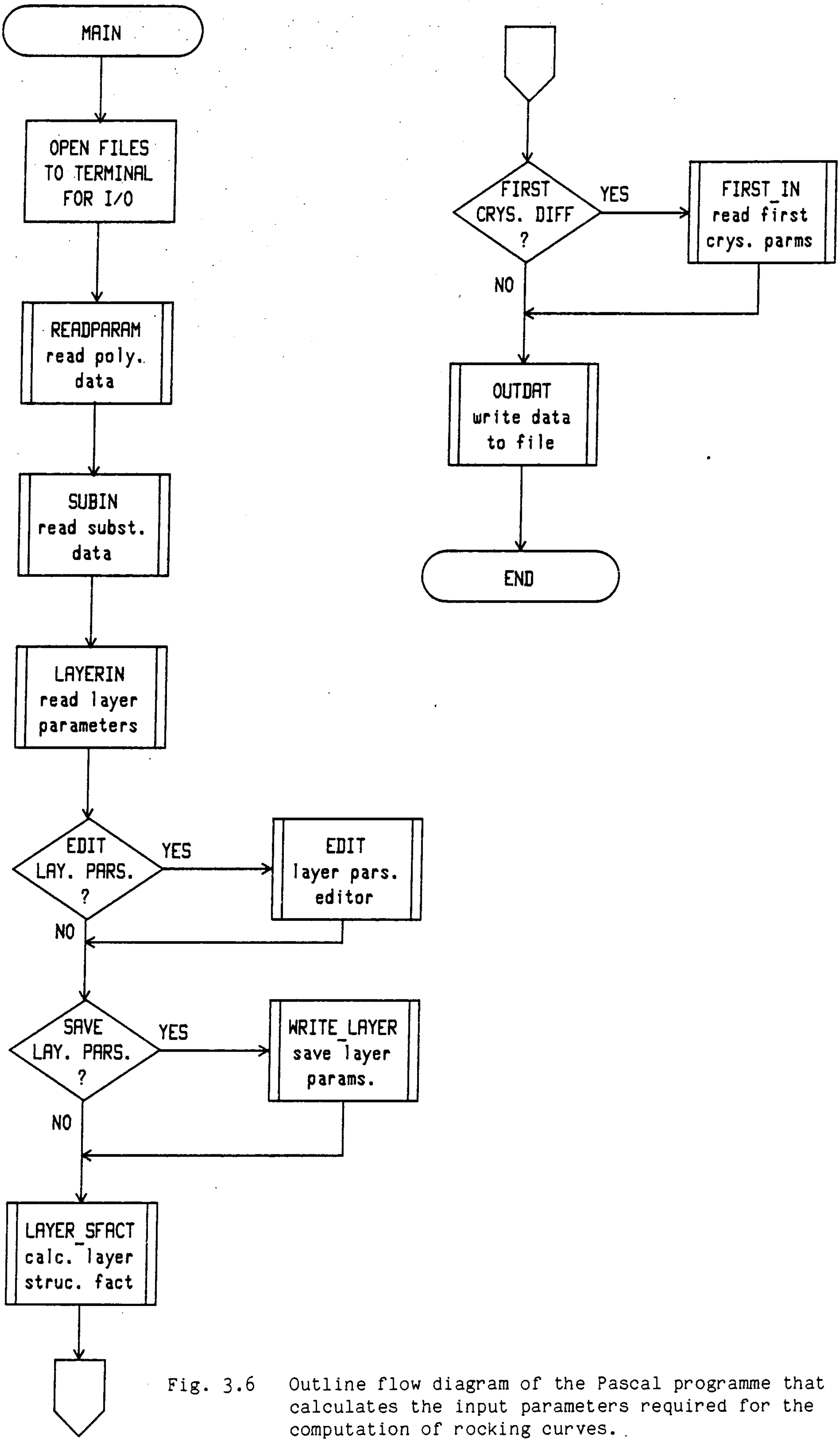

MAIN
OPEN FILES TO TERMINAL FOR I/O
READPARAM read poly. data
SUBIN read subst. data
LAYERIN read layer parameters
EDIT LAY. PARS. ?
YES
EDIT layer pars. editor
NO
SAVE LAY. PARS. ?
YES
WRITE_LAYER save layer params.
NO
LAYER_SFACT calc. layer struc. fact
FIRST CRYS. DIFF ?
YES
FIRST_IN read first crys. parms
NO
OUTDAT write data to file
END

Fig. 3.6 Outline flow diagram of the Pascal programme that calculates the input parameters required for the computation of rocking curves.

diagram, fig. 3.6. Appendix A contains a complete listing of the programme. The analytical and dispersion approximation data is contained within a data file to allow additions for extra elements to be easily made. The material names entered by the user are converted into sets of elements by comparing the names with those expected for the various III-V combinations. If the material name is not recognised it is asked for again. Detection of an element within the set is then easily made. Thus code numbers do not need to be used for each material, hence making altering the programme straightforward.

In order to avoid entering the individual laminae/layers for graded and multiquantum wells, routines are provided to calculate these parameters. Once this data has been generated it can still be edited to allow for any additional variations. The layer parameters can also be saved in a disc file to be reused in a future calculation.

The layer editing routine allows any of the layer parameters to be altered and additionally for layers to be deleted or extra layers inserted.

For graded composition layers with a non linear variation with depth the layer parameters are assumed to lie on a variable radius curve. The variation from non linearity is entered as a percentage value with 100% corresponding to a curve with radius equal to the layer thickness and 0% corresponding to a linear variation.

Fig. 3.7 shows complete detailed flow diagrams of the

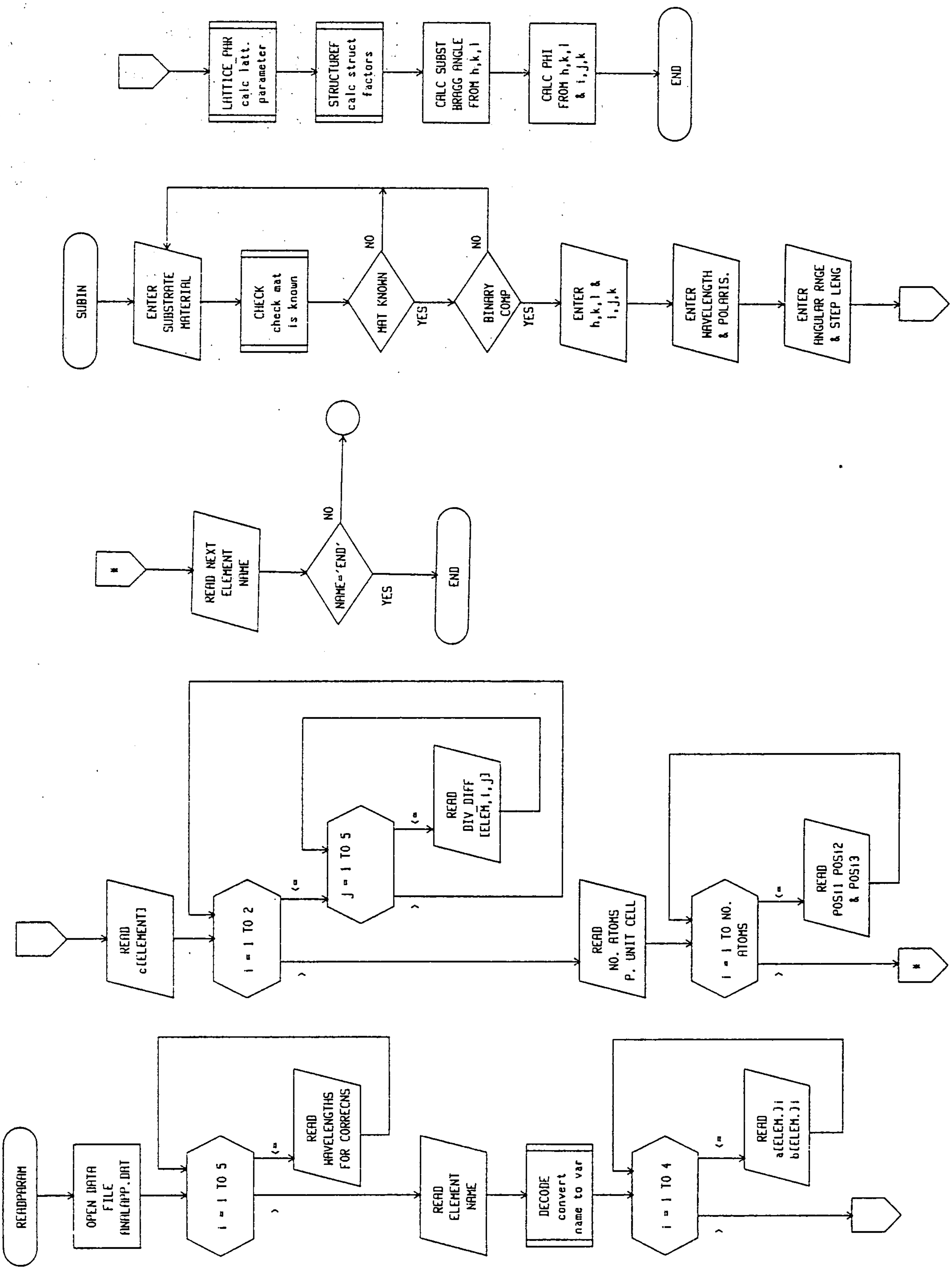
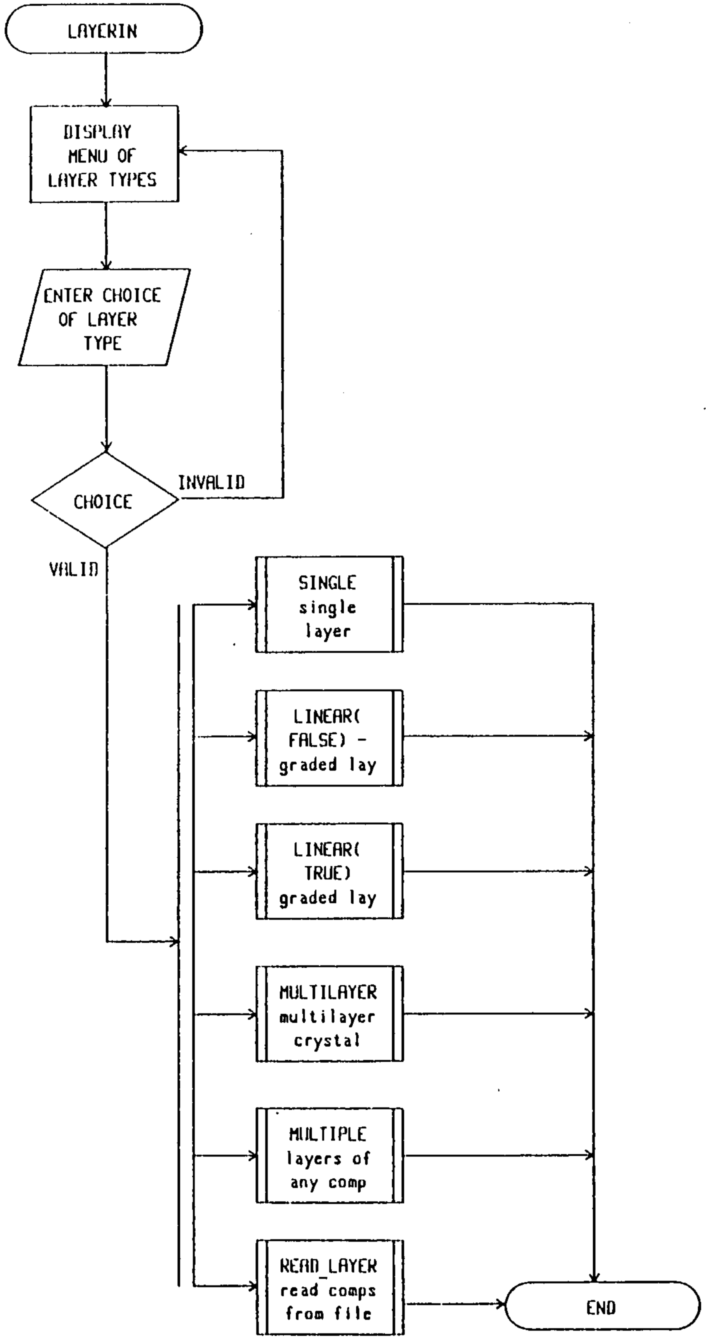
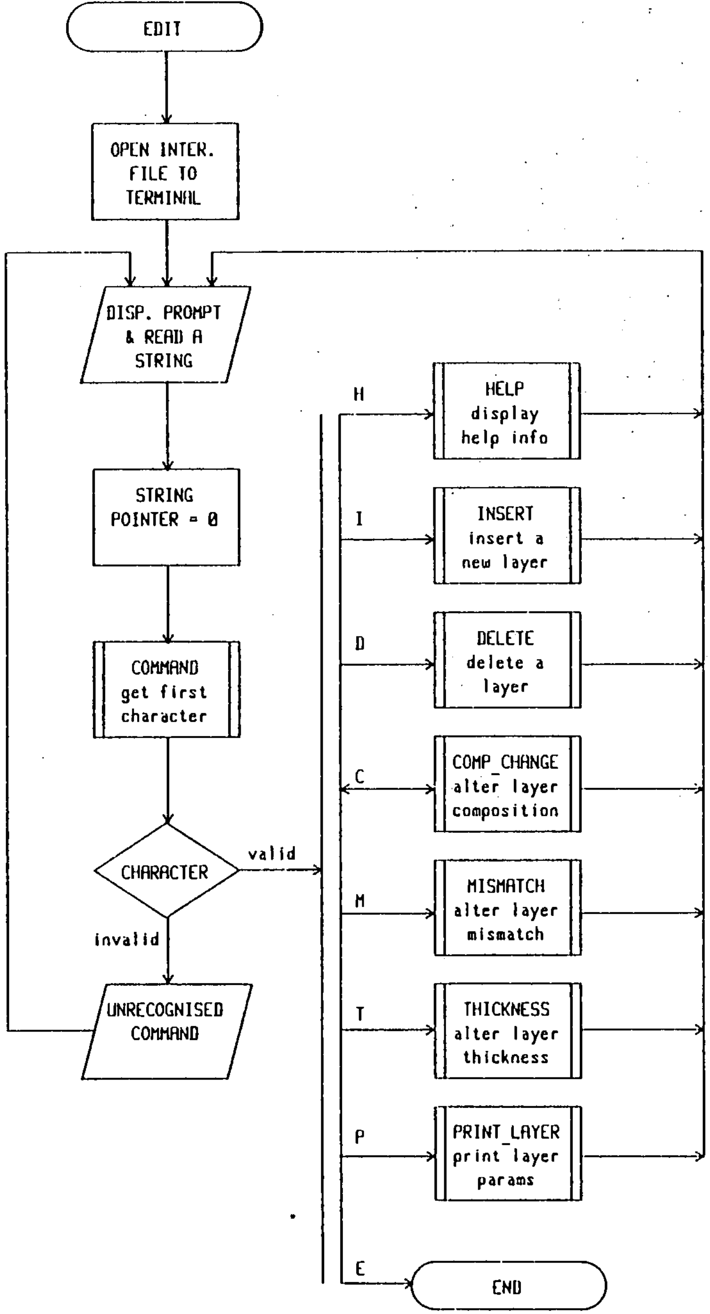
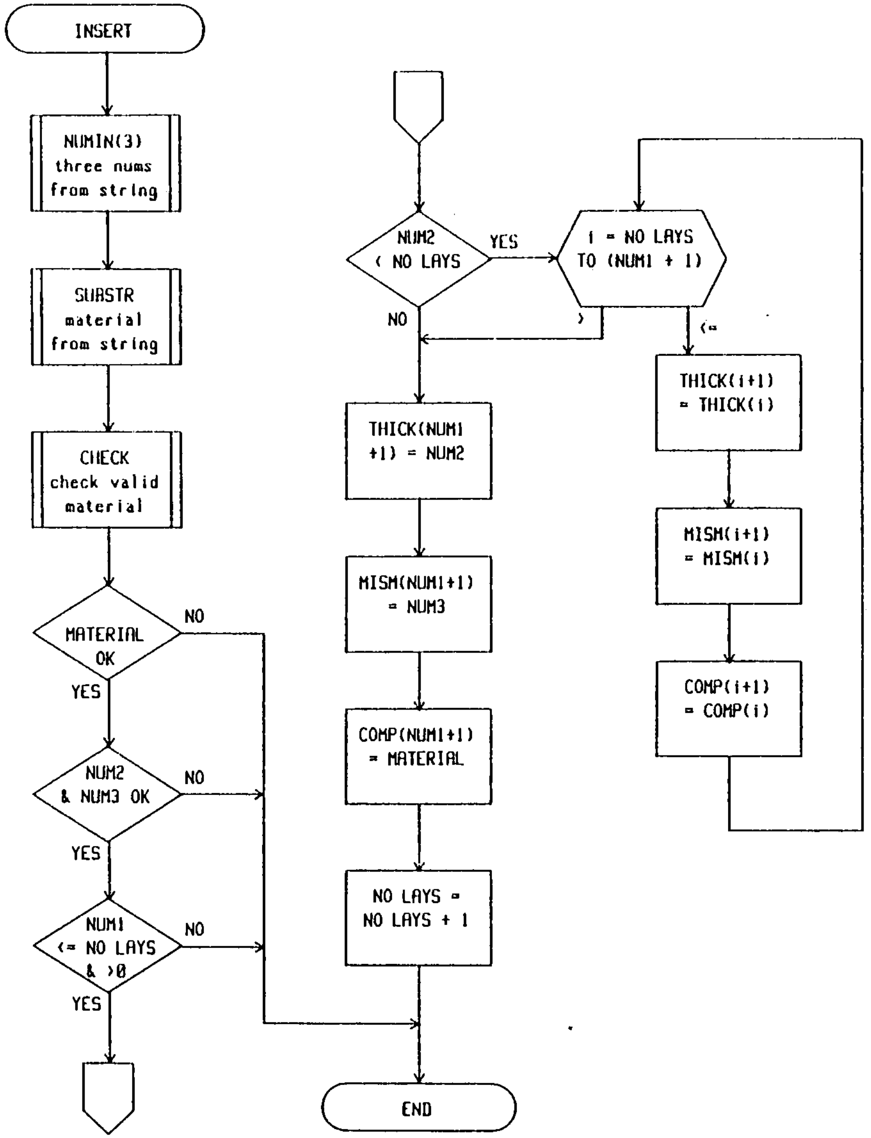
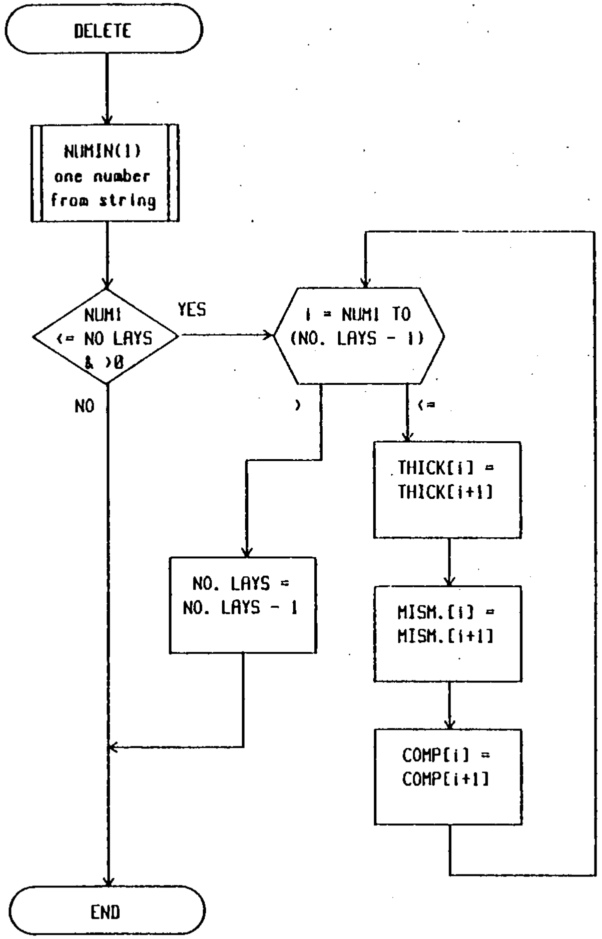


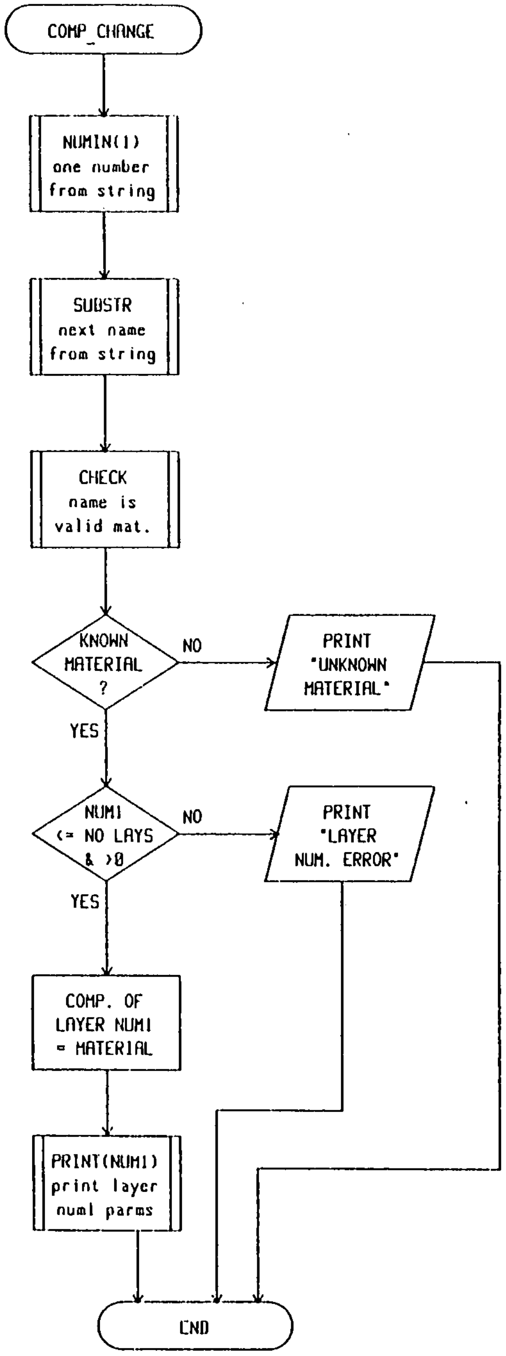
Fig. 3.7   Detailed flow diagram of the Pascal programme.

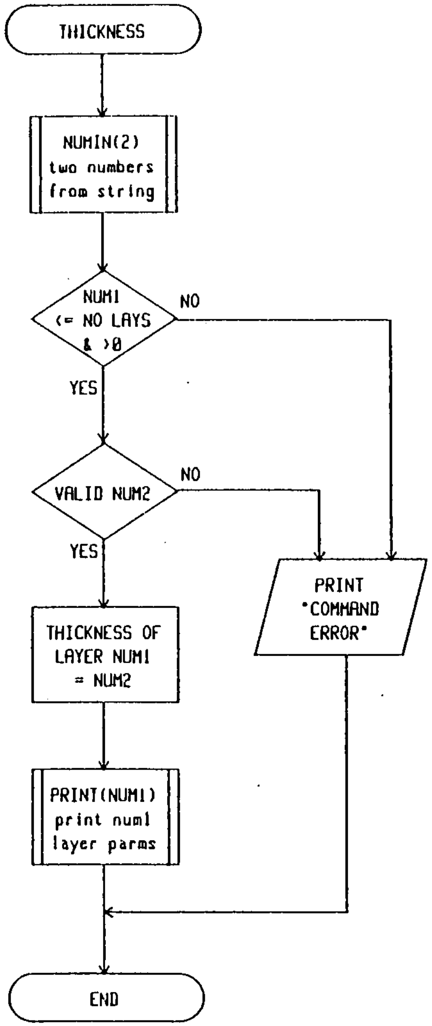

LAYERIN
DISPLAY MENU OF LAYER TYPES
ENTER CHOICE OF LAYER TYPE
CHOICE
INVALID
VALID
SINGLE single layer
LINEAR( FALSE) - graded lay
LINEAR( TRUE) graded lay
MULTILAYER multilayer crystal
MULTIPLE layers of any comp
READ_LAYER read_comps from file
END

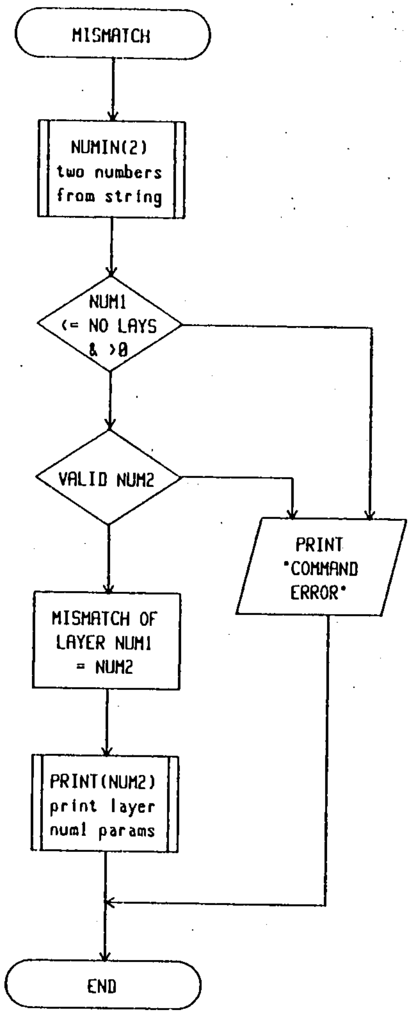

EDIT
OPEN INTER. FILE TO TERMINAL
DISP. PROMPT & READ A STRING
STRING POINTER = 0
COMMAND get first character
CHARACTER
valid
invalid
UNRECOGNISED COMMAND
H
HELP display help info
I
INSERT insert a new layer
D
DELETE delete a layer
C
COMP_CHANGE alter layer composition
M
MISMATCH alter layer mismatch
T
THICKNESS alter layer thickness
P
PRINT_LAYER print_layer params
E
END

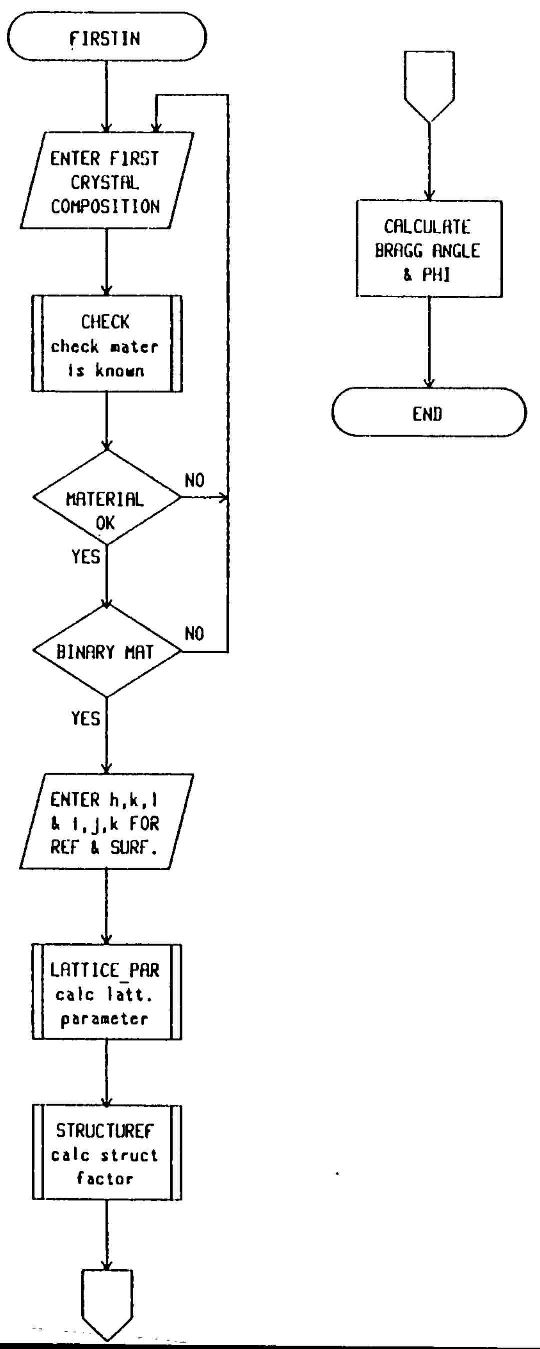

INSERT
NUMIN(3)
three nums
from string
SUBSTR
material
from string
CHECK
check valid
material
MATERIAL
OK
NO
YES
NUM2
& NUM3 OK
NO
YES
NUM1
<= NO LAYS
& >0
NO
YES
NUM2
< NO LAYS
YES
NO
i = NO LAYS
TO (NUM1 + 1)
>
<=
THICK(i+1)
= THICK(i)
MISM(i+1)
= MISM(i)
COMP(i+1)
= COMP(i)
THICK(NUM1
+1) = NUM2
MISM(NUM1+1)
= NUM3
COMP(NUM1+1)
= MATERIAL
NO LAYS =
NO LAYS + 1
END

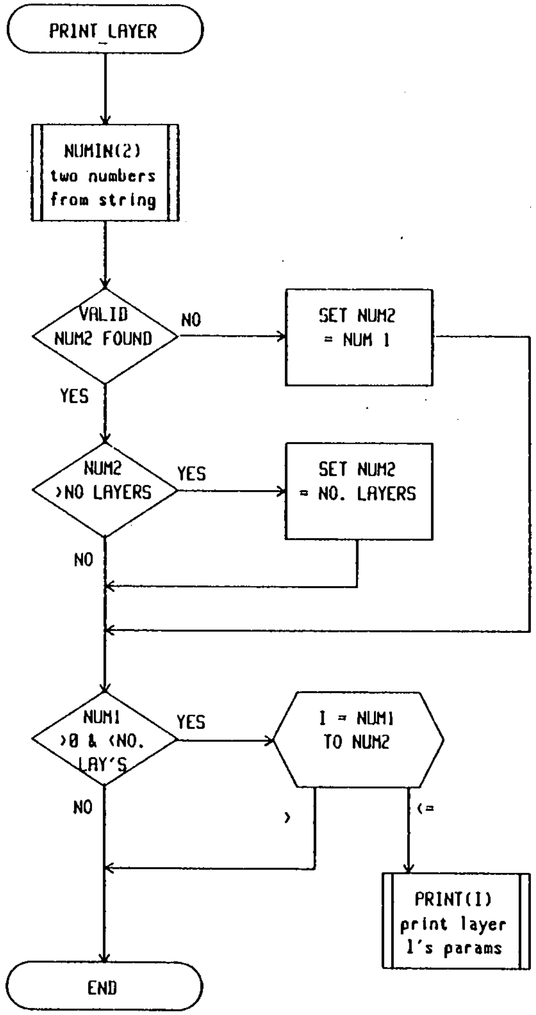

DELETE
NUMIN(1)
one number
from string
NUM1
<= NO LAYS
& >0
YES
NO
i = NUM1 TO
(NO. LAYS - 1)
>
<=
THICK[i] =
THICK[i+1]
MISM.[i] =
MISM.[i+1]
COMP[i] =
COMP[i+1]
NO. LAYS =
NO. LAYS - 1
END

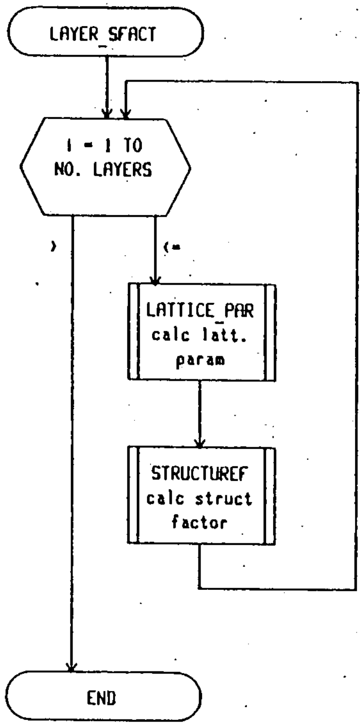

COMP_CHANGE
NUMIN(1) one number from string
SUBSTR next name from string
CHECK name is valid mat.
KNOWN MATERIAL ?
NO
PRINT "UNKNOWN MATERIAL"
YES
NUM1 <= NO LAYS & >0
NO
PRINT "LAYER NUM. ERROR"
YES
COMP. OF LAYER NUM1 = MATERIAL
PRINT(NUM1) print layer num1 parms
END

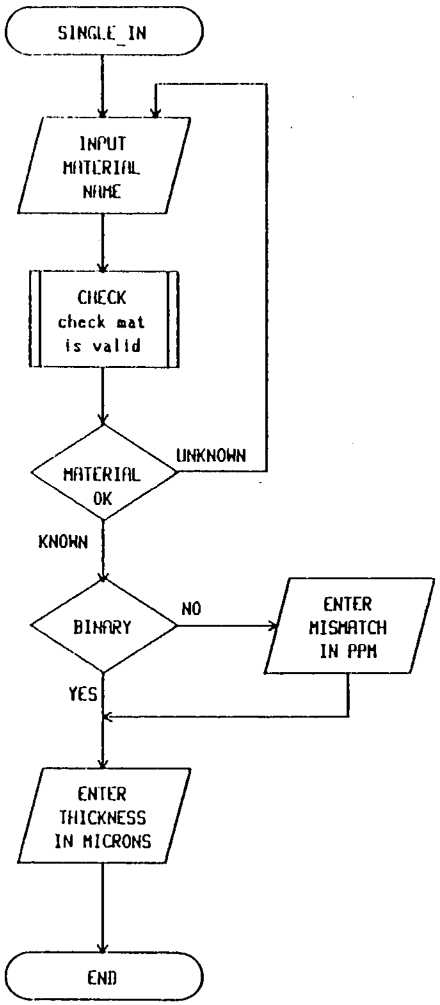

THICKNESS
NUMIN(2) two numbers from string
NUM1 <= NO LAYS & >0
NO
YES
VALID NUM2
NO
YES
PRINT "COMMAND ERROR"
THICKNESS OF LAYER NUM1 = NUM2
PRINT(NUM1) print num1 layer parms
END

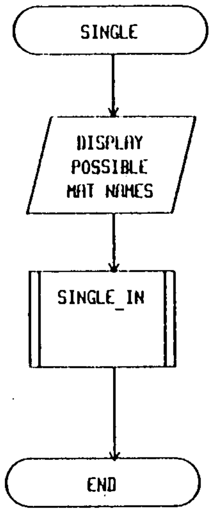

MISMATCH
NUMIN(2) two numbers from string
NUM1 <= NO LAYS & >0
VALID NUM2
PRINT "COMMAND ERROR"
MISMATCH OF LAYER NUM1 = NUM2
PRINT(NUM2) print layer num1 params
END

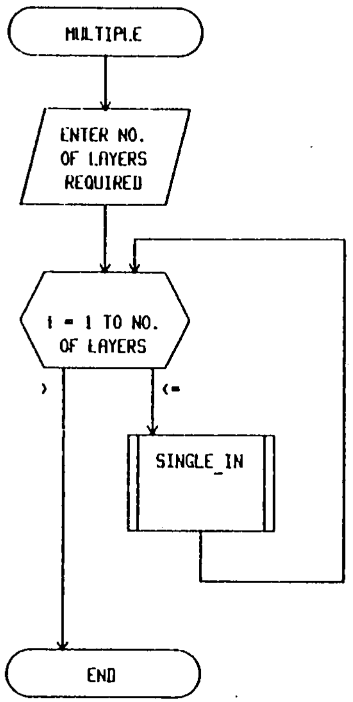

FIRSTIN
ENTER FIRST CRYSTAL COMPOSITION
CHECK check mater is known
MATERIAL OK
NO
YES
BINARY MAT
NO
YES
ENTER h,k,l & i,j,k FOR REF & SURF.
LATTICE_PAR calc latt. parameter
STRUCTUREF calc struct factor
CALCULATE BRAGG ANGLE & PHI
END

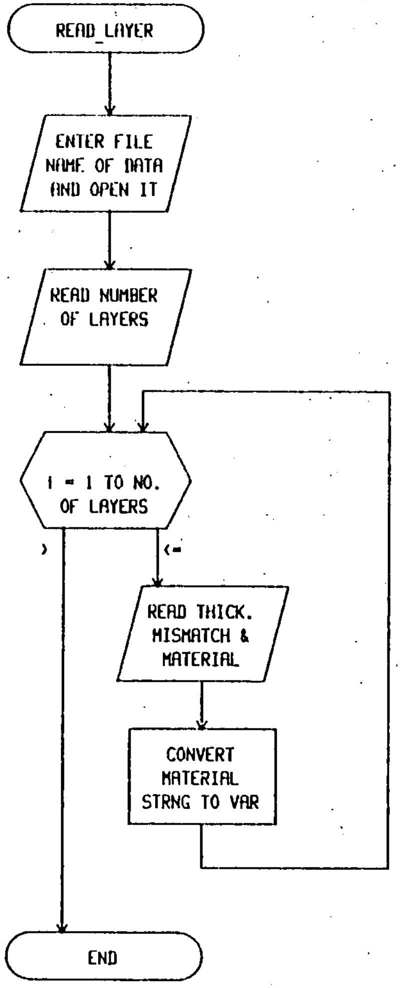

PRINT_LAYER
NUMIN(2) two numbers from string
VALID NUM2 FOUND
NO
SET NUM2 = NUM 1
YES
NUM2 >NO LAYERS
YES
SET NUM2 = NO. LAYERS
NO
NUM1 >0 & <NO. LAY'S
YES
I = NUM1 TO NUM2
>
<=
PRINT(I) print layer I's params
NO
END

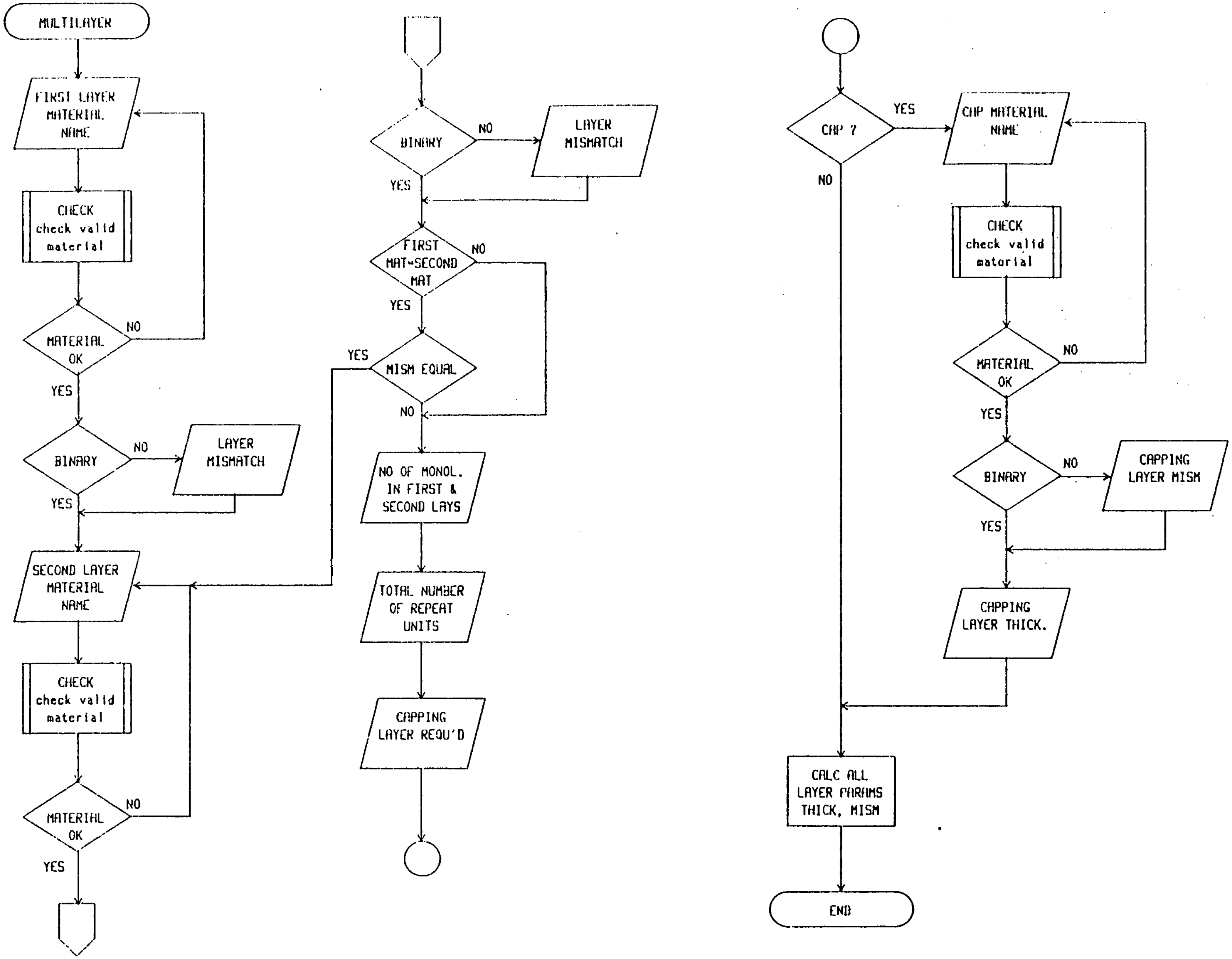

LAYER_SFACT
I = 1 TO NO. LAYERS
>
<=
LATTICE_PAR calc latt. param
STRUCTUREF calc struct factor
END

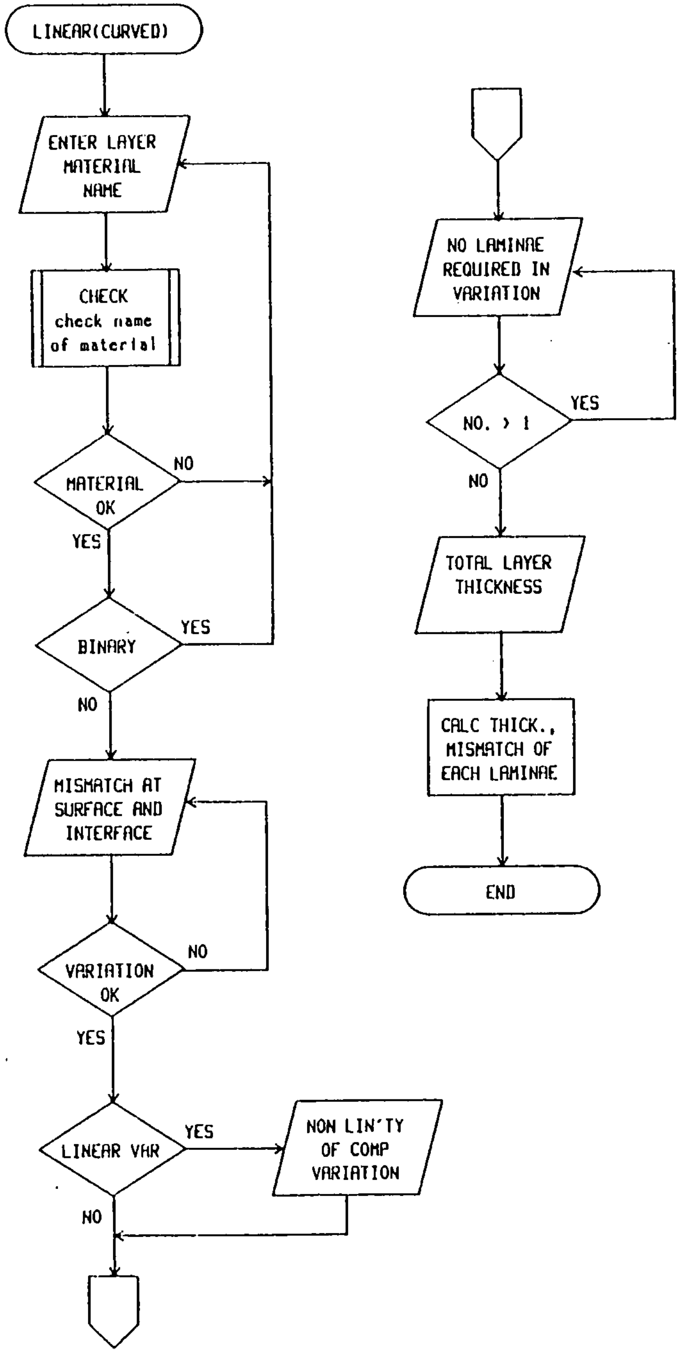

SINGLE_IN
INPUT MATERIAL NAME
CHECK check mat is valid
MATERIAL OK
UNKNOWN
KNOWN
BINARY
NO
ENTER MISMATCH IN PPM
YES
ENTER THICKNESS IN MICRONS
END

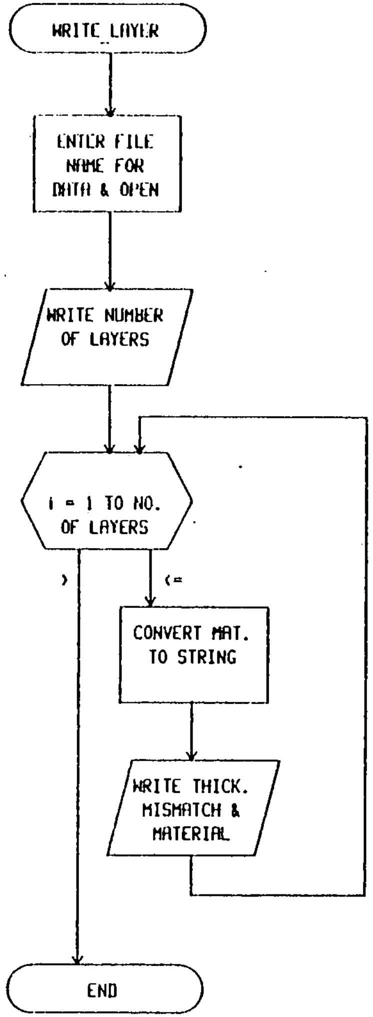

SINGLE
DISPLAY POSSIBLE MAT NAMES
SINGLE_IN
END

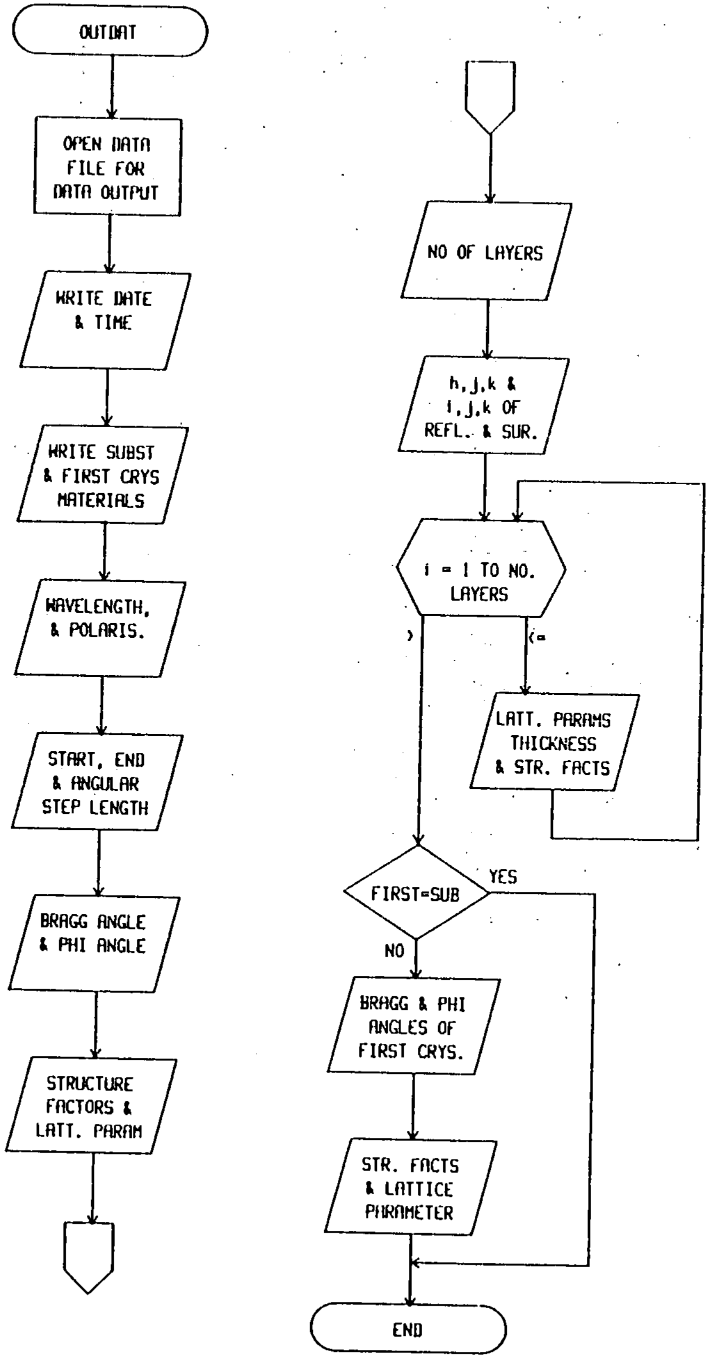

MULTIPLE
ENTER NO. OF LAYERS REQUIRED
I = 1 TO NO. OF LAYERS
>
<=
SINGLE_IN
END


READ_LAYER
ENTER FILE NAME OF DATA AND OPEN IT
READ NUMBER OF LAYERS
I = 1 TO NO. OF LAYERS
>
<=
READ THICK. MISMATCH & MATERIAL
CONVERT MATERIAL STRNG TO VAR
END


MULTILAYER
FIRST LAYER MATERIAL NAME
CHECK check valid material
MATERIAL OK
NO
YES
BINARY
NO
LAYER MISMATCH
YES
SECOND LAYER MATERIAL NAME
CHECK check valid material
MATERIAL OK
NO
YES
BINARY
NO
LAYER MISMATCH
YES
FIRST MAT=SECOND MAT
NO
YES
MISM EQUAL
YES
NO
NO OF MONOL. IN FIRST & SECOND LAYS
TOTAL NUMBER OF REPEAT UNITS
CAPPING LAYER REQU'D
CAP ?
YES
NO
CAP MATERIAL NAME
CHECK check valid material
MATERIAL OK
NO
YES
BINARY
NO
CAPPING LAYER MISM
YES
CAPPING LAYER THICK.
CALC ALL LAYER PARAMS THICK, MISM
END


LINEAR(CURVED)
ENTER LAYER MATERIAL NAME
CHECK check name of material
MATERIAL OK
NO
YES
BINARY
YES
NO
MISMATCH AT SURFACE AND INTERFACE
VARIATION OK
NO
YES
LINEAR VAR
YES
NON LIN'TY OF COMP VARIATION
NO
NO LAMINAE REQUIRED IN VARIATION
NO. > 1
YES
NO
TOTAL LAYER THICKNESS
CALC THICK., MISMATCH OF EACH LAMINAE
END


WRITE_LAYER
ENTER FILE NAME FOR DATA & OPEN
WRITE NUMBER OF LAYERS
i = 1 TO NO. OF LAYERS
>
<=
CONVERT MAT. TO STRING
WRITE THICK. MISMATCH & MATERIAL
END


OUTDAT
OPEN DATA FILE FOR DATA OUTPUT
WRITE DATE & TIME
WRITE SUBST & FIRST CRYS MATERIALS
WAVELENGTH, & POLARIS.
START, END & ANGULAR STEP LENGTH
BRAGG ANGLE & PHI ANGLE
STRUCTURE FACTORS & LATT. PARAM
NO OF LAYERS
h,j,k & i,j,k OF REFL. & SUR.
i = 1 TO NO. LAYERS
>
<=
LATT. PARAMS THICKNESS & STR. FACTS
FIRST=SUB
YES
NO
BRAGG & PHI ANGLES OF FIRST CRYS.
STR. FACTS & LATTICE PARAMETER
END

individual routines, and their operation is fairly straightforward. In particular the editing routine operates by inputting one command string, from which the individual functions remove their required number of variables, delimitted by spaces. This method allows a variable number of parameters to be entered in the command string while still retaining a single line entry system.

At all stages of the programme unrecognised materials are detected and the user prompted for a new material. Additionally, ternary and higher alloys cannot be selected as substrates and composition variations will not be allowed for binary alloys. The percentage composition of each element in a chosen material is determined from the mismatch, according to the equations given in chapter 1. For quaternary alloys the band gap is also used. The material structure factors are calculated by multiplying the structure factor for the elements by their percentage in the material. This will be valid only if the various elements in the material are completely randomly distributed. Finally, the parameters required by the Fortran programme are output for every layer.

### 3.5.2 The Fortran Programme

Operation of this programme is outlined by the flow diagrams shown in fig 3.8. Appendix A contains a complete listing of the programme.

Once the data has been read from the data file the

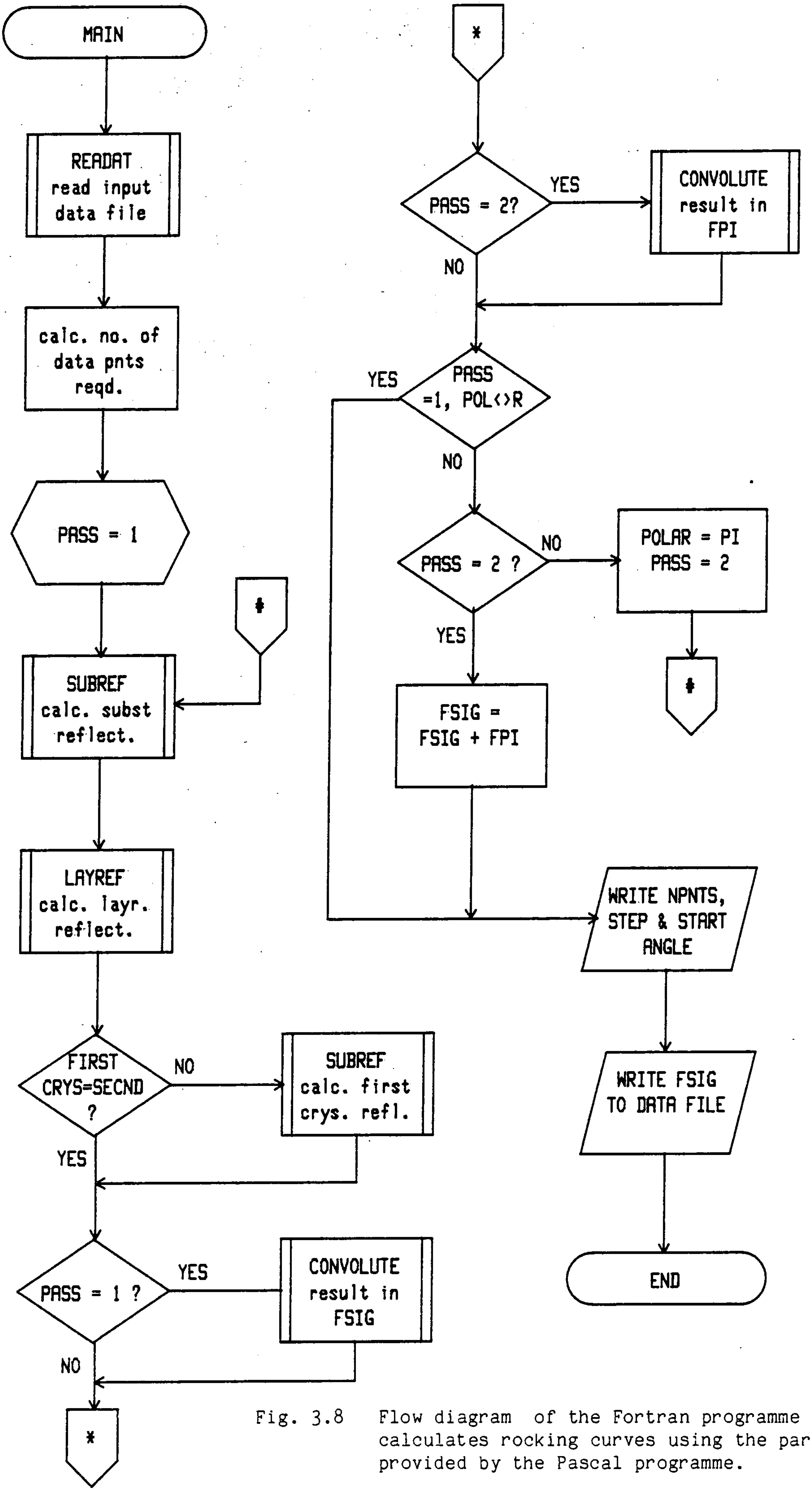


Fig. 3.8 Flow diagram of the Fortran programme that calculates rocking curves using the parameters provided by the Pascal programme.

reflectivities for the first and second crystals are calculated, over the range of angles required. If the first crystal is the same as the substrate of the second crystal the substrate reflectivity is used for the first crystal reflectivity. If the first crystal reflection is different from that of the second crystal, giving a non parallel arrangement, dispersion should be taken into account. However, the programme does not account for this effect making it only valid for the synchrotron source where the beam divergence is so small that dispersion can be neglected. The reflectivities are calculated using equations (3.58) and (3.59). This programme requires no knowledge of the materials being used and can therefore be used to calculate rocking curves from any material, provided that it is supplied with the appropriate structure factors and lattice parameters.

For each layer in the second crystal the deviation parameter $\alpha_h$ is calculated, taking into account the difference in Bragg and phi angles for the layer and substrate. This is necessary since the range of angles specified are taken relative to the substrate Bragg angle. Thus, if we take $\Delta\theta_{S,L}$ to be the angular deviation from the exact Bragg angle for the substrate and layer respectively, we have

for low angles of incidence

$$\Delta\theta_L = \Delta\theta_S + \theta_{BS} - \theta_{BL} - \varphi_S + \varphi_L \qquad (3.62)$$

and for high angles of incidence

$$\Delta\theta_L = \Delta\theta_S + \theta_{BS} - \theta_{BL} + \varphi_S - \varphi_L. \qquad (3.63)$$

For ease of calculation the convolution is calculated over the same range and interval as the single crystal reflectivities, although the routines can handle both a different range and interval. The convolution is calculated according to equation (2.17). For each value of $\beta$ a multiplied curve is calculated and the area under it determined to give the reflectivity at the angle $\beta$. To calculate the multiplied curve the first crystal reflectivity at the angle $\alpha$ is multiplied by the second crystal reflectivity at the angle $(\alpha-\beta)$. If the angle $(\alpha-\beta)$ does not correspond to a data point on the other reflectivity curve linear interpolation between the nearest points is used. The curve is then generated by repeating this operation over the range of angles $\alpha$. Outside this range the reflectivities are taken to be zero.

If random polarisation is selected the entire calculation is repeated and the two convoluted curves added.

Finally, the rocking curve data is written to the output file, to be plotted by another programme. This approach allows the curve to be plotted on different scales without recalculating the curve. Additionally, the effect of sample curvature is added by the plotting programme, allowing the effect of different sample curvatures to be observed without recalculating the curve.

The plotting programme is straightforward and hence will not be described in detail. A complete listing is contained in Appendix A. Rocking curve data is read from a file and plotted using GHOST routines. The plotfile produced by these routines can be plotted either on a graphics terminal or on a hardcopy plotter.

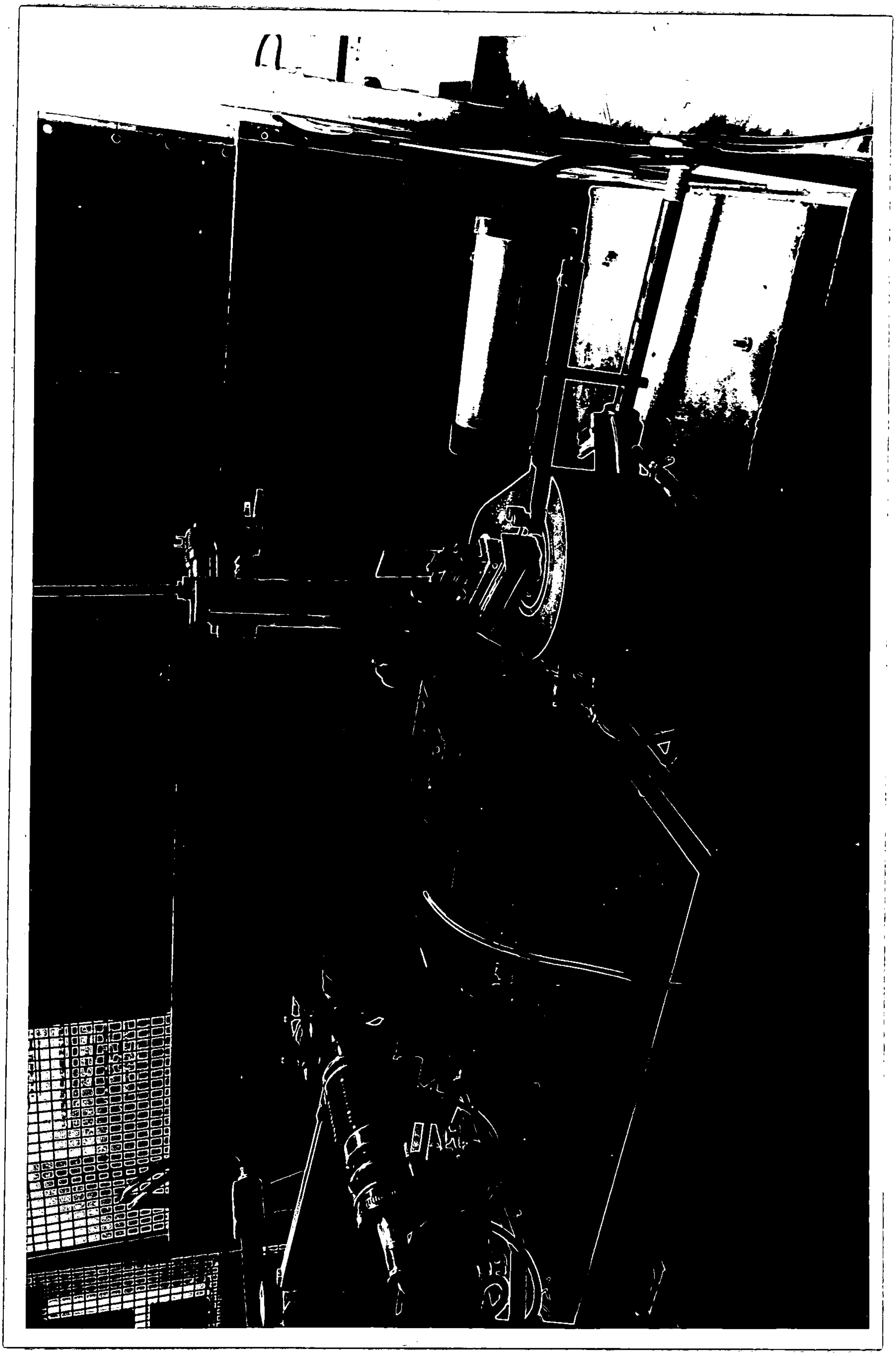


Fig. 4.1 Photograph of the six inch double crystal camera and the GX-6 x-ray generator at Durham. The camera is rigidly mounted on a separate table while the lm collimator is equipped with an optical light source and adjustable slits.

# CHAPTER 4

# EXPERIMENTAL TECHNIQUES AND INSTRUMENTATION

Two types of double crystal camera have been used to measure the rocking curves presented here. A six inch axial separation camera, similar to that described by Hart (1980) and by Meriam Abdul Gani (1982), was used at Durham in conjunction with a Marconi-Elliot GX-6 rotating anode generator and with a Philips 1010 sealed tube generator. At the SRS, Daresbury Laboratory, a twelve inch axial separation camera, as described by Bowen and Davies (1983), was used on the Port 7 station. Both of these cameras have similar axial drive mechanisms, driving the axis via a stepper motor/gearbox/micrometer combination pushing a tangential arm attached to the axis. On the larger camera several other axes are motor driven including the detector arm and the camera itself. Computer control of the cameras was implemented at Durham with a BBC microcomputer and Minicam system, and at Daresbury with a PDP11/04 minicomputer and a CAMAC system.

## 4.1 The Six Inch Camera at Durham

A photograph of the double axis camera, positioned to receive X-rays from the GX-6 generator, is shown in fig. 4.1. With 200 steps per revolution motors the drive system gives 0.177 and 0.121 arc seconds per motor step for the first and second axes respectively. The camera is rigidly

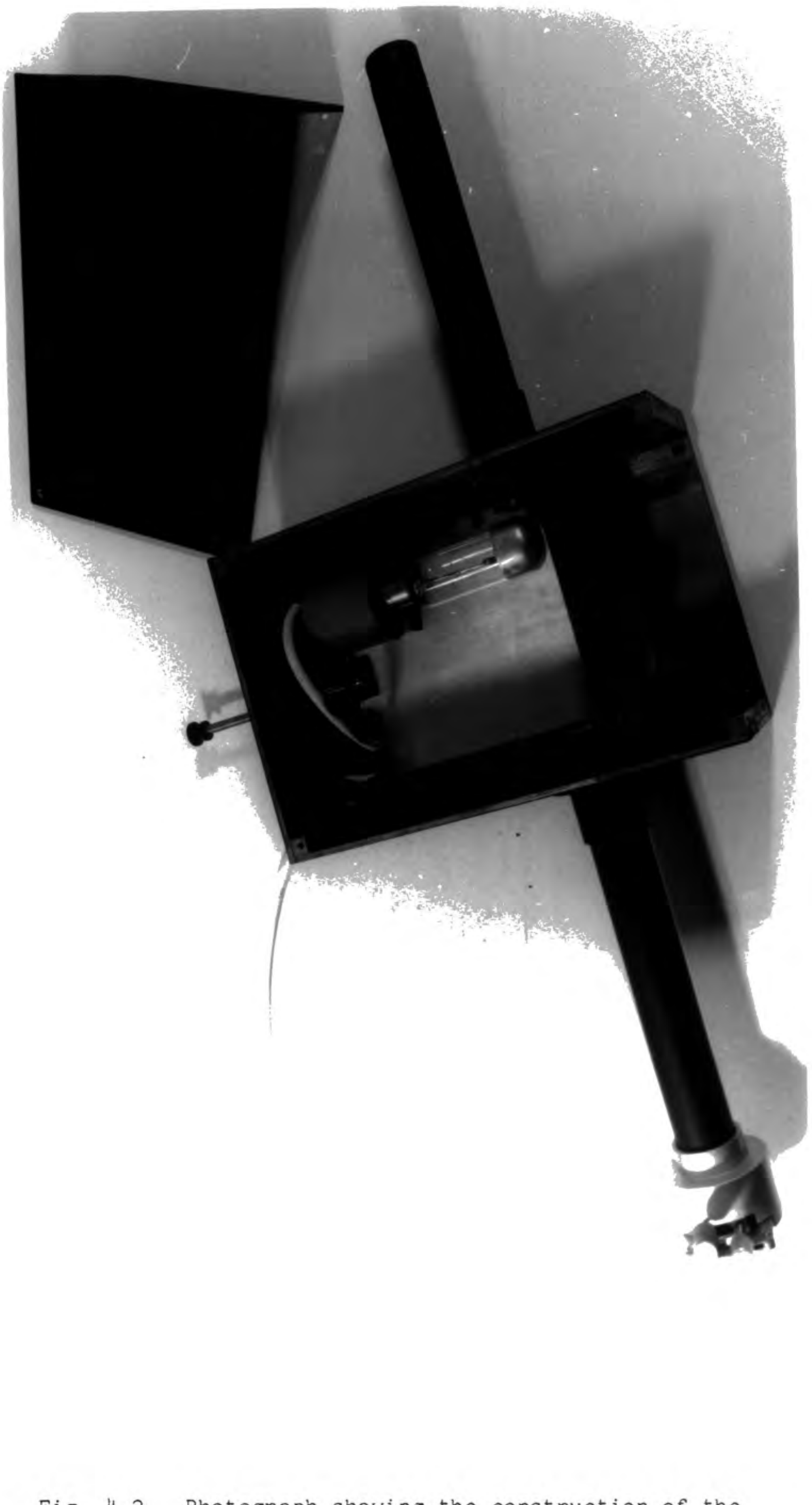


Fig. 4.2 Photograph showing the construction of the collimator and removable optical light source. The projector lamp and steel tube are mounted on a sliding mechanism allowing either the lamp or the steel tube to be positioned in line with the collimator tube.

mounted on a separate table to remove any vibration present in the generator. A collimator approximately 1m long is used, fixed at the optimum take-off angle with respect to the anode. In this arrangement the X-ray beam is horizontal so that an angled baseplate is not required for the camera, as it is with a vertically mounted sealed tube source.

In order to ease camera alignment a removable optical light source, consisting of a projector lamp mounted on a sliding arm, was fitted to the collimator. By placing the lamp in the centre of the collimator the light follows the same path as the X-rays, thus facilitating easy camera alignment. Additionally, where the crystals used are highly polished they can also be set close to the correct Bragg angle, for surface symmetric reflections, using this system. The construction of the light source is clear from the photograph shown in fig. 4.2. The cover has been removed, while the sliding part is slightly displaced from its 'home' position, where the steel tube mounted on the slide fits tightly between the two sections of collimator tube. This arrangement helps to reduce the amount of scattered radiation in the box.

### 4.1.1 Computer control

Previously this double axis camera had been controlled by a PET microcomputer and a memory mapped Minicam system (Meriam Abdul Gani, 1982). In this configuration the Minicam system appears as part of the microprocessor's memory, being controlled by code within the microcomputer. Meriam Abdul Gani (1982) has described many of the boards available. Recently the Minicam system was improved by the addition of an intelligent controller card (ICC) based on a 6502 microprocessor. This board replaces the memory mapped interface with all the control software contained on the board. Communication to a computer is provided by an IEEE488 interface. The boards are then controlled by passing binary data along this bus, an 8 bit number selecting the operation required and additional variables being passed as two 16 bit numbers.

Unfortunately, the majority of inexpensive microcomputers are not equipped with an IEEE488 interface, notably the BBC microcomputer. Thus, even though this computer offers far greater performance, including high resolution graphics, than the PET computer communication with a Minicam system is not possible. Consequently, since the BBC computer, and many other inexpensive computers, is equipped with a serial RS232 port it was appropriate to add such a serial interface to the Minicam system. A full description of the design is given below. Since the RS232 standard is designed to transmit Ascii encoded data only 7 bits are required (although 8 bits are often available), thus making the transmission of binary data not straightforward. In order to avoid this problem the Minicam firmware was rewritten to allow Ascii transmission. Full details of the new software can be found in Appendix B.

Once the BBC microcomputer and the Minicam system can communicate with each other the actual control of the double crystal camera is easily implemented. Routines similar in outline to those described by Meriam Abdul Gani (1982) were implemented, with the addition of graphics routines to display the data collection in real time. Rocking curve data can be saved on floppy disc to be redisplayed later or plotted on a digital plotter such as the HP7470A or the PIXY3. The BBC microcomputer is also connected to the laboratory's local area network thus allowing this data to be transferred onto the University's mainframe computer for analysis and comparison with simulated rocking curves. With the addition of the Workstation Rom (available from the Computer Centre, University of Sussex) the BBC microcomputer can also act as a graphics terminal to the mainframe. A full description of the control software is given in Appendix C.

X-ray intensities were measured with a standard scintillator/photomultiplier system with a Harwell rack mounted H.T. supply and amplifier. Two of the 16 bit Minicam counters were cascaded to provide one 32 bit counter which was required when long counting times were used for the measurement of weak satellite peaks from superlattices. Since good counting statistics are required to observe such satellites long counting times are needed and the counter can overflow if a main diffraction peak is encountered.

### 4.1.2 Design of the ICC RS232 Serial Interface

The fundamental requirements of the serial interface were that it should conform to the RS232 standard, operate at a variable baud rate, have a variable word format and utilise hardwire handshaking (CTS/DTR type). High transmission speeds were also required to provide fast communication. Since many computers are equipped with only one serial interface once this is connected to the Minicam system it is no longer possible to use a serial printer or plotter without needing a data switch. Therefore, any plotter being used to plot rocking curves cannot be used under programme control without user intervention. Consequently, a second serial port was included in the ICC interface to enable data to be passed from one serial port to the other. Baud rates and word formats need to be individually controlled especially if a slow speed peripheral device is being used.

Since the Minicam bus is not a true microprocessor bus, being controlled by two interface adaptors, it is limited in speed and has several vital signals missing (particularly the processor clock required for synchronisation and the $R/\overline{W}$, $\overline{IRQ}$ and $\overline{NMI}$ signals). Therefore, standard 6500/6800 peripheral IC's cannot be directly attached to this bus, hence it was decided to attach the serial interface directly to the ICC board and the microprocessor bus using the expansion connector located in the centre of the board. Since the Minicam manual does not give the pin-out of this

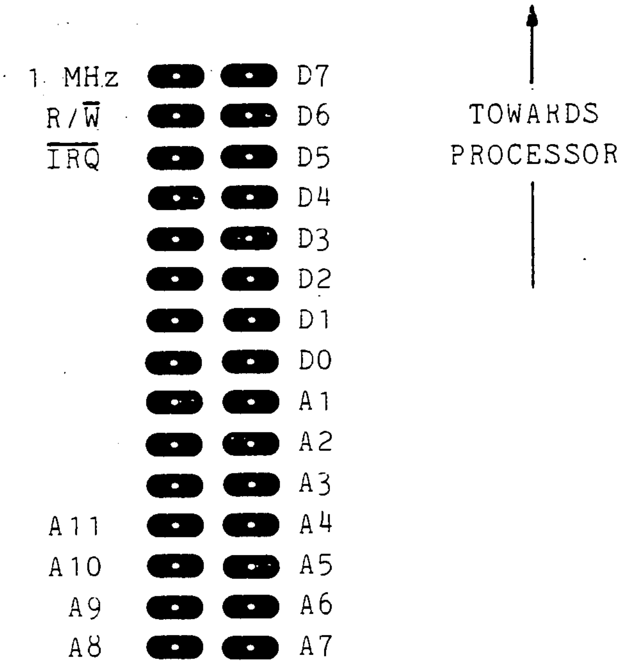


Fig. 4.3 Pin out of the connector in the centre of the Minicam ICC board. This connector allows access to the unbuffered microprocessor signals unlike the minicam bus.

connector it is shown in fig. 4.3 for reference.

The serial interface was designed around the 6850 ACIA which provides all the features required, including hardwire handshaking, and is directly compatible with the 6502 processor. The receive and transmit clock inputs were tied together and the clock signal provided by a 2.4576 MHz crystal oscillator and a 4702B programmable divider. The 4702B is specifically designed for this use and provides a rate 16 times the majority of widely used baud rates. Therefore, the 6850 should be programmed, after reset, to divide the receive and transmit clocks by 16. Two of these combinations are included in the design to provide the two independent rates for the two 6850s. In order to provide a variable word format an 8-way dual-in-line switch, buffered by a 74LS244, is provided which can be read as a memory location. This value can then be used to programme the word selection bits of the 6850s.

The main ICC board decodes addresses within the range $8000-$803F to provide the chip enable signals for the two 6522 VIAs and the 68488 GPIA. However, only the $8000, $8010 and $8020 lines are used leaving the $8030 line free. A wire patch on the main board is used to link this signal with a spare pin on the expansion connector. This line plus the address lines A2 and A3 are connected to half of a 74LS139 dual 2 to 4 line decoder to provide the three enable lines for the two 6850s and the 74LS244. Thus the addresses of these chips are $8030, $8034 and $8038.

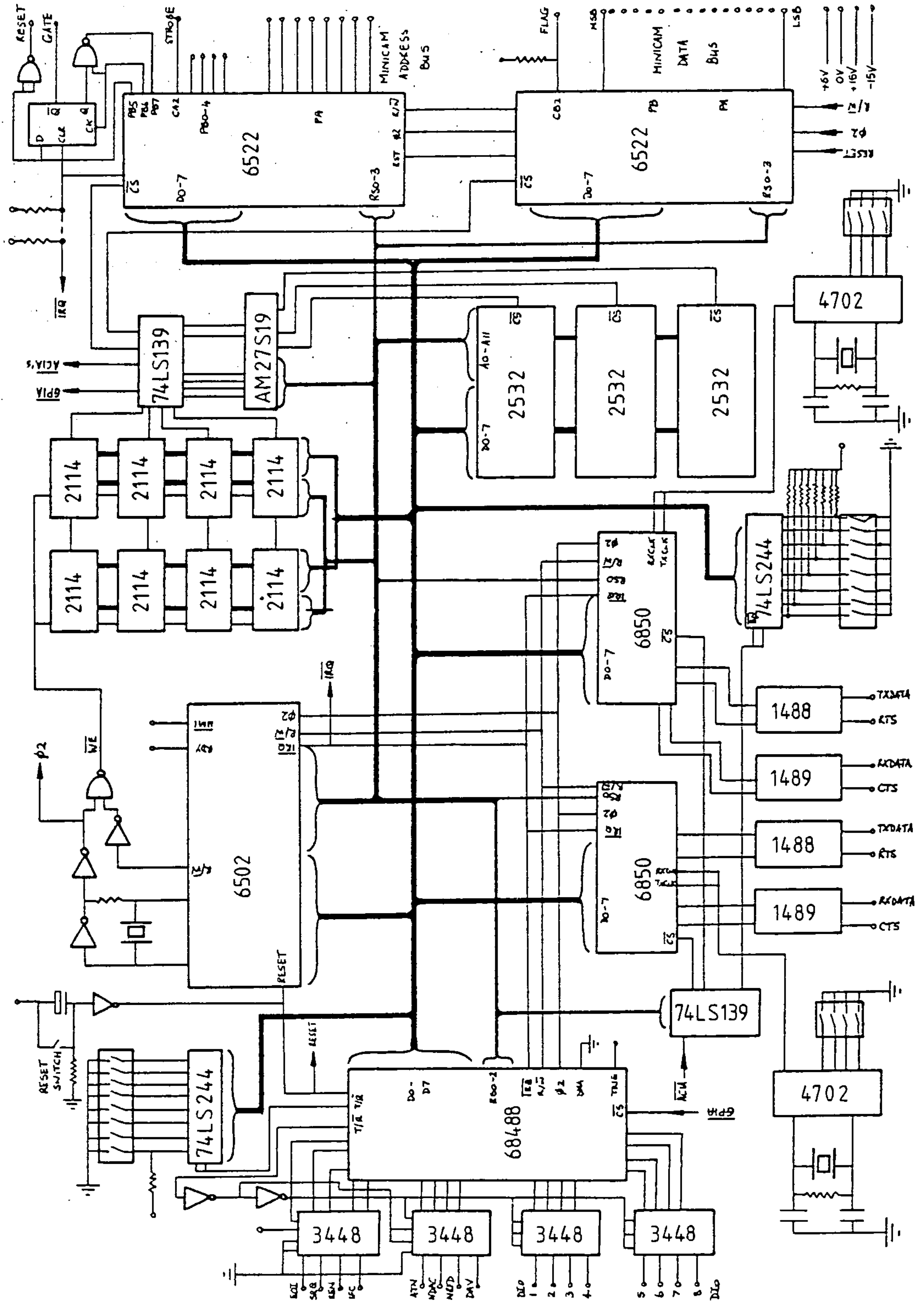


Fig. 4.4 Circuit diagram of the Minicam ICC board complete with the twin channel RS232 serial interface.

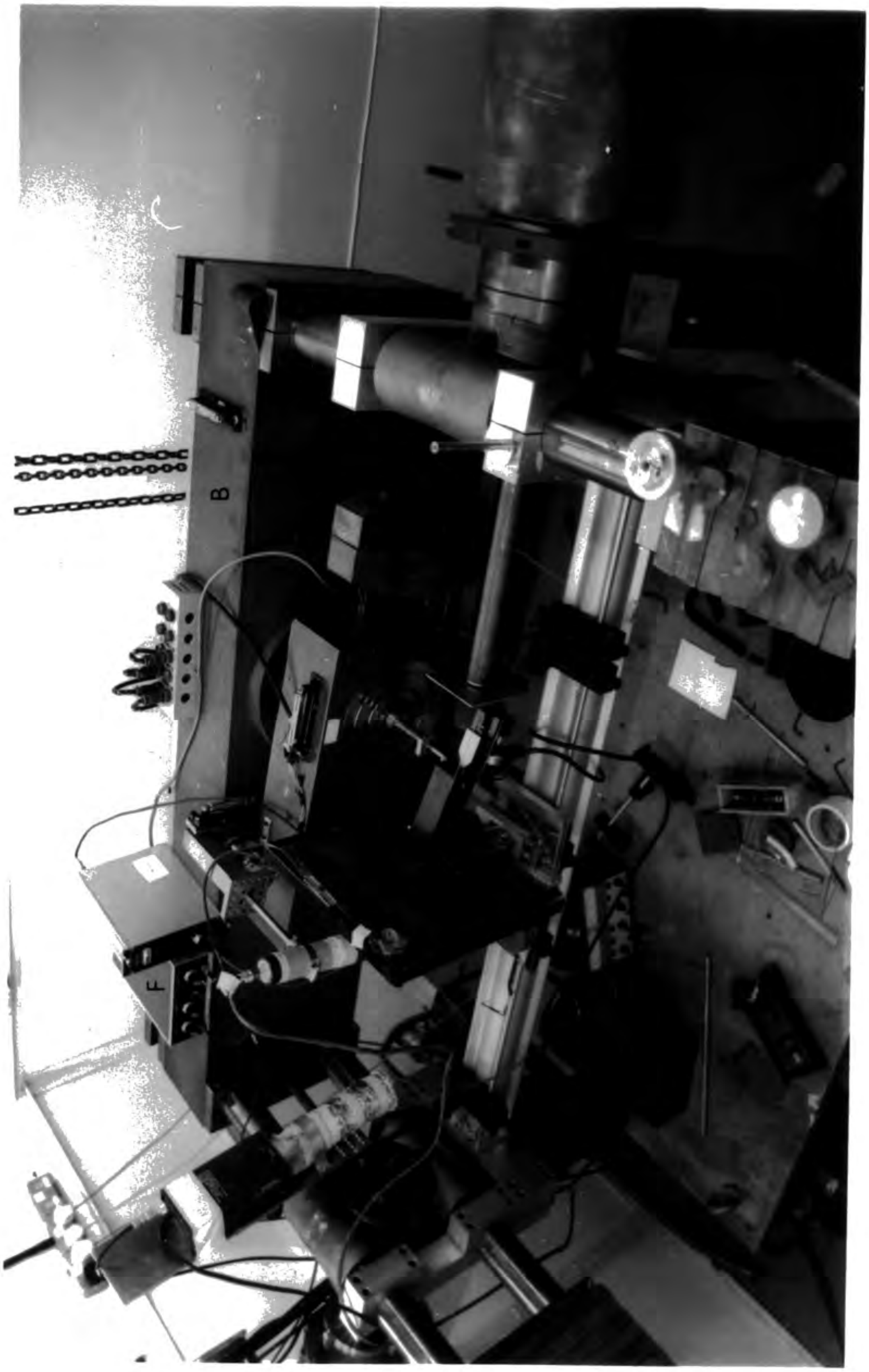


Fig. 4.5 Photograph of the 12 inch double crystal camera on the TOP2 station at the SRS, Daresbury Laboratory. The camera A, is mounted on a rotary table attached to a massive aluminium casting, B, which can be rotated about the x-ray beam. The black box monochromator, C, is shown as are the TV detector, D, and the scintillation detector, E. The H.T. supply and current amplifier, F, are used for the ionisation detector which is positioned behind the slits, G.

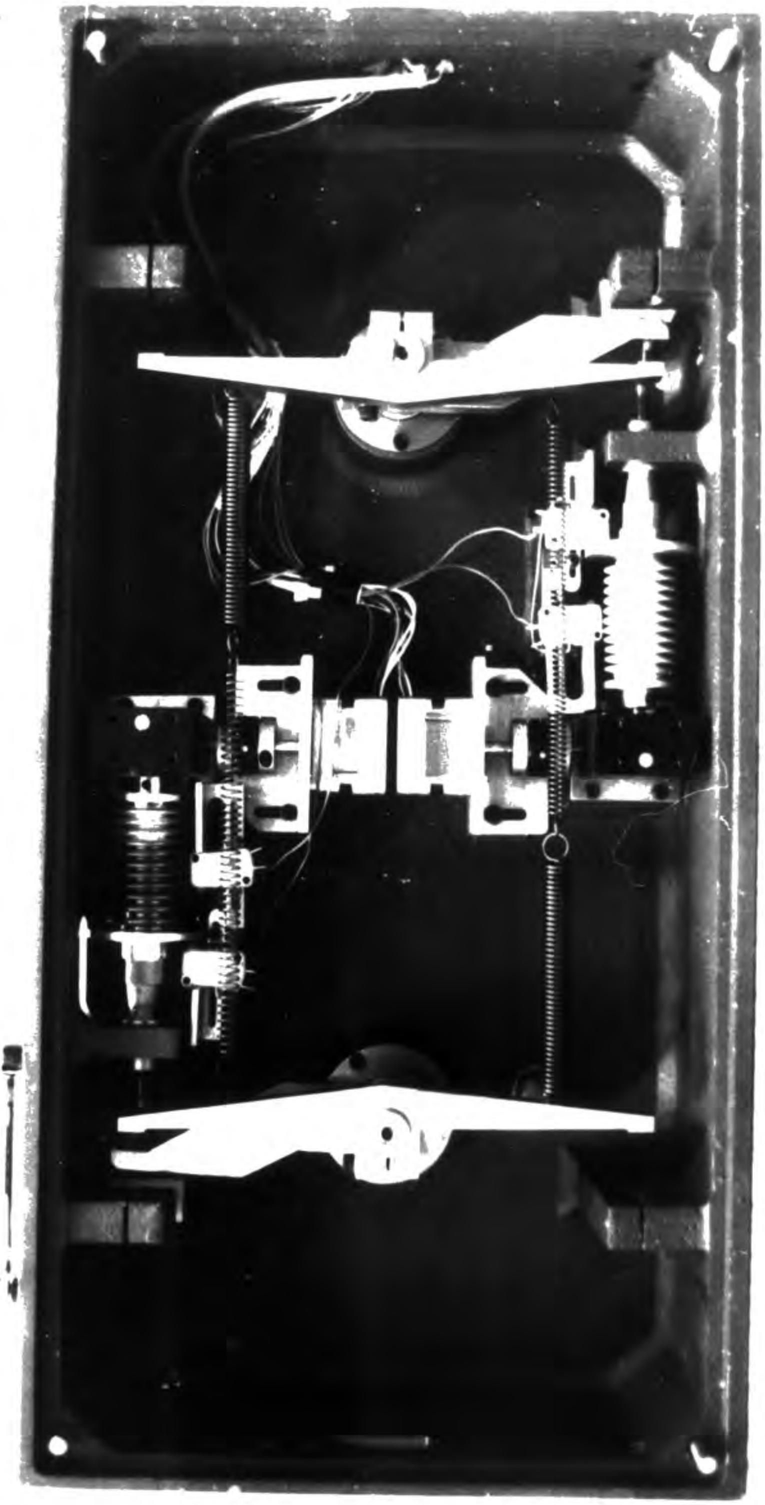


Fig. 4.6 Photograph of the 12 inch camera axial drives. The stepper motors, A, drive the tangent arm, B, via the gearbox and micrometer, C. A constant length micrometer is used.

The receive and transmit data lines plus the two hardwire handshake lines CTS and RTS are converted to RS232 levels by 1488 and 1489 line drivers and receivers. Spare drivers on the 1488 are used to provide permanently asserted RTS, DCD and DSR signals.

The final circuit design, including the original ICC circuit, is shown in fig. 4.4.

## 4.2 The Twelve Inch Camera at Daresbury

A photograph of the twelve inch camera, as installed in station 7.4 on the SRS, is shown in fig. 4.5. The micrometer/tangent arm drive system can be seen in fig. 4.6, which with 48 steps per revolution gives 0.20 arc seconds per motor step on both axes. The camera is mounted on a stepper motor driven rotary table, attached to a massive aluminium casting. This casting can be rotated around the horizontal X-ray beam to alter the polarisation state used for diffraction. With the camera vertical, ie the axes horizontal, $\sigma$ polarisation is selected. As can be seen from fig. 4.5 the optical bench mounted between the casting pivots can be used to hold slits, shielding and TV detectors in position. The scintillation detector is mounted on the detector arm which is also stepper motor driven. Spirit levels are attached to the majority of the movable arms to enable easy alignment. The beam available is approximately 2 cm high and 12 cm wide and the source is about 60 m away. The photon flux as a function of energy as produced by the

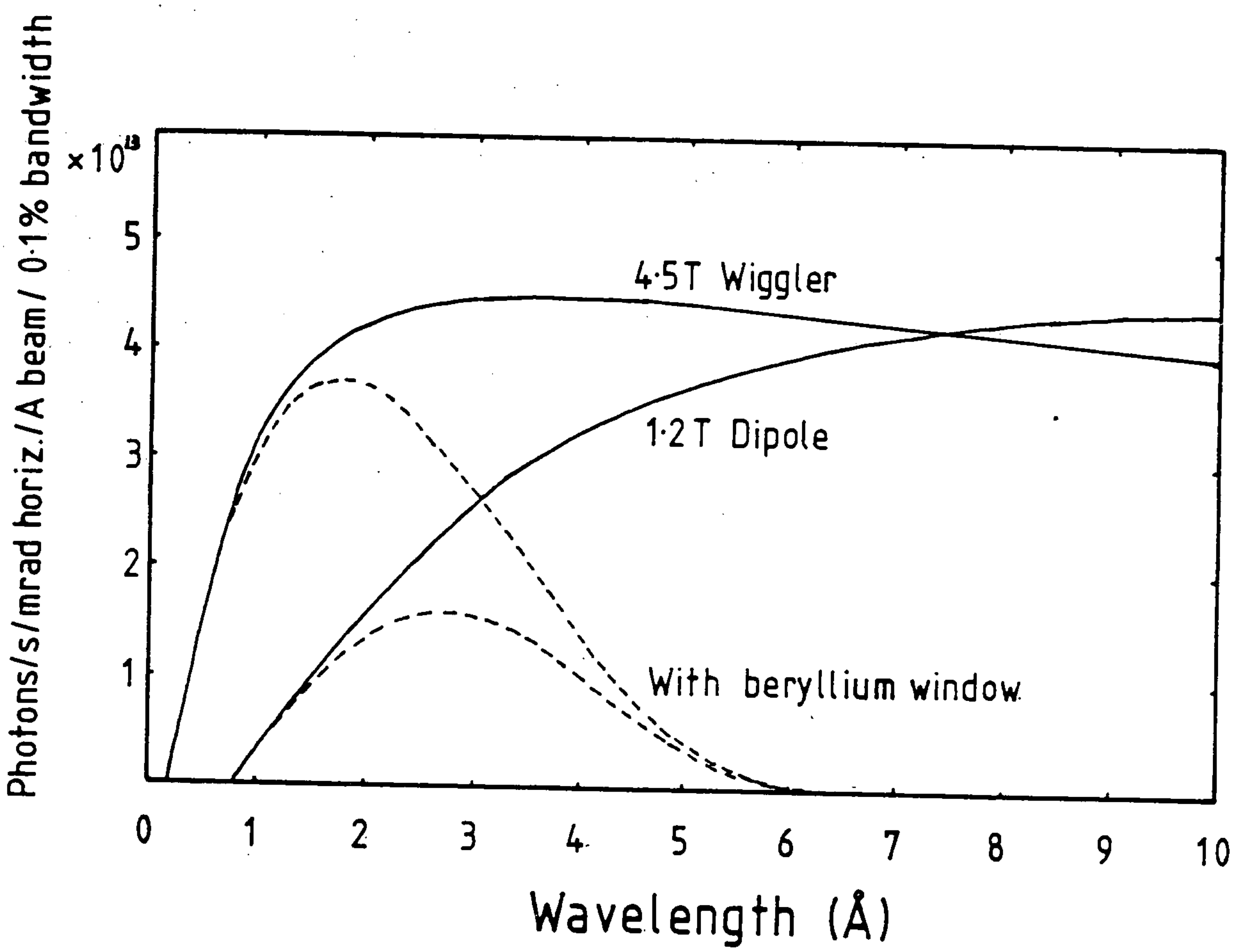


Fig. 4.7 The photon flux available at the SRS as a function of wavelength. The TOP2 station is on beam line 7 which is from a dipole magnet while the new TOP3 station is on the wiggler line. The dashed curves show the flux after passing through a eryllluim window as on the TOP2 station.

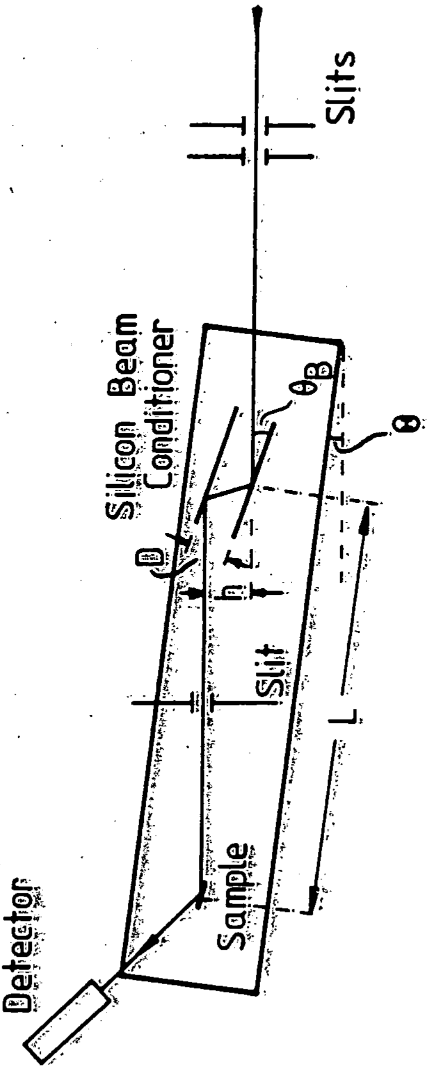


Fig. 4.8 The beam displacement which occurs when using the black box monochromator, giving an overall (+m,-m,+n) setting.

SRS is shown in fig. 4.7. This gives a minimum wavelength of about 0.6 Å while, since the station is an air station, the maximum useable at the camera is about 2.5 Å. Even if helium beam tubes were to be used the maximum wavelength is limited by the absorption of the beryllium window to approximately 5 Å.

A double reflecting 'black box' monochromator, consisting of two parallel Si (111) crystals mounted in an aluminium housing, is also available for use on this double crystal camera. The X-ray beam is doubly diffracted by two Si (111) reflections before being incident on the sample crystal. A diagram of the X-ray beam path is shown in fig. 4.8. The X-ray beam emerging from the monochromator is parallel to the incident beam, ie horizontal, but vertically displaced. Thus the camera has to be rotated in order to displace the second axis such that the beam passes through it. Since the Bragg angle of the silicon reflections is fairly small, of the order of 10 degrees for the wavelengths used, it does not change much with wavelength. The vertical displacement of the X-ray beam, h, is given by

$$h = 2\ D \cos\theta_B, \tag{4.1}$$

where D is the crystal separation and $\theta_B$ the Bragg angle. Hence, over the range of 0.5 A to 2.0 Å this separation only changes by about 1 mm. Additionally, by rotating one of the silicon crystals with respect to the other supression of harmonics is possible (Hart, 1980). A recent paper by Bonse et al (1983) also outlines the design of some similar harmonic rejecting multi-reflection monochromators. Due to the low divergence of the X-ray beam on the SRS, approximately 5 x $10^{-6}$ radians for a 0.1mm entrance slit, the beam emerging from the monochromator is very nearly a plane wave. Consequently, a overall dispersive arrangement, ie (+n, -n, +m) as with a different sample reflection, can be successfully used without introducing any noticeable broadening of the rocking curve. Therefore, variable wavelength experiments can easily be performed since only the angle of the monochromator needs to be altered, any small change in the displacement of the X-ray beam being accounted for by adjusting the height of the entrance slits infront of the monochromator.

### 4.2.1 Computer Control and Instrumentation

This camera is controlled by a PDP11/04 minicomputer and a CAMAC system, as are the majority of cameras at Daresbury. The CAMAC system is attached directly to the PDP11 UNIBUS and provides a wide variety of plug-in interface cards. A VT100 is used as the terminal. CAMAC cards control the stepper motors and floppy disc drives as well as providing the pulse counter and TV display driver. Initially, software to control the camera was written in Catex, a language similar to Basic but with additional commands to control the CAMAC system. A comprehensive manual control system was implemented, using a pocket terminal as a keypad to select motors and speeds. Rocking curve collection routines, similar to those used on the six inch camera, were also added. Recently these programmes have been rewritten in Fortran by Daresbury Laboratory staff.

Rocking curves can be plotted on a HP7470A plotter and the data saved on a high capacity Winchester disc. An 8 inch floppy drive can also be used to save the data and transfer it to other computers. A standard PET 5¼ inch floppy drive was also interfaced to the PDP11 via a CAMAC IEEE488 interface card, allowing data to be saved on these discs and read into a Cifer 2684, with a similarly interfaced PET disc drive, for replotting in Durham. The Cifer also allowed this data to be transferred onto the University's mainframe computer.

X-ray intensity was measured with a Nuclear Enterprises scintillation detector with NIM based power supply, amplifier and discriminator. The pulses were counted with a CAMAC pulse counter with a pulse generator providing the necessary timing pulses.

### 4.3 Aligning the Cameras and Recording Rocking Curves

The techniques for aligning both cameras are fairly similar, although at the SRS it is rather more straightforward due to the smooth, continuous wavelength spectrum.

### 4.3.1 The Six Inch Camera

This camera is set in the zero position by adjusting it so that the X-ray beam passes through the centre of both axes. This is easily accomplished using the optical light source, but can be checked with X-rays using either a fluorescent screen or dental film. The camera is then rigidly secured to the table. For a particular reflection and wavelength the camera is then set at $2\theta_B$ to the X-ray beam using the rotary table.

The first crystal is then mounted on the first goniometer and set near the Bragg angle. Again, the optical light source can be used if a surface symmetric reflection is being used with a highly polished crystal. With the detector mounted to look down the axis the angle of the first crystal is adjusted to obtain a maximum count rate. Even if the adjustment of the camera to $2\theta_B$ is slightly out the wide aperture of the detector should ensure that the reflection is found. This process of finding the reflection can be greatly speeded up by rotating the goniometer with a length of wire. Dental film is then used to deterine the position of the diffracted beam with respect to the second axis. If the beam does not pass through the axis the $2\theta_B$ angle of the camera should be altered to bring it into line and the angle of the first crystal readjusted.

Recently Fewster (1985) has explained a similar method using Kevlar twine to pull the first goniometer. If the twine is then securely fixed the camera can be rotated without altering the angle of the goniometer with respect to the incident beam. With a narrow slit positioned infront of the detector the camera angle can then easily be adjusted to the optimum position.

The second crystal is then mounted, ensuring that the X-ray beam will intercept the surface. With the crystal set near the appropriate Bragg angle the reflection can be found. A computer searching technique can be used or, for broad reflections, a wire pull technique similar to that used for the first crystal. For the majority of III-V crystal reflections with CuK$\alpha_1$ radiation the latter technique can usually be used.

The tilt adjustment of the second crystal is then performed to provide the narrowest possible reflection. A suitable technique based on the change in angular position of the peak as a function of the tilt angle has been described by Fewster (1985).

Once all adjustments have been made the rocking curve can be recorded. As small a beam as possible should be used to reduce the effect of sample curvature and area variations in composition or thickness.

### 4.3.2 The Twelve Inch Camera

Alignment with a two crystal (+-) arrangement is similar to that described for the six inch camera. Instead of altering the angle of the camera to ensure that the diffracted beam from the first crystal passes through the second axis the angle of the first crystal alone can be altered. This will change the wavelength diffracted slightly. The angle of the crystals can be set relative to the horizontal X-ray beam with a variable angle set square mounted on the horizontal optical bench. The reflection from the second crystal is easily found with a simple TV imaging detector, consisting of a fluorescent screen mounted on the front of an image intensifier tube (an obsolete Mullard model) with the back screen viewed by a closed circuit TV camera. The peak is best found with a beam roughly the same size as the sample.

When aligning the camera with the black box monochromator, a slightly different technique is required. The angle, $\theta$, at which the camera should be set is given by

$$\sin\theta = 2\ D \cos\theta_B\ /\ L, \qquad (4.2)$$

where L is the axis separation and D the spacing of the crystals in the monochromator. The easiest alignment process is to initially set the camera horizontal using a precision spirit level. The camera is then rotated downwards from the horizontal by the angle $\theta - \theta_B$ and the monochromator set horizontal, again using a spirit level. Rotating the camera then back to the required angle above the horizontal puts the monochromator at the required angle. Since the monochromator is not channel cut but consists of two separate crystals, they need to be aligned to allow the X-ray beam through. One of the crystals can be rotated by a motorised screw or an electromagnet / magnet pair. The correct alignment can then be found by viewing the diffracted X-ray beam with either the TV detector or an ionisation detector. A pointer in the second axis can be used to ensure that the beam passes through this axis. The second crystal can then be mounted and the reflection found using a similar technique to that previously described with a normal (+-) arrangement.

We note that with the SRS the beam divergence is so small that the double crystal arrangement does not need to be adjusted for tilt, as with a conventional source. This greatly increases the speed at which rocking curves can be recorded and helps overcome the delays introduced in finding the reflections by not being able to rotate the crystals with the wire pull technique!

# CHAPTER 5

## SINGLE UNIFORM LAYERS

Rocking curves from single layers would be expected to be the easiest to interpret. It is straightforward to measure the perpendicular and parallel mismatches by measuring the layer and substrate peak separation using symmetric and asymmetric reflections. For a coherent layer/substrate interface, ie with no mismatch dislocations, and a surface symmetric reflection if the peak separation is $\Delta\theta$ then, from the Bragg equation, we obtain

$$\frac{\Delta d}{d} = \Delta\theta \cot\theta_B \qquad (5.1)$$

where d is the lattice spacing of the substrate and $\Delta\theta = (\theta_S - \theta_L)$. The relaxed lattice mismatch is then found using equation (1.34) which gives the composition according to Vegard's Law (see section 1.4). Any angle, $\Delta\varphi$, between the 001 planes of the layer and substrate can be measured by reversing the X-ray path and recording two rocking curves. From the two peak separations, $\Delta\theta_1$ and $\Delta\theta_2$ we find that

$$\frac{\Delta d}{d} = \tfrac{1}{2}(\Delta\theta_1 + \Delta\theta_2) \cot\theta_B \qquad (5.2)$$

$$\Delta\varphi = \tfrac{1}{2}(\Delta\theta_1 - \Delta\theta_2) \qquad (5.3)$$

If asymmetric reflections are used an additional peak separation will be introduced due to the difference in the angle between the reflection plane and surface for the layer and substrate. If $\varphi_L$ and $\varphi_S$ are these angles for the layer

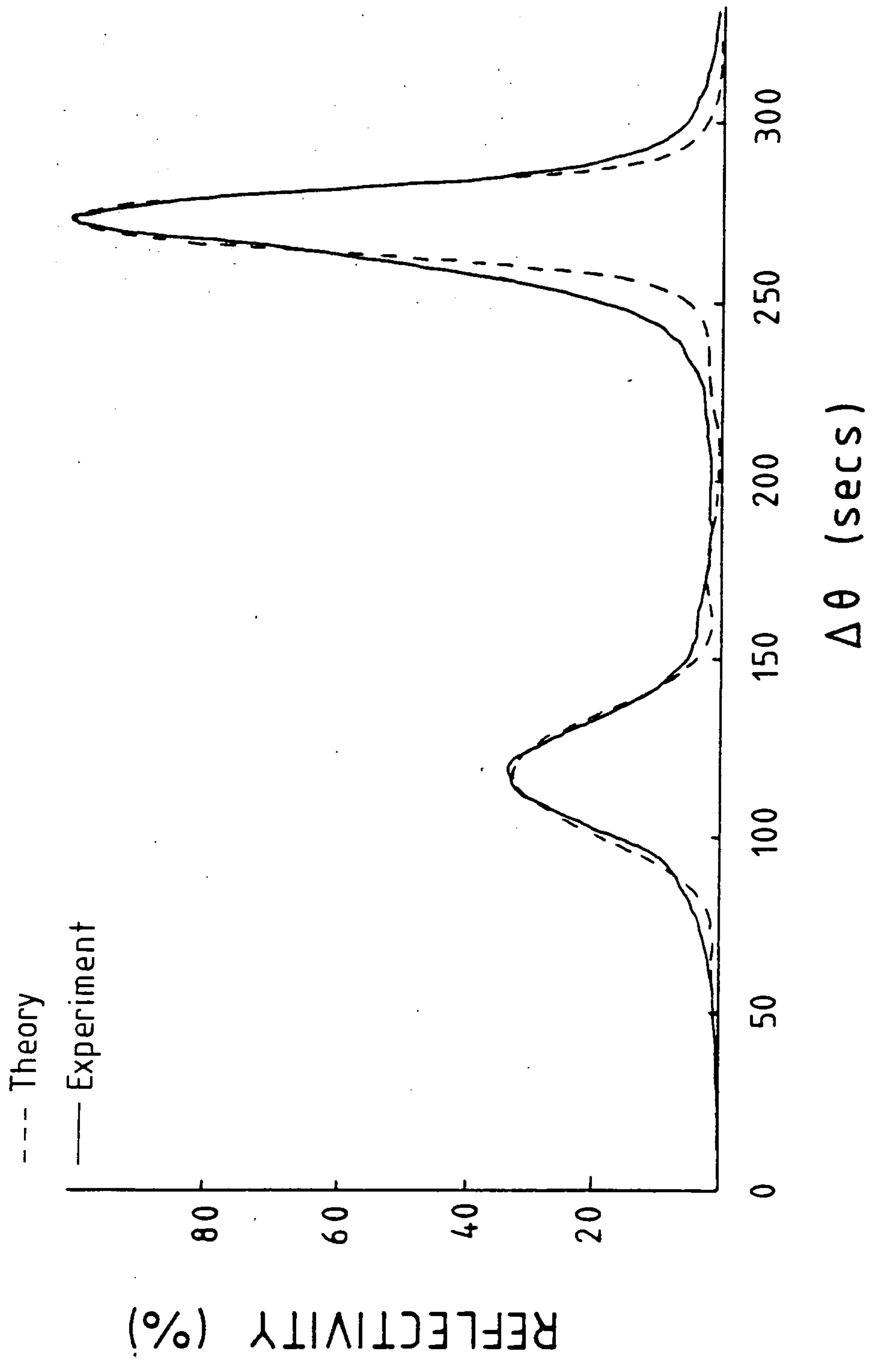


Fig. 5.1 Experimental and computed 004 rocking curves from a 650 ppm mismatched, λ = 1·3μm GaInAsP sample. The solid curve is the experimental curve and the dashed curve the theoretical best fit curve. CuK$\alpha_1$ radiation was used and the layer thickness is 0.45 μm.

and substrate respectively then when the angle of incidence is $(\theta-\varphi)$ the peak separation is $(\theta_S - \theta_L - \varphi_S + \varphi_L)$ and when the angle of incidence is $(\theta+\varphi)$ it is $(\theta_S - \theta_L + \varphi_S - \varphi_L)$.

However, the layer thickness cannot be determined from these measurements. It can be determined by comparing the experimental rocking curve with computed curves for various layer thicknesses, assuming the lattice mismatch already calculated.

## 5.1 Surface symmetric reflections

Fig. 5.1 shows the experimental 004 rocking curve with $CuK\alpha_1$ radiation for a sample with a single quaternary layer, with a band gap corresponding to radiation of wavelength 1.3 µm. The substrate is, as usual, 001 InP. The mismatch is determined to be 650 ppm from the separation of the peaks, giving the layer composition as $Ga_{0.27}In_{0.73}As_{0.61}P_{0.39}$. We note that the substrate peak is considerably broader than the theoretical width, indicating that the sample is curved. The amount of broadening provides a measure of the curvature. The calculated curve that best fits this experimental curve is also shown in fig. 5.1, with a layer thickness of 0.45 µm. Agreement between the curves is fairly good, although the tails of the substrate peak are slightly broader than in the calculated curve. The weak Bragg case Pendellosung fringes seen in the computed curve are not visible in the experimental curve. Clearly, the presence of threading dislocations, in the layer, introducing a local lattice curvature, will blurr out this

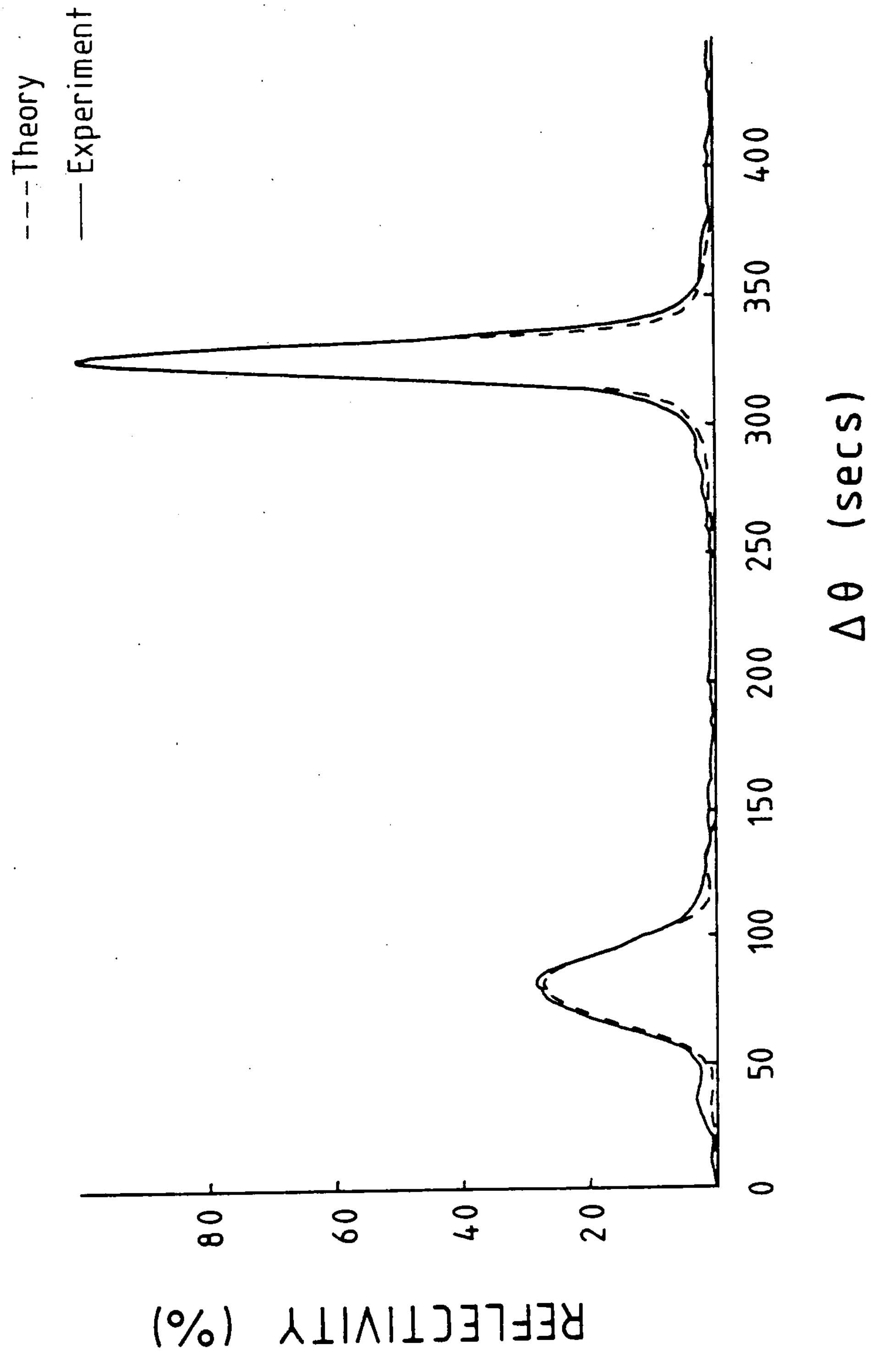


Fig. 5.2 Experimental and theoretical 004 rocking curves from a 960 ppm mismatched GalnAs sample. CuK$\alpha_1$ radiation was used and the solid curve is the experimental rocking curve. The layer thickness is 0.55 μm.

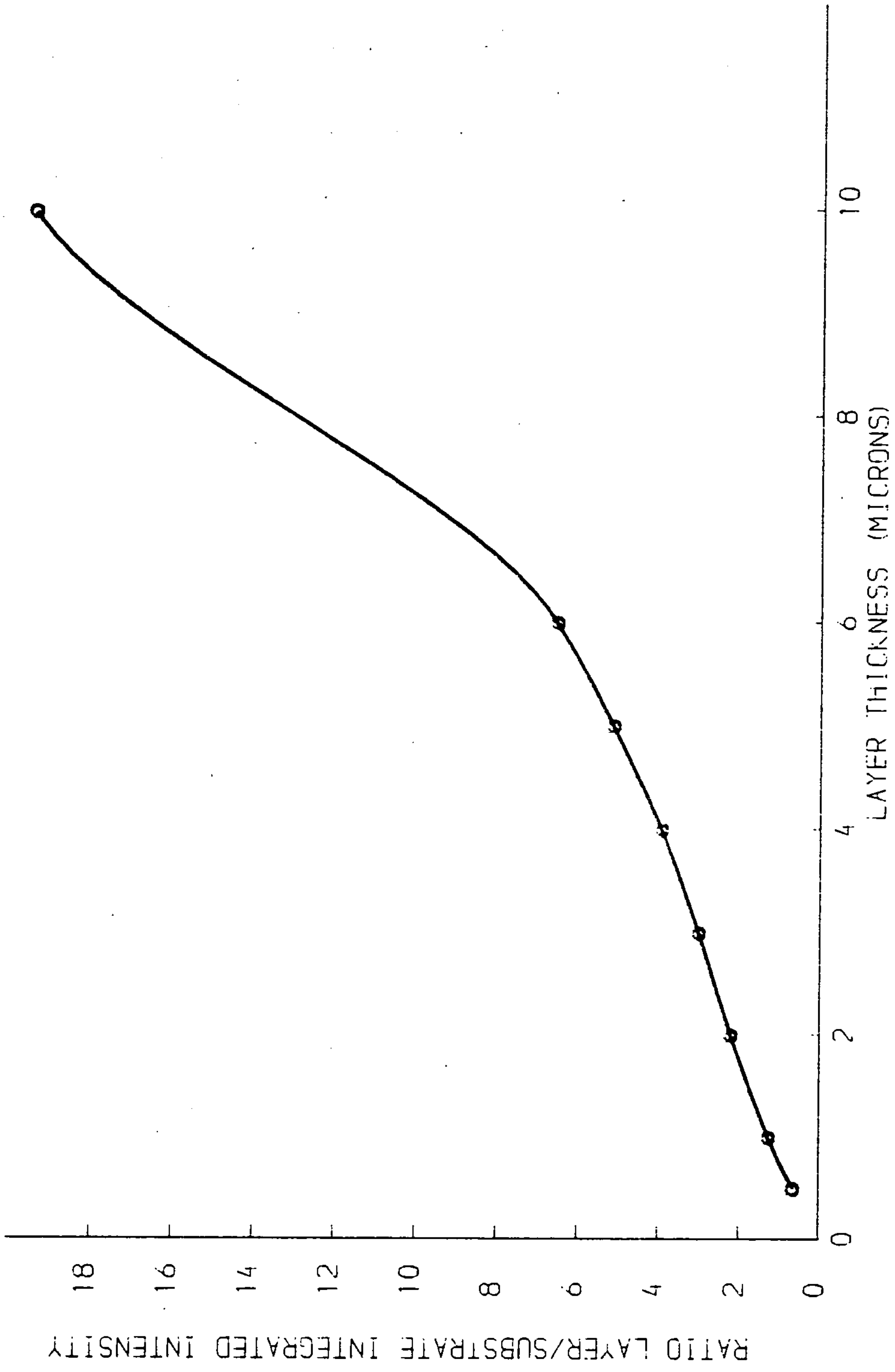


Fig. 5.3 The ratio of integrated intensities from the layer and substrate as a function of layer thickness 004 refl. This curve is for CuK$\alpha_1$ radiation and a GaInAs layer of 600 ppm mismatch.

weak structure. Measurement of the rocking curve with the X-ray path reversed showed that there was no tilt angle between the 001 planes of the layer and substrate.

A similar example is shown in fig. 5.2, for another ternary layer. The relaxed lattice mismatch is determined to be 960 ppm, giving $Ga_{0.45}In_{0.55}As$, and the computed curve for a 0.55 µm layer thickness is also shown in fig. 5.2. The half width of the substrate peak is nearer the theoretical value than in the previous example. Agreement between the computed and experimental curves is extremely good, being slightly better than the fit in the previous example.

Instead of calculating a number of rocking curves for each sample in order to determine the layer thickness the ratio of integrated intensities of the layer and substrate peaks can be used, provided that the peaks do not overlap. The ratio of peak heights could also be used but these are affected by the sample curvature, as shown in chapter 1. A graph of the ratio of integrated intensities as a function of the layer thickness is shown in fig. 5.3, for the 004 reflection with $CuK\alpha_1$ radiation and these values are also given in Table 5.1. These values were calculated for a GaInAs ternary layer of 600 ppm mismatch but, due to the very small effect that changing the composition has on the structure factors this curve can be used for any mismatch. Further, since the structure factors for quaternary GaInAsP alloys are only slightly different from those of the ternary

GaInAs this curve can also be used for these types of layers without introducing any large errors.

TABLE 5.1

**Values of the ratio of integrated intensities of the layer and substrate peaks for a 600 ppm mismatched GaInAs layer as a function of layer thickness.**

| Layer Thickness (μm) | Ratio of layer/substrate integrated intensity (±0.05) |
|---|---|
| 0.5 | 0.6 |
| 1.0 | 1.3 |
| 2.0 | 2.2 |
| 3.0 | 3.0 |
| 4.0 | 3.9 |
| 5.0 | 5.1 |
| 6.0 | 6.5 |
| 7.0 | 9.4 |
| 8.0 | 13.1 |
| 9.0 | 17.1 |
| 10.0 | 19.4 |

If the layer is only slightly mismatched from the substrate the peaks will begin to overlap. Their separation can be increased by either increasing the wavelength or narrowing the peaks. Increasing the wavelength will alter the peak separation according to the Bragg equation but will also increase the half width of the peaks. Additionally, due to the increased absorption within the layer the thickness of crystal sampled will be reduced. Alternatively, reducing the wavelength will produce the opposite effect but, due to the reduced Bragg angle, the effect of sample curvature will be increased since the X-ray beam will intercept a larger area of the sample. Fig. 5.4

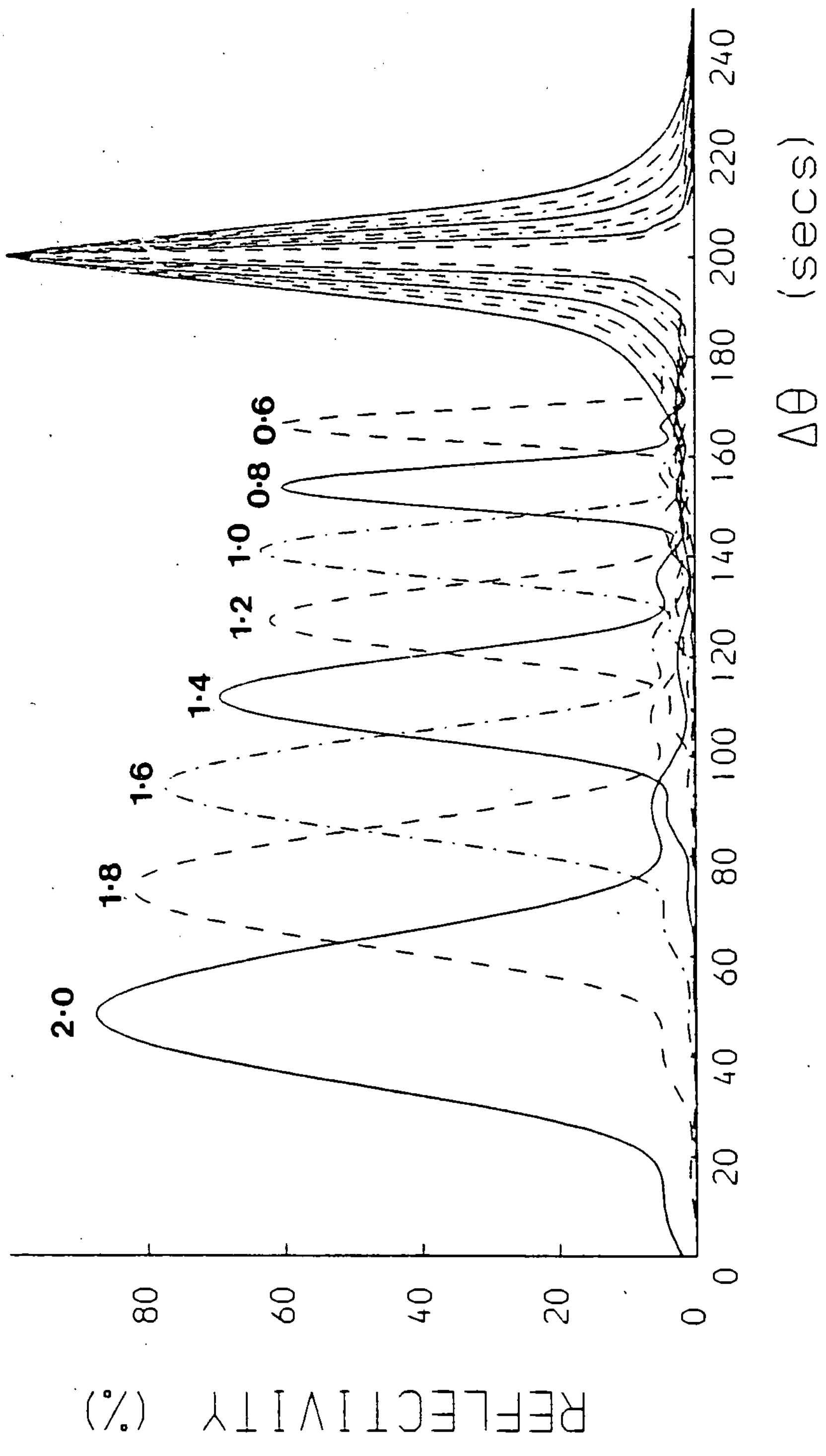


Fig. 5.4 004 Theoretical Rocking curves from a 400 ppm, mismatched, 1μm thick GaInAs layer as a function of wavelength.

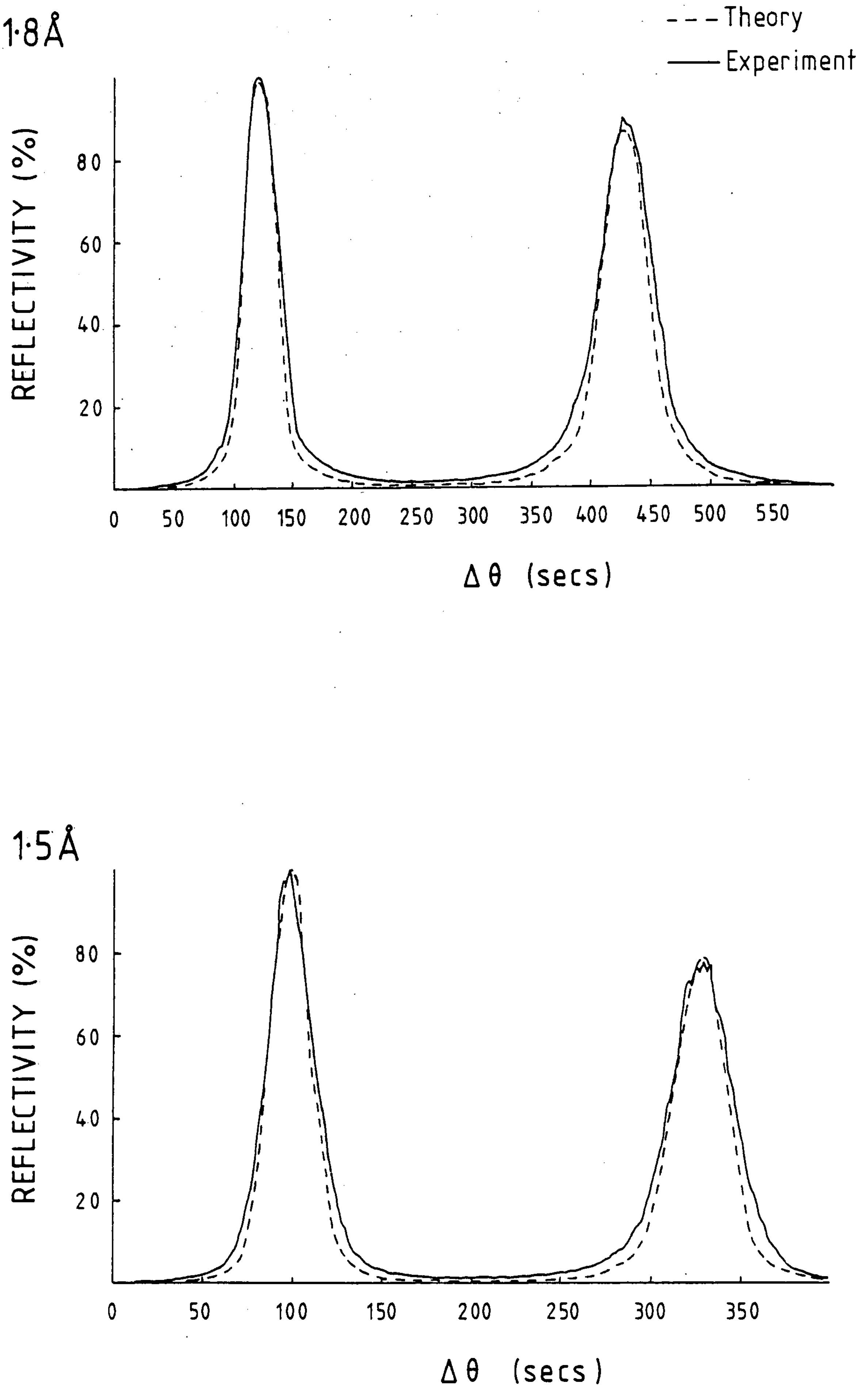
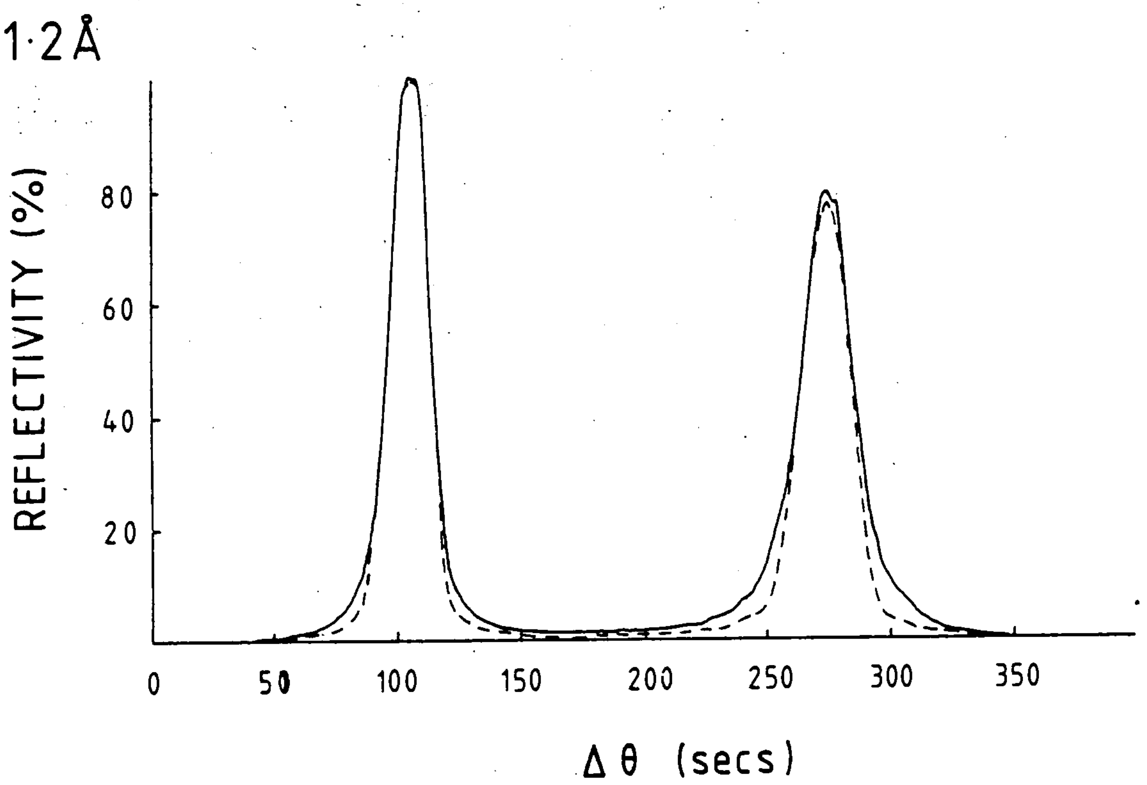
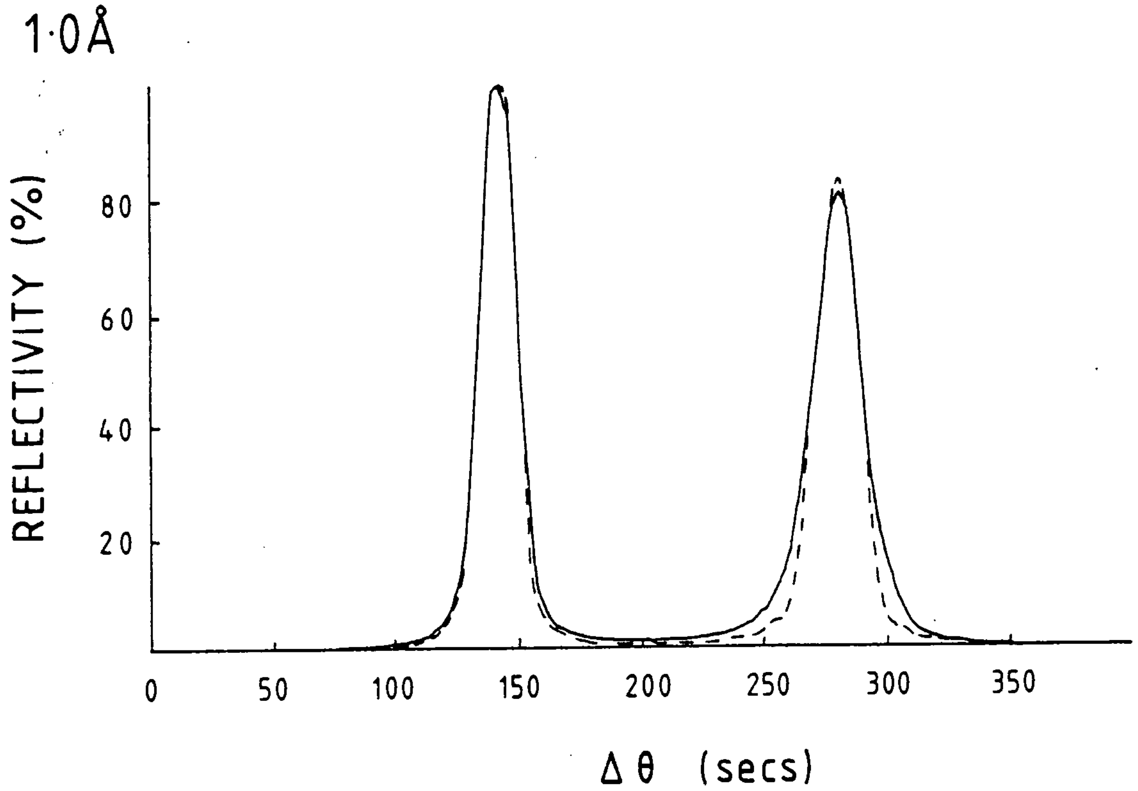


Fig. 5.5 Experimental and calculated 004 rocking curves from a 950 ppm mismatched 0.90 μm thick layer at four wavelengths. The layer material is GaInAsP with λ = 1.55 μm.



1·2Å
REFLECTIVITY (%)
80
60
40
20
0
50
100
150
200
250
300
350
Δ θ (secs)



1·0Å
REFLECTIVITY (%)
80
60
40
20
0
50
100
150
200
250
300
350
Δ θ (secs)

shows the effect of altering the wavelength on the 004 rocking curve calculated for a 1 μm, 400 ppm mismatch ternary layer.

Since the X-ray parameters used in the calculation change as a function of the wavelength we need to check that the curve calculated as the best fit to the experimental curve at one wavelength still agrees with the experimental curve when the wavelength is altered. This will check that all the variable wavelength parameters, such as the dispersion corrections to the structure factors, are calculated correctly. Fig. 5.5 shows a series of experimental rocking curves for a λ=1.55 μm GaInAsP layer at 1.0, 1.2, 1.5 and 1.8 Å. These rocking curves were recorded using the double crystal camera at Daresbury in conjunction with the double reflecting monochromator. The wavelength can then be easily altered without needing to adjust the angle of the camera. From the rocking curve at 1.5 A the mismatch was calculated to be -950 ppm, corresponding to $Ga_{0.43}In_{0.57}As_{0.89}P_{0.11}$, and the best fit computed curve, also shown in fig. 5.5, was found for a 0.8 μm layer thickness. Theoretical curves at the other wavelengths are also shown in fig. 5.5 for comparison with the experimental curves. There is good agreement between these curves at all the wavelengths used, both in the angular separation, peak widths and heights. This confirms that using just one rocking curve at one wavelength gives a layer thickness that is consistent with curves recorded at other wavelengths.

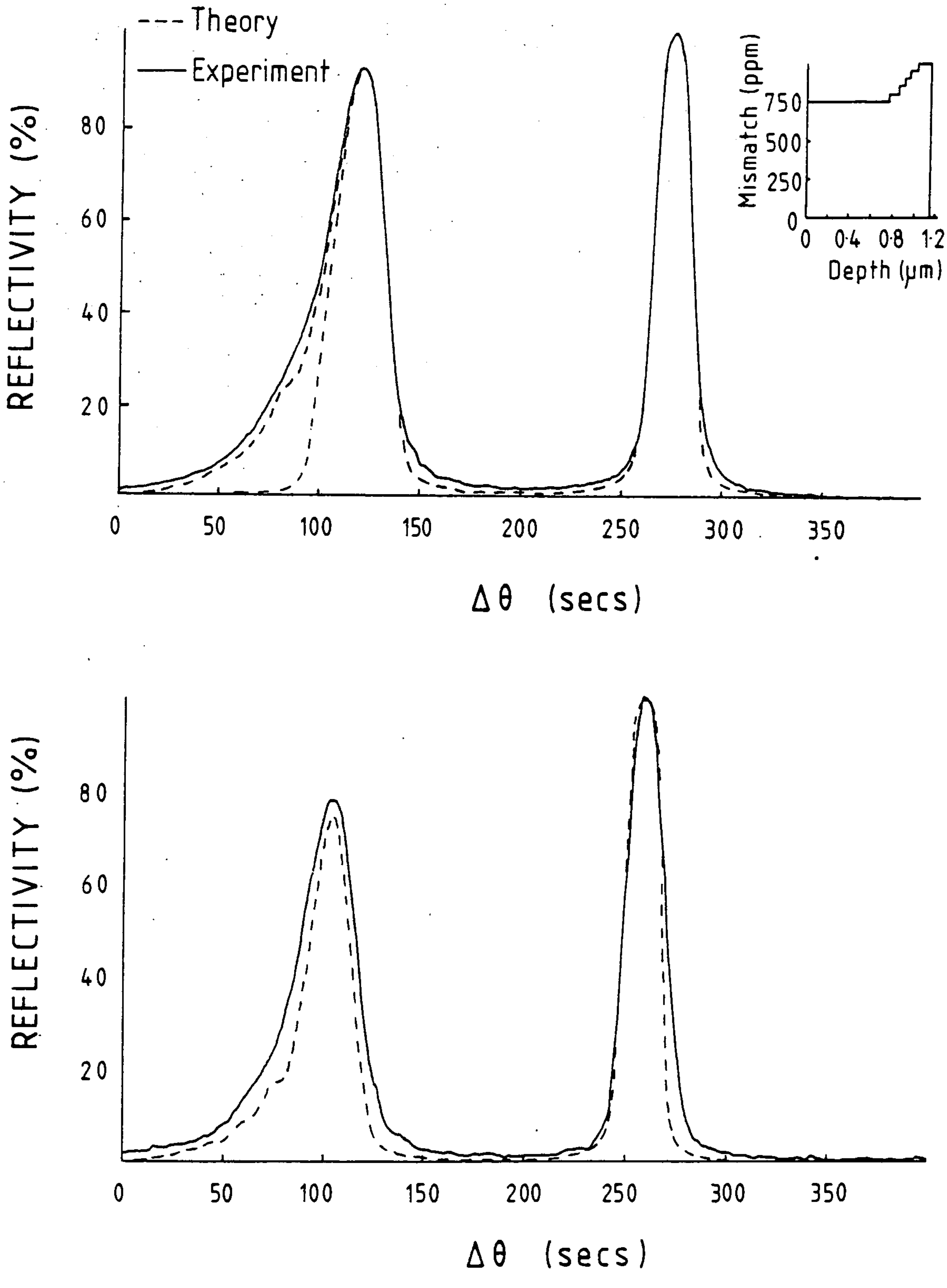


Fig. 5.6 (a) Theoretical and experimental 004 rocking curves from a 750 ppm mismatched, 1.15 μm thick GaInAs layer. Two computed curves are shown, one for a 0.9 μm uniform layer and one from a 1.15 μm thick layer with a region near the interface of larger mismatch, as shown on the inset. The wavelength used was 1.3 Å with σ polarisation.

(b) The computed and experimental rocking curves from the same sample at the same wavelength but with the double crystal camera rotated to give π polarisation.

Since only surface symmetric reflections have been used no measure of the interface coherency can be obtained. However, for the majority of samples studied mismatch dislocations could not be observed using Lang topography, hence the parallel mismatch must be zero and the interface coherent. Dislocations are often observed but these are usually threading dislocations existing in both the substrate and layers and are not formed due to layer mismatch. Consequently, the use of a single 004 reflection is a perfectly valid means of measuring the layer composition.

Since with a conventional laboratory generator randomly polarised radiation is produced in the characteristic lines any calculated rocking curves should include the contribution to the rocking curve from the $\pi$ component as well as from the $\sigma$ component. This effect is included in the calculation by adding the rocking curves for each polarisation state, according to equation (2.17). The contribution due to the $\pi$ component can be examined experimentally by using the double crystal camera at the SRS with the diffraction plane rotated through 90° to the horizontal. This setting corresponds to $\pi$ polarisation for an X-ray beam lying exactly in the plane of the electron orbit.

Fig. 5.6(a) shows the 004 rocking curve at 1.3 Å for a sample with a single GaInAs layer with $\sigma$ polarisation. The separation of the substrate and layer peaks gives the mismatch as 750 ppm. The computed curve for a layer thickness of 0.9 μm is also shown in fig. 5.6(a), but does not agree particularly well with the layer peak which is considerably asymmetric with an intense tail on the low angle side. This would suggest that a region of slightly larger lattice parameter was present within the layer and fig. 5.6(a) also shows the computed curve for a layer where a narrow region of larger lattice parameter is included next to the layer/substrate interface, with a total layer thickness of 1.15 μm. The relaxed lattice mismatch as a function of depth is shown in the inset in fig. 5.6(a), corresponding to a range of Indium content from 51.6% at the interface to 52.0% at the surface. It is not unexpected that such a region could exist near to the interface as many similar observations have been reported (see Chapter 2). The agreement between the computed curve and the experimental curve is now considerably improved, with the asymmetric peak being extremely well simulated by the calculated curve.

The double crystal camera was then rotated towards the horizontal, π polarisation position, in 15° steps with the rocking curve being recorded at each stage. Initially, with the black box monochromator the only effect observable was a considerable broadening of the peaks due to the increased dispersion present from the wide X-ray source. The experiment was then repeated with a good quality InP crystal as the first crystal giving an exactly parallel (+-)

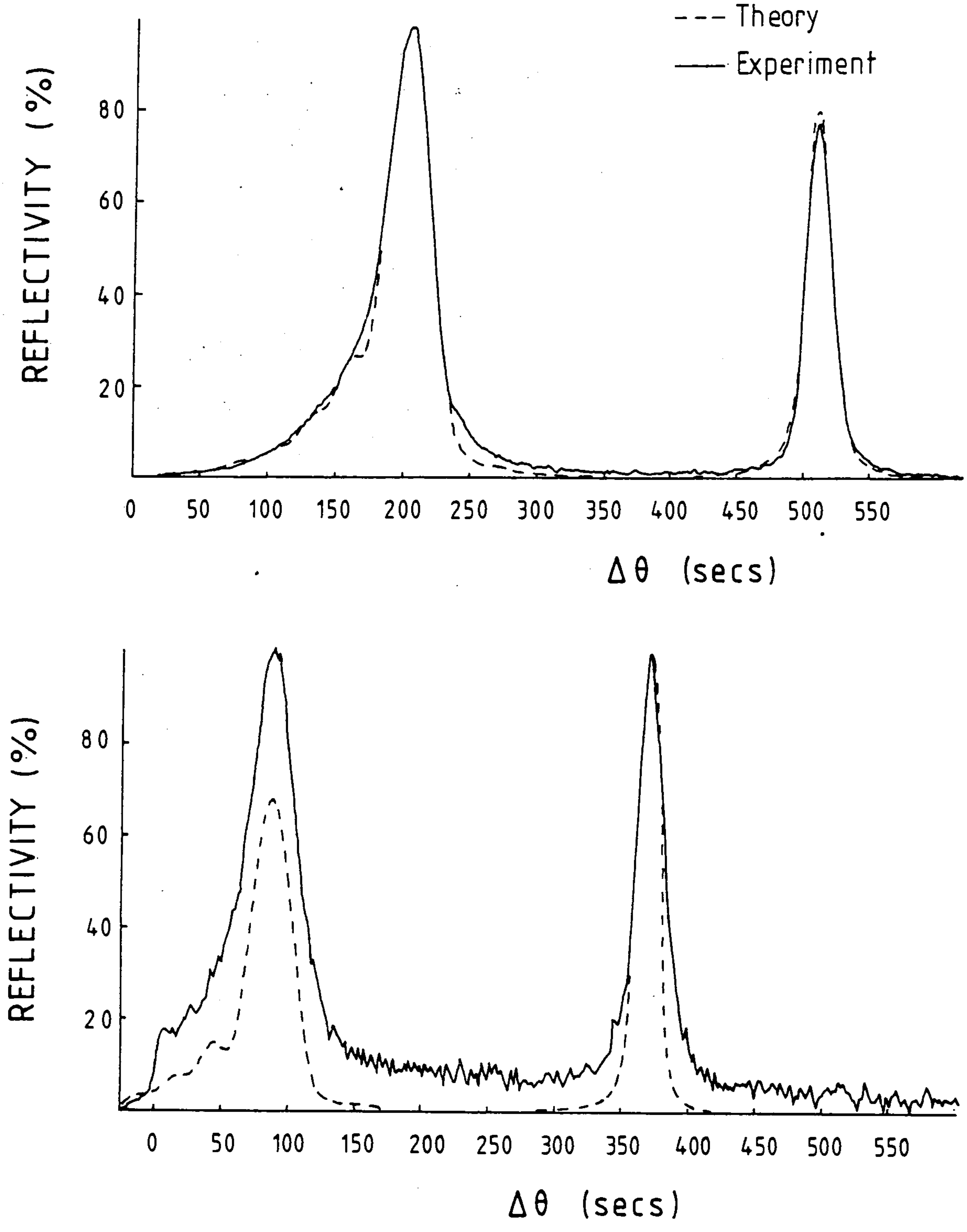


Fig. 5.7 (a) Rocking curves from the same sample used in fig. 5.6 but at 2.0 Å and σ polarisation. The composition used to give the calculated curve is that shown in the inset in fig. 5.6(a).

(b) Computed and experimental curves at the same wavelength but with π polarisation.

arrangement. Although the peaks no longer increased in width little other effects could be observed until complete π polarisation was selected. This rocking curve is shown in fig. 5.6(b). We would expect this result as the diffracted intensity with π polarisation will be smaller than that with σ polarisation thus the σ component will dominate the rocking curve until complete π polarisation is selected. Since the extinction length should be longer with π polarisation the diffracted intensity from the layer would be expected to be smaller than with σ polarisation. Additionally, the half width of the peaks should also be reduced. However, when the effect of sample curvature is included in the calculation of the rocking curve both of these effects are reduced. The experimental rocking curve does show a small reduction in the height of the layer peak, but no change in the peak half widths can be observed. The computed rocking curve, also shown in fig. 5.6(b) does agree fairly well with this experimental curve, although the calculated layer peak is slightly narrower and less intense. Clearly, any part of the X-ray beam lying slightly out of the plane of the electron orbit will not be completely polarised and can provide some σ polarisation contribution to the rocking curve.

This experiment was then repeated at 2.0 Å, giving a Bragg angle of 43°, where the polarisation factor C is much more significant than at 1.3 Å. The σ polarisation rocking curve is shown in fig. 5.7(a) along with the computed curve assuming the composition and thickness already determined. Agreement between the curves is again particularly good. We note that the substrate peak is narrower than at 1.3 Å due to the decreased effect of sample curvature at this higher Bragg angle. Fig. 5.7(b) shows the π polarisation rocking curve plus the computed curve. There is rather more discrepancy between these two rocking curves, with the computed curve showing a much less intense layer peak, although the shapes agree fairly well. Due to the very low diffracted intensity the background of the experimental curve is very noisy. Clearly, when the polarisation term is so significant and the diffracted intensity so low any very weak σ polarisation contribution will have a marked effect on the rocking curve. It should be noted that similar experiments, as reported by Bonse, Krasnicki and Teworte (1983), are not straightforward. In their case an additional monochromator crystal was used, with a near 45° Bragg angle, to reduce the amount of the σ component present in the beam.

## 5.2 The Use of Highly Asymmetric Reflections to Study Thin Layers

For very thin layers, less than 0.5 μm, the diffracted intensity from the layer, using the 004 reflection, becomes very small compared to that from the substrate. Additionally, the half width of the layer peak is greatly increased when low angles of incidence are used. Typically, the extinction distance for the 004 reflection is of the

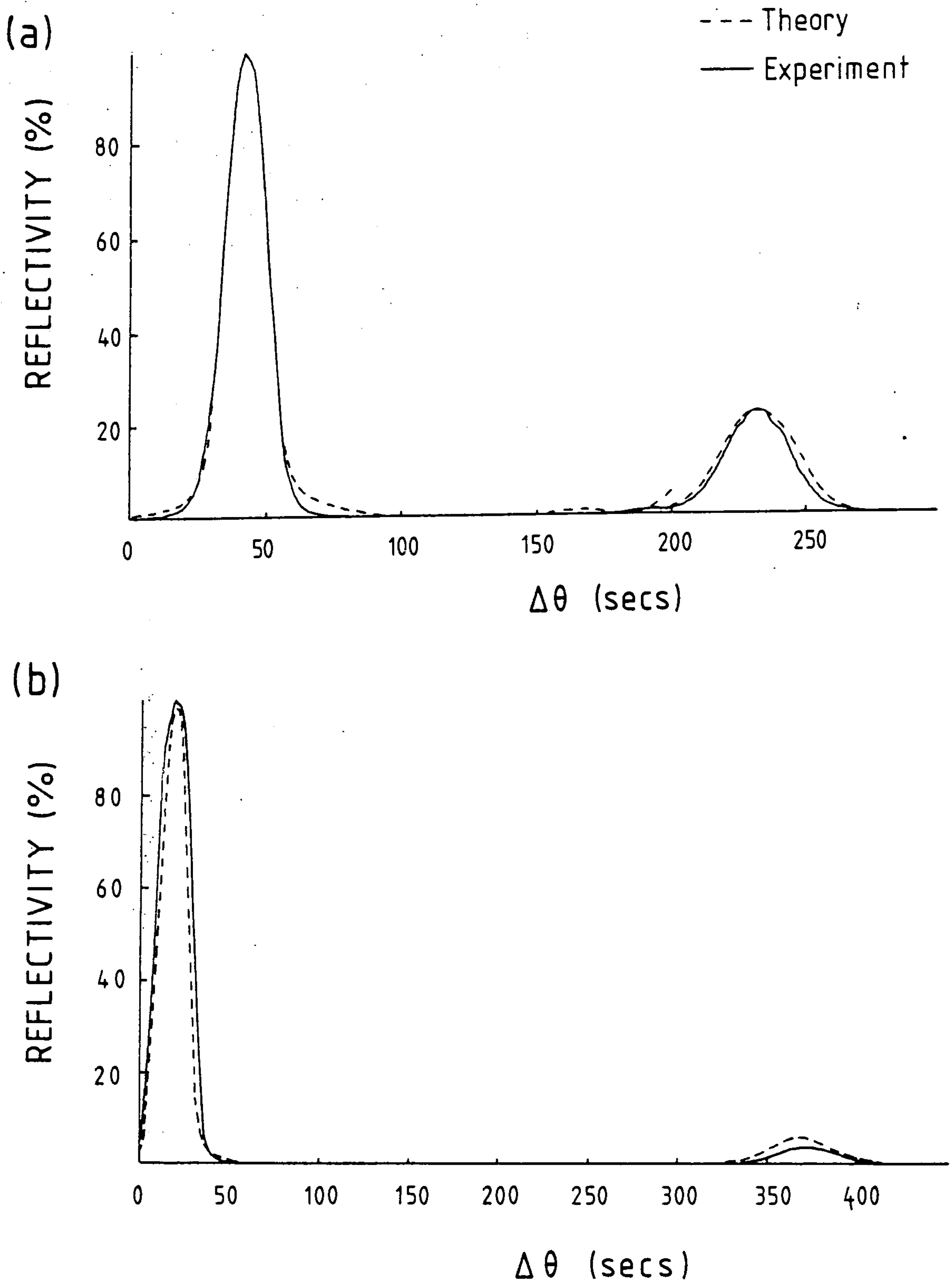


Fig. 5.8 (a) Experimental and computed 004 rocking curves from a 0.3 μm, λ = 1.31 μm, -780 ppm mismatched GaInAsP/lnP sample. The wavelength is 1.5 Å and σ polarisation from the SRS.

(b) The 115 rocking curves from the same sample and with the same experimental conditions.

order of 10 μm, which is large compared to such layer thicknesses. However, by using a highly asymmetric reflection the extinction distance can be greatly reduced due to the factor $(\gamma_0|\gamma_h|)^{\frac{1}{2}}$. Specifically, by reducing the incident or exit angle to below 5° the factor $\sin(\theta-\varphi)$ will ensure that the extinction distance approaches zero. The diffracted intensity from the layer should then be dramatically increased.

A series of rocking curves were recorded for a $\sim$ 0.4 μm, $\lambda$=1.31 μm GaInAsP layer, grown on an InP substrate, as a function of wavelength for several reflections using the Daresbury double crystal camera. Fig. 5.8(a) shows the 004 rocking curve at 1.5 Å, with the double reflecting monochromator as the 'first' crystal. From the peak separation the mismatch is determined as -780 ppm, giving the layer composition as $Ga_{0.30}In_{0.70}As_{0.62}P_{0.38}$ and the computed rocking curve for a 0.3 μm thick layer is shown also shown in fig. 5.8(a) for comparison. The two curves again agree extremely well although, as expected, the diffracted intensity from the layer is only approximately 20% of the intensity of the substrate peak.

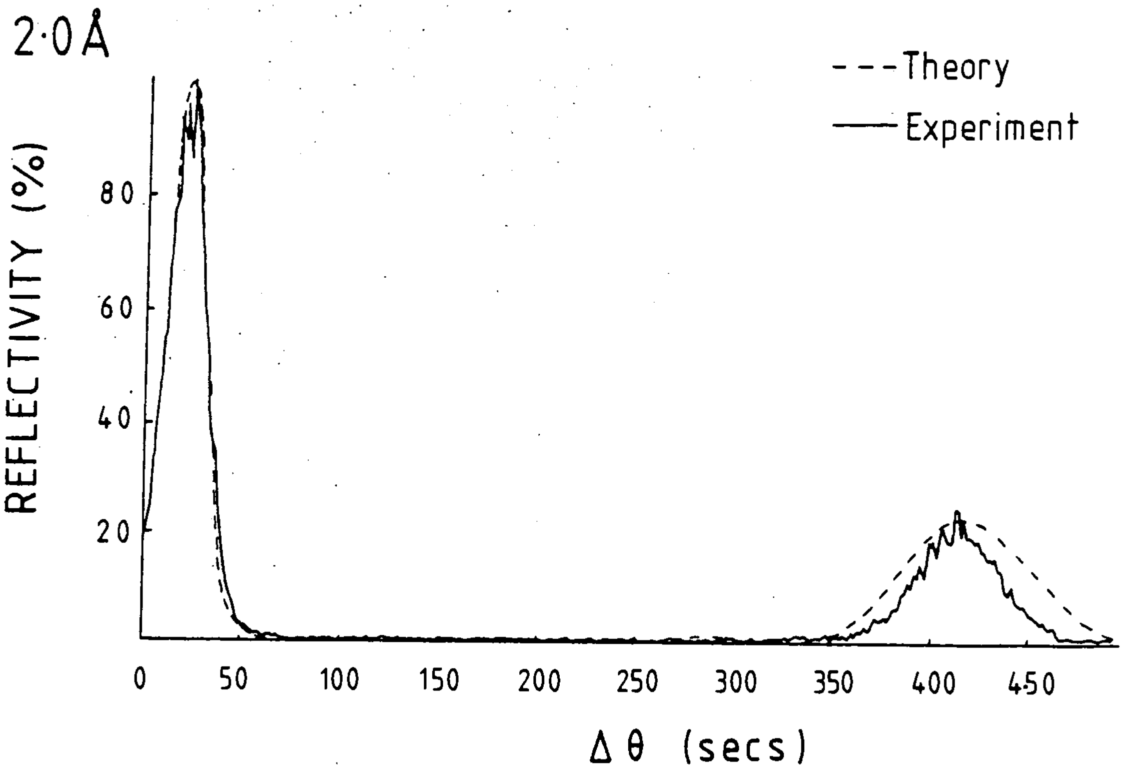
The 115 rocking curve was also measured at 1.5 Å and is shown in fig. 5.8(b). The diffracted intensity from the layer is less than that for the 004 reflection, at approximately 5% of the substrate peak intensity. This is expected since the 115 structure factors are smaller than those for the 004 reflection, giving a larger extinction distance, and the asymmetry factors are not yet important since θ-φ is still 25.8°. The theoretical curve is also shown in fig. 5.8(b) and again agrees well with the experimental curve. Since the positions of the peaks are in good agreement the parallel mismatch must be negligible. Clearly, there is no advantage in using this reflection over the 004 reflection except when the parallel mismatch is to be determined.

**TABLE 5.2**

**Values of the peak half width and extinction distance for the 224 and 404 reflections as a function of wavelength for an (001) InP crystal.**

| Wavel. (Å) | 224 Reflection | | 404 Reflection | |
|---|---|---|---|---|
| | FWHM (secs) | Extin. Len. (μm) | FWHM (secs) | Extin. Len. (μm) |
| 2.00 | 14.96 | 5.99 | 19.72 | 7.09 |
| 1.80 | 14.04 | 5.30 | 13.63 | 6.09 |
| 1.60 | 15.52 | 4.17 | 15.95 | 4.20 |
| 1.58 | 15.94 | 4.01 | 16.98 | 3.88 |
| 1.56 | 16.45 | 3.84 | 18.24 | 3.55 |
| 1.54 | 17.10 | 3.65 | 20.09 | 3.18 |
| 1.52 | 17.90 | 3.44 | 22.98 | 2.73 |
| 1.50 | 18.96 | 3.21 | 28.44 | 2.18 |
| 1.48 | 20.37 | 2.96 | 44.41 | 1.38 |
| 1.47 | 21.70 | 2.75 | 93.78 | 0.65 |
| 1.46 | 22.38 | 2.66 | - | - |
| 1.44 | 25.46 | 2.31 | - | - |
| 1.42 | 30.96 | 1.88 | - | - |
| 1.40 | 44.88 | 1.28 | - | - |
| 1.39 | 70.03 | 0.82 | - | - |

The asymmetry factor becomes especially important for the 224 and 404 reflections, where below 1.6 Å the value of θ-φ is less than 7°. Values of the extinction distance for the 224 and 404 reflections, with a low incidence angle, as a function of wavelength are shown in Table 5.2. Since the



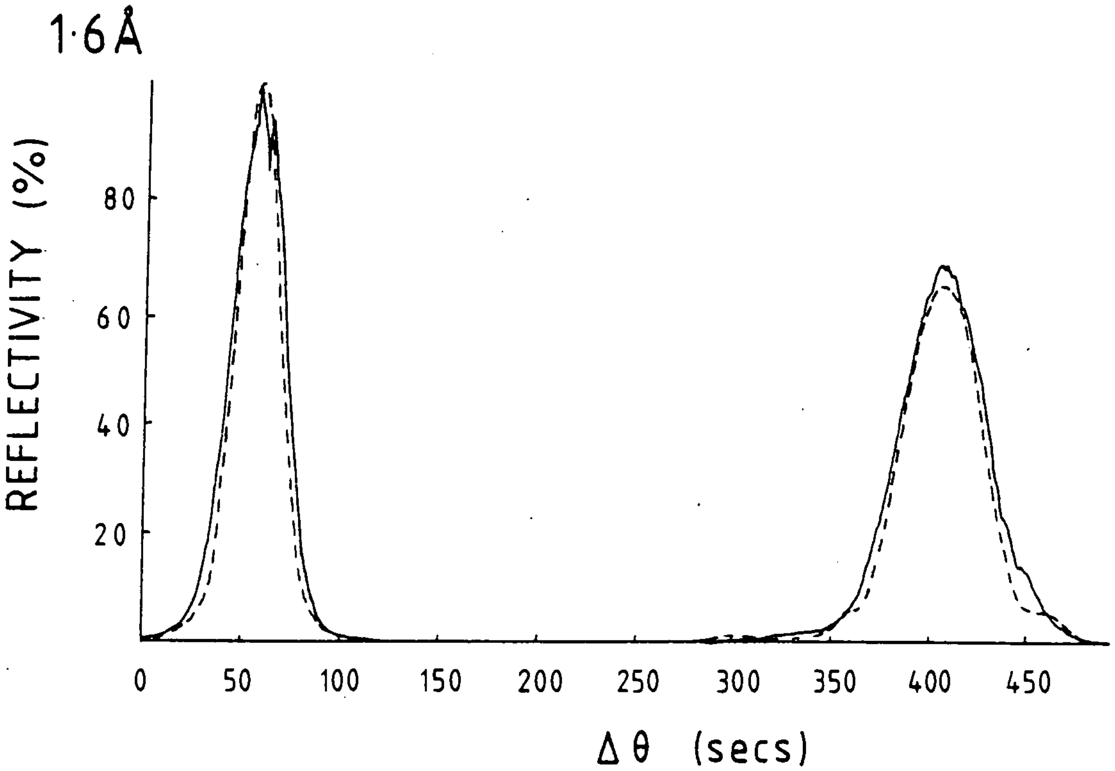
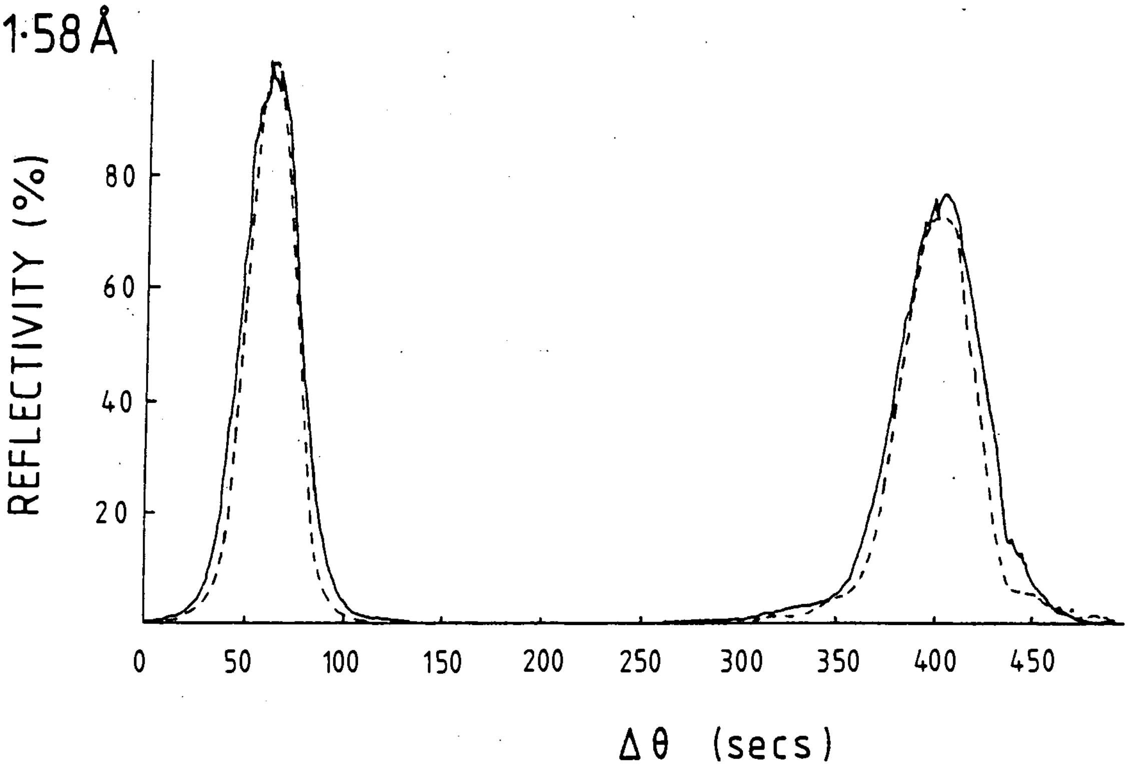
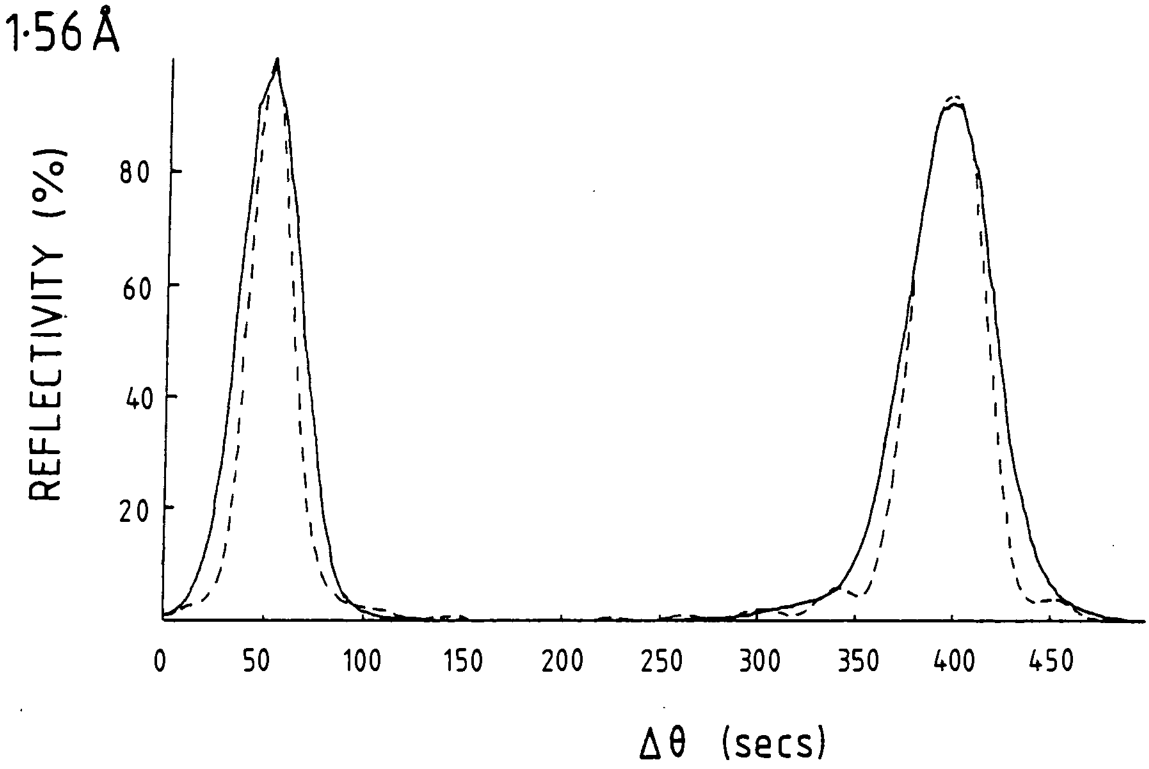
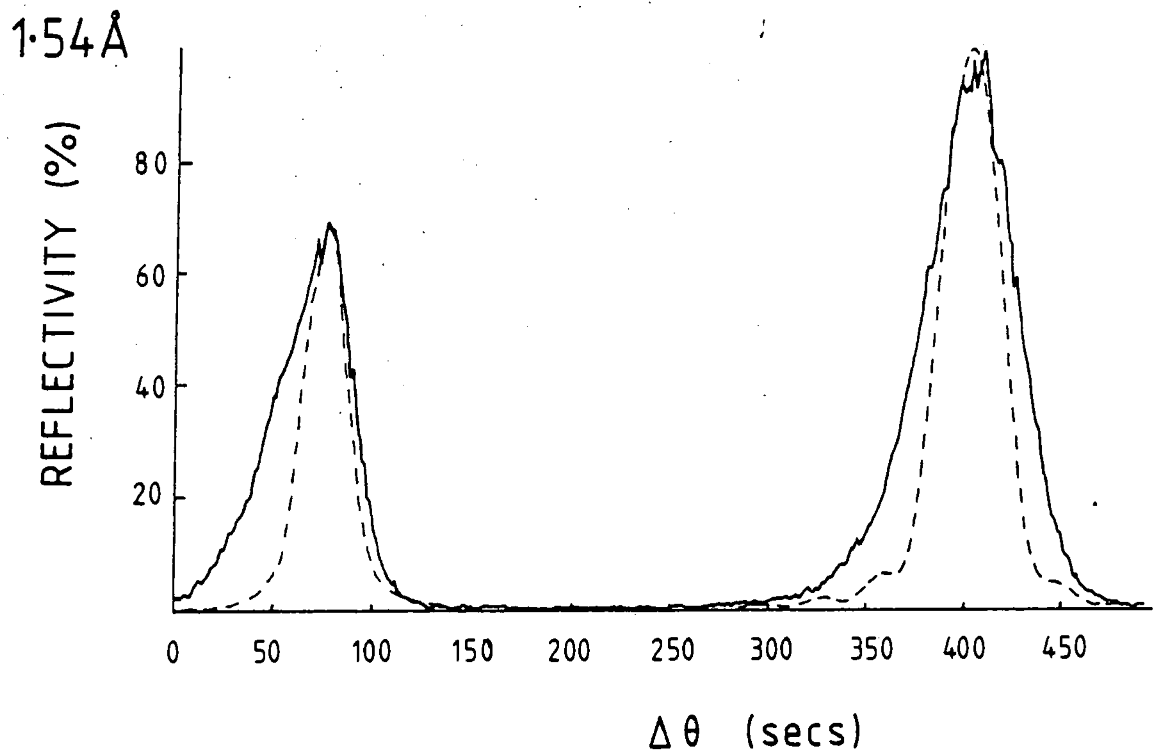
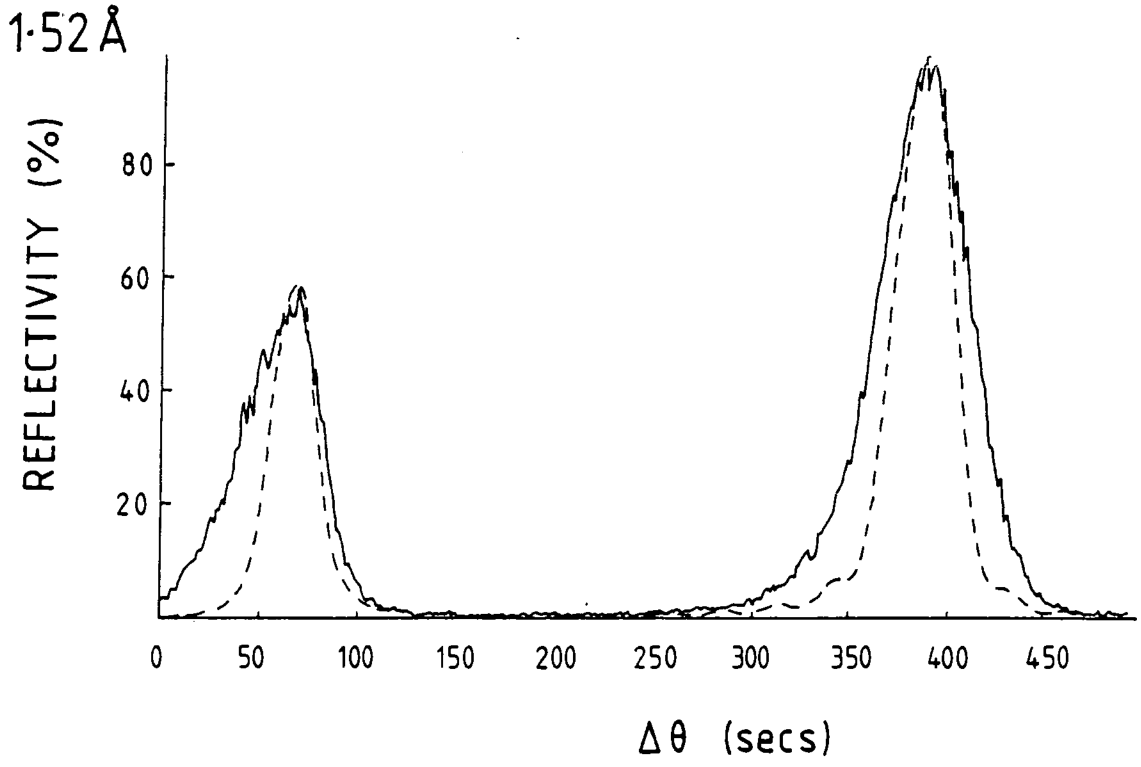


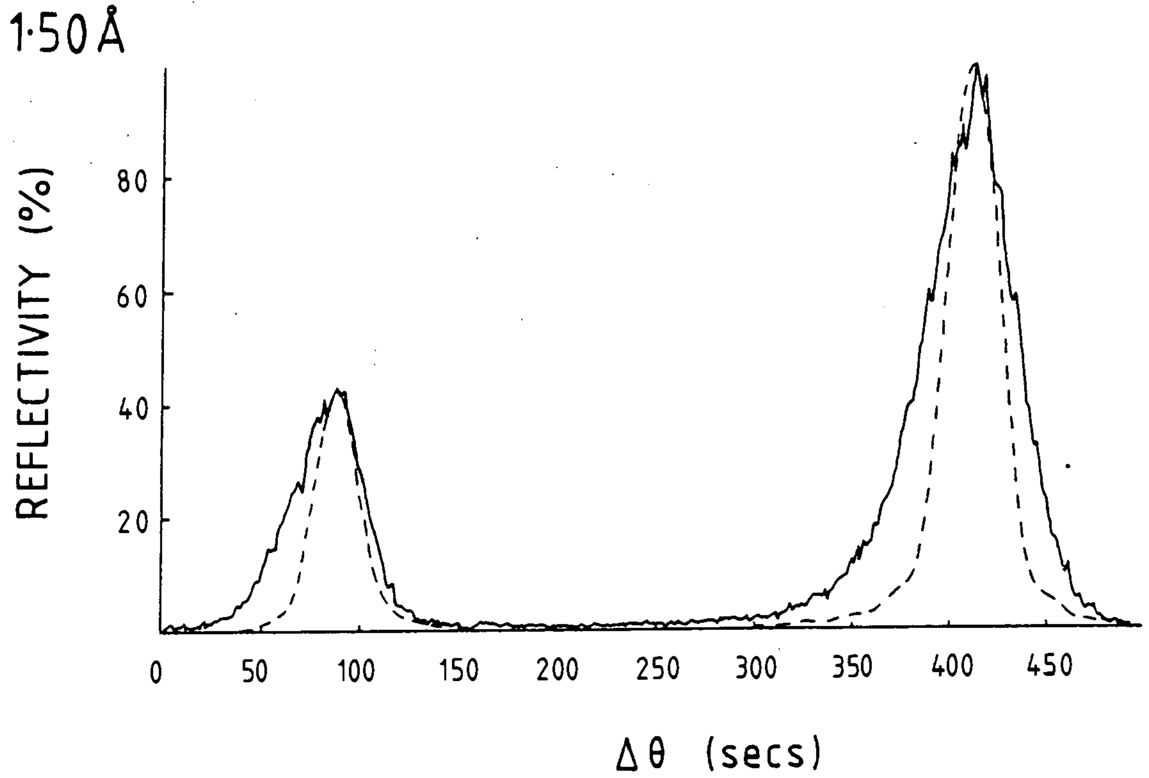
Fig. 5.9 Series of 224 experimental and computed rocking curves from the same sample used in fig. 5.8 at various wavelengths in the range 2.0 Å to 1.40 Å.

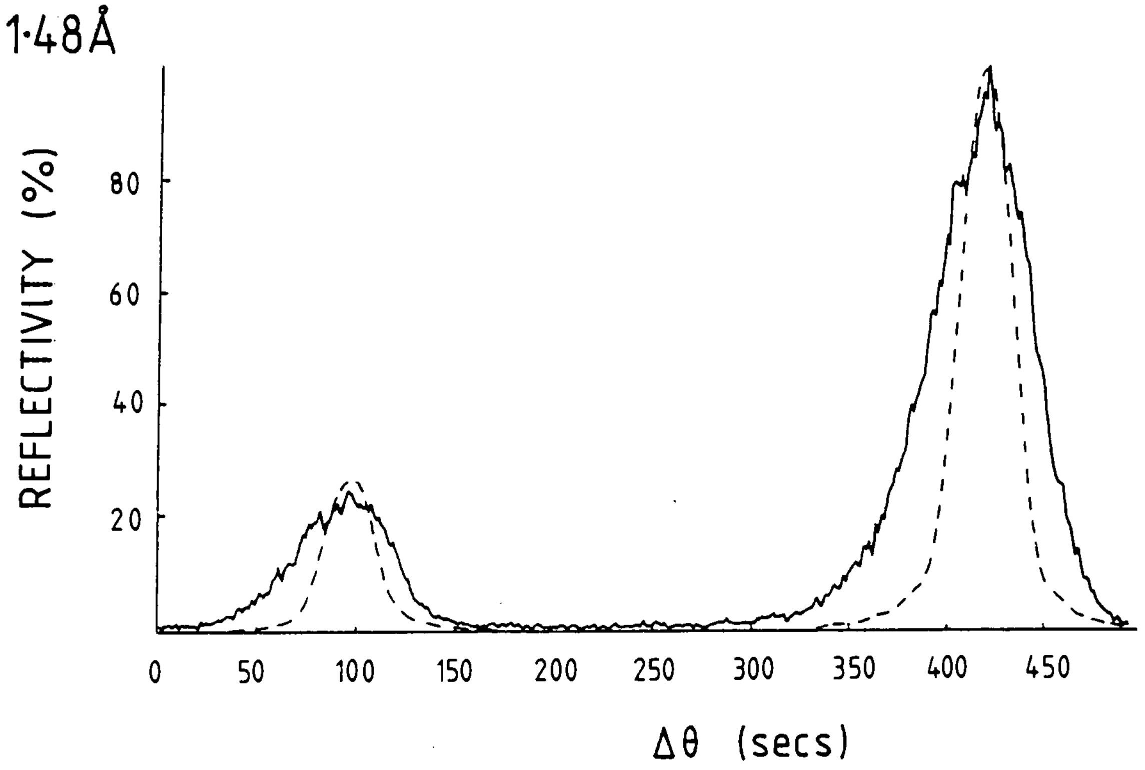

1·58 Å
REFLECTIVITY (%)
80
60
40
20
0
50
100
150
200
250
300
350
400
450
Δθ (secs)


1·56 Å
REFLECTIVITY (%)
80
60
40
20
0
50
100
150
200
250
300
350
400
450
Δθ (secs)

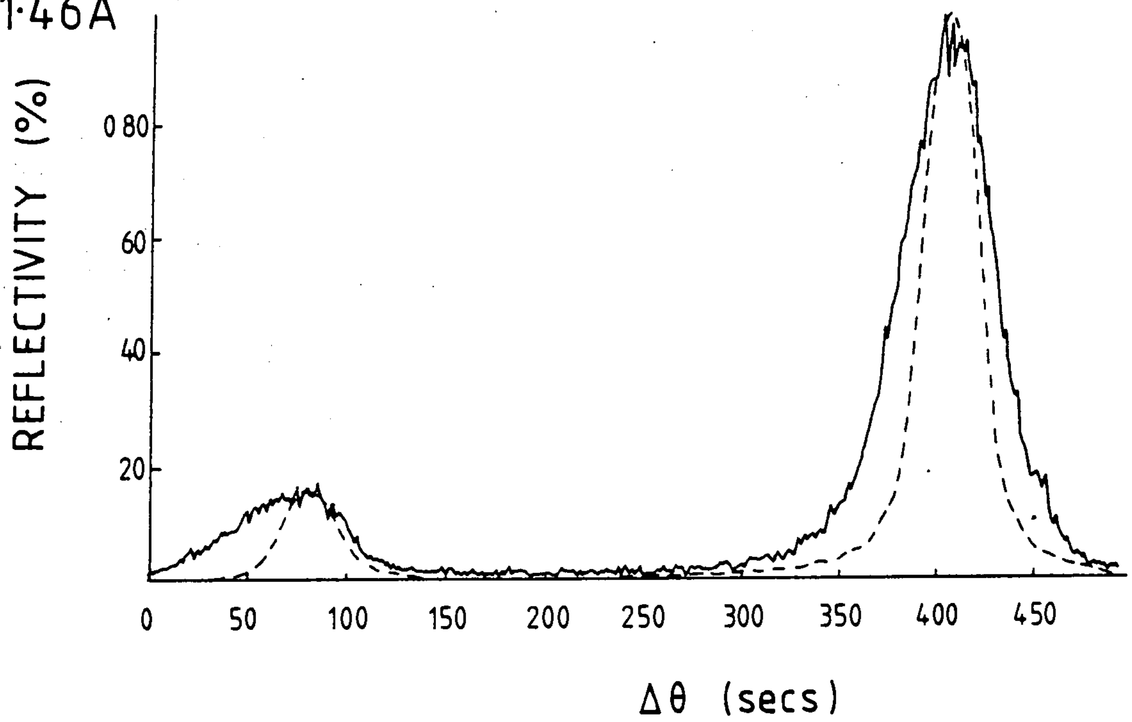

1·54Å
REFLECTIVITY (%)
80
60
40
20
0
50
100
150
200
250
300
350
400
450
Δθ (secs)


1·52Å
REFLECTIVITY (%)
80
60
40
20
0
50
100
150
200
250
300
350
400
450
Δθ (secs)

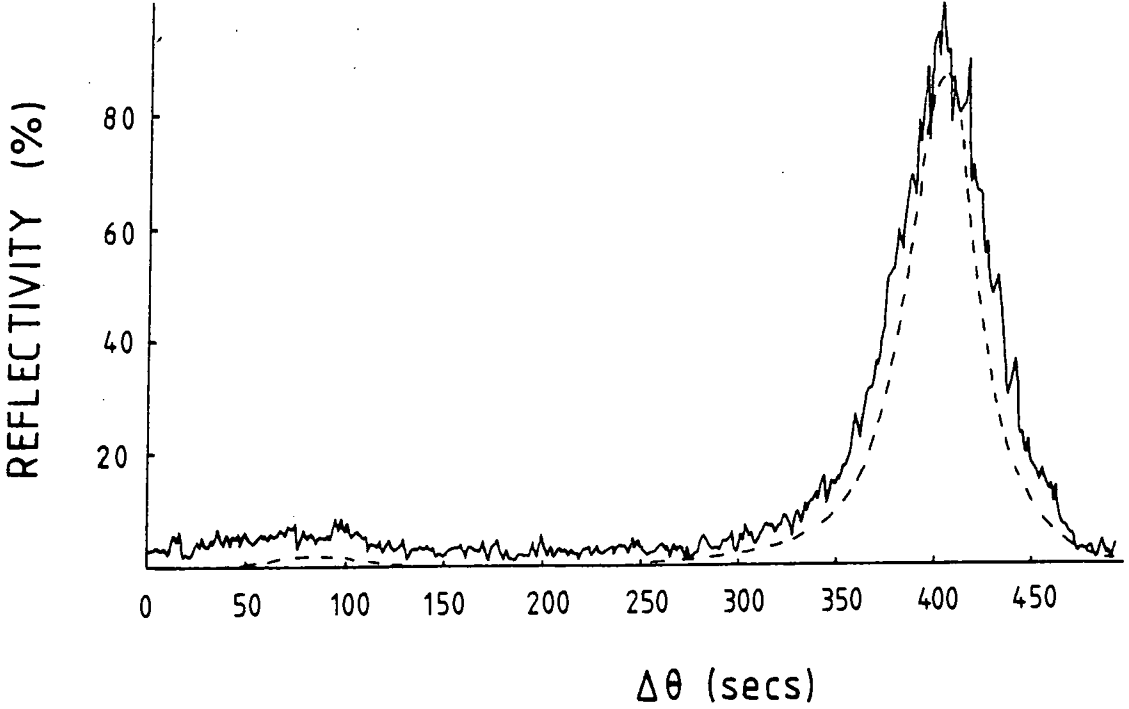

1·50Å
REFLECTIVITY (%)
20
40
60
80
0
50
100
150
200
250
300
350
400
450
Δθ (secs)

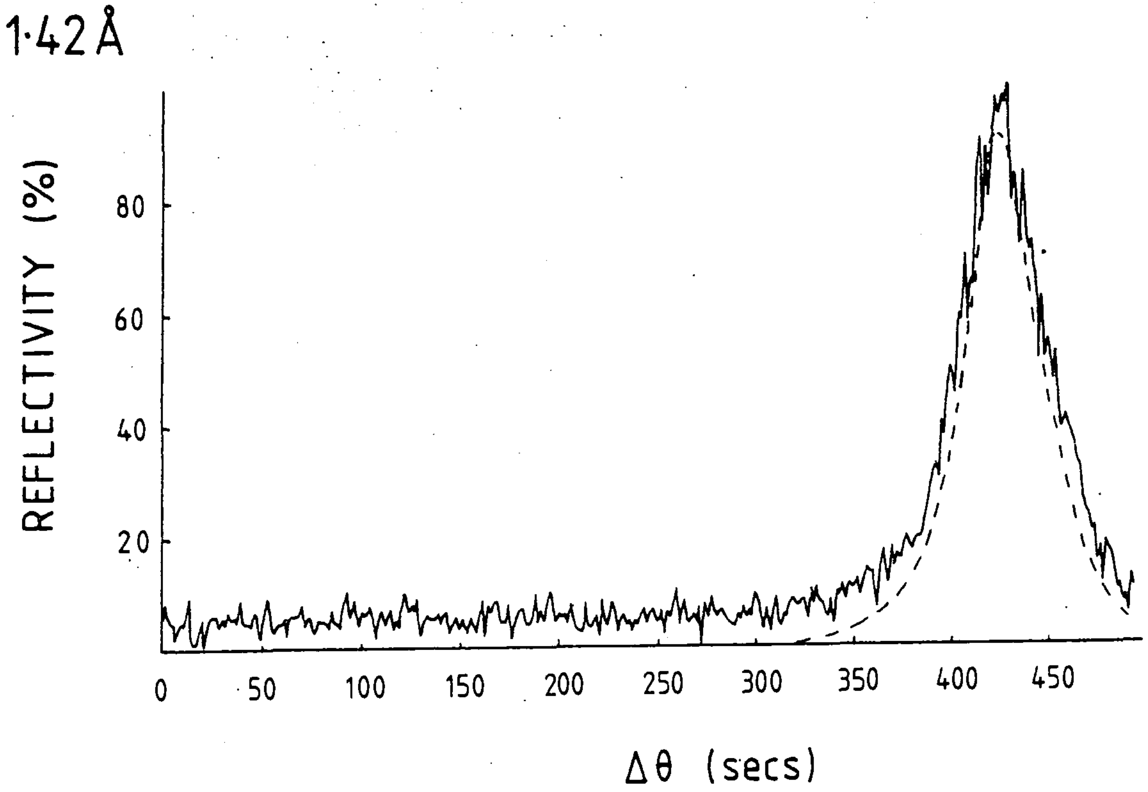

1·48Å
REFLECTIVITY (%)
20
40
60
80
0
50
100
150
200
250
300
350
400
450
Δθ (secs)

1·46Å


REFLECTIVITY (%)
0 80
60
40
20
0
50
100
150
200
250
300
350
400
450
Δθ (secs)

1·44Å


REFLECTIVITY (%)
80
60
40
20
0
50
100
150
200
250
300
350
400
450
Δθ (secs)


1·42 Å
REFLECTIVITY (%)
80
60
40
20
0
50
100
150
200
250
300
350
400
450
Δθ (secs)

extinction distance decreases so rapidly the peak half width must similarly be affected and values for half widths of the 224 and 404 reflections are also shown in Table 5.2. Fig. 5.9 shows a series of 224 rocking curves recorded at the SRS and using the black-box monochromator. The wavelength is easily altered with this arrangement and over the range 1.6 A to 1.4 Å the height of the X-ray beam alters very little. A low angle of incidence was used with a 100 µm vertical slit used to limit the area of sample illuminated. Since, even with a very narrow slit, a long length of sample is illuminated due to the low incidence angle an exit slit infront of the detector was used to reduce the area of sample contributing to the rocking curve.

The 1.60 Å rocking curve shows that even at 6.6° incidence angle the diffracted intensity from the layer is already greatly increased over the 004 and 115 reflections. The layer peak is broader than the substrate peak and about 75% of its intensity. Clearly, already there is little doubt as to the usefulness of this reflection for the characterisation of layers with thicknesses less than 0.5 µm. As the wavelength is further reduced, the layer peak rapidly becomes more intense at the expense of the substrate peak. At about 1.54 Å the peaks are almost equally intense with the layer peak again being the broadest. Due to the very low angle of incidence the effect of bulk sample curvature is especially important and further increases the width of the peaks as the wavelength is decreased. At 1.52

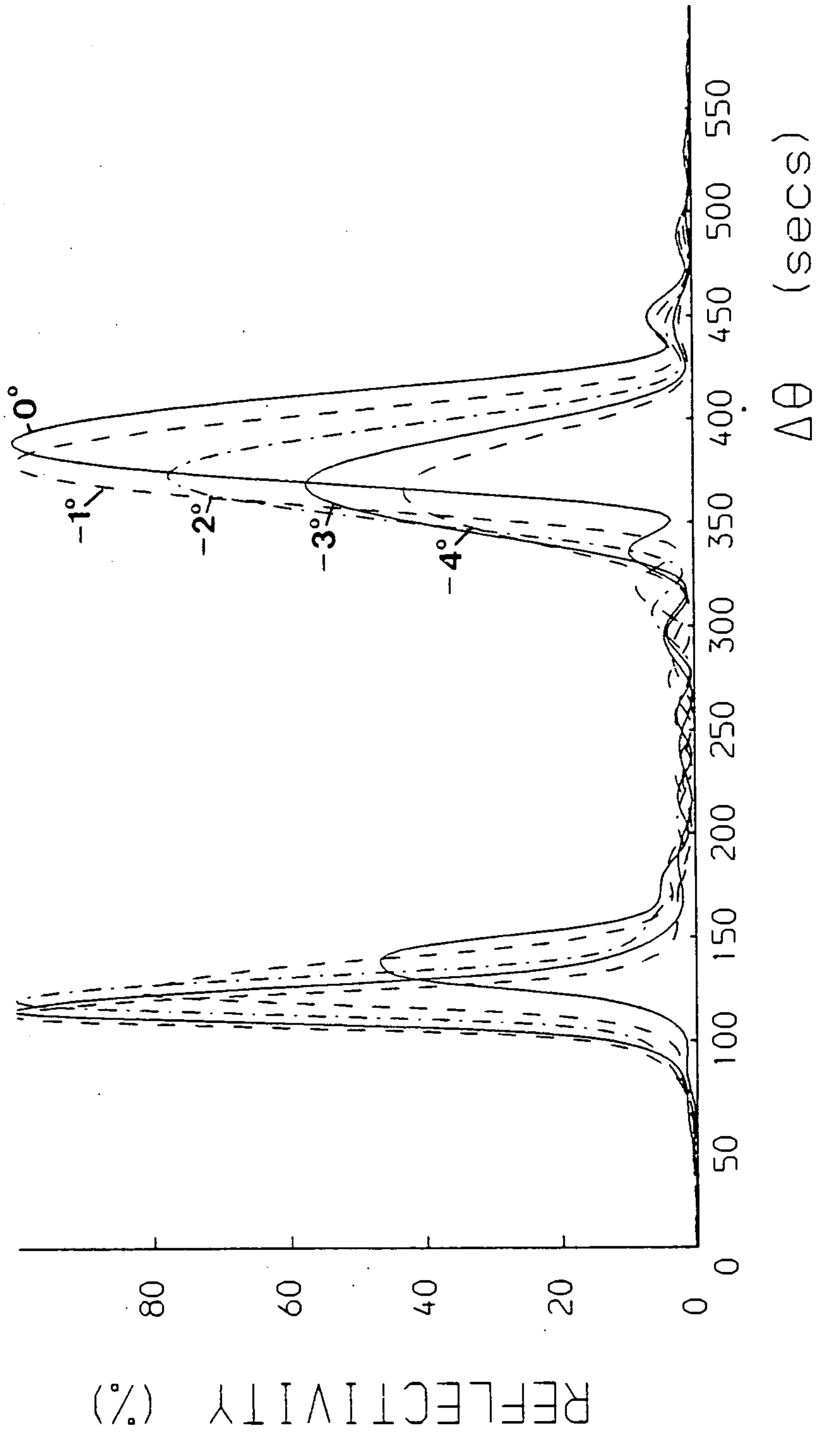


Fig. 5.10 Theoretical 224 reflections for a 0.2 µm thick -600 ppm mismatched GaInAs layer for different values of surface misorientation. Positive angles are taken to decrease the angle between the surface and the x-ray beam for the (θ-ϕ) angle of incidence case. The wavelength used is 1.6 Å with σ polarisation.

A and below the substrate peak rapidly diminishes and becomes noticeably asymmetric, with a more intense tail on the low angle side. The overall diffracted intensity is also rapidly diminished as the wavelength is further reduced giving rise to considerably more noisy rocking curves. At these low incidence angles the X-ray beam will become increasingly strongly reflected from the crystal surface due to refraction and less intensity will be available in the crystal to be diffracted. From the refractive index of X-rays the critical angle for total external reflection is approximately 0.5°. At 1.40 Å, which corresponds to an incidence angle of only 0.49° only very weak diffraction from the layer can be observed with no intensity detectable from the substrate.

Since the rocking curve is, clearly, so dependent on the value of $\theta - \varphi$ under these conditions it will be particularly sensitive to any surface misorientation. This effect is shown in fig. 5.10 where rocking curves for a 0.2µm, -600 ppm mismatched GaInAs layer for 1.6 Å radiation have been calculated including an increasing additional tilt angle. The negative sign indicates that the additional angle increases the incidence angle with respect to the crystal surface. We note how sensitive the layer peak is to this tilt angle, with a -4° tilt reducing its intensity to less than 50% of its value with no additional tilt. Substrates are often cut upto 3° off orientation in order to improve layer growth. With MBE growth the steps thus produced on the surface are usually filled up before the layer continues to grow but this may not be the case for LPE grown layers. Consequently, this surface misorientation must be taken into account when computing rocking curves as best fits to the experimental curves. This appeared to be particularly true for this sample as the layer intensities in the calculated curves were considerably greater than those in the experimental curves at the same wavelengths. Although there could easily be an error of 0.05 Å in the wavelength used in the experiment this would not account for these discrepancies and, additionally, the peak separations would not be consistent. However, by including a surface misorientation of 2°, in a direction such as to increase the incidence angle, reasonable fits could be made at the wavelengths expected.

Fig. 5.9 shows such calculated rocking curves for comparison with the experimental curves. Agreement between these curves is reasonable, particularly at the higher wavelengths. At the lower wavelengths used the asymmetry exhibited by the experimental substrate peak is not present in the calculated curve. This is not unexpected as the calculations do not include the effect of the X-ray beam reflected from the surface. Hartwig (1977,1978) has pointed out that for highly asymmetric reflections this reflected beam cannot be neglected and the conventional two beam approximation can no longer be used. He showed that the inclusion of this beam in the calculation of the Bragg reflection profile does produce asymmetric peaks. Note that artificial symmetry of the rocking curve due to a parallel (+-) arrangement will not be present in rocking curves recorded with the black box monochromator. Additionally, any changes in the layer composition and thickness over the area of the sample will also increasingly affect the rocking curve as the wavelength is reduced, due to the increased area of the sample illuminated. Further, since the rocking curve is also more sensitive to thin regions of the layer any rapid changes in composition near the layer/substrate interface, as reported for LPE grown layers, will also become important.

Although the layer used was not especially thin we have seen that the use of highly asymmetric reflections does lead to a greatly increased layer intensity making characterisation rather more straightforward. The use of reflections where the totally externally reflected beam from the crystal surface cannot be neglected increases the difficulty of calculating rocking curves to best fit the experimental ones and, consequently, hinders the determination of layer thickness. Therefore, 224 reflections at approximately 1.50 Å are more useful. Ample intensity is diffracted by the layer for the peak position to be accurately determined and theoretical curves can easily be computed that agree well with the experimental curve, thus allowing a reasonably accurate determination of layer thickness.

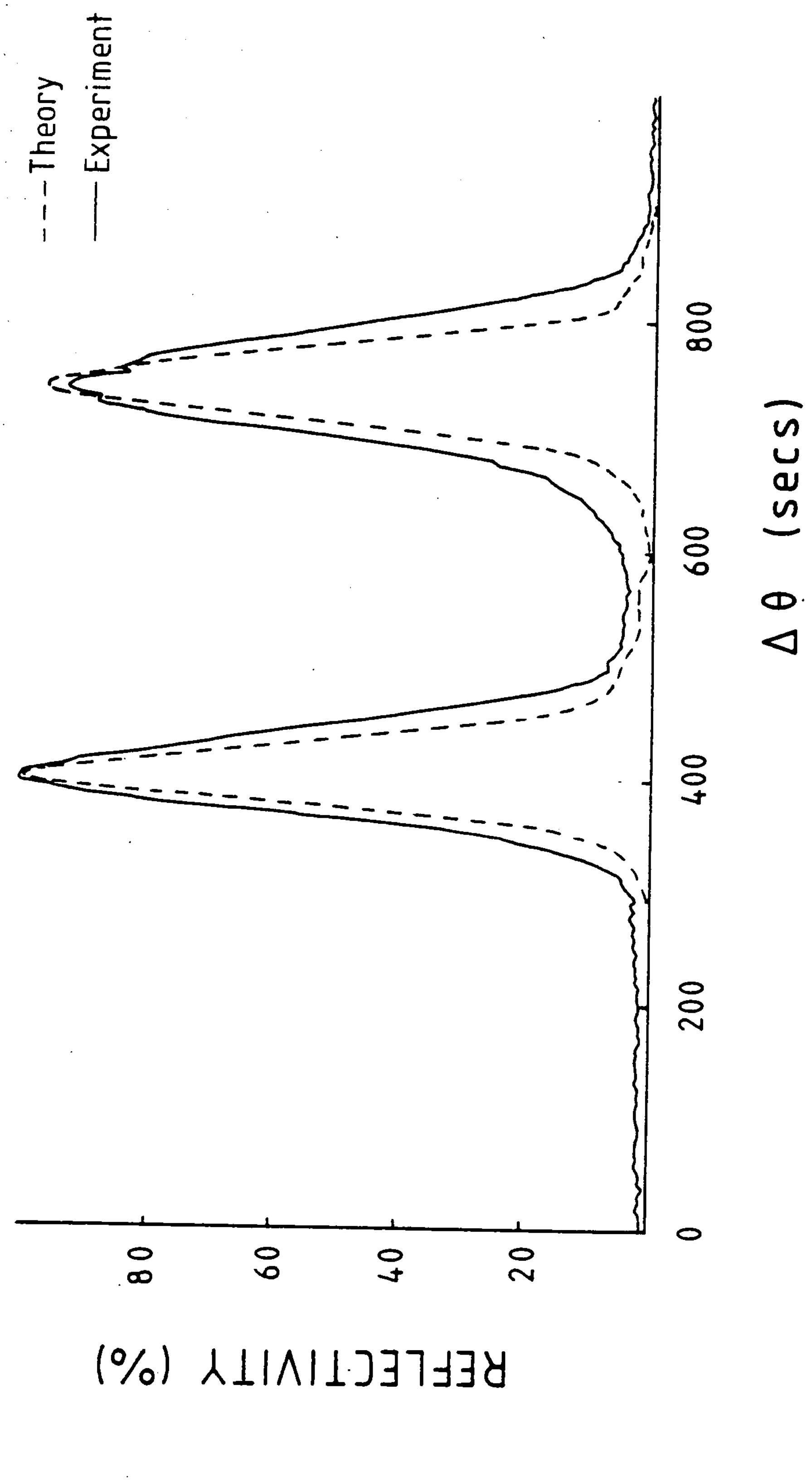


Fig. 5.11 Experimental and computed 404 rocking curves for the same sample used in fig. 5.9 at a wavelength of 1.55 Å.

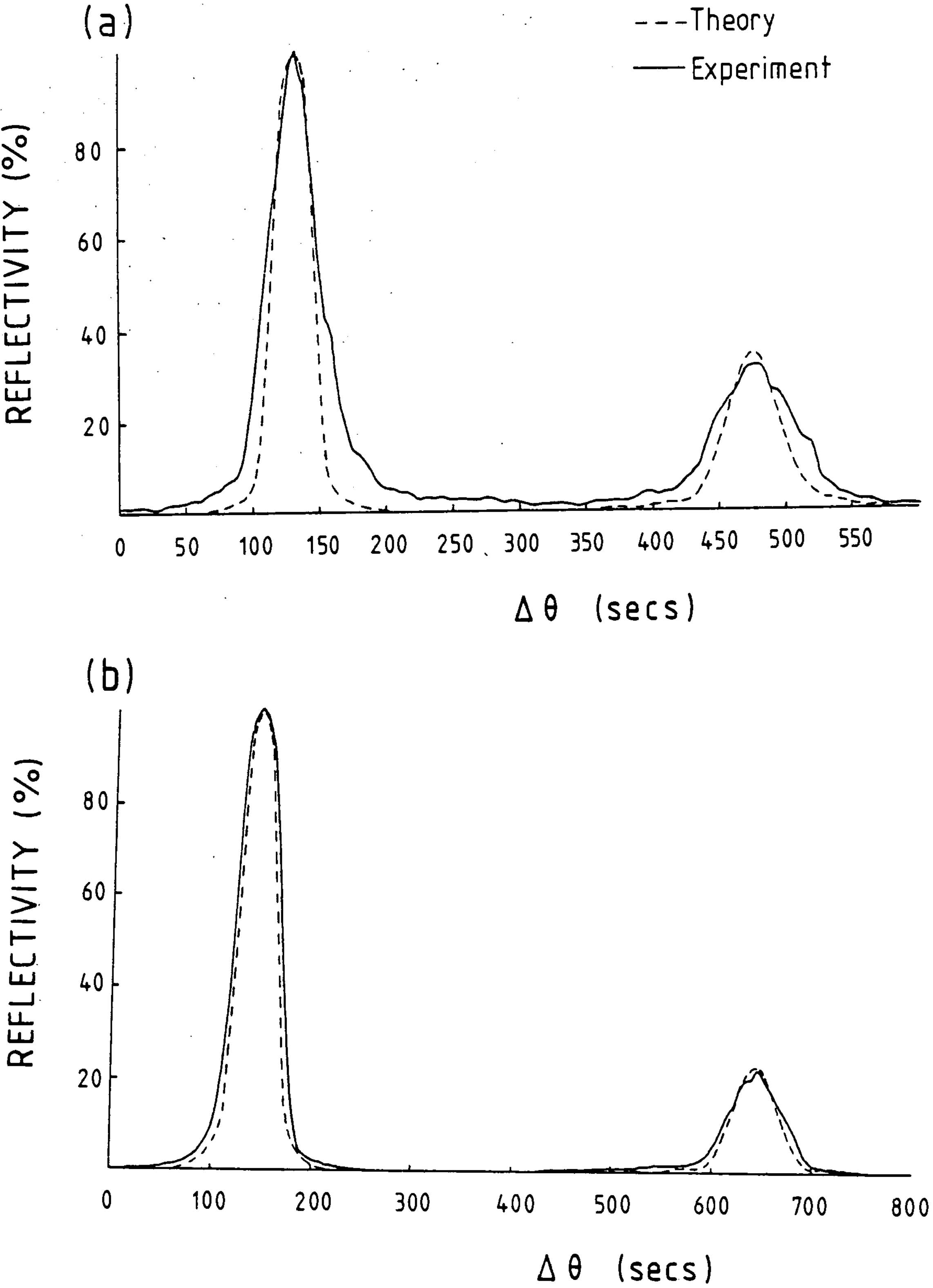


Fig. 5.12 (a) Experimental and theoretical 004 rocking curves from a 0.45 µm thick, $\lambda$ = 10 µm, -1100 ppm mismatched GaInAsP layer on an InP substrate. The wavelength is 1.54 Å.

(b) The experimental and theoretical 115 rocking curves from the same sample at 1.5 Å.

Fig. 5.9(a) shows the experimental and calculated rocking curves for the same sample at 2.0 Å. Since the asymmetry factor is no longer as important the layer peak is reduced to a similar height to that observed in the 115 and 004 reflections. Consequently, there is no advantage in using this reflection at similar wavelengths over the 004 reflection. If only a non-tunable wavelength source is available, as is a conventional X-ray generator, good results can be obtained with CuK$\alpha_1$ radiation.

Similar effects are observed with the 404 reflection at wavelengths in the range 1.45 Å to 1.7 Å. Fig. 5.11 shows the experimental and calculated rocking curves for this reflection with the same sample at 1.55 Å. Again the layer diffracts considerably more intensity than from the 004 and 115 reflections making its position and integrated intensity easily measured. The 404 reflection is slightly weaker than the 224 reflection and consequently has a larger extinction length at the longer wavelengths. The calculated rocking curve again agrees reasonably well with the experimental curve although both experimental peaks are considerably broader than those in the computed curve. A layer thickness of 0.3μm again gave the best agreement.

Similar results were obtained with another sample with a 0.45 μm, λ=1.0 μm, GaInAsP layer. The 004 rocking curve at 1.54 Å is shown in fig. 5.12(a) giving the mismatch as -1100 ppm and the calculated rocking curve, assuming a layer thickness of 0.45 μm, is also shown in fig. 5.12(a). The

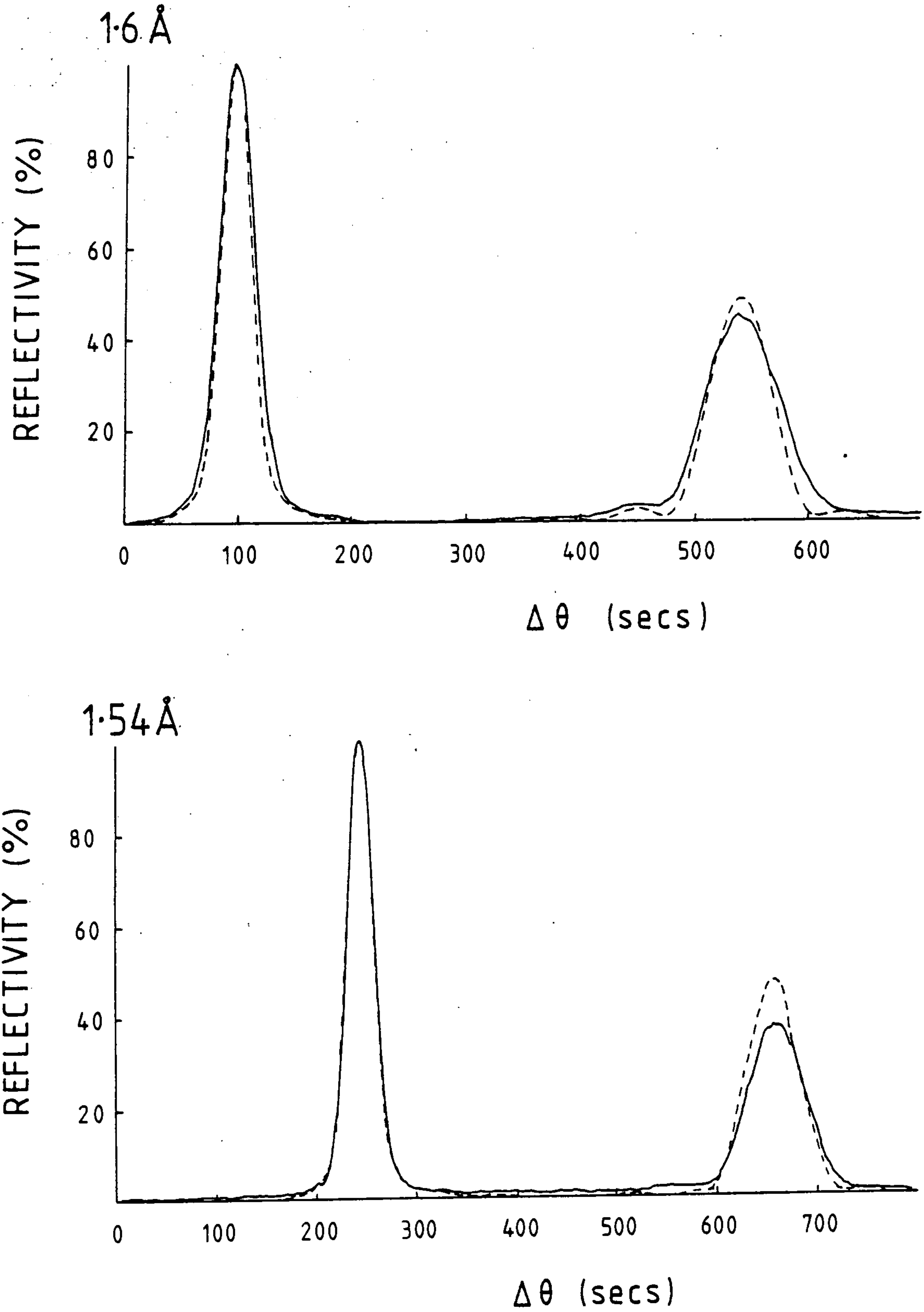
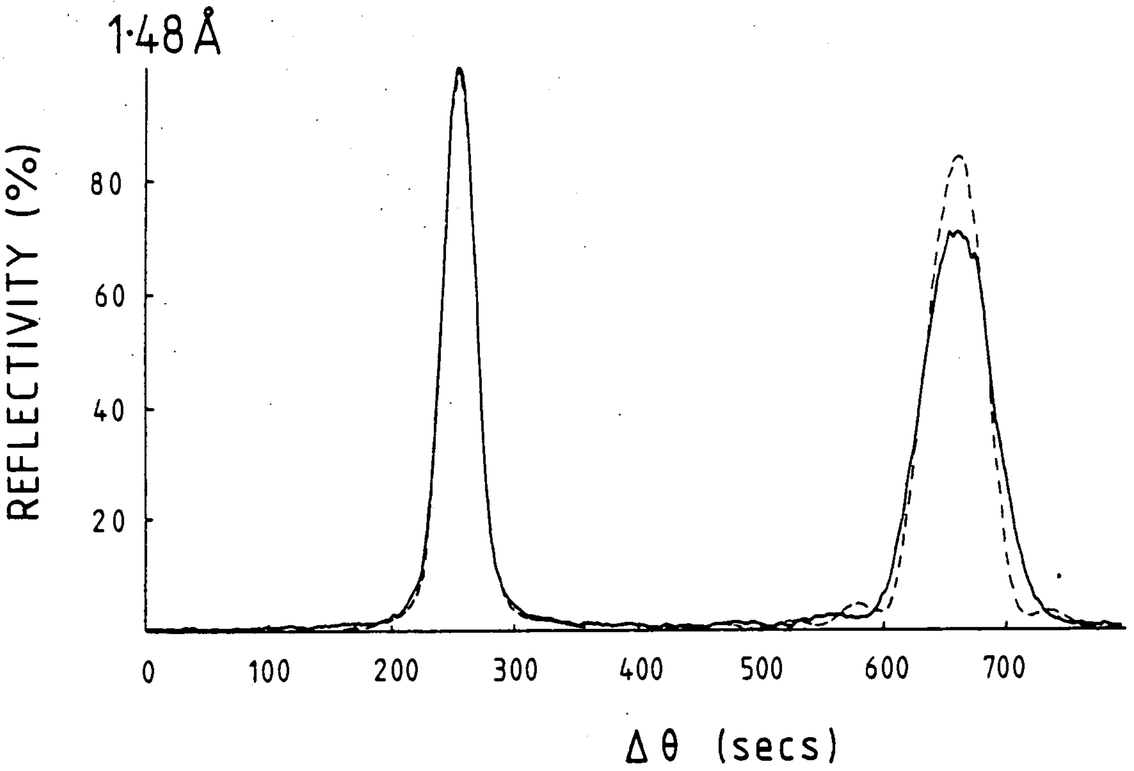
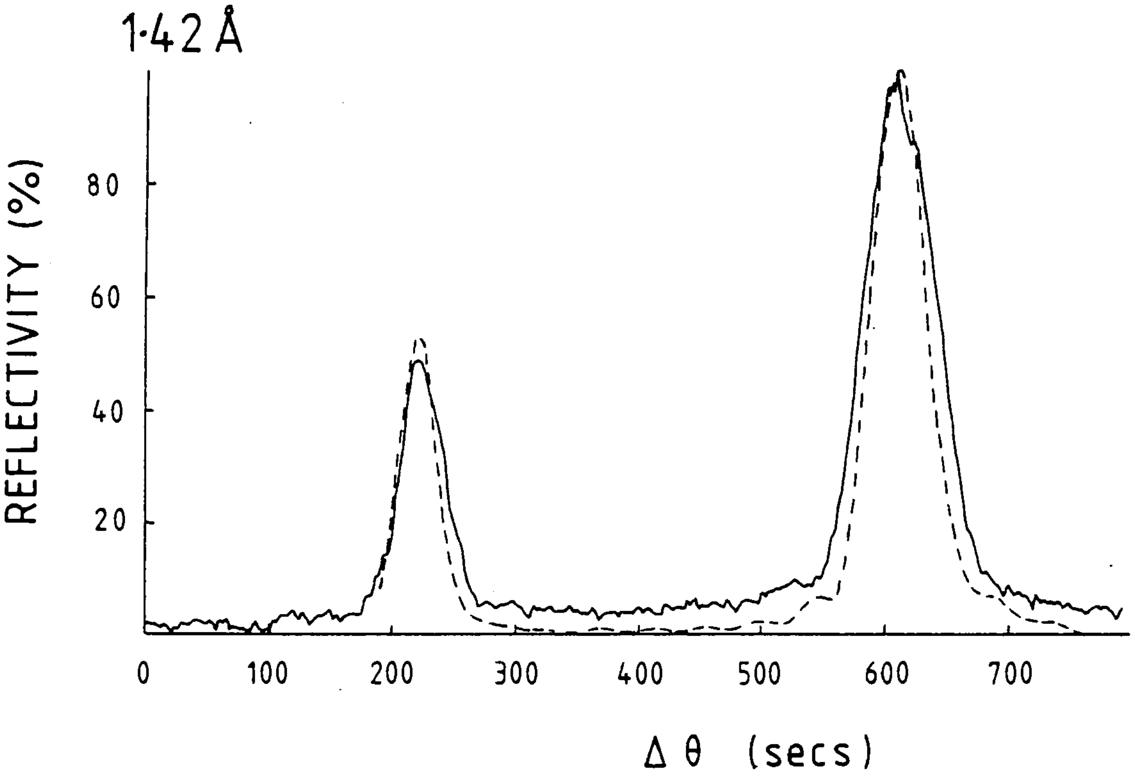


Fig. 5.13 224 experimental and computed rocking curves from the same sample used in fig. 5.12. Wavelengths in the range 1.6Å to 1.42Å. The computed curves are for a layer thickness of 0.2μm.


1·48 Å
REFLECTIVITY (%)
80
60
40
20
0
100
200
300
400
500
600
700
Δ θ (secs)


1·42 Å
REFLECTIVITY (%)
80
60
40
20
0
100
200
300
400
500
600
700
Δ θ (secs)

composition of the layer is determined to be $Ga_{0.09}In_{0.91}As_{0.15}P_{0.85}$. Once again agreement is particularly good with the effect of sample curvature included in the calculation to give equal half widths for the substrate peak. Measurement of the rocking curve with the X-ray path reversed indicated that there was no tilt angle present between the layer and substrate. The 115 reflection was also measured at 1.5 Å and the rocking curve is shown in fig. 5.12(b), again showing only small intensity from the layer. The computed rocking curve is also shown in fig. 5.12(b) and again agrees reasonably well.

Fig. 5.13 shows a number of 224 rocking curves recorded for this sample over the range 1.6 A to 1.42 Å. The rocking curve behaves in a similar manner that of the previous sample, with the layer peak rapidly becoming more intense than the substrate peak. Good agreement between computed and these experimental curves could only be achieved for a layer thickness of 0.2 µm, suggesting that a different region of the sample was being illuminated by the X-rays. Since the layer did not extend over the entire region of the substrate it is quite possible that a thinner region near to the edge of the layer was studied. Computed rocking curves for this layer thickness are also shown in each figure for comparison and good agreement is obtained. The asymmetric peaks shown in the experimental curves from the previous sample are not observed in these rocking curves. similar agreements obtained as with the previous sample.

In all of the above examples a low incidence angle was used, giving $\gamma_0$ less than $|\gamma_h|$. A high incidence angle could also be used which would similarly decrease the extinction length, due to the factor $(\gamma_0|\gamma_h|)^{\frac{1}{2}}$, but would decrease the half width of the peaks, due to the factor $(|\gamma_h|/\gamma_0)^{\frac{1}{2}}$. At these extremely low exit angles the exit beam would be increasingly totally internally reflected at the crystal surface (Hartwig, 1978). Consequently, the two beam approximation used in the calculation of the rocking curve would again be in error. Hartwig has shown that in this case peaks with an even higher asymmetry are produced.

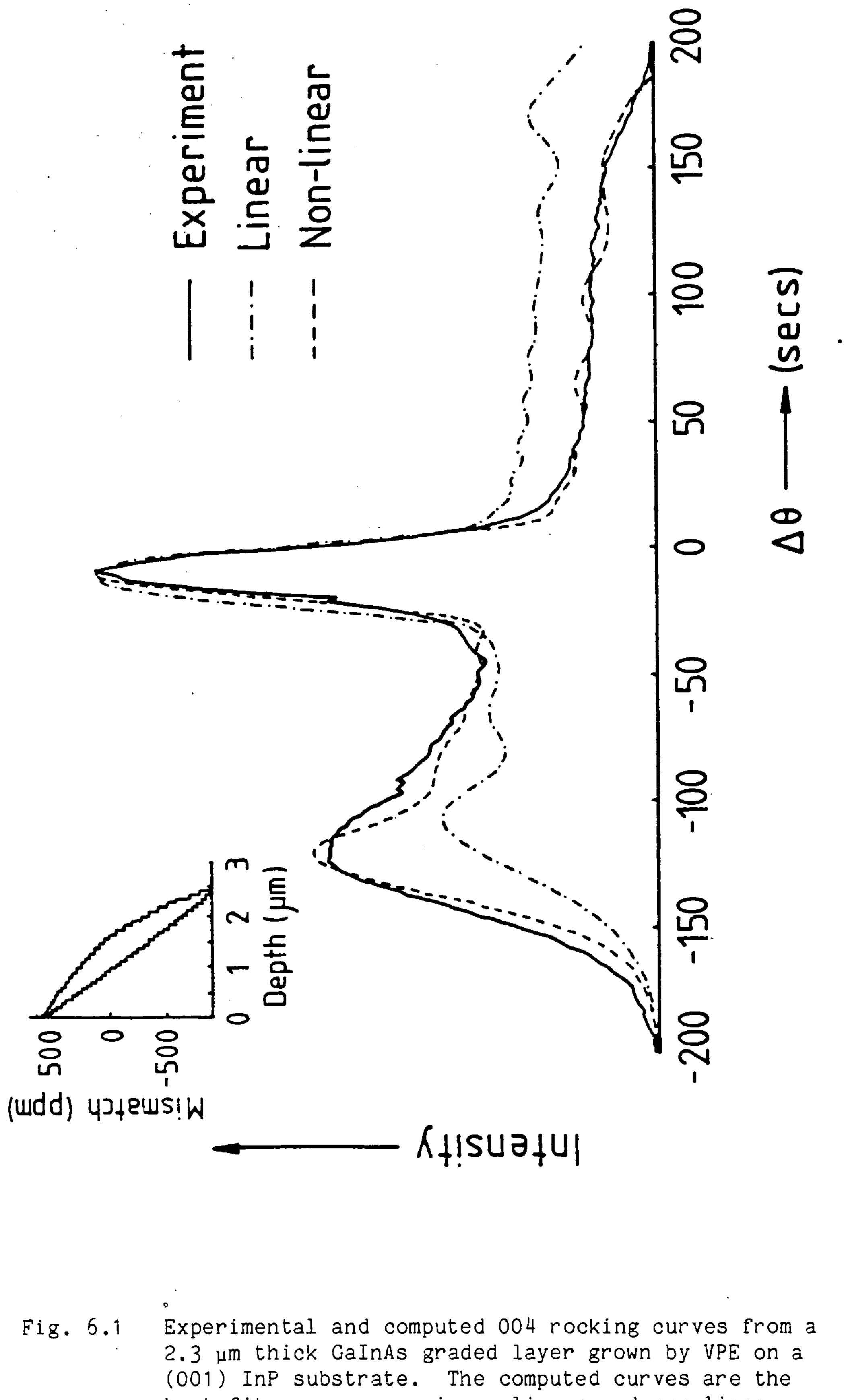


Fig. 6.1 Experimental and computed 004 rocking curves from a 2.3 μm thick GaInAs graded layer grown by VPE on a (001) InP substrate. The computed curves are the best fit curves assuming a linear and non linear composition variation. The composition variations used are shown in the inset. CuK$\alpha_1$ radiation was used.

# CHAPTER 6

# SINGLE LAYERS WITH A DEPTH DEPENDENT COMPOSITION

Epitaxial layers in which the composition varies continuously as a function of depth are often referred to as graded layers and are generally grown by the VPE method. For (Ga,In)As ternary layers this composition change can be attributed to a change in the concentration of the two type III species as a function of time (see Chapter 1).

The usual 004 double crystal rocking curves from such structures often appear similar to that shown in fig. 6.1. This rocking curve was recorded with CuK$\alpha_1$ radiation from a GaInAs layer grown on a 001 InP substrate by VPE. The narrow central peak visible in the curve is from the substrate while the broad, asymmetric peak extending on either side of substrate peak is from the layer. Since the layer peak extends on both sides of the substrate peak the composition must vary to give lattice parameters both larger and smaller than that of the substrate. The wedge-like shape of the layer peak, having an almost level top with only small undulations, immediately suggests that the lattice parameter decreases continuously with depth since the section of the peak on the low angle side of the substrate peak, corresponding to a larger lattice parameter, has the greatest intensity and, therefore, will be due to the region of the layer nearer to the surface. For VPE grown layers we expect that the composition change should be

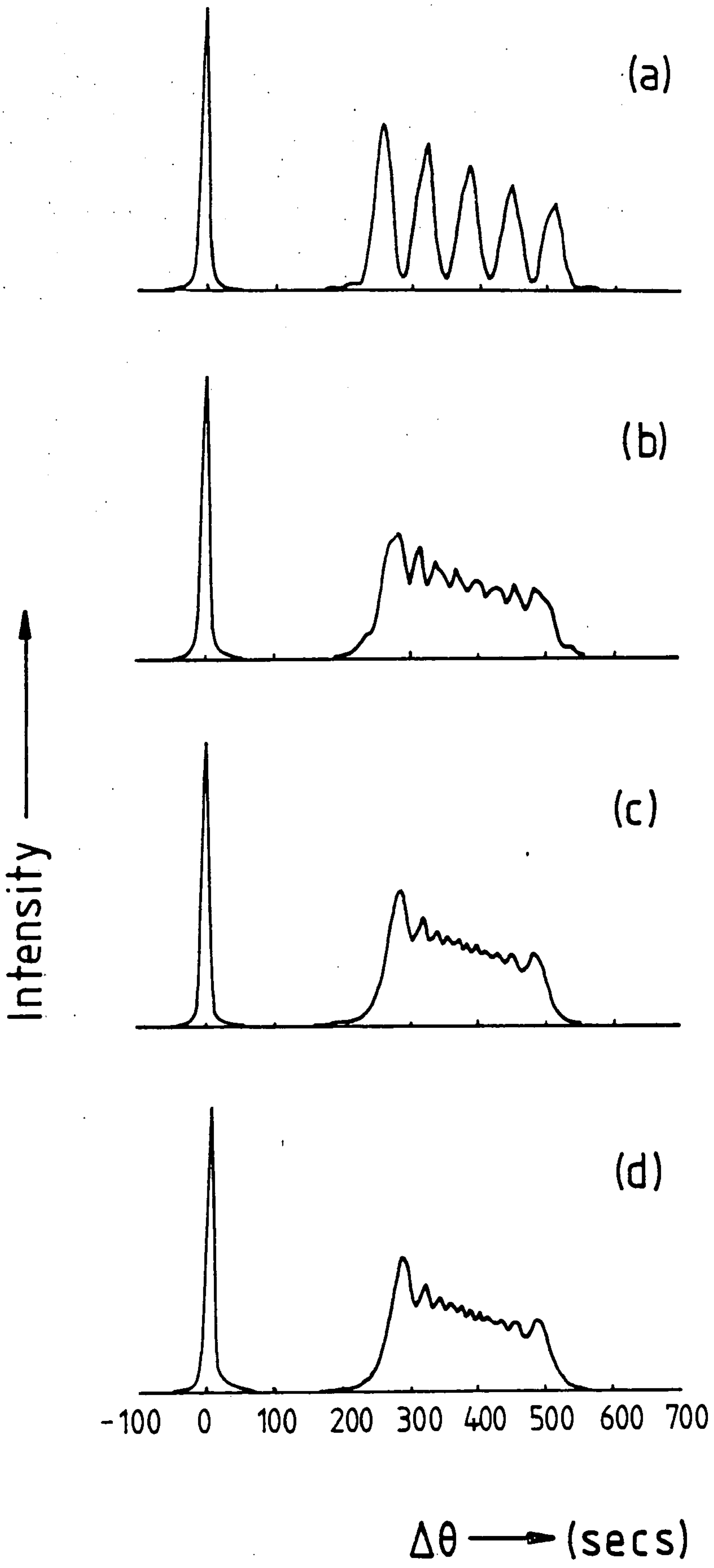


Fig. 6.2 004 computed rocking curves for a 3µm thick GaInAs graded layer with a mismatch range of -200 ppm at the interface and -1000 ppm at the surface, for an increasing number of laminae: (a) 5 laminae, (b) 10 laminae, (c) 20 laminae and (d) 40 laminae.

continuous and monotonic which has been confirmed by X-ray fluorescence microanalysis on bevelled edges (Halliwell, Juler and Norman, 1983). Usually the lattice parameter increases towards the surface due to the layer becoming increasingly indium rich.

In order to calculate the rocking curve from this type of structure the layer must be divided into a number of laminae of constant composition, as already discussed in Chapter 3. It is required to determine into how many laminae the layer should be divided to produce a rocking curve that is a good approximation to one assuming a continuous variation. Clearly, in order to reduce the computation time to a minimum we require the least number that will produce an adequate approximation.

Fig. 6.2 shows a series of rocking curves calculated for an increasing number of laminae. The GaInAs ternary layer is 3μm thick and has a composition range of -2000 ppm at the substrate/layer interface to -1000 ppm at the surface. This corresponds to compositions of $Ga_{49.8}In_{50.2}As$ at the interface and $Ga_{48.4}In_{51.6}As$ at the surface. Rocking curves are shown for 5, 10, 20 and 40 laminae assuming a linear variation in composition with depth. For the 5 laminae curve the number of peaks is equal to the number of laminae with the individual peaks well resolved. However, for the 10 laminae curve already the number of peaks, although increased, no longer equals the number of laminae. A wedge-like peak shape has developed with only small

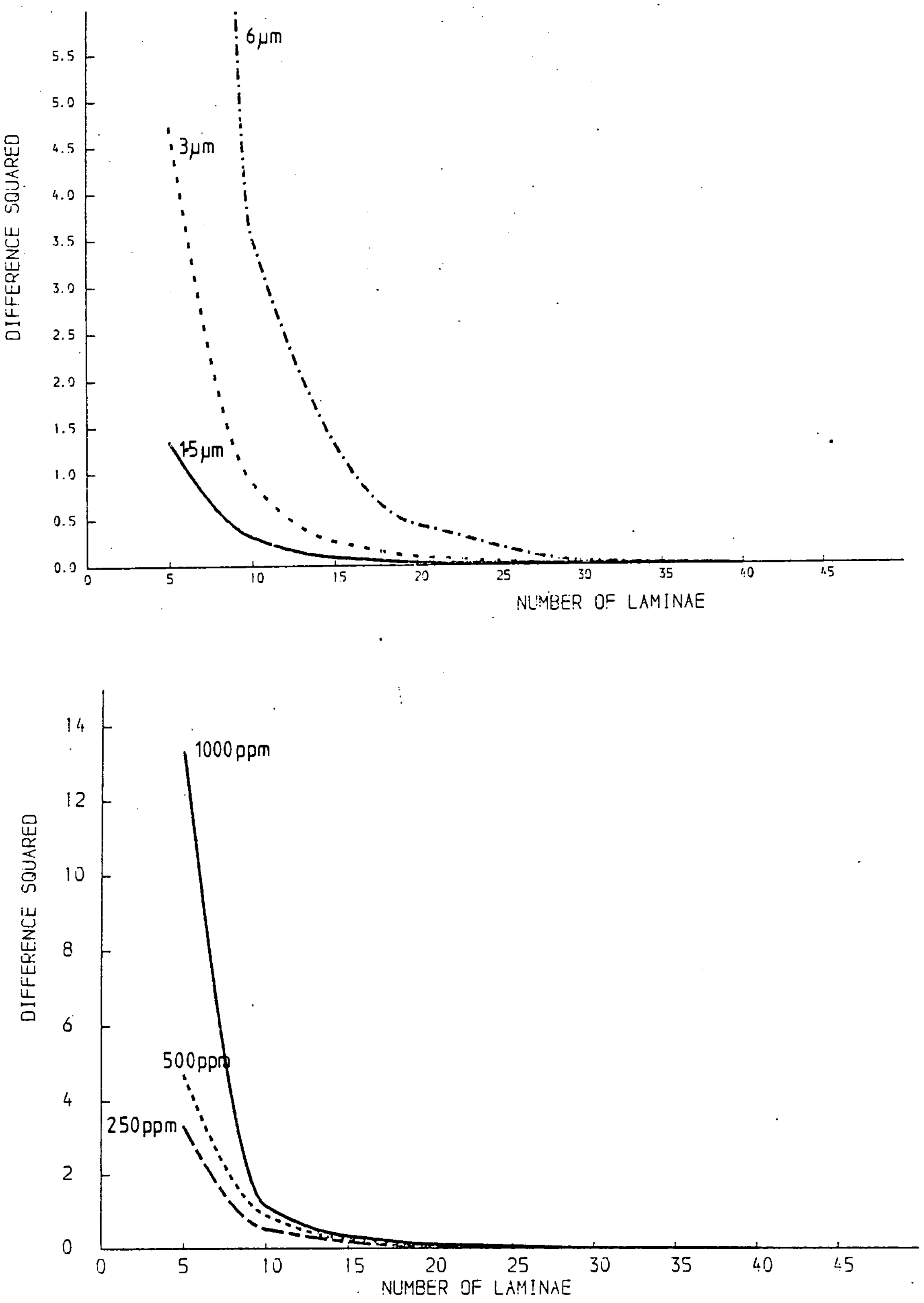


Fig. 6.3 Curves showing the square of the difference between computed rocking curves for an increasing number of laminae and the 40 laminae curve. Curves are shown for a number of mismatch ranges and layer thicknesses.

undulations being visible along the top of the peak. For 20 laminae the peak has fully developed to its final shape and there is no difference between this curve and the one for 40 laminae. This suggests that the peak shape rapidly converges towards that assuming a continuous variation. We note that the number of peaks in the rocking curve cannot be taken as an indication of the number of laminae in the layer.

In order to determine a suitable rule to allow the minimum number of laminae required to give an adequate approximation for a given layer thickness and composition range, curves were calculated for a number of different layer parameters. Assuming that 40 laminae gives a curve that is very close to that required the differences between this curve and those with fewer laminae were calculated. The squares of the difference in intensity at each point in the curve were calculated and are shown in fig. 6.3. Several mismatch ranges and layer thicknesses were used and these curves are also shown in fig. 6.3. From these curves we can deduce that doubling the layer thickness requires double the number of laminae but changing the mismatch range is not as important. As a general rule, for mismatch ranges of the order of 1000 ppm a laminae thickness of 0.1 μm should be perfectly adequate.

Using this approach of calculating rocking curves the curve shown in fig. 6.1 was computed as the best fit to the experimental curve, assuming a linear variation in composition. The effect of sample curvature has been included so as to equate the half width of the central substrate peak. The composition variation used is shown in the inset in fig. 6.1. Agreement between the calculated and experimental curves is not particularly good, the calculated curve has considerably less intensity on the low angle side of the layer peak, yet has rather more intensity on the high angle side. Fine structure present in the calculated curve is somewhat reduced after including sample curvature but is still more pronounced than in the experimental curve. This discrepancy can be accounted for by the the local lattice deviations introduced by the presence of a large number of threading dislocations ($\sim 10^{7}$ $m^{-2}$) known to exist within the layer and substrate. The presence of these dislocations was determined by a TEM study at British Telecom Research Laboratories; the absence of any mismatch dislocations was also revealed. Consequently, there must be very little parallel mismatch and the layer/substrate interface can be considered coherent.

Since there is no fundamental reason for the lattice parameter to vary linearly with depth we must be able to calculate the rocking curve for any arbitrary non linear variation. Often the composition will vary rapidly near the interface, ie at the start of growth, when equilibrium conditions have not yet been reached. Using such a variable non linear change in composition the curve shown in fig. 6.1 was calculated. The mismatch variation which gave this best

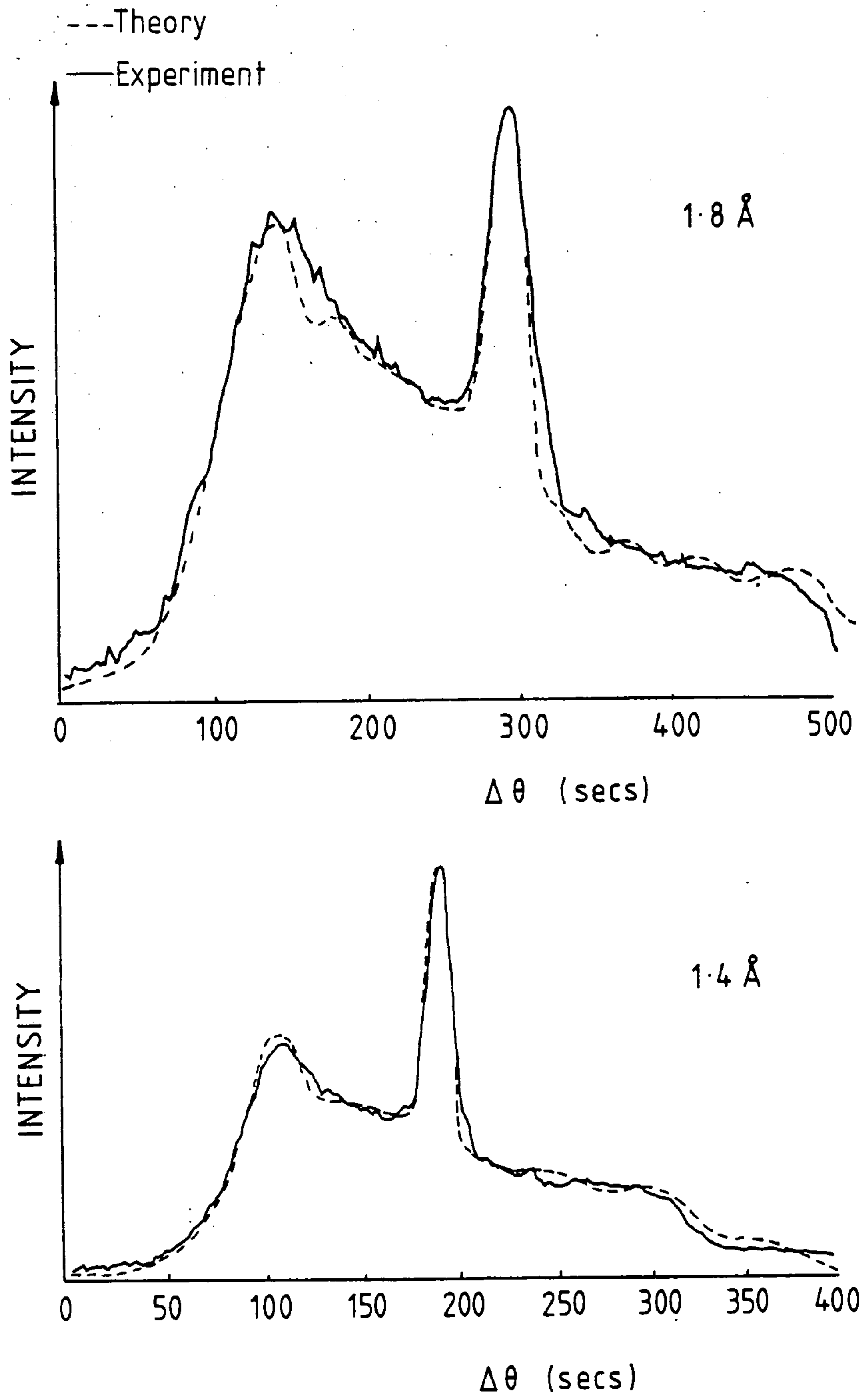
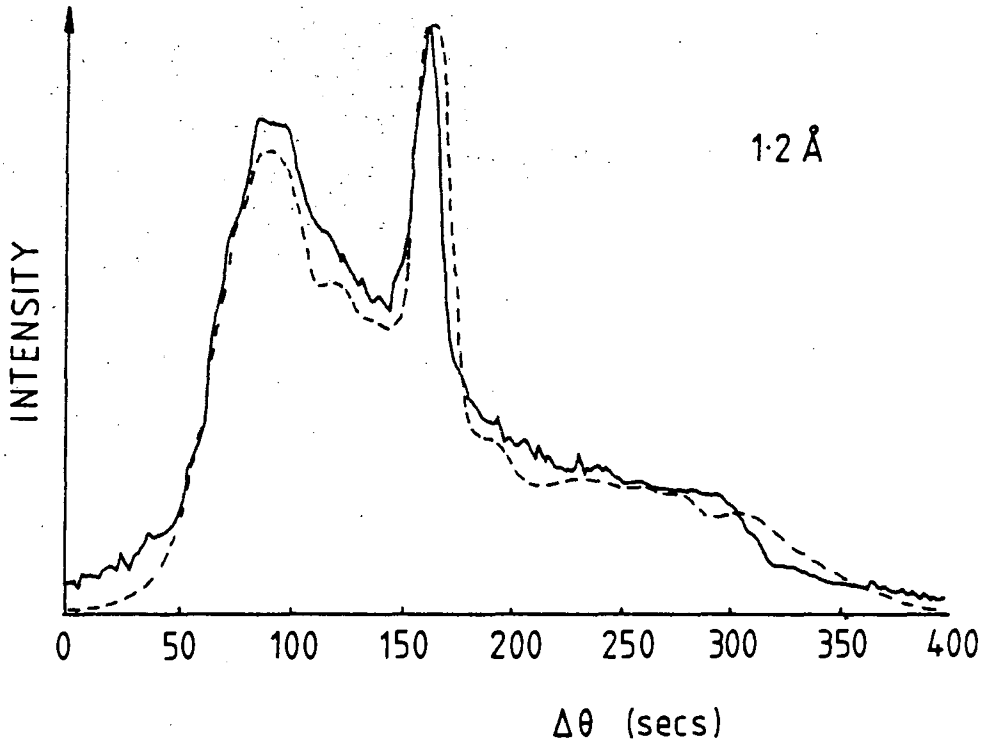
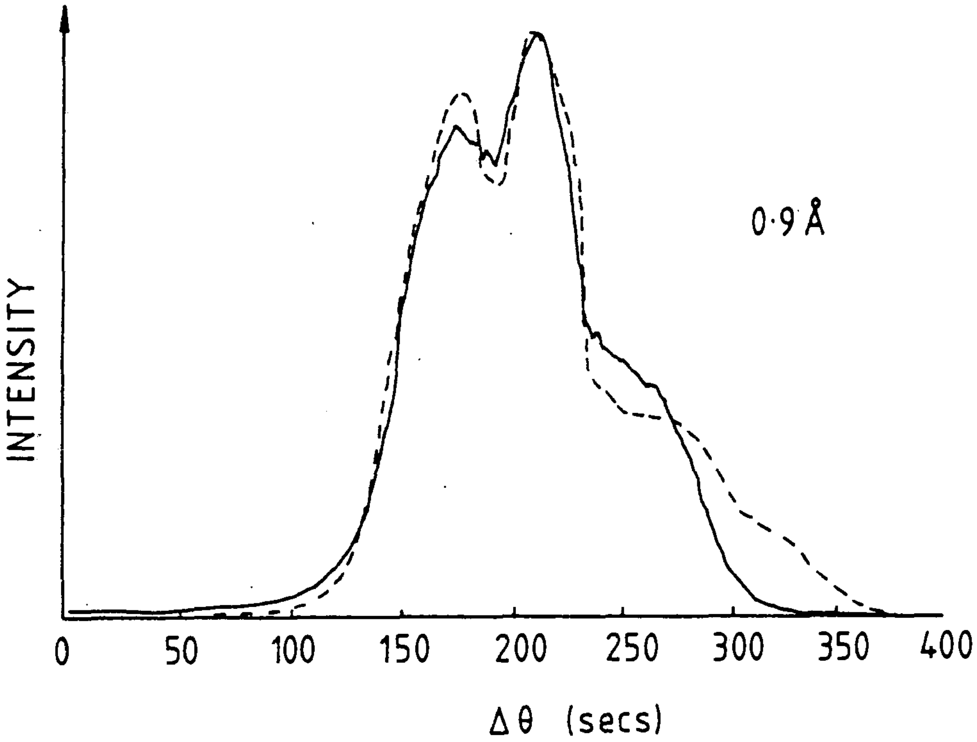


Fig. 6.4 Experimental and computed rocking curves from the sample used in fig. 6.1 at wavelengths of 1.0Å, 1.2Å, 1.6Å and 1.8Å. The non linear variation given in the inset in fig. 6.1 was used for the computed curves.


INTENSITY
1·2 Å
0
50
100
150
200
250
300
350
400
Δθ (secs)


INTENSITY
0·9 Å
0
50
100
150
200
250
300
350
400
Δθ (secs)

fit curve is shown in the inset in fig. 6.1, showing that the composition changes more rapidly towards the interface. The agreement between this curve and the experimental curve is extremely good with the only discrepancies occurring on the high angle side of the layer peak. Here rather more fine structure is present in the computed curve. However, the central substrate peak and the low angle side of the layer peak do match very well. The total layer thickness determined is 2.3 µm which agrees well with that expected from the growth conditions.

Since only one reflection at one wavelength has been used to determine the composition variation there must be some doubt as to its uniqueness. In other words, would the variation deduced be different if another set of X-ray optical conditions had been used. Using the double crystal camera at the SRS, the wavelength can easily be altered and the rocking curve recorded as a function of wavelength. If the black-box, double reflecting, monochromator is used the vertical displacement of the beam alters very little making this experiment very straightforward, especially as all the relevant axes, including the detector arm, are motorised. Such rocking curves were recorded over the range 0.8 Å to 2.0 Å at intervals of 0.2 Å and a selection of these are shown in fig. 6.4. As the wavelength is reduced the overall width of the rocking curve decreases, as expected from the differential of the Bragg equation. The half width of the central substrate peak would be expected to decrease with decreasing wavelength since, for a flat crystal it is directly proportional to the wavelength, but is, instead, observed to increase for the curves below 1 Å. This effect will be due to the increased effect of bulk sample curvature as the Bragg angle is decreased. When finding the reflection with the TV detector the peak can be observed to sweep across the sample due to its curvature and this displacement as a function of the rotation angle can be used to estimate the curvature. Fine detail present in the rocking curves at the longer wavelengths also becomes increasingly blurred out at the shorter wavelengths due to this effect. The ratio of integrated intensity under the layer and substrate peaks changes as a function of both the absorption coefficient and the X-ray path length within the layer, giving a maximum corresponding to the Ga absorption edge. Values of this ratio as a function of wavelength are given in Table 6.1.

Using the composition variation determined from the rocking curve at 1.54 Å computed curves were calculated at each wavelength and are also shown in fig. 6.4. Agreement between these curves and the experimental curves is reasonably good, with better matching obtained at the longer wavelengths. At the shorter wavelengths the same sized X-ray beam will intercept a larger area of the sample making the rocking curve more susceptible to area variations in both thickness and composition. However, for all the wavelengths used better fits could not be obtained by

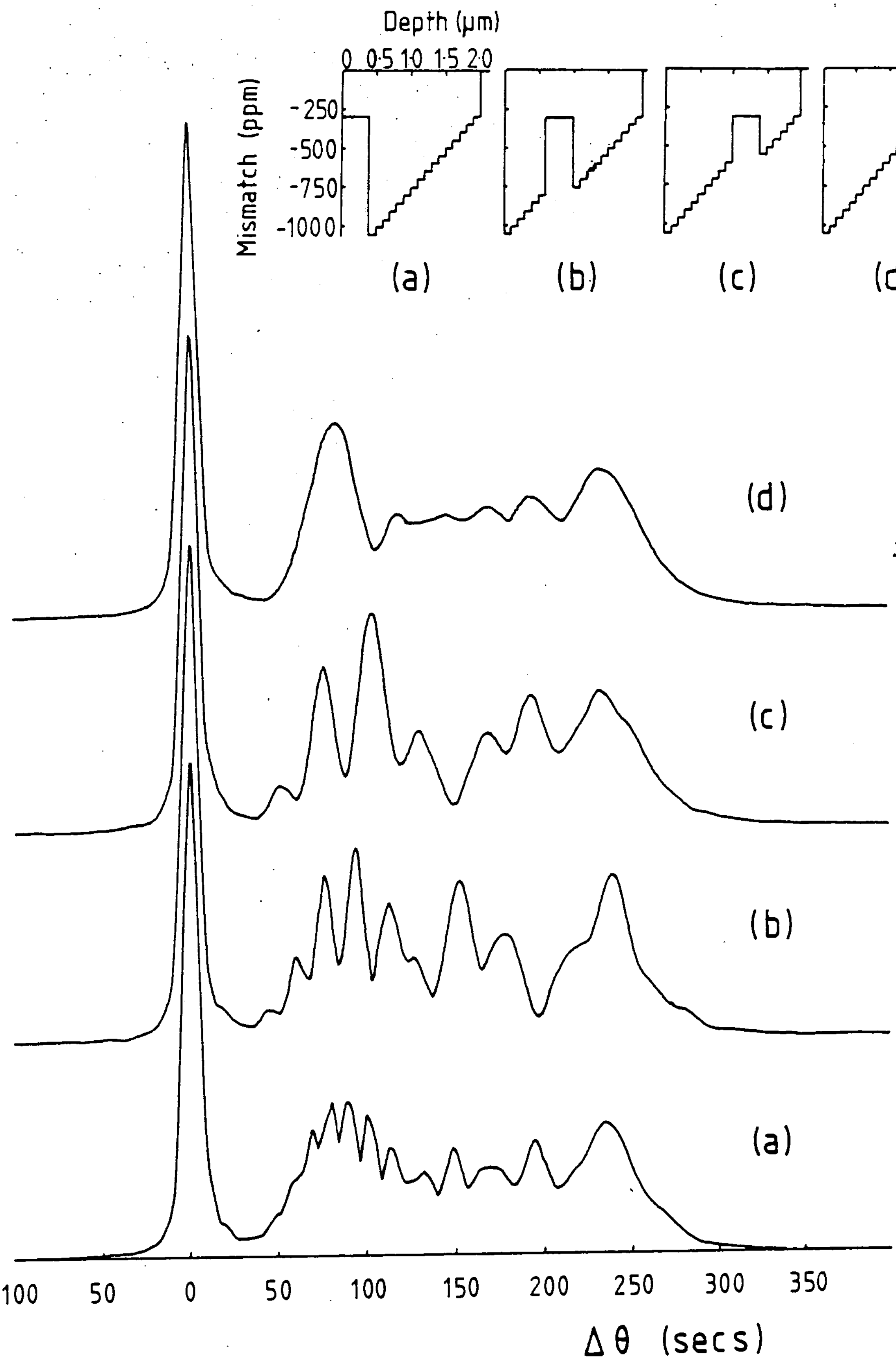


Fig. 6.5 Theoretical 004 rocking curves for a GaInAs graded layer with a 0.4 μm region of -300 ppm mismatch at different depths within the layer. The total layer thickness is 2.0 μm and the composition variations used are shown in the insets. The wavelength used is 1.5Å.

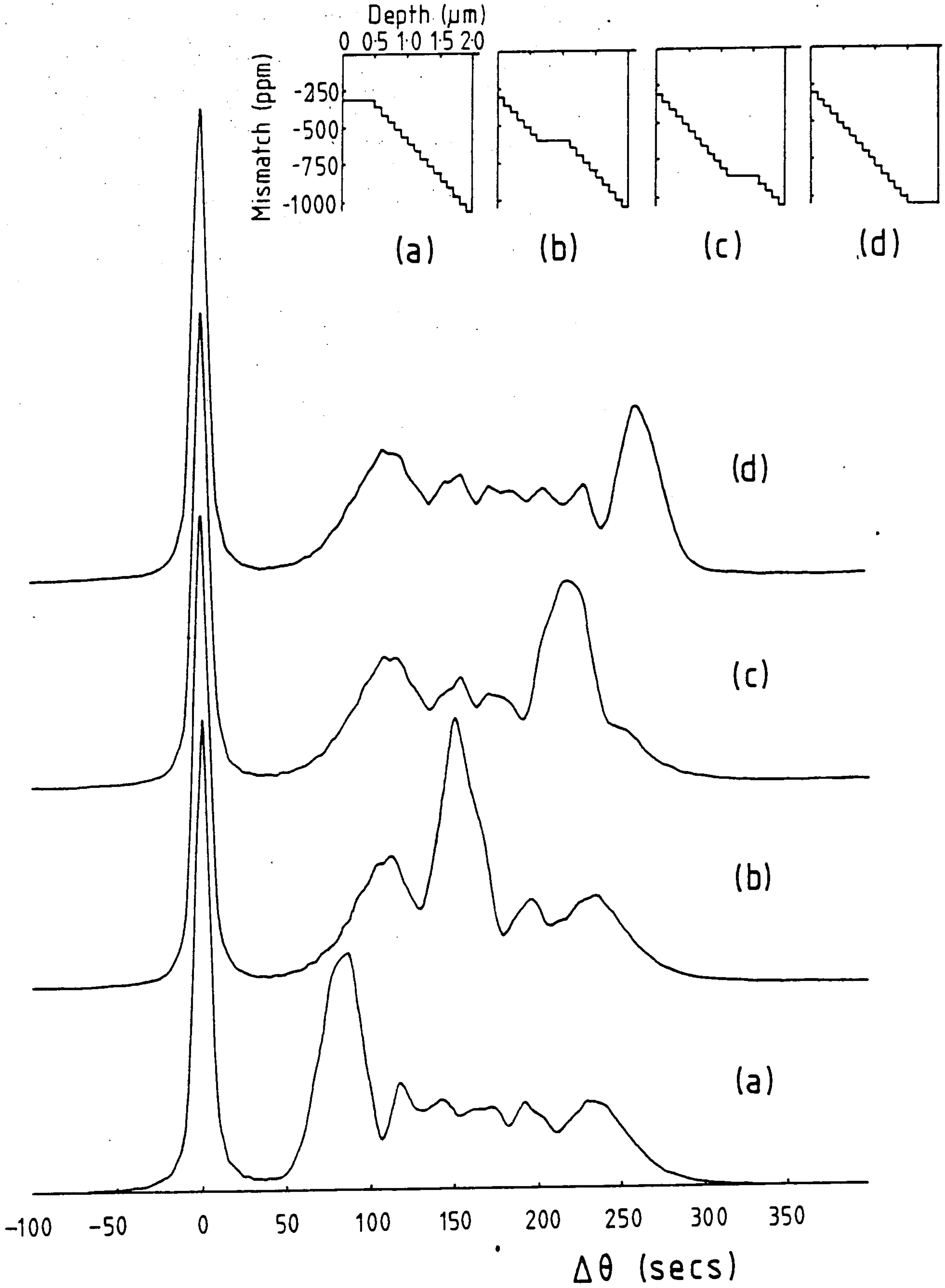


Fig. 6.6 Theoretical 004 rocking curves for a GaInAs graded layers with a 0.4 μm region of constant mismatch at different depths, as shown in the insets. The total layer thickness and overall mismatch range is the same as that in fig. 6.5. The wavelength used is 1.5Å.

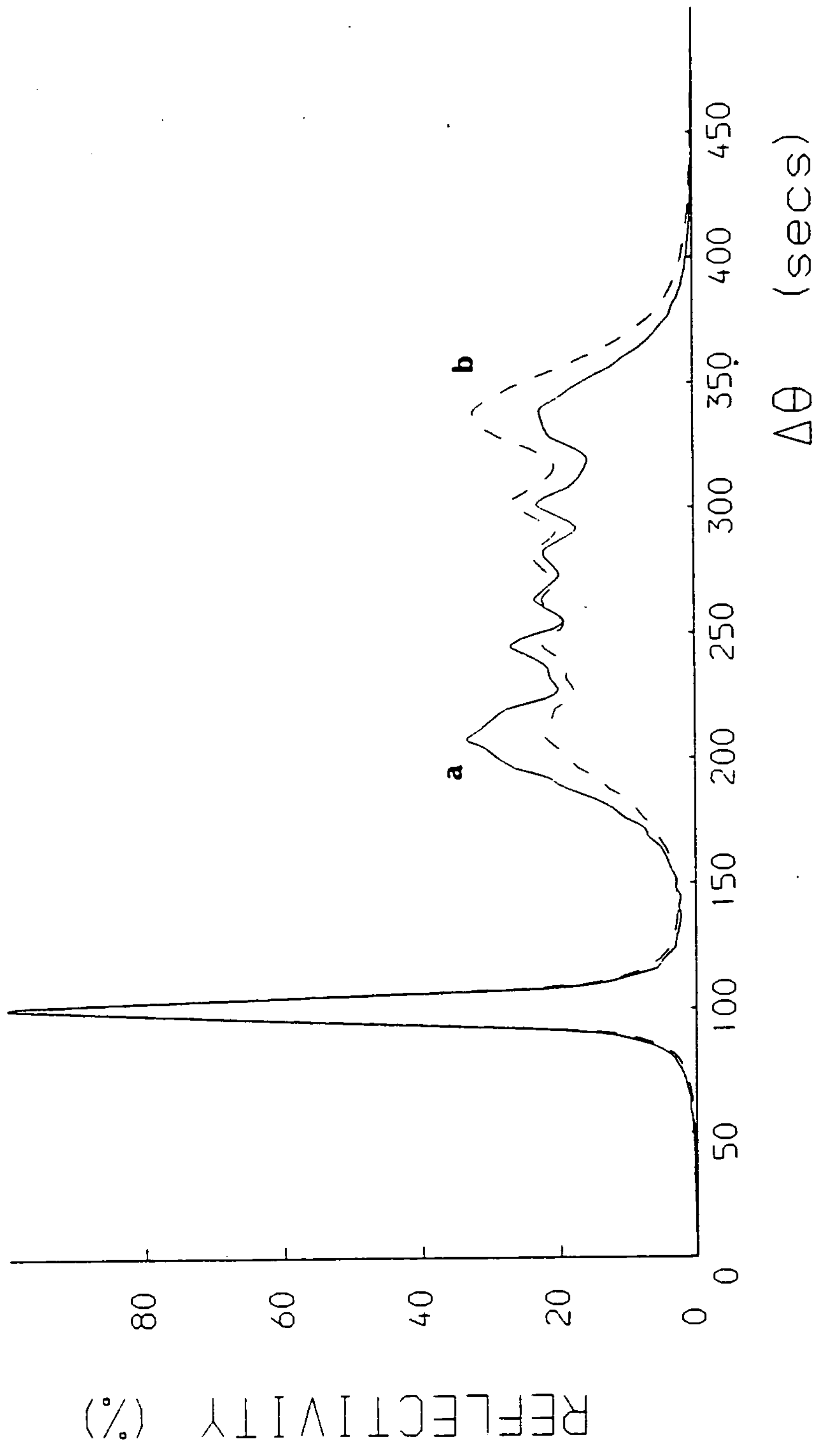


Fig. 6.7 Theoretical 004 rocking curves for a 2.0 μm thick GaInAs graded layer with a mismatch range of -1050 ppm to -300 ppm. The two curves are for the composition variation reversed, i.e. (a) is for -300 ppm at the surface and -1050 ppm at the interface and (b) for -1050 ppm at the surface and -300 ppm at the interface. A linear composition variation is used.

assuming a different composition variation or layer thickness, confirming the use of a single rocking curve to determine these parameters.

TABLE 6.1

**Values for the ratio of the areas under the layer and substrate peaks for the experimental curves for a range of wavelengths.**

| Wavelength<br>(Å)<br>(±0.02) | Ratio of areas under layer<br>and substrate peaks<br>(±0.05) |
|---|---|
| 0.8 | 1.32 |
| 0.9 | 1.53 |
| 1.0 | 2.50 |
| 1.1 | 3.21 |
| 1.2 | 2.91 |
| 1.3 | 2.21 |
| 1.4 | 1.77 |
| 1.5 | 1.85 |
| 1.8 | 2.36 |

Polarisation experiments similar to those performed with a single uniform layer were also carried out. Even with a highly perfect InP crystal as the first crystal no effects could be observed that could be ascribed to the polarisation change, although significant modifications were predicted theoretically.

The rocking curve is extremely sensitive to the composition variation used as is demonstrated by the rocking curves shown in figs. 6.5, 6.6 and 6.7. In fig. 6.5 a region of constant composition, 0.4 μm thick and -300 ppm mismatched, has been positioned at different depths in a linearly graded layer in which the mismatch varies from -300 ppm at the interface to -1050 ppm at the surface. As can be seen, the shape of the rocking curve is extremely sensitive to the position of the constant region, while the wedge shape with a nearly level top is no longer produced. Consequently, it is obvious that a smooth variation in lattice parameter does not exist in the layer. Further, if such a region was known to exist within the layer its position in the layer could be determined. It would be expected that increasing the depth of the constant layer would reduce its effect on the rocking curve just from a consideration of absorption, however, even increasing its thickness to compensate for this effect will not restore the original shape due to the phases of the beams from each laminae also being affected. Fig. 6.6 shows the effect of including a region where the composition remains constant at different depths in a graded layer. The mismatch of the layer varies from -1050 ppm mismatch at the interface and -300 ppm at the surface. Again the layer peak is particularly sensitive to the position of the constant region and the differences when this region is next to the surface or next to the interface are very marked. Further, fig. 6.7 shows the effect on the rocking curve on having the linear grading reversed. The solid curve shows the rocking curve when the mismatch at the interface is -1050 ppm and at the surface is -300 ppm while the broken curve is for -300 ppm at the interface and -1050 ppm at the surface. The curves are completely different in shape and, consequently, from the rocking curves there can be no doubt as to the sign

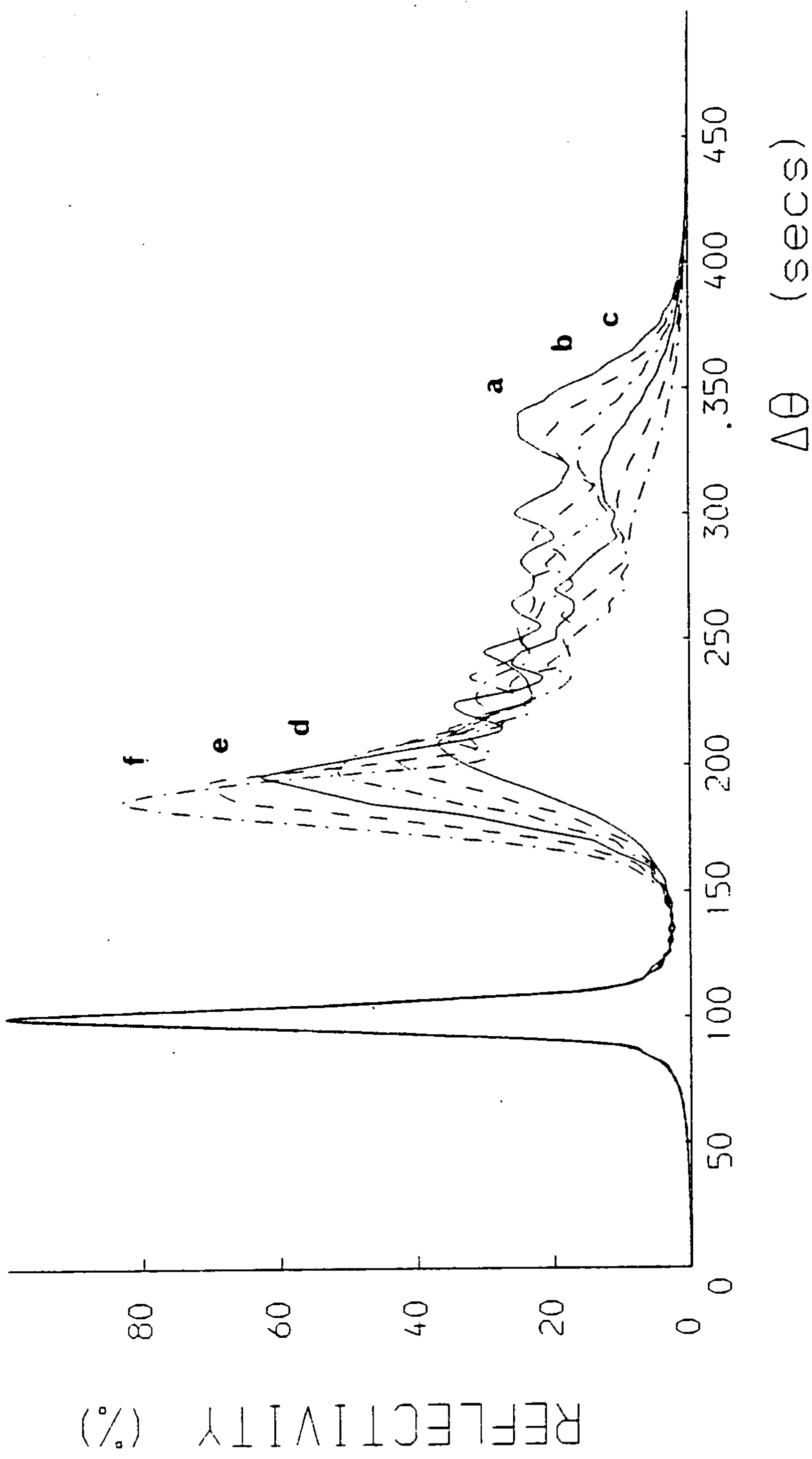


Fig. 6.8 Theoretical rocking curves for the same composition range as in fig. 6.7 with increasing non linearity, giving a greater range of change at the interface. The curves are for (a) linear (b) 20% (c) 40% (d) 60% (e) 80% and (f) 100% non linearity. 100% non linearity corresponds to an elliptical variation which is perpendicular at the intersection with the axes.

of the composition gradient. Some degree of mirror symmetry does exist between the two curves, as might be expected. However, differences do exist between one curve and the mirror image of the other.

Additionally, fig. 6.8 shows the affect of increasing the non linearity of the composition change. The mismatches used are -1050 ppm at the interface and -300 ppm at the surface, for a 2.0 µm layer, and in each consecutive curve the variation is further from linearity. The slope of the top of the wedge-like layer peak is rapidly increased as the rate of change of composition next to the interface is increased and, for the high non linearity curves, the shape approaches that of a single peak with an extended tail on the high angle side. Consequently, if a composition variation can be found that gives a rocking curve that agrees well with the experimental one such a variation can be taken to be a good measure of the actual variation. This same assumption has been extensively used in the characterisation of ion or diffusion implanted layers by rocking curves, as discussed in Chapter 2.

A series of closely related VPE grown GaInAs ternary layers, on (001) InP substrates, have also been studied. In addition to measuring the rocking curves these samples were electrically characterised at British Telecom Research Laboratories. Lang topographs were also recorded to determine the presence of mismatch dislocations that would indicate an incoherent interface.

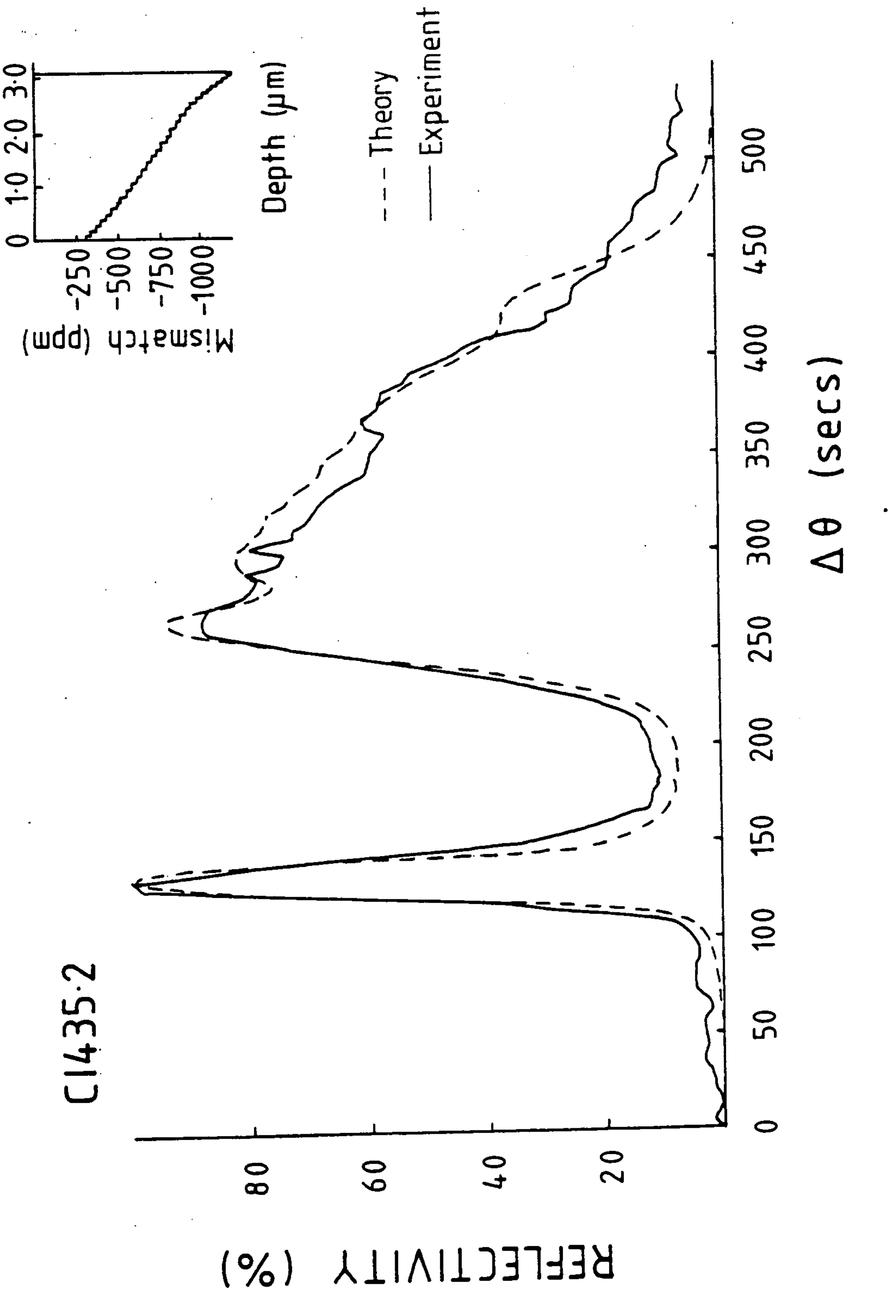
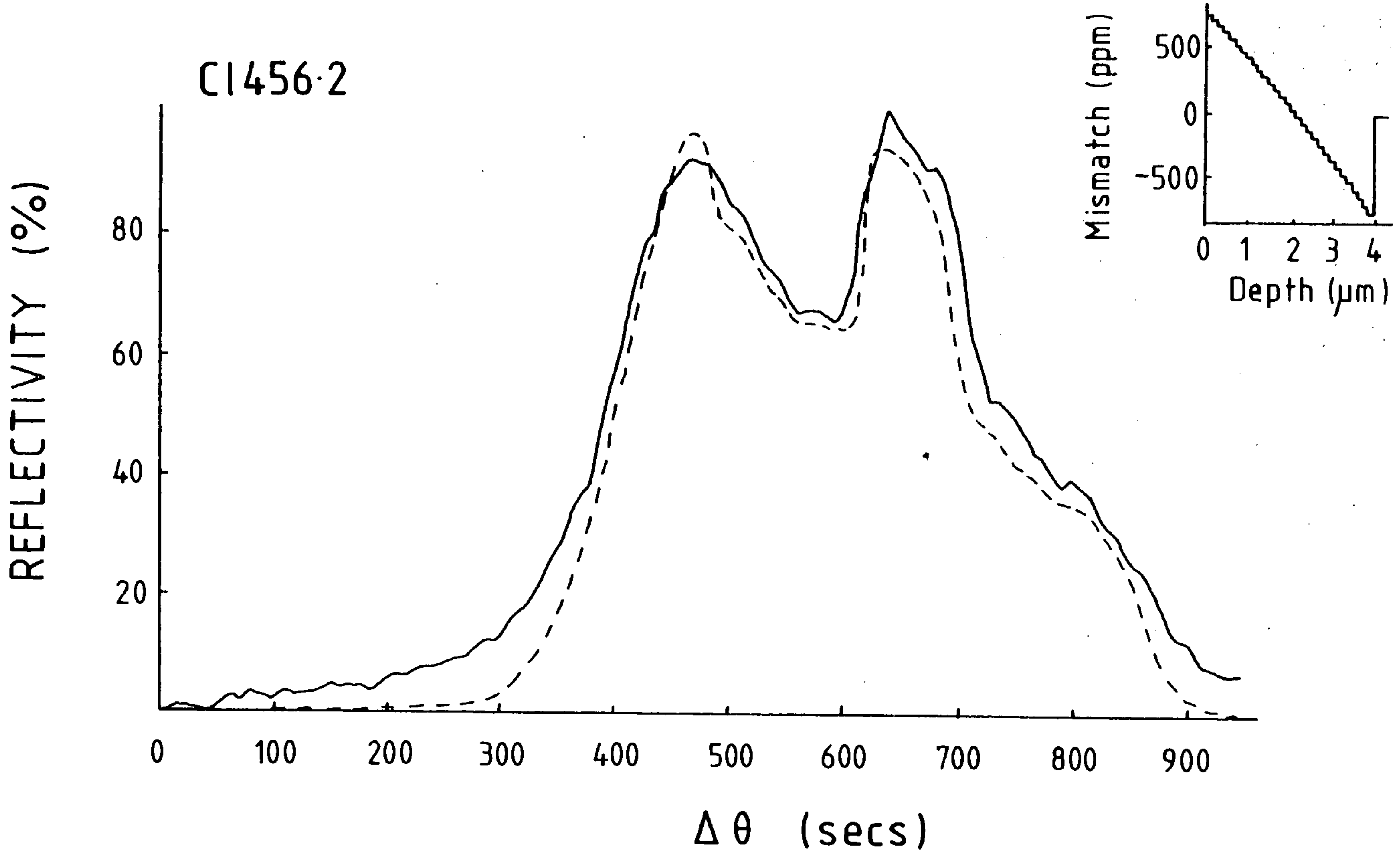
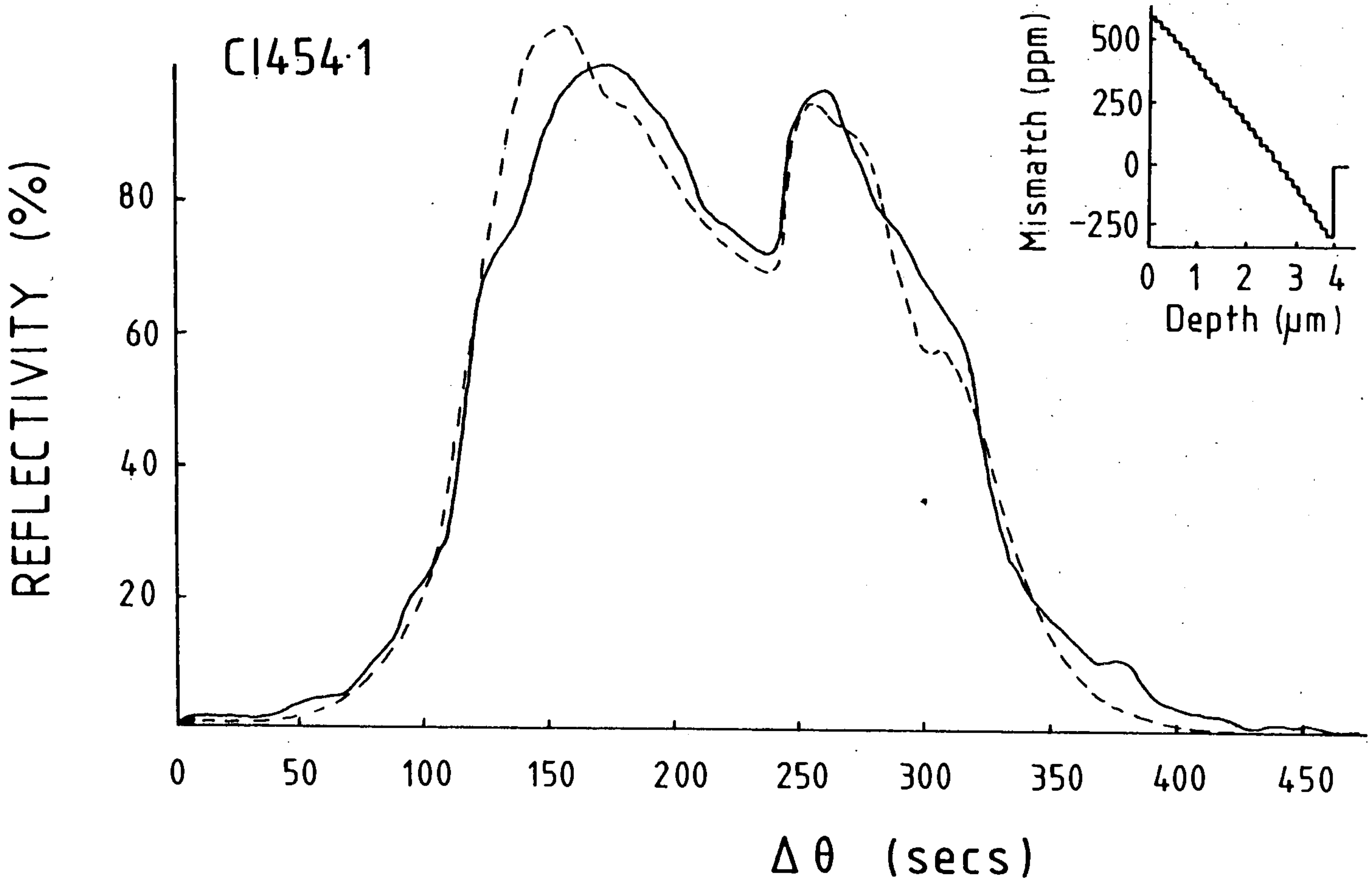
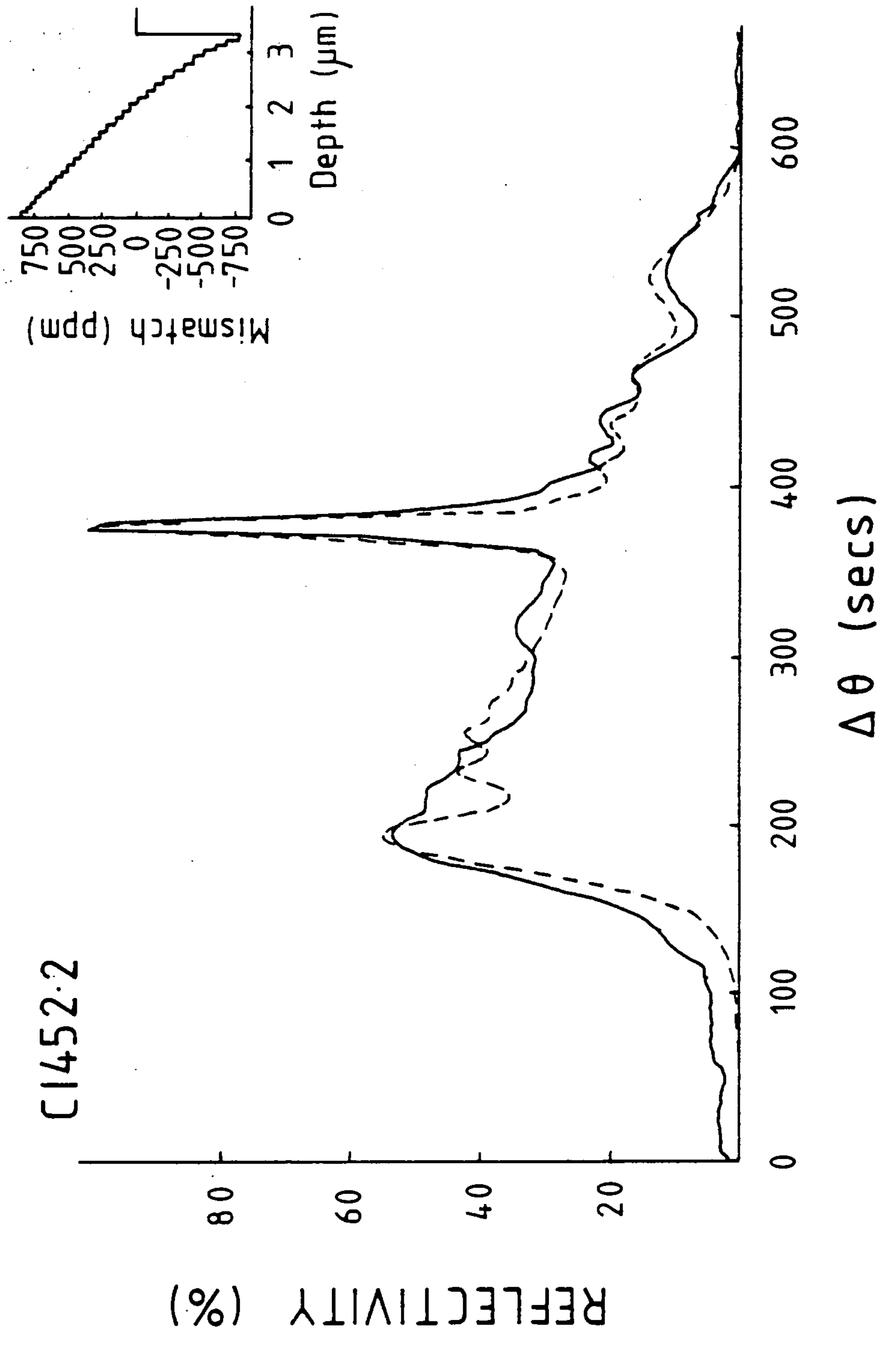


Fig. 6.9 004 experimental and computed rocking curves for a number of GaInAs graded layers, grown by VPE on a (001) InP substrate. The composition variations used to give the computed curves are shown in the insets. All rocking curves were recorded with CuK$\alpha_1$ radiation.

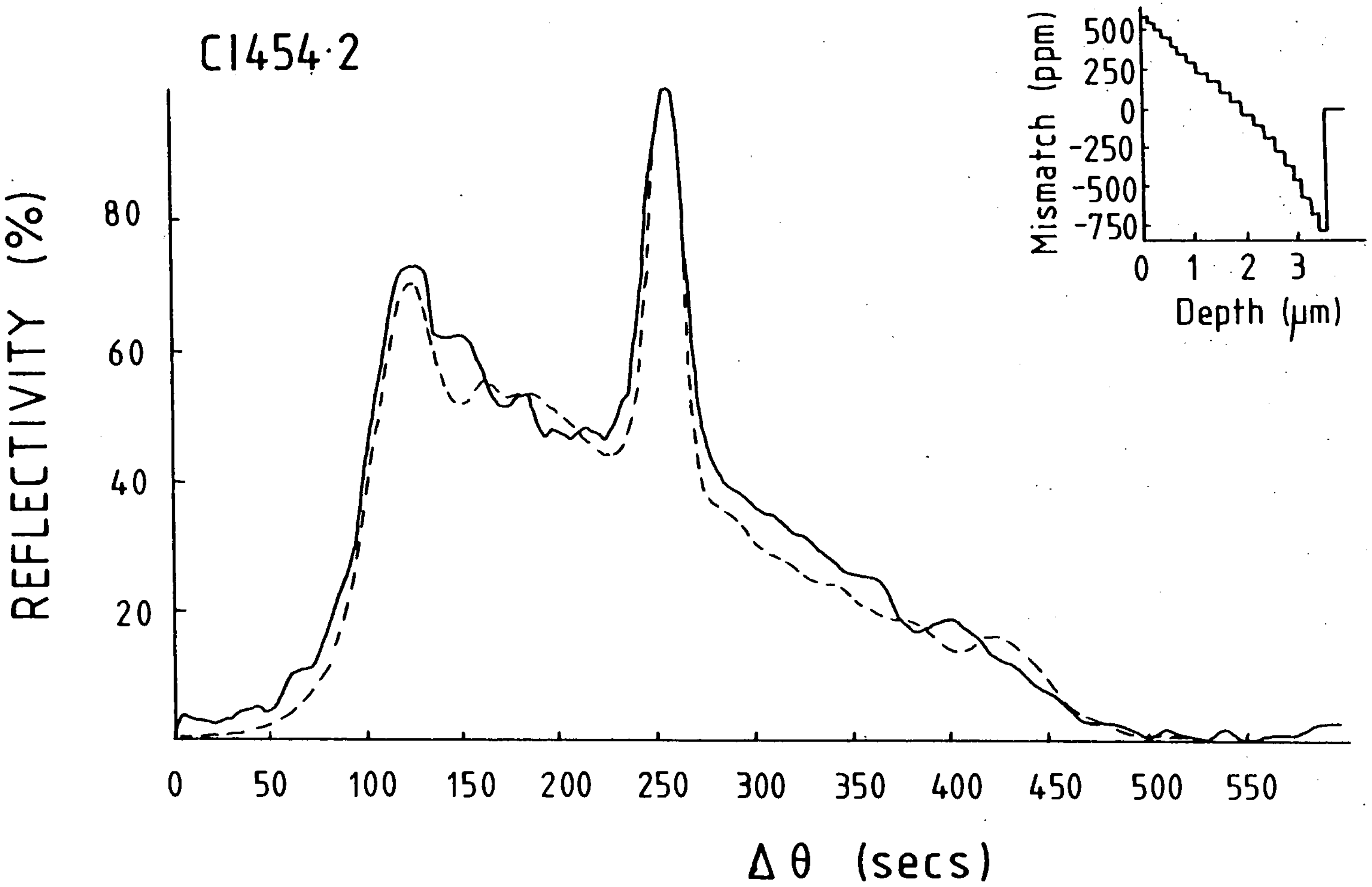


C1456·2
REFLECTIVITY (%)
80
60
40
20
0
100
200
300
400
500
600
700
800
900
Δθ (secs)
Mismatch (ppm)
500
0
-500
0
1
2
3
4
Depth (μm)


C1454·1
REFLECTIVITY (%)
80
60
40
20
0
50
100
150
200
250
300
350
400
450
Δθ (secs)
Mismatch (ppm)
500
250
0
−250
0
1
2
3
4
Depth (μm)



C1452·2
REFLECTIVITY (%)
80
60
40
20
0
100
200
300
400
500
600
Δ θ (secs)
Mismatch (ppm)
750
500
250
0
-250
-500
-750
0
1
2
3
Depth (μm)


C1454·2
REFLECTIVITY (%)
80
60
40
20
0
50
100
150
200
250
300
350
400
450
500
550
Δθ (secs)
Mismatch (ppm)
500
250
0
-250
-500
-750
0
1
2
3
Depth (μm)

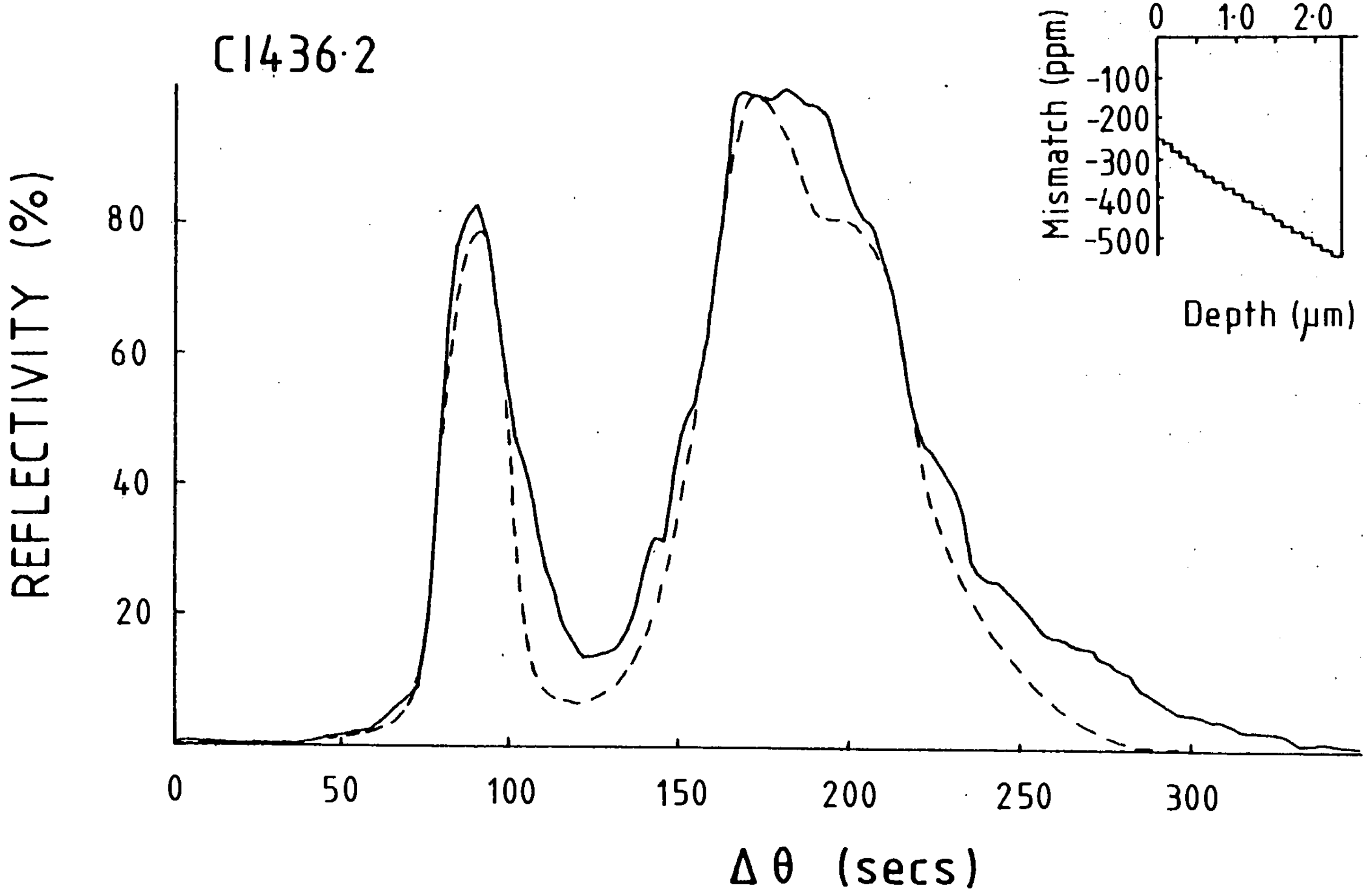

C1436·2
REFLECTIVITY (%)
80
60
40
20
0
50
100
150
200
250
300
Δθ (secs)
Mismatch (ppm)
-100
-200
-300
-400
-500
0
1·0
2·0
Depth (μm)

Fig. 6.9 shows the double crystal 004 rocking curves for these samples, recorded with CuK$\alpha_1$ radiation and an InP first crystal. There is a very wide variation in the shapes of these rocking curves with two samples in particular giving very broad substrate peaks. All of the curves exhibit a very broad layer peak indicative of a graded layer. Computed rocking curves found to best fit these curves are also shown in fig. 6.9 for comparison, the mismatch variations as a function of depth are shown in the inset in each figure.

The rocking curve from sample CI452.2 shows a narrow, central substrate peak close in half width to the theoretical value. Considerable fine detail is present in the layer peak, although not as much as in the computed curve which does agree reasonably well with the experimental curve. This sample does contain mismatch dislocations which will reduce the overall sample curvature, as shown in Chapter 1, consequently reducing the broadening of the substrate peak and the blurring out of the fine detail. The mismatch variation determined shows some deviation from linearity, with an increased rate of change near to the interface. However, due to the presence of the mismatch dislocations the composition range deduced will not be accurate.

The central substrate peak in the rocking curve from sample CI454.2 is again fairly narrow and only slightly broader than the theoretical value. This sample does not contain any mismatch dislocations which will reduce the sample curvature. However, since the composition range gives lattice parameters both larger and smaller than that of the substrate the curvature will be reduced as the two regions will produce curvature in the opposite sense. The computed curve does fairly well with the experimental curve, again giving a composition variation that changes more rapidly towards the interface. Fine detail is present in both curves due to the reduced effect of sample curvature.

In the rocking curves from both CI436.2 and CI435.2 the substrate and layer peaks are well separated, with the lattice parameter of the layer always smaller than that of the substrate. The curve from sample CI436.2 is rather unusual in shape, with a flat top on the low angle side followed by a steeply sloping tail on the high angle side. The calculated curve does not agree particularly well with the experimental one, with the flat topped region proving difficult to simulate. Also the extended high angle tail is less intense than in the experimental curve which would suggest that the layer thickness is larger than that determined by matching the heights of the maximum of the layer peak and the substrate peak. The composition variation determined is close to linear. The layer peak from sample CI435.2 is again rather unusually shaped. It is close to a wedge shape but appears to have some extra intensity in the centre and has a steeper slope on the high angle side of this 'bulge'. This would suggest that two different rates of change of composition are present and the computed curve, which agrees fairly well with the experimental one, does confirm this. The composition variation determined can almost be divided into two regions of linear variation, with a more rapidly varying section next to the interface. This sample contains a large number of threading dislocations hindering the determination of the presence of mismatch dislocations.

Both the rocking curves from samples CI456.2 and CI454.1 show very broadened central substrate peaks and no fine detail is visible in either curve. Sample CI454.1 contains a large number of mismatch dislocations and sample CI456.2 contains so many threading dislocations that the presence of mismatch dislocations could not be determined. Consequently, we would expect fine detail in both curves to be blurred out with the broadened substrate peaks due to a small radius of curvature. Computed curves that agree fairly well with these experimental curves could be found after including this sample curvature, with both requiring a nearly linear composition variation with slightly greater rates of change near to the interface.

A summary of both the X-ray and electrical results is shown in Table 6.2. There does appear to be some correlation between the presence of mismatch dislocations and the dark current. The composition ranges given in the table are those determined from the rocking curves. Generally, the layer thicknesses found from the simulated rocking curves were found to agree fairly well with those obtained from the electrical measurements. For all the samples composition variations close to linear were found to give computed rocking curves that agree reasonably well with the experimental ones, with many requiring a greater rate of change near to the interface, ie at the start of growth.

These results confirm that the computer simulation of rocking curves, as a best fit to the experimental ones, is an extremely valuable technique for the determination of the composition variation with depth for graded layers. No other technique can match the sensitivity to the composition achieved with this X-ray method. However, due to this high sensitivity and, additionally, sample curvature and defect density, it is not straightforward to achieve good agreement between the computed and experimental rocking curves as there are many parameters to alter.

TABLE 6.2

SUMMARY OF X-RAY AND ELECTRICAL CHARACTERISATION OF VPE GROWN SAMPLES

| SAMPLE | CI435.2 | CI454.1 | CI454.2 | CI436.2 | CI452.2 | CI456.2 |
|---|---|---|---|---|---|---|
| Mismatch Network | ?[1] | yes | no | no | yes | ?[1] |
| Electrical Comment[2] | Quite good | Graded from 2 µm | well behaved | pretty good | v. graded from 1 µm | v.poor |
| Dark Current[2] (mA/cm$^2$) | 0.1 | 2.0 | 0.17 | 0.06 | 0.4 | >2 |
| Carrier Conc.[2] ($10^{16}$) | 3 | 1 | 2 | 4 | 2 | 3-84 |
| Mismatch at surface (ppm) | -300 | 650 | 610 | -250 | 850 | 750 |
| Mismatch at interface (ppm) | -1250 | -300 | -800 | -550 | -780 | -800 |
| Indium content range (%) | 52.7 - 51.3 | 54.0 - 52.7 | 54.0 - 51.9 | 52.7 - 52.3 | 54.3 - 52.0 | 54.2 - 51.9 |
| Layer Thickness (X-ray) | 3.15 | 3.8 | 3.8 | 2.35 | 3.4 | 4.0 |
| Layer Thickness[2] (Elec.) | 4.6 | ∼4 | ∼4 | 2.8 | ∼4 | ∼4 |

[1] The ? indicates that the presence of mismatch dislocations could not be determined due to the high density of threading dislocations.
[2] These results were provided by British Telecom Research Laboratories.

# CHAPTER 7

# MULTIPLE AND MULTILAYER STRUCTURES

Multiple layer structures, especially multiquantum well structures, are probably one of the most important types of structures for device applications with the majority of devices being developed consisting of more than one layer. Clearly, there is little difference in calculating the complete rocking curve for these structures from calculating the curves for graded layers. Here the layers used in the calculation are the actual physical layers existing in the structure.

Even for a small number of layers great care needs to be taken in the interpretation of the rocking curves. The number of peaks observed will very rarely reflect the number of layers present, with both greater or smaller numbers of peaks than layers possible. This is illustrated by the 004 theorectical curves shown in fig. 7.1, for an increasing number of subdivisions of a GaInAs layer into layers of alternating composition, keeping the total layer thickness and the average mismatch constant. The radiation used is CuK$\alpha_1$, with an InP first crystal. The two layer mismatches used are -500 ppm and -1000 ppm, corresponding to compositions of $Ga_{47.6}In_{52.4}As$ and $Ga_{48.4}In_{51.6}As$. For curve (a), consisting of one pair of such layers each of thickness 0.5 µm, two peaks are formed as expected. In addition, low intensity Bragg case Pendellosung oscillations

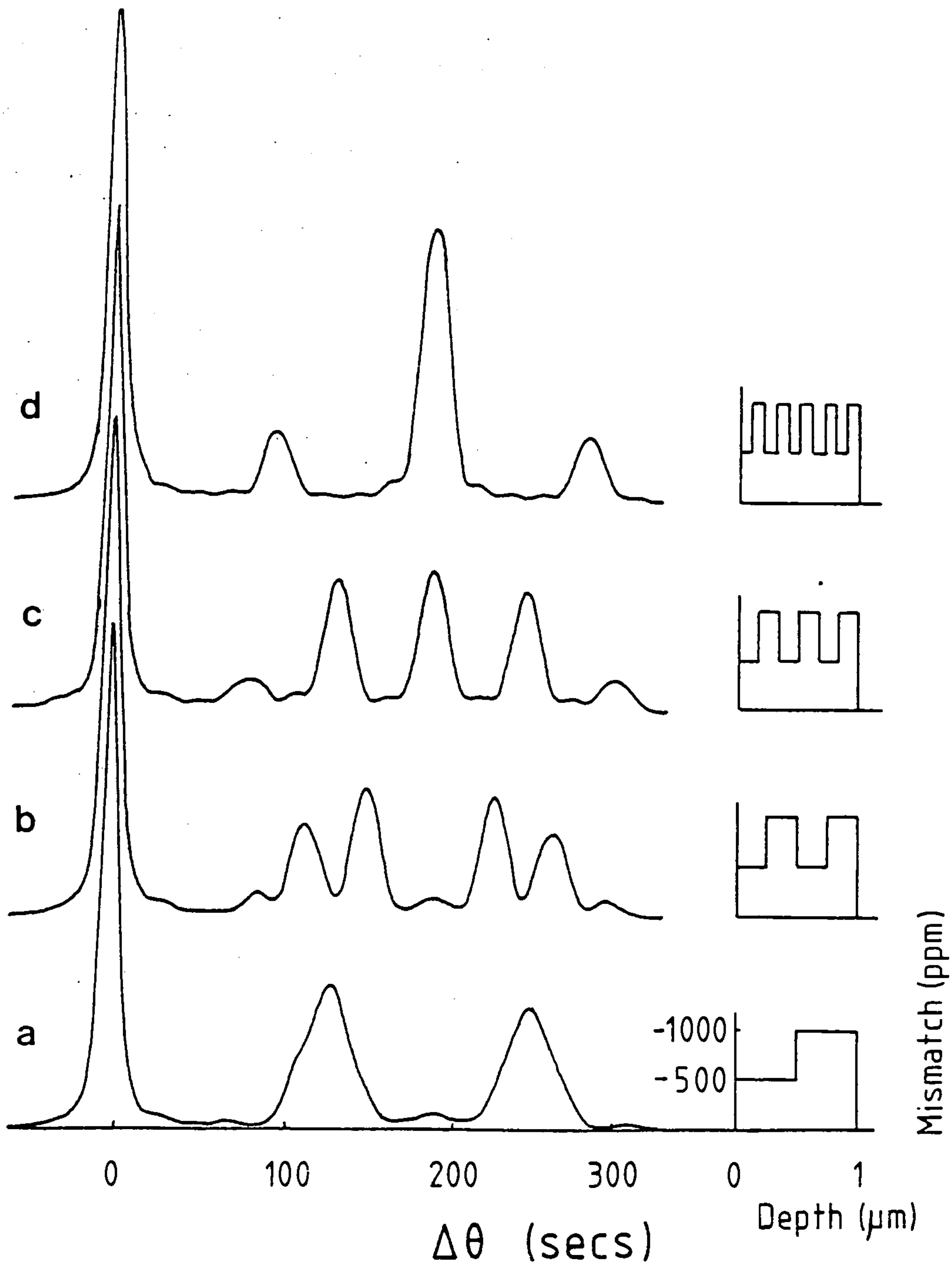


Fig. 7.1 004 theoretical curves with CuK$\alpha_1$ radiation for an increasing number of subdivisions of a 1 μm GaInAs layer into alternating layers of -500 ppm and -1000 ppm mismatch. The curves are for (a) 2 layers, (b) 4 layers, (c) 6 layers and (d) 10 layers. The mismatch variations are shown in the inset.

are visible. For the second curve, (b), for a sample consisting of two such pairs of layers, with individual layer thicknesses of 0.25 µm, there are now four peaks visible in the rocking curve. This is not expected as only two compositions are present in the layer and arises because of the phase differences between the beams from each layer. The angular position of the peaks are different from those in curve (a), the peaks in each pair being on either side of the two peaks in curve(a). This suggests that some degree of destructive interference has occurred at the Bragg angle corresponding to the composition of the two layers. Again weak oscillations are present between and either side of the two pairs of peaks.

As the number of subdivisions is increased to three pairs, as in fig. 7.1(c), the peaks produced are again greatly different from those in the preceding curves. Now three main layer peaks are produced, the central peak in a position exactly midway between the peaks of curve (a). The outer two peaks correspond to the peaks in curve (a), although they are somewhat narrower. Also present in this curve are the beginnings of two satellite peaks either side of the three main peaks.

When the layer has been divided into 5 pairs, as in curve (d), the central peak has developed to be clearly the most intense of all the peaks from the layer. Two satellite peaks on either side of the main peak, equispaced from it, are clearly visible. Their separation is determined by the thickness of the repeat unit of the pairs of layers within the layer. As the number of divisions are further increased this same pattern of peaks is continued, ie a central peak and satellite peaks. This is the predicted behaviour of a superlattice, as described in Chapter 2. The central peak is referred to as the zeroth order peak and its displacement from the substrate peak gives the average composition of the layer. The two satellite peaks seen in curve (d) are referred to as the +1 and -1 peaks, with positive indices being on the high angle side of the zeroth order peak. Higher order, weaker satellite peaks, again equispaced from the zeroth order peak, will also be produced.

If the superlattice period is considered as a lattice spacing, the peaks correspond to different orders of reflection from this lattice at the wavelength used. Clearly, the satellite peaks will be symmetrically spaced about the zeroth order peak, not from the substrate peak. Therefore, their separation can be used to determine the superlattice period. As more subdivisions are made, more higher order satellite peaks will become visible. The relative intensities of the satellites will depend on the particular structure of the superlattice. Additionally, since superlattices can be constructed of any combination of binary, ternary, quaternary or higher alloys, its average composition will vary not only as a function of the relative thicknesses of the layers but also as a function of the composition of the individual layers. For example, for a

$n_1$-GaAs/$n_2$-AlAs superlattice, where $n_i$ is the number of monolayers in each layer, the separation of the substrate and zeroth order peaks will depend only on the ratio $n_1:n_2$, but for a $n_1$-$Ga_{1-x}Al_xAs$/$n_2$-GaAs superlattice it will also depend on the value of x.

Since the individual layers forming the superlattice are usually very thin compared to the extinction distance, the kinematical theory of X-ray diffraction has been extensively used to calculate the satellite intensities, see Chapter 2. However, many of the superlattices have relatively thick buffer layers grown either between the substrate and superlattice or as a cap on top of the superlattice or even in both positions. If this buffer is of a similar composition to that of one of the components of the superlattice the position and/or intensity of the zeroth order peak will be affected. Consequently, an approach based on the dynamical diffraction theory is required and has been used throughout the calculations presented here. For a perfectly periodic superlattice, in which all of the repeat periods are identical, the diffracted intensity calculation can be simplified according to the scheme of Vardanyan, Manoukyan and Petrosyan (1985) and the computational approach is no longer required. However, if imperfect superlattices are to be considered the computational approach must be used.

If the growth of a superlattice is considered, in which the interfaces between the individual layers are perfectly

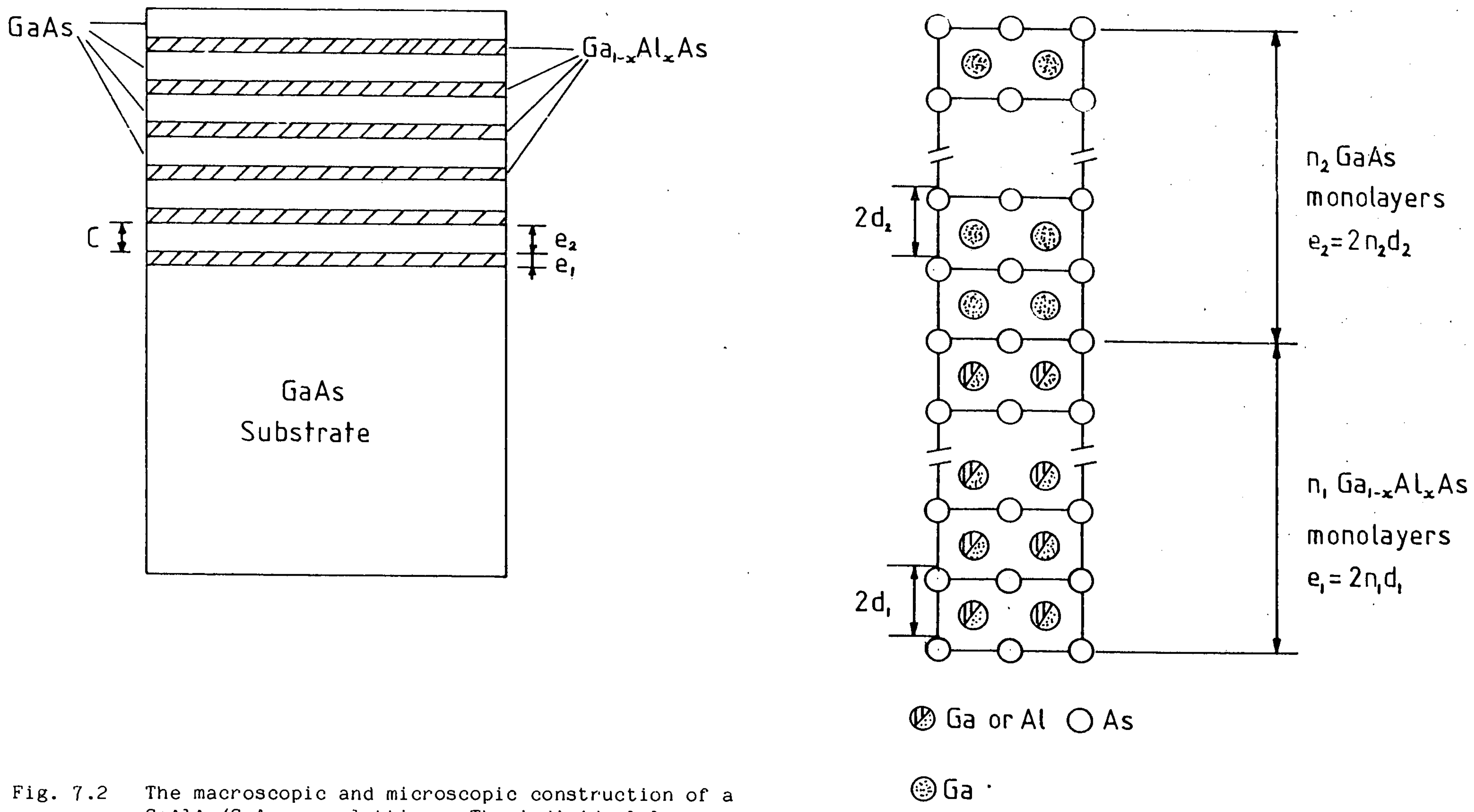


Fig. 7.2 The macroscopic and microscopic construction of a GaAlAs/GaAs superlattice. The individual layers are an integer number of monolayers which is half a unit cell.

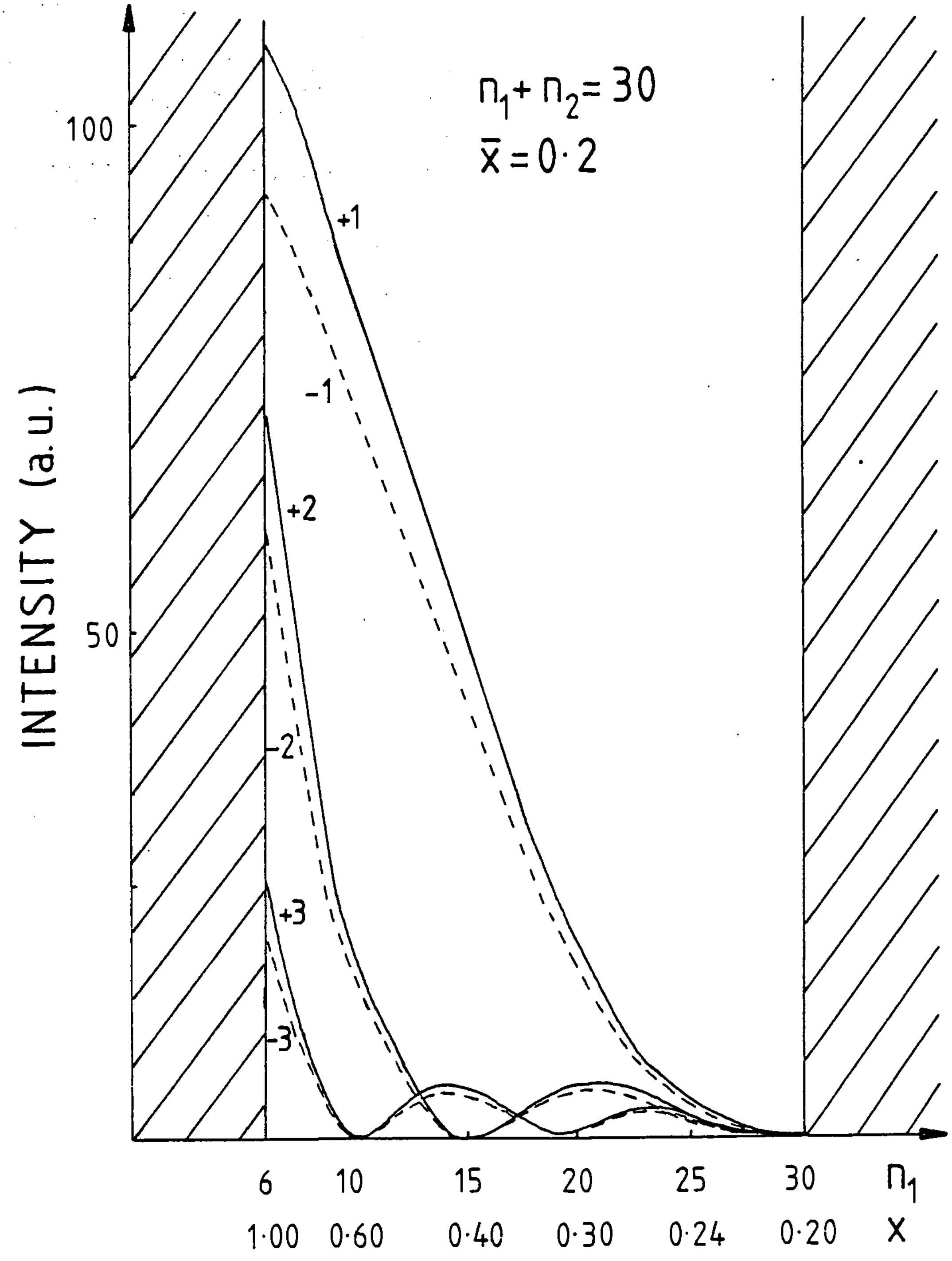


Fig. 7.3 Satellite intensities for the 002 reflection from a $n_1$GaAlAs/$n_2$GaAs superlattice as a function $n_1$ with $n_1$ + $n_2$ constant and $\bar{x}$ constant.

sharp, the structure formed on an atomic scale will be similar to that shown in fig. 7.2. Each layer must be an integer number of monolayers and not necessarily an integer number of unit cells. One monolayer is half a unit cell for a III-V compound. Hence, the individual layers are considered as $n_i$ monolayers throughout the calculations. In the computer programme the section that allows the automatic entry of multiquantum well layer parameters works in terms of monolayers and calculates the individual layer thicknesses from this number and the lattice parameter, taking into account tetragonal distortion if required.

For a $n_1$-$Ga_{1-x}Al_xAs/n_2$-GaAs superlattice, grown on a (001) GaAs substrate, the satellite intensities as a function of $n_1$ and $n_2$, with $n_1+n_2$=constant, are shown in fig. 7.3 for the 002 reflection. From this graph we note that the i-th order satellite peak has zero intensity when $i=k(1+n_1/n_2)$, where k is an integer. Thus for a superlattice where $n_1=n_2$ the even ordered satellites will be absent. Also the intensities of different order satellites do not change with the same function of $n_1:n_2$ , which can be particularly important in the interpretaion of rocking curves. Since defects in the periodicity of the superlattice will tend to reduce the intensity of the satellites, particularly the high order ones, the ratio of intensity of just one satellite to the zeroth order peak may give a false indication of the structure of the superlattice. However, by including the ratio of another

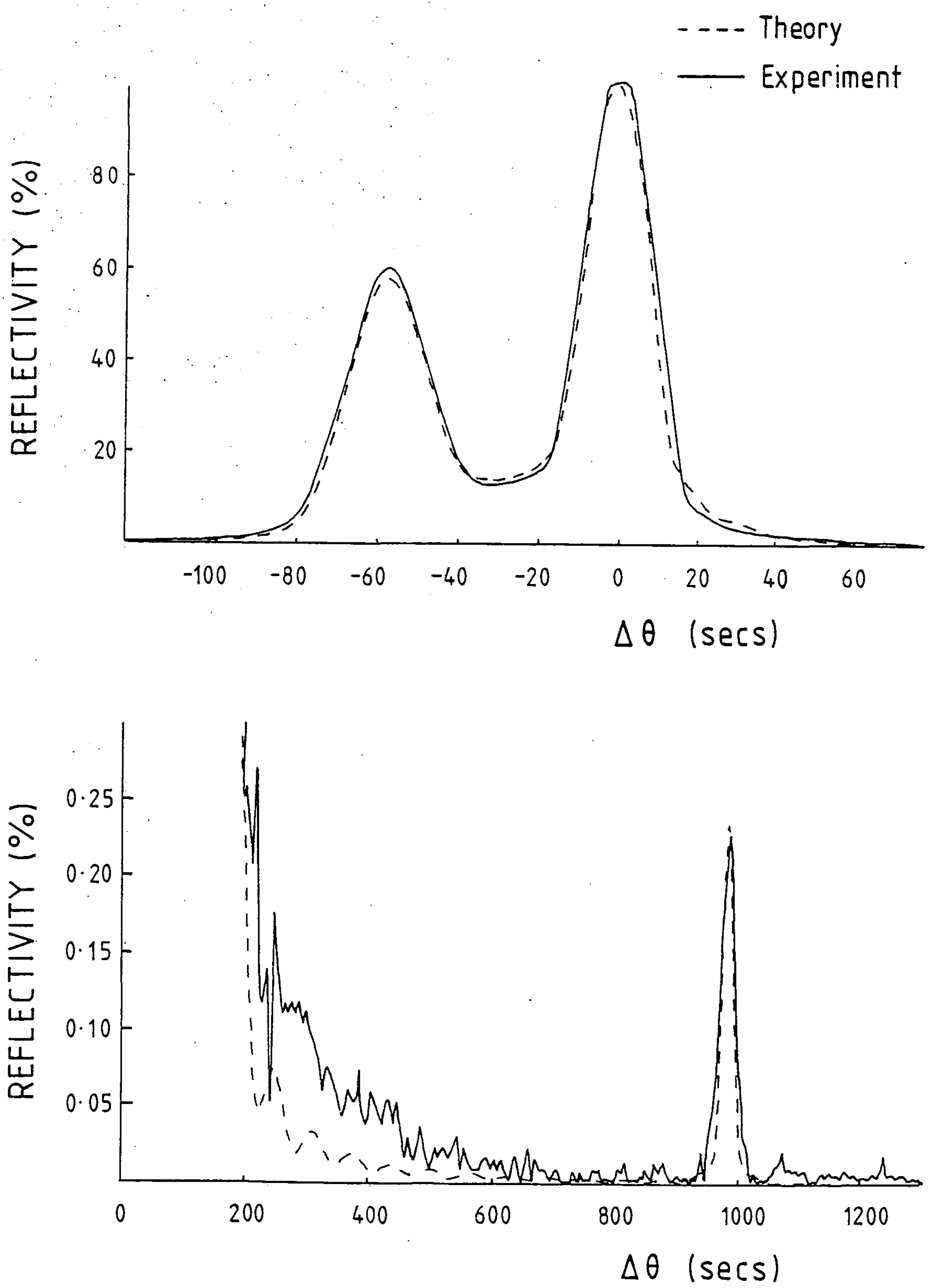


Fig. 7.4 Experimental and theoretical 004 rocking curves from the first $n_1$GaAlAs/$n_2$GaAs sample, at 1.54 Å. Curves (a) show the region around the 004 GaAs substrate reflection and curves (b) show the region out to +1200 secs from the substrate reflection and on an expanded vertical scale, revealing the +1 satellite peak. The computed curve is for $n_1$ = 43, $n_2$ = 28 and x =0.35.

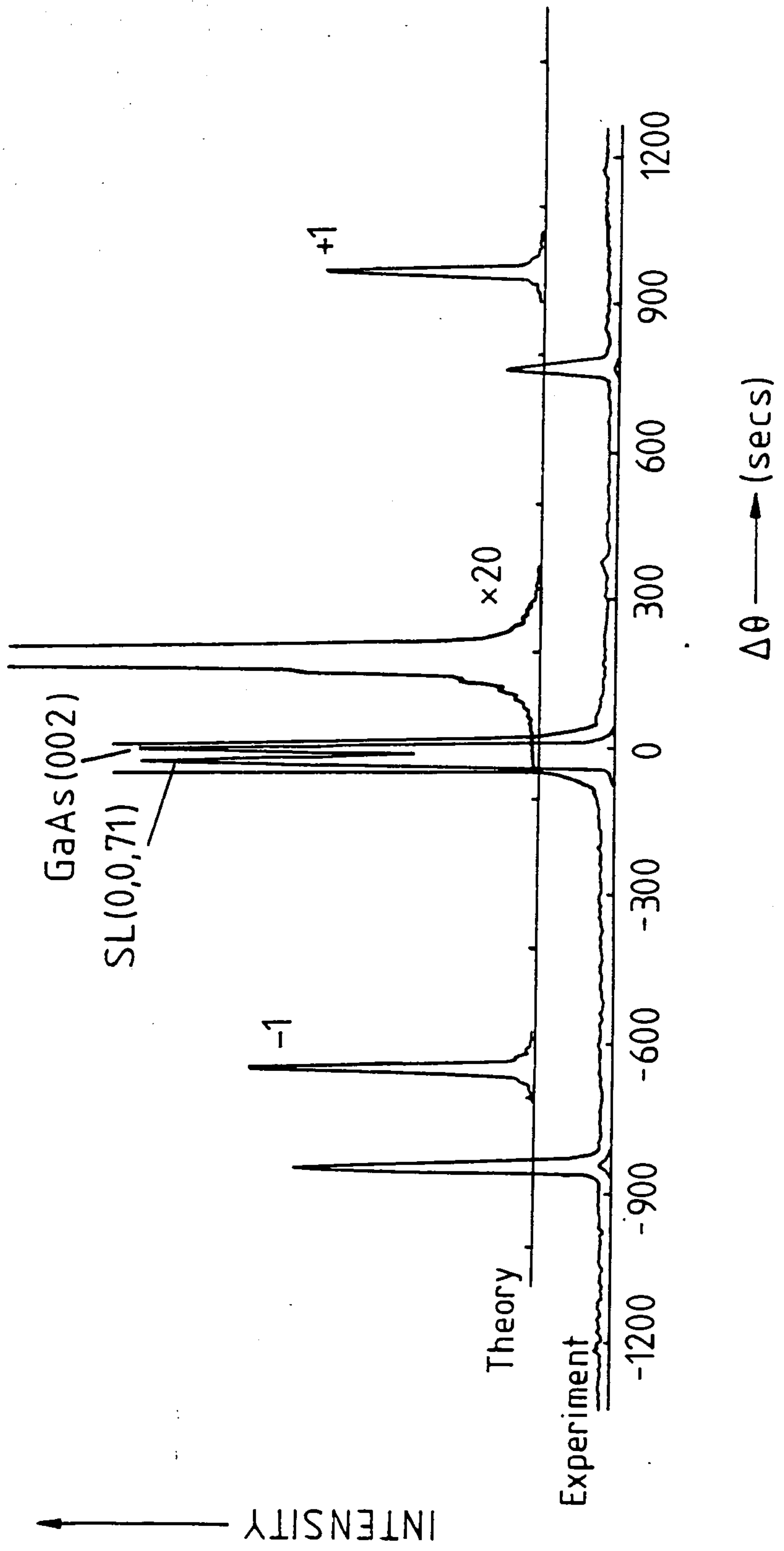


Fig. 7.5 Experimental and computed 002 rocking curves from the same sample used for fig. 7.4. Both the +1 and -1 satellite peaks are easily visible on the vertically expanded scale. The wavelength is 1.5Å.

peak, whose intensity changes as a different function of $n_1:n_2$, this error can be reduced.

## 7.1 Experimental Results

Fig. 7.4 shows the 004 rocking curve for a $Ga_{1-x}Al_xAs$/GaAs superlattice on a (001) GaAs substrate, grown by MBE. The superlattice consists of 50 repeat units and has a GaAs cap of approximately 0.3 µm. The rocking curve was recorded with $CuK\alpha_1$ radiation and an InP first crystal. The zeroth order and substrate peaks are well resolved, but slightly broadened from the theoretical widths due to sample curvature. Only the +1 satellite peak, which is the most intense, could easily be observed and is visible in the expanded region of the curve. No other satellites could be observed near to the 004 GaAs reflection.

Additionally, the 002 rocking curve was recorded at 1.5 Å using the double crystal camera at the SRS and is shown in fig. 7.5. The black-box monochromator was used as the 'first' crystal. Even with the synchrotron source long counting times are needed to provide the extremely good counting statistics required. Consequently, over the recording period the beam intensity decreases due to the decay of the current in the storage ring. Therefore, the incident beam intensity was monitored with an ionisation detector in order that the rocking curve could be compensated for this effect. Because the 002 reflection is a pseudo forbidden reflection for GaAs, being only present due to the difference in scattering factors of the Ga and As atoms, the presence of Al in the alloy provides a relatively large modulation in the structure factor of the layers. Consequently, the satellite intensities are greater than those near to the 004 reflection and the +1 and -1 satellites are easily visible in the rocking curve.

From the separation of the +1 satellite and the zeroth order peak near the 004 reflection the value of $L=n_1+n_2$ can be determined. If we label the zeroth order peak as 2L and the +1 peak as 2L+1 we have

$$(2L+1)\lambda = 2C\sin\theta_{2L+1} \qquad (7.1)$$

and

$$2L\lambda = 2C\sin\theta_{2L} \qquad (7.2)$$

from Bragg's Law, with C the superlattice period. Thus,

$$\frac{\sin\theta_{2L+1}}{2L+1} = \frac{\sin\theta_{2L}}{2L} \qquad (7.3)$$

ie

$$2L = \frac{\sin\theta_{2L}}{\sin\theta_{2L+1} - \sin\theta_{2L}} \qquad (7.4)$$

With the values of $\theta_{2L}$ and $\theta_{2L+1}$ determined from the displacement from the substrate peak we find that 2L=142, to the nearest integer value. In fact L rarely turns out to be an exact integer due to deviations in the periodicity of the superlattice and the nearest value should be taken. With this value of 2L and the absolute position of the zeroth order peak the superlattice period can be determined and is found to be C=200.8 Å. The zeroth order and +1 peaks can

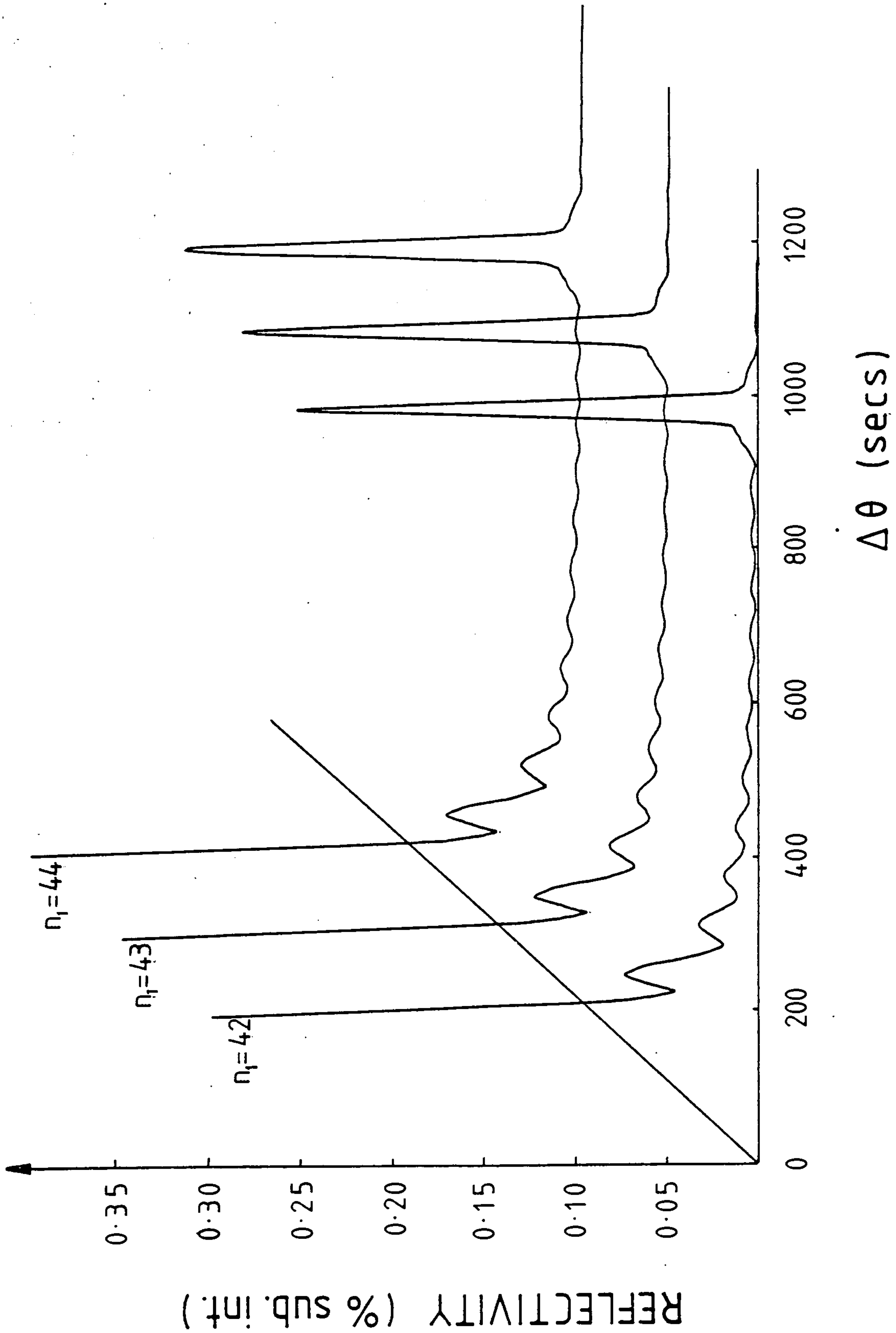


Fig. 7.6 Computed 004 rocking curves for CuK$\alpha_1$ radiation for values of $n_1$ = 44, 43 and 42 showing the variation in +1 satellite intensity. The overall parameters are as for the first GaAlAs/GaAs sample.

then be labelled as the (0,0,142) and (0,0,143) reflections from a lattice spacing of 200.8 Å. This value is confirmed from the spacing of the +1 and -1 satellites near to the 002 GaAs reflection. These peaks can be labelled as (0,0,70), (0,0,71) and (0,0,72) for the -1, 0 and +1 respectively.

From the separation of the zeroth order and 004 peaks the average mismatch of the superlattice can be determined. Assuming coherent interfaces this then gives the average composition, $\bar{x}$, which is defined as $\bar{x} = n_1x/(n_1+n_2)$. The value found, assuming Poisson's ratio for GaAs and GaAlAs are equal, is $\bar{x}$=0.185 or 213 ppm mismatch.

Rocking curves can then be calculated for this superlattice for varying values of the ratio $n_1:n_2$ with $n_1+n_2$ and $\bar{x}$ constant. A value of the sample curvature was used to give the best fit to the widths of the 004 and zeroth order peaks. The best fit 004 and 002 rocking curves are shown in figs. 7.4 and 7.5 for comparison with the experimental curves. Values of $n_1$ and $n_2$ thus determined are $n_1$=43 and $n_2$=28, giving x=0.35. Fig. 7.6 demonstrates the sensitivity of the +1 satellite intensity to the value of the ratio $n_1:n_2$ for the 004 reflection. The three curves shown are for the ratios 42:29, 43:28 and 44:27. The satellite intensity of the 42:29 curve is approximately 9% greater than that of the 43:28 curve, while that of the 44:27 curve is approximately 8% weaker. Oscillatory structure on the high angle side of the substrate peak is not changed nor are the shapes of the substrate and zeroth

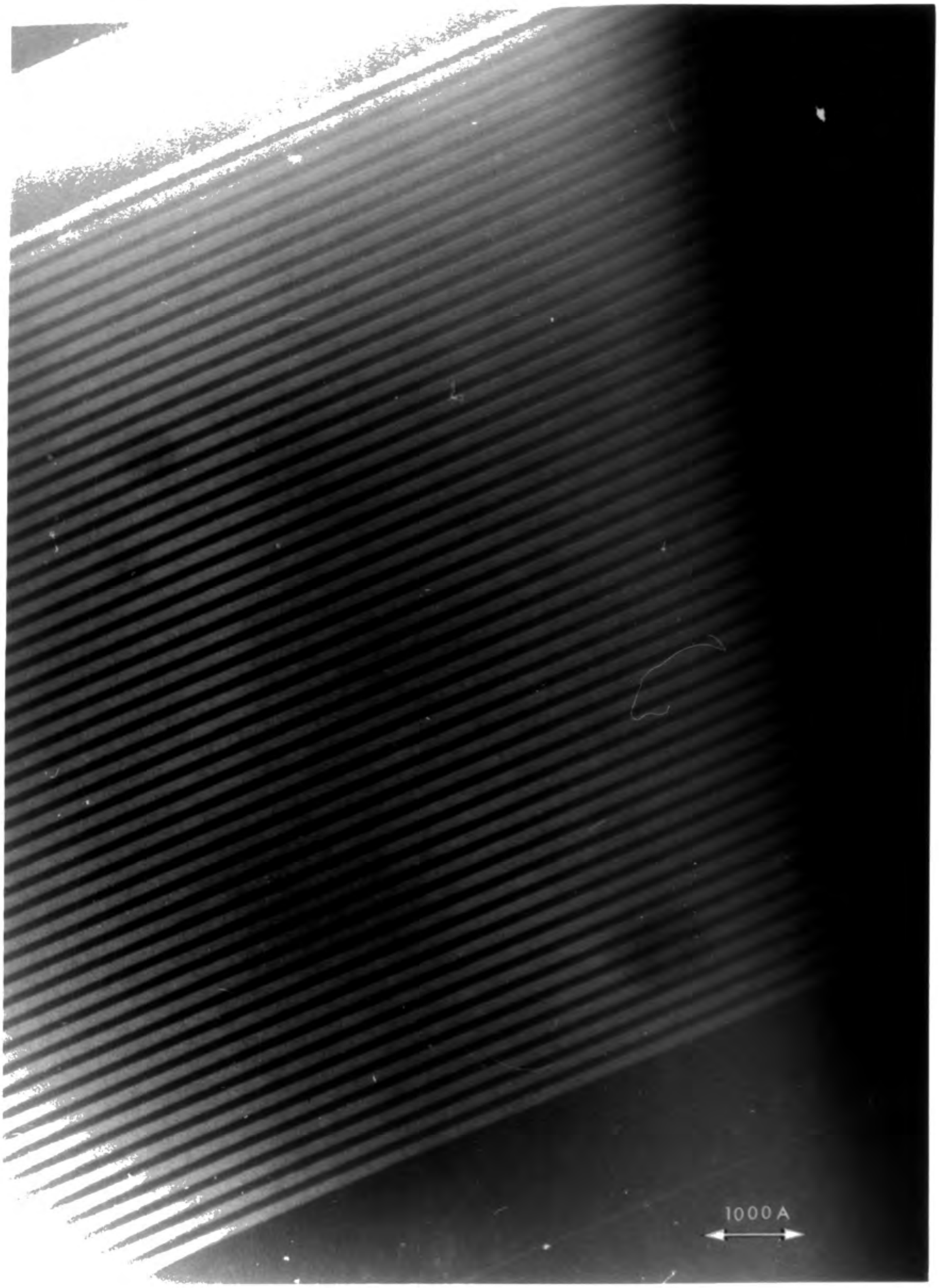


Fig. 7.7 The 002 dark field transmission electron micrograph of the first GaAlAs/GaAs superlattice. The GaAlAs layers are the lighter regions due to the enhanced diffracted intensity.

order peaks. Therefore, this fitting procedure will be unaffected by any superlattice imperfections that reduce the +1 satellite intensity by less than approximately 5%. Imperfections that reduce the intensity by a greater amount would give a larger value of $n_1$ using this fitting procedure. The intensities of the satellites near the 002 GaAs reflection, particularly the -1 peak, also agree well with those calculated for $n_1=43$ and $n_2=28$. However, the intensity ratio of a higher order satellite, whose intensity varies as a different function of $n_1:n_2$, could not be used to improve the fitting procedure since no such peaks could be observed either near the 004 or 002 GaAs reflections. The value of x deduced agrees well with that expected from the growth conditions. Using this composition the intensity of the -1 satellite was calculated and was found to be much weaker than the +1 satellite, explaining why it could not be observed in the experimental rocking curve.

A transmission electron microscopy study of this sample was also carried out at British Telecom Research Laboratories. Taking a thin section perpendicular to the surface the micrograph shown in fig. 7.7 was obtained. This represents the 002 dark field image, with the GaAs and GaAlAs layers easily distinguished by the enhanced intensity diffracted by the GaAlAs layers. Although the exact magnification could not be accurately determined the values of the relative layer thicknesses agree extremely well with those obtained from the rocking curve. The micrograph is extremely useful for showing any deviations in the repeat period and for showing the thickness uniformity along the layers.

Some discrepancies do exist between the calculated and experimental rocking curves, particularly in the intensity of the satellite peaks in the region of the 002 GaAs reflection. Further, satellites greater than the first order could not be observed. Both of these facts indicate that the superlattice is not perfect; either the repeat unit varies throughout the superlattice about some mean value or the interfaces are not sharp due to interdiffusion between the GaAlAs and GaAs layers. The effect of these variations on the intensity of the satellite peaks is discussed later.

Another example of a GaAlAs/GaAs superlattice was also studied. This sample consisted of 100 layers of GaAs and 99 layers of GaAlAs with an encapsulating layer of 1 μm of GaAlAs and 200 A of GaAs, as shown schematically in fig. 7.8.

The diffracted intensity in the region of the 002 GaAs reflection was recorded in a similar manner to that used for the previous sample and is shown in fig. 7.9. Again only the +1 and -1 satellites could be observed, with considerably smaller intensities than those obtained for the previous sample. The intensities of both satellites were approximately one thousandth of the intensity of the zeroth order peak, which could not be completely resolved from the 002 GaAs peak.

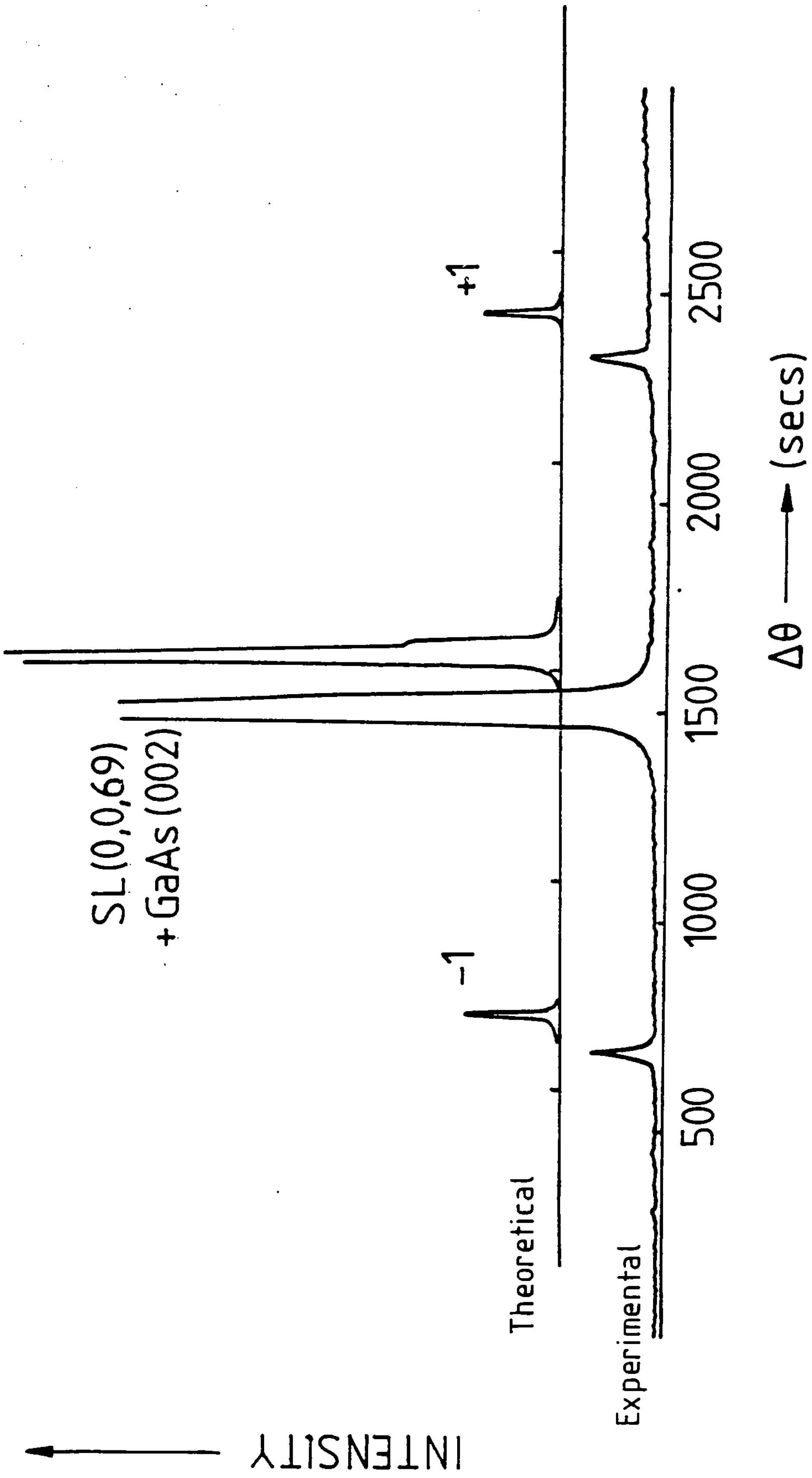


Fig. 7.9 The 002 experimental and computed rocking curves from the second GaAlAs/GaAs superlattice, its construction is shown in fig. 7.8. The vertical scale is expanded to show the +1 and -1 satellite peaks. CuK$\alpha_1$ radiation was used. The computed curve as for $n_1$ = 35, $n_2$ = 34 and x = 0.30.

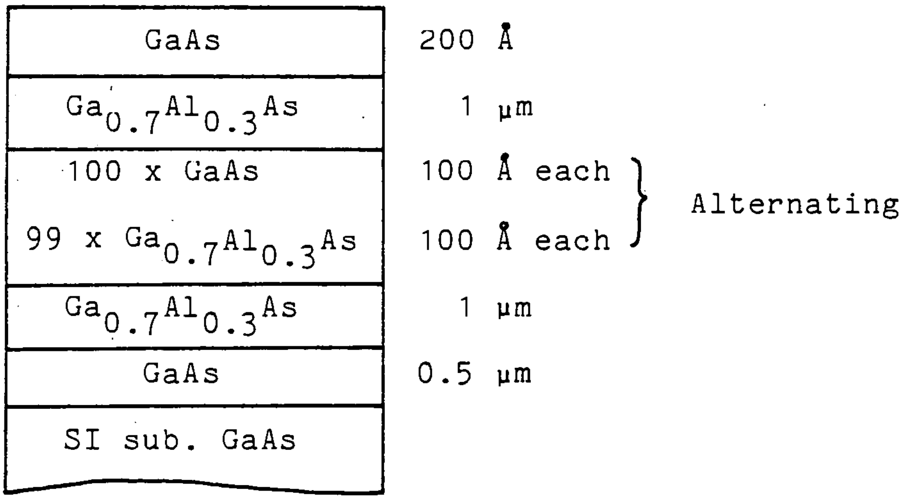


Fig. 7.8 Schematic diagram of the construction of the second GaAlAs/GaAs superlattice.

From the satellite separations the value of $n_1+n_2$ is found to be 69. Thus, from the position of the zeroth order peak the superlattice period is 195 Å, compared to the 200 A expected from the growth conditions. The best fit calculated curve is shown in fig. 7.9, for $n_1=35$ and $n_2=34$. Agreement bewteen the satellite positions is very good, although the calculated peaks are slightly more intense and narrower. Since $n_1 \approx n_2$ we would expect the second order satellite peaks to be particularly weak which explains why they could not be observed. Since the satellite intensities are smaller than those predicted theoretically we again expect there to be some dispersion in the superlattice repeat period.

A similarly designed sample, its construction as shown in fig. 7.10, was also studied. The 002 rocking curve is shown in fig. 7.11, again revealing only satellites up to

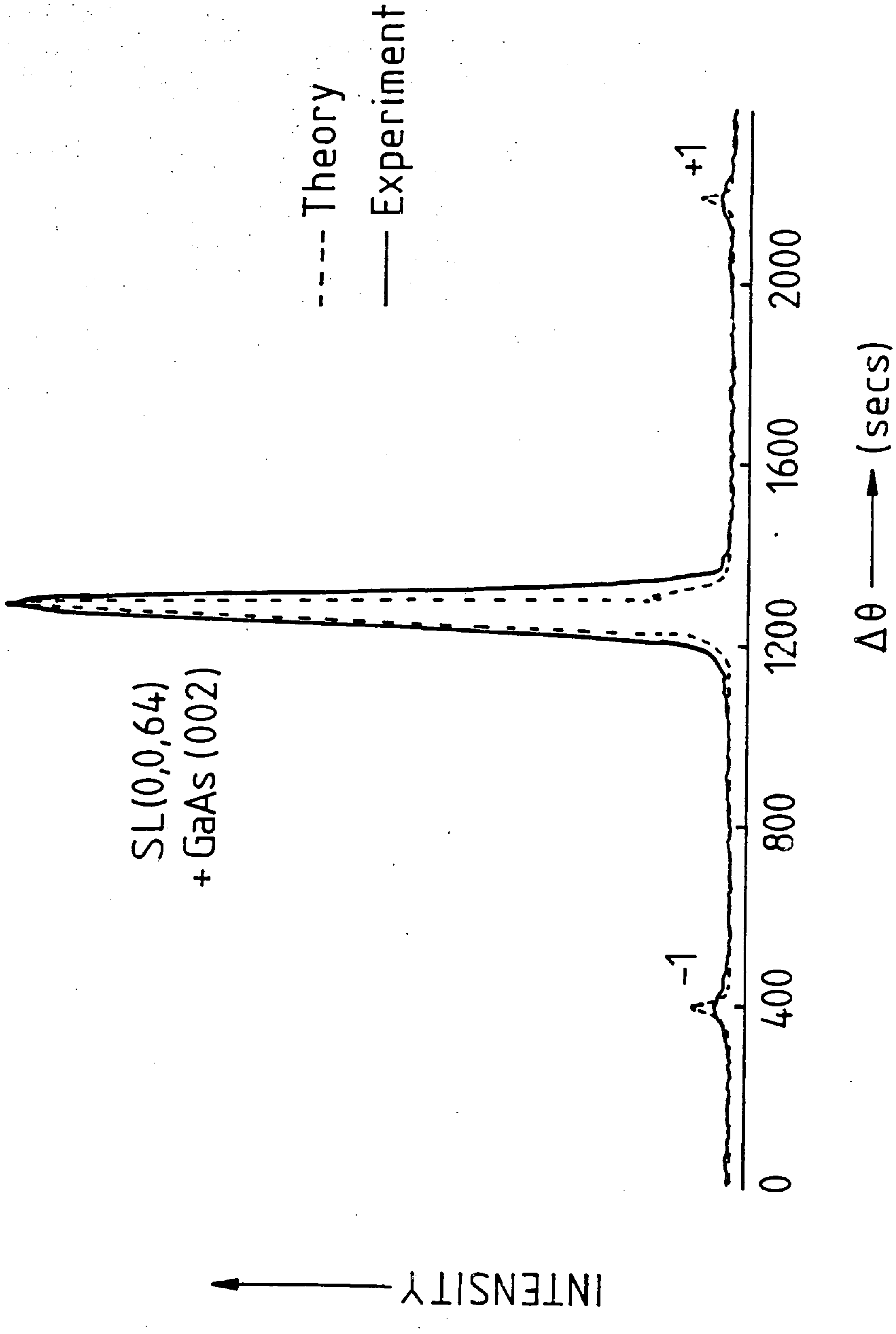


Fig. 7.11 002 experimental and computed rocking curves from the third GaAlAs/GaAs superlattice. The CuK$\alpha_1$ radiation was used.

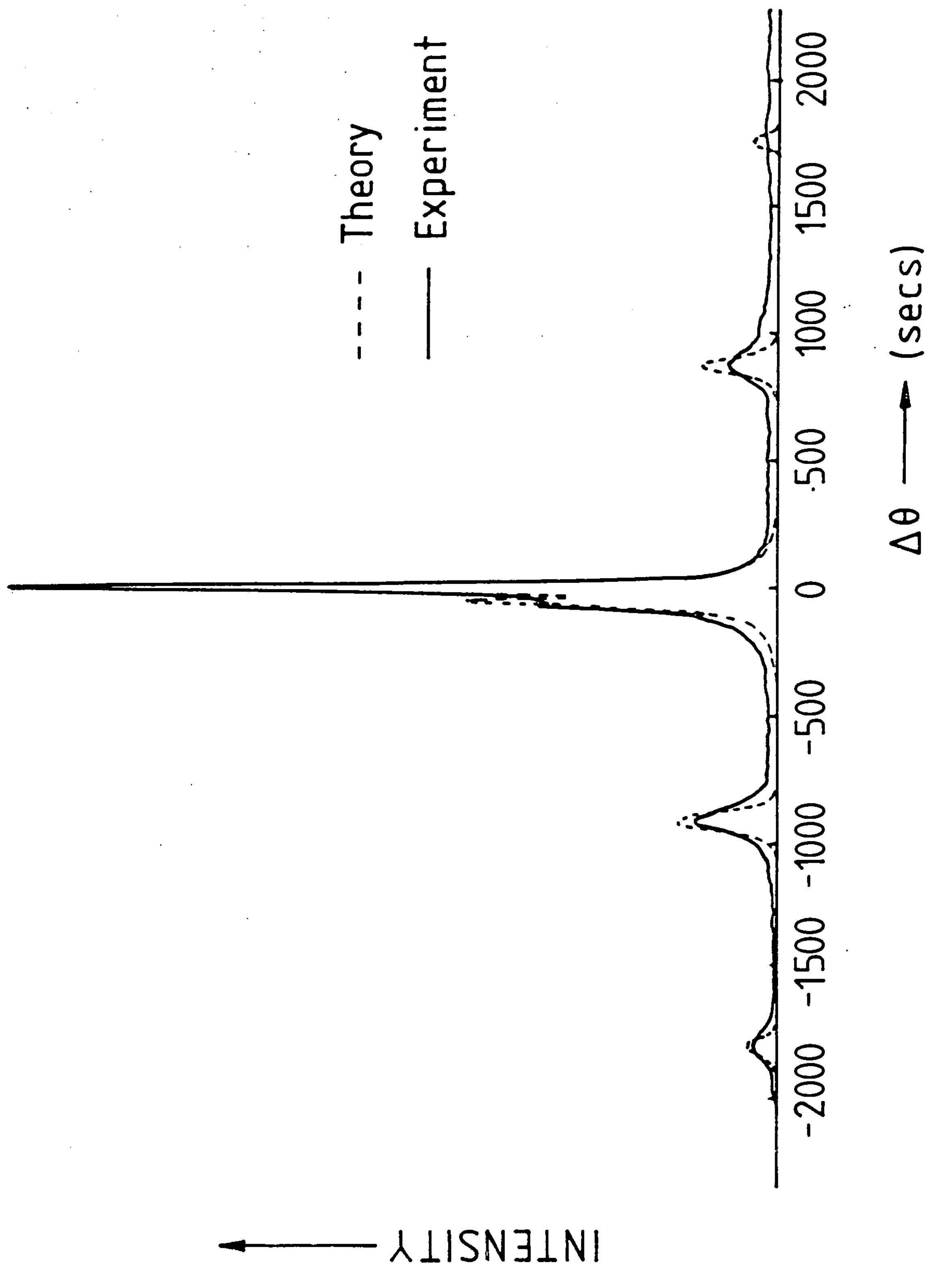


Fig. 7.12 002 experimental and computed rocking curves, with CuK$\alpha_1$ radiation, from the GaInAs/InP superlattice. The +1, -1 and -2 satellites are easily visible in the experimental curve. The computed curve is for $n_1$ = 38, $n_2$ = 26 and x = 0.537.

the first order. Satellite intensities were similar to those for the previous sample, their spacing giving $L=n_1+n_2=64$. From the position of the zeroth order satellite peak, the (0,0,64) reflection, the superlattice period is found to be C=160 Å, which agrees reasonably well with the 175 A expected. The best fit calculated curve for $n_1=38$, $n_2=26$ is also shown in fig. 7.11. Once again the calculated satellite intensities are slightly smaller than those in the experimental curve.

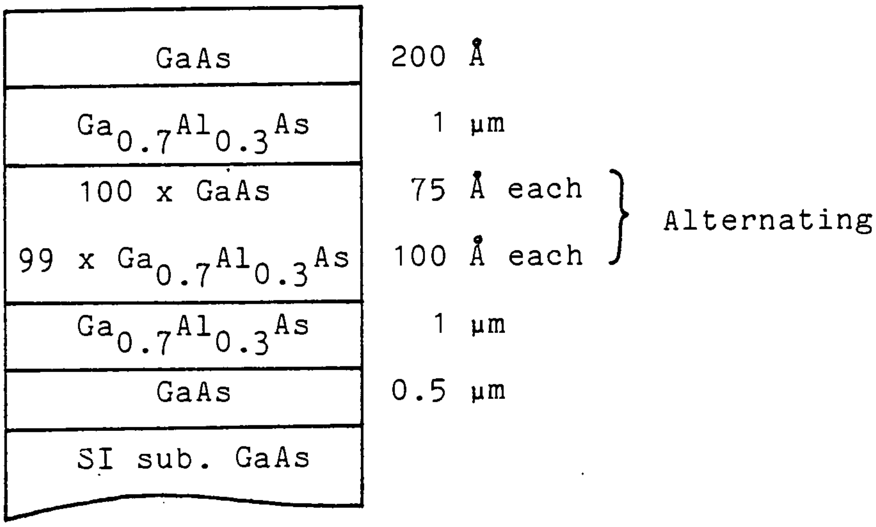


Fig. 7.10 Schematic diagram of the construction of the third GaAlAs/GaAs superlattice sample.

A $n_1$-GaInAs/$n_2$-InP superlattice, grown by MOCVD on an (001) InP substrate (Moss and Spurdens, 1984), has also been studied. The 002 rocking curve is shown in fig. 7.12, which reveals satellites up to the second order. The -1 and -2 satellites are easily visible, being approximately 10% of the intensity of the zeroth order peak, but only the +1 satellite can be observed on the high angle side of the zeroth order peak. The zeroth order peak is only partially resolved from the 002 InP substrate peak and is on the low angle side. Since for the 002 reflection the difference in structure factor between GaInAs and InP is much greater than that between GaAlAs and GaAs the satellites are much more intense. From the separation of these satellites the value of $L=n_1+n_2$ is calculated and found to be 64, giving a superlattice period of C=188 Å. The best fit calculated curve was found for $n_1=38$, $n_2=26$ and is also shown in fig. 7.12. Here, the ratio of intensities of the -1 and -2 satellites was used to aid the determination of $n_1:n_2$ and help reduce the errors introduced by superlattice imperfections. From these values and the value of $\bar{x}$, determined from the separation of the zeroth order peak and the substrate peak, x is found to be 0.537.

There are again some discrepancies between the computed and experimental rocking curves, with the experimental satellite peaks being both less intense and broader than the computed ones. Further, the +2 satellite could not be observed in the experimental curve yet is easily visible in the computed curve. These facts again lead to the assumption that the superlattice is imperfect with the most likely deviation being in its repeat period. This is not unexpected for this sample since the control of the growth times for each layer was performed by hand and it is known that perfect timing was not achieved. This was confirmed by a TEM study of the sample and the 002 dark field micrograph

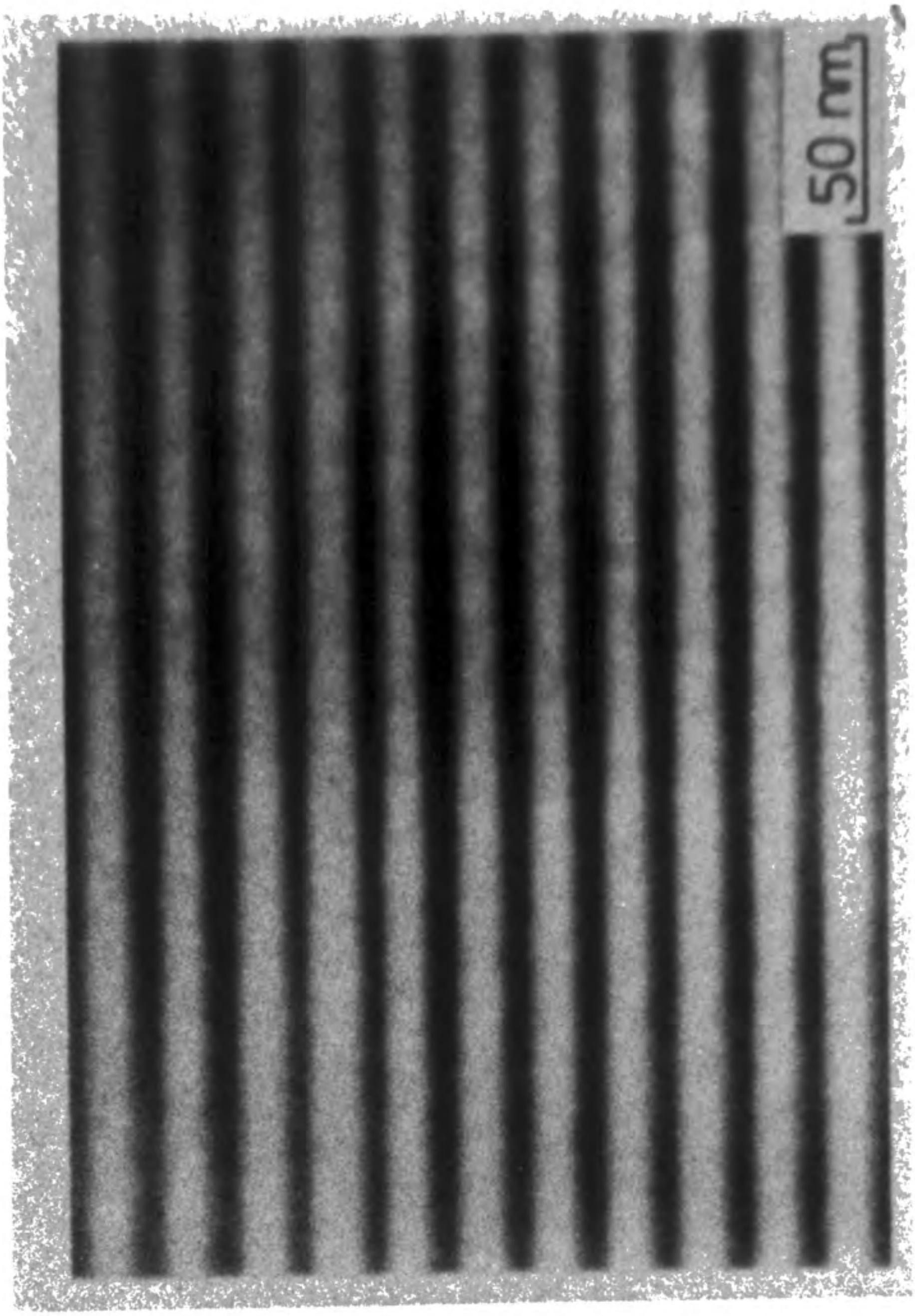


Fig. 7.13 002 transmission electron micrograph of the $n_1$GaInAs/$n_2$InP superlattice, revealing the non uniformity in both repeat unit and layer thickness along the layers.

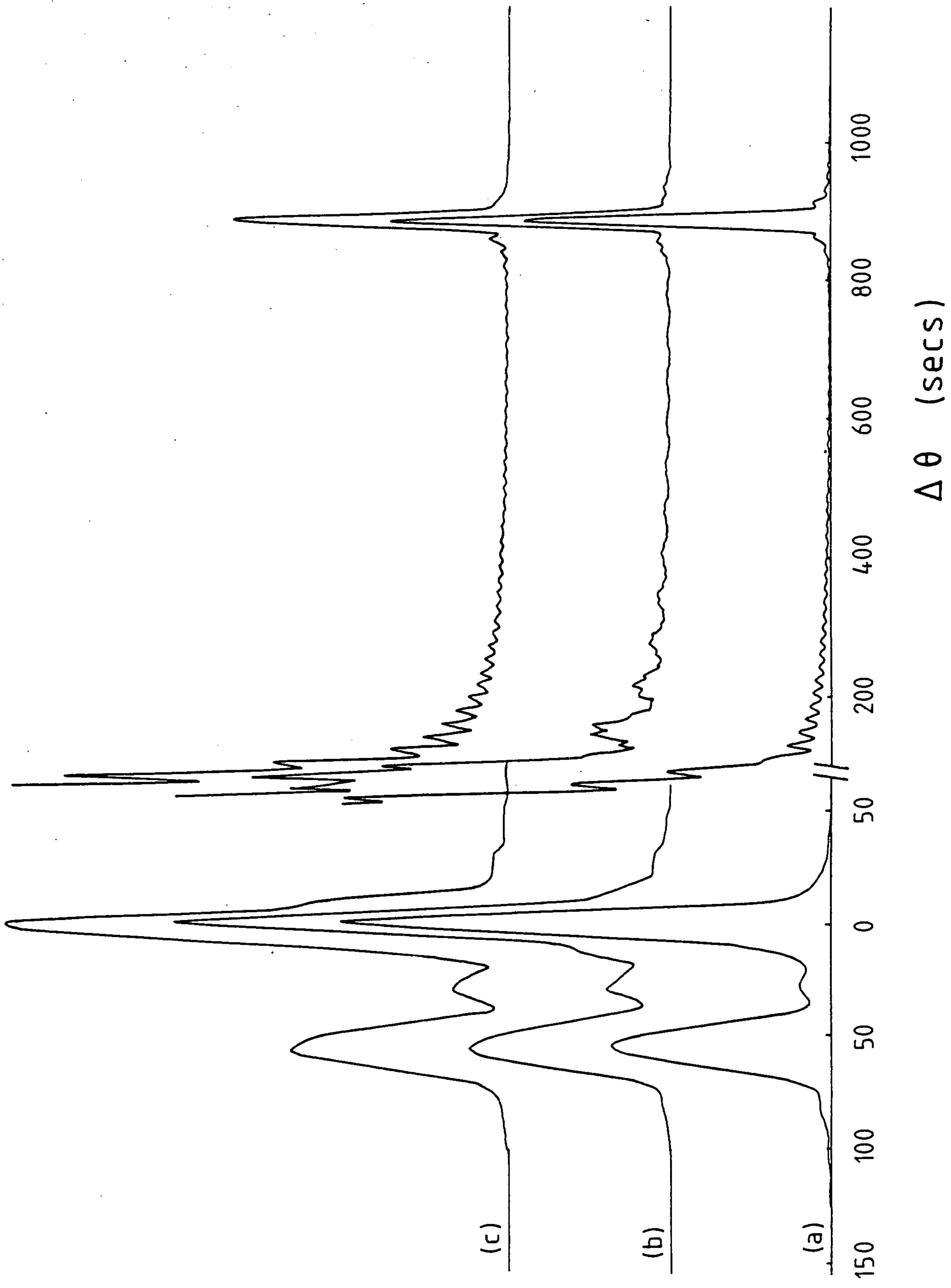


Fig. 7.14 The effect on the 004 computed rocking curve for the first GaAlAs/GaAs superlattice of adding a GaAs capping layer. Curve (a) is for no cap, (b) for 0.3 μm cap and (c) for a 1.0 μm cap. The +1 satellites are shown in the vertically expanded regions. CuK$\alpha_1$ radiation is used.

is shown in fig. 7.13. The individual layers are easily distinguishable by the enhanced intensity from the InP layers. There is noticeably a lack of uniformity in thickness along the layers as well as dispersion in the superlattice period.

## 7.2 The Effect of Adding a Capping Layer to the Superlattice

The effect of adding a relatively thick capping layer, of similar composition to one of the components of the superlattice, is demonstrated by the 004 rocking curves shown in fig. 7.14. The radiation is CuK$\alpha_1$ and the structure used is that of the first GaAlAs/GaAs sample but the effect of sample curvature has not been included. If the capping layer is sufficiently thick to diffract strongly it is essential that the dynamical theory is used to calculate the overall diffracted intensities and significant changes in the shape of the rocking curve do occur as seen in fig. 7.14. The substrate peak has been considerably affected by the 1 µm cap, being both broadened and with an intense 'bulge' on the high angle side. Additionally, the weak peak between the substrate and zeroth order peak is much more intense than in the curve for the sample without a cap. These main peaks are also similarly affected, although to a somewhat lesser extent, by the addition of a 0.3 µm cap. Also shown in the figures are the expanded regions revealing the +1 satellite peak. Again, for the 1 µm cap significant modifications are observed with the oscillation on the side of the substrate peak changed as well as the satellite intensity reduced. The position of the satellite peak is not changed with respect to the substrate peak. For the 0.3 µm cap the oscillations are again heavily modified but only an extremely small change in satellite intensity occurs. Since the amplitudes of all of these oscillations on the side of the substrate peak are so small they would not be expected to be observed in any experimental rocking curves. Clearly, if an even thicker cap was to be added the modifications would be greater but these rocking curves show that the presence of a cap that is only about 10% of the extinction distance can make a significant difference to the intensity of the satellite peaks. Further, even the addition of only a 0.3 µm cap does have a significant effect on the shape of the substrate and zeroth order peaks. However, when sample curvature is included in these calculations the effects do become less noticeable.

## 7.3 The Effect of Dispersion in the Superlattice Period and of Interdiffusion between the layers.

The effect of dispersion in the superlattice period, ie the thickness of the layers forming the repeat unit varies throughout the layer, is discussed first. This effect can easily be modelled by introducing a random deviation in the number of monolayers, about a mean value, for each layer used in the calculation. The size of the random deviation can then be increased and rocking curves calculated. Note

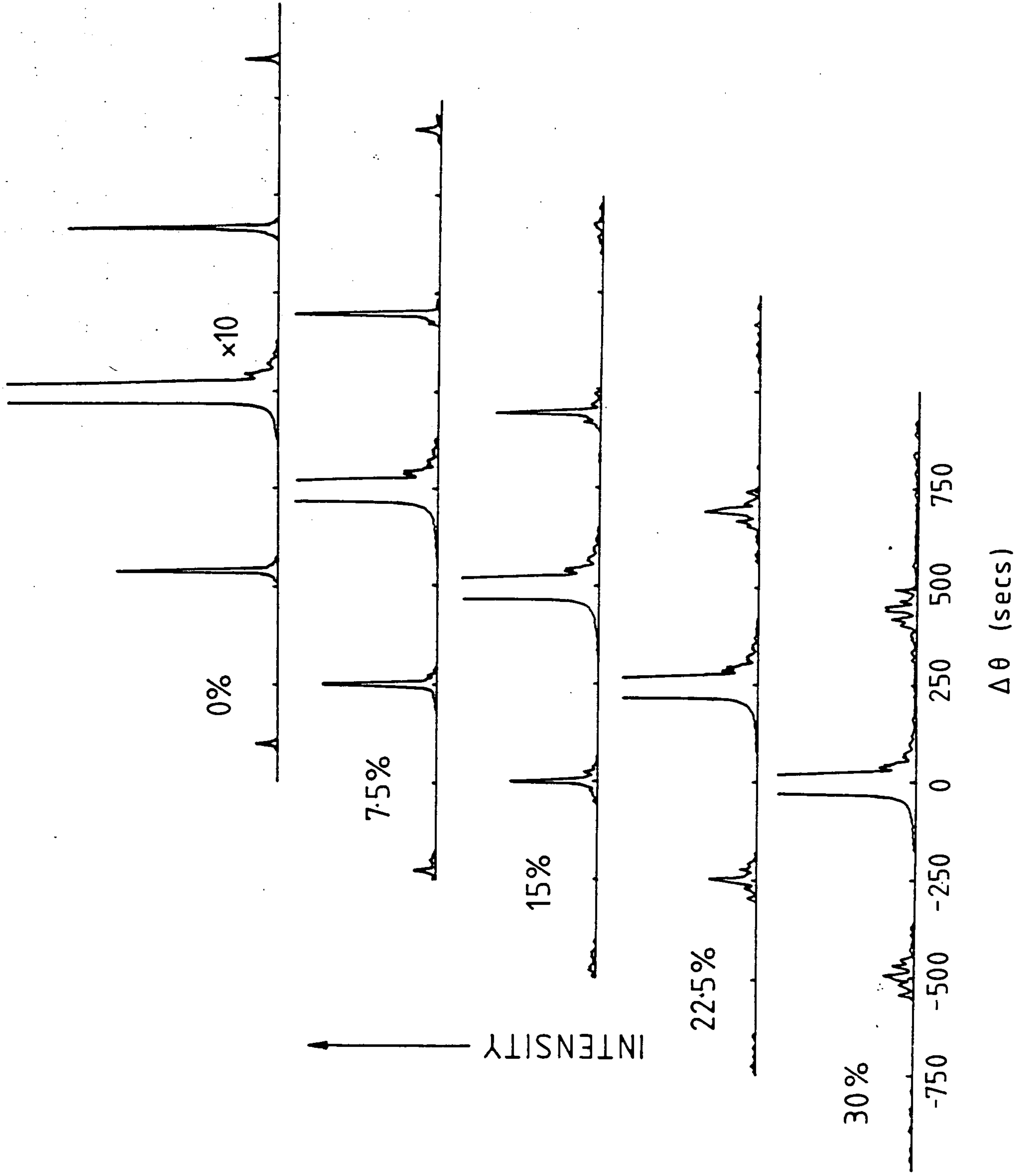


Fig. 7.15 The effect of dispersion in the superlattice period on 002 computed curves for the GaInAs/lnP superlattice. CuK$\alpha_1$ radiation is used. The individual layer thicknesses vary randomly about the perfect superlattice values within the range (mean - percentage of mean) to (mean + percentage of mean) where the percentage used is as indicated by the curves.

that the satellite intensity formulae developed for perfect superlattices by other workers (see for example Vardanyan, Manoukyan and Petrosyan, 1985) are not well suited to modelling this effect. Fig. 7.15 shows the effect of increasing the size of the random fluctuation in the 002 rocking curve from a GaInAs/InP superlattice with a construction similar to that found for the sample studied experimentally. The amount of random deviation is taken as a percentage of the mean number of monolayers forming each layer. In the curve for zero deviation satellite peaks up to the second order are easily visible with well formed, narrow peaks. However, as the size of the fluctuations are increased the second order peaks rapidly disappear, while the intensity of the first order peaks is less rapidly affected. For the 7.5% fluctuations the second order satellite peaks have already lost their well formed shape and are less intense. The +1 satellites have also lost intensity but still retain a similar shape. The intensity of the satellites is not affected equally on either side of the zeroth order peak which is particularly important and the ratio of intensities of the +1 and -1 satellites does not remain the same; the +1 satellite, which is the most intense, is reduced in intensity more rapidly than is the -1 peak. When the fluctuations are as large as 15% the second order satellites have already reduced to a very spread out peak which would not be visible in any experimental curve. Again the first order satellites have been reduced in intensity but also have become considerably broadened in the tails. Beyond 15% the first order satellites are further reduced in intensity and broadened and by 30% have only become a very spread out, badly formed peak. This agrees well with the observations of other workers (see for example Kevarec et al, 1984).

Since we have easily observed one second order satellite peak in the experimental curve from the GaInAs/InP superlattice the fluctuations in repeat period must be less than about 10% by comparison with the above computed curves. This agrees well with the measurements taken from the TEM micrograph. Variations in layer thicknesses along the layers would also be expected to have a significant effect on the rocking curve, but this effect is not so easily modelled and would require calculating a number of rocking curves for different layer thicknesses and then adding them.

If interdiffusion has occurred between the layers (for example of Al in a GaAlAs/GaAs superlattice) the interfaces will become less sharp. If a large degree of interdiffusion has occurred the mismatch and composition variations throughout the superlattice will tend towards sinusoidal. Clearly, to model this effect with a computational procedure designed for stepwise variation would require a large number of layers, with the sinusoidal variation modelled by dividing each layer into a number of laminae of constant composition. Consequently, the number of layers required for the computation can easily be up to an order of magnitude greater than the actual number of layers in the structure, greatly increasing the computation time. Calculations based on a Fourier expansion of the composition variation are perhaps better suited to studying this effect and have been used successfully by a number of workers (see for example Fleming et al, 1980). Kervarec et al (1984) have also calculated the satellite intensities as a function of the interface width by adding a sinusoidal variation to the stepwise variation. This showed that the satellite intensities decrease as the interface width is increased, with the higher order satellites more rapidly affected.

From the TEM results for two of the samples studied the interfaces did appear to be reasonably sharp, which agrees well with the previous observations of Dernier et al (1977) where the GaAl/GaAlAs interfaces were determined to be less than two monolayers wide. The most obvious irregularity in the superlattice was in the repeat period (particularly for the GaInAs/InP sample) and, as shown, this has a very large effect on satellite intensities explaining why those observed were less intense than the computed ones.

# DISCUSSION AND SUGGESTION FOR FUTHER WORK

The computational method developed has been shown to produce rocking curves that agree extremely well with experimental ones for single uniform layers, single layers with depth dependent compositions and multiquantum well structures. Often there are discrepancies between the computed and experimental rocking curve but these can be accounted for by imperfection in the samples. The technique is particularly useful in the determination of the composition variation in graded layers and provides an accurate means of determining layer thicknesses for all types of samples. However, due to the high sensitivity of the rocking curve to composition and the large numbers of parameters that can be altered it can be extremely difficult to fit experimental rocking curves exactly, especially for graded layers. Bulk sample curvature also has a very major affect on the rocking curve and can mask much of the fine detail present in the computed curves neglecting this effect.

It has also been found that the rocking curve from samples with only a few layers of similar composition are not straightforward to interpret. The number of peaks observed in the rocking curve very rarely equal the number of layers. Further problems are encountered if the layer is thin relative to the extinction distance, making the diffracted intensity from the layer only a few percent of that of the substrate peak. Also, if thin layers are present buried between thicker layers of a similar composition, as in many of the device structures currently being developed, their detection can be extremely difficult. For the thin, single layers the use of highly asymmetric reflections in the low angle of incidence case has been studied. Diffracted intensities from the layer were found to increase dramatically as the angle was reduced to near grazing incidence. However, due to the high sensitivity of the layer peak to the incident angle difficulties arose in the simulation of such rocking curves. This effect could be due to surface misorientation and further experiments are required to ascertain this. For thin layers sandwiched between thicker ones this technique could also be used with angles above those of near grazing incidence so that the top, thick layer does not diffract all of the intensity. When using a synchrotron source the incident angle can be adjusted to tailor the extinction distance to that required and it would be important to assess whether similar techniques could be used with a laboratory source. Since a different first crystal reflection would have to be used in order to prevent either extreme beam compression or expansion a channel cut monochromator would be extremely useful to reduce the effect of the dispersion introduced. The use of a high incidence angle with a grazing exit beam may also be useful for these studies but as yet no experimental results have been obtained to confirm this.

The use of gap fringes may also become extremely useful for the study of these structures with thin, buried layers, as recently discussed by Tanner and Hill (1985). Here the thin layers produce strong gap fringes in the rocking curve which vary as a function of both their depth and thickness. To date only theoretical curves have been produced and good quality experimental data is required for comparison.

Due to the lack of perfection in the superlattices studied only a small number of satellites were observable even with the 002 forbidden reflection and the extremely good counting statistics afforded by the high intensity of the synchrotron source. The effect of dispersion in the superlattice period has been modelled and found to have a dramatic effect on the satellite intensities. However, the effect of both interdiffusion between the layers and thickness irregularities along the layers remain to be modelled. In order to study the effect of interdiffusion it may well be better to use a different computational approach based on numerically integrating the Takagi-Taupin equations with a continuous composition modulation. If the stepwise approach is used the composition variation would need to be modelled by an extremely large number of laminae, which combined with the large angular ranges required in the calculation will greatly increase the computational time. This may not be a problem with access to a large, powerful computer. However, this approach would have the advantage that the combined effect of both dispersion in the periodicity and interdiffusion could be studied.

Additionally, this technique could also be used to simulate rocking curves from ion implanted superlattices to compare with experimental results similar to those presented by Terauchi et al (1984).

Little use has been made of double crystal topography to study these materials mainly due to the effect of sample curvature. Since any good quality heteroepitaxial layer must produce sample curvature only extremely small regions of the crystal can be imaged in one topograph. Consequently, single crystal topography has been more extensively used to examine the samples for the presence of mismatch dislocations.

Finally, since the need for this technique of rocking curve simulation is becoming more widespread, particularly by the semiconductor manufacturers, ways by which the computation can be become more easily available are required. With the advent of more and more powerful microcomputers, particularly those based around the 68000 series of processors, it may well be feasible to perform the computations on these machines. If this step could be made it is certain that this technique could become more and more important. Further, it ought to be perfectly feasible to automatically analyse rocking curves from single uniform layers provided that well resolved peaks are obtained. The analysis is greatly simplified when the mismatch and areas under each peak are determined. It is unlikely that such a technique could be developed for graded layers due to the large number of laminae required and the sensitivity of the rocking curve to small variations in the composition profile. Both sample curvature and large dislocation densities would also greatly hinder such an analaysis. For multiple layered structures further problems would be encountered due to the additional interference peaks present in the rocking curves. However, even if automated analysis cannot be easily implemented there remains no doubt as to the usefulness of rocking curve simulation to provide a highly sensitive, non-destructive characterisation method.

# APPENDIX A

## COMPUTER PROGRAMMED FOR THE COMPUTATION OF ROCKING CURVES

This appendix contains complete listings of the computer programme required to calculate and plot rocking curves from III and IV semi-conductor crystals with epitaxial layers of any composition. The programmes run on an IBM 4341/I at Durham University.

```
{ PROGRAM TO CALCULATE INPUT PARAMETERS FOR THE CALCULATION
  OF ROCKING CURVES USING THE TAKAGI TAUPIN EQUATIONS. DATA
  IS OUTPUT TO THE FILE 'INPDAT' TO BE USED BY THE PROGRAM
  CRUNCH WHICH CALCULATES THE ROCKING CURVE

  (C) M.J. HILL 1984/5 }

program INPUT_ROCK(input, output);

const
  PI = 3.1415927;

type
  material = (Im, Ga, As, P, Al, Si);
  mat_data = array [material] of real;
  matset = set of material;
  complex = record
              re, im: real;
            end;
  parameters = record
                 napuc: integer;
                 coord: array [1 .. 8, 1 .. 3] of real;
               end;
  comstring = array [1 .. 10] of string(25);
  comnumber = array [1 .. 10] of real;
  comset = array [1 .. 10] of set of material;
  matstring = array [material] of string(25);

const
  com_name = comstring('InP', 'GaAs', 'GaInAs', 'GaInAsP',
                       'GaAlAs', 'AlAs', 'Si');
  com_param = comnumber(5.8688E-10, 5.6535E-10, 0.0, 0.0, 0.0,
                        5.6535E-10, 5.4307E-10);
  com_type = comset([Im, P], [Ga, As], [Im, Ga, As],
                    [Im, Ga, As, P], [Ga, As, Al], [As, Al],
                    [Si]);
  mat_name = matstring('In', 'Ga', 'As', 'P', 'Al', 'Si');

var
  substrate: string(25);
  first: string(25);
  subcomp: matset;
  firstcomp: matset;
  layer_mat: array [1 .. 200] of matset;
  msource, msink, output_file: text;
  wavelength: real;
  no_layers: integer;
  fh1, fh2, fo: complex;
  ffh1, ffh2, ffo: complex;
  lfh1, lfh2, lfo: array [1 .. 200] of complex;
  start, finish, alpha_step: real;
  bstart, bfin, beta_step: real;
  stheta, cphi, d_sub: real;
  ftheta, fphi, d_first: real;
  h, k, l: integer;
  ii, jj, kk: integer;
  fh, fk, fl, fii, fjj, fkk: integer;
  thick, c: array [1 .. 200] of real;
  data: array [material] of parameters;
  percent: mat_data;
  div_diff: array [material, 1 .. 2, 1 .. 5] of real;
  pnts: array [1 .. 5] of real;
  query: string(25);
  polar: string(25);
  first_diff: boolean;
  approx: array [material] of record
                                 a: array [1 .. 4] of real;
                                 b: array [1 .. 4] of real;
                                 c: real;
                                end;
(* procedure to calculate lattice parameter and percentages of *)
(* each element in the composition *)


procedure lattice_param(composition: matset; mismatch: real;
                         var d: real);

  var
    i: integer;
    eg : real;


  procedure initialise;

    var
      element: material;

    begin
      for element := Im to Si do begin
        percent[element] := 1.0;
        end;
      end {initialise};


  begin {lattice_param}
    if composition = [Im, Ga, As]
      then begin
        percent[Im] := (5.8688 * mismatch + 0.2153) / 0.405;
        percent[Ga] := 1.0 - percent[Im];
        percent[As] := 1.0;
        d := d_sub * (1.0 + 2.0 * mismatch);
        end
      else
        if composition = [Ga, As, Al]
          then begin
            percent[Al] := 5.6535 * mismatch / 0.0065;
            percent[Ga] := 1.0 - percent[Al];
            percent[As] := 1.0;
            d := d_sub * (1.0 + 2.0 * mismatch);
            end
          else
            if composition = [Im, Ga, As, P]
              then begin
                eg := 1.0E-06;
                eg := 6.6E-34*3.0E8/1.6E-19/eg;
                percent[As] := (0.72 - sqrt(sqr(0.72)-4.0*0.12*
                                (1.35-eg))) / 0.24;
                percent[Ga] := mismatch * 5.8688 - percent[As] *
                                (6.0590 - 5.8688);
                percent[Ga] := percent[Ga] / (percent[As]
                                * 5.6535
                                + (1 - percent[As]) * 5.4512
                                - percent[As] * 6.0590
                                - (1 - percent[As]) * 5.8688);
                    percent[Im] := 1.0 - percent[Ga];
                    percent[P]  := 1.0 - percent[As];
                    d := d_sub * (1.0 + 2.0 * mismatch);
                    end
               else begin
                 for i := 1 to 7 do begin
                   if composition = com_type[i]
                     then
                       d := com_param[i];
                   end;
                 initialise;
                 end;
          end {lattice_param};
(* procedure to calculate structure factors for any material *)


procedure structuref(composition: matset; ds: real; ah, ak, al:
                     integer; var ft1, ft2, fto: complex);

  var
    element: material;
    f1, f2, f: complex;

(* procedure to obtain the scattering factor from the *)
(* analytical approximation, table is in file ANALAPP.DAT *)

  procedure scatt_fact(index: real; element: material;
                       var f: real);

    var
      i: integer;

    begin
      f := 0.0;
      for i := 1 to 4 do begin
        f := f + approx[element].a[i] *
                exp(- approx[element].b[i] * sqr(index));
        end;
      f := f + approx[element].c;
      end {scatt_fact};
(* procedure to calculate absorption and dispersion *)
(* corrections *)


  procedure corrections(element: material; var df1, df2: real);

    var
      lamda, mult: real;
      fcorr: array [1 .. 2] of real;
      i, j: integer;

    begin
      lamda := wavelength * 1.0E10;
      for i := 1 to 2 do begin
        mult := 1.0;
        fcorr[i] := 0.0;
        for j := 1 to 5 do begin
          fcorr[i] := fcorr[i] + mult * div_diff[element, i, j];
          mult := mult * (lamda - pnts[j]);
          end;
        end;
      df1 := fcorr[1];
      df2 := fcorr[2];
      end {corrections};
(* procedure to calculate structure factors for an element *)
(* corresponding to the h, k, l; -h, -k, -l & 0, 0, 0 *)
(* reflections *)


  procedure sfact(element: material; ds: real;
                  var f0, f1, f2: complex);

    var
      index, f: real;
      df1, df2: real;


    procedure eval(hh, kk, ll: integer; var ff: complex);

      var
        temp: complex;
        i: integer;
        position: real;

      begin
        temp.re := 0.0;
        temp.im := 0.0;
        for i := 1 to data[element].napuc do begin
          position := hh * data[element].coord[i, 1] +
                      kk * data[element].coord[i, 2] +
                      ll * data[element].coord[i, 3];
          temp.re := temp.re + cos(2.0 * PI * position);
          temp.im := temp.im + sin(2.0 * PI * position);
          end;
        ff.re := temp.re * f - temp.im * df2;
        ff.im := temp.re * df2 + temp.im * f;
        end {eval};


    begin {sfact}
      index := sqrt(float(sqr(ah) + sqr(ak) + sqr(al))) /
               (ds * 1.0E10 * 2.0);
      scatt_fact(index, element, f);
      corrections(element, df1, df2);
      f := f + df1;
      eval(ah, ak, al, f1);
      eval(- ah, - al, - ak, f2);
      index := 0.0;
      scatt_fact(index, element, f);
      f := f + df1;
      eval(0, 0, 0, f0);
      end {sfact};


  begin {structuref}
    ft1.re := 0.0;
    ft1.im := 0.0;
    ft2.re := 0.0;
    ft2.im := 0.0;
    fto.re := 0.0;
    fto.im := 0.0;
    for element := Im to Si do begin
      if element IN composition
        then begin
          sfact(element, ds, f, f1, f2);
          ft1.re := ft1.re + f1.re * percent[element];
          ft1.im := ft1.im + f1.im * percent[element];
          ft2.re := ft2.re + f2.re * percent[element];
          ft2.im := ft2.im + f2.im * percent[element];
          fto.re := fto.re + f.re * percent[element];
          fto.im := fto.im + f.im * percent[element];
          end;
      end;
    end {structuref};
(* procedure to check material exists, if it doesn't rc=0, *)
(* else rc=1, code returns the number of elements in       *)
(* the composition *)


procedure check(inpdat: string(25); var rc, code: integer;
                var composition: matset);

  var
    j: integer;
    elem: material;

  begin
    rc := 1;
    code := 0;
    for j := 1 to 7 do begin
      if inpdat = com_name[j]
        then begin
          composition := com_type[j];
          rc := 0;
          end;
      end;
    if rc = 0
      then begin
        for elem := Im to Si do begin
          if elem IN composition
            then
              code := code + 1;
          end;
        end;
    end {decomp};
(* procedure to read analytic approximation data and *)
(* dispersion and absorption correction data *)


procedure readparam;

  var
    element: material;
    indata: string(25);
    i, j, m: integer;
(* procedure to convert string input to element type *)
```

```
  procedure decode(inpdat: string(25); var element: material);

    var
      elem: material;

    begin
      for elem := Im to Si do begin
        if inpdat = mat_name[elem]
          then
            element := elem;
        end;
      end {decode};


  begin {readparam}
    reset(input, 'file=PHK0:ANALAPP.DAT');
    writeln(msink, 'Reading scattering factor data for',
                   ' elements..wait');
    for i := 1 to 5 do begin
      read(input, pnts[i]);
      end;
    read(input, indata);
    repeat
      decode(indata, element);
      for i := 1 to 4 do begin
        read(input, approx[element].a[i], approx[element].b[i]);
        end;
      read(input, approx[element].c);
      for i := 1 to 2 do begin
        for j := 1 to 5 do begin
          read(input, div_diff[element, i, j]);
          end;
        end;
      read(input, data[element].napuc);
      for i := 1 to data[element].napuc do begin
        read(input, data[element].coord[i, 1],
                    data[element].coord[i, 2],
                    data[element].coord[i, 3]);
        end;
      read(input, indata);
      until indata = 'end';
    end {readparam};
(* procedure to list material types available for InP *)
(* substrates *)


procedure IPchoice(var name: string(25));

  begin
    writeln(msink, 'Please choose a layer material from the ',
                   'following types');
    writeln(msink);
    writeln(msink, '       InP');
    writeln(msink, '       GaInAs');
    writeln(msink, '       GaInAsP');
    writeln(msink);
    writeln(msink, 'Layer material ? ');
    readln(msource, name);
    end {IPchoice};
(* procedure to list material types for GaAs substrates *)


procedure GAchoice(var name: string(25));

  begin
    writeln(msink, 'Please choose a layer material from the ',
                   'following types');
    writeln(msink);
    writeln(msink, '      GaAs');
    writeln(msink, '      AlAs');
    writeln(msink, '      GaAlAs');
    writeln(msink);
    writeln(msink, 'Layer material ? ');
    readln(msource, name);
    end {GAchoice};
(* procedure to calculate layer parameters for a linear *)
(* or graded lattice parameter variation *)


procedure linear(curved : boolean);

  var
    total_thick: real;
    curve, xcoord, ycoord, origin, radius : real;
    i, rcode, mcode: integer;
    mism_int, mism_sur: real;
    layer_name: string(25);

  begin
    page(msink);
    writeln(msink, 'GRADED LAYER PARAMETER INPUT');
    repeat
      if subcomp = [Im, P]
        then
          IPchoice(layer_name)
        else
          GAchoice(layer_name);
      check(layer_name, rcode, mcode, layer_mat[1]);
      if rcode <> 0
        then
          writeln(msink, 'I don''t know this material - ',
                         'please choose again');
      if mcode < 2
        then
          writeln(msink, 'Can''t have linear composition',
                         ' variation with binary composition',
                         ' - choose again');
      until (rcode = 0) AND (mcode > 2);
     repeat
      writeln(msink, 'Enter mismatch at layer/substrate ',
                     'interface (ppm)');
      readln(msource, mism_int);
      writeln(msink, 'Enter mismatch at surface of layer',
                     ' (ppm)');
      readln(msource, mism_sur);
      mism_sur := mism_sur * 1.0E-6;
      mism_int := mism_int * 1.0E-6;
      if mism_sur = mism_int
```

```
        then
          writeln(msink, 'No variation possible - choose again');
      until mism_sur <> mism_int;
    if curved
      then begin
        writeln(msink, 'Degree of non-linearity (0-100)');
        readln(msource, curve);
        end
      else
        curve := 0.0;
    repeat
      writeln(msink, 'How many laminae required');
      readln(msource, no_layers);
      if no_layers <= 0
        then
          writeln(msink, 'Impossible number of laminae - ',
                         'try again');
      if no_layers = 1
        then
          writeln(msink, 'Variation with one laminae ?',
                         ' - try again');
      until no_layers > 1;
    writeln(msink, 'Total thickness of epitaxial layer ',
                  '(microns)');
    readln(msource, total_thick);
    total_thick := total_thick * 1.0E-6;
    thick[1] := total_thick / float(no_layers);
    if curve > 0.05
      then begin
        curve := (1.0/sqrt(2.0) - 0.5) * curve / 100.0 + 0.5;
        origin := total_thick * (0.5 - sqr(curve)) /
                  (1.0 - 2.0 * curve);
        radius := 2.0 * sqr(curve * total_thick - origin);
        for i := 1 to no_layers do begin
          xcoord := total_thick * (no_layers - i) /
                    (no_layers - 1);
          ycoord := sqrt(radius - sqr(xcoord - origin)) + origin;
          c[i] := ycoord * (mism_sur - mism_int) / total_thick +
                  mism_int;
          layer_mat[i] := layer_mat[1];
          thick[i] := thick[1];
          end
        end
      else begin
        xcoord := (mism_sur - mism_int) / float(no_layers - 1);
        c[1] := mism_int;
        for i := 2 to no_layers do begin
          mism_int := mism_int + xcoord;
          c[i] := mism_int;
          layer_mat[i] := layer_mat[1];
          thick[i] := total_thick / float(no_layers);
          end
        end;
    end {linear};
(* procedure to calculate multi-layer parameters *)


procedure multilayer;

  var
  code: array [1 .. 2] of integer;
  i, j, no_mono1, no_mono2, rcode: integer;
  mism1, mism2: real;
  cap: string(10);
  layer_names: array [1 .. 2] of string(25);

begin
  page(msink);
  writeln(msink, 'MULTI LAYER SYSTEM CALCULATION');
  writeln(msink);
  writeln(msink, 'For first layer of repeat pair');
  repeat
    if subcomp = [Im, P]
      then
        IPchoice(layer_names[1])
      else
        GAchoice(layer_names[1]);
    check(layer_names[1], rcode, code[1], layer_mat[1]);
    if rcode <> 0
      then
        writeln(msink, 'I don''t know this material - ',
                       'please choose again');
    until rcode = 0;
  if (code[1] > 2)
    then begin
      writeln(msink);
      writeln(msink, 'Enter the mismatch for this layer',
                     ' (ppm)');
      readln(msource, mism1);
      mism1 := mism1 * 1.0E-6;
      end
    else
      mism1 := 0.0;
  writeln(msink);
  writeln(msink, 'and for the second layer of the repeat',
                 ' pair');
  repeat
    repeat
      if subcomp = [Im, P]
        then
          IPchoice(layer_names[2])
        else
          GAchoice(layer_names[2]);
      check(layer_names[2], rcode, code[2], layer_mat[2]);
      if rcode <> 0
        then
          writeln(msink, 'I don''t know this material - ',
                         'please choose again');
      until rcode = 0;
      if (code[2] > 2)
        then begin
          writeln(msink);
          writeln(msink, 'Enter the mismatch for this ',
                         'material (ppm)');
          readln(msource, mism2);
          mism2 := mism2 * 1.0E-6;
          end
        else
          mism2 := 0.0;
    if (layer_names[1] = layer_names[2]) AND (mism1 = mism2)
    then begin
      writeln(msink, 'You have the same material for both ',
                     'layers with the same composition');
      writeln(msink, 'Please choose the second layer again');
      end;
  until (layer_names[1] <> layer_names[2]) OR
        ((layer_names[1] = layer_names[2]) AND
        (mism1 <> mism2));
writeln(msink, 'Enter the number of monolayers in the',
               ' first layer');
readln(msource, no_mono1);
writeln(msink, 'and the number of monolayers in the second',
               ' layer');
readln(msource, no_mono2);
writeln(msink, 'Enter the total number of repeat units',
               ' required');
writeln(msink, 'ie half the total number of layers');
readln(msource, no_layers);
lattice_param(layer_mat[1], mism1, c[1]);
lattice_param(layer_mat[2], mism2, c[2]);
thick[1] := float(no_mono1) * c[1] / 2.0;
thick[2] := float(no_mono2) * c[2] / 2.0;
writeln(msink, 'Thickness of first layer  = ', thick[1]);
writeln(msink, 'Thickness of second layer = ', thick[2]);
writeln(msink);
writeln(msink, 'Total layer thickness = ', (thick[1] +
                thick[2]) * no_layers);
writeln(msink);
writeln(msink, 'Do you require a capping layer (Y/N)');
readln(msource, cap);
if (cap[1] = 'Y') OR (cap[1] = 'y')
  then begin
    i := no_layers * 2 + 1;
    writeln(msink, 'Enter material for capping layer');
    repeat
      if subcomp = [Im,P]
        then IPchoice(layer_names[1])
        else GAchoice(layer_names[1]);
      check(layer_names[1], rcode, code[1], layer_mat[i]);
      if rcode <> 0
        then
          writeln(msink, 'Unknown material - choose again');
      until rcode = 0;
    writeln(msink, 'Enter thickness of capping layer ',
                   '(microns)');
    readln(msource, thick[i]);
    thick[i] := thick[i] * 1.0E-6;
    if code[1] > 2
      then begin
        writeln(msink, 'Enter capping layer mismatch (ppm)');
        readln(msource, c[i]);
        c[i] := c[i] * 1.0E-6;
        end
      else
        c[i] := 0.0;
    end;
writeln(msink, 'OK - calculating multi layer parameters');
j := 1;
for i := 1 to no_layers do begin
  thick[j] := thick[1];
```

```
      thick[j + 1] := thick[2];
      c[j] := mism1;
      c[j + 1] := mism2;
      layer_mat[j] := layer_mat[1];
      layer_mat[j + 1] := layer_mat[2];
      j := j + 2;
      end;
    no_layers := no_layers * 2;
    if (cap[1] = 'Y') OR (cap[1] = 'y')
      then
        no_layers := no_layers + 1;
    end {multilayer};
(* procedure to input layer parameters *)


procedure single_in(num : integer);

  var
    mism: real;
    rcode, mcode: integer;
    name: string(25);

  begin
    repeat
      readln(msource, name);
      check(name, rcode, mcode, layer_mat[num]);
      if rcode <> 0
        then
          writeln(msink, 'Sorry don''t know this material - ',
                         'choose again');
      until rcode = 0;
    if mcode > 2
      then begin
        writeln(msink, 'Enter mismatch (ppm)');
        readln(msource, mism);
        mism := mism * 1.0E-6;
        end
      else
        mism := 0.0;
    writeln(msink, 'Enter thickness of layer (microns)');
    readln(msource, thick[num]);
    thick[num] := thick[num] * 1.0E-6;
    c[num] := mism;
    end {single_in};
(* procedure to input a single layer *)


procedure single;

  begin
    page(msink);
    writeln(msink, 'SINGLE LAYER COMPOSITION');
    if subcomp = [Im,P]
      then begin
        writeln(msink);
        writeln(msink, 'Enter the layer material - either ',
                       'GaInAs or GaInAsP');
        end
      else begin
        writeln(msink);
        writeln(msink, 'Enter the layer material - either ',
                       'GaAlAs or AlAs');
        end;
    single_in(1);
    no_layers := 1;
    end {single};
(* procedure to input multiple arbitrary layers *)

procedure multiple;

  var
    ls : integer;

  begin
    page(msink);
    writeln(msink, 'MULTIPLE LAYER INPUT');
    writeln(msink);
    writeln(msink, 'Enter the number of layers required');
    readln(msource, no_layers);
    if subcomp = [Im, P]
      then begin
        writeln(msink, 'Choose the layers from GaInAs, GaInAsP ',
                       'or InP only');
        writeln(msink);
        end
      else begin
        writeln(msink, 'Choose the layers from GaAlAs, GaAs or ',
                       'AlAs only');
        writeln(msink);
        end;
    for ls := 1 to no_layers do begin
      writeln;
      writeln(msink, 'Enter the layer material for layer ',
                     'number ',ls:3);
      single_in(ls);
      end;
    end {multiple};

(* procedure to read layer parameters from a data file *)

procedure read_layer;

  var
    filename : string(25);
    infile : text;
    js, ks : integer;

  begin
    writeln(msink);
    writeln(msink, 'Enter file name containing layer',
                   ' parameters');
    readln(msource, filename);
    filename := 'File=' || filename;
    reset(infile, filename);
    readln(infile, no_layers);
    for js := 1 to no_layers do begin
      readln(infile, thick[js], c[js], filename);
      for ks := 1 to 7 do begin
        if filename = com_name[ks]
          then layer_mat[js] := com_type[ks];
        end;
      end;
    close(infile);
    end;

(* procedure to input type of layer variation required and *)
(* call the appropriate routine *)


procedure layerin;

  var
    choice: integer;

  begin
    repeat
      writeln(msink, 'The following layer types are available');
      writeln(msink);
      writeln(msink, '       Single layer....................1');
      writeln(msink, '       Linear variation................2');
      writeln(msink, '       Non-linear variation............3');
      writeln(msink, '       Multilayer structure............4');
      writeln(msink, '       Other variation.................5');
      writeln(msink, '       Read variation from file........6');
      writeln(msink);
      writeln(msink, 'Please enter your choice');
      readln(msource, choice);
      until (choice > 0) AND (choice < 7);
    case choice of
      1: single;
      2: linear(false);
      3: linear(true);
      4: multilayer;
      5: multiple;
      6: read_layer;
      end;
    end {layerin};
(* procedure to calculate layer structure factors *)


procedure layer_sfacts;

  var
    i: integer;

  begin
    page(msink);
    writeln(msink, 'Calculating layer structure factors');
    for i := 1 to no_layers do begin
      lattice_param(layer_mat[i], c[i], c[i]);
      structuref(layer_mat[i], c[i], h, k, l, lfh1[i],
                 lfh2[i], lfo[i]);
      end;
    end {layer_sfacts};
(* procedure to input substrate parameters etc *)


procedure subin;

  var
```

```
    rcode, mcode: integer;

  begin
    repeat
      writeln(msink, 'Enter material type for substrate of ',
                     'the second crystal');
      readln(msource, substrate);
      check(substrate, rcode, mcode, subcomp);
      if rcode <> 0
        then
          writeln(msink, 'Sorry I don''t know of this material');
      if mcode > 2
        then
          writeln(msink, 'You have chosen either a ternary or ',
                         'quaternary for the substrate !');
      if (rcode <> 0) OR (mcode > 2)
        then begin
          writeln(msink, 'Please enter the second crystal ',
                         'substrate again');
          writeln(msink, ' ');
          end;
      until (rcode = 0) AND (mcode < 3);
    writeln(msink, 'Enter h, k, l for reflection');
    readln(msource, h, k, l);
    writeln(msink, 'Enter i, j, k for surface of crystal');
    readln(msource, ii, jj, kk);
    writeln(msink, 'Enter wavelength (A)');
    readln(msource, wavelength);
    wavelength := wavelength * 1.0E-10;
    repeat
      writeln(msink, 'Polarization state : s - sigma; p - pi;',
                     'r - random');
      readln(msource, polar);
      until (polar[1] = 's') OR (polar[1] = 'p') OR
            (polar[1] = 'r');
    if polar[1] = 's'
      then
        polar := 'SIGMA'
      else
        if polar[1] = 'p'
          then
            polar := 'PI'
          else
            polar := 'RANDOM';
    writeln(msink, 'Range of angles for reflectivity relative',
                   ' to Bragg angle (secs)');
    readln(msource, start, finish);
    writeln(msink, 'Angular step between points (secs)');
    readln(msource, alpha_step);
    lattice_param(subcomp, 0, d_sub);
    structuref(subcomp, d_sub, h, k, l, fh1, fh2, fo);
    stheta := wavelength * sqrt(float(sqr(h) + sqr(k) + sqr(l)))
              / (d_sub * 2.0);
    cphi := sqrt(float(sqr(h * ii + k * jj + l * kk)) /
            float((sqr(h) + sqr(k) + sqr(l)) *
            (sqr(ii) + sqr(jj) + sqr(kk))));
    end {subin};

procedure firstin;
```

```
  var
    rcode, mcode: integer;

  begin
    repeat
      writeln(msink, 'Enter material type for first crystal');
      readln(msource, first);
      check(first, rcode, mcode, firstcomp);
      if rcode <> 0
        then
          writeln(msink, 'Sorry I don''t know of this material');
      if mcode > 2
        then
          writeln(msink, 'You have chosen either a ternary or',
                         ' quaternary for the first crystal !');
      if (rcode <> 0) OR (mcode > 2)
        then begin
          writeln(msink,'Please enter the first crystal again');
          writeln(msink);
          end;
      until (rcode = 0) AND (mcode < 3);
    writeln(msink,'Enter h, k, l for reflection');
    readln(msource, fh, fk, fl);
    writeln(msink,'Enter i, j, k for the surface');
    readln(msource, fii, fjj, fkk);
    lattice_param(firstcomp, 0, d_first);
    structuref(firstcomp, d_first, fh, fk, fl, ffh1, ffh2, ffo);
    ftheta := wavelength * sqrt(float(sqr(fh)+sqr(fk)+sqr(fl)))
              / (d_first * 2.0);
    fphi := sqrt(float(sqr(fh*fii + fk*fjj + fl*fkk)) /
            float((sqr(fh)+sqr(fk)+sqr(fl)) *
            (sqr(fii)+sqr(fjj)+sqr(fkk))));
    end {firstin};

(* procedure to output data to file INPDAT *)

procedure outdat;

  var
    m: integer;
    date, time : alfa;

  begin
    rewrite(output,'File=PHKO:INPDAT');
    writeln(msink);
    writeln(msink,'Writing data to file INPDAT ....',
                  ' please wait');
    datetime(date, time);
    writeln(date);
    writeln(time);
    if NOT(first_diff) then first := substrate;
    writeln(substrate);
    writeln(first);
    if first_diff
      then
        writeln('YES')
      else
        writeln('NO');
    writeln(wavelength);
    writeln(polar);
    writeln(start, finish);
    writeln(alpha_step);
    writeln(start, finish);
    writeln(alpha_step);
    writeln(stheta);
    writeln(cphi);
    writeln(fh1.re, ' ', fh1.im);
    writeln(fh2.re, ' ', fh2.im);
    writeln(fo.re, ' ', fo.im);
    writeln(d_sub);
    writeln(no_layers);
    writeln(h, k, l);
    writeln(ii, jj, kk);
    for m := 1 to no_layers do begin
      writeln(c[m]);
      writeln(thick[m]);
      writeln(lfh1[m].re, ' ', lfh1[m].im);
      writeln(lfh2[m].re, ' ', lfh2[m].im);
      writeln(lfo[m].re, ' ', lfo[m].im);
      end;
    if first_diff
      then begin
        writeln(ftheta);
        writeln(fphi);
        writeln(ffh1.re,' ',ffh1.im);
        writeln(ffh2.re,' ',ffh2.im);
        writeln(ffo.re, ' ',ffo.im);
        writeln(d_first);
        end;
    end {outdat};

(* procedure to write layer parameters to a file *)

procedure write_layer;

  var
    filename : string(25);
    outfile : text;
    js, ks : integer;

  begin
    writeln(msink);
    writeln(msink, 'Enter file name for layer parameters');
    readln(msource, filename);
    filename := 'File=' || filename;
    rewrite(outfile, filename);
    writeln(outfile, no_layers);
    for js := 1 to no_layers do begin
      for ks := 1 to 7 do begin
        if layer_mat[js] = com_type[ks]
          then filename := com_name[ks];
        end;
      writeln(outfile, thick[js], c[js], filename);
      end;
    close(outfile);
    end {write_layer};

(* procedure to edit layer parameters *)
procedure edit;
```

```
var
  instring: string(25);
  index, ascii : integer;
  comchar: char;
  number : array [1 .. 3] of real;

procedure command(var ks: integer; stn: string(25);
                  var out: char);

  begin {command}
    repeat
      ks := ks + 1;
      out := stn[ks];
      until (out <> ' ') OR (ks = length(stn));
    end;

procedure substr(var ks: integer; const stn: string(25);
                 var stout : string(25));

   var
     ls : integer;
     out: char;

   begin {substr}
     repeat
       ks := ks + 1;
       out := stn[ks];
       until (out <> ' ') OR (ks = length(stn));
     ls := 1;
     stout := '         ';
     repeat
       stout[ls] := out;
       ls := ls + 1;
       ks := ks + 1;
       if ks <= length(stn) then out := stn[ks];
       until (out = ' ') OR (out = ',') OR (ks > length(stn));
     end {substr};

procedure num_in(n : integer);

  var
    outstring : string(25);
    js : integer;

  begin
    for js := 1 to n do begin
      if index < length(instring)
        then begin
          substr(index, instring, outstring);
          readstr(outstring, number[js]);
          end
        else number[js] := -999;
      end;
    end {num_in};

procedure print(num : integer);

  var
    js : integer;
    name : string(25);
  begin
    name := '                    ';
    for js := 1 to 7 do begin
      if layer_mat[num] = com_type[js]
        then name := com_name[js];
      end;
    writeln('      ',num:3,'          ',thick[num]*1.0E6:8:4,
            '               ',c[num]*1.0E6:8:4,
            '        ',name);
    end {print};

procedure print_layers;

  var
    js, n1, n2: integer;
    outstring : string(25);

  begin
    num_in(2);
    n1 := trunc(number[1]); n2 := trunc(number[2]);
    if n2 > no_layers then n2 := no_layers;
    if n2 = -999 then n2 := n1;
    if (n2 >= n1) AND (n1 <= no_layers)
      then begin
        writeln(' Layer number   thickness (microns)   ',
                ' mismatch (ppm)  material');
        writeln;
        for js := n1 to n2 do begin
          print(js);
          end
        end
      else writeln(' layer number range error');
    end;

procedure delete;

  var
    n1, js : integer;
    outstring : string(25);

  begin
    num_in(1);
    n1 := trunc(number[1]);
    if (n1 > 0) AND (n1 <= no_layers)
      then begin
        for js := n1 to (no_layers - 1) do begin
          thick[js] := thick[js+1];
          c[js] := c[js+1];
          layer_mat[js] := layer_mat[js+1];
          end;
        no_layers := no_layers - 1;
        writeln(' Layer number ',n1:3,' deleted');
        end
      else writeln(' Layer number out of range');
    end {delete};

procedure insert;

  var
```

```
    js, n1, rcode, mcode : integer;
    mat : matset;
    outstring : string(25);

  begin
    num_in(3);
    if index < length(instring)
      then begin
        substr(index, instring, outstring);
        check(outstring, rcode, mcode, mat);
        end;
    n1 := trunc(number[1]);
    if (n1 > 0) AND (number[2] <> -999) AND (number[3] <> -999)
       AND (rcode = 0) AND (n1 <= no_layers)
      then begin
        if (n1 < no_layers) then begin
          for js := no_layers downto (n1+1) do begin
            thick[js+1] := thick[js];
            c[js+1] := c[js];
            layer_mat[js+1] := layer_mat[js];
            end;
          end;
        thick[n1+1] := number[2] * 1.0E-6;
        c[n1+1] := number[3] * 1.0E-6;
        layer_mat[n1+1] := mat;
        no_layers := no_layers + 1;
        print(n1+1);
        end;
    end {insert};

procedure mismatch;

  var
    n1 : integer;

  begin
    num_in(2);
    n1 := trunc(number[1]);
    if (n1 > 0) AND (n1 <= no_layers) AND (number[2] <> -999)
      then begin
        c[n1] := number[2] * 1.0E-6;
        writeln;
        print(n1);
        end
      else
        writeln(' Input command error');
    end {mismatch};

procedure thickness;

  var
    n1 : integer;

  begin
    num_in(2);
    n1 := trunc(number[1]);
    if (n1 > 0) AND (n1 <= no_layers) AND (number[2] <> -999)
      then begin
        thick[n1] := number[2] * 1.0E-6;
        print(n1);
        end
      else
        writeln(' Input command error');
    end {thickness};

procedure comp_change;

  var
    n1, rcode, mcode : integer;
    outstring : string(25);
    mat : matset;

  begin
    num_in(1);
    n1 := trunc(number[1]);
    if index < length(instring)
      then begin
        substr(index, instring, outstring);
        check(outstring, rcode, mcode, mat);
        end;
    if rcode <> 0 then writeln(' Unkown material type - ',
                               'request ignored');
    if (n1 > 0) AND (n1 <= no_layers) AND (rcode = 0)
      then begin
        layer_mat[n1] := mat;
        print(n1);
        end
      else if rcode = 0 then writeln(' Layer number error');
    end {comp_change};

procedure help;

  begin
    writeln;
    writeln(' The following commands are available');
    writeln;
    writeln(' P <layer no> <layer no>   -  Prints layer ',
            'parameters starting');
    writeln('                              at the first layer',
            ' no. to the');
    writeln('                              second no. The ',
            'second no. is');
    writeln('                              optional.');
    writeln;
    writeln(' T <layer no> <number>     - Changes thickness of',
            ' layer no. to');
    writeln('                             <number>');
    writeln;
    writeln(' M <layer no> <number>     - Changes mismatch of',
            ' layer no. to');
    writeln('                             <number>');
    writeln;
    writeln(' C <layer no> <material>   - Changes material of',
            ' layer no. to');
    writeln('                             <material>');
    writeln;
    writeln(' D <layer no>              - Deletes data for ',
            'layer no.');
    writeln;
    writeln(' I <layer no> <number1> <number2> <material> - ',
              'Inserts layer');
      writeln;
      writeln(' E      -       Exit from editor mode');
      writeln;
      writeln(' Note - all input should be separated by one or',
              ' more spaces');
      writeln;
      end {help};

  procedure illegal;

    begin
      if (comchar <> 'E') AND (comchar <> 'e')
        then writeln(' command ?');
      end {illegal};

  begin {edit}
    reset(input, 'File=*MSOURCE*,Interactive');
    rewrite(output, 'File=*MSINK*, NOCC');
    page;
    writeln(' When prompted enter command  - H for help');
    writeln;
    repeat
      writeln('&*');
      readln(instring);
      index := 0;
      command(index, instring, comchar);
      ascii := ord(comchar);
      if (ascii > 192) then ascii := ascii -64;
      case ascii of
        151 : print_layers;
        163 : thickness;
        148 : mismatch;
        131 : comp_change;
        132 : delete;
        136 : help;
        137 : insert
        otherwise  illegal
        end;
      until (comchar = 'E') OR (comchar = 'e');
    end {edit};

(* main program *)

begin {INPUT_ROCK}
  reset(msource, 'File=*MSOURCE*,Interactive');
  rewrite(msink, 'File=*MSINK*');
  page(msink);
  readparam;
  page(msink);
  subin;
  layerin;
  page(msink);
  writeln(msink,'Do you want to edit the layer parameters ',
                '(Y/N)');
  readln(msource, query);
  if (query[1] = 'Y') OR (query[1] = 'y')
    then edit;
  writeln(msink);
  writeln(msink, 'Do you want to save the layer parameters ',
               '(Y/N)');
readln(msource, query);
if (query[1] = 'Y') OR (query[1] = 'y')
  then write_layer;
layer_sfacts;
writeln(msink, 'Do you want a different first crystal (Y/N)');
readln(msource, query);
if (query[1] = 'Y') OR (query[1] = 'y')
  then begin
    first_diff := TRUE;
    firstin;
    end
  else
    first_diff := FALSE;
outdat;
writeln(msink);
end;
```

```
C
C ***************************************************************
C * THIS PROGRAM DOES THE NUMBER CRUNCHING FOR ROCKING CURVES *
C *           USING THE DATA FROM FILE INPDAT                  *
C *   WHICH IS GENERATED BY THE PASCAL PROGRAM INPUT.ROCK.P    *
C ***************************************************************
C
      REAL ALPHA(2000), FSUB(2000), THETA, PHI, PHIF, PI
      REAL CLAYR(200), THICK(200), FLAY(2000), FSIG(2000)
      REAL BETA(2000), FPI(2000)
      INTEGER H, K, L, II, JJ, KK, PASS
      COMPLEX FH1, FH2, FO, ZSUB(2000), FFH1, FFH2, FFO
      COMPLEX LFH1(200), LFH2(200), LFO(200)
      CHARACTER*10 POLAR, SUBST, FIRST, DIFF
      COMMON /AREA1/ ALPHA, ALSTEP, START, FINISH, NOPNTS, WAVE,
     1       POLAR, SUBST, FIRST, DIFF
      COMMON /AREA2/ FH1, FH2, FO, DSUB, THETA, PHI
      COMMON /AREA3/ FFH1, FFH2, FFO, DFIRST, THETF, PHIF
      COMMON /AREA4/ BETA, BSTEP, BSTART, BFIN
      COMMON /AREA5/ LFH1, LFH2, LFO, CLAYR, THICK, NOLAYS,
     1       H, K, L, II, JJ, KK
C
C  Read data from file INPDAT
C
      CALL READAT
C
C  No of data points in single crystal reflectivity curves
C
      NOPNTS = INT((FINISH - START)/ALSTEP + 1.5)
      PASS = 1
      WRITE (6,10)
   10 FORMAT ('&Enter additional tilt angle (degrees) ? ')
      READ (5,*) TILT
      TILT = TILT * 3.142 / 180.
C
C  Calculate substrate reflectivity curve
C
   20 CALL SUBREF(FH1, FH2, FO, DSUB, THETA, PHI, FSUB, ZSUB,
     1             TILT)
C
C  Calculate layer reflectivity curve
C
      CALL LAYREF(FSUB, ZSUB, FLAY, TILT)
C
C  If first crystal <> substrate calculate its reflectivity
C
      IF (DIFF(1:1) .EQ. 'Y') CALL SUBREF(FFH1, FFH2, FFO,
     1    DFIRST, THETF, PHIF, FSUB, ZSUB, 0.)
C
C  Convolute reflectivity of first and second crystals
C
      IF (PASS .EQ. 1) CALL CONVOL(FSUB, FLAY, FSIG)
      IF (PASS .EQ. 2) CALL CONVOL(FSUB, FLAY, FPI)
C
C  If random polarisation redo with pi polarisation
C
      IF ((POLAR(1:1) .NE. 'R') .AND. (PASS .EQ. 1)) GO TO 50
      IF (PASS .EQ. 2) GO TO 30
      POLAR = 'PI'
      PASS = 2
      GO TO 20
C
C  Add sigma and pi polarisations if random
C
   30 DO 40 I = 1, NOPNTS
        FSIG(I) = FSIG(I) + FPI(I)
   40 CONTINUE
C
C  Write data to output file
C
   50 WRITE (1,*) NOPNTS
      WRITE (1,*) BSTEP
      WRITE (1,*) BETA(1)
      DO 60 I = 1, NOPNTS
        WRITE (1,*) FSIG(I)
   60 CONTINUE
      STOP
      END
C
C ********************************************************
C *   SUBREF - CALCULATES INFINITE CRYSTAL REFLECTIVITY   *
C *    OVER A RANGE OF ANGLES RELATIVE TO BRAGG ANGLE     *
C ********************************************************
C
      SUBROUTINE SUBREF(FH1, FH2, FO, DS, THETA, PHI, FSUB,
     1   ZSUB, TILT)
      REAL ALPHA(2000), FSUB(2000), THETA, PHI, DS, PI
      COMPLEX ZSUB(2000), FH1, FH2, FO, CHIH1, CHIH2, CHIO
      COMPLEX CSQRT, CABS, SQ, B, CH1, CH2
      CHARACTER*10 POLAR, SUBST, FIRST, DIFF
      COMMON /AREA1/ ALPHA, ALSTEP, START, FINISH, NOPNTS, WAVE,
     1       POLAR, SUBST, FIRST, DIFF
      WRITE (6,10)
   10 FORMAT (' CALCULATING SUBSTRATE REFLECTIVITY')
      PI = 3.1415927
      CER = 2.817E-15
      VUC = DS ** 3
      FFACT = -(WAVE**2*CER) / (VUC*PI)
      CHIH1 = FH1 * FFACT
      CHIH2 = FH2 * FFACT
      CHIO = FO * FFACT
      GAMMAO = SIN(THETA - PHI - TILT)
      GAMMAH = SIN(THETA + PHI + TILT)
      CHIH1 = CHIH1 / GAMMAH
      CHIH2 = CHIH2 / GAMMAO
      ALPHA(1) = START
      WRITE (6,20) THETA, PHI
   20 FORMAT (' THETA = ', F10.5, ' PHI = ', F10.5, ' RADIANS')
      DO 40 I = 1, NOPNTS
        THS = ALPHA(I) * PI / (3600.0*180.0)
        CH1 = CHIH1
        CH2 = CHIH2
        IF (POLAR(1:1) .NE. 'P') GO TO 30
        CH1 = CHIH1 * COS(2.0*(THETA + THS))
        CH2 = CHIH2 * COS(2.0*(THETA + THS))
   30   CONTINUE
        B = (CHIO/GAMMAO + CHIO/GAMMAH + 2.0*THS*SIN(2.0*THETA)/
     1       GAMMAH) / 2.0
        SQ = CSQRT(B**2 - CH1*CH2)
        ZSUB(I) = -(B + SQ*SIGN(1.,AIMAG(SQ))) / CH2
        FSUB(I) = (CABS(ZSUB(I))) ** 2
        ALPHA(I + 1) = ALPHA(I) + ALSTEP
  40  CONTINUE
      RETURN
      END
C
C ***************************************************
C *  LAYREF - CALCULATES LAYER REFLECTIVITIES USING  *
C *  REFLECTIVITIES FROM SUBSTRATE AS STARTING POINT *
C ***************************************************
C
      SUBROUTINE LAYREF(FSUB, ZSUB, FLAY, TILT)
      REAL ALPHA(2000), FSUB(2000), CLAYR(200), THICK(200), PHI
      REAL FLAY(2000), PI
      INTEGER H, K, L, II, JJ, KK
      COMPLEX ZSUB(2000), LFH1(200), LFH2(200), LFO(200)
      COMPLEX FH1, FH2, FO, B, SQ, NUM, DEN, CSQRT, CCOS, CSIN
      COMPLEX CHIH1, CHIH2, CHIO, CH1, CH2, CTN
      CHARACTER*10 POLAR, SUBST, FIRST, DIFF
      COMMON /AREA1/ ALPHA, ALSTEP, START, FINISH, NOPNTS, WAVE,
     1       POLAR, SUBST, FIRST, DIFF
      COMMON /AREA2/ FH1, FH2, FO, DSUB, THETA, PHI
      COMMON /AREA5/ LFH1, LFH2, LFO, CLAYR, THICK, NOLAYS,
     1       H, K, L, II, JJ, KK
      PI = 3.1415927
      CER = 2.817E-15
      WRITE (6,10)
  10  FORMAT (' CALCULATING LAYER REFLECTIVITY')
      AH = FLOAT(H)
      AK = FLOAT(K)
      AL = FLOAT(L)
      AII = FLOAT(II)
      AJJ = FLOAT(JJ)
      AKK = FLOAT(KK)
      DO 50 I = 1, NOLAYS
        VUC = DSUB ** 2 * CLAYR(I)
        FFACT = -(WAVE**2*CER) / (PI*VUC)
        CHIH1 = LFH1(I) * FFACT
        CHIH2 = LFH2(I) * FFACT
        CHIO = LFO(I) * FFACT
        DLAY = SQRT(1/(AH**2/DSUB**2 + AK**2/DSUB**2 +
     1         AL**2/CLAYR(I)**2))
        THETL = ARSIN(WAVE/(2.0*DLAY))
        PHIL = (AH*AII/DSUB**2 + AK*AJJ/DSUB**2 +
     1          AL*AKK/CLAYR(I)**2) ** 2
        PHIL = PHIL / ((AH**2/DSUB**2 + AK**2/DSUB**2 + AL**2/
     1  CLAYR(I)**2)*(AII**2/DSUB**2 + AJJ**2/DSUB**2 + AKK**2/
     1  CLAYR(I)**2))
        PHIL = SQRT(PHIL)
        PHIL = ARCOS(PHIL)
        GAMMAO = SIN(THETL - PHIL - TILT)
        GAMMAH = SIN(THETL + PHIL + TILT)
        D = PI / WAVE * SQRT(GAMMAH/GAMMAO)
        CHIH1 = CHIH1 / GAMMAH
        CHIH2 = CHIH2 / GAMMAO
        WRITE (6,20) I, THETL, PHIL
  20    FORMAT (' LAYER NO ', I3, '  THETA = ', F10.5, ' PHI = ',
     1          F10.5)
        DO 40 J = 1, NOPNTS
          THS = ALPHA(J) * PI / (3600.0*180.) + THETA - THETL +
     1          PHIL - PHI
          CH1 = CHIH1
          CH2 = CHIH2
          IF (POLAR(1:1) .NE. 'P') GO TO 30
          CH1 = CHIH1 * COS(2.0*(THETL + THS))
          CH2 = CHIH2 * COS(2.0*(THETL + THS))
   30     CONTINUE
          B = (CHIO/GAMMAH + CHIO/GAMMAO +
     1         2.0*THS*SIN(2.0*THETL)/GAMMAH) / 2.0
          SQ = CSQRT(B**2 - CH1*CH2)
          CTN = CSIN(SQ*D*(-THICK(I))) / CCOS(SQ*D*(-THICK(I)))
          NUM = ZSUB(J) * SQ + (0.0,1.0) * (CH1 + ZSUB(J)*B) * CTN
          DEN = SQ - (0.0,1.0) * (B + ZSUB(J)*CH2) * CTN
          ZSUB(J) = NUM / DEN
          FLAY(J) = CABS(ZSUB(J)) ** 2
   40   CONTINUE
   50 CONTINUE
      RETURN
      END
C
C ************************************************************
C * CONVOL - CALCULATES CONVOLUTION OF DATA SETS FSUB & FLAY *
C ************************************************************
C
      SUBROUTINE CONVOL(FSUB, FLAY, FCON)
      REAL FSUB(2000), FLAY(2000), BETA(2000), FCON(2000)
      REAL ALPHA(2000), F(2000)
      CHARACTER*10 POLAR, SUBST, FIRST, DIFF
      COMMON /AREA1/ ALPHA, ALSTEP, START, FINISH, NOPNTS, WAVE,
     1       POLAR, SUBST, FIRST, DIFF
      COMMON /AREA4/ BETA, BSTEP, BSTART, BFIN
      NPNTS = INT((BFIN - BSTART)/BSTEP + 1.5)
      WRITE (6,10)
   10 FORMAT (' CALCULATING CONVOLUTION')
      BETA(1) = BSTART
      DO 20 I = 1, NPNTS
        CALL MULT(FSUB, FLAY, BETA(I), F, NDP)
        FCON(I) = AINTG(ALPHA,F,NDP)
        BETA(I + 1) = BETA(I) + BSTEP
   20 CONTINUE
      RETURN
      END
C
C ***************************************************************
C * MULTIPLIES TWO DATA SETS, ONE OFFSET FROM THE OTHER BY BETA *
C ***************************************************************
C
      SUBROUTINE MULT(FSUB, FLAY, BETA, F, NDP)
      REAL FSUB(2000), FLAY(2000), F(2000), ALPHA(2000)
      CHARACTER*10 POLAR, SUBST, FIRST, DIFF
      COMMON /AREA1/ ALPHA, ALSTEP, START, FINISH, NOPNTS, WAVE,
     1       POLAR, SUBST, FIRST, DIFF
      NDP = 0
      N = NOPNTS - 1
      L = 1
      IF ((ALPHA(1) + BETA) .GT. ALPHA(NOPNTS)) RETURN
      IF ((ALPHA(NOPNTS) + BETA) .LT. ALPHA(1)) RETURN
      DO 50 I = 1, NOPNTS
        IF ((ALPHA(I) + BETA) .GT. ALPHA(NOPNTS)) GO TO 60
        IF ((ALPHA(I) + BETA) .LT. ALPHA(1)) GO TO 40
        DO 20 J = L, N
          H = ALPHA(I) + BETA - ALPHA(J)
          SH = ALPHA(J + 1) - ALPHA(J)
          IF (H .GT. SH) GO TO 10
          G = FLAY(J) * (1 - H/SH) + FLAY(J + 1) * H / SH
          GO TO 30
   10     CONTINUE
   20   CONTINUE
   30   NDP = NDP + 1
        F(NDP) = FSUB(I) * G
        L = J
   40   CONTINUE
   50 CONTINUE
   60 CONTINUE
      RETURN
      END
C
C *******************************************************
C * AINTG - CALCULATES AREA UNDER MULTIPLIED DATA SET F *
C *******************************************************
C
      FUNCTION AINTG(ALPHA, F, NDP)
      REAL ALPHA(2000), F(2000)
      N = NDP - 1
      AINTG = 0.0
      DO 10 I = 1, N
        H = ALPHA(I + 1) - ALPHA(I)
        AINTG = AINTG + (F(I) + F(I + 1)) * H / 2.0
   10 CONTINUE
      RETURN
      END
C
C ******************************************
C *  READAT - READS DATA FROM FILE INPDAT  *
C ******************************************
C
      SUBROUTINE READAT
      REAL ALPHA(2000), BETA(2000), FSUB(2000), PHI, PHIF
      REAL THICK(200), CLAYR(200)
      INTEGER H, K, L, II, JJ, KK
      COMPLEX ZSUB(2000), FH1, FH2, FO, FFH1, FFH2, FFO
      COMPLEX LFH1(200), LFH2(200), LFO(200)
      CHARACTER*10 POLAR, SUBST, FIRST, DIFF, DATE, TIME
      COMMON /AREA1/ ALPHA, ALSTEP, START, FINISH, NOPNTS, WAVE,
     1       POLAR, SUBST, FIRST, DIFF
      COMMON /AREA2/ FH1, FH2, FO, DSUB, THETA, PHI
      COMMON /AREA3/ FFH1, FFH2, FFO, DFIRST, THETF, PHIF
      COMMON /AREA4/ BETA, BSTEP, BSTART, BFIN
      COMMON /AREA5/ LFH1, LFH2, LFO, CLAYR, THICK, NOLAYS,
     1       H, K, L, II, JJ, KK
C
C  Read data from file INPDAT
C
      CALL FTNCMD('ASSIGN 3=INPDAT;')
      READ (3,10) DATE
      READ (3,10) TIME
      READ (3,10) SUBST
      READ (3,10) FIRST
      READ (3,10) DIFF
      READ (3,*) WAVE
      READ (3,10) POLAR
      READ (3,*) START, FINISH
      READ (3,*) ALSTEP
      READ (3,*) BSTART, BFIN
      READ (3,*) BSTEP
   10 FORMAT (' ', A10)
      WRITE (6,20) TIME, DATE
   20 FORMAT (' DATA GENERATED AT ', A10, ' ON ', A10)
      WRITE (6,30) FIRST, SUBST
   30 FORMAT (' First crystal is ', A10, '  Second crystal is ',
     1        A10)
      WRITE (6,40) WAVE
   40 FORMAT (' Wavelength = ', E12.5)
      WRITE (6,50) POLAR
   50 FORMAT (' Polarization state is ', A10)
      WRITE (1,*) DATE
      WRITE (1,*) TIME
      WRITE (1,*) SUBST
      WRITE (1,*) FIRST
      WRITE (1,*) WAVE
      WRITE (1,*) POLAR
C
C  Read substrate variables
C
      READ (3,*) THETA
      READ (3,*) PHI
      READ (3,*) FH1
      READ (3,*) FH2
      READ (3,*) FO
      READ (3,*) DSUB
      THETA = ARSIN(THETA)
      PHI = ARCOS(PHI)
      WRITE (1,*) THETA, PHI, DSUB
C
C  Read layer variables
C
      READ (3,*) NOLAYS
      READ (3,*) H, K, L
      READ (3,*) II, JJ, KK
      DO 60 I = 1, NOLAYS
        READ (3,*) CLAYR(I)
        READ (3,*) THICK(I)
        READ (3,*) LFH1(I)
        READ (3,*) LFH2(I)
        READ (3,*) LFO(I)
   60 CONTINUE
      WRITE (1,*) NOLAYS
      WRITE (1,*) H, K, L
      WRITE (1,*) II, JJ, KK
      DO 70 I = 1, NOLAYS
        WRITE (1,*) CLAYR(I), THICK(I)
   70 CONTINUE
C
C  Read first crystal variables if necessary
C
      IF (DIFF(1:1) .EQ. 'N') GO TO 90
      WRITE (6,80)
   80 FORMAT (' Reading first crystal parameters')
      READ (3,*) THETF
      READ (3,*) PHIF
      READ (3,*) FFH1
      READ (3,*) FFH2
      READ (3,*) FFO
      READ (3,*) DFIRST
      THETF = ARSIN(THETF)
      PHIF = ARCOS(PHIF)
   90 CONTINUE
      RETURN
      END
```

```
C  ****************************************************************
C  *                         PLOTTER                              *
C  *                                                              *
C  *  This programme plots the  reflectivity against angle        *
C  *  of incidence from the convoluted data. The effect of        *
C  *  curvature can be added by smoothing over a given no.        *
C  *  of data points.                                             *
C  *           THE *GHOST SYSTEM IS USED.                         *
C  ****************************************************************
C
      REAL MAX, MIN, REFS(2000), W(200), M(200), MM, MMIN
      REAL BETA(2000), WOL(200)
      LOGICAL*1 CHOICE(4), EQUC
      INTEGER FILE
      INTEGER PNOL, H
      DIMENSION SUBST(3), FIRST(3), POLAR(3), DATE(3), TIME(3)
C
C  Define constants used to determine scaling of graph
C
      WRITE (6,10)
   10 FORMAT (' Please wait ... reading data from disc file')
      FILE = 1
      RMAX = 0.
      RMIN = 1E20
      MAX = 0.
      MIN = 10000.
      MM = 0.
      TOL = 0.
      MMIN = 1000.
C
C Read data from disc file
C
      READ (FILE,20) DATE
      READ (FILE,20) TIME
   20 FORMAT (3A4)
      WRITE (6,30) TIME, DATE
   30 FORMAT (' DATA GENERATED AT ', 3A4, ' ON ', 3A4)
      READ (FILE,50) SUBST
      READ (FILE,50) FIRST
      READ (FILE,*) WAVE
      READ (FILE,50) POLAR
      READ (FILE,*) THETA, PHI, DSUB
      READ (FILE,*) NOLAYS
      READ (FILE,*) H, K, L
      READ (FILE,*) II, JJ, KK
      DO 40 I = 1, NOLAYS
        READ (FILE,*) M(I), W(I)
        M(I) = (M(I) - DSUB) / DSUB * 1.0E6
        WOL(I) = W(I) * 1.0E6
        TOL = TOL + WOL(I)
        IF (M(I) .GT. MM) MM = M(I)
        IF (M(I) .LT. MMIN) MMIN = M(I)
   40 CONTINUE
   50 FORMAT (3A4)
      READ (FILE,*) NOPNTS
      READ (FILE,*) BSTEP
      READ (FILE,*) BETA(1)
      DO 60 J = 1, NOPNTS
        READ (FILE,*) REFS(J)
        IF (BETA(J) .GT. MAX) MAX = BETA(J)
```

```
          IF (BETA(J) .LT. MIN) MIN = BETA(J)
          IF (REFS(J) .GT. RMAX) RMAX = REFS(J)
          IF (REFS(J) .LT. RMIN) RMIN = REFS(J)
          BETA(J + 1) = BETA(J) + BSTEP
   60 CONTINUE
C
C  Input of range of data to be plotted
C
        WRITE (6,70) BSTEP
   70 FORMAT (' The data has an angular step interval of ', F5.1,
     1        ' secs')
        WRITE (6,80)
   80 FORMAT (' Enter the change in angle along the incident',
     1        ' X-ray beam (secs)')
        READ (5,*) ASMTH
        ISMTH = INT(ASMTH/BSTEP)
        IF (ISMTH .LE. 1) GO TO 110
        RMAX = 0.
        RMIN = 1E20
        NOPNTS = NOPNTS - ISMTH
        DO 100 J = 1, NOPNTS
          REFL = 0.
          DO 90 I = 1, ISMTH
            REFL = REFL + REFS(I + J - 1)
   90     CONTINUE
          REFS(J) = REFL
          IF (REFS(J) .GT. RMAX) RMAX = REFS(J)
          IF (REFS(J) .LT. RMIN) RMIN = REFS(J)
  100 CONTINUE
  110 WRITE (6,120)
  120 FORMAT (' Do you require a logarithmic scale (Y/N)')
        READ (5,130) CHOICE
  130 FORMAT (4A1)
        IF ( .NOT. (EQUC(CHOICE(1),'Y'))) GO TO 150
        RMAX = 1E-20
        RMIN = 1E20
        DO 140 I = 1, NOPNTS
          REFS(I) = ALOG10(REFS(I))
          IF (REFS(I) .GT. RMAX) RMAX = REFS(I)
          IF (REFS(I) .LT. RMIN) RMIN = REFS(I)
  140 CONTINUE
  150 WRITE (6,160)
  160 FORMAT (' NORMALIZING DATA')
        DO 170 I = 1, NOPNTS
          REFS(I) = (REFS(I) - RMIN) / (RMAX - RMIN) * 100.
  170 CONTINUE
        RMAX = 100.
        RMIN = 0.
        WRITE (6,180)
  180 FORMAT (' Range of data to plot is:-')
        WRITE (6,190) MIN, MAX, RMIN, RMAX
  190 FORMAT (' Xmin=', F10.4, 2X, 'Xmax=', F9.4, 2X, 'Ymin=',
     1          F5.2, 'Ymax=', F10.4)
        WRITE (6,200)
  200 FORMAT (' Enter range to be plotted in X direction')
        WRITE (6,210)
  210 FORMAT ('&       from ? ')
        READ (5,*) MIN
        WRITE (6,220)
  220 FORMAT ('&         to ? ')
```

```
      READ (5,*) MAX
      WRITE (6,230)
 230  FORMAT (' Range to be plotted in Y direction')
      WRITE (6,240)
 240  FORMAT ('&      to ? ')
      READ (5,*) RMAX
C
C Plot defined region of data using *GHOST routines
C
C Information box first
C
      CALL PAPER(1)
      CALL BLKPEN
      CALL PSPACE(0.80, 1.0, 0., 0.56)
      CALL CSPACE(0.80, 1.0, 0., 0.56)
      CALL MAP(0., 1., 0., 1.)
      CALL CTRMAG(7)
      CALL BORDER
      CALL PLACE(0, 0)
      CALL CRLNFS(4)
      CALL TYPECS(' DATA GENERATED AT : ', 21)
      CALL TYPECS(TIME, 10)
      CALL CRLNFD
      CALL TYPECS('                   ON : ', 21)
      CALL TYPECS(DATE, 10)
      CALL CRLNFS(3)
      CALL TYPECS(' SECOND CRYSTAL : ', 19)
      CALL TYPECS(SUBST, 10)
      CALL CRLNFS(3)
      CALL TYPECS(' FIRST CRYSTAL : ', 17)
      CALL TYPECS(FIRST, 10)
      CALL CRLNFS(3)
      CALL TYPECS(' POLARIZATION : ', 16)
      CALL TYPECS(POLAR, 10)
      CALL CRLNFS(3)
      CALL TYPECS(' WAVELENGTH = ', 14)
      CALL TYPENE(WAVE, 5)
      CALL TYPECS(' M', 2)
      CALL CRLNFS(3)
      CALL TYPECS(' BRAGG ANGLE = ', 16)
      CALL TYPENE(THETA, 5)
      CALL CRLNFS(3)
      CALL TYPECS(' PHI = ', 7)
      CALL TYPENE(PHI, 5)
      CALL CRLNFS(3)
      CALL TYPECS(' REFLECTION = ', 14)
      CALL TYPENI(H, 2)
      CALL TYPENI(K, 2)
      CALL TYPENI(L, 2)
      CALL CRLNFS(3)
      CALL TYPECS(' SURFACE = ', 11)
      CALL TYPENI(II, 2)
      CALL TYPENI(JJ, 2)
      CALL TYPENI(KK, 2)
      CALL CRLNFS(3)
      CALL TYPECS(' LAYER THICKNESS = ', 19)
      CALL TYPENF(TOL, 5)
      CALL CTRSET(4)
      CALL TYPECS(' M', 2)
      CALL CTRSET(1)
      CALL TYPECS('M', 1)
      CALL CRLNFS(3)
      CALL TYPECS(' NO OF LAYERS : ', 16)
      CALL TYPENI(NOLAYS, 3)
      IF (ISMTH .LE. 1) GO TO 250
      CALL CRLNFS(3)
      CALL TYPECS(' CURVATURE ANGLE ', 17)
      ASMTH = BSTEP * ISMTH
      CALL TYPENF(ASMTH, 5)
      CALL TYPECS(' SECS', 5)
  250 CONTINUE
C
C Now plot the rocking curve and axes
C
C Labels first
C
      CALL PSPACE(.01, .74, 0.08, .5)
      CALL CSPACE(.0, 1., .0, 1.)
      CALL MAP(0., 1., 0., 1.)
      CALL CTRMAG(20)
      CALL CTRSET(3)
      CALL PLOTCS(0.70, 0.005, 'D', 1)
      CALL CTRSET(4)
      CALL TYPECS('J', 1)
      CALL CTRSET(2)
      CALL TYPECS('  (SECS)', 8)
      CALL CTRSET(1)
      CALL CTRORI(1.0)
      CALL PLOTCS(0.02, 0.35, 'REFLECTIVITY (%)', 16)
      CALL CTRORI(0.0)
      CALL PSPACE(.10, .74, 0.14, 0.5)
      CALL CSPACE(.00, .80, 0.00, 0.6)
      CALL MAP(MIN, MAX, RMIN, RMAX)
      CALL CTRMAG(12)
C
C Draw axes
C
      CALL MARK(MAX - MIN, SX)
      CALL MARK(RMAX, SY)
      CALL AXESSI(SX, SY)
C
C Plot the rocking curve
C
      CALL NSCURV(BETA, REFS, 1, NOPNTS)
C
C Draw graph of layer mismatch vs depth
C
      CALL PSPACE(0.80, 1.00, 0.00, 0.20)
      CALL THICK(1)
      CALL BLKPEN
      CALL CSPACE(.80, 1.0, .00, 0.2)
      CALL MAP(0., 0.2, 0., 0.4)
      CALL CTRMAG(6)
      CALL PLOTCS(0.05, 0.02, 'DEPTH BELOW SURFACE', 19)
      CALL CTRSET(4)
      CALL TYPECS(' M', 2)
      CALL CTRSET(1)
      CALL TYPECS('M', 1)
      CALL POSITN(0.01, 0.08)
      CALL CTRORI(1.0)
      CALL TYPECS('LATTICE MISMATCH (PPM)', 22)
      CALL CTRORI(0.)
      IF (MMIN .GT. 0.) MMIN = 0.
      CALL PSPACE(.85, .95, .04, .16)
      CALL MAP(0., TOL, MMIN, MM)
      CALL MARK(TOL*2., SX)
      CALL CTRMAG(3)
      CALL MARK((MM - MMIN)*2., SY)
      CALL AXESSI(SX, SY)
      CALL POSITN(0., M(NOLAYS))
      WW = WOL(NOLAYS)
      PNOL = NOLAYS
      IF (NOLAYS .EQ. 1) GO TO 270
      NL = NOLAYS - 1
      DO 260 L = 1, NL
        CALL JOIN(WW, M(PNOL))
        CALL JOIN(WW, M(PNOL - 1))
        WW = WW + WOL(PNOL - 1)
        PNOL = PNOL - 1
  260 CONTINUE
  270 CALL JOIN(WW, M(PNOL))
      CALL JOIN(WW, 0.)
C
C End plotting
C
      CALL GREND
      STOP
      END
C
C  routine to calculate tick lengths along axes
C
      SUBROUTINE MARK(RANGE, STEP)
      IB = IFIX(ALOG10(RANGE))
      A = RANGE / 10.0 ** IB
      IF ((A .GE. 1.) .AND. (A .LT. 3.))
     1    STEP = 2. * 10. ** (IB - 1)
      IF ((A .GE. 3.) .AND. (A .LT. 7.))
     1    STEP = 5. * 10. ** (IB - 1)
      IF ((A .GE. 7.) .AND. (A .LT. 10.))
     1    STEP = 1 * 10. ** IB
      RETURN
      END
C
C  routine to provide n crlnfds
C
      SUBROUTINE CRLNFS(N)
      DO 10 I = 1, N
        CALL CRLNFD
   10 CONTINUE
      RETURN
      END
```

# APPENDIX B

## PROGRAMMING THE MINICAM I.C.C. BOARD FOR ASCII SERIAL DATA

The software required to operate the two channel I.C.C. serial interface is described. A a full 6502 disassembler and a memory dump utility are also implemented. These follow, in principle, those described in 'Beyond Games: System Software for your 6502 Personal Computer' by Ken Skier. This new software plus the original, slightly modified, 'Minicon' software occupies about 3k of code and therefore requires two of the board's 2k, 2532 type, EPROMS.

The board control software, 'Minicon', occupies the EPROM located at $F000 to $FFFF, and also contains the 6502 reset routines. The disassembler and monitor are contained in the EPROM from $E000 to $EFFF. Since the monitor uses code located within the other EPROM a jump table is provided at $F000 onwards to enable these routines to be located even if their absolute location within the other EPROM is altered during development. The jump instructions within this table are guaranteed not to be altered, thus avoiding the need to reprogramme all the EPROMs if code in one is changed. This system is similar to the operating system jump table provided on the BBC microcomputer (OSWRCH, OSBYTE, OSWORD, OSFILE etc).

During RESET the stack is setup and interrupts disabled. The least significant five bits of the address switch on the main board are used to set the GPIB address register in the 68488. Once the two 6850's have been master reset, the word format switch on the serial board is read and the least significant four bits and most significant four bits used to program the second and first 6850's respectively. The clock divide ratio in both chips is set to 16, since the 4702 produces 16 times the standard baud rates. The top three bits of the address switch are then used to determine which mode of operation is required. The location of the required code is copied on to the zero page from the table RESTAB to enable an indirect jump to the code.

If interrupts are enabled by any code the 6502 is vectored to a piece of code that jumps to the location held in $00,$01. This location is initially set to the remainder of the interrupt handler in the EPROM $F000-$FFFF. However, if the user wishes to provide another interrupt handler the location held in $00,$01 can easily be altered.

Routines using the GPIB bus as the host interface set the flags IEEFLG, used to select the input device in the character input routine, and IEEEPR, used to select the output device during character output. Ascii modes also set the flag NEWFLG, which is used by the board control routines to determine whether the existing binary output routines or the new Ascii ones should be used. Additionally, the character output routine uses the flags HOST and PRINTR to determine whether output should go to the host or printer serial ports, and the flag IEEEPR for the GPIB port as output. Note that more than one of the flags HOST, PRINTR and IEEEPR can be set at once, enabling output to be sent to more than one device simultaneously.

The operating modes supported at present are:

Monitor - development mode with disassembler and memory dump options. Uses the host serial port or GPIB port.

Minicon - mode for control of modules, can be used either in binary or ascii. Ascii with host serial port or GPIB port, binary only with GPIB port.

GPIB to serial - takes data from the GPIB port and transmits it via the host serial port. Hardwire handshaking can be used to control the flow of data from the serial port. At present only unidirectional.

Serial to serial - takes data from the host serial port and transmits it via the printer serial port. No buffering is perfromed so the printer port should operate at baud rates greater than or equal to the host port baud rate.

## Monitor

This routine is designed to operate with a dumb terminal. Once the routine is entered all variables are initialised by a jump to INIT. The 6850 is set with RTS low, to prevent input from the terminal. The state of the flag IEEFLG is checked to determine whether the GPIB port or the host serial port has been selected. For GPIB operation the flags HOST and PRINTR are cleared and IEEEPR set, while for serial operation HOST is set and PRINTR, IEEEPR cleared. A start up message is then transmitted plus the prompt character, '*', which is useful to check that communication formats are correctly set. If serial control is selected the RTS line is then set high, allowing input.

Characters are received and stored in the input buffer located at $0300. If either a backspace or delete is detected action is taken to remove the last character from the buffer, provided that it is not already empty. The character received is echoed back to the terminal unless it was a delete or backspace with an empty buffer, when an ascii bell is echoed. Once a carriage return is received the input routine is exited and the first character checked for a valid command. The commands currently available are selected as below.

| Character | Variables | Command |
|---|---|---|
| D | Start, End Addresses | Disassemble from start address to end address |
| M | Start, End Addresses | Dump memory between start address and end address |
| E | none | Exit to Minicon |

| | | |
|---|---|---|
| P | none | Toggle serial printer flag on and off to provide hardcopy |
| H | none | Help - displays this information |

If a valid character is found the relevant routine is entered otherwise the message '? COMMAND' is transmitted followed by a jump back to the start of this routine. For those commands requiring input of variables, ie D and M, the routine STEASA is entered. Starting from the end of the input buffer each character, corresponding to a hexadecimal digit, is removed and converted to binary. If conversion to binary produces an error because the character was not a valid hexadecimal digit the message 'COMMAND ERROR' is transmitted and the routine terminated with the carry flag set. Once a valid binary value is found it added to the start or end address value as necessary, after mulitplying by the required factor of 16 (left shift four times). The two addresses should be separated by a space character. If STEASA is successful in obtaining two valid 16 bit addresses the carry flag is cleared and the routine terminated.

The disassembly and memory dump routines are then entered once valid start and end addresses have been found. Since the table driven disassembler and memory dump are similar to those described in Ken Skier's book they will not be described here. There are. however, some errors present in the tables contained within the book. Output from these routines is performed by the routine PRTCHR, which transmits output according to the previously described flags. The P command can be used to toggle the flag PRINTR on and off to enable hardcopy to be obtained on a serial printer attached to the printer port.

## Minicon

Once entered this routine selects the output devices in a similar manner to the monitor routine. Characters received are placed in the same input buffer but no interpretation of backspace or delete is made. No characters are echoed back to the terminal/microcomputer. This routine is designed, primarily, to operate with a microcomputer where echoing of the input would be unecessary. Once the currently selected input terminator byte is received the character entry routine is entered and the buffer decoded. Since the number of commands provided by this routine is far greater than those offered by the monitor a rather more sophisticated decoding process is implemented in order to save memory space. Commands are selected by a two letter code and a one number option, eg AD1, thus allowing up to 260 possible commands. The input data, if required, should then follow, separated from the command name by a comma with each number deliminated also by a comma (eg. ST1,32,1000,100).

The first character in the buffer is removed from the buffer and checked to be in the range 'A' to 'Z' by subtracting 65 and testing for the byte to be in the range >=0 and <26. If the byte is out of this range the message 'BAD COMMAND' is transmitted and the input section reentered. A valid byte is then used as an index to the table of valid second characters CHTAB. This character is compared with the second character in the input buffer and if they do not agree the 'BAD COMMAND' error message is returned as above. The third character is then removed from the buffer and checked for validity by subtracting 48 to give a range of 0 to 9. If it is non valid the error message 'BAD OPTION' is returned. The table JMPTAB contains the addresses minus one of the possible routines and is accessed by the following technique. The valid first character byte is used to obtain an offset number from the table OFFSET which gives the offset of the start of the addresses for a particular command from the start of JMPTAB. The byte obtained from the option number is multiplied by two and added to this offset thus pointing to the high byte of the required address. This value is fetched and pushed onto the stack. The preceeding low byte value is then also fetched and pushed onto the stack. An RTS instruction then fetches this address off the stack and sets the program counter to this address plus one which will be the start of the required code.

The majority of the routines accessed in this way then remove the required number of input parameters from the input buffer, converting from ascii decimal to binary, with commas as number delimiters. Conversion from decimal to binary is not so straightforward as from hexadecimal to binary. In order to avoid binary multiplication and increase the speed of the conversion a tabular method is used. The decimal number removed from the buffer is checked for validity. if it is not a '0' to '9' the error 'VARIABLE ERROR' is returned. A valid byte is then used to access tables of 10's, 100's, 1000's, 10000's which contain the binary value of 10,20...90; 100,200....900 etc. The values obtained from these tables are added to the units value to give the binary number. The numbers obtained are stored consequetively from address VA ($40) which correspond to the original Minicon input values. A jump to the original module handling routines is then made. These routines, slightly modified, are contained in the $F000 EPROM.

Some of the original Minicon routines have an overlap function, returning data to complete the communication process before actually operating the module, allowing the controlling computer to perform some other tasks while, for example, a stepper motor is driving a large number of steps. This option is implemented in the new software as another option number, with the message 'OK' being returned.

When a routine has finished the software informs the controlling computer by transmitting a message or value read from a module. The value read from the module is converted to ascii before transmission if the flag NEWFLG is set. When NEWFLG is clear the original binary GPIB output routine is used. For modules that are output devices the message 'OK' is returned, while for input modules that can generate errors, such as an overrange on an A to D module, a meaningful error message is transmitted. Possible error messages are 'OVERRANGE' for an A to D overrange or 'TIMED OUT' for either a faulty or non present A to D or an incomplete handshake on a GPIB (general purpose input board).

If a Minicon routine generates a negative value, such as an A to D, the routine sets the flag NEGFLG, informing the output routine that the value returned is in two's complement. The output routine REPLAN then takes the two's complement of this value before converting to ascii and transmits a minus sign before infront of the ascii coded value. Conversion to ascii is performed by using the well known double babble method. This routine, BINASC, returns with three bytes in the output buffer representing six BCD numbers. The output bytes are then converted to ascii, 4 bits at a time, by adding 48. The flag LEADZ is used to suppress the transmission of leading zeros, but if the output routine ends with LEADZ still clear the value is zero so one zero is transmitted.

In addition to the existing Minicon routines a number of new ones have been added, particularly those enabling use of the extra serial ports. These are detailed along with the other commands available below.

## MINICON COMMAND DESCRIPTIONS

| Command | Input | Description |
|---|---|---|
| AD1 | address | reads A to D number ‹address›, same as old usr(5). |
| AD2 | add., n1, n2 | operates 16 channel autoranging A to D, same as usr(10). |
| CO1 | address, n1 | counts for ‹n1› 100th's second using scalar module number ‹address›. Same as usr(2). |
| CO2 | address, n1 | same as CO1 but overlap option is implemented. |
| DA1 | address, n1 | outputs value ‹n1› to two channel D to A number ‹address›, same as usr(4) without 512 added to address. |
| DA2 | address, n1 | outputs value ‹n1› to 4 channel D to A number ‹address›, same as usr(4) with 512 added to address. |
| GP1 | address | inputs 16 bit value from general purpose input board (GPIB), same as usr(8). |
| MO1 | nothing | jumps to the binary version of Minicon for use via the GPIB interface. |
| MO2 | nothing | jumps to the new ascii monitor with the GPIB interface selected as the input and output port. |
| MO3 | nothing | jumps to the ascii version of Minicon with the GPIB interface seleced. |
| MO4 | nothing | jumps to the new ascii monitor with the host serial port selected. |
| MO5 | nothing | jumps to the GPIB to host serial port routine. |
| PR1 | string | Outputs the characters in the string via the printer serial port. The carriage return is transmitted plus any other termination byte as selected by PR2 (eg linefeed). |
| PR2 | byte | the character ‹byte› is used as the terminator byte by PR1. |
| RD1 | nothing | returns value held in output buffer, same as usr(0). |
| RD2 | address | reads counts held in scalar ‹address›, same as usr(3). |
| ST1 | add., n1, n2 | operates the stepper motor board number ‹add.› for ‹n1› steps with ‹n2› x 0.1 millisecs between steps. Same as usr(1) |
| ST2 | add., n1, n2 | same as ST1 but overlap option implemented. |
| ST3 | n1, n2, n3 | sets speeds for accelerating stepper motor driver ST4. n1 is the start/stop speed (in 0.1 msecs), n2 is the top speed (in 0.1 msecs) and n3 is the change in speed per step (in 0.1 msecs). |
| ST4 | add., n1 | drives motor ‹add.› ‹n1› number of steps. Speeds set by ST3. |

| | | |
|---|---|---|
| VP1 | add., n1, n2 | vector plot option using two channel D to A board ‹address›, same as usr(6). |
| VP2 | add., n1, n2 | as VP1 but is the same as usr(7). |

Parallel to serial conversion

This routine is entered either by the setting of the address switch at reset or by the MO5 command in Minicon. Bytes are taken from the GPIB interface and transmitted via the host serial port completely transparently. No buffering takes place as bytes are only removed from the GPIB interface when the serial port has transmitted the previous byte. Hardwire handshaking can be used to control the flow of data from the serial port, the speed at which data is taken from the GPIB interface will, therefore, also reflect this hanshaking. The routine is exited by the receipt of three consecutive escape codes (ascii 27), none of which will be transmitted via the serial port. Single and double escape sequences can, therefore, be sent through the interface for control of Epson printers etc. When the routine exits it returns to the Minicon routine with the GPIB interface selected.

```
          ;
          ;  MAIN MINICON ROUTINES. OCCUPIES EPROM
          ;  $F000 TO $FFFF. CONTAINS BOARD CONTROL
          ;  ROUTINES PLUS BINARY AND ASCII USER
          ;  INTERFACES
          ;
          ;  M.J. HILL 1985
          ;
          ;  SETUP VARIABLES
          ;
0006      USRCMD   EQU $06        vector for user cmmd.
0010      USRNO    EQU $10
0011      SAVX     EQU $11
0012      TMPO     EQU $12        output bytes
0014      WKSP     EQU $14        work space for A to D
001F      STATUS   EQU $1F        third byte of output
0020      VLIST    EQU $20        three input variables
0020      VA       EQU $20        A%
0022      N        EQU $22        N%
0024      T        EQU $24        and T%
0026      TEMPT    EQU $26        temporary T% location
0028      TVECLO   EQU $28
002A      TVECHI   EQU $2A
002C      WKLOC    EQU $2C
0030      NEWFLG   EQU $30        new ascii version flg
0031      NEGFLG   EQU $31        output is negative
0032      BUFFEN   EQU $32        buffer end pointer
0034      BPOINT   EQU $34        buffer fill pointer
0036      CHAR     EQU $36        output char store
0037      HGHBIT   EQU $37        highbyte of output
0038      LOWBIT   EQU $38        lowbyte of output
0039      CREG     EQU $39        command reg of ACIA 1
003A      CREG2    EQU $3A        command reg of ACIA 2
003B      IEEFLG   EQU $3B        IEEE488 flag
003C      IEEEPR   EQU $3C        IEEE488 as printer
003D      PDELM    EQU $3D        printer delimiter
003E      DELIM    EQU $3E        command delimiter
003F      LEADZ    EQU $3F        leading zero strip
0040      INCNT    EQU $40
0041      TPOINT   EQU $41        table pointer
0043      SELECT   EQU $43
0045      GETPTR   EQU $45
0047      PRINTR   EQU $47        serial printer flag
0048      HOST     EQU $48        serial host flag
0049      RETURN   EQU $49        return address loctn
0070      TOP      EQU $70        top speed of motor
0072      BOTTOM   EQU $72        bottom speed of motor
0074      INC      EQU $74        speed increment
0075      FAST     EQU $75        top speed flag
0076      SLOW     EQU $76        slow down flag
0077      DECL     EQU $77        number of steps to
          ;                       decelerate
          ;
          ; BOARD ADDRESSES ETC
          ;
8000      VIA      EQU $8000      via's
8020      IEE      EQU $8020      GPIA chip (68488)
801F      DATALO   EQU VIA+$1F    data bus low byte
```

```
        8010      DATAHI    EQU VIA+$10     data bus high byte
        8025      SRQ       EQU IEE+5       service request byte
        8030      ACIA1     EQU $8030       ACIA 1 (host)
        8038      ACIA2     EQU $8038       ACIA 2 (printer)
        8034      SWITCH    EQU $8034       parity switch
                  ;
                  ; BUFFERS
                  ;
        0300      BUFF      EQU $0300       input buffer
        0350      OUTBUF    EQU $0350       output buffer
        0200      TVHBUF    EQU $0200       work space for vector
        0220      TVLBUF    EQU $0220       generator
                  ;
                  ; CONSTANTS
                  ;
        000D      NCMDS     EQU $0D         number of commands
        0000      APNT      EQU $00
        0002      NPNT      EQU $02
        0004      TPNT      EQU $04
        00FF      ON        EQU $FF
        0000      OFF       EQU $0
        000D      CR        EQU $0D
        000A      LF        EQU $0A
        007F      TEX       EQU $7F         defines start of text
        00FF      ETX       EQU $FF         and end of text
                  ;
                  ; ENTRY JUMP TABLE IN $E000 EPROM
                  ;
        E000      MON       EQU $E000       location of monitor
                  ;
   F000                     ORG $F000
                  ;
                  ; JUMP TABLE FOR ROUTINES REQUIRED BY $E000 EPROM
                  ;
   F000 4C 4EFC             JMP RESET       these routines are
   F003 4C D3F7             JMP PRTCHR      required by code not
   F006 4C 20F8             JMP PRINT       contained in this
   F009 4C 82F7             JMP HOSTON      Eprom
   F00C 4C 87F7             JMP HOSTOF
   F00F 4C 8CF7             JMP PRTON
   F012 4C 91F7             JMP PRTOFF
   F015 4C 96F7             JMP RTSON
   F018 4C A0F7             JMP RTSOFF
   F01B 4C AAF7             JMP RSRES
   F01E 4C 4AF8             JMP PUSHSL
   F021 4C 5DF8             JMP POPSL
   F024 4C 90F8             JMP GETCHR
   F027 4C 28F7             JMP RDOB
   F02A 4C 70F8             JMP GETSL
   F02D 4C 9BF8             JMP MNSTRT
                  ;
                  ; INITIALISE VIA'S
                  ;
   F030 A9 00     INIT      LDA #0
   F032 8D 0F80             STA VIA+$0F     address = 0, no oper.
   F035 8D 1280             STA VIA+$12     data lines
   F038 8D 1380             STA VIA+$13     as inputs
   F03B 8D 1B80             STA VIA+$1B     disable special
```

```
F03E 8D 1C80            STA VIA+$1C    features
F041 A9 A0              LDA #$A0
F043 8D 0280            STA VIA+2      control lines
F046 A9 FF              LDA #$FF       address byte
F048 8D 0380            STA VIA+3
F04B A9 E0              LDA #$E0       enable timers
F04D 8D 0B80            STA VIA+$0B
F050 A9 0A              LDA #$0A
F052 8D 0C80            STA VIA+$0C    strobe operation
F055 A9 7F              LDA #$7F
F057 8D 1E80            STA VIA+$1E    disable interrupts
F05A 8D 0E80            STA VIA+$0E
F05D A9 A0              LDA #$A0       except timer 2
F05F 8D 0E80            STA VIA+$0E
F062 A9 84              LDA #$84
F064 8D 0480            STA VIA+4      timers for count
F067 A9 13              LDA #$13
F069 8D 0580            STA VIA+5
F06C A9 20              LDA #TVLBUF
F06E 85 28              STA TVECLO
F070 A9 00              LDA #TVHBUF
F072 85 2A              STA TVECHI
F074 A9 02              LDA #TVLBUF/256
F076 85 29              STA TVECLO+1
F078 A9 02              LDA #TVHBUF/256
F07A 85 2B              STA TVECHI+1
F07C 60                 RTS
                ;
                ; RESTART ROUTINE FOR BOARD CONTROL ROUTINES
                ;
F07D 60         RESTRT  RTS            all routines pass
                ;                      through here on
                ;                      completion, can be
                ;                      used to add code
                ;
                ; DUMMY USR(0)
                ;
F07E 20 4EF3    USR0    JSR PRTBUF     prints output bytes
F081 4C 7DF0            JMP RESTRT     again
                ;
                ; MOTOR USR(1)
                ;
F084 A5 21      USR1    LDA VA+1       is overlap option on
F086 F0 03              BEQ MT1
F088 20 2DF3            JSR ZEROUT     send reply if on
F08B 18         MT1     CLC            2's complement number
F08C A5 22              LDA N          of steps for ease of
F08E 49 FF              EOR #$FF       counting. Maximum
F090 69 01              ADC #1         steps are therefore
F092 85 22              STA N          32767
F094 A5 23              LDA N+1
F096 49 FF              EOR #$FF
F098 69 00              ADC #0
F09A 85 23              STA N+1
F09C 18                 CLC
F09D A5 24              LDA T          2's complement number
F09F 49 FF              EOR #$FF       of 100 microsecond
F0A1 69 01              ADC #1         delays per motor step
```

```
F0A3 85 24              STA T
F0A5 A5 25              LDA T+1
F0A7 49 FF              EOR #$FF
F0A9 69 00              ADC #0
F0AB 85 25              STA T+1
F0AD A5 20              LDA VA          set board address on
F0AF 8D 0F80            STA VIA+$0F     bus
F0B2 A9 62              LDA #$62        set 100 microsecond
F0B4 8D 0680            STA VIA+$06     count on timer 1
F0B7 20 CCF0   MCONT2   JSR STEP        step motor
F0BA E6 22              INC N           increment number of
F0BC D0 F9              BNE MCONT2      steps
F0BE E6 23              INC N+1         and loop until
F0C0 D0 F5              BNE MCONT2      non left
F0C2 A5 21              LDA VA+1
F0C4 D0 03              BNE MQ2
F0C6 20 2DF3            JSR ZEROUT      reply that finished
F0C9 4C 7DF0   MQ2      JMP RESTRT
               ;
               ; STEP MOTOR AFTER WAITING T LOTS OF 0.1
               ; MILLISECS
               ;
F0CC A5 24     STEP     LDA T           copy number of delays
F0CE 85 26              STA TEMPT       into temp location
F0D0 A5 25              LDA T+1
F0D2 85 27              STA TEMPT+1
F0D4 A9 00     ML1      LDA #0          start timer 1
F0D6 8D 0780            STA VIA+7       and clear timed out
F0D9 AD 0480            LDA VIA+4       flag
F0DC A9 40     ML2      LDA #$40        test timed out flag
F0DE 2C 0D80            BIT VIA+$0D
F0E1 F0 F9              BEQ ML2         loop until set
F0E3 E6 26              INC TEMPT       increment counter
F0E5 D0 ED              BNE ML1         and loop until
F0E7 E6 27              INC TEMPT+1     zero reached
F0E9 D0 E9              BNE ML1
F0EB AD 0180            LDA VIA+1       strobe to step motor
F0EE 60                 RTS             and return
               ;
               ; STEPPER MOTOR(3) OR USR(11) - SETS MIN,MAX
               ; SPEEDS AND ACCLN RATES FOR STEPPER MOTOR(4)
               ; OR USR(12)
               ;
F0EF 18        USR11    CLC
F0F0 A5 20              LDA VA          2's complement first
F0F2 49 FF              EOR #$FF        var is the start/stop
F0F4 69 01              ADC #1          speed
F0F6 85 72              STA BOTTOM      save it as bottom
F0F8 A5 21              LDA VA+1        speed
F0FA 49 FF              EOR #$FF
F0FC 69 00              ADC #0
F0FE 85 73              STA BOTTOM+1
F100 18                 CLC
F101 A5 22              LDA N           2's complement of 2nd
F103 49 FF              EOR #$FF        var is the fast speed
F105 69 01              ADC #1
F107 85 70              STA TOP
F109 A5 23              LDA N+1
```

```
F10B 49 FF               EOR #$FF
F10D 69 00               ADC #0
F10F 85 71               STA TOP+1
F111 18                  CLC
F112 A5 24               LDA T          2's complement of 3rd
F114 49 FF               EOR #$FF       var is the change in
F116 69 01               ADC #1         speed per motor step
F118 85 74               STA INC        ie the accln rate
F11A A5 25               LDA T+1
F11C 49 FF               EOR #$FF
F11E 69 00               ADC #0
F120 85 75               STA INC+1
F122 20 2DF3             JSR ZEROUT     reply routine
F125 4C 7DF0             JMP RESTRT     finished and return
                ;
                ; STEPPER MOTOR(4) OR USR(12) - ACCELERATING
                ; AND DECELERATING STEPPER MOTOR DRIVER
                ;
F128 A5 72      USR12    LDA BOTTOM     start off at bottom
F12A 85 24               STA T          speed
F12C A5 73               LDA BOTTOM+1
F12E 85 25               STA T+1
F130 A9 00               LDA #0
F132 85 75               STA FAST       clear fast speed flag
F134 85 76               STA SLOW       and start decelerating
F136 85 77               STA DECL       zero number of steps
F138 85 78               STA DECL+1     needed to decelerate
F13A 18                  CLC
F13B A5 22               LDA N          2's complement no. of
F13D 49 FF               EOR #$FF       steps ie maximum is
F13F 69 00               ADC #0         32767
F141 85 22               STA N
F143 A5 23               LDA N+1
F145 49 FF               EOR #$FF
F147 69 00               ADC #0
F149 85 23               STA N+1
F14B D0 04               BNE OK         if zero no. of steps
F14D A5 22               LDA N          exit routine without
F14F F0 28               BEQ MEXIT      doing anything
F151 A5 20      OK       LDA VA
F153 8D 0F80             STA VIA+$0F    set board address
F156 A9 64               LDA #$64       100 microsec clock
F158 8D 0680             STA VIA+$06    for execution time
F15B 20 CCF0    MAINLP   JSR STEP       step motor after wait
F15E A5 75               LDA FAST       reached top speed yet
F160 30 03               BMI JMP1       yes - don't accel
F162 20 7FF1             JSR ACCL       no so accelerate
F165 A5 76      JMP1     LDA SLOW       decelerating yet
F167 F0 03               BEQ GO         no so continue
F169 20 BDF1             JSR DECEL      yes so decelerate
F16C E6 22      GO       INC N
F16E D0 04               BNE TESTDC     not yet zero so test
                ;                       if decelerate yet
F170 E6 23               INC N+1
F172 F0 05               BEQ MEXIT      zero steps left exit
F174 20 ACF1    TESTDC   JSR TSTDCL     test if decel point
                ;                       reached yet
F177 90 E2               BCC MAINLP     continue stepping
```

```
F179 20 2DF3   MEXIT    JSR ZEROUT      routine has finished
F17C 4C 7DF0            JMP RESTRT      return
               ;
               ;
F17F 18        ACCL     CLC             add accel increment
F180 A5 24              LDA T           to 2's complement of
F182 65 74              ADC INC         delay thus decreasing
F184 85 24              STA T           delay time and
F186 A5 25              LDA T+1         increasing speed
F188 65 75              ADC INC+1
F18A 85 25              STA T+1
F18C B0 0B              BCS PASTF       delay gone past zero
F18E 18                 CLC             - use top speed
F18F C5 71              CMP TOP+1       top speed reached ?
F191 90 12              BCC AEXIT       no continue
F193 A5 24              LDA T
F195 C5 70              CMP TOP
F197 90 0C              BCC AEXIT
F199 A5 70     PASTF    LDA TOP         yes set speed to top
F19B 85 24              STA T           speed
F19D A5 71              LDA TOP+1
F19F 85 25              STA T+1
F1A1 A9 FF              LDA #ON
F1A3 85 75              STA FAST
F1A5 E6 77     AEXIT    INC DECL        increment no. of steps
F1A7 D0 02              BNE ACONT       required to decelerate
F1A9 E6 78              INC DECL+1
F1AB 60        ACONT    RTS             return
               ;
F1AC 18        TSTDCL   CLC             add no. of decel steps
F1AD A5 22              LDA N           to no. of motor steps
F1AF 65 77              ADC DECL
F1B1 A5 23              LDA N+1
F1B3 65 78              ADC DECL+1
F1B5 90 04              BCC NOTYET      if number passed zero
F1B7 A9 FF              LDA #ON         start decelerating so
F1B9 85 76              STA SLOW        set flag
F1BB 18        NOTYET   CLC             carry cleared before
F1BC 60                 RTS             returning
               ;
F1BD 38        DECEL    SEC
F1BE A5 24              LDA T           subtract increment
F1C0 E5 74              SBC INC         from step delay thus
F1C2 85 24              STA T           making speed slower
F1C4 A5 25              LDA T+1
F1C6 E5 75              SBC INC+1
F1C8 85 25              STA T+1
F1CA 90 0B              BCC UNDER
F1CC 18                 CLC             compare delay with
F1CD C5 73              CMP BOTTOM+1    bottom speed
F1CF B0 0E              BCS DEXIT       if less use bottom
F1D1 A5 24              LDA T           speed
F1D3 C5 72              CMP BOTTOM
F1D5 B0 08              BCS DEXIT
F1D7 A5 72     UNDER    LDA BOTTOM
F1D9 85 24              STA T
F1DB A5 73              LDA BOTTOM+1
F1DD 85 25              STA T+1
```

```
F1DF 60         DEXIT    RTS
                ;
                ; COUNT USR(2)
                ;
F1E0 A5 21      USR2     LDA VA+1       check overlap option
F1E2 F0 03               BEQ CT1
F1E4 20 2DF3             JSR ZEROUT
F1E7 A9 A0      CT1      LDA #$A0
F1E9 8D 0E80             STA VIA+$0E
F1EC A9 00               LDA #0
F1EE 8D 0080             STA VIA
F1F1 A5 24               LDA T          set timer 2 to count
F1F3 8D 0880             STA VIA+8      down delay time
F1F6 A5 25               LDA T+1
F1F8 8D 0980             STA VIA+9
F1FB A9 FF               LDA #$FF
F1FD 8D 0080             STA VIA
F200 A9 20               LDA #$20
F202 2C 0D80    CL1      BIT VIA+$0D    check if timer 2 has
F205 F0 FB               BEQ CL1        counted down to zero
                ;
                ; READ USR(3)
                ;
F207 A5 20      USR3     LDA VA         set up board address
F209 8D 0F80             STA VIA+$0F    on address bus
F20C AD 1080             LDA DATAHI     read data bus
F20F 49 FF               EOR #$FF       and invert data bits
F211 85 13               STA TMP0+1
F213 AD 1F80             LDA DATALO     and for low byte
F216 49 FF               EOR #$FF
F218 85 12               STA TMP0       put data in output
F21A A5 21               LDA VA+1       buffer
F21C D0 03               BNE RED1
F21E 20 4EF3             JSR PRTBUF     output data to user
F221 4C 7DF0    RED1     JMP RESTRT     and back to start
                ;
                ; D TO A (MK 1)
                ;
F224 A5 20      USR4     LDA VA         set up VIA's
F226 8D 0F80             STA VIA+$0F
F229 A9 FF               LDA #$FF
F22B 8D 1280             STA VIA+$12
F22E 8D 1380             STA VIA+$13
F231 A9 02               LDA #$02
F233 24 21               BIT VA+1       see if 512 added to
F235 F0 03               BEQ DA1        address - 4 channel
F237 4C F7F2             JMP DAC4       board if yes
F23A A5 23      DA1      LDA N+1        put data on data bus
F23C 8D 1080             STA DATAHI
F23F A5 22               LDA N
F241 8D 1F80             STA DATALO
F244 AD 0180             LDA VIA+1      and latch it (strobe)
F247 20 2DF3             JSR ZEROUT     reply to user
F24A 4C 7DF0             JMP RESTRT
                ;
                ; A TO D USR(5)
                ;
F24D A5 20      USR5     LDA VA         set address
```

```
F24F 8D 0F80          STA VIA+$0F
F252 A9 00            LDA #0
F254 8D 1C80          STA VIA+$1C
F257 AD 1080          LDA VIA+$10    clear flag and
F25A AD 0180          LDA VIA+1      start conversion
F25D A9 FF            LDA #$FF       2's complement of
F25F 85 26            STA TEMPT      time out value for
F261 A9 FE            LDA #$FE       board error
F263 85 27            STA TEMPT+1
F265 A9 08     AD1    LDA #$08       test if conversion
F267 2C 1D80          BIT VIA+$1D    has begun
F26A D0 07            BNE ST1        yes - coninue
F26C 20 E0F2          JSR TOUT       no see if timed out
F26F 90 F4            BCC AD1        no test bit again
F271 B0 78            BCS TIMERR     yes - error
F273 A9 40     ST1    LDA #$40
F275 8D 1C80          STA VIA+$1C
F278 8D 1080          STA VIA+$10    clear flag
F27B A9 08            LDA #$08
F27D 2C 1D80   AD2    BIT VIA+$1D    conversion complete ?
F280 F0 FB            BEQ AD2        no - loop
F282 A9 10            LDA #$10
F284 2C 1080          BIT DATAHI     overrange ?
F287 D0 4B            BNE OVERR
F289 A9 20            LDA #$20
F28B 2C 1080          BIT DATAHI     check sign bit
F28E D0 1E            BNE PLUS       clear if positive
F290 A9 FF     NEG    LDA #ON        set negative flag
F292 85 31            STA NEGFLG     for output routine
F294 18               CLC
F295 AD 1F80          LDA DATALO     2's complement data
F298 49 FF            EOR #$FF       before returning
F29A 69 01            ADC #01
F29C 48               PHA
F29D AD 1080          LDA DATAHI
F2A0 29 0F            AND #$0F
F2A2 49 FF            EOR #$FF
F2A4 69 00            ADC #0
F2A6 85 13            STA TMP0+1
F2A8 68               PLA
F2A9 85 12            STA TMP0       data in output buffer
F2AB 18               CLC
F2AC 90 12            BCC ZDELY      delay required to
               ;                     limit access speed
F2AE AD 1F80   PLUS   LDA DATALO
F2B1 48               PHA
F2B2 AD 1080          LDA DATAHI
F2B5 29 0F            AND #$0F
F2B7 85 13            STA TMP0+1
F2B9 68               PLA
F2BA 85 12            STA TMP0       data in output buffer
F2BC A9 00            LDA #OFF
F2BE 85 31            STA NEGFLG     clear negative flag
F2C0 A2 0A     ZDELY  LDX #$0A       simple delay loop
F2C2 A0 C8     ZD2    LDY #$C8
F2C4 88        ZD1    DEY
F2C5 D0 FD            BNE ZD1
F2C7 CA               DEX
```

```
F2C8 D0 F8                 BNE ZD2
F2CA 20 4EF3               JSR PRTBUF      print value to user
F2CD A9 00                 LDA #OFF        clear negative flag
F2CF 85 31                 STA NEGFLG
F2D1 4C 7DF0               JMP RESTRT      and restart
F2D4 A9 08     OVERR       LDA #08         set bit in status
F2D6 05 1F                 ORA STATUS      byte
F2D8 85 1F                 STA STATUS
F2DA 20 40F7               JSR OVER        output error code
F2DD 4C 7DF0               JMP RESTRT
F2E0 18        TOUT        CLC             increment time out
F2E1 E6 26                 INC TEMPT       counter
F2E3 D0 05                 BNE TO1
F2E5 E6 27                 INC TEMPT+1
F2E7 D0 01                 BNE TO1
F2E9 38                    SEC             carry set if timed
F2EA 60        TO1         RTS             out
F2EB A9 02     TIMERR      LDA #2
F2ED 05 1F                 ORA STATUS      set status bit
F2EF 85 1F                 STA STATUS
F2F1 20 49F7               JSR TMOUT       output error code
F2F4 4C 7DF0               JMP RESTRT

               ;
               ; 4-CHANNEL DAC DRIVER
               ; USR(4)
               ;
F2F7 A5 22     DAC4        LDA N           get low data byte
F2F9 8D 1F80               STA DATALO
F2FC A9 00                 LDA #0          zero high byte
F2FE 8D 1080               STA DATAHI
F301 AD 0180               LDA VIA+$01     latch lower 4 bits
F304 EE 1080               INC DATAHI      add one to high byte
F307 4E 1F80               LSR DATALO      put top 4 bits of
F30A 4E 1F80               LSR DATALO      low byte into bottom
F30D 4E 1F80               LSR DATALO      4 bits
F310 4E 1F80               LSR DATALO
F313 AD 0180               LDA VIA+$01     and latch it in
F316 EE 1080               INC DATAHI      address for top 4
F319 A5 23                 LDA N+1         bits and fetch high
F31B 8D 1F80               STA DATALO      byte of data
F31E AD 0180               LDA VIA+$01     and latch it in
F321 EE 1080               INC DATAHI      and latch start
F324 AD 0180               LDA VIA+$01     conversion
F327 20 2DF3               JSR ZEROUT      reply to user
F32A 4C 7DF0               JMP RESTRT
               ;
               ;
F32D A5 30     ZEROUT      LDA NEWFLG      test if new ascii
F32F F0 03                 BEQ IEEE        mode - no old routine
F331 4C 3EFB               JMP REPLOK      yes - new routine
F334 A9 20     IEEE        LDA #$20        set output ready in
F336 8D 2580               STA SRQ         serial poll byte
F339 A9 00                 LDA #0          output two null bytes
F33B 20 33F7               JSR WROB        via the GPIB
F33E A9 00                 LDA #0
F340 20 33F7               JSR WROB
F343 A5 1F                 LDA STATUS      and the status byte
F345 20 33F7               JSR WROB
```

```
F348 A9 00                LDA #0           serial poll byte to
F34A 8D 2580              STA SRQ          ready for input
F34D 60                   RTS
F34E A5 30      PRTBUF    LDA NEWFLG       test ascii mode
F350 F0 03                BEQ IEPRT
F352 4C DAFB              JMP REPLAN
F355 A9 20      IEPRT     LDA #$20         output ready in
F357 8D 2580              STA SRQ          serial poll byte
F35A A5 12                LDA TMPO         and write out the
F35C 20 33F7              JSR WROB         output buffer
F35F A5 13                LDA TMPO+1
F361 20 33F7              JSR WROB
F364 A5 1F                LDA STATUS       plus the status byte
F366 20 33F7              JSR WROB
F369 A9 00                LDA #0
F36B 8D 2580              STA SRQ          clear serial poll
F36E 60                   RTS
                ;
F36F A5 21      USR7      LDA VA+1
F371 29 FE                AND #$FE
F373 4A                   LSR A
F374 85 11                STA SAVX         set up speed
F376 A5 20                LDA VA
F378 8D 0F80              STA VIA+$0F
F37B A9 FF                LDA #$FF
F37D 8D 1280              STA VIA+$12
F380 8D 1380              STA VIA+$13
F383 A0 02                LDY #NPNT
F385 20 A8F3              JSR GETV
F388 A0 07                LDY #7
F38A 20 B1F3              JSR PUT          X into table
F38D 18                   CLC
F38E 20 E7F4              JSR PLOT         output it
F391 A0 04                LDY #TPNT
F393 20 A8F3              JSR GETV
F396 A0 04                LDY #4
F398 20 B1F3              JSR PUT          Y into table
F39B 38                   SEC
F39C 20 E7F4              JSR PLOT         output it
F39F 20 17F5              JSR DELAY        delay for plotter
                ;                          speed
F3A2 20 2DF3              JSR ZEROUT       reply to user
F3A5 4C 7DF0              JMP RESTRT
                ;
F3A8 B9 2000    GETV      LDA VLIST,Y      get variable from
F3AB AA                   TAX              location in VLIST
F3AC C8                   INY
F3AD B9 2000              LDA VLIST,Y
F3B0 60                   RTS
                ;
F3B1 91 2A      PUT       STA (TVECHI),Y put variable in
F3B3 8A                   TXA              table TVE
F3B4 91 28                STA (TVECLO),Y
F3B6 60                   RTS
                ;
                ; VECTOR GENERATOR
                ;
F3B7 A5 21      USR6      LDA VA+1
```

```
F3B9 85 11                STA SAVX        set up speed
F3BB 46 11                LSR SAVX
F3BD 29 01                AND #01
F3BF 85 21                STA VA+1        quick reply ?
F3C1 F0 03                BEQ VEC2
F3C3 20 2DF3              JSR ZEROUT
F3C6 A5 20       VEC2     LDA VA          set board address
F3C8 8D 0F80              STA VIA+$0F
F3CB A9 FF                LDA #$FF
F3CD 8D 1280              STA VIA+$12
F3D0 8D 1380              STA VIA+$13
F3D3 A0 02                LDY #NPNT
F3D5 20 A8F3              JSR GETV
F3D8 A0 08                LDY #8
F3DA 20 B1F3              JSR PUT         X2 into table
F3DD A0 04                LDY #TPNT
F3DF 20 A8F3              JSR GETV
F3E2 A0 05                LDY #5
F3E4 20 B1F3              JSR PUT         Y2 into table
F3E7 20 33F4              JSR DELTAX
F3EA 20 45F4              JSR DELTAY
F3ED A0 01                LDY #1
F3EF 20 70F4              JSR CLEAR       clear accumulator
F3F2 A5 20                LDA VA
F3F4 A0 07       VLOP     LDY #7
F3F6 18                   CLC
F3F7 20 E7F4              JSR PLOT        plot X1,Y1
F3FA A0 04                LDY #4
F3FC 38                   SEC
F3FD 20 E7F4              JSR PLOT
F400 A0 08                LDY #8
F402 20 77F4              JSR SUBTR
F405 A5 28                LDA TEMPT+2
F407 F0 20                BEQ RETRN
F409 A0 01       TESTA    LDY #1
F40B 20 9EF4              JSR TEST
F40E A5 28                LDA TEMPT+2
F410 10 06                BPL UP
F412 20 D2F4              JSR LOW
F415 4C 26F4              JMP BACK
F418 F0 06       UP       BEQ GOOD
F41A 20 BDF4              JSR HIGH
F41D 4C 26F4              JMP BACK
F420 20 D2F4     GOOD     JSR LOW
F423 20 BDF4              JSR HIGH
F426 4C F4F3     BACK     JMP VLOP
F429 A5 21       RETRN    LDA VA+1
F42B D0 03                BNE VRET
F42D 20 2DF3              JSR ZEROUT
F430 4C 7DF0     VRET     JMP RESTRT
                 ;
F433 A0 08       DELTAX   LDY #08
F435 20 77F4              JSR SUBTR
F438 88                   DEY
F439 20 A9F4              JSR SIGMA
F43C A0 02                LDY #2
F43E 20 5AF4              JSR MOD
F441 20 10F5              JSR STORE
```

```
F444 60                  RTS
                ;
F445 A0 05      DELTAY   LDY #5
F447 20 77F4             JSR SUBTR
F44A 88                  DEY
F44B 20 A9F4             JSR SIGMA
F44E A0 00               LDY #0
F450 20 5AF4             JSR MOD
F453 20 5FF4             JSR NEGATE
F456 20 10F5             JSR STORE
F459 60                  RTS
                ;
F45A 18         MOD      CLC
F45B A5 28               LDA TEMPT+2
F45D 10 10               BPL VRET1
F45F A9 FF      NEGATE   LDA #$FF
F461 45 26               EOR TEMPT
F463 69 01               ADC #1
F465 85 26               STA TEMPT
F467 A9 FF               LDA #$FF
F469 45 27               EOR TEMPT+1
F46B 69 00               ADC #0
F46D 85 27               STA TEMPT+1
F46F 60         VRET1    RTS
                ;
F470 A9 00      CLEAR    LDA #0
F472 91 28               STA (TVECLO),Y
F474 91 2A               STA (TVECHI),Y
F476 60                  RTS
                ;
F477 38         SUBTR    SEC
F478 B1 28               LDA (TVECLO),Y
F47A 88                  DEY
F47B F1 28               SBC (TVECLO),Y
F47D 85 26               STA TEMPT
F47F C8                  INY
F480 B1 2A               LDA (TVECHI),Y
F482 88                  DEY
F483 F1 2A               SBC (TVECHI),Y
F485 85 27               STA TEMPT+1
F487 10 05      TEST2    BPL EUAL
F489 A9 FF               LDA #$FF
F48B 85 28               STA TEMPT+2     sign flag
F48D 60                  RTS
                ;
F48E D0 09      EUAL     BNE POS
F490 A5 26               LDA TEMPT
F492 D0 05               BNE POS
F494 A9 00               LDA #0
F496 85 28               STA TEMPT+2
F498 60                  RTS
                ;
F499 A9 01      POS      LDA #1
F49B 85 28               STA TEMPT+2
F49D 60                  RTS
                ;
F49E B1 28      TEST     LDA (TVECLO),Y
F4A0 85 26               STA TEMPT
```

```
F4A2 B1 2A                  LDA (TVECHI),Y
F4A4 85 27                  STA TEMPT+1
F4A6 4C 87F4                JMP TEST2
                  ;
F4A9 A5 28        SIGMA     LDA TEMPT+2     sign of delta
F4AB 10 05                  BPL ZER2
F4AD 91 28        PUTIT     STA (TVECLO),Y
F4AF 91 2A                  STA (TVECHI),Y
F4B1 60                     RTS
                  ;
F4B2 D0 02        ZER2      BNE POSIT
F4B4 F0 F7                  BEQ PUTIT
F4B6 91 28        POSIT     STA (TVECLO),Y
F4B8 A9 00                  LDA #0
F4BA 91 2A                  STA (TVECHI),Y
F4BC 60                     RTS
                  ;
F4BD A0 07        HIGH      LDY #7
F4BF 20 FFF4                JSR ADD
F4C2 A0 07                  LDY #7
F4C4 20 10F5                JSR STORE
F4C7 A0 01                  LDY #1
F4C9 20 FFF4                JSR ADD
F4CC A0 01                  LDY #1
F4CE 20 10F5                JSR STORE
F4D1 60                     RTS
                  ;
F4D2 A0 04        LOW       LDY #4
F4D4 20 FFF4                JSR ADD
F4D7 A0 04                  LDY #4
F4D9 20 10F5                JSR STORE
F4DC A0 02                  LDY #2
F4DE 20 FFF4                JSR ADD
F4E1 A0 01                  LDY #1
F4E3 20 10F5                JSR STORE
F4E6 60                     RTS
                  ;
F4E7 A5 20        PLOT      LDA VA          output data in table
F4E9 69 00                  ADC #0          to D to A
F4EB 8D 0F80                STA VIA+$0F
F4EE B1 28                  LDA (TVECLO),Y
F4F0 8D 1F80                STA DATALO
F4F3 B1 2A                  LDA (TVECHI),Y
F4F5 8D 1080                STA DATAHI
F4F8 AD 0180                LDA VIA+1       latch it
F4FB 20 17F5                JSR DELAY       delay for pen speed
F4FE 60                     RTS
                  ;
F4FF 18           ADD       CLC
F500 B1 28                  LDA (TVECLO),Y
F502 88                     DEY
F503 71 28                  ADC (TVECLO),Y
F505 85 26                  STA TEMPT
F507 C8                     INY
F508 B1 2A                  LDA (TVECHI),Y
F50A 88                     DEY
F50B 71 2A                  ADC (TVECHI),Y
F50D 85 27                  STA TEMPT+1
```

```
F50F 60                RTS
                  ;
F510 A6 26        STORE      LDX TEMPT
F512 A5 27                   LDA TEMPT+1
F514 4C B1F3                 JMP PUT
                  ;
F517 A6 11        DELAY      LDX SAVX
F519 F0 08                   BEQ DRET
F51B A0 64        DL2        LDY #$64
F51D 88           DL1        DEY
F51E D0 FD                   BNE DL1
F520 CA                      DEX
F521 D0 F8                   BNE DL2
F523 60           DRET       RTS
                  ;
                  ; GPIP MODULE DRIVER USR(8)
                  ;
F524 A5 20        USR8       LDA VA          board address
F526 8D 0F80                 STA VIA+$0F
F529 A9 00                   LDA #0
F52B 8D 0080                 STA VIA         clear flag
F52E A9 FF                   LDA #$FF
F530 8D 0080                 STA VIA
F533 AD 1080                 LDA VIA+$10
F536 A9 40                   LDA #$40
F538 8D 1C80                 STA VIA+$1C     set up handshake
F53B AD 0180                 LDA VIA+1       set DRQ
F53E A9 FF                   LDA #$FF
F540 85 26                   STA TEMPT       set up time out
F542 A9 80                   LDA #$80
F544 85 27                   STA TEMPT+1
F546 A9 08                   LDA #8
F548 2C 1D80      GPL1       BIT VIA+$1D     wait for reply
F54B D0 08                   BNE GPL2
F54D 20 E0F2                 JSR TOUT
F550 90 F6                   BCC GPL1
F552 4C EBF2                 JMP TIMERR
F555 AD 1080      GPL2       LDA DATAHI
F558 49 FF                   EOR #$FF
F55A 85 13                   STA TMP0+1
F55C AD 1F80                 LDA DATALO
F55F 49 FF                   EOR #$FF
F561 85 12                   STA TMP0
F563 20 4EF3                 JSR PRTBUF
F566 4C 7DF0                 JMP RESTRT
                  ;
                  ; USR(9) - JUMP TO MONITOR AT $E000
                  ;
F569 AD 00E0      USR9       LDA $E000
F56C C9 4C                   CMP #$4C        is jump instruction at
                  ;                          $E000 present
F56E D0 03                   BNE MONERR      no - not there !
F570 4C 00E0                 JMP MON         yes - so jump to it
F573 A5 30        MONERR     LDA NEWFLG      error - monitor chip
F575 D0 06                   BNE MNERR       not installed
F577 20 2DF3                 JSR ZEROUT      old mode so reply
F57A 4C 00F0                 JMP RESET       reset routine
F57D 20 20F8      MNERR      JSR PRINT       print error message
```

```
F580 7F4D4F4E           DB TEX,'MONITOR EPROM NOT PRESENT',$0D,ETX
F584 49544F52
F588 20455052
F58C 4F4D204E
F590 4F542050
F594 52455345
F598 4E540DFF
F59C 4C 00F0            JMP RESTRT     and reset 6502
     F569      MONITR   EQU USR9
               ;
               ; AUTORANGING A TO D DRIVER
               ;
F59F A5 24     USR10    LDA T
F5A1 F0 03              BEQ S1
F5A3 4C 1BF6            JMP AUTO
F5A6 A9 00     S1       LDA #0
F5A8 85 31              STA NEGFLG
F5AA 20 B0F5            JSR SCONV
F5AD 4C 08F7            JMP RET
               ;
F5B0 A5 20     SCONV    LDA VA
F5B2 8D 0F80            STA VIA+$0F
F5B5 A9 FF              LDA #$FF
F5B7 8D 1280            STA VIA+$12
F5BA 8D 1380            STA VIA+$13
F5BD A5 23              LDA N+1
F5BF 8D 1080            STA DATAHI
F5C2 A5 22              LDA N
F5C4 8D 1F80            STA DATALO
F5C7 AD 0180            LDA VIA+1
F5CA A2 50              LDX #$50
F5CC CA        DECR     DEX
F5CD D0 FD              BNE DECR
F5CF A9 00              LDA #0
F5D1 8D 1280            STA VIA+$12
F5D4 8D 1380            STA VIA+$13
F5D7 A9 80              LDA #$80
F5D9 8D 0F80            STA VIA+$0F
F5DC A9 00              LDA #0
F5DE 8D 0F80            STA VIA+$0F
F5E1 A9 00              LDA #0
F5E3 8D 1C80            STA VIA+$1C
F5E6 8D 1080            STA VIA+$10
F5E9 A2 FF              LDX #$FF
F5EB AD 1D80   WCONV    LDA VIA+$1D
F5EE 29 08              AND #08
F5F0 D0 12              BNE DONE
F5F2 CA                 DEX
F5F3 F0 03              BEQ ATOUT
F5F5 4C EBF5            JMP WCONV
F5F8 A9 FF     ATOUT    LDA #$FF
F5FA 85 13              STA TMP0+1
F5FC 85 12              STA TMP0
F5FE 20 4EF3            JSR PRTBUF
F601 4C 7DF0            JMP RESTRT
F604 A9 80     DONE     LDA #$80
F606 8D 0F80            STA VIA+$0F
F609 AD 1080            LDA DATAHI
```

```
F60C 29 0F                 AND #$0F
F60E 85 13                 STA TMPO+1
F610 AD 1F80               LDA DATALO
F613 85 12                 STA TMPO
F615 A9 00                 LDA #0
F617 8D 0F80               STA VIA+$0F
F61A 60                    RTS
                  ;
F61B 20 B0F5      AUTO     JSR SCONV
F61E A5 13                 LDA TMPO+1
F620 85 2D                 STA WKLOC+1
F622 A5 12                 LDA TMPO
F624 85 2C                 STA WKLOC
F626 A9 00                 LDA #0
F628 85 2E                 STA WKLOC+2
F62A A5 22                 LDA N
F62C 29 10                 AND #$10
F62E D0 3E                 BNE BIP
F630 A9 00        UNI      LDA #0
F632 85 31                 STA NEGFLG
F634 A5 2D                 LDA WKLOC+1
F636 F0 10                 BEQ A3
F638 29 08        A1       AND #$08
F63A D0 09                 BNE A2
F63C E6 2E                 INC WKLOC+2
F63E 06 2D                 ASL WKLOC+1
F640 A5 2D                 LDA WKLOC+1
F642 4C 38F6               JMP A1
F645 4C 64F6      A2       JMP A7
F648 A5 2C        A3       LDA WKLOC
F64A F0 15                 BEQ A6
F64C 29 08        A4       AND #$08
F64E D0 09                 BNE A5
F650 E6 2E                 INC WKLOC+2
F652 06 2C                 ASL WKLOC
F654 A5 2C                 LDA WKLOC
F656 4C 4CF6               JMP A4
F659 A2 04        A5       LDX #$04
F65B 20 12F7               JSR INCR
F65E 4C 64F6               JMP A7
F661 4C 08F7      A6       JMP RET
F664 A5 2E        A7       LDA WKLOC+2
F666 85 22                 STA N
F668 20 B0F5               JSR SCONV
F66B 4C 18F7               JMP WASL
                  ;
F66E A5 2D        BIP      LDA WKLOC+1
F670 29 08                 AND #$08
F672 F0 42                 BEQ ANEG
F674 A9 00        APOS     LDA #0
F676 85 31                 STA NEGFLG
F678 A5 2D                 LDA WKLOC+1
F67A 29 07                 AND #$07
F67C F0 10                 BEQ P3
F67E A5 2D        P1       LDA WKLOC+1
F680 29 04                 AND #$04
F682 D0 07                 BNE P2
F684 E6 2E                 INC WKLOC+2
```

```
F686 06 2D                ASL WKLOC+1
F688 4C 7EF6              JMP P1
F68B 4C AAF6    P2        JMP P7
F68E A5 2C      P3        LDA WKLOC
F690 F0 15                BEQ P6
F692 29 80      P4        AND #$80
F694 D0 09                BNE P5
F696 E6 2E                INC WKLOC+2
F698 06 2C                ASL WKLOC
F69A A5 2C                LDA WKLOC
F69C 4C 92F6              JMP P4
F69F A2 03      P5        LDX #$03
F6A1 20 12F7              JSR INCR
F6A4 4C AAF6              JMP P7
F6A7 4C 08F7    P6        JMP RET
F6AA A5 2E      P7        LDA WKLOC+2
F6AC 49 10                EOR #$10
F6AE 85 22                STA N
F6B0 20 B0F5              JSR SCONV
F6B3 4C 18F7              JMP WASL
                ;
F6B6 A9 FF      ANEG      LDA #$FF
F6B8 85 31                STA NEGFLG
F6BA A5 2D                LDA WKLOC+1
F6BC 49 07                EOR #$07
F6BE 85 2D                STA WKLOC+1
F6C0 A5 2C                LDA WKLOC
F6C2 49 FF                EOR #$FF
F6C4 85 2C                STA WKLOC
F6C6 E6 2C                INC WKLOC
F6C8 D0 02                BNE N0
F6CA E6 2D                INC WKLOC+1
F6CC A5 2D      N0        LDA WKLOC+1
F6CE F0 10                BEQ N3
F6D0 29 04      N1        AND #$04
F6D2 D0 09                BNE N2
F6D4 E6 2E                INC WKLOC+2
F6D6 06 2D                ASL WKLOC+1
F6D8 A5 2D                LDA WKLOC+1
F6DA 4C D0F6              JMP N1
F6DD 4C FCF6    N2        JMP N7
F6E0 A5 2C      N3        LDA WKLOC
F6E2 F0 15                BEQ N6
F6E4 29 80      N4        AND #$80
F6E6 D0 09                BNE N5
F6E8 E6 2E                INC WKLOC+2
F6EA 06 2C                ASL WKLOC
F6EC A5 2C                LDA WKLOC
F6EE 4C E4F6              JMP N4
F6F1 A2 03      N5        LDX #$03
F6F3 20 12F7              JSR INCR
F6F6 4C FCF6              JMP N7
F6F9 4C 08F7    N6        JMP RET
F6FC A5 2E      N7        LDA WKLOC+2
F6FE 49 10                EOR #$10
F700 85 22                STA N
F702 20 B0F5              JSR SCONV
F705 4C 18F7              JMP WASL
```

```
                      ;
F708 20 4EF3      RET        JSR PRTBUF
F70B A9 00                   LDA #0
F70D 85 31                   STA NEGFLG
F70F 4C 7DF0                 JMP RESTRT
                      ;
F712 E6 2E        INCR       INC WKLOC+2
F714 CA                      DEX
F715 D0 FB                   BNE INCR
F717 60                      RTS
                      ;
F718 A2 04        WASL       LDX #$04
F71A 06 2E                   ASL WKLOC+2
F71C CA                      DEX
F71D D0 F9                   BNE WASL
F71F A5 2E                   LDA WKLOC+2
F721 45 13                   EOR TMP0+1
F723 85 13                   STA TMP0+1
F725 4C 08F7                 JMP RET
                      ;
                      ; READ BYTE USING 68488 GPIA CHIP
                      ;
F728 A9 01        RDOB       LDA #1
F72A 2C 2080      RL1        BIT IEE
F72D F0 FB                   BEQ RL1
F72F AD 2780                 LDA IEE+7
F732 60                      RTS
                      ;
                      ; WRITE OUT BYTE USING GPIA CHIP
                      ;
F733 48           WROB       PHA
F734 A9 40        WR1        LDA #$40
F736 2C 2080                 BIT IEE
F739 F0 F9                   BEQ WR1
F73B 68                      PLA
F73C 8D 2780                 STA IEE+7
F73F 60                      RTS
                      ;
                      ; OVERRANGE
                      ;
F740 A5 30        OVER       LDA NEWFLG       test if ascii mode
F742 F0 02                   BEQ IEOVR        no - binary reply
F744 D0 0D                   BNE RSOVER       yes - send error
F746 4C 2DF3      IEOVR      JMP ZEROUT       binary reply
                      ;
                      ; TIME OUT
                      ;
F749 A5 30        TMOUT      LDA NEWFLG       test if ascii mode
F74B F0 03                   BEQ IETIM
F74D D0 14                   BNE RSTIM
F74F 60                      RTS
F750 4C 2DF3      IETIM      JMP ZEROUT       binary reply
                      ;
F753 20 20F8      RSOVER     JSR PRINT
F756 7F4F5645                DB TEX,'OVERRANGE',$0D,ETX
F75A 5252414E
F75E 47450DFF
F762 60                      RTS
```

```
                         ;
F763 20 20F8     RSTIM      JSR PRINT
F766 7F54494D               DB TEX,'TIMED OUT',$0D,ETX
F76A 4544204F
F76E 55540DFF
F772 60                     RTS
                         ;
                         ; VARIABLE INITIALISATION
                         ;
F773 A9 00       VINIT      LDA #OFF
F775 85 47                  STA PRINTR
F777 85 48                  STA HOST
F779 85 36                  STA CHAR
F77B 85 3C                  STA IEEEPR
F77D A9 0D                  LDA #$0D
F77F 85 3E                  STA DELIM
F781 60                     RTS
                         ;
                         ; PRINT ROUTINES FOR OUTPUT CONTROL
                         ;
F782 A9 FF       HOSTON     LDA #ON
F784 85 48                  STA HOST
F786 60                     RTS
F787 A9 00       HOSTOF     LDA #OFF
F789 85 48                  STA HOST
F78B 60                     RTS
F78C A9 FF       PRTON      LDA #ON
F78E 85 47                  STA PRINTR
F790 60                     RTS
F791 A9 00       PRTOFF     LDA #OFF
F793 85 47                  STA PRINTR
F795 60                     RTS
F796 A5 39       RTSON      LDA CREG
F798 29 9F                  AND #$9F
F79A 8D 3080                STA ACIA1
F79D 85 39                  STA CREG
F79F 60                     RTS
F7A0 A5 39       RTSOFF     LDA CREG
F7A2 09 40                  ORA #$40
F7A4 8D 3080                STA ACIA1
F7A7 85 39                  STA CREG
F7A9 60                     RTS
F7AA 18          RSRES      CLC              read parity select
F7AB AD 3480                LDA SWITCH       switch
F7AE 29 70                  AND #$70         mask bits for ACIA 1
F7B0 4A                     LSR A            shift to position
F7B1 4A                     LSR A            in command register
F7B2 09 43                  ORA #$43         add master reset bit
F7B4 8D 3080                STA ACIA1        put in command reg
F7B7 29 FD                  AND #$FD         clear reset bit
F7B9 8D 3080                STA ACIA1        store it
F7BC 85 39                  STA CREG         copy to CREG
F7BE 18                     CLC              same again for ACIA 2
F7BF AD 3480                LDA SWITCH
F7C2 29 07                  AND #$07         both have /16 clock
F7C4 0A                     ASL A            and RTS set low
F7C5 0A                     ASL A
F7C6 09 03                  ORA #$03         copy of command reg
```

```
F7C8 8D 3880            STA ACIA2      kept in RAM as is a
F7CB 29 FD              AND #$FD       read only register
F7CD 8D 3880            STA ACIA2      in the chip
F7D0 85 3A              STA CREG2
F7D2 60                 RTS
F7D3 C9 00     PRTCHR   CMP #0         null character ?
F7D5 F0 1D              BEQ PEXIT      yes - exit
F7D7 85 36              STA CHAR       save it
F7D9 A5 48              LDA HOST       host port selected
F7DB F0 05              BEQ IFPR       no - try printer
F7DD A5 36              LDA CHAR       yes - get character
F7DF 20 F5F7            JSR SEND1      and transmit it
F7E2 A5 47     IFPR     LDA PRINTR     test printer flag
F7E4 F0 05              BEQ IFIEE      no - try GPIB flag
F7E6 A5 36              LDA CHAR       yes - get character
F7E8 20 02F8            JSR SEND2      and print it
F7EB A5 3C     IFIEE    LDA IEEEPR     GPIB as output ?
F7ED F0 05              BEQ PEXIT      no - exit
F7EF A5 36              LDA CHAR       yes - get character
F7F1 20 33F7            JSR WROB       and transmit it
F7F4 60        PEXIT    RTS
F7F5 AD 3080   SEND1    LDA ACIA1      test transmit enable
F7F8 29 02              AND #2         bit in ACIA
F7FA F0 F9              BEQ SEND1      not set - wait
F7FC A5 36              LDA CHAR       is set - put char
F7FE 8D 3180            STA ACIA1+1    in output buffer
F801 60                 RTS
F802 AD 3880   SEND2    LDA ACIA2      same for ACIA 2
F805 29 02              AND #2
F807 F0 F9              BEQ SEND2
F809 A5 36              LDA CHAR
F80B 8D 3980            STA ACIA2+1
F80E 60                 RTS
F80F A9 0D     CRLF     LDA #CR        print a carriage
F811 20 D3F7            JSR PRTCHR     return followed
F814 A9 0A              LDA #LF        by a linefeed
F816 20 D3F7            JSR PRTCHR
F819 60                 RTS
F81A A9 20     SPACE    LDA #$20       print a space
F81C 20 D3F7            JSR PRTCHR
F81F 60                 RTS
F820 68        PRINT    PLA            print message which
F821 AA                 TAX            follows the JSR PRINT
F822 68                 PLA            instruction
F823 A8                 TAY            recover calling
F824 20 4AF8            JSR PUSHSL     address
F827 86 43              STX SELECT     make select=address
F829 84 44              STY SELECT+1
F82B 20 89F8            JSR INCSL      increment it (TEX)
F82E 20 89F8   NEXTCH   JSR INCSL      increment select
F831 20 70F8            JSR GETSL      get byte pointed to
F834 C9 FF              CMP #ETX       is it ETX ?
F836 F0 06              BEQ ENDIT      yes - exit
F838 20 D3F7            JSR PRTCHR     no - print character
F83B 18                 CLC
F83C 90 F0              BCC NEXTCH     and loop
F83E A6 43     ENDIT    LDX SELECT     restore select
F840 A4 44              LDY SELECT+1   pointer
```

```
F842 20 5DF8                JSR POPSL
F845 98                     TYA                push address of ETX
F846 48                     PHA                onto stack
F847 8A                     TXA
F848 48                     PHA
F849 60                     RTS                and return to next
F84A 68         PUSHSL      PLA                pull return address
F84B 85 49                  STA RETURN         and save it
F84D 68                     PLA
F84E 85 4A                  STA RETURN+1
F850 A5 44                  LDA SELECT+1       push select onto
F852 48                     PHA                stack
F853 A5 43                  LDA SELECT
F855 48                     PHA
F856 A5 4A                  LDA RETURN+1       push return address
F858 48                     PHA                back onto the stack
F859 A5 49                  LDA RETURN
F85B 48                     PHA
F85C 60                     RTS
F85D 68         POPSL       PLA                save return address
F85E 85 49                  STA RETURN         from top of stack
F860 68                     PLA
F861 85 4A                  STA RETURN+1
F863 68                     PLA
F864 85 43                  STA SELECT         get select from stack
F866 68                     PLA
F867 85 44                  STA SELECT+1
F869 A5 4A                  LDA RETURN+1       push return back to
F86B 48                     PHA                the stack
F86C A5 49                  LDA RETURN
F86E 48                     PHA
F86F 60                     RTS
F870 A5 45      GETSL       LDA GETPTR         fetch byte pointed
F872 48                     PHA                to by select
F873 A6 46                  LDX GETPTR+1
F875 A5 43                  LDA SELECT
F877 85 45                  STA GETPTR         using GETPTR if
F879 A5 44                  LDA SELECT+1       select is not on
F87B 85 46                  STA GETPTR+1       the zero page
F87D A0 00                  LDY #0
F87F B1 45                  LDA (GETPTR),Y
F881 A8                     TAY
F882 68                     PLA
F883 85 45                  STA GETPTR
F885 86 46                  STX GETPTR+1
F887 98                     TYA
F888 60                     RTS
F889 E6 43      INCSL       INC SELECT         increment select
F88B D0 02                  BNE BRANCH         pointer
F88D E6 44                  INC SELECT+1
F88F 60         BRANCH      RTS
F890 AD 3080    GETCHR      LDA ACIA1          get character from
F893 29 01                  AND #1             host serial port
F895 F0 F9                  BEQ GETCHR         loop unitl bit set
F897 AD 3180                LDA ACIA1+1        get character
F89A 60                     RTS
                ;
                ; START OF ASCII MINICON ROUTINES
```

```
                     ;
F89B 20 73F7   MNSTRT   JSR VINIT       initialise variables
F89E A9 FF              LDA #ON
F8A0 85 30              STA NEWFLG      minicon routines to
                     ;                  return ascii output
F8A2 A5 3B              LDA IEEFLG
F8A4 F0 0D              BEQ SER3
F8A6 A9 FF              LDA #ON
F8A8 85 3C              STA IEEEPR
F8AA 20 87F7            JSR HOSTOF
F8AD 20 91F7            JSR PRTOFF
F8B0 4C B9F8            JMP MINCON
F8B3 20 96F7   SER3     JSR RTSON       RTS high, input OK
F8B6 20 82F7            JSR HOSTON      host port for output
F8B9 20 30F0   MINCON   JSR INIT        initialise VIA's
F8BC A2 00              LDX #0          buffer counter=0
F8BE 86 32              STX BUFFEN
F8C0 86 31              STX NEGFLG      clear negative flag
F8C2 A5 3B              LDA IEEFLG
F8C4 F0 0D              BEQ INLOOP
F8C6 20 28F7   INLP     JSR RDOB
F8C9 9D 0003            STA BUFF,X
F8CC C5 3E              CMP DELIM
F8CE F0 10              BEQ DECODE
F8D0 E8                 INX
F8D1 D0 F3              BNE INLP
F8D3 20 90F8   INLOOP   JSR GETCHR      get character
F8D6 9D 0003            STA BUFF,X      store in buffer
F8D9 C5 3E              CMP DELIM       is it the delimiter
F8DB F0 03              BEQ DECODE      yes - process buffer
F8DD E8                 INX             no - increment pointer
F8DE D0 F3              BNE INLOOP      and loop
F8E0 86 32     DECODE   STX BUFFEN      save buffer fill pntr
F8E2 A2 00              LDX #0          get first character
F8E4 BD 0003            LDA BUFF,X
F8E7 38                 SEC             prepare to subtract
F8E8 E9 41              SBC #$41        subtract $41, 'A'=0
F8EA 90 04              BCC BADCHR      char was less than $41
F8EC C9 1B              CMP #27         is character'Z'
F8EE 90 03              BCC GOODCH
F8F0 4C 26F9   BADCHR   JMP INERR       input command error
F8F3 A8        GOODCH   TAY             char as index to table of
               ;                        second chars
F8F4 E8                 INX
F8F5 BD 0003            LDA BUFF,X      next char from buffer
F8F8 D9 91FD            CMP CHTAB,Y     agree with 2 nd char
F8FB D0 F3              BNE BADCHR      no
F8FD 20 3AF9            JSR GETIND      next char as index
F900 B0 11              BCS BADOPT      no. was out of range
F902 0A                 ASL A           x2 for low, high byte
F903 18                 CLC
F904 79 ABFD            ADC OFFSET,Y    add offset from table
F907 A8                 TAY             use as index to jump
F908 C8                 INY             table,high byte first
F909 B9 C5FD            LDA JUMPTB,Y    form indexed jump
F90C 48                 PHA             from vectors in
F90D 88                 DEY             table JUMPTB
F90E B9 C5FD            LDA JUMPTB,Y
```

```
F911 48                     PHA
F912 60                     RTS
F913 20 20F8    BADOPT      JSR PRINT
F916 7F424144               DB TEX,'BAD OPTION',CR,ETX
F91A 204F5054
F91E 494F4E0D
F922 FF
F923 4C B9F8                JMP MINCON
F926 20 20F8    INERR       JSR PRINT
F929 7F424144               DB TEX,'BAD COMMAND',CR,ETX
F92D 20434F4D
F931 4D414E44
F935 0DFF
F937 4C B9F8                JMP MINCON
F93A A2 02      GETIND      LDX #2
F93C BD 0003                LDA BUFF,X
F93F 38                     SEC
F940 E9 30                  SBC #$30
F942 90 04                  BCC GBAD
F944 C9 06                  CMP #06
F946 90 02                  BCC GGOOD
F948 38         GBAD        SEC
F949 60                     RTS
F94A 18         GGOOD       CLC
F94B 60                     RTS
                ;
                ; ASCII MINICON ROUTNIES - VARIABLE
                ; CONVERSION TO OLD FORMAT
                ;
F94C 20 60FA    STPM1       JSR THREIN      stepper motor no 1
F94F B0 03                  BCS J1          get 3 input params
F951 20 84F0                JSR USR1        and jump to usr(1)
F954 4C B9F8    J1          JMP MINCON
F957 20 60FA    STPM2       JSR THREIN      stepper motor no 2
F95A B0 07                  BCS J2          as above but set
F95C A9 01                  LDA #1          high byte of board
F95E 85 21                  STA VA+1        address for quick
F960 20 84F0                JSR USR1        return
F963 4C B9F8    J2          JMP MINCON
F966 20 60FA    STPM3       JSR THREIN      stepper motor no 3
F969 B0 F8                  BCS J2          is used to set accln
F96B 20 EFF0                JSR USR11       & decln rates for
F96E 4C B9F8                JMP MINCON      no. 4
F971 20 6FFA    STPM4       JSR TWOIN       stepper motor 4,
F974 B0 ED                  BCS J2          only requires board
F976 20 28F1                JSR USR12       address & no. steps
F979 4C B9F8                JMP MINCON
F97C 20 6FFA    CO1         JSR TWOIN       scalar no 1 (counter)
F97F B0 0B                  BCS J3          two inputs required
F981 A5 22                  LDA N           but second variable
F983 85 24                  STA T           needs to be made T%
F985 A5 23                  LDA N+1         variable
F987 85 25                  STA T+1         usr(2)
F989 20 E0F1                JSR USR2
F98C 4C B9F8    J3          JMP MINCON
F98F 20 6FFA    CO2         JSR TWOIN       scalar no 2
F992 B0 0F                  BCS J4          high byte
F994 A5 22                  LDA N           of address set
```

```
F996 85 24              STA T
F998 A5 23              LDA N+1
F99A 85 25              STA T+1
F99C A9 01              LDA #1
F99E 85 21              STA VA+1
F9A0 20 E0F1            JSR USR2
F9A3 4C B9F8   J4       JMP MINCON
F9A6 20 7DFA   ADC1     JSR ONEIN      A to D no 1
F9A9 B0 03              BCS J5         board address reqd
F9AB 20 4DF2            JSR USR5       usr(5)
F9AE 4C B9F8   J5       JMP MINCON
F9B1 20 60FA   ADC2     JSR THREIN     A to D no 2
F9B4 B0 03              BCS J6         multilpexed A to D
F9B6 20 9FF5            JSR USR10      requires more input
F9B9 4C B9F8   J6       JMP MINCON
F9BC 20 6FFA   DAC1     JSR TWOIN      D to A no 1
F9BF B0 03              BCS J7         two inputs only
F9C1 20 24F2            JSR USR4       usr(4)
F9C4 4C B9F8   J7       JMP MINCON
F9C7 20 6FFA   DAC2     JSR TWOIN      D to A no 2
F9CA B0 07              BCS J8         address high byte set
F9CC A9 02              LDA #2         for 4 channel board
F9CE 85 21              STA VA+1
F9D0 20 24F2            JSR USR4
F9D3 4C B9F8   J8       JMP MINCON
F9D6 20 7DFA   GP1      JSR ONEIN      GPI no 1
F9D9 B0 03              BCS J9         only address required
F9DB 20 24F5            JSR USR8       usr(8)
F9DE 4C B9F8   J9       JMP MINCON
F9E1 4C 11FD   MO0      JMP OLDST      old monitor (IEEE)
F9E4 4C 11FD   MO1      JMP OLDST      old minicon (IEEE)
F9E7 4C 9FFC   MO2      JMP IEEMON     IEEE ascii monitor
F9EA 4C A6FC   MO3      JMP IEECAM     IEEE ascii minicom
F9ED 4C 69F5   MO4      JMP MONITR     RS232 monitor
F9F0 4C ADFC   MO5      JMP IEESER     IEEE to RS232 routine
F9F3 20 60FA   VP1      JSR THREIN     vector plot no 1
F9F6 B0 03              BCS J10
F9F8 20 B7F3            JSR USR6       usr(6)
F9FB 4C B9F8   J10      JMP MINCON
F9FE 20 60FA   VP2      JSR THREIN     vector plot no 2
FA01 B0 03              BCS J11
FA03 20 6FF3            JSR USR7       usr(7)
FA06 4C B9F8   J11      JMP MINCON
FA09 20 7EF0   RD1      JSR USR0       usr(0)
FA0C 4C B9F8            JMP MINCON     no inputs
FA0F 20 7DFA   RD2      JSR ONEIN      usr(3)
FA12 B0 03              BCS J12        address only required
FA14 20 07F2            JSR USR3
FA17 4C B9F8   J12      JMP MINCON
FA1A 20 87F7   PR1      JSR HOSTOF     output to printer
FA1D 20 8CF7            JSR PRTON      port only
FA20 A9 00              LDA #OFF
FA22 85 3C              STA IEEEPR
FA24 A2 03              LDX #3
FA26 E8        OUTLP    INX
FA27 BD 0003            LDA BUFF,X     char from input buff
FA2A 48                 PHA            save it temporarily
FA2B 20 D3F7            JSR PRTCHR     print it
```

```
FA2E 68                PLA          return character
FA2F C9 0D             CMP #13      carriage return ?
FA31 D0 F3             BNE OUTLP    no - continue
FA33 A5 3D             LDA PDELM    printer delimiter
FA35 C9 0D             CMP #13
FA37 F0 03             BEQ PREX     if CR have printed it
FA39 20 D3F7           JSR PRTCHR   else print current
                ;                   printer delimiter
FA3C 20 91F7    PREX   JSR PRTOFF   turn printer off
FA3F A5 3B             LDA IEEFLG   see which port is host
FA41 F0 07             BEQ PSER     serial port - branch
FA43 A9 FF             LDA #ON      IEEE488 port as host
FA45 85 3C             STA IEEEPR
FA47 4C 4DFA           JMP PREX2
FA4A 20 82F7    PSER   JSR HOSTON   serial port 1 as host
FA4D 20 3EFB    PREX2  JSR REPLOK   reply to user
FA50 4C B9F8           JMP MINCON
FA53 A2 04      PR2    LDX #4       sets printer delimiter
FA55 BD 0003           LDA BUFF,X   to 5th character in
FA58 85 3D             STA PDELM    buffer
FA5A 20 3EFB           JSR REPLOK
FA5D 4C B9F8           JMP MINCON
FA60 A9 03      THREIN LDA #3       buffer empty pointer
FA62 85 34             STA BPOINT   to 4 th character
FA64 85 40             STA INCNT    three inputs required
FA66 20 9CFA           JSR INNVAL   input values
FA69 B0 01             BCS VERROR   carry set gives error
FA6B 60                RTS
FA6C 4C C5FB    VERROR JMP VINERR   error in variables
FA6F A9 03      TWOIN  LDA #3
FA71 85 34             STA BPOINT
FA73 A9 02             LDA #2       two inputs required
FA75 85 40             STA INCNT
FA77 20 9CFA           JSR INNVAL
FA7A B0 F0             BCS VERROR
FA7C 60                RTS
FA7D A9 03      ONEIN  LDA #3       one input required
FA7F 85 34             STA BPOINT
FA81 A9 01             LDA #1
FA83 85 40             STA INCNT
FA85 20 9CFA           JSR INNVAL
FA88 B0 E2             BCS VERROR
FA8A 60                RTS
FA8B A6 34      ADVAN  LDX BPOINT   buffer empty pointer
FA8D E8         ADLP   INX          step it on one place
FA8E BD 0003           LDA BUFF,X   get character
FA91 C9 2C             CMP #C','    is it a comma
FA93 F0 04             BEQ ADEND    yes - return
FA95 C9 0D             CMP #13      is it CR
FA97 D0 F4             BNE ADLP     no loop
FA99 86 34      ADEND  STX BPOINT   save empty pointer
FA9B 60                RTS
FA9C A0 00      INNVAL LDY #0       loop counter
FA9E 98         ILOOP  TYA          save Y register
FA9F 48                PHA          on stack
FAA0 20 8BFA           JSR ADVAN    empty pointer to ,
FAA3 20 BAFA           JSR ASCBIN   decimal ascii to bin
FAA6 68                PLA
```

```
FAA7 A8                   TAY             recover Y register
FAA8 A5 38                LDA LOWBIT      low byte of input
FAAA 99 2000              STA VA,Y        variable list+Y
FAAD A5 37                LDA HGHBIT      high byte
FAAF 99 2100              STA VA+1,Y      in variable list
FAB2 C8                   INY             increment Y register
FAB3 C8                   INY             twice
FAB4 C6 40                DEC INCNT       decrement  counter
FAB6 D0 E6                BNE ILOOP       if non zero loop
FAB8 18                   CLC
FAB9 60                   RTS
FABA A9 00      ASCBIN    LDA #0          initialise low and
FABC 85 38                STA LOWBIT      high bytes
FABE 85 37                STA HGHBIT
FAC0 A6 34                LDX BPOINT      get empty pointer
FAC2 CA                   DEX             to charcter to left
FAC3 BD 0003              LDA BUFF,X      and fetch it
FAC6 20 2FFB              JSR TSTEND      reached end of number
FAC9 B0 43                BCS EXIT3
FACB 85 38                STA LOWBIT      -$30 gives low byte
FACD CA                   DEX             get next character
FACE BD 0003              LDA BUFF,X
FAD1 20 2FFB              JSR TSTEND      test it
FAD4 B0 38                BCS EXIT3
FAD6 C9 00                CMP #0
FAD8 F0 0F                BEQ B2
FADA A8                   TAY             index to table
FADB B9 31FE              LDA TABLE1,Y    number equiv in hex
FADE 18                   CLC
FADF 65 38                ADC LOWBIT      add to low byte
FAE1 85 38                STA LOWBIT
FAE3 A5 37                LDA HGHBIT      add carry to high
FAE5 69 00                ADC #0          byte
FAE7 85 37                STA HGHBIT
FAE9 A9 3B      B2        LDA #TABLE2     tpointer points to
FAEB 85 41                STA TPOINT      table2
FAED A9 FE                LDA #(TABLE2/256)
FAEF 85 42                STA TPOINT+1
FAF1 20 10FB              JSR ADD_IN      add next character
FAF4 B0 18                BCS EXIT3       to low and high bytes
FAF6 A9 4F                LDA #TABLE3     repeat with table3
FAF8 85 41                STA TPOINT
FAFA A9 FE                LDA #(TABLE3/256)
FAFC 85 42                STA TPOINT+1
FAFE 20 10FB              JSR ADD_IN
FB01 B0 0B                BCS EXIT3
FB03 A9 63                LDA #TABLE4     and table4
FB05 85 41                STA TPOINT
FB07 A9 FE                LDA #(TABLE4/256)
FB09 85 42                STA TPOINT+1
FB0B 20 10FB              JSR ADD_IN
FB0E 18         EXIT3     CLC
FB0F 60                   RTS
FB10 CA         ADD_IN    DEX             move to next left
FB11 BD 0003              LDA BUFF,X
FB14 20 2FFB              JSR TSTEND      test it
FB17 B0 15                BCS EXIT2
FB19 C9 00                CMP #0
```

```
FB1B F0 10                  BEQ EXIT1
FB1D 0A                     ASL A           mulitiply by two
FB1E A8                     TAY             use it as index
FB1F B1 41                  LDA (TPOINT),Y  equiv in hex from
FB21 18                     CLC             table
FB22 65 38                  ADC LOWBIT      add to low byte
FB24 85 38                  STA LOWBIT
FB26 C8                     INY             high byte part and
FB27 B1 41                  LDA (TPOINT),Y  add to high byte
FB29 65 37                  ADC HGHBIT
FB2B 85 37                  STA HGHBIT
FB2D 18         EXIT1       CLC
FB2E 60         EXIT2       RTS
FB2F C9 2C      TSTEND      CMP #C','       is character a ,
FB31 F0 09                  BEQ COMMA       yes - set carry
FB33 C9 20                  CMP #$20        is it a space
FB35 F0 05                  BEQ COMMA
FB37 38                     SEC             prepare to subtract
FB38 E9 30                  SBC #$30        subtract $30
FB3A 18                     CLC
FB3B 60                     RTS
FB3C 38         COMMA       SEC
FB3D 60                     RTS
                ;
                ; ASCII ROUTINE FOR BOARD ROUTINE COMPLETED
                ; OK AND ALL QUICK REPLIES
                ;
FB3E 20 20F8    REPLOK      JSR PRINT       routine completed ok,
FB41 7F4F4B0D               DB TEX,'OK',CR,ETX
FB45 FF
FB46 60                     RTS
                ;
                ; BINARY TO ASCII DECIMAL CONVERSION
                ;
FB47 A9 00      BINASC      LDA #0          zero output buffer
FB49 8D 5003                STA OUTBUF      double babble method
FB4C 8D 5103                STA OUTBUF+1
FB4F 8D 5203                STA OUTBUF+2
FB52 A2 08                  LDX #8
FB54 A9 80                  LDA #128
FB56 48         BLOOP       PHA             save acc on stack
FB57 25 37                  AND HGHBIT      high bit of high byte
FB59 F0 03                  BEQ BNEXT       if clear branch
FB5B 20 84FB                JSR ADD1        else add one to
FB5E 20 A5FB    BNEXT       JSR TIMTWO      output x 2
FB61 68                     PLA             return acc
FB62 4A                     LSR A           left one place
FB63 CA                     DEX             test loop counter
FB64 D0 F0                  BNE BLOOP       if non zero loop
FB66 A2 07                  LDX #7          now do seven bits of
FB68 A9 80                  LDA #128        low byte
FB6A 48         BLOOP2      PHA
FB6B 25 38                  AND LOWBIT
FB6D F0 03                  BEQ BNEXT2
FB6F 20 84FB                JSR ADD1
FB72 20 A5FB    BNEXT2      JSR TIMTWO
FB75 68                     PLA
FB76 4A                     LSR A
```

```
FB77 CA                     DEX
FB78 D0 F0                  BNE BLOOP2
FB7A A9 01                  LDA #1          test low bit and add
FB7C 25 38                  AND LOWBIT      one to output if set
FB7E F0 03                  BEQ BEXIT
FB80 20 84FB                JSR ADD1
FB83 60           BEXIT     RTS
FB84 F8           ADD1      SED             addition in BCD
FB85 18                     CLC             clear carry flag
FB86 AD 5003                LDA OUTBUF      add one to low byte
FB89 69 01                  ADC #1
FB8B 8D 5003                STA OUTBUF
FB8E 90 12                  BCC BEND
FB90 AD 5103                LDA OUTBUF+1    carry to next byte
FB93 69 00                  ADC #0
FB95 8D 5103                STA OUTBUF+1
FB98 90 08                  BCC BEND
FB9A AD 5203                LDA OUTBUF+2    carry to next byte
FB9D 69 00                  ADC #0          gives 24 bit BCD
FB9F 8D 5203                STA OUTBUF+2    number
FBA2 18           BEND      CLC
FBA3 D8                     CLD
FBA4 60                     RTS
FBA5 F8           TIMTWO    SED             output buffer
FBA6 18                     CLC             by two using BCD
FBA7 AD 5003                LDA OUTBUF      add it to itself
FBAA 6D 5003                ADC OUTBUF
FBAD 8D 5003                STA OUTBUF
FBB0 AD 5103                LDA OUTBUF+1
FBB3 6D 5103                ADC OUTBUF+1
FBB6 8D 5103                STA OUTBUF+1
FBB9 AD 5203                LDA OUTBUF+2
FBBC 6D 5203                ADC OUTBUF+2
FBBF 8D 5203                STA OUTBUF+2
FBC2 18                     CLC
FBC3 D8                     CLD
FBC4 60                     RTS
FBC5 20 20F8      VINERR    JSR PRINT       print VARIABLE ERROR
FBC8 7F564152               DB TEX,'VARIABLE ERROR',CR,ETX
FBCC 4941424C
FBD0 45204552
FBD4 524F520D
FBD8 FF
FBD9 60                     RTS
                  ;
                  ; OUTPUT ASCII VALUE OF OUTPUT BYTES
                  ;
FBDA A5 31        REPLAN    LDA NEGFLG      negative flag set
FBDC F0 14                  BEQ CONT        if number is negative
FBDE A5 12                  LDA TMP0        if negative perform
FBE0 18                     CLC             two's complement
FBE1 49 FF                  EOR #$FF        first before
FBE3 69 01                  ADC #1          converting to ascii
FBE5 48                     PHA
FBE6 A5 13                  LDA TMP0+1
FBE8 49 FF                  EOR #$FF
FBEA 69 00                  ADC #0
FBEC 85 13                  STA TMP0+1
```

```
FBEE 68                 PLA
FBEF 85 12              STA TMPO
FBF1 18                 CLC
FBF2 A9 00     CONT     LDA #0          clear leading zeros
FBF4 85 3F              STA LEADZ       flag
FBF6 A5 12              LDA TMPO        output bytes to
FBF8 85 38              STA LOWBIT      variable LOWBIT &
FBFA A5 13              LDA TMPO+1      HGHBIT
FBFC 85 37              STA HGHBIT
FBFE 20 47FB            JSR BINASC      convert to decimal
FC01 A5 31              LDA NEGFLG
FC03 F0 05              BEQ CONT2
FC05 A9 2D              LDA #C'-'       print minus sign
FC07 20 D3F7            JSR PRTCHR      if negative
FC0A 18        CONT2    CLC
FC0B A2 03              LDX #3          three BCD bytes to be
FC0D CA        BRAN     DEX             output
FC0E 18                 CLC
FC0F A9 F0              LDA #$F0        4 MSB's of byte
FC11 3D 5003            AND OUTBUF,X    transmitted first
FC14 4A                 LSR A
FC15 4A                 LSR A
FC16 4A                 LSR A
FC17 4A                 LSR A
FC18 69 30              ADC #$30        as ascii
FC1A 20 3BFC            JSR PRNUM
FC1D A9 0F              LDA #$0F        then 4 LSB's
FC1F 18                 CLC
FC20 3D 5003            AND OUTBUF,X
FC23 69 30              ADC #$30
FC25 20 3BFC            JSR PRNUM
FC28 E0 00              CPX #0
FC2A D0 E1              BNE BRAN        and loop for next two
FC2C A5 3F              LDA LEADZ       if still not printed
FC2E D0 05              BNE END         anything number=0 so
FC30 A9 30              LDA #$30        print ascii 0
FC32 20 D3F7            JSR PRTCHR
FC35 A9 0D     END      LDA #CR         end with cr
FC37 20 D3F7            JSR PRTCHR
FC3A 60                 RTS
FC3B C9 30     PRNUM    CMP #$30        is number a zero
FC3D D0 07              BNE CONT3
FC3F A5 3F              LDA LEADZ
FC41 D0 01              BNE OUT         no so print it
FC43 60                 RTS             else don't print it
FC44 A9 30     OUT      LDA #$30
FC46 20 D3F7   CONT3    JSR PRTCHR
FC49 A9 FF              LDA #ON         have now printed char
FC4B 85 3F              STA LEADZ       so zeros are no
FC4D 60                 RTS             longer leading
               ;
               ; RESET ROUTINE ENTERED ON 6502 RESET
               ;
FC4E 78        RESET    SEI             disable interrupts
FC4F D8                 CLD             binary addition etc
FC50 A2 FF              LDX #$FF
FC52 9A                 TXS
               ;
```

```
                    ; INITIALISE VIAS
                    ;
FC53 20 30F0                JSR INIT
FC56 A9 00                  LDA #OFF
FC58 85 3B                  STA IEEFLG
FC5A 85 30                  STA NEWFLG
FC5C A9 0D                  LDA #$0D        printer and input
FC5E 85 3D                  STA PDELM       delimiters to CR
FC60 85 3E                  STA DELIM
FC62 A2 05                  LDX #5
FC64 BD 7BFD   RSLP         LDA RESTAB,X    copy reset, IRQ and
FC67 95 00                  STA 0,X         NMI vectors to RAM
FC69 CA                     DEX
FC6A 10 F8                  BPL RSLP
                    ;
                    ; 68488 GPIA RESET
                    ;
FC6C A9 80                  LDA #$80
FC6E 8D 2380                STA $8023
FC71 A9 00                  LDA #0
FC73 8D 2380                STA $8023
FC76 8D 2080                STA $8020
FC79 8D 2280                STA $8022
FC7C AD 2480                LDA $8024
FC7F 29 1F                  AND #$1F
FC81 8D 2480                STA $8024
                    ;
                    ; 6850 ACIAS RESET
                    ;
FC84 20 AAF7                JSR RSRES
                    ;
                    ; DETERMINE START ROUTINE FROM TOP 3 BITS
                    ; OF ADDRESS SWITCH
                    ;
FC87 AD 2480                LDA $8024       top 3 bits of address
FC8A 4A                     LSR A           switch as option
FC8B 4A                     LSR A           select number
FC8C 4A                     LSR A
FC8D 4A                     LSR A           have number x 2
FC8E 29 0D                  AND #$0E        clear other bits
FC90 A8                     TAY
FC91 B9 81FD                LDA STRTAB,Y    low byte of routine
FC94 85 08                  STA $08         save it
FC96 C8                     INY
FC97 B9 81FD                LDA STRTAB,Y    high byte
FC9A 85 09                  STA $09
FC9C 6C 0800                JMP ($0008)     jump to this routine
                    ;
                    ; SOME OF THE POSSIBLE STARTING ROUTINES
                    ;
                    ; GPIB BASED ASCII MONITOR
                    ;
FC9F A9 FF     IEEMON       LDA #ON
FCA1 85 3B                  STA IEEFLG
FCA3 4C 69F5                JMP MONITR
                    ;
                    ; GPIB BASED ASCII MINICON
                    ;
```

```
FCA6 A9 FF      IEECAM   LDA #ON
FCA8 85 3B               STA IEEFLG
FCAA 4C 9BF8             JMP MNSTRT
                ;
                ; GPIB TO SERIAL ROUTINE
                ;
FCAD 20 82F7    IEESER   JSR HOSTON      data recieved via
FCB0 A9 FF               LDA #ON         GPIB is transmitted
FCB2 85 3B               STA IEEFLG      to host serial port
FCB4 20 91F7             JSR PRTOFF      until three
FCB7 20 96F7             JSR RTSON       consequetive ESC's
FCBA A9 00               LDA #OFF        are received
FCBC 85 3C               STA IEEEPR
FCBE 20 28F7    IEERPT   JSR RDOB        character from GPIB
FCC1 C9 1B               CMP #$1B        is character ESC ?
FCC3 F0 06               BEQ ESCAPE      yes - branch
FCC5 20 D3F7             JSR PRTCHR      no - transmit it
FCC8 4C BEFC             JMP IEERPT      then loop
FCCB 20 28F7    ESCAPE   JSR RDOB        get next character
FCCE C9 1B               CMP #$1B        is it an ESC
FCD0 D0 0A               BNE ESEXT1      no - branch
FCD2 20 28F7             JSR RDOB        get next character
FCD5 C9 1B               CMP #$1B        is it an ESC
FCD7 D0 10               BNE ESEXT2      no branch
FCD9 4C 9BF8             JMP MNSTRT      three ESC's recieved
                ;                        - return to minicon
FCDC 48         ESEXT1   PHA             save non ESC char
FCDD A9 1B               LDA #$1B        transmit ESC
FCDF 20 D3F7             JSR PRTCHR
FCE2 68                  PLA
FCE3 20 D3F7             JSR PRTCHR      then non ESC char
FCE6 4C BEFC             JMP IEERPT      and back to loop
FCE9 48         ESEXT2   PHA             as above but send
FCEA A9 1B               LDA #$1B        two ESC's before
FCEC 20 D3F7             JSR PRTCHR      character
FCEF A9 1B               LDA #$1B
FCF1 20 D3F7             JSR PRTCHR
FCF4 68                  PLA
FCF5 20 D3F7             JSR PRTCHR
FCF8 4C BEFC             JMP IEERPT
                ;
                ; HOST SERIAL PORT TO PRINTER SERIAL PORT
                ;
FCFB 20 87F7    SERSER   JSR HOSTOF
FCFE 20 8CF7             JSR PRTON
FD01 A9 00               LDA #OFF
FD03 85 3C               STA IEEEPR
FD05 20 96F7             JSR RTSON
FD08 20 90F8    SERRPT   JSR GETCHR
FD0B 20 D3F7             JSR PRTCHR
FD0E 4C 08FD             JMP SERRPT
                ;
                ; ORIGINAL BINARY MINICON USER INTERFACE
                ;
FD11 20 30F0    OLDST    JSR INIT        initialise VIA's
FD14 A9 10               LDA #$10        set serial poll byte
FD16 8D 2580             STA SRQ
FD19 20 28F7             JSR RDOB        get byte from GPIB
```

```
FD1C 85 10              STA USRNO      save it as the usr
FD1E A2 00              LDX #0         number - then get
FD20 20 28F7   GLP      JSR RDOB       six more bytes and
FD23 95 20              STA VLIST,X    save in VLIST
FD25 E8                 INX
FD26 E0 06              CPX #6
FD28 D0 F6              BNE GLP
FD2A 20 30FD            JSR BEGIN      choose the routine
FD2D 4C 11FD            JMP OLDST      then start again
               ;
               ;

FD30 A9 00     BEGIN    LDA #0         set serial poll byte
FD32 8D 2580            STA SRQ
FD35 85 1F              STA STATUS     and status byte
               ;
               ; CHOOSE ROUTINE
               ;
FD37 A5 10     CHOOSE   LDA USRNO
FD39 A2 0C              LDX #(NCMDS-1)
FD3B DD 54FD   CHLP1    CMP USR,X      compare first char
FD3E D0 0E              BNE CHLP2      with table USR
FD40 8A                 TXA
FD41 0A                 ASL A
FD42 AA                 TAX
FD43 E8                 INX
FD44 BD 61FD            LDA ADRR,X     get address low
FD47 48                 PHA            and high bytes
FD48 CA                 DEX            put on the stack
FD49 BD 61FD            LDA ADRR,X
FD4C 48                 PHA            and jump to it with
FD4D 60                 RTS            an RTS instruction
FD4E CA        CHLP2    DEX
FD4F 10 EA              BPL CHLP1      loop until agreement
FD51 6C 0600            JMP (USRCMD)   no agreement - user
               ;                       supplied rouine
               ; TABLE OF FIRST CHARACTERS
               ;
FD54 00010203  USR      DB 0,1,2,3,4,5,6,7,8,9,10,11,12
FD58 04050607
FD5C 08090A0B
FD60 0C

               ;
               ; ADDRESSES OF ROUTINES
               ;
FD61 7DF0      ADRR     ADDR (USR0-1)
FD63 83F0               ADDR (USR1-1)
FD65 DFF1               ADDR (USR2-1)
FD67 06F2               ADDR (USR3-1)
FD69 23F2               ADDR (USR4-1)
FD6B 4CF2               ADDR (USR5-1)
FD6D B6F3               ADDR (USR6-1)
FD6F 6EF3               ADDR (USR7-1)
FD71 23F5               ADDR (USR8-1)
FD73 68F5               ADDR (USR9-1)
FD75 9EF5               ADDR (USR10-1)
FD77 EEF0               ADDR (USR11-1)
FD79 27F1               ADDR (USR12-1)

               ;
```

```
                    ; TABLE OF RESET, IRQ AND NMI VECTORS
                    ;
FD7B 7DFE           RESTAB   ADDR IRQVEC
FD7D 7EFE                    ADDR NMIVEC
FD7F 9BF8                    ADDR MNSTRT
                    ;
                    ; TABLE OF VECTORS FOR START UP IN ASCII
                    ;
FD81 11FD           STRTAB   ADDR OLDST
FD83 11FD                    ADDR OLDST
FD85 9FFC                    ADDR IEEMON
FD87 A6FC                    ADDR IEECAM
FD89 69F5                    ADDR MONITR
FD8B 9BF8                    ADDR MNSTRT
FD8D ADFC                    ADDR IEESER
FD8F FBFC                    ADDR SERSER
                    ;
                    ; TABLE OF SECOND CHARACTERS FOR ASCII
                    ; ROUTINES
                    ;
FD91 44424F41       CHTAB    DB 'DBOAEFPHIJKLONORQDTTUPWXYZ'
FD95 45465048
FD99 494A4B4C
FD9D 4F4E4F52
FDA1 51445454
FDA5 55505758
FDA9 595A
                    ;
                    ; TABLE OF OFFSETS FROM START OF JUMPTB
                    ;
FDAB 00000C18       OFFSET   DB 0,0,12,24,0,0,36,0,0,0,0,0,48,0,0,96,0
FDAF 00002400
FDB3 00000000
FDB7 30000060
FDBB 00
FDBC 3C480000                DB 60,72,0,0,84,0,0,0,0
FDC0 54000000
FDC4 00
                    ;
                    ; TABLE OF ADDRESSES OF ASCII MINICON
                    ; ROUTINES
                    ;
FDC5 A5F9           JUMPTB   ADDR (ADC1-1)
FDC7 A5F9                    ADDR (ADC1-1)
FDC9 B0F9                    ADDR (ADC2-1)
FDCB 0000                    ADDR 0
FDCD 0000                    ADDR 0
FDCF 0000                    ADDR 0
FDD1 7BF9                    ADDR (CO1-1)
FDD3 7BF9                    ADDR (CO1-1)
FDD5 8EF9                    ADDR (CO2-1)
FDD7 0000                    ADDR 0
FDD9 0000                    ADDR 0
FDDB 0000                    ADDR 0
FDDD BBF9                    ADDR (DAC1-1)
FDDF BBF9                    ADDR (DAC1-1)
FDE1 C6F9                    ADDR (DAC2-1)
FDE3 0000                    ADDR 0
```

```
FDE5 0000                  ADDR 0
FDE7 0000                  ADDR 0
FDE9 D5F9                  ADDR (GP1-1)
FDEB D5F9                  ADDR (GP1-1)
FDED D5F9                  ADDR (GP1-1)
FDEF 0000                  ADDR 0
FDF1 0000                  ADDR 0
FDF3 0000                  ADDR 0
FDF5 E0F9                  ADDR (MO0-1)
FDF7 E3F9                  ADDR (MO1-1)
FDF9 E6F9                  ADDR (MO2-1)
FDFB E9F9                  ADDR (MO3-1)
FDFD ECF9                  ADDR (MO4-1)
FDFF EFF9                  ADDR (MO5-1)
FE01 08FA                  ADDR (RD1-1)
FE03 08FA                  ADDR (RD1-1)
FE05 0EFA                  ADDR (RD2-1)
FE07 0000                  ADDR 0
FE09 0000                  ADDR 0
FE0B 0000                  ADDR 0
FE0D 4BF9                  ADDR (STPM1-1)
FE0F 4BF9                  ADDR (STPM1-1)
FE11 56F9                  ADDR (STPM2-1)
FE13 65F9                  ADDR (STPM3-1)
FE15 70F9                  ADDR (STPM4-1)
FE17 0000                  ADDR 0
FE19 F2F9                  ADDR (VP1-1)
FE1B F2F9                  ADDR (VP1-1)
FE1D FDF9                  ADDR (VP2-1)
FE1F 0000                  ADDR 0
FE21 0000                  ADDR 0
FE23 0000                  ADDR 0
FE25 19FA                  ADDR (PR1-1)
FE27 19FA                  ADDR (PR1-1)
FE29 52FA                  ADDR (PR2-1)
FE2B 0000                  ADDR 0
FE2D 0000                  ADDR 0
FE2F 0000                  ADDR 0
                ;
                ; TABLES OF BINARY EQUIVALENTS OF DECIMAL
                ;
FE31 000A141E   TABLE1     DB 0,10,20,30,40,50,60,70,80,90
FE35 28323C46
FE39 505A
FE3B 0000       TABLE2     ADDR 0
FE3D 6400                  ADDR 100
FE3F C800                  ADDR 200
FE41 2C01                  ADDR 300
FE43 9001                  ADDR 400
FE45 F401                  ADDR 500
FE47 5802                  ADDR 600
FE49 BC02                  ADDR 700
FE4B 2003                  ADDR 800
FE4D 8403                  ADDR 900
FE4F 0000       TABLE3     ADDR 0
FE51 E803                  ADDR 1000
FE53 D007                  ADDR 2000
FE55 B80B                  ADDR 3000
```

```
FE57 A00F                   ADDR 4000
FE59 8813                   ADDR 5000
FE5B 7017                   ADDR 6000
FE5D 581B                   ADDR 7000
FE5F 401F                   ADDR 8000
FE61 2823                   ADDR 9000
FE63 0000        TABLE4     ADDR 0
FE65 1027                   ADDR 10000
FE67 204E                   ADDR 20000
FE69 3075                   ADDR 30000
FE6B 409C                   ADDR 40000
FE6D 50C3                   ADDR 50000
FE6F 60EA                   ADDR 60000
FE71 60EA                   ADDR 60000
FE73 60EA                   ADDR 60000
FE75 60EA                   ADDR 60000
                 ;
                 ; IRQ AND NMI HANDLERS
                 ;
FE77 6C 0000     IRQ        JMP ($00)       6502 here on interrupt
FE7A 6C 0200     NMI        JMP ($02)       and NMI - jump to RAM
                 ;                          vector
FE7D 40          IRQVEC     RTI             which point here
FE7E 40          NMIVEC     RTI             initially. User can
                 ;                          alter RAM vectors
```

```
                    ;
                    ; MONITOR. PROVIDES MEMORY DUMP AND
                    ; DISASSEMBLER VIA GPIB OR RS232 INTERFACES.
                    ; ASCII TRANSMISSION IS USED WITH TERMINAL
                    ; ECHO
                    ;
                    ; M.J. HILL 1985
                    ;
                    ; VARIABLES
                    ;
0043        SELECT   EQU $43        select pointer
0045        GETPTR   EQU $45        zero page pointer
0050        NUMBER   EQU $50        temp number store
0032        BUFFEN   EQU $32        buffer fill pointer
0034        BPOINT   EQU $34        buffer empty pointer
0052        COUNT    EQU $52        counts lines dumped
0054        NUMLIN   EQU $54        number lines to dump
0056        EA       EQU $56        end address
0058        SA       EQU $58        start address
005A        COLUMN   EQU $5A        data cell for dump
0047        PRINTR   EQU $47        printer flag
0048        HOST     EQU $48        output device flag
003B        IEEFLG   EQU $3B        flag for GPIB as host
003C        IEEEPR   EQU $3C        GPIB as printer flag
0036        CHAR     EQU $36        char to be output
005C        REPEAT   EQU $5C        loop counter
005D        TEMPX    EQU $5D        temporary X register
0049        RETURN   EQU $49        temp return address
005E        DISLNS   EQU $5E        no. lines disass'ed
005F        LINUM    EQU $5F        no. to disassemble
0060        LETTER   EQU $60        letter from mnem
0061        SUBPTR   EQU $61        pointer to sub'tine
0063        OPBYTS   EQU $63        data cell
0064        OPCHRS   EQU $64        data cell
0065        ADRCOL   EQU $65        start column of field
                    ;
                    ; CONSTANTS
                    ;
00FF        ON       EQU $FF
0000        OFF      EQU $0
000D        CR       EQU $0D
000A        LF       EQU $0A
007F        TEX      EQU $7F
00FF        ETX      EQU $FF
                    ;
                    ; BUFFERS
                    ;
0300        BUFF     EQU $0300      input buffer
0350        OUTBUF   EQU $0350      output buffer
                    ;
                    ; LOCATION OF JUMP INSTRUCTIONS FOR
                    ; ROUTINES IN $F000 EPROM
                    ;
F000        RESET    EQU $F000
F003        PRTCHR   EQU $F003
F006        PRINT    EQU $F006
F009        HOSTON   EQU $F009
F00C        HOSTOF   EQU $F00C
```

```
     F00F       PRTON     EQU $F00F
     F012       PRTOFF    EQU $F012
     F015       RTSON     EQU $F015
     F018       RTSOFF    EQU $F018
     F01B       RSRES     EQU $F01B
     F01E       PUSHSL    EQU $F01E
     F021       POPSL     EQU $F021
     F024       GETCHR    EQU $F024
     F027       RDOB      EQU $F027
     F02A       GETSL     EQU $F02A
     F02D       MINCON    EQU $F02D
                ;
E000                      ORG X'E000'
                ;
E000 4C 79E3              JMP START
                ;
E003 A5 44      PRADR     LDA SELECT+1   print high byte of
E005 20 D7E0              JSR PRTBYT     pointer
E008 A5 43                LDA SELECT     and low byte
E00A 20 D7E0              JSR PRTBYT
E00D 60                   RTS
E00E 20 2AF0    DUMPSL    JSR GETSL      get byte from pointer
E011 20 D7E0              JSR PRTBYT     and print it
E014 60                   RTS
E015 20 B4E0    PRDUMP    JSR GOTOSA     pointer=start address
E018 20 60E0    HXLOOP    JSR PRLINE     dump one line
E01B 10 FB                BPL HXLOOP     loop if A<&FF
E01D 20 C6E0              JSR CRLF       print cr,lf
E020 60                   RTS
E021 38         SETADS    SEC
E022 A5 57                LDA EA+1       compare high byte of
E024 C5 59                CMP SA+1       start & end addresses
E026 90 0A                BCC TOOLOW     if end  start
E028 D0 06                BNE SEXIT
E02A A5 56                LDA EA         compare low byte of
E02C C5 58                CMP SA         start & end addresses
E02E 90 02                BCC TOOLOW     if end  start
E030 18         SEXIT     CLC            return carry clear
E031 60                   RTS
E032 38         TOOLOW    SEC            return with carry set
E033 60                   RTS
E034 A9 24      PRSA      LDA #C'$'      print $
E036 20 03F0              JSR PRTCHR
E039 A5 59                LDA SA+1       print high byte of
E03B 20 D7E0              JSR PRTBYT     start address
E03E A5 58                LDA SA         and low byte
E040 20 D7E0              JSR PRTBYT
E043 60                   RTS
E044 A9 24      PREA      LDA #C'$'      print $
E046 20 03F0              JSR PRTCHR
E049 A5 57                LDA EA+1       print high byte of
E04B 20 D7E0              JSR PRTBYT     end address
E04E A5 56                LDA EA         and low byte
E050 20 D7E0              JSR PRTBYT
E053 60                   RTS
E054 20 34E0    RANGE     JSR PRSA       print start address
E057 A9 2D                LDA #C'-'      hyphen
E059 20 03F0              JSR PRTCHR
```

```
E05C 20 44E0            JSR PREA        and end address
E05F 60                 RTS
E060 20 C6E0   PRLINE   JSR CRLF        print cr,lf
E063 A5 43              LDA SELECT      low byte of pointer
E065 48                 PHA             save on stack
E066 29 0F              AND #$0F        mask high byte
E068 85 5A              STA COLUMN      save in col counter
E06A 68                 PLA             fetch back from stack
E06B 29 F0              AND #$F0        make pointer multiple
E06D 85 43              STA SELECT      of 16
E06F 20 03E0            JSR PRADR       print pointer
E072 A2 03              LDX #$3         print three spaces
E074 20 EAE0            JSR SPACES
E077 A5 5A              LDA COLUMN      get coulmn counter
E079 F0 0C              BEQ COLOK       if zero branch
E07B A2 03     PRLOOP   LDX #$3         print three spaces
E07D 20 EAE0            JSR SPACES
E080 20 3DE1            JSR INCSL       increment pointer
E083 C6 5A              DEC COLUMN      and decrease column
E085 D0 F4              BNE PRLOOP      if not zero loop
E087 20 0EE0   COLOK    JSR DUMPSL      byte from pointer
E08A 20 D1E0            JSR SPACE       print space
E08D 20 9BE0            JSR NEXTSL      increment pointer
E090 30 08              BMI EXIT        loop if A<&FF
E092 A5 43     NOTEA    LDA SELECT      low byte of pointer
E094 29 0F              AND #$0F        mask high bits
E096 C9 00              CMP #0          not dumped 16 bytes
E098 D0 ED              BNE COLOK       then loop
E09A 60        EXIT     RTS
E09B 38        NEXTSL   SEC             prepare to subtract
E09C A5 44              LDA SELECT+1    high byte of pointer
E09E C5 57              CMP EA+1        less than high byte
E0A0 90 09              BCC SLOK        end address continue
E0A2 D0 0D              BNE NOINC       if greater then end
E0A4 38                 SEC
E0A5 A5 43              LDA SELECT      low byte of pointer
E0A7 C5 56              CMP EA          > end address ?
E0A9 B0 06              BCS NOINC
E0AB 20 3DE1   SLOK     JSR INCSL       increment pointer
E0AE A9 00              LDA #0          return with A=0
E0B0 60                 RTS
E0B1 A9 FF     NOINC    LDA #$FF        return with A=&FF
E0B3 60                 RTS
E0B4 A5 58     GOTOSA   LDA SA          pointer equals
E0B6 85 43              STA SELECT      start address
E0B8 A5 59              LDA SA+1
E0BA 85 44              STA SELECT+1
E0BC 60                 RTS
E0BD A5 43     DECSL    LDA SELECT      decrease pointer
E0BF D0 02              BNE BRAN2       low byte
E0C1 C6 44              DEC SELECT+1    and high byte if
E0C3 C6 43     BRAN2    DEC SELECT      necessary
E0C5 60                 RTS
E0C6 A9 0D     CRLF     LDA #CR         print CR
E0C8 20 03F0            JSR PRTCHR
E0CB A9 0A              LDA #LF         and line feed
E0CD 20 03F0            JSR PRTCHR
E0D0 60                 RTS
```

```
EOD1 A9 20      SPACE    LDA #$20      print a space
EOD3 20 O3FO             JSR PRTCHR
EOD6 60                  RTS
EOD7 48         PRTBYT   PHA           save acc
EOD8 4A                  LSR A         move top 4 bits to
EOD9 4A                  LSR A         lower 4 bits
EODA 4A                  LSR A
EODB 4A                  LSR A
EODC 20 2FE1             JSR ASCII     convert to ascii
EODF 20 O3FO             JSR PRTCHR    and print
EOE2 68                  PLA           get back acc
EOE3 20 2FE1             JSR ASCII     and print lower 4
EOE6 20 O3FO             JSR PRTCHR    bits in ascii
EOE9 60                  RTS
EOEA A9 20      SPACES   LDA #$20      print X number of
EOEC 86 5C      CHARS    STX REPEAT    spaces
EOEE 48         RPLOOP   PHA
EOEF A6 5C               LDX REPEAT
EOF1 FO O9               BEQ RPTEND
EOF3 C6 5C               DEC REPEAT
EOF5 20 O3FO             JSR PRTCHR
EOF8 68                  PLA
EOF9 18                  CLC
EOFA 90 F2               BCC RPLOOP
EOFC 68         RPTEND   PLA
EOFD 60                  RTS
EOFE 86 5C      CRLFS    STX REPEAT    print X number of
E100 A6 5C      CRLOOP   LDX REPEAT    CR's and LF's
E102 FO O7               BEQ ENDCR
E104 C6 5C               DEC REPEAT
E106 20 C6EO             JSR CRLF
E109 90 F5               BCC CRLOOP
E10B 60         ENDCR    RTS
E10C 86 5D      PRMSG    STX TEMPX     save X register
E10E B5 O1               LDA 1,X       save message pointer
E110 48                  PHA
E111 B5 OO               LDA O,X
E113 48                  PHA
E114 A6 5D      LOOP     LDX TEMPX     X points to message
E116 A1 OO               LDA (O,X)     pointer on zero page
E118 C9 FF               CMP #ETX      get character, ETX ?
E11A FO OC               BEQ MSGEND    yes - end routine
E11C F6 OO               INC O,X       no increment pointer
E11E DO O2               BNE NEXT
E120 F6 O1               INC 1,X
E122 20 O3FO    NEXT     JSR PRTCHR    print character
E125 18                  CLC
E126 90 EC               BCC LOOP      and loop
E128 68         MSGEND   PLA           restore original
E129 95 OO               STA O,X       message pointer
E12B 68                  PLA
E12C 95 O1               STA 1,X
E12E 60                  RTS
E12F O8         ASCII    PHP           save decimal flag
E130 D8                  CLD           clear decimal flag
E131 29 OF               AND #$OF      clear 4 MSB's
E133 C9 OA               CMP #$OA      is acc›9
E135 30 O2               BMI DECIML    if not must be O-9
```

```
E137 69 06              ADC #6          if so add &36, A-F
E139 69 30    DECIML    ADC #$30        add &30 => ascii
E13B 28                 PLP             restore decimal flag
E13C 60                 RTS
E13D E6 43    INCSL     INC SELECT      increment select
E13F D0 02              BNE BRANCH      pointer
E141 E6 44              INC SELECT+1
E143 60       BRANCH    RTS
E144 A6 32    ADDRES    LDX BUFFEN      X with buffer pointer
E146 A0 02              LDY #2          Y as loop counter
E148 88       ADLOOP    DEY
E149 CA                 DEX             X points to next char
E14A BD 0003            LDA BUFF,X      to the left in buffer
E14D 20 95E1            JSR BINARY      convert to binary
E150 C9 FF              CMP #$FF        if acc=&FF error
E152 F0 20              BEQ ADERR
E154 99 5000            STA NUMBER,Y    save as low 4 bits
E157 CA                 DEX             next character to
E158 BD 0003            LDA BUFF,X      the left
E15B 20 95E1            JSR BINARY      convert to binary
E15E C9 FF              CMP #$FF
E160 F0 12              BEQ ADERR       if error
E162 0A                 ASL A           move 4 LSB's to
E163 0A                 ASL A           4 MSB's
E164 0A                 ASL A
E165 0A                 ASL A
E166 19 5000            ORA NUMBER,Y    add to 4 LSB's
E169 99 5000            STA NUMBER,Y    and save it
E16C C0 00              CPY #0          is loop ended
E16E D0 D8              BNE ADLOOP      no - loop
E170 86 32              STX BUFFEN      update buffer pointer
E172 18                 CLC             clear carry and
E173 60                 RTS             return
E174 38       ADERR     SEC             set carry and
E175 60                 RTS             return
E176 20 44E1  STEASA    JSR ADDRES      4 characters from
E179 B0 18              BCS SERR        buffer to 16 bit no.
E17B A5 51              LDA NUMBER+1    and make it the end
E17D 85 56              STA EA          address
E17F A5 50              LDA NUMBER
E181 85 57              STA EA+1
E183 C6 32              DEC BUFFEN      jump past space
E185 20 44E1            JSR ADDRES      and repeat for start
E188 B0 09              BCS SERR        address
E18A A5 51              LDA NUMBER+1
E18C 85 58              STA SA
E18E A5 50              LDA NUMBER
E190 85 59              STA SA+1
E192 60                 RTS
E193 38       SERR      SEC             if error set carry
E194 60                 RTS
E195 38       BINARY    SEC             prepare to subtract
E196 E9 30              SBC #$30        subtract &30
E198 90 0A              BCC BAD         if char < &30 bad
E19A C9 0A              CMP #$0A        was it &30 - &39
E19C 90 08              BCC GOOD        yes - now have binary
E19E E9 07              SBC #7          subtract 7
E1A0 C9 10              CMP #$10        was it &41-&46
```

```
E1A2 90 02              BCC GOOD        yes - now have binary
E1A4 A9 FF     BAD      LDA #$FF        acc with &FF
E1A6 60        GOOD     RTS
E1A7 20 B4E0   PRDIS    JSR GOTOSA      select pointer=›start
E1AA 20 C6E0            JSR CRLF        address, print cr,lf
E1AD 20 B3E1   PRLP     JSR DSLINE      disassemble one line
E1B0 10 FB              BPL PRLP        loop until last line
E1B2 60                 RTS             done then return
E1B3 20 2AF0   DSLINE   JSR GETSL       get byte from select
E1B6 48                 PHA             save it
E1B7 20 C8E1            JSR MNEMON      mnemonic for opcode
E1BA 20 D1E0            JSR SPACE       print space
E1BD 68                 PLA             restore opcode
E1BE 20 E1E1            JSR OPERAN      print operand
E1C1 20 31E2            JSR FINISH      finish line
E1C4 20 9BE0            JSR NEXTSL      increment select
E1C7 60                 RTS
E1C8 A2 03     MNEMON   LDX #3          3 letters to print
E1CA 86 60              STX LETTER
E1CC AA                 TAX             use opcode as index
E1CD BD FAE5            LDA MCODES,X    get mnemonic code for
E1D0 AA                 TAX             that opcode, as index
E1D1 BD FAE7   MNLOOP   LDA MNAMES,X    get letter from table
E1D4 86 5D              STX TEMPX       save X register
E1D6 20 03F0            JSR PRTCHR      print it
E1D9 A6 5D              LDX TEMPX       increment table point
E1DB E8                 INX
E1DC C6 60              DEC LETTER      decrease counter
E1DE D0 F1              BNE MNLOOP      if not yet zero loop
E1E0 60                 RTS
E1E1 AA        OPERAN   TAX             lookup addressing mode
E1E2 BD FAE6            LDA MODES,X     for the opcode
E1E5 AA                 TAX             X indicates addressing
E1E6 20 EAE1            JSR MODEX       mode, handle it
E1E9 60                 RTS
E1EA BD 3AE3   MODEX    LDA SUBS,X      low byte of X pointer
E1ED 85 61              STA SUBPTR      table of subroutines
E1EF E8                 INX
E1F0 BD 3AE3            LDA SUBS,X      high byte
E1F3 85 62              STA SUBPTR+1
E1F5 6C 6100            JMP (SUBPTR)    jump to chosen routine
E1F8 20 3DE1   ONEBYT   JSR INCSL       advance to byte after
E1FB 20 0EE0            JSR DUMPSL      opcode and dump it
E1FE 60                 RTS
E1FF 20 3DE1   TWOBYT   JSR INCSL       get byte after opcode
E202 20 2AF0            JSR GETSL
E205 48                 PHA             save it on stack
E206 20 3DE1            JSR INCSL       get high byte
E209 20 0EE0            JSR DUMPSL      dump it
E20C 68                 PLA             recover low byte
E20D 20 D7E0            JSR PRTBYT      dump it
E210 60                 RTS
E211 A9 28     LPAREN   LDA #C'('       print left parenthesis
E213 D0 02              BNE SENDIT
E215 A9 29     RPAREN   LDA #C')'       print right
E217 20 03F0   SENDIT   JSR PRTCHR      parenthesis
E21A 60                 RTS
E21B A9 2C     XINDEX   LDA #C','       print comma
```

```
E21D 20 03F0         JSR PRTCHR
E220 A9 58           LDA #C'X'      followed by an X
E222 20 03F0         JSR PRTCHR
E225 60              RTS
E226 A9 2C   YINDEX  LDA #C','      print a comma
E228 20 03F0         JSR PRTCHR
E22B A9 59           LDA #C'Y'      followed by a Y
E22D 20 03F0         JSR PRTCHR
E230 60              RTS
E231 85 64   FINISH  STA OPCHRS     save length of the
E233 86 63           STX OPBYTS     operand in chars and
E235 CA              DEX            in bytes,decrease
E236 30 06           BMI SELOK      pointer to point to
E238 20 BDE0 LOOP1   JSR DECSL      opcode
E23B CA              DEX
E23C 10 FA           BPL LOOP1      select =› opcode
E23E 08      SELOK   PHP            save decimal flag
E23F D8              CLD            clear decimal flag
E240 38              SEC            space over to column
E241 A5 65           LDA ADRCOL     for the address
E243 E9 04           SBC #4         operand field started
E245 E5 64           SBC OPCHRS     4 and includes opchrs
E247 28              PLP            restore decimal flag
E248 AA              TAX            print enough spaces
E249 20 EAE0         JSR SPACES     to reach address col
E24C 20 03E0         JSR PRADR      print address
E24F 20 D1E0 LOOP2   JSR SPACE      space once
E252 20 0EE0         JSR DUMPSL     dump selected byte
E255 20 3DE1         JSR INCSL      increment pointer
E258 C6 63           DEC OPBYTS     dumped last byte?
E25A 10 F3           BPL LOOP2      no - loop
E25C 20 BDE0         JSR DECSL      select points to last
             ;                      byte in operand
E25F 20 C6E0 FINEND  JSR CRLF       goto a new line
E262 60              RTS
E263 20 FFE1 ABSLUT  JSR TWOBYT     print two byte operan
E266 A2 02           LDX #2         operand has two bytes
E268 A9 04           LDA #4         and four characters
E26A 60              RTS
E26B 20 63E2 ABSX    JSR ABSLUT
E26E 20 1BE2         JSR XINDEX     print comma and an X
E271 A2 02           LDX #2         two bytes
E273 A9 06           LDA #6         six characters
E275 60              RTS
E276 20 63E2 ABSY    JSR ABSLUT
E279 20 26E2         JSR YINDEX     print comma and a Y
E27C A2 02           LDX #2         two bytes
E27E A9 06           LDA #6         six characters
E280 60              RTS
E281 A9 41   ACC     LDA #C'A'      print letter A
E283 20 03F0         JSR PRTCHR
E286 A2 00           LDX #0         no bytes
E288 A9 01           LDA #1         one character
E28A 60              RTS
E28B A2 00   IMPLID  LDX #0         no bytes
E28D A9 00           LDA #0         and no characters
E28F 60              RTS
E290 A9 23   IMMEDT  LDA #C'#'      print a #
```

```
E292 20 03F0           JSR PRTCHR
E295 A9 24             LDA #C'$'     then a $ to show hex
E297 20 03F0           JSR PRTCHR
E29A 20 F8E1           JSR ONEBYT    print one byte
E29D A2 01             LDX #1        one byte
E29F A9 04             LDA #4        and four characters
E2A1 60                RTS
E2A2 20 11E2  INDRCT   JSR LPAREN    left parenthesis
E2A5 20 63E2           JSR ABSLUT    two byte operand
E2A8 20 15E2           JSR RPAREN    right parenthesis
E2AB A9 06             LDA #6        six characters
E2AD A2 02             LDX #2        and two bytes
E2AF 60                RTS
E2B0 20 11E2  INDX     JSR LPAREN
E2B3 20 07E3           JSR ZEROX     print zero page
              ;                      address, comma and X
E2B6 20 15E2           JSR RPAREN    right parenthesis
E2B9 A2 01             LDX #1        one byte
E2BB A9 08             LDA #8        eight characters
E2BD 60                RTS
E2BE 20 11E2  INDY     JSR LPAREN
E2C1 20 FAE2           JSR ZEROPG
E2C4 20 15E2           JSR RPAREN
E2C7 20 26E2           JSR YINDEX
E2CA A2 01             LDX #1
E2CC A9 08             LDA #8
E2CE 60                RTS
E2CF 20 3DE1  RELATV   JSR INCSL     select next byte
E2D2 20 1EF0           JSR PUSHSL    save select pointer
E2D5 20 2AF0           JSR GETSL     get operand byte
E2D8 48                PHA           save it on stack
E2D9 20 3DE1           JSR INCSL     increment pointer
E2DC 68                PLA           restore operand byte
E2DD C9 00             CMP #0        is it plus or minus
E2DF 10 02             BPL FORWRD    plus forward branch
E2E1 C6 44             DEC SELECT+1  branching backwards
              ;                      is like branching
              ;                      forwards from one
              ;                      page lower
E2E3 08       FORWRD   PHP           save callers flags
E2E4 D8                CLD           binary arithmetic
E2E5 18                CLC           prepare to add
E2E6 65 43             ADC SELECT    add operand
E2E8 90 02             BCC RELEND
E2EA E6 44             INC SELECT+1
E2EC 85 43    RELEND   STA SELECT    select => branch
E2EE 28                PLP           restore decimal flag
E2EF 20 03E0           JSR PRADR     print address
E2F2 20 21F0           JSR POPSL     restore select
E2F5 A2 01             LDX #1        one byte
E2F7 A9 04             LDA #4        and four characters
E2F9 60                RTS
E2FA A9 00    ZEROPG   LDA #0        print two ascii '0's
E2FC 20 D7E0           JSR PRTBYT
E2FF 20 F8E1           JSR ONEBYT    print 1 byte operand
E302 A2 01             LDX #1        one byte
E304 A9 04             LDA #4        and four characters
E306 60                RTS
```

```
E307 20 FAE2   ZEROX    JSR ZEROPG      zero page address
E30A 20 1BE2            JSR XINDEX      comma and an X
E30D A2 01              LDX #1          one byte
E30F A9 06              LDA #6          and six characters
E311 60                 RTS
E312 20 FAE2   ZEROY    JSR ZEROPG
E315 20 26E2            JSR YINDEX
E318 A9 01              LDA #1
E31A A9 06              LDA #6
E31C 60                 RTS
E31D 68        TXMODE   PLA             pop return address to
E31E 68                 PLA             operan
E31F 68                 PLA             and return address to
E320 68                 PLA             dsline
E321 20 9BE0            JSR NEXTSL      advance past TEX
E324 30 0D              BMI TXEXIT      return if reached end
E326 20 2AF0            JSR GETSL       get character
E329 C9 FF              CMP #ETX        is it ETX
E32B F0 06              BEQ TXEXIT      yes - exit
E32D 20 03F0            JSR PRTCHR      print character
E330 18                 CLC             loop next character
E331 90 EE              BCC TXMODE+4
E333 20 C6E0   TXEXIT   JSR CRLF        print cr,lf
E336 20 9BE0            JSR NEXTSL      advance next opcode
E339 60                 RTS
E33A 8BE2      SUBS     ADDR IMPLID     mode zero is invalid
E33C 81E2               ADDR ACC
E33E 90E2               ADDR IMMEDT
E340 FAE2               ADDR ZEROPG
E342 07E3               ADDR ZEROX
E344 12E3               ADDR ZEROY
E346 63E2               ADDR ABSLUT
E348 6BE2               ADDR ABSX
E34A 76E2               ADDR ABSY
E34C 8BE2               ADDR IMPLID
E34E CFE2               ADDR RELATV
E350 B0E2               ADDR INDX
E352 BEE2               ADDR INDY
E354 A2E2               ADDR INDRCT
E356 1DE3               ADDR TXMODE
               ;
               ; USER INTERFACE SECTION
               ;
E358 A9 00     INIT     LDA #0          initialise variables
E35A 85 52              STA COUNT
E35C 85 5A              STA COLUMN
E35E 85 36              STA CHAR
E360 85 5C              STA REPEAT
E362 85 5D              STA TEMPX
E364 85 49              STA RETURN
E366 85 4A              STA RETURN+1
E368 85 5F              STA LINUM
E36A 85 60              STA LETTER
E36C A9 04              LDA #4
E36E 85 54              STA NUMLIN
E370 A9 05              LDA #5
E372 85 5E              STA DISLNS
E374 A9 10              LDA #16
```

```
E376 85 65                STA ADRCOL
E378 60                   RTS
E379 20 58E3    START     JSR INIT        initialise variables
E37C 20 1BF0              JSR RSRES       reset 6850
E37F 20 18F0              JSR RTSOFF      RTS low
E382 A5 3B                LDA IEEFLG      RS232 or IEEE488 ?
E384 F0 0D                BEQ SERIAL
E386 A9 FF                LDA #ON
E388 85 3C                STA IEEEPR      IEEE as output
E38A 20 0CF0              JSR HOSTOF
E38D 20 12F0              JSR PRTOFF
E390 4C 9DE3              JMP START2
E393 20 09F0    SERIAL    JSR HOSTON      use host for output
E396 20 12F0              JSR PRTOFF
E399 A9 00                LDA #OFF
E39B 85 3C                STA IEEEPR
E39D 20 C6E0    START2    JSR CRLF        print a couple of
E3A0 20 C6E0              JSR CRLF        cr's and lf's
E3A3 20 06F0              JSR PRINT       print title
E3A6 7F                   DB TEX
E3A7 23232320             DB '### I.C.C. RS232/IEEE MONITOR
E3AB 492E432E                VS 1.0 ###'
E3AF 432E2052
E3B3 53323332
E3B7 2F494545
E3BB 45204D4F
E3BF 4E49544F
E3C3 52205653
E3C7 20312E30
E3CB 20232323
E3CF 0D0A                 DB CR,LF
E3D1 28632920             DB '(c) M.J.HILL 1984/5'
E3D5 4D2E4A2E
E3D9 48494C4C
E3DD 20313938
E3E1 342F35
E3E4 0D0AFF               DB CR,LF,ETX
E3E7 20 C6E0              JSR CRLF
E3EA 20 06F0    VISMON    JSR PRINT       print prompt
E3ED 7F0D0A2A             DB TEX,CR,LF,'*',ETX
E3F1 FF
E3F2 A5 3B                LDA IEEFLG
E3F4 F0 1A                BEQ SER2        IEEE or RS232 input
E3F6 A2 00                LDX #0
E3F8 20 27F0    NXTIN     JSR RDOB        get byte from GPIB
E3FB 9D 0003              STA BUFF,X
E3FE C9 08                CMP #8
E400 F0 2B                BEQ DELETE
E402 C9 7F                CMP #$7F
E404 F0 27                BEQ DELETE
E406 20 40E4              JSR ECHO
E409 C9 0D                CMP #13
E40B F0 39                BEQ PROCES
E40D E8                   INX
E40E D0 E8                BNE NXTIN
E410 20 15F0    SER2      JSR RTSON       RTS high
E413 A2 00                LDX #0
E415 20 24F0    NEXTIN    JSR GETCHR      characters from 6850
```

```
E418 9D 0003              STA BUFF,X      and place in buffer
E41B C9 08                CMP #8          handle delete and
E41D F0 0E                BEQ DELETE      backspace alike
E41F C9 7F                CMP #$7F
E421 F0 0A                BEQ DELETE
E423 20 40E4              JSR ECHO        echo to terminal
E426 C9 0D                CMP #13         stop when CR received
E428 F0 1C                BEQ PROCES
E42A E8                   INX
E42B D0 E8                BNE NEXTIN
E42D E0 00      DELETE    CPX #0          buffer pointer back
E42F D0 08                BNE NONZ        one but not past zero
E431 A9 07                LDA #7
E433 20 40E4              JSR ECHO
E436 18                   CLC
E437 90 DC                BCC NEXTIN
E439 CA         NONZ      DEX
E43A 20 40E4              JSR ECHO
E43D 18                   CLC
E43E 90 D5                BCC NEXTIN
E440 48         ECHO      PHA
E441 20 03F0              JSR PRTCHR
E444 68                   PLA
E445 60                   RTS
E446 86 32      PROCES    STX BUFFEN      save buffer pointer
E448 A2 00                LDX #0
E44A BD 0003              LDA BUFF,X      get first character
E44D C9 0D                CMP #13         if CR back to monitor
E44F F0 99                BEQ VISMON
E451 C9 44                CMP #C'D'       if D then disassemble
E453 D0 0B                BNE VB1         if not test another
E455 20 76E1              JSR STEASA      decode addresses
E458 B0 4F                BCS ERROR       carry set if error
E45A 20 A7E1              JSR PRDIS       jump to disas
E45D 4C EAE3              JMP VISMON      finished
E460 C9 45      VB1       CMP #C'E'       if E then exit to
E462 D0 03                BNE VB2         minicon routines
E464 4C 2DF0              JMP MINCON
E467 C9 4D      VB2       CMP #C'M'       if M then dump
E469 D0 0B                BNE VB3
E46B 20 76E1              JSR STEASA
E46E B0 39                BCS ERROR
E470 20 15E0              JSR PRDUMP
E473 4C EAE3              JMP VISMON
E476 C9 50      VB3       CMP #C'P'       if P then toggle
E478 D0 10                BNE VB4         printer on or off
E47A A5 47                LDA PRINTR
E47C F0 06                BEQ PON         off so put on
E47E 20 12F0              JSR PRTOFF      on so put off
E481 4C EAE3              JMP VISMON
E484 20 0FF0    PON       JSR PRTON
E487 4C EAE3              JMP VISMON
E48A C9 48      VB4       CMP #C'H'
E48C D0 06                BNE VB5
E48E 20 C4E4              JSR MENU
E491 4C EAE3              JMP VISMON
E494 20 06F0    VB5       JSR PRINT       unrecognised command
E497 7F0D0A               DB TEX,CR,LF
```

```
E49A 3F20434F           DB '? COMMAND'
E49E 4D4D414E
E4A2 44
E4A3 0D0AFF             DB CR,LF,ETX
E4A6 4C EAE3            JMP VISMON     back to monitor
E4A9 18        ERROR    CLC            command error
E4AA 20 06F0            JSR PRINT
E4AD 7F0D0A             DB TEX,CR,LF
E4B0 434F4D4D           DB 'COMMAND ERROR'
E4B4 414E4420
E4B8 4552524F
E4BC 52
E4BD FF                 DB ETX
E4BE 20 C6E0            JSR CRLF
E4C1 4C EAE3            JMP VISMON
E4C4 20 06F0   MENU     JSR PRINT
E4C7 7F0D0A             DB TEX,CR,LF
E4CA 4D4F4E49           DB 'MONITOR COMMANDS :',CR,LF,LF
E4CE 544F5220
E4D2 434F4D4D
E4D6 414E4453
E4DA 203A0D0A
E4DE 0A
E4DF 4D203C73           DB 'M ‹start address› ‹end address›
E4E3 74617274               Dumps memory between'
E4E7 20616464
E4EB 72657373
E4EF 3E203C65
E4F3 6E642061
E4F7 64647265
E4FB 73733E20
E4FF 20204475
E503 6D707320
E507 6D656D6F
E50B 72792062
E50F 65747765
E513 656E
E515 20616464           DB ' addresses given (in hex)',CR,LF
E519 72657373
E51D 65732067
E521 6976656E
E525 2028696E
E529 20686578
E52D 290D0A
E530 44203C73           DB 'D ‹start address› ‹end address›
E534 74617274              Disassembles memory '
E538 20616464
E53C 72657373
E540 3E203C65
E544 6E642061
E548 64647265
E54C 73733E20
E550 20204469
E554 73617373
E558 656D626C
E55C 6573206D
E560 656D6F72
E564 7920
```

```
      E566 62657477
      E56A 65656E20             DB 'between given addresses',CR,LF
      E56E 67697665
      E572 6E206164
      E576 64726573
      E57A 7365730D
      E57E 0A
      E57F 50202020
      E583 20202020             DB 'P
      E587 20202020                Toggles printer port'
      E58B 20202020
      E58F 20202020
      E593 20202020
      E597 20202020
      E59B 20202020
      E59F 2020546F
      E5A3 67676C65
      E5A7 73207072
      E5AB 696E7465
      E5AF 7220706F
      E5B3 7274
      E5B5 206F6E20
      E5B9 6F72206F             DB ' on or off',CR,LF
      E5BD 66660D0A
      E5C1 45202020
      E5C5 20202020             DB 'E
      E5C9 20202020                Exits to minicon'
      E5CD 20202020
      E5D1 20202020
      E5D5 20202020
      E5D9 20202020
      E5DD 20202020
      E5E1 20204578
      E5E5 69747320
      E5E9 746F206D
      E5ED 696E6963
      E5F1 6F6E
      E5F3 0D0A0AFF
      E5F7 4C EAE3              DB CR,LF,LF,ETX
                                JMP VISMON          back to monitor
                       ;
                       ; TABLES FOR DISASSEMBLER
                       ;
      E5FA 226A0101    MCODES   DB $22,$6A,$01,$01,$01,$6A,$0A,
      E5FE 016A0A01                $01,$70,$6A
      E602 706A
      E604 0A01016A             DB $0A,$01,$01,$6A,$0A,$01,$1F,
      E608 0A011F6A                $6A,$01,$01
      E60C 0101
      E60E 016A0A01             DB $01,$6A,$0A,$01,$2B,$6A,$01,
      E612 2B6A0101                $01,$01,$64,$0A
      E616 01640A
      E619 01580701             DB $01,$58,$07,$01,$01,$16,$07,
      E61D 01160779                $79,$01,$76
      E621 0176
      E623 07790116             DB $07,$79,$01,$16,$07,$79,$01,
      E627 07790119                $19,$07,$01
      E62B 0701
      E62D 01010779             DB $01,$01,$07,$79,$01,$88,$07,
E631 01880701                 $01,$01
E635 01
E636 01077901         DB $01,$07,$79,$01,$7F,$49,$01,
E63A 7F490101            $01,$01,$49
E63E 0149
E640 64016D49         DB $64,$01,$6D,$49,$64,$01,$55,
E644 64015549            $49,$64,$01
E648 6401
E64A 25490101         DB $25,$49,$01,$01,$01,$49,$64,
E64E 01496401            $01,$31,$49
E652 3149
E654 01010149         DB $01,$01,$01,$49,$64,$01,$82,
E658 64018204            $04,$01,$01
E65C 0101
E65E 01047C01         DB $01,$04,$7C,$01,$73,$04,$7C,
E662 73047C01            $01,$55,$04
E666 5504
E668 7C012804         DB $7C,$01,$28,$04,$01,$01,$01,
E66C 01010104            $04,$7C,$01
E670 7C01
E672 8E040101         DB $8E,$04,$01,$01,$01,$04,$7C,
E676 01047CAC            $AC,$01,$91
E67A 0191
E67C 01019791         DB $01,$01,$97,$91,$94,$01,$46,
E680 94014601            $01,$A3,$01
E684 A301
E686 97919401         DB $97,$91,$94,$01,$0D,$91,$01,
E68A 0D910101            $01,$97,$91
E68E 9791
E690 9401A991         DB $94,$01,$A9,$91,$A6,$01,$01,
E694 A6010191            $91,$01,$01
E698 0101
E69A 615B5E01         DB $61,$5B,$5E,$01,$61,$5B,$5E,
E69E 615B5E01            $01,$9D,$5B
E6A2 9D5B
E6A4 9A01615B         DB $9A,$01,$61,$5B,$5E,$01,$10,
E6A8 5E01105B            $5B,$01,$01
E6AC 0101
E6AE 615B5E01         DB $61,$5B,$5E,$01,$34,$5B,$9E,
E6B2 345B9E01            $01,$61,$5B
E6B6 615B
E6B8 5E013D37         DB $5E,$01,$3D,$37,$01,$01,$3D,
E6BC 01013D37            $37,$40,$01
E6C0 4001
E6C2 52374301         DB $52,$37,$43,$01,$3D,$37,$40,
E6C6 3D374001            $01,$1C,$37
E6CA 1C37
E6CC 01010137         DB $01,$01,$01,$37,$40,$01,$2E,
E6D0 40012E37            $37,$01,$01
E6D4 0101
E6D6 01374001         DB $01,$37,$40,$01,$3A,$85,$01,
E6DA 3A850101            $01,$3A,$85
E6DE 3A85
E6E0 4C014F85         DB $4C,$01,$4F,$85,$67,$01,$3A,
E6E4 67013A85            $85,$4C,$01
E6E8 4C01
E6EA 13850101         DB $13,$85,$01,$01,$01,$85,$4C,
E6EE 01854C01            $01,$8B,$85
```

```
E6F2 8B85
E6F4 01010185                DB $01,$01,$01,$85,$4C,$01
E6F8 4C01
E6FA 12160000  MODES         DB 18,22,00,00,00,06,06,00,18,04,02
E6FE 00060600
E702 120402
E705 00000C0C                DB 00,00,12,12,00,20,24,00,00,00
E709 00141800
E70D 0000
E70F 0E0E0012                DB 14,14,00,18,16,00,00,00,22,22
E713 10000000
E717 1616
E719 000C1600                DB 00,12,22,00,00,06,06,06,00,18
E71D 00060606
E721 0012
E723 0402000C                DB 04,02,00,12,12,12,00,20,24,00
E727 0C0C0014
E72B 1800
E72D 00000808                DB 00,00,08,08,00,18,16,00,00,00
E731 00121000
E735 0000
E737 0E0E0012                DB 14,14,00,18,22,00,00,00,06,06
E73B 16000000
E73F 0606
E741 00120402                DB 00,18,04,02,00,12,12,12,00,20
E745 000C0C0C
E749 0014
E74B 18000000                DB 24,00,00,00,08,08,00,18,16,00
E74F 08080012
E753 1000
E755 00000E0E                DB 00,00,14,14,00,18,22,00,00,00
E759 00121600
E75D 0000
E75F 06060012                DB 06,06,00,18,04,02,00,26,12,12
E763 0402001A
E767 0C0C
E769 00141800                DB 00,20,24,00,00,00,08,08,00,18
E76D 00000808
E771 0012
E773 10000000                DB 16,00,00,00,14,14,28,00,22,00
E777 0E0E1C00
E77B 1600
E77D 00060606                DB 00,06,06,06,00,18,00,18,00,12,12
E781 00120012
E785 000C0C
E788 0C001418                DB 12,00,20,24,00,00,08,08,10,00
E78C 00000808
E790 0A00
E792 12101200                DB 18,16,18,00,00,14,00,00,04,22
E796 000E0000
E79A 0416
E79C 04000606                DB 04,00,06,06,06,00,18,04,18,00
E7A0 06001204
E7A4 1200
E7A6 0C0C0C00                DB 12,12,12,00,20,24,00,00,08,08
E7AA 14180000
E7AE 0808
E7B0 0A001410                DB 10,00,20,16,18,00,14,14,16,00
```

```
E7B4 12000E0E
E7B8 1000
E7BA 04160000              DB 04,22,00,00,06,06,06,00,18,04
E7BE 06060600
E7C2 1204
E7C4 12000C0C              DB 18,00,12,12,12,00,20,24,00,00
E7C8 0C001418
E7CC 0000
E7CE 00080800              DB 00,08,08,00,18,16,00,00,00,14
E7D2 12100000
E7D6 000E
E7D8 0E000416              DB 14,00,04,22,00,00,06,06,06,00
E7DC 00000606
E7E0 0600
E7E2 12041200              DB 18,04,18,00,12,12,12,00,20,24
E7E6 0C0C0C00
E7EA 1418
E7EC 00000008              DB 00,00,00,08,08,00,12,16,00,00
E7F0 08000C10
E7F4 0000
E7F6 000E0E00              DB 00,14,14,00
E7FA 7F         MNAMES     DB TEX
E7FB 42414441              DB 'BAD','ADC','AND','ASL','BCC',
E7FF 4443414E                 'BCS','BEQ'
E803 4441534C
E807 42434342
E80B 43534245
E80F 51
E810 42495442              DB 'BIT','BMI','BNE','BPL','BRK',
E814 4D49424E                 'BVC','BVS'
E818 4542504C
E81C 42524B42
E820 56434256
E824 53
E825 434C4343              DB 'CLC','CLD','CLI','CLV','CMP',
E829 4C44434C                 'CPX','CPY'
E82D 49434C56
E831 434D5043
E835 50584350
E839 59
E83A 44454344              DB 'DEC','DEX','DEY','EOR','INC',
E83E 45584445                 'INX','INY'
E842 59454F52
E846 494E4349
E84A 4E58494E
E84E 59
E84F 4A4D504A              DB 'JMP','JSR','LDA','LDX','LDY',
E853 53524C44                 'LSR','NOP'
E857 414C4458
E85B 4C44594C
E85F 53524E4F
E863 50
E864 4F524150              DB 'ORA','PHA','PHP','PLA','PLP',
E868 48415048                 'ROL','ROR'
E86C 50504C41
E870 504C5052
E874 4F4C524F
E878 52
```

```
E879 52544952           DB 'RTI','RTS','SBC','SEC','SED',
E87D 54535342              'SEI','STA'
E881 43534543
E885 53454453
E889 45495354
E88D 41
E88E 53545853           DB 'STX','STY','TAX','TAY','TSX',
E892 54595441              'TXA','TXS'
E896 58544159
E89A 54535854
E89E 58415458
E8A2 53
E8A3 54594154           DB 'TYA','TEX'
E8A7 4558
E8A9 FF                 DB ETX
0000                    END
```

APPENDIX C

# DOUBLE CRYSTAL CONTROL PROGRAMS FOR THE BBC MICROCOMPUTER

## INTRODUCTION

The double crystal control programs are all written in standard BBC Basic and employ no machine code, thus making them easily altered. Since the programs are designed to run in the high resolution mode (Mode 0) only a small amount of RAM is available for program use (about 10K). The control system is, therefore, segmented into a number of small programs which are individually run as required by a menu program. This makes it necessary to keep the 'system' disc permanently in one drive, with data being stored on a second drive. A considerable amount of code is, therefore, common to all of the programs, particularly the device handling routines. In order to exchange data between programs use is made of the resident integer variables A% to Z%, which keep their values all the time the machine is powered. Data from the rocking curve measuring routine is stored in a temporary file on the system disc for access from the replotting routines. In order to permanently save this data on the second drive the SAVE routine should be used.

The individual programs required to form the complete system are described below with the common sections of code described afterwards. The programs required are:

| | | |
|---|---|---|
| START | - | initialisation routines |
| MENU | - | main menu routine which executes programs below |
| COUNT | - | counts the photomultiplier pulses |
| SCAN | - | repeatedly steps an axis and counts the PM pulses |
| STEP | - | moves an axis |
| PLOT | - | steps an axis through given range recording PM counts |
| TILT | - | tilts a goniometer by specified angle |
| REPLOT | - | reads disc data and plots it on the screen |
| HPLOT | - | reads disc data and plots it on hardcopy device |
| AUTO | - | automatically optimises the sample tilt |
| TABLE | - | moves the XY scanning table |
| PEAK | - | monitors the counts and drives the axis to remain on peak |
| UTILIT | - | utilities package, caculates Bragg angles etc |
| EPSPLOT | - | reads disc data and plots it on an Epson printer |
| SAVE | - | makes a permanent copy of the temporary data file |
| HELP | - | help package, instructions on the use of each routine |
| CALIB | - | resets the positions of the axes etc as held in memory |
| FIND | - | moves an axis and counts PM pulses until peak is found |

## DESCRIPTION OF PROGRAMS

### Start

This program is run by the autoboot file $.!BOOT. The display is partitioned so that the positions of the axes, the time and date, the users name and the count rate can be continuously displayed. The time and date plus the positions of the two tilt micrometers are entered by the user. The positions of the axes are stored in the variables S%, T%, U% and V% for the two axes and goniometers and K% and L% for the position of the XY table. These values are converted to the appropriate units by the variables step(1) to step(6). The communication protocol to the minicam interface is set to 4800 baud, 8 data bits and no parity. The MENU program is loaded and run when finished.

Procedures used are PROCupdate, PROCtime and PROCtitle.

### Menu

The main system menu is displayed on the screen and the user requested to input a two letter code corresponding to each option. While the program idles, waiting for keys to be pressed, the time is updated in the display at the top of the screen using the procedure PROCtime. Once a valid two letter code is entered the corresponding program is loaded and run. This program also contains the code for altering the sample description file maintained on the system disc which is added to a data file during permanent saving, although this code is not a procedure due to the restraints of the error handling mechanism it is described in the procedures section for clarity.

Procedures used are PROCdirectory, PROCtime and PROCtitle.

### Count

The scalar module defined by the variable scalar1% is used to count the PM pulses for a given number of 100th's of a second. The counts are then displayed on the screen and in the partitioned section. This process is repeated indefinitely until a key is pressed. The user is then able to repeat this operation or return to the menu.

Procedures used are PROCtitle, PROCcommand, PROCtime and FNminicam.

### Scan

Once the user enters the required axis to be scanned its previous direction of motion is displayed according to whether the relevant bit is set in the variable E%. The angle step and counting time are then enetered and the routine performs the step and count operation the required number of times. After each step the counts are displayed and the position of the axis held in memory updated and displayed in the partitioned section of the screen. The step-count repeat operation can be stopped prematurely by pressing any key. Once finished the routine can be rerun or return to the menu.

Procedures used are PROCtitle, PROCcommand, FNmotor, PROCstep_scan, PROCstep_count, PROCprint, PROCprint, PROCtime and FNminicam.

Step

Similar to SCAN but only drives an axis by a given angle after which it updates the postions held in memory.

Procedures used are PROCtitle, PROCcommand, PROCupdate, FNmotor, PROCstep_scan, PROCtime, PROCprint and FNminicam.

Plot

The chosen axis is repeatedly stepped and the PM pulses counted for a given number of times. The counts are displayed graphically while they are being collected with automatic scaling to display from the minimum count to the maximum count. If a new count is collected that is either greater than the previous maximum or smaller than the previous minimum the graph is erased and redrawn with a new scale. The repeated step_count operation can be stopped early by pressing any key, the graph is the redrawn with a new x-axis. The temporary data file is emptied and then filled with the counts collected and the angular step length. Once completed the axis is driven back to the starting point. Again the routine can be repeated or the user returned to the menu.

Procedures used are PROCaxes, FNminicam, PROCplot, PROCpoint, PROCerase, PROCredraw, PROCsave, PROCtitle, PROCcommand and PROCtime.

Replot

The filename containing the data is requested and if return is entered it is defaulted to the temporary file. If no drive number is given a default of 2 is added to the file name. The file is opened and if unsuccessful the error is trapped and the user requested for a new file name if the previously entered one does not exist or the name was too long. Data is read to determine the range of angles and counts and the user asked for which range to be plotted. Data is then read again from the disc to avoid entering it into memory and plotted on the screen. Again the routine can be repeated or the user returned to the manu.

Procedures used are PROCplot, PROCaxes, PROCpaper, PROCpspace, PROCmspace, PROCmove, PROCpoint, PROCdraw_rel, PROCnotch, PROClabel_x, PROClabel_y, PROCmark, PROCplotcs, PROCtitle, PROCtime and PROCcommand.

### Hplot

This routine is identical to the REPLOT routine but plots the graph on the hardcopy device attached to the printer port of the minicam interface. Devices catered for are the PIXY3 and HP7470A plotters. If the data is read from a permanent file the data consisting of the sample name, date, time and generator conditions is also read and added to the plot.

Procedures used are PROCplot, PROCaxes, PROCpaper, PROCpspace, PROCmspace, PROCmove, PROCpoint, PROCdraw_rel, PROCnotch, PROClabel_x, PROClabel_y, PROCmark, PROCplotcs, PROCtitle, PROCtime, PROCminicam and PROCcommand.

### Epsplot

Again this routine is identical to the REPLOT routine and plots the disc data on the screen. Once the data has been plotted on the screen the screen is dumped onto on Epson compatible printer. A standard screen dump routine is used.

### Calib

Allows positions of the axes, tilt goiniometers and scanning table to be altered. If return is pressed when a new position is requested the value is unaltered. If a position is altered the new value is displayed in the partitioned screen and the appropriate resident integer variable updated.

Procedure used is PROCtitle.

### Find

Similar in operation to the scan routine but instead of performing a given number of repeat step-counts it continues to step-count until either a count equal to a given limit is recorded or a given maximum angular scan range is exceeded. If a count greater than the limit is encountered the count is repeated to check that it was not due to spurious noise, if it is less than the limit the scan is recommenced. If the maximum scan range is reached and a scan in both directions from the start position was requested the axis is driven back to the start and step-counted in the opposite direction. The counts measured and the position of the axis are updated after each step-count. The routine can again be stopped before either of the stop conditions are encountered by pressing any key.

Procedures used are FNmotor, PROCstep_scan, PROCstep_count, PROCtitle, PROCcommand, PROCprint, FNminicam, PROCupdate and PROCtime.

### Tilt

Once a goniometer is chosen the previous direction is displayed according to the state of the relevant bit in the variable G%. The goniometer is tilted by driving the micrometer a given number of millimetres. If the given number would take the micrometer past either of its limits it is driven only to this limit and the user warned.

Procedures used are FNmotor, PROCstep_scan, PROCtitle, PROCcommand, FNminicam, PROCprint and PROCtime.

### Save

This routine reads the data stored in the temporary file and stores it in the given file. If temporary file cannot be opened the error is trapped and the user returned to the main menu. The sample description file is also read and the data added to the given file allowing a record of the x-ray, generator and sample conditions to be kept. The current date and time are also added to the file. If the file cannot be opened the error is trapped and the user prompted for another name. However, an existing file will be overwritten as the filing system does not detect this as an error.

Procedures used are PROCtitle and PROCcommand.

### Peak

This routine will keep the axis positioned either on top or on the flanks of a peak and is suitable for recording topographs over a long time period. The axis should be positioned as required before the routine is used. The count rate is measured at this position and the standard error calculated. The count rate is then measured every 5 seconds and if it lies outside the error bar the axis is moved until the count rate lies within the error bar. If the count rate drops to below the background rate the routine stops and reports an error. Axis positions are updated as required on the screen so that the drift can be measured.

Procedures used are FNmotor, PROCtitle, PROCcommand, FNminicam, PROCprint, PROCflank, PROCtop, PROCincrease, PROCdecrease, PROCdelay, PROCintensity.

### Auto

This routine only drives the second axis. The axis is scanned with steps and counting time as specified by the user until the peak is found. The criterion for finding a peak is that the count should be twice the background count. Once the peak is found the axis is stepped until backround is again reached. The peak is then scanned to find the maximum count. The goniometer is then tilted by the user supplied number of millimetres and the process of finding the peak maximum repeated. This process is repeated so that the peak height is measured with the tilt on either side of the starting position. The optimum setting is determined and the axis and goniometer moved to this position. The whole process is then repeated with the tilt step length halved. During the process of finding the peak the axis is scanned in one direction up to a maximum range specified by the user and then scanned in the other direction. If no peak can be found the routine stops and informs the user.

Procedures used are FNmotor, PROCtitle, PROCcommand, FNminicam, PROCprint, PROCtilt_opt, PROCtilt, PROCmove, PROCfind, PROCupdate and PROCtime.

Help

The help menu is displayed and the user requested for a two letter code corresponding to each of the choices. If a valid choice is made the data file on the system disc containing the relevant help text is opened and the data displayed on the screen as it is read. Pressing any key once this is done returns the user to the help menu. The option QUIT returns the user to the main system menu.

Procedures used are PROCtitle, PROCcommand and PROCdisplay.

Utilit

The utilities menu is displayed and a two letter code requested. If a valid choice is entered the relevent program is loaded from disc and run. This routine is, therefore, similar in operation to MENU but provides access to some of the less often used routines. Routines available at present are:

Bragg - calculates the Bragg angle for a given reflection, wavelength and lattice parameter.
Refl - calculates the single crystal reflectivity curve for a given material and structure factors assuming the crystal to be infinitely thick.
Spdata - produces a text file of rocking curve data as stored in another disc file. Is useful for data transfer to another computer.

Table

The XY scanning table is moved to the user requested position. The distance required to drive the micrometers is calculated by subtracting the requested position from the present position as held in memory. It is therefore important that the position in memory is correct. The position is maintained in millimetres in the partitioned screen.

Procedures used are PROCupdate, PROCtitle, PROCcommand, PROCtime, FNminicam and PROCprint.

COMMON PROCEDURES

PROCdirectory

The drive number is requested and if it lies between 0 and 3 the disc operating system command *CAT ‹Drive number› is called using the OSCLI (Operacting System Command Line Interpreter) vector at &FFF7. Pressing any key exits the procedure.

PROCupdate no parameters

The screen window corresponding to the partitioned region at the top of the screen is set using the VDU 28 command and inverse video selected. The number output format is set to 10 figures and 5 decimal places with the @% variable. The resident integer variables corresponding to the positions of the axes, goniometers and XY table are multiplied by their scaling factors held in step(1 .. 6) and printed in the relevant positions.

PROCtime no parameters

The number of seconds, minutes and hours are calculated from the TIME variable and printed in standard format in the partitioned region of the screen. Leading zeros are used if the values are less than 10. Note that TIME = ((hours*60 + minutes)*60 + seconds)*100.

PROCtitle (xcoord, ycoord, string)

Sets current screen to the partitioned region and selects inverse video then prints the string at the position xcoord, ycoord. Xcoord and ycoord are in text units. Screen is reset to main area minus top partition and bottom line.

PROCcommand (string)

Sets current screen to the bottom line only of the display. Selects inverse video and clears the screen. String is then printed centred on this line. Screen is then reset to main area minus the top partition and bottom line.

FNminicam (string, address, time, number) returns integer

Calls the procedure PROCtime to update the time display. Sets print output to the RS423 serial port and prints the following:

string;address;time;number;carriage_return

Print output is then set to dummy, ie no output, and input is selected as the serial port. One string is then input and the input is selected as the keyboard. The value of the string is then returned.

FNmotor (string) returns one integer

If the string is "I" the relevent bit in the variable E% is set (bit 0 is axis 1 and bit 1 is axis 2) otherwise the bit is cleared. According to which axis is selected (held in the variable M%) a stepper motor number is selected from the variables held in AX%(1..4). Allowance is made in the selection for the fact that driving the motor on both axes in the same direction drives the axes in opposite senses of theta. The variable P% is set to the number of motor steps (MO%) with the sign according to whether increasing or decreasing theta. P% can then be used to update the position counters S%,T%,U% and V%.

PROCstep_scan no parameters

FNmotor is called to get the address of the stepper motor. The position of the first count value to be printed is set in the variables X% and Y%. PROCcommand is called to display the 'Press any key to stop' message. A tight loop is then executed in which PROCstep_count is called to step the axis and count the PM pulses and the count returned printed at the position X%,Y% on the main screen. X% and Y% are increased to the next position, if the bottom of the screen is reached two linefeeds are printed and Y% kept at this line number. The number of step-counts is counted and the loop stops if this count equals the number selected by the user or if a key has been pressed. The procedure then terminates.

PROCstep_count (address, number, time)

The minicam controlled stepper motor with address ‹address› is moved ‹number› of steps with speed given by speed%. The position of the axis is then updated and displayed on the screen by calling the procedure PROCprint. The minicam scalar is then used to count for ‹time› 100th's of a second and the count returned displayed on the screen by calling the procedure PROCtitle. The variable R% is returned containing the count.

PROCprint (xcoord, ycoord, number)

Similar in operation to PROCtitle but the value of ‹number› number is printed at the position xcoord, ycoord in the partitioned screen.

PROCaxes no parameters

Routine to draw axes on the currently selected output device. Since this routine is independent of the output device being used it can be used for either the screen or hardcopy devices. The x and y ranges are determined from the maximum and minimum coordinates of the plotter space currently being used. Once the range is calculated the tick spacing is determined by the procedure PROCmark. The axes are then drawn using the procedures PROCdraw_rel and PROCnotch which respectively draw a line relative to the present cursor position and draw a tick. The axes are then labelled using the procedures PROClabel_x and PROClabel_y.

PROCpspace (xmin, xmax, ymin, ymax)

This routine sets the scaling factors required to produce output on the plotter device. The plotting space is defined assuming the total area runs from 0,0 to 1,1. The users coordinates should then be mapped onto this area by calling PROCmspace. The variables xscale% and yscale% are the actual ranges of numbers possible for the graphic device.

PROCmspace (xmin, xmax, ymin, ymax)

This routine scales the plotter space being used, as defined by calling PROCpspace, to the user units. Consequent calls to draw lines etc can then be called with variables in the user units - the scaling factors produced by this procedure are then used to convert the user units into the device units.

PROCmove (xcoord, ycoord)

Routine to move the cursor or pen to the coordinates xcoord, ycoord. Xcoord and ycoord are in user units and are converted to the device coordinates using the scaling factors determined by a previous call to PROCpspace and PROCmspace. This routine is device dependent, the particular device command for moving the cursor/pen is used.

PROCpoint (xcoord, ycoord)

PROCmove is called to move the cursor/pen to the position xcoord, ycoord and a dot is then made. This routine is device dependent, the command to produce a dot depends on the device.

PROCdraw_rel (xinc, yinc)

This routine calls the device's relative drawing command after scaling the variables xinc and yinc.

PROCnotch (switch)

If switch is 1 a tick in the y direction is drawn and if swicth is 2 a tick in the x direction is drawn. This routine is device dependent, the device's relative drawing command is used.

PROClabel_x (xcoord)

The cursor/pen is moved to the xcoord position along the x axis and the value of xcoord is written below the position. The device's printing commands are used so it is not device indepenedent.

PROClabel_y (ycoord)

Similar to the above routine but writes the value of ycoord to the left of the position.

PROCmark (range)

According to the magnitude of range a suitable value of tick spacing is chosen and returned in the variable SX (which is in user units). Logarithms to the base ten are used to convert range into a variable in the range >=1 and <10.

PROCplotcs (xcoord, ycoord, string)

PROCmove is called to move the cursor/pen to the position xcoord, ycoord and the string is then printed using the device's printing command. It is, therefore, device dependent.

PROCdraw (xcoord, ycoord)

Draws a line from the present position to the position xcoord, ycoord after scaling to the device units. Calls the device's absolute drawing routine so it is device dependent.

PROCpen (pen)

Selects the plotter pen by calling the device's pen select command for choosing pen number <pen>.

PROCminicam (string)

This is very similar to the function FNminicam but only the string is printed followed by a carriage return. It is used by the hardcopy plotting routines to pass the string to the minicam interface which then passes the string to the device via the serial printer port. This enables the plotters to be used without unplugging the minicam from the BBC and also converts to the plotter's baud rate and word format.

PROCflank no parameters

Routine that adjusts the position of an axis to keep on the flank of a peak. The count rate at the starting position is determined by calling FNintensity which counts the PM pulses for 5 seconds. The standard error is then determined and if the count rate strays more than three standard errors from the start position PROCincrease or PROCdecrease are called to move the motor. If the count rate is in the required range a five second delay is used to prevent needless hunting.

PROCtop no parameters

Similar to PROCflank but adjusts the axis so as to remain on the top of a peak. A slightly tighter adjustment loop is used as the direction required to move to refind the peak is not known, for PROCflank it is known which direction will increase or decrease the count rate depending on which side of the peak is being used.

PROCincrease no parameters

Moves the stepper motor determined by calling FNmotor a given number of steps with speed given by speed%. Display and resident variables are updated as required.

PROCdecrease no parameters

Similar to PROCincrease but moves the motor in the opposite direction.

PROCfind no parameters

The present position of the second axis is saved as now%. The stepper motor address is determined for increasing theta by calling FNmotor and the command 'looking for peak' displayed on the bottom line of the display. PROCstep_count is then called repeatedly until a count greater than twice the background is found or the maximum scan range reached. If the maximum scan range is reached the motor is driven back to the start position and step scanned in the opposite direction. If a peak is not found the variable fault is set to true and the routine exited. If a peak is found stepping is continued in the decreasing theta direction until background is found again. The peak is then scanned in the increasing theta direction until a maximum is found. The maximum count and the axis position of the maximum are returned in the variables maximum% and maxpos% respectively.

PROCsave no parameters

The temporary data file on the system disc (D.RTEMP) is deleted by calling the disc operating system's delete file function. This is called by setting up a file descriptor block in memory containing the file name. X% and Y% are then set to the low and high bytes of a memory address that contains the address of the file descriptor block. A% is set to 6 and the disc operating system called via the vector OSFILE at &FFDD. The temporary data file is then opened and the data consisting of the number of counts, the angular step between the counts, and the counts is then stored in the file. This process prevents the 'Can't extend' error which occurs if an existing file is opened for output and more data stored in it than when it was first created.

Sample descriptor file updating

This routine updates the sample descriptor file maintained on the system disc. It is not a procedure since it needs to trap the 'file doesn't exist' error when it is first run and no file exists. The file is opened and the contents displayed on the screen so that they can be altered. Once the user has finished updating the descriptions entering return to the command prompt writes the data back to the disc file. Data that is maintained consists of seven strings; the experimental site, the sample name, the generator voltage, the generator current, the first crystal, the wavelength and the reflection being used.

## DATA FORMAT USED FOR THE STORAGE OF ROCKING CURVES

Rocking curve data is stored in disc files in the following format:

1. Number of counts in the file (integer).
2. Angular step between each count in arc seconds (real).
3. The actual counts stored sequentially as integers.
4. The date (string).
5. The time (integer).
6. Experimental site name (string).
7. Sample name (string).
8. Generator voltage (string).
9. Generator current (string).
10. First crystal (string).
11. Wavelength (string).
12. Reflection (string).

Since BBC Basic stores numbers and strings in internal format these files cannot be typed or transferred to another computer without first converting to a text file. This operation is easily performed using the SPOOL DATA option avaiable in the utilities package.

```
 10 REM ----------------------------------------
 20 REM -----  THIS IS THE STARTING PROGRAM  -----
 30 REM ----------------------------------------
 40 :
 50 MODE0
 60 DIM step(6): step(1)=0.1776: step(2)=0.121: step(3)=1.0/(96.0*50):
    step(4)=1.0/(96.0*50)
 70 step(5)=1.0/96.0: step(6)=1.0/96.0
 80 PROCinit
 90 PRINT''" This is the new double crystal control program - welcome"
100 INPUT''" Please enter your name "name$
110 PROCtitle(12,1,name$)
120 PRINTTAB(0,7); : INPUT"  and your group "group$
130 PROCtitle(12,2,group$)
140 PRINTTAB(0,9); : INPUT"  and the date (dd/mm/yy) "date$
150 PROCtitle(12,3,date$)
160 PRINTTAB(0,11); : INPUT"  and the time ho,mi,se "ho%,mi%,se%
170 TIME=((ho%*60+mi%)*60+se%)*100
180 PROCtime
190 PRINTTAB(0,13); : INPUT"  Enter the micrometer reading for the 1st
    goniometer "a
200 U%=a/step(3)
210 PROCupdate
220 PRINTTAB(0,15); : INPUT"  and the reading for the 2nd goniometer "a
230 V%=a/step(4)
240 PROCupdate
250 PRINTTAB(0,17); : INPUT"  Enter the generator H.T. (kV) "kV
260 PROCtitle(64,4,STR$(kV))
270 PRINTTAB(0,19); : INPUT"  and the generator current (mA) "mA
280 PROCtitle(64,5,STR$(mA))
290 CHAIN"P.MENU"
300 :
310 REM init procedure - produces table of axes etc
320 :
330 DEFPROCinit
340 VDU28,0,6,79,0: REM text window at top of screen
350 COLOUR0: COLOUR129: CLS
360 PRINT TAB(3,1); "USER"; TAB(3,2); "GROUP"; TAB(3,3); "DATE";
    TAB(3,4); "TIME"; TAB(3,5 ); "OPTION";
370 PRINT TAB(25,1); "Axis 1"; TAB(25,2); "Axis 2"; TAB(25,3); "Gon  1";
    TAB(25,4); "Gon  2"; TAB(25,5); "Slit 1"
380 PRINT TAB(44,1)"Scan-Table (mm)"; TAB(44,2)"X"; TAB(44,3)"Y";
390 PRINT TAB(61,1); "Scint."; TAB(61,2); "Ionis."; TAB(61,4); "kV";
    TAB(61,5); "mA";
400 PROCupdate
410 VDU28,0,31,79,7: COLOUR1: COLOUR128: CLS
420 *FX8,6
430 *FX7,6
440 ENDPROC
450 :
460 DEFPROCupdate
470 VDU28,0,6,79,0: COLOUR0: COLOUR129
480 @%=&2050A
490 PRINTTAB(34,1); S%*step(1)/3600; TAB(34,2); T%*step(2)/3600;
    TAB(34,3); U%*step(3); TAB(34,4); V%*step(4)
500 PRINT TAB(47,2); K%*step(5); TAB(47,3); L%*step(6);
510 @%=10
520 VDU28,0,31,79,7: COLOUR1: COLOUR128
530 ENDPROC
540 :
```

```
550 DEFPROCtitle(X%,Y%,string$)
560 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); string$;
570 VDU28,0,31,79,7: COLOUR1: COLOUR128
580 ENDPROC
590 :
600 DEFPROCtime
610 se%=(TIME DIV 100) MOD 60
620 mi%=(TIME DIV 6000)MOD 60
630 ho%=(TIME DIV 360000)MOD 24
640 VDU28,0,6,79,0: COLOUR0: COLOUR129
650 PRINTTAB(12,4); : IF ho%<10 PRINT"0"; ho%;  ELSE PRINT; ho%;
660 IF mi%<10 PRINT":0"; mi%;  ELSE PRINT":"; mi%;
670 IF se%<10 PRINT":0"; se%;  ELSE PRINT":"; se%;
680 VDU28,0,31,79,7: COLOUR1: COLOUR128
690 ENDPROC


 10 REM -----------------------------------------------------
 20 REM -------  This program prints the menu, asks for input ------
 30 REM -------  then chains the relevant program.           ------
 40 REM -----------------------------------------------------
 50 DIM C 20
 60 PROCtitle(12,5,"MENU")
 70 CLS
 80 X%=20
 90 PRINT
100 PRINTTAB(X%); "Calibrate axis positions.............CA"
110 PRINTTAB(X%); "Count................................CO"
120 PRINTTAB(X%); "Scan axis and count..................SC"
130 PRINTTAB(X%); "Step axis without count..............SS"
140 PRINTTAB(X%); "Tilt goniometer......................TI"
150 PRINTTAB(X%); "Find peak............................FI"
160 PRINTTAB(X%); "Plot rocking curve and collect data..PL"
170 PRINTTAB(X%); "Replot data from disc................RP"
180 PRINTTAB(X%); "Epson printer plot of disc data......EP"
190 PRINTTAB(X%); "Automatic tilt optimise..............AU"
200 PRINTTAB(X%); "Peak hold............................PK"
210 PRINTTAB(X%); "Move scanning table..................MT"
220 PRINTTAB(X%); "Disc directory of data...............DI"
230 PRINTTAB(X%); "Save temporary data file.............SA"
240 PRINTTAB(X%); "Update your data header file.........UP"
250 PRINTTAB(X%); "Run the utilities package............UT"
260 PRINTTAB(X%); "Help.................................HE"
270 PRINT
280 PRINTTAB(0,20); "                                        "
290 COLOUR0: COLOUR129: PRINTTAB(0,20); "Enter command >> ";
300 COLOUR1: COLOUR128
310 choice$=""
320 REPEAT
330   A$=INKEY$(50)
340   PROCtime
350   IF A$<>"" choice$=choice$+A$: PRINTTAB(18,20); choice$;
355   IF ASC(A$)=127 AND LEN(choice$)=1 choice$="" ELSE IF ASC(A$)=127
      choice$=LEFT$(choice$,LEN(choice$)-2)
360   UNTIL LEN(choice$)=3
370 choice$=LEFT$(choice$,2)
380 IF choice$="CA" CHAIN"P.CALIB"
390 IF choice$="CO" CHAIN"P.COUNT"
400 IF choice$="SC" CHAIN"P.SCAN"
410 IF choice$="SS" CHAIN"P.STEP"
```

```
420 IF choice$="TI" CHAIN"P.TILT"
430 IF choice$="FI" CHAIN"P.FIND"
440 IF choice$="PL" CHAIN"P.PLOT"
450 IF choice$="RP" CHAIN"P.REPLOT"
460 IF choice$="MT" CHAIN"P.TABLE"
470 IF choice$="HP" CHAIN"P.HPLOT"
480 IF choice$="AU" CHAIN"P.AUTO"
490 IF choice$="PK" CHAIN"P.PEAK"
500 IF choice$="EP" CHAIN"P.EPSPLOT"
510 IF choice$="DI" PROCdirectory: GOTO10
520 IF choice$="UP" GOTO780
530 IF choice$="SA" CHAIN"P.SAVE"
540 IF choice$="UT" CHAIN"P.UTILIT"
550 IF choice$="HE" CHAIN"P.HELP"
560 VDU7: GOTO280
570 :
580 DEFPROCtitle(X%,Y%,string$)
590 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); string$;
600 FORI%=1TO(10-LEN(string$)): VDU32: NEXT
610 VDU28,0,31,79,7: COLOUR1: COLOUR128
620 ENDPROC
630 :
640 DEFPROCdirectory
650 PROCtitle(12,5,"DIRECT.")
660 CLS
670 INPUT''" Which drive's directory do you require "drive%
680 PRINT''
690 IF drive%<0 AND drive%>3 CLS: PRINT'" Illegal drive number - try
    again": GOTO670
700 CLS
710 A$="CAT"+STR$(drive%): $C=A$: X%=C MOD 256: Y%=C DIV 256: CALL &FFF7
720 VDU28,0,31,79,31: COLOUR0: COLOUR129: CLS: A$="Press any key for
    menu"
730 PRINTTAB(INT(40-LEN(A$)/2)); A$;
740 VDU28,0,31,79,7: COLOUR1: COLOUR128
750 A%=GET
760 ENDPROC
770 :
780 PROCtitle(12,5,"UPDATE")
790 CLS
800 PRINT''"  Reading your header file from disc"
810 ON ERROR GOTO 1220
820 X%=OPENIN("D.HEADER")
830 INPUT#X%,place$
840 INPUT#X%,sname$
850 INPUT#X%,volt$
860 INPUT#X%,current$
870 INPUT#X%,first_crystal$
880 INPUT#X%,wave$
890 INPUT#X%,refl$
900 CLOSE#X%
910 CLS
920 PRINTTAB(2,2); " Your current header file contains the following
    information".
930 PRINTTAB(5,4)"1. Experiment running at
    "; TAB(30,4); place$
940 PRINTTAB(5,6); "2. Sample name                    "; TAB(20,6);
    sname$
950 PRINTTAB(5,8); "3. Generator voltage              "; TAB(26,8); volt$
960 PRINTTAB(5,10); "4. Generator current               "; TAB(26,10);
```

```
    current$
 970 PRINTTAB(5,12); "5. First crystal                        "; TAB(22,12);
     first_crystal$
 980 PRINTTAB(5,14); "6. Wavelength                         "; TAB(19,14);
     wave$
 990 PRINTTAB(5,16); "7. Reflection being used                               ";
     TAB(31,16); refl$
1000 PRINTTAB(1,20); "                                                          ";
1010 COLOUR0: COLOUR129: PRINTTAB(1,18); "  Enter choice to update or <CR>
     for menu "
1020 COLOUR1: COLOUR128: choice=GET
1030 IF choice=13 GOTO 1120
1040 IF choice=&31 INPUT'"  Enter experimental site "place$: GOTO920
1050 IF choice=&32 INPUT'"  Enter sample name "sname$: GOTO920
1060 IF choice=&33 INPUT'"  Enter generator voltage "volt$: GOTO920
1070 IF choice=&34 INPUT'"  Enter generator current "current$: GOTO920
1080 IF choice=&35 INPUT'"  Enter first crystal "first_crystal$: GOTO920
1090 IF choice=&36 INPUT'"  Enter wavelength "wave$: GOTO920
1100 IF choice=&37 INPUT'"  Enter reflection being used "refl$: GOTO920
1110 VDU7: GOTO920
1120 X%=OPENOUT("D.HEADER")
1130 PRINT#X%,place$
1140 PRINT#X%,sname$
1150 PRINT#X%,volt$
1160 PRINT#X%,current$
1170 PRINT#X%,first_crystal$
1180 PRINT#X%,wave$
1190 PRINT#X%,refl$
1200 CLOSE#X%
1210 GOTO1250
1220 IF ERR=222 CLS: PRINTTAB(5,2); "You have no header file as yet"
1230 place$="": sname$="": volt$="": current$="": first_crystal$="":
     wave$="": refl$=""
1240 IF ERR=222 GOTO 930 ELSE REPORT: END
1250 GOTO10
1260 :
1270 DEFPROCtime
1280 se%=(TIME DIV 100)MOD 60: mi%=(TIME DIV 6000)MOD 60: ho%=(TIME DIV
     360000)MOD 24
1290 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(12,4);
1300 IF ho%<10 PRINT"0"; ho%;  ELSE PRINT; ho%;
1310 IF mi%<10 PRINT":0"; mi%;  ELSE PRINT":"; mi%;
1320 IF se%<10 PRINT":0"; se%;  ELSE PRINT":"; se%;
1330 VDU28,0,30,79,7: COLOUR1: COLOUR128
1340 ENDPROC
```

```
  10 REM ------------------------------
  20 REM -----  COUNTING ROUTINE  -----
  30 REM ------------------------------
  40 :
  50 scaler1%=56: scaler2%=55
  60 PROCtitle(12,5,"COUNT")
  70 CLS: INPUT''"  Enter counting time, in 100th's secs "TT%
  80 PROCcommand("Press any key to stop")
  90 X%=5: Y%=5
 100 REPEAT
 110   R%=FNminicam("CO1",scaler1%,TT%,0) + 256*256*FNminicam("RD2",
       scaler2%,0,0)
 120   R$=STR$(R%): PROCtitle(68,1,R$)
```

```
130    PRINT TAB(X%,Y%); R%;
140    X%=X%+15: IF X%=80 X%=5: Y%=Y%+1
150    IF (Y%=23 AND X%=5) VDU10,10: Y%=22
160    UNTIL INKEY(0)<>-1
170 PROCcommand("Press <CR> for repeat, any other key for menu")
180 A=GET
190 IF A=13 PROCcommand(" "): GOTO 70
200 CHAIN"P.MENU"
210 :
220 DEFPROCtitle(X%,Y%,string$)
230 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "            ";
240 PRINTTAB(X%,Y%); string$;
250 VDU28,0,30,79,7
260 COLOUR1: COLOUR128
270 ENDPROC
280 :
290 DEFPROCcommand(string$)
300 VDU28,0,31,79,31: COLOUR0: COLOUR129: CLS
310 PRINTTAB(INT(40-LEN(string$)/2)); string$;
320 VDU28,0,30,79,7: COLOUR1: COLOUR128
330 ENDPROC
340 :
350 DEFFNminicam(usr$,AD%,NN%,TT%)
360 PROCtime
370 *FX3,7
380 PRINTusr$; ","; AD%; ","; NN%; ","; TT%; CHR$(13);
390 *FX3,6
400 *FX2,1
410 INPUT""return$
420 *FX2,2
430 *FX3,0
440 =VAL(return$)
450 :
460 DEFPROCtime
470 se%=(TIME DIV 100)MOD 60: mi%=(TIME DIV 6000)MOD 60: ho%=(TIME DIV
    360000)MOD 24
480 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(12,4);
490 IF ho%<10 PRINT"0"; ho%;  ELSE PRINT; ho%;
500 IF mi%<10 PRINT":0"; mi%;  ELSE PRINT":"; mi%;
510 IF se%<10 PRINT":0"; se%;  ELSE PRINT":"; se%;
520 VDU28,0,30,79,7: COLOUR1: COLOUR128
530 ENDPROC
```

```
 10 REM -----------------------------------------------------
 20 REM -----    STEP SCAN ROUTINE; SCANS AXIS AND COUNTS    -----
 30 REM -----------------------------------------------------
 40 DIM AX%(8)
 50 AX%(1)=34: AX%(2)=35: AX%(3)=36: AX%(4)=37
 60 AX%(5)=33: AX%(6)=32: AX%(7)=33: AX%(8)=32
 70 scaler1%=56: scaler2%=55: speed%=20
 80 DIM step(2)
 90 step(1)=0.1776: step(2)=0.121
100 PROCtitle(12,5,"SCAN-COUNT")
110 CLS
120 INPUT''" Which axis to scan (1 or 2) "M%
130 IF M%<>1 AND M%<>2 THEN 110
140 F%=(E%ANDM%)
150 IF F% increasing=TRUE ELSE increasing=FALSE
160 PRINT'" Previous motion was ";
```

```
170 IF increasing PRINT"increasing"; ELSE PRINT "decreasing";
180 PRINT" theta"
190 INPUT'" Counting time, in 100th's secs "TT%
200 INPUT'" Step length per cycle (secs) "step_length
210 MO%=INT(step_length/step(M%))
220 INPUT'" Number of cycles "NU%
230 INPUT'" Increasing or decreasing theta (I/D) "inc$
240 PROCstep_scan
250 PROCcommand("Press <CR> for repeat, any other key for menu")
260 A=GET: IF A=13 PROCcommand(" "): GOTO 100
270 CHAIN"P.MENU"
280 :
290 DEFFNmotor(dir$)
300 IFdir$="I" E%=(E% OR M%) ELSE E%=(E% AND(&FF EOR M%))
310 IF dir$="I" THEN F%=1 ELSE F%=0
320 IF M%=1 THEN stepper%=AX%(M%+1-F%)
330 IF M%=2 THEN stepper%=AX%(M%+1+F%)
340 I%=0: IF dir$="I" P%=MO% ELSE P%=-MO%
350 =stepper%
360 :
370 DEFPROCstep_scan
380 IF NU%=0 ENDPROC
390 stepper%=FNmotor(inc$)
400 X%=5: Y%=15
410 PROCcommand("Press any key to stop")
420 REPEAT
430   I%=I%+1
440   PROCstep_count(stepper%,MO%,TT%)
450   PRINTTAB(X%,Y%); R%;
460   X%=X%+15: IFX%=80 X%=5: Y%=Y%+1
470   IF (Y%=23 AND X%=5) VDU10,10: Y%=22
480   Z%=INKEY(0)
490   UNTIL I%=NU% OR Z%<>-1
500 ENDPROC
510 :
520 DEFPROCtitle(X%,Y%,string$)
530 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "            ";
540 PRINTTAB(X%,Y%); string$;
550 VDU28,0,30,79,7
560 COLOUR1: COLOUR128
570 ENDPROC
580 :
590 DEFPROCcommand(string$)
600 VDU28,0,31,79,31: COLOUR0: COLOUR129: CLS
610 PRINTTAB(INT(40-LEN(string$)/2)); string$;
620 VDU28,0,30,79,7: COLOUR1: COLOUR128
630 ENDPROC
640 :
650 DEFPROCstep_count(stepper%,MO%,TT%)
660 A%=FNminicam("ST1",stepper%,MO%,speed%)
670 @%=&2050A
680 IF M%=1 S%=S%+P%: R=S%*step(1)/3600: PROCprint(34,1,R)
690 IF M%=2 T%=T%+P%: R=T%*step(2)/3600: PROCprint(34,2,R)
700 @%=10
710 R%=FNminicam("CO1",scaler1%,TT%,0)+256*256*FNminicam("RD2",
    scaler2%,0,0)
720 R$=STR$(R%): PROCtitle(68,1,R$)
730 ENDPROC
740 :
750 DEFFNminicam(usr$,AD%,NN%,TT%)
```

```
 760 PROCtime
 770 *FX2,2
 780 *FX3,7
 790 PRINTusr$; ","; AD%; ","; NN%; ","; TT%; CHR$(13);
 800 *FX3,6
 810 *FX2,1
 820 INPUT""reply$
 830 *FX2,2
 840 *FX3,0
 850 =VAL(reply$)
 860 :
 870 DEFPROCprint(X%,Y%,R)
 880 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "                    "
 890 PRINTTAB(X%,Y%); R;
 900 VDU28,0,30,79,7
 910 COLOUR1: COLOUR128
 920 ENDPROC
 930 :
 940 DEFPROCtime
 950 se%=(TIME DIV 100)MOD 60: mi%=(TIME DIV 6000)MOD 60: ho%=(TIME DIV
     360000)MOD 24
 960 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(12,4);
 970 IF ho%<10 PRINT"0"; ho%;  ELSE PRINT; ho%;
 980 IF mi%<10 PRINT":0"; mi%;  ELSE PRINT":"; mi%;
 990 IF se%<10 PRINT":0"; se%;  ELSE PRINT":"; se%;
1000 VDU28,0,30,79,7: COLOUR1: COLOUR128
1010 ENDPROC
```

```
  10 REM -------------------------------------------
  20 REM -----    STEP ROUTINE; SCANS AXIS ONLY  -----
  30 REM -------------------------------------------
  40 DIM AX%(8)
  50 AX%(1)=34: AX%(2)=35: AX%(3)=36: AX%(4)=37
  60 AX%(5)=33: AX%(6)=32: AX%(7)=33: AX%(8)=32
  70 scaler1%=56: scaler2%=55: speed%=20
  80 DIM step(2)
  90 step(1)=0.1776: step(2)=0.121
 100 PROCtitle(12,5,"SCAN-AXIS")
 110 CLS
 120 INPUT''"  Which axis to scan (1 or 2) "M%
 130 IF M%<>1 AND M%<>2 THEN 110
 140 F%=(E%ANDM%)
 150 IF F% increasing=TRUE ELSE increasing=FALSE
 160 PRINT'"  Previous motion was ";
 170 IF increasing PRINT"increasing";  ELSE PRINT "decreasing";
 180 PRINT" theta"
 190 INPUT'"  Scan length required (secs) "step_length
 200 MO%=INT(step_length/step(M%))
 210 INPUT'"  Increasing or decreasing theta (I/D) "inc$
 220 PRINT''"  Driving axis number "; M%
 230 PROCstep_scan
 240 VDU7
 250 PROCcommand("Press <CR> for repeat, any other key for menu")
 260 A=GET: IF A=13 PROCcommand(" "): GOTO 100
 270 CHAIN"P.MENU"
 280 :
 290 DEFFNmotor(dir$)
 300 IFdir$="I" E%=(E% OR M%) ELSE E%=(E% AND(&FF EOR M%))
 310 IF dir$="I" THEN F%=1 ELSE F%=0
```

```
320 IF M%=1 THEN stepper%=AX%(M%+1-F%)
330 IF M%=2 THEN stepper%=AX%(M%+1+F%)
340 I%=0: IF dir$="I" P%=MO% ELSE P%=-MO%
350 =stepper%
360 :
370 DEFPROCstep_scan
380 stepper%=FNmotor(inc$)
390 IF MO%>&7FFF A=FNminicam("ST1",stepper%,&7FFF,speed%): MO%=MO%-&7FFF:
    GOTO390
400 PROCupdate
410 A=FNminicam("ST1",stepper%,MO%,speed%)
420 PROCupdate
430 ENDPROC
440 :
450 DEFPROCtitle(X%,Y%,string$)
460 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "                ";
470 PRINTTAB(X%,Y%); string$;
480 VDU28,0,30,79,7
490 COLOUR1: COLOUR128
500 ENDPROC
510 :
520 DEFPROCcommand(string$)
530 VDU28,0,31,79,31: COLOUR0: COLOUR129: CLS
540 PRINTTAB(INT(40-LEN(string$)/2)); string$;
550 VDU28,0,30,79,7: COLOUR1: COLOUR128
560 ENDPROC
570 :
580 DEFPROCupdate
590 @%=&2050A
600 IF M%=1 S%=S%+P%: R=S%*step(1)/3600: PROCprint(34,1,R)
610 IF M%=2 T%=T%+P%: R=T%*step(2)/3600: PROCprint(34,2,R)
620 @%=10
630 ENDPROC
640 :
650 DEFFNminicam(usr$,AD%,NN%,TT%)
660 PROCtime
670 *FX2,2
680 *FX3,7
690 PRINTusr$; ","; AD%; ","; NN%; ","; TT%; CHR$(13);
700 *FX3,6
710 *FX2,1
720 INPUT""reply$
730 *FX2,2
740 *FX3,0
750 =VAL(reply$)
760 :
770 DEFPROCprint(X%,Y%,R)
780 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "              "
790 PRINTTAB(X%,Y%); R;
800 VDU28,0,30,79,7
810 COLOUR1: COLOUR128
820 ENDPROC
830 :
840 DEFPROCtime
850 se%=(TIME DIV 100)MOD 60: mi%=(TIME DIV 6000)MOD 60: ho%=(TIME DIV
    360000)MOD 24
860 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(12,4);
870 IF ho%<10 PRINT"0"; ho%;  ELSE PRINT; ho%;
880 IF mi%<10 PRINT":0"; mi%;  ELSE PRINT":"; mi%;
890 IF se%<10 PRINT":0"; se%;  ELSE PRINT":"; se%;
```

```
900 VDU28,0,30,79,7: COLOUR1: COLOUR128
910 ENDPROC
```

```
  1 REM -------------------------------------------------------
  2 REM ----  PLOT; COLLECTS DATA AND DISPLAYS IN REAL TIME ----
  3 REM -------------------------------------------------------
 10 DIM AX%(4),step(2),CO%(400),C% 20: scaler1%=56: scaler2%=55:
    speed%=50
 20 AX%(1)=34: AX%(2)=35: AX%(3)=36: AX%(4)=37
 30 step(1)=0.1776: step(2)=0.121
 40 max%=&7FFFFFFF: min%=0: min_count%=&7FFFFFFF: max_count%=0
 50 xorig%=100: yorig%=100
 60 PROCtitle(12,5,"PLOT")
 70 CLS
 80 INPUT''" Which axis to scan "M%
 90 PRINT'"  Recording data can only be performed with increasing theta"
100 INPUT'"  Counting time, in 100th's secs "TT%
110 INPUT'"  Step length per cycle (secs) "step_length
120 MO%=step_length/step(M%)
130 INPUT'"  Number of data points required "NU%
140 xstep=(1200-xorig%)/NU%: ystep=(700-yorig%)/(max%-min%)
150 CLS
160 PROCsetup
170 PROCaxes: VDU5: MOVE100,85: PRINT"0": MOVE1100,85: PRINT
    NU%*MO%*step(M%)
180 MOVE0,725: PRINT""; max%: MOVE0,yorig%: PRINT""; min%: VDU4
190 PROCcommand("Press any key to stop")
200 PROCplot
210 PROCerase: xstep=(1200-xorig%)/I%: PROCredraw
220 PROCsave
230 VDU7
240 PROCcommand("Press <CR> to repeat, any other key for menu")
250 REPEAT: A=INKEY(50): PROCtime: UNTIL A<>-1
260 IF A=13 PROCcommand(" "): RUN
270 CHAIN"P.MENU"
280 DEFPROCtitle(X%,Y%,string$)
290 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "                    "
300 PRINTTAB(X%,Y%); string$;
310 VDU28,0,30,79,7: COLOUR1: COLOUR128
320 ENDPROC
330 DEFPROCcommand(string$)
340 VDU28,0,31,79,31: COLOUR0: COLOUR129: CLS
350 PRINTTAB(INT(40-LEN(string$)/2)); string$;
360 VDU28,0,30,79,7: COLOUR1: COLOUR128
370 ENDPROC
380 DEFPROCprint(X%,Y%,R)
390 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "                    ";
400 PRINTTAB(X%,Y%); R;
410 VDU28,0,30,79,7: COLOUR1: COLOUR128
420 ENDPROC
430 DEFPROCaxes
440 MOVExorig%,yorig%: DRAW1200,yorig%: MOVExorig%,yorig%: DRAW
    xorig%,675
450 FOR I%=1 TO 9: J%=(1200-xorig%)*I%/10: MOVE J%+xorig%,yorig%
460   DRAW J%+xorig%,yorig%+10: NEXT
470 FORI%=1TO9: J%=(700-yorig%)*I%/10: MOVExorig%,J%+yorig%
480   DRAW xorig%+10,J%+yorig%: NEXT
490 MOVE600,80: VDU5: PRINT"Seconds": VDU4
500 ENDPROC
510 :
520 DEFFNminicam(usr$,AD%,NN%,TT%)
530 PROCtime
540 *FX3,7
550 PRINTusr$; ","; AD%; ","; NN%; ","; TT%; CHR$(13);
560 *FX3,6
570 *FX2,1
580 INPUT""reply$
590 *FX2,2
600 *FX3,0
610 =VAL(reply$)
620 :
630 DEFPROCplot
640 IF M%=1 stepper%=AX%(1) ELSE stepper%=AX%(4)
650 MOVExorig%,yorig%
660 I%=0
```

```
670 REPEAT
680   I%=I%+1: A=FNminicam("ST1",stepper%,MO%,speed%)
690   @%=&2050A
700   IF M%=1 S%=S%+MO%: R=S%*step(1)/3600: PROCprint(34,1,R)
710   IF M%=2 T%=T%+MO%: R=T%*step(2)/3600: PROCprint(34,2,R)
720   @%=10
730   R%=FNminicam("CO1",scaler1%,TT%,0)+256*256*FNminicam("RD2",
      scaler2%,0,0)
740   CO%(I%)=R%
750   IF R%>max_count% max_count%=R%
760   IF R%<min_count% min_count%=R%
770   PROCtitle(68,1,STR$(R%))
780   PROCpoint(I%,R%)
790   UNTIL I%=NU% OR INKEY(0)<>-1
800 PROCcommand("Stopped....rewinding axis to start position")
810 IF M%=1 stepper%=AX%(2) ELSE stepper%=AX%(3)
820 A=FNminicam("ST1",stepper%,MO%*I%,speed%)
830 @%=&2050A
840 IF M%=1 S%=S%-MO%*I%: R=S%*step(1)/3600: PROCprint(34,1,R)
850 IF M%=2 T%=T%-MO%*I%: R=T%*step(2)/3600: PROCprint(34,2,R)
860 @%=10
870 ENDPROC
880 :
890 DEFPROCpoint(I%,R%)
900 PLOT69,I%*xstep+xorig%,(R%-min%)*ystep+yorig%
910 IF I%<5 ENDPROC
920 IF I%=5 PROCerase: ystep=(700-yorig%)/(max_count%-min_count%):
    max%=max_count%: min%=min_count%: PROCredraw: ENDPROC
930 IF max_count%<=max% AND min_count%>=min% ENDPROC
940 PROCerase: ystep=(700-yorig%)/(max_count%-min_count%):
    max%=max_count%: min%=min_count%: PROCredraw
950 ENDPROC
960 DEFPROCsetup
970 VDU24,0; 0; 1279; 740;
980 ENDPROC
990 DEFPROCerase
1000 FOR J%=1 TO I%: PLOT70,J%*xstep+xorig%,(CO%(J%)-min%)*ystep+yorig%:
     NEXT
1010 MOVE0,yorig%: VDU5: PRINT"       "; : FORK%=1TO6: VDU127: NEXT: VDU4
1020 MOVE0,725: VDU5: PRINT"                "; : FORK%=1TO15: VDU127: NEXT:
     VDU4
1030 ENDPROC
1040 DEFPROCredraw
1050 MOVE0,yorig%: VDU5: PRINT""; min_count%: MOVE0,725: PRINT"";
     max_count%: VDU4
1060 FORJ%=1 TO I%: PLOT69,J%*xstep+xorig%,(CO%(J%)-min%)*ystep+yorig%:
     NEXT
1070 ENDPROC
1080 DEFPROCsave
1090 $C%="D.RTEMP": ?(C%+19)=C% MOD 256: ?(C%+20)=C% DIV 256: X%=(C%+19)
     MOD 256: Y%=(C%+19) DIV 256: A%=6: CALL &FFDD
1100 X%=OPENOUT("D.RTEMP")
1110 PRINT#X%,I%: PRINT#X%,step(M%)*MO%: FOR J%=1 TO I%: PRINT#X%,CO%(J%):
     NEXT
1120 CLOSE#X%
1130 ENDPROC
1140 DEFPROCtime
1150 O%=(TIME DIV 100)MOD 60: P%=(TIME DIV 6000)MOD 60: Q%=(TIME DIV
     360000)MOD 24
1160 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(12,4);
```

```
1170 IF Q%<10 PRINT"0"; Q%;  ELSE PRINT; Q%;
1180 IF P%<10 PRINT":0"; P%;  ELSE PRINT":"; P%;
1190 IF O%<10 PRINT":0"; O%;  ELSE PRINT":"; O%;
1200 VDU28,0,30,79,7: COLOUR1: COLOUR128
1210 ENDPROC


  10 REM ------------------------------------------
  20 REM ---   FIND ROUTINE - SCANS AXIS AND COUNTS UNTIL PEAK FOUND  ---
  30 REM ------------------------------------------
  40 DIM AX%(8)
  50 AX%(1)=34: AX%(2)=35: AX%(3)=36: AX%(4)=37
  60 AX%(5)=33: AX%(6)=32: AX%(7)=33: AX%(8)=32
  70 scaler1%=56: scaler2%=55: speed%=50
  80 DIM step(2): step(1)=0.1776: step(2)=0.121
  90 PROCtitle(12,5,"FIND PEAK")
 100 CLS
 110 INPUT''" Which axis to scan (1 or 2) "M%
 120 IF M%<>1 AND M%<>2 THEN 100
 130 F%=(E%ANDM%)
 140 IF F% increasing=TRUE ELSE increasing=FALSE
 150 PRINT'" Previous motion was ";
 160 IF increasing PRINT"increasing"; ELSE PRINT "decreasing";
 170 PRINT" theta"
 180 INPUT'" Counting time, in 100th's secs "TT%
 190 INPUT'" Step length per cycle (secs) "step_length
 200 MO%=INT(step_length/step(M%))
 210 INPUT'" Counts required to assume peak found "LT%
 220 INPUT'" Increasing or decreasing theta (I/D) or both (B) "inc$
 230 both=FALSE: rewind=FALSE
 240 IF inc$="B" both=TRUE: INPUT'" Which direction first (I/D) "inc$
 250 INPUT'" Maximum scan range (degrees) "max_scan
 260 max_scan=max_scan*3600/step(M%)
 270 PROCstep_scan
 280 IF end_scan THEN 310
 290 IF both AND inc$="I" inc$="D": rewind=TRUE: PROCstep_scan: GOTO 310
 300 IF both AND inc$="D" inc$="I": rewind=TRUE: PROCstep_scan
 310 PROCcommand("Press <CR> for repeat, any other key for menu")
 320 A=GET: IF A=13 PROCcommand(" "): GOTO 90
 330 CHAIN"P.MENU"
 340 :
 350 DEFFNmotor(dir$)
 360 IFdir$="I" E%=(E% OR M%) ELSE E%=(E% AND(&FF EOR M%))
 370 IF M%=2 THEN IF dir$="I" F%=1 ELSE F%=0
 380 IF M%=1 THEN stepper%=AX%(M%+1-F%)
 390 IF M%=2 THEN stepper%=AX%(M%+1+F%)
 400 I%=0: IF dir$="I" P%=MO% ELSE P%=-MO%
 410 =stepper%
 420 :
 430 DEFPROCstep_scan
 440 stepper%=FNmotor(inc$)
 450 IFrewind THEN A%=FNminicam("ST1",stepper%,MO%*JJ%,speed%):
     PROCupdate(M%,P%*JJ%)
 460 X%=5: Y%=20
 470 PROCcommand("Press any key to stop")
 480 I%=0
 490 REPEAT
 500   I%=I%+1
 510   CO%=FNstep_count(stepper%,MO%,TT%)
 520   PRINTTAB(X%,Y%); CO%;
 530    X%=X%+15: IFX%=80 X%=5: Y%=Y%+1
 540    IF (Y%=23 AND X%=5) VDU10,10: Y%=22
 550    Z%=INKEY(0)
 560    UNTIL CO%>=LT% OR Z%<>-1 OR I%*MO%>=max_scan
 570 IF CO%>=LT% THEN CO%=FNminicam("CO1",scaler1%,TT%,0) +
     256*256*FNminicam("RD2",scaler2%,0,0): IF CO%<LT% THEN 490
 580 IF I%*MO%>=max_scan THEN end_scan=FALSE ELSE end_scan=TRUE
 590 JJ%=I%
 610 ENDPROC
 620 :
 630 DEFPROCtitle(X%,Y%,string$)
 640 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "                  ";
 650 PRINTTAB(X%,Y%); string$;
 660 VDU28,0,30,79,7
 670 COLOUR1: COLOUR128
 680 ENDPROC
 690 :
 700 DEFPROCcommand(string$)
 710 VDU28,0,31,79,31: COLOUR0: COLOUR129: CLS
 720 PRINTTAB(INT(40-LEN(string$)/2)); string$;
 730 VDU28,0,30,79,7: COLOUR1: COLOUR128
 740 ENDPROC
 750 :
 760 DEFFNstep_count(stepper%,MO%,TT%)
 770 A%=FNminicam("ST1",stepper%,MO%,speed%)
 780 PROCupdate(M%,P%)
 790 R%=FNminicam("CO1",scaler1%,TT%,0)+256*256*FNminicam("RD2",
     scaler2%,0,0)
 800 R$=STR$(R%): PROCtitle(55,1,R$)
 810 =R%
 820 :
 830 DEFFNminicam(usr$,AD%,NN%,TT%)
 840 *FX3,7
 850 PRINTusr$; ","; AD%; ","; NN%; ","; TT%; CHR$(13);
 860 *FX3,6
 870 *FX2,1
 880 INPUT""reply$
 890 *FX2,2
 900 *FX3,0
 910 =VAL(reply$)
 920 :
 930 DEFPROCupdate(M%,P%)
 940 @%=&2050A
 950 IF M%=1 S%=S%+P%: R=S%*step(1)/3600: PROCprint(34,1,R)
 960 IF M%=2 T%=T%+P%: R=T%*step(2)/3600: PROCprint(34,2,R)
 970 @%=10
 980 ENDPROC
 990 :
1000 DEFPROCprint(X%,Y%,R)
1010 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "                  ";
1020 PRINTTAB(X%,Y%); R;
1030 VDU28,0,30,79,7: COLOUR1: COLOUR128
1040 ENDPROC
```

```
  10 REM -----------------------------------
  20 REM -----    GONIOMETER TILT ROUTINE  -----
  30 REM -----------------------------------
  40 DIM AX%(8)
  50 AX%(1)=34: AX%(2)=35: AX%(3)=36: AX%(4)=37
```

```
 60 AX%(5)=33: AX%(6)=32: AX%(7)=33: AX%(8)=32
 70 scaler1%=56: scaler2%=55: speed%=100
 80 DIM step(2): step(1)=1.0/(96.0*50): step(2)=1.0/(96.0*50)
 90 PROCtitle(12,5,"TILT")
100 CLS
110 INPUT''" Which goniometer to tilt (1 or 2) "M%
120 IF M%<>1 AND M%<>2 THEN 100
130 F%=(G%ANDM%)
140 IF F% increasing=TRUE ELSE increasing=FALSE
150 PRINT'" Previous motion was ";
160 IF increasing PRINT"compressing"; ELSE PRINT "expanding";
170 PRINT" spring"
180 INPUT'" Distance to drive micrometer (mm) "step_length
190 MO%=step_length/step(M%)
200 INPUT'" Compressing or expanding spring (C/E) "inc$
210 PROCstep_scan
220 VDU7: PROCcommand("Press <CR> for repeat, any other key for menu")
230 A=GET: IF A=13 PROCcommand(" "): GOTO 100
240 CHAIN"P.MENU"
250 :
260 DEFFNmotor(dir$)
270 IFdir$="C" G%=(G% OR M%) ELSE G%=(G% AND(&FF EOR M%))
280 IF dir$="C" F%=1 ELSE F%=0
290 IF M%=1 THEN stepper%=AX%(M%+4+F%)
300 IF M%=2 THEN stepper%=AX%(M%+5+F%)
310 I%=0: IF dir$="C" P%=MO% ELSE P%=-MO%
320 =stepper%
330 :
340 DEFPROCstep_scan
350 stepper%=FNmotor(inc$)
360 IF (V%+P%)<0 OR (V%+P%)>(25/step(M%)) PRINT'" Limit will be passed
    by this scan - driving to limit"
370 IF (V%+P%)<0 MO%=V%
380 IF (V%+P%)>(25/step(M%)) MO%=25/step(M%)-V%
390 stepper%=FNmotor(inc$)
400 PRINT''" OK - tilting goniometer "; M%; ", "; MO%; " steps"
410 IF MO%>&FFFF A%=FNminicam("ST1",stepper%,&7FFF,speed%): MO%=MO%-
    &7FFF: GOTO410
420 A=FNminicam("ST1",stepper%,MO%,speed%)
430 @%=&2050A
440 IF M%=1 U%=U%+P%: R=U%*step(1): PROCprint(34,3,R)
450 IF M%=2 V%=V%+P%: R=V%*step(2): PROCprint(34,4,R)
460 @%=10
470 ENDPROC
480 :
490 DEFPROCtitle(X%,Y%,string$)
500 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "          ";
510 PRINTTAB(X%,Y%); string$;
520 VDU28,0,30,79,7
530 COLOUR1: COLOUR128
540 ENDPROC
550 :
560 DEFPROCcommand(string$)
570 VDU28,0,31,79,31: COLOUR0: COLOUR129: CLS
580 PRINTTAB(INT(40-LEN(string$)/2)); string$;
590 VDU28,0,30,79,7: COLOUR1: COLOUR128
600 ENDPROC
610 :
620 DEFFNminicam(usr$,AD%,NN%,TT%)
630 PROCtime
```

```
640 *FX2,2
650 *FX3,7
660 PRINTusr$; ","; AD%; ","; NN%; ","; TT%; CHR$(13);
670 *FX3,6
680 *FX2,1
690 INPUT""reply$
700 *FX2,2
710 *FX3,0
720 =VAL(reply$)
730 :
740 DEFPROCprint(X%,Y%,R)
750 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "          ";
760 PRINTTAB(X%,Y%); R;
770 VDU28,0,30,79,7
780 COLOUR1: COLOUR128
790 ENDPROC
800 :
810 DEFPROCtime
820 se%=(TIME DIV 100)MOD 60: mi%=(TIME DIV 6000)MOD 60: ho%=(TIME DIV
    360000)MOD 24
830 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(12,4);
840 IF ho%<10 PRINT"0"; ho%;  ELSE PRINT; ho%;
850 IF mi%<10 PRINT":0"; mi%;  ELSE PRINT":"; mi%;
860 IF se%<10 PRINT":0"; se%;  ELSE PRINT":"; se%;
870 VDU28,0,30,79,7: COLOUR1: COLOUR128
880 ENDPROC
```

```
 10 REM ----------------------------------------------------
 20 REM -----  REPLOT ROUTINE - PLOTS DATA FROM DISC FILE  -----
 30 REM ----------------------------------------------------
 40 DIM PS(4),MS(4),map(4)
 50 PROCtitle(12,5,"REPLOT")
 60 max_count%=0: min_count%=&7FFFFFFF
 70 CLS
 80 PRINT'" When prompted you can enter the name of a data file":
    PRINT" or press RETURN to plot the last recorded curve"
 90 INPUT'" Enter the data file to be plotted "file$
100 IF file$="" file$=":0.D.RTEMP"
110 IF INSTR(file$,".")=0 file$=":2.D."+file$
120 DER%=0
130 ON ERROR GOTO 380
140 X%=OPENIN(file$)
150 IF DER%<>0 AND ERR=214 INPUT'" File does not exist - choose another"
    file$: DER%=0: GOTO100
160 IF DER%<>0 AND ERR=204 INPUT'" File name too long - choose another"
    file$: DER%=0: GOTO 100
170 INPUT#X%,NU%
180 IF DER%<>0 AND ERR=222 INPUT'" File does not exist - choose another"
    file$: CLOSE#X%: DER%=0: GOTO100
190 INPUT#X%,step
200 FOR J%=1 TO NU%
210   INPUT#X%,A%
220   IF A%>max_count% THEN max_count%=A%
230   IF A%<min_count% THEN min_count%=A%
240   NEXT
250 CLOSE#X%
260 IF max_count%=0 THEN max_count%=100
270 CLS: PRINT'" X-range of data is 0 to "; step*NU%; " secs"
280 INPUT'" Enter X-range to be plotted "xmin,xmax
```

```
290 PRINT'"  Y-range of data is "; min_count%; " to "; max_count%; "
    counts"
300 INPUT'"  Enter range to be plotted "min_count%,max_count%
310 CLS
320 PROCplot
330 PROCcommand("Press <CR> for repeat, any other key for menu")
340 *FX21,0
350 REPEAT: A=INKEY(50): PROCtime: UNTIL A<>-1
360 IF A=13 PROCcommand(" "): GOTO 60
370 CHAIN"P.MENU"
380 DER%=DER%+1
390 IF ERR=204 OR ERR=199 THEN 150
400 IF ERR=222 THEN 180 ELSE REPORT: END
410 DEFPROCplot
420 PROCpaper
430 PROCpspace(0,1,0.05,1): PROCmspace(0,1,0,1)
440 A$="EXPERIMENTAL ROCKING CURVE ": PROCplotcs(0.2,0.96,A$)
450 PROCplotcs(0.6,0.96,file$)
460 A$="RELATIVE BRAGG ANGLE (secs)": PROCplotcs(0.5,0.03,A$)
470 PROCpspace(0.1,0.9,0.15,0.92):
    PROCmspace(xmin,xmax,min_count%,max_count%)
480 PROCaxes
490 PROCmove(map(1),map(3))
500 X%=OPENIN(file$): INPUT#X%,NU%: INPUT#X%,step
510 FOR J%=1 TO NU%
520   INPUT#X%,A%
530   IF J%*step<xmin THEN 580
540   IF J%*step>xmax THEN 600
550   IF A%>max_count% THEN 580
560   IF A%<min_count% THEN 580
570   PROCpoint(J%*step,A%)
580   NEXT
590 CLOSE#X%
600 ENDPROC
610 DEFPROCpaper
620 xscale%=1279: yscale%=800
630 VDU24,0; 0; 1279; 800;
640 CLG
650 ENDPROC
660 DEFPROCpspace(PS(1),PS(2),PS(3),PS(4))
670 ZX%=xscale%*PS(1): ZY%=yscale%*PS(3)
680 VX%=xscale%*PS(2)-ZX%: VY%=yscale%*PS(4)-ZY%
690 ENDPROC
700 DEFPROCmspace(MS(1),MS(2),MS(3),MS(4))
710 map(1)=MS(1): map(2)=MS(2): map(3)=MS(3): map(4)=MS(4)
720 IF map(1)<0 AND map(2)>0 THEN xorig=0 ELSE xorig=map(1)
730 IF map(3)<0 AND map(4)>0 THEN yorig=0 ELSE yorig=map(3)
740 ENDPROC
750 DEFPROCmove(x,y)
760 A%=(x-map(1))*VX%/(map(2)-map(1))+ZX%
770 B%=(y-map(3))*VY%/(map(4)-map(3))+ZY%
780 PLOT4,A%,B%
790 ENDPROC
800 DEFPROCpoint(x,y)
810 PROCmove(x,y)
820 PLOT65,0,0
830 ENDPROC
840 DEFPROCaxes
850 xrange=map(2)-map(1): yrange=map(4)-map(3)
860 PROCmark(xrange)
```

```
 870 PROCmove(map(1),yorig): PROCnotch(1)
 880 step%=xrange/SX
 890 FOR J%=1 TO step%
 900   PROCdraw_rel(SX,0)
 910   PROCnotch(1)
 920   NEXT
 930 PROCdraw(map(2),yorig)
 940 FOR J%=step% TO 0 STEP -1
 950   x=J%*SX+map(1)
 960   PROClabel_x(x)
 970   NEXT
 980 PROCmark(yrange)
 990 PROCmove(xorig,map(3)): PROCnotch(2)
1000 step%=yrange/SX
1010 FOR J%=1 TO step%
1020   PROCdraw_rel(0,SX)
1030   PROCnotch(2)
1040   NEXT
1050 PROCdraw(xorig,map(4))
1060 FOR J%=step% TO 0 STEP-1
1070   x=J%*SX+map(3)
1080   PROClabel_y(x)
1090   NEXT
1100 ENDPROC
1110 DEFPROCdraw_rel(x,y)
1120 A%=x*VX%/(map(2)-map(1))
1130 B%=y*VY%/(map(4)-map(3))
1140 PLOT1,A%,B%
1150 DEFPROCnotch(C%)
1160 IF C%=1 THEN PLOT1,0,10: PLOT1,0,-10
1170 IFC%=2 THEN PLOT1,10,0: PLOT1,-10,0
1180 ENDPROC
1190 DEFPROClabel_x(x)
1200 A%=(x-map(1))*VX%/(map(2)-map(1))+ZX%-8*LEN(STR$(x))
1210 B%=(yorig-map(3))*VY%/(map(4)-map(3))+ZY%-22
1220 PLOT4,A%,B%: VDU5: PRINTSTR$(x): VDU4
1230 ENDPROC
1240 DEFPROClabel_y(x)
1250 A%=(xorig-map(1))*VX%/(map(2)-map(1))+ZX%-16*LEN(STR$(x))-20
1260 B%=(x-map(3))*VY%/(map(4)-map(3))+ZY%
1270 PLOT4,A%,B%: VDU5: PRINTSTR$(x): VDU4
1280 ENDPROC
1290 DEFPROCmark(range)
1300 B%=LOG(range)
1310 A=range/10.^B%
1320 IF A>=1 AND A<3 THEN SX=2*10^(B%-1)
1330 IF A>=3 AND A<7 THEN SX=5*10^(B%-1)
1340 IF A>=7 AND A<10 THEN SX=1*10^B%
1350 ENDPROC
1360 DEFPROCplotcs(x,y,A$)
1370 PROCmove(x,y): VDU5: PRINTA$: VDU4: ENDPROC
1380 DEFPROCtitle(X%,Y%,string$)
1390 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "                        ";
1400 PRINTTAB(X%,Y%); string$;
1410 VDU28,0,30,79,7: COLOUR1: COLOUR128
1420 ENDPROC
1430 DEFPROCcommand(string$)
1440 VDU28,0,31,79,31: COLOUR0: COLOUR129: CLS
1450 PRINTTAB(40-INT(LEN(string$)/2)); string$;
1460 VDU28,0,30,79,7: COLOUR1: COLOUR128
```

```
1470 ENDPROC
1480 DEFPROCdraw(x,y)
1490 A%=(x-map(1))*VX%/(map(2)-map(1))+ZX%: B%=(y-map(3))*VY%/(map(4)-
     map(3))+ZY%
1500 PLOT5,A%,B%: ENDPROC
1510 :
1520 DEFPROCtime
1530 se%=(TIME DIV 100)MOD 60: mi%=(TIME DIV 6000)MOD 60: ho%=(TIME DIV
     360000)MOD 24
1540 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(12,4);
1550 IF ho%<10 PRINT"0"; ho%;  ELSE PRINT; ho%;
1560 IF mi%<10 PRINT":0"; mi%;  ELSE PRINT":"; mi%;
1570 IF se%<10 PRINT":0"; se%;  ELSE PRINT":"; se%;
1580 VDU28,0,30,79,7: COLOUR1: COLOUR128
1590 ENDPROC
```

```
  10 REM ---------------------------------------------------
  20 REM -----  HARDCOPY PLOT ROUTINE- PLOTS DATA FROM DISC  -----
  30 REM ---------------------------------------------------
  40 DIM PS(4),MS(4),map(4)
  50 PROCtitle(12,5,"HARDCOPY")
  60 max_count%=0: min_count%=&7FFFFFFF
  70 CLS
  80 PRINT''" When prompted you can enter the name of a data file":
     PRINT" or press RETURN to plot the last recorded curve"
  90 INPUT'" Enter the data file to be plotted "file$
 100 IF file$="" file$=":0.D.RTEMP"
 110 IF INSTR(file$,".")=0 file$=":2.D."+file$
 120 DER%=0
 130 ON ERROR GOTO 370
 140 X%=OPENIN(file$)
 150 IF DER%<>0 AND ERR=214 INPUT'" File does not exist - choose another"
     file$: DER%=0: GOTO100
 160 IF DER%<>0 AND ERR=204 INPUT'" File name too long - choose another"
     file$: DER%=0: GOTO 100
 170 INPUT#X%,NU%
 180 IF DER%<>0 AND ERR=222 INPUT'" File does not exist - choose another"
     file$: CLOSE#X%: DER%=0: GOTO100
 190 INPUT#X%,step
 200 FOR J%=1 TO NU%
 210   INPUT#X%,A%
 220   IF A%>max_count% THEN max_count%=A%
 230   IF A%<min_count% THEN min_count%=A%
 240   NEXT
 250 CLOSE#X%
 260 IF max_count%=0 THEN max_count%=100
 270 CLS: PRINT'" X-range of data is 0 to "; step*NU%; " secs"
 280 INPUT'" Enter X-range to be plotted "xmin,xmax
 290 PRINT'" Y-range of data is "; min_count%; " to "; max_count%; "
     counts"
 300 INPUT'" Enter range to be plotted "min_count%,max_count%
 310 PRINT''"          "; : COLOUR0: COLOUR129: PRINT"Plotting data from
     disc file "; file$: COLOUR1: COLOUR128
 320 PROCplot
 330 PROCcommand("Press <CR> for repeat, any other key for menu")
 340 *FX21,0
 350 A=GET: IF A=13 PROCcommand(" "): GOTO 60
 360 CHAIN"P.MENU"
 370 DER%=DER%+1
```

```
380 IF ERR=204 OR ERR=199 THEN 150
390 IF ERR=222 THEN 180 ELSE REPORT: END
400 DEFPROCplot
410 PROCpaper: PROCpen(1)
420 PROCpspace(0,1,0,1): PROCmspace(0,1,0,1)
430 PROCctrmag(6): PROCplotcs(0.1,0.92,"EXPERIMENTAL ROCKING CURVE ")
440 PROCplotcs(0.7,0.92,file$)
450 PROCctrmag(4): PROCplotcs(0.5,0.03,"RELATIVE BRAGG ANGLE (secs)")
460 PROCctrori(1): PROCplotcs(0.025,0.3,"COUNTS (unit time)"):
    PROCctrori(0)
470 PROCpspace(0.1,0.9,0.1,0.9):
    PROCmspace(xmin,xmax,min_count%,max_count%)
480 PROCaxes
490 PROCmove(map(1),map(3))
500 X%=OPENIN(file$): INPUT#X%,NU%: INPUT#X%,step
510 FOR J%=1 TO NU%
520   INPUT#X%,CO%
530   IF J%*step<xmin THEN 580
540   IF J%*step>xmax THEN 590
550   IF CO%>max_count% THEN 580
560   IF CO%<min_count% THEN 580
570   PROCdraw((J%-1)*step,CO%): PROCdraw(J%*step,CO%)
580   NEXT
590 PROCpen(0)
600 ENDPROC
610 DEFPROCpaper
620 xscale%=2400: yscale%=1800
630 PROCminicam("PR2,"+CHR$(10))
640 ENDPROC
650 DEFPROCpspace(PS(1),PS(2),PS(3),PS(4))
660 ZX%=xscale%*PS(1): ZY%=yscale%*PS(3)
670 VX%=xscale%*PS(2)-ZX%: VY%=yscale%*PS(4)-ZY%
680 ENDPROC
690 DEFPROCmspace(MS(1),MS(2),MS(3),MS(4))
700 map(1)=MS(1): map(2)=MS(2): map(3)=MS(3): map(4)=MS(4)
710 IF map(1)<0 AND map(2)>0 THEN xorig=0 ELSE xorig=map(1)
720 IF map(3)<0 AND map(4)>0 THEN yorig=0 ELSE yorig=map(3)
730 ENDPROC
740 DEFPROCmove(x,y)
750 A%=(x-map(1))*VX%/(map(2)-map(1))+ZX%
760 B%=(y-map(3))*VY%/(map(4)-map(3))+ZY%
770 PROCminicam("PR1,"+"M"+STR$(A%)+","+STR$(B%))
780 ENDPROC
790 DEFPROCaxes
800 xrange=map(2)-map(1): yrange=map(4)-map(3)
810 PROCmark(xrange)
820 PROCmove(map(1),yorig): PROCnotch(1)
830 step%=xrange/SX
840 FOR J%=1 TO step%
850   PROCdraw_rel(SX,0)
860   PROCnotch(1)
870   NEXT
880 PROCdraw(map(2),yorig)
890 FOR J%=step% TO 0 STEP -1
900   x=J%*SX+map(1)
910   PROClabel_x(x)
920   NEXT
930 PROCmark(yrange)
940 PROCmove(xorig,map(3)): PROCnotch(2)
950 step%=yrange/SX
```

```
 960 FOR J%=1 TO step%
 970   PROCdraw_rel(0,SX)
 980   PROCnotch(2)
 990   NEXT
1000 PROCdraw(xorig,map(4))
1010 FOR J%=step% TO 0 STEP-1
1020   x=J%*SX+map(3)
1030   PROClabel_y(x)
1040   NEXT
1050 ENDPROC
1060 DEFPROCdraw_rel(x,y)
1070 A%=x*VX%/(map(2)-map(1))
1080 B%=y*VY%/(map(4)-map(3))
1090 PROCminicam("PR1,"+"I"+STR$(A%)+","+STR$(B%))
1100 DEFPROCnotch(C%)
1110 IF C%=1 THEN PROCminicam("PR1,I0,25,0,-25")
1120 IFC%=2 THEN PROCminicam("PR1,I25,0,-25,0")
1130 ENDPROC
1140 DEFPROClabel_x(x)
1150 A%=(x-map(1))*VX%/(map(2)-map(1))+ZX%-8*LEN(STR$(x))
1160 B%=(yorig-map(3))*VY%/(map(4)-map(3))+ZY%-65
1170 PROCminicam("PR1,"+"M"+STR$(A%)+","+STR$(B%))
1180 PROCminicam("PR1,"+"P"+STR$(x))
1190 ENDPROC
1200 DEFPROClabel_y(x)
1210 A%=(xorig-map(1))*VX%/(map(2)-map(1))+ZX%-16*LEN(STR$(x))-100
1220 B%=(x-map(3))*VY%/(map(4)-map(3))+ZY%
1230 PROCminicam("PR1,"+"M"+STR$(A%)+","+STR$(B%))
1240 PROCminicam("PR1,"+"P"+STR$(x))
1250 ENDPROC
1260 DEFPROCmark(range)
1270 B%=LOG(range)
1280 A=range/10.^B%
1290 IF A>=1 AND A<3 THEN SX=2*10^(B%-1)
1300 IF A>=3 AND A<7 THEN SX=5*10^(B%-1)
1310 IF A>=7 AND A<10 THEN SX=1*10^B%
1320 ENDPROC
1330 DEFPROCplotcs(x,y,C$)
1340 PROCmove(x,y): C$="PR1,"+"P"+C$: PROCminicam(C$)
1350 DEFPROCtitle(X%,Y%,string$)
1360 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "                   ";
1370 PRINTTAB(X%,Y%); string$;
1380 VDU28,0,30,79,7: COLOUR1: COLOUR128
1390 ENDPROC
1400 DEFPROCcommand(string$)
1410 VDU28,0,31,79,31: COLOUR0: COLOUR129: CLS
1420 PRINTTAB(40-INT(LEN(string$)/2)); string$;
1430 VDU28,0,30,79,7: COLOUR1: COLOUR128
1440 ENDPROC
1450 DEFPROCdraw(x,y)
1460 A%=(x-map(1))*VX%/(map(2)-map(1))+ZX%: B%=(y-map(3))*VY%/(map(4)-
     map(3))+ZY%
1470 A$="PR1,"+"D"+STR$(A%)+","+STR$(B%): PROCminicam(A$)
1480 ENDPROC
1490 DEFPROCpen(C%)
1500 A$="PR1,"+"J"+STR$(C%): PROCminicam(A$)
1510 ENDPROC
1520 DEFPROCctrmag(C%)
1530 ctrsize%=C%+1: A$="PR1,"+"S"+STR$(C%): PROCminicam(A$)
1540 ENDPROC
```

```
1550 DEFPROCctrori(C%)
1560 A$="PR1,"+"Q"+STR$(C%): PROCminicam(A$): ENDPROC
1570 DEFPROCminicam(A$)
1580 *FX3,7
1590 *FX2,2
1600 PRINTA$; CHR$(13);
1610 *FX3,6
1620 *FX2,1
1630 INPUT""A$
1640 *FX2,2
1650 *FX3,0
1660 ENDPROC


  10 REM ----------------------------------------
  20 REM -----   AUTOMATIC TILT OPTIMISE ROUTINE  -----
  30 REM ----------------------------------------
  40 DIM AX%(8)
  50 AX%(1)=34: AX%(2)=35: AX%(3)=36: AX%(4)=37
  60 AX%(5)=33: AX%(6)=32: AX%(7)=33: AX%(8)=32
  70 scaler1%=56: scaler2%=55: speed%=20: gspeed%=100
  80 DIM step(4),max%(9),position%(9)
  90 step(1)=0.1776: step(2)=0.121: step(3)=1.0/(96.0*50):
     step(4)=1.0/(96.0*50)
 100 here%=T%: tilt_here%=V%
 110 PROCtitle(12,5,"AUTO TILT")
 120 CLS
 130 PRINT''"  Second axis only is driven by this routine"
 140 M%=2
 150 F%=(E%ANDM%)
 160 IF F% increasing=TRUE ELSE increasing=FALSE
 170 PRINT'"  Previous motion of this axis was ";
 180 IF increasing PRINT"increasing";  ELSE PRINT "decreasing";
 190 PRINT" theta"
 200 INPUT'"  Counting time for peak scanning, in 100th's secs "TT%
 210 INPUT'"  Step length for peak scanning (secs) "step_length
 220 MO%=INT(step_length/step(M%))
 230 INPUT'"  Maximum range to be scanned when looking for peaks (secs) "
     range
 240 limit%=range/step(M%)
 250 INPUT'"  Tilt step length for tilt optimisation "tilt_step
 260 INPUT'"  Enter the background count rate (counts/sec) "background
 270 tstep%=tilt_step/step(M%+2)
 280 FOR C%=1 TO 3
 290   PROCtilt_opt
 300   tstep%=tstep%/2
 310   NEXT C%
 320 PROCcommand("Press <CR> for repeat, any other key for menu")
 330 A=GET: IF A=13 PROCcommand(" "): GOTO 110
 340 CHAIN"P.MENU"
 350 :
 360 DEFFNmotor(dir$)
 370 IFdir$="I" E%=(E% OR M%) ELSE E%=(E% AND(&FF EOR M%))
 380 IF dir$="I" THEN F%=1 ELSE F%=0
 390 IF M%=1 THEN stepper%=AX%(M%+1-F%)
 400 IF M%=2 THEN stepper%=AX%(M%+1+F%)
 410 I%=0: IF dir$="I" P%=MO% ELSE P%=-MO%
 420 =stepper%
 430 :
 440 DEFPROCtitle(X%,Y%,string$)
 450 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "                    ";
 460 PRINTTAB(X%,Y%); string$;
 470 VDU28,0,30,79,7
 480 COLOUR1: COLOUR128
 490 ENDPROC
 500 :
 510 DEFPROCcommand(string$)
 520 VDU28,0,31,79,31: COLOUR0: COLOUR129: CLS
 530 PRINTTAB(INT(40-LEN(string$)/2)); string$;
 540 VDU28,0,30,79,7: COLOUR1: COLOUR128
 550 ENDPROC
 560 :
 570 DEFPROCstep_count(stepper%,MO%,TT%)
 580 A%=FNminicam("ST1",stepper%,MO%,speed%)
 590 @%=&2050A
 600 IF M%=1 S%=S%+P%: R=S%*step(1)/3600: PROCprint(34,1,R)
 610 IF M%=2 T%=T%+P%: R=T%*step(2)/3600: PROCprint(34,2,R)
 620 @%=10
 630 R%=FNminicam("CO1",scaler1%,TT%,0)+256*256*FNminicam("RD2",
     scaler2%,0,0)
 640 R$=STR$(R%): PROCtitle(68,1,R$)
 650 ENDPROC
 660 :
 670 DEFFNminicam(usr$,AD%,NN%,TT%)
 680 *FX2,2
 690 *FX3,7
 700 PRINTusr$; ","; AD%; ","; NN%; ","; TT%; CHR$(13);
 710 *FX3,6
 720 *FX2,1
 730 INPUT""reply$
 740 *FX2,2
 750 *FX3,0
 760 =VAL(reply$)
 770 :
 780 DEFPROCprint(X%,Y%,R)
 790 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "                    "
 800 PRINTTAB(X%,Y%); R;
 810 VDU28,0,30,79,7
 820 COLOUR1: COLOUR128
 830 ENDPROC
 840 :
 850 DEFPROCtilt_opt
 860 CLS: PRINTTAB(8); "Tilt (mm)"; TAB(18);. "Maximum count"; TAB(28);
     "Axis position of max"
 870 FOR I%=1 TO 5
 880   PROCfind
 890   PRINT'TAB(10); V%*step(4); TAB(20); maximum%; TAB(30);
       maxpos%*step(2)/3600
 900   max%(I%)=maximum%: position%(I%)=maxpos%
 910   IF I%<>5 PROCtilt("I",tstep%)
 920   NEXT
 930 PROCtilt("D",5*tstep%)
 940 IF (T%-here%)>0 PROCmove("D",T%-here%) ELSE PROCmove("I",here%-T%)
 950 FOR I%=6 TO 9
 960   PROCfind
 970   max%(I%)=maximum%: position%(I%)=maxpos%
 980   PRINT'TAB(10); V%*step(4); TAB(20); maximum%; TAB(30);
       maxpos%*step(2)/3600
 990   IF I%<>9 PROCtilt("D",tstep%)
1000   NEXT
```

```
1010 maximum%=0: FOR I%=1 TO 9: IF max%(I%)>maximum% maximum%=max%(I%):
     J%=position%(I%): Z%=I%: NEXT
1020 IF Z%<=5 PROCtilt("I",(Z%+3)*tstep%) ELSE PROCtilt("I",(9-Z%)*tstep%)
1030 IF (T%-J%)>0 PROCmove("D",T%-J%) ELSE PROCmove("I",J%-T%)
1040 ENDPROC
1050 :
1060 DEFPROCtilt(dir$,step%)
1070 IF dir$="I" stepper%=AX%(9): P%=step% ELSE stepper%=AX%(8): P%=-step%
1080 IF step%>&7FFF A%=FNminicam("ST1",stepper%,&7FFF,gspeed%):
     step%=step%-&7FFF:GOTO 1080
1090 A%=FNminicam("ST1",stepper%,step%,gspeed%)
1100 @%=&2050A: V%=V%+P%: R=V%*step(4): PROCprint(34,4,R)
1110 @%=10
1120 ENDPROC
1130 :
1140 DEFPROCmove(dir$,step%)
1150 stepper%=FNmotor(dir$)
1160 IF step%>&7FFF A%=FNminicam("ST1",stepper%,&7FFF,speed%):
     step%=step%-&7FFF:GOTO 1160
1170 A%=FNminicam("ST1",stepper%,step%,speed%)
1180 ENDPROC
1190 :
1200 DEFPROCfind
1210 now%=T%
1220 stepper%=FNmotor("I")
1230 step_count
1240 REPEAT
1250   PROCstep_count(stepper%,MO%,100)
1260   UNTIL (R%>(2*background)) OR (T%>=now%+limit%)
1270 IF R%>(2*background) THEN stepper%=FNmotor("D"):GOTO 1350
1280 PROCmove("D",T%-now%)
1290 stepper%=FNmotor("D")
1300 REPEAT
1310   PROCstep_count(stepper%,MO%,100)
1320   UNTIL (R%>(2*background)) OR (T%<=now%-limit%)
1330 IF R%>(2*background) THEN 1350
1340 fault=TRUE: ENDPROC
1350 REPEAT: PROCstep_count(stepper%,MO%,100): UNTIL R%<(1.5*background):
     maximum%=0: stepper%=FNmotor("I")
1360 REPEAT
1370   PROCstep_count(stepper%,MO%,100): IF R%>maximum% maximum%=R%:
       maxpos%=T%
1380   UNTIL R%<(1.5*background) AND maximum%>(1.5*background)
1390 ENDPROC
```

```
  10 REM --------------------------------
  20 REM -----   MOVE SCANNING TABLE   -----
  30 REM --------------------------------
  40 DIM AX%(12)
  50 AX%(1)=34: AX%(2)=35: AX%(3)=36: AX%(4)=37
  60 AX%(5)=33: AX%(6)=32: AX%(7)=33: AX%(8)=32
  70 AX%(9)=32: AX%(10)=33: AX%(11)=34: AX%(12)=35
  80 speed%=50
  90 DIM step(2)
 100 step(1)=1.0/96.0: step(2)=1.0/96.0
 110 PROCtitle(12,5,"SCAN TABLE")
 120 CLS
 130 PRINT''"  The present position of the scanning table is"
 140 PRINT''"              X = "; K%*step(1); " mm"
```

```
150 PRINT"               Y = "; L%*step(2); " mm"
160 INPUT''"  Enter the X coord and Y coord of desired position (mm) "
    new_x,new_y
170 PRINT''"  OK - moving to desired position"
180 PROCmove
190 PROCcommand("Press <CR> for repeat, any other key for menu")
200 A=GET: IF A=13 PROCcommand(" "): GOTO 120
210 CHAIN"P.MENU"
220 :
230 DEFPROCmove
240 steps%=(new_x/step(1)-K%)
250 IF steps%>=0 THEN A%=FNminicam("ST1",AX%(9),steps%,speed%) ELSE
    A%=FNminicam("ST1",AX%(10),-steps%,speed%)
260 PROCupdate(1,steps%)
270 steps%=(new_y/step(2)-L%)
280 IF steps%>=0 THEN A%=FNminicam("ST1",AX%(11),steps%,speed%) ELSE
    A%=FNminicam("ST1",AX%(12),-steps%,speed%)
290 PROCupdate(2,steps%)
300 ENDPROC
310 :
320 DEFPROCupdate(motor%,steps%)
330 @%=&2050A
340 IF motor%=1 K%=K%+steps%: R=K%*step(1): PROCprint(47,2,R)
350 IF motor%=2 L%=L%+steps%: R=L%*step(2): PROCprint(47,3,R)
360 @%=10
370 ENDPROC
380 :
390 DEFPROCtitle(X%,Y%,string$)
400 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "          ";
410 PRINTTAB(X%,Y%); string$;
420 VDU28,0,30,79,7
430 COLOUR1: COLOUR128
440 ENDPROC
450 :
460 DEFPROCcommand(string$)
470 VDU28,0,31,79,31: COLOUR0: COLOUR129: CLS
480 PRINTTAB(INT(40-LEN(string$)/2)); string$;
490 VDU28,0,30,79,7: COLOUR1: COLOUR128
500 ENDPROC
510 :
520 DEFFNminicam(usr$,AD%,NN%,TT%)
530 PROCtime
540 *FX2,2
550 *FX3,7
560 PRINTusr$; ","; AD%; ","; NN%; ","; TT%; CHR$(13);
570 *FX3,6
580 *FX2,1
590 INPUT""reply$
600 *FX2,2
610 *FX3,0
620 =VAL(reply$)
630 :
640 DEFPROCprint(X%,Y%,R)
650 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "          ";
660 PRINTTAB(X%,Y%); R;
670 VDU28,0,30,79,7: COLOUR1: COLOUR128
680 ENDPROC
690 :
700 DEFPROCtime
710 se%=(TIME DIV 100)MOD 60: mi%=(TIME DIV 6000)MOD 60: ho%=(TIME DIV
    360000)MOD 24
720 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(12,4);
730 IF ho%<10 PRINT"0"; ho%;  ELSE PRINT; ho%;
740 IF mi%<10 PRINT":0"; mi%;  ELSE PRINT":"; mi%;
750 IF se%<10 PRINT":0"; se%;  ELSE PRINT":"; se%;
760 VDU28,0,30,79,7: COLOUR1: COLOUR128
770 ENDPROC


 10 REM -------------------------------------------------
 20 REM -----    PEAK HOLD ROUTINE - FOR TOPOGRAPHS  -----
 30 REM -------------------------------------------------
 40 DIM AX%(8)
 50 AX%(1)=34: AX%(2)=35: AX%(3)=36: AX%(4)=37
 60 AX%(5)=33: AX%(6)=32: AX%(7)=33: AX%(8)=32
 70 scaler1%=56: scaler2%=55: speed%=20
 80 DIM step(2)
 90 step(1)=0.1776: step(2)=0.121
100 decr$="": incr$="": fault=FALSE: end=FALSE
110 PROCtitle(12,5,"PEAK HOLD")
120 CLS
130 PRINT''"  Only the second axis is driven by this routine"
140 M%=2
150 F%=(E%ANDM%)
160 IF F% increasing=TRUE ELSE increasing=FALSE
170 PRINT'"  The last motion of this axis was ";
180 IF increasing PRINT"increasing";  ELSE PRINT "decreasing";
190 PRINT" theta"
200 PRINT'"  Are you positioned at"
210 PRINT'"            1. Top of peak"
220 PRINT "            2. High angle side"
230 PRINT "            3. Low angle side"
240 INPUT'"  Enter which one (1,2 or 3) "position%
250 INPUT'"  Step size required for adjusments (secs) "step_length
260 MO%=INT(step_length/step(M%))
270 INPUT'"  Enter the background count rate (counts/sec) "background
280 CLS: PRINT'''"     Holding peak position": PROCcommand("Press any key
    to stop")
290 IF position%=1 PROCtop ELSE PROCflank
300 IF fault=TRUE CLS: PRINT'''"   Count rate is less than background -
    check generator"
310 PROCcommand("Press <CR> for repeat, any other key for menu")
320 A=GET: IF A=13 PROCcommand(" "): GOTO 110
330 CHAIN"P.MENU"
340 :
350 DEFFNmotor(dir$)
360 IFdir$="I" E%=(E% OR M%) ELSE E%=(E% AND(&FF EOR M%))
370 IF dir$="I" THEN F%=1 ELSE F%=0
380 IF M%=1 THEN stepper%=AX%(M%+1-F%)
390 IF M%=2 THEN stepper%=AX%(M%+1+F%)
400 I%=0: IF dir$="I" P%=MO% ELSE P%=-MO%
410 =stepper%
420 :
430 :
440 DEFPROCtitle(X%,Y%,string$)
450 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "          ";
460 PRINTTAB(X%,Y%); string$;
470 VDU28,0,30,79,7
480 COLOUR1: COLOUR128
490 ENDPROC
```

```
 500 :
 510 DEFPROCcommand(string$)
 520 VDU28,0,31,79,31: COLOUR0: COLOUR129: CLS
 530 PRINTTAB(INT(40-LEN(string$)/2)); string$;
 540 VDU28,0,30,79,7: COLOUR1: COLOUR128
 550 ENDPROC
 560 :
 570 DEFFNminicam(usr$,AD%,NN%,TT%)
 580 *FX2,2
 590 *FX3,7
 600 PRINTusr$; ","; AD%; ","; NN%; ","; TT%; CHR$(13);
 610 *FX3,6
 620 *FX2,1
 630 INPUT""reply$
 640 *FX2,2
 650 *FX3,0
 660 =VAL(reply$)
 670 :
 680 DEFPROCprint(X%,Y%,R)
 690 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "          "
 700 PRINTTAB(X%,Y%); R;
 710 VDU28,0,30,79,7
 720 COLOUR1: COLOUR128
 730 ENDPROC
 740 :
 750 DEFPROCflank
 760 rate=FNminicam("CO1",scalar1%,500,0)+256*256*FNminicam("RD2",
     scalar2%,0,0)
 770 error=SQR(rate)*3/5: rate=rate/5
 780 if position%=2 decr$="I": incr$="D" ELSE decr$="D": incr$="I"
 790 IF end=TRUE ENDPROC ELSE count=FNintensity
 800 IF count<(background-SQR(background)) fault=TRUE: ENDPROC
 810 IF (count<(rate+error)) AND (count>(rate-error)) PROCdelay(500): GOTO
     790
 820 IF count<(rate-error) PROCincrease: GOTO790
 830 IF count>(rate+error) PROCdecrease: GOTO790
 840 ENDPROC
 850 :
 860 DEFPROCincrease
 870 stepper%=FNmotor(incr$)
 880 A%=FNminicam("ST1",stepper%,MO%,speed%)
 890 @%=&2050A
 900 IF M%=1 S%=S%+P%: R=S%*step(1)/3600: PROCprint(34,1,R)
 910 IF M%=2 T%=T%+P%: R=T%*step(2)/3600: PROCprint(34,2,R)
 920 @%=10
 930 ENDPROC
 940 :
 950 DEFPROCdecrease
 960 stepper%=FNmotor(decr$)
 970 A%=FNminicam("ST1",stepper%,MO%,speed%)
 980 @%=&2050A
 990 IF M%=1 S%=S%+P%: R=S%*step(1)/3600: PROCprint(34,1,R)
1000 IF M%=2 T%=T%+P%: R=T%*step(2)/3600: PROCprint(34,2,R)
1010 @%=10
1020 ENDPROC
1030 :
1040 PROCdelay(time%)
1050 A=INKEY(time%)
1060 IF A<>-1 end=TRUE
1070 ENDPROC
```

```
1080 :
1090 DEFPROCtop
1100 rate=FNminicam("CO1",scalar1%,500,0)+256*256*FNminicam("RD2",
     scalar2%,0,0)
1110 error=SQR(rate)*3/5: rate=rate/5
1120 decr$="D": incr$="I"
1130 IF end=TRUE ENDPROC ELSE count=FNintensity
1140 IF count<(background-SQR(background)) fault=TRUE: ENDPROC
1150 IF (count<(rate+error)) AND (count>(rate-error)) PROCdelay(500):
     GOTO1130
1160 last_count=count: PROCdecrease: count=FNintensity: IF
     count<last_count GOTO 1180
1170 IF count>(rate-error) GOTO 1130 ELSE GOTO 1160
1180 last_count=count: PROCincrease: count=FNintensity: IF
     count<last_count GOTO 1160
1190 IF count>(rate-error) GOTO 1130 ELSE GOTO 1180
1200 :
1210 DEFFNintensity
1220 A%=FNminicam("CO1",scalar1%,500,0)+256*256*FNminicam("RD2",
     scalar2%,0,0)
1230 =A%/5


  10 REM ----------------------------
  20 REM -----  UTILITIES PROGRAM  -----
  30 REM ----------------------------
  40 :
  50 PROCtitle(12,5,"UTILITIES"): CLS: PROCcommand(" ")
  60 PROCmenu
  70 PRINTTAB(0,19); "                         "; : COLOUR0: COLOUR129: PRINT"Enter
     utilities subject >> "; : COLOUR1: COLOUR128: PRINT"  ";
  80 INPUT""choice$
  90 IF LEN(choice$)>2 VDU7: GOTO70
 100 IF choice$="BR" CHAIN"P.BRAGG"
 110 IF choice$="RE" CHAIN"P.REFL"
 120 IF choice$="SP" CHAIN"P.SPDATA"
 130 IF choice$="QU" CHAIN"P.MENU"
 140 VDU7: GOTO70
 150 :
 160 DEFPROCtitle(X%,Y%,string$)
 170 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "          ";
 180 PRINTTAB(X%,Y%); string$;
 190 VDU28,0,30,79,7: COLOUR1: COLOUR128
 200 ENDPROC
 210 :
 220 DEFPROCcommand(string$)
 230 VDU28,0,31,79,31: COLOUR0: COLOUR129: CLS
 240 PRINTTAB(INT(40-LEN(string$)/2)); string$;
 250 VDU28,0,30,79,7: COLOUR1: COLOUR128
 260 ENDPROC
 270 DEFPROCmenu
 280 PRINT'
 290 PRINT'"  BRagg angle - calculate Bragg angle for a material"
 300 PRINT'"  REflectivity - calculates single crystal relflectivity
     curve"
 310 PRINT'"  SPool data - spools a data file to an ascii file"
 320 PRINT'"  QUit  -  leave help, return to main menu"
 330 PRINT''"Select choice by keying in the two letters in capitals shown
     above"
 340 ENDPROC
 10 REM -------------------------------------------------
 20 REM -----   SAVE ROUTINE - COPIES TEMPORARY FILE   -----
 30 REM -------------------------------------------------
 40 :
 50 DIM CO%(600)
 60 PROCtitle(12,5,"SAVE DATA"): CLS
 70 PRINT''" When prompted enter the file name for data storage,":
    PRINT" if no drive number is given drive 2 is assumed"
 80 INPUT'" Enter file name "file$
 90 IF INSTR(file$,".")=0 file$="D."+file$
100 IF INSTR(file$,":")=0 file$=":2."+file$
110 PRINT''" Reading data from file :0.D.RTEMP, writing data to file ";
    file$
120 ON ERROR GOTO 510
130 DER%=0
140 X%=OPENIN(":0.D.RTEMP")
150 INPUT#X%,NU%
160 IF DER%<>0 AND ERR=222 PRINT'" There is no temporary data file -
    press any key for menu": A=GET: CLOSE#X%: CHAIN"P.MENU"
170 INPUT#X%,step
180 FOR I%=1 TO NU%: INPUT#X%,CO%(I%): NEXT
190 CLOSE#X%
200 DER%=0
210 X%=OPENIN(":0.D.HEADER")
220 INPUT#X%,place$
230 IF DER%<>0 AND ERR=222 PRINT'" There is no header file - press key
    for menu and use UPDATE": A=GET: CLOSE#X%: CHAIN"P.MENU"
240 INPUT#X%,sname$
250 INPUT#X%,volt$
260 INPUT#X%,current$
270 INPUT#X%,first_crystal$
280 INPUT#X%,wave$
290 INPUT#X%,refl$
300 CLOSE#X%
310 DER%=0
320 X%=OPENOUT(file$)
330 IF DER%<>0 AND ERR=204 PRINT'" Chosen file name too long - choose
    another "; : INPUTfile$: GOTO 90
340 PRINT#X%,NU%
350 PRINT#X%,step
360 FORI%=1 TO NU%
370   PRINT#X%,CO%(I%)
380   NEXT
390 PRINT#X%,place$
400 PRINT#X%,sname$
410 PRINT#X%,volt$
420 PRINT#X%,current$
430 PRINT#X%,first_crystal$
440 PRINT#X%,wave$
450 PRINT#X%,refl$
460 CLOSE#X%
470 :
480 PROCcommand("Press <CR> for repeat, any other key for menu")
490 A=GET: IF A=13 PROCcommand(""): CLS: GOTO70
500 CHAIN"P.MENU"
510 REM ERROR TRAP FOR ILLEGAL DISC ACCESS
520 DER%=DER%+1
530 IF ERR=214 PRINT'" File does not exist"
```

```
540 IF ERR=204 PRINT'"  File name too long"
550 IF ERR=222 PRINT'"  File does not exist"
560 IF ERR<>214 AND ERR<>204 AND ERR<>222 REPORT: END
570 IF ERL<=190 GOTO 160
580 IF ERL>190 AND ERL<=300 GOTO 230
590 IF ERL>300 GOTO 330
600 :
610 DEFPROCtitle(X%,Y%,string$)
620 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "               ";
630 PRINTTAB(X%,Y%); string$;
640 VDU28,0,30,79,7: COLOUR1: COLOUR128
650 ENDPROC
660 :
670 DEFPROCcommand(string$)
680 VDU28,0,31,79,31: COLOUR0: COLOUR129: CLS
690 PRINTTAB(INT(40-LEN(string$)/2)); string$;
700 VDU28,0,30,79,7: COLOUR1: COLOUR128
710 ENDPROC
```

```
 10 REM -------------------------
 20 REM -----  HELP PROGRAM  -----
 30 REM -------------------------
 40 :
 50 PROCtitle(12,5,"HELP"): CLS: PROCcommand(" ")
 60 PROCmenu
 70 PRINTTAB(0,19); "                   "; : COLOUR0: COLOUR129: PRINT"Enter
    help subject >> "; : COLOUR1: COLOUR128: PRINT"  ";
 80 INPUT""choice$
 90 IF LEN(choice$)>2 VDU7: GOTO70
100 IF choice$="IN" PROCintro: CLS: GOTO60
110 IF choice$="MM" PROCmove: CLS: GOTO60
120 IF choice$="PL" PROCplot: CLS: GOTO60
130 IF choice$="DA" PROCsave: CLS: GOTO60
140 IF choice$="TR" PROCtransfer: CLS: GOTO60
150 IF choice$="QU" CHAIN"P.MENU"
160 VDU7: GOTO70
170 :
180 DEFPROCtitle(X%,Y%,string$)
190 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "               ";
200 PRINTTAB(X%,Y%); string$;
210 VDU28,0,30,79,7: COLOUR1: COLOUR128
220 ENDPROC
230 :
240 DEFPROCcommand(string$)
250 VDU28,0,31,79,31: COLOUR0: COLOUR129: CLS
260 PRINTTAB(INT(40-LEN(string$)/2)); string$;
270 VDU28,0,30,79,7: COLOUR1: COLOUR128
280 ENDPROC
290 DEFPROCmenu
300 PRINT'
310 PRINT'"  INtro  -  gives a brief introduction to the use of these
    programs"
320 PRINT'"  Moving Motors  -  info on driving the axes and updating of
    poistions"
330 PRINT'"  PLotting  -  how to record a rocking curve and obtain hard
    copy etc"
340 PRINT'"  DAta storage  -  how to save data and its format on disc"
350 PRINT'"  TRansfer  -  how to transfer data to a remote host etc"
360 PRINT'"  QUit  -  leave help, return to main menu"
```

```
370 PRINT''"Select choice by keying in the two letters in capitals shown
    above"
380 ENDPROC
390 :
400 DEFPROCintro
410 CLS: PRINT'
420 X%=OPENIN("T.INTRO")
430 PROCdisplay
440 ENDPROC
450 :
460 DEFPROCmove
470 CLS: PRINT'
480 X%=OPENIN("T.MOTORS")
490 PROCdisplay
500 ENDPROC
510 :
520 DEFPROCdisplay
530 REPEAT: A%=BGET#X%: VDUA%: IF A%=13: VDU10
540   UNTIL EOF#X%
550 CLOSE#X%
560 PROCcommand("Press any key for menu")
570 A=GET
580 PROCcommand(" ")
590 ENDPROC
600 :
610 DEFPROCplot
620 CLS
630 X%=OPENIN("T.PLOTS")
640 PROCdisplay
650 ENDPROC
660 :
670 DEFPROCsave
680 CLS: PRINT'
690 X%=OPENIN("T.SAVE")
700 PROCdisplay
710 ENDPROC
720 :
730 DEFPROCtransfer
740 CLS: PRINT'
750 X%=OPENIN("T.TRANS")
760 PROCdisplay
770 ENDPROC
```

```
 10 REM -----------------------------------
 20 REM -----  RESET POSITIONS ROUTINE  -----
 30 REM -----------------------------------
 40 :
 50 DIM step(6): step(1)=0.1776: step(2)=0.121: step(3)=1.0/(96.0*50):
    step(4)=1.0/(96.0*50): step(5)=1.0/96.0: step(6)=1.0/96.0
 60 PROCtitle(12,5,"RESET")
 70 CLS
 80 PRINT''" The current positions of the axes are as shown above"
 90 PRINT'" When prompted enter the new position or <CR> to leave
    unaltered"
100 INPUT'" Enter the position of axis 1 (deg) "val$
110 IF val$<>"" S%=VAL(val$)*3600/step(1): PROCtitle(34,1,val$)
120 PRINTTAB(0,6)""
130 INPUT'" Enter the position of axis 2 (deg) "val$
140 !F val$<>"" T%=VAL(val$)*3600/step(2): PROCtitle(34,2,val$)
```

```
150 PRINTTAB(0,8)""
160 INPUT'"  Enter the position of goniometer 1 (mm) "val$
170 IF val$<>"" U%=VAL(val$)/step(3): PROCtitle(34,3,val$)
180 PRINTTAB(0,10)""
190 INPUT'"  Enter the position of goniometer 2 (mm) "val$
200 IF val$<>"" V%=VAL(val$)/step(4): PROCtitle(34,4,val$)
210 PRINTTAB(0,12)""
220 INPUT'"  Enter the position of X-table micrometer (mm) "val$
230 IF val$<>"" K%=VAL(val$)/step(5): PROCtitle(47,2,val$)
240 PRINTTAB(0,14)""
250 INPUT'"  Enter the position of Y-table micrometer (mm) "val$
260 IF val$<>"" L%=VAL(val$)/step(6): PROCtitle(47,3,val$)
270 CHAIN"P.MENU"
280 :
290 DEFPROCtitle(X%,Y%,string$)
300 VDU28,0,6,79,0: COLOUR0: COLOUR129: PRINTTAB(X%,Y%); "            ";
310 PRINTTAB(X%,Y%); string$;
320 VDU28,0,30,79,7
330 COLOUR1: COLOUR128
340 ENDPROC
```

# REFERENCES

AINSLIE, B.J., DAY, C.R., FRANCE, P.W., BEALES, K.J. and NEWNS, G.R. (1979) Electron. Lett. **15**, 411-413.

ALLISON, S.K. and WILLIAMS, J.H. (1930) Phys. Rev. **35**, 1476-1490.

ALLISON, S.K. (1932) Phys. Rev. **41**, 1-20.

BARTELS, W.J. and NIJMAN, W. (1978) J. Cryst. Growth, **44**, 518-525.

BARTELS, W.J. (1983) J. Vac. Sci. Technol. **B 1(2)**, 338-345.

BATTERMAN, B.W. and COLE, H.C. (1964) Rev. Mod. Phys. **36**, 3, 681-717.

BIEFILED, R.M., OSBOURN, G.C., GOURLEY, P.L. and FRITZ, I.J. (1983) J. Electron. Mater. **12** 903-908.

BOHM, H. (1975) Acta Cryst. **A31**, 622-628.

BONSE, U., KRASNICKI, S. and TEWORTE, R. (1983) Nucl. Instrum. Methods, **208**, 711-712.

BONSE, U., OLTHOFF-MUNTER, K. and RUMPF, A. (1983) J. Appl. Cryst. 524-531.

BOWEN, D.K. and DAVIES, S.T. (1983) Nucl. Instrum. Methods, **208**, 725-729.

BURGEAT, J. and COLELLA, R. (1969) J. Appl. Phys. **40**, 3505-3509.

BURGEAT, J., QUILLEC, M., PRIMOT, J., LeROUX, G. and LAUNOIS, H. (1981) Appl. Phys. Lett. **38**(7), 711-712.

CHANG, L.L., ESAKI, L., HOWARD, W.E. and LUDEKE, R. (1973) J. Vac. Sci. Technol., **10**, 11-17.

CHANG, L.L., SEGMULLER, A. and ESAKI, L. (1976) Appl. Phys. Lett., **28**(1), 39-41.

CHANG, L.I. and KOMA, A. (1976) Appl. Phys. Lett. **29**, 138-140.

CHANG, S.L., PATEL, N.B., NANNICHI, Y. and DePRINCE, F.C. (1979) J. Appl. Phys., **50**, 2975-2976.

CHANG, L.L. (1980) Handbook Of Semiconductors, edited by S.P. Keller (Amsterdam: North Holland).

CHANG, S.L. (1980) Appl. Phys. Lett. **37**, 819-821.

CHATERJEE, A.K., FAKTOR, M.M., LYONS, M.H. and MOSS, R.H. (1982), J. Cryst. Growth, **56**, 591-604.

CHO, A.Y. and ARTHUR, J.R. (1975) Prog. Soild. State. Chem., **10**, 157-160.

CHU, S.N.G., MACRANDER, A.T., STREGE, K.E. and JOHNSTON Jr., W.D. (1985) J. Appl. Phys. **57**(2), 249-257.

COMPTON, A.H. (1931) Rev. Sci. Instrum., **2**, 365-371.

COMPTON, A.H. and ALLISON, S.K. (1935) X-Rays In Theory And Experiment, New York: Van Norstrand.

DANIEL, V. and LIPSON, H. (1943) Proc. Roy. Soc. London Ser A, **181**, 378.

DANIEL, V. and LIPSON, H. (1944) Proc. Roy. Soc. London Ser A, **182**, 378.

DAVIDENKOV, N.N. (1961) Sov. Phys. Solid State **2**, 2595-2598.

DEHLINGER, U. (1927) Z. Kristallogr., **65**, 615-631.

DINGLE, R., GOSSARD, A.C. and WEIGMANN, W. (1975) Phys. Rev. Lett., **34**, 21, 1327-1330.

DINGLE. R. (1975) Advances In Solid State Physics Vol XV, edited by H.J. Quesser (Braunshweig: Pergammon-Vieweg).

DINGLE, R. (1977) J. Vac. Sci. Technol. **14**, 1006-1011.

DINGLE, R., STORMER, H.L., GOSSARD, A.C. and WEIGMANN, W. (1978) Appl. Phys. Lett., **33**, 665-667.

DOHLER, G.H. (1972) Phys. Stat. Sol., (b) **52**, 79-92.

DOHLER, G.H. (1972) Phys. Stat. Sol., (b) **52**, 533-545.

DOHLER, G.H., KUNZEL, H., OLEGO, D., PLOOG, K., RUDEN, P., STOLTZ, H.J. and ABSTREITER, G. (1981) Phys. Rev. Lett., **47**, 864-867.

DuMOND, J.W.M. (1937) Phys. Rev. **52**, 872-883.

DuMOND, J.W.M. and MARLOW, D. (1937) Rev. Sci. Instrum. **8**, 112-121.

EBERT, G., Von KLITZING, K., PROBST, C. and PLOOG, K. (1982) Solid State Commun. **44**, 95-98.

ENTIN, I.R. (1978) Phys. Stat. Sol. (b) **90**, 575-584.

ESAKI, L. and TSU, R. (1970) IBM Jnl. Res. Dev. **14**, 61-65.

ESAKI, L. CHANG, L.L. and TSU, R. (1971) Proceedings Of 12th International Conference on Low Temperature Physics, pp 531-553. Edited by Eizo Kanda, Tokyo:Keigaku.

ESAKI, L. and CHANG, L.L. (1974) Phys. Rev. Lett. **33**, 495-498.

FEWSTER, P.F. (1985) J. Appl. Cryst. (In Press).

FLEMING, R.M., McWHAN, D.B., GOSSARD, A.C., WEIGMANN, W. and LOGAN, R.A. (1980) J. Appl. Phys. **51**(1), 357-363.

FONTAINE, D.De, (1966) Local Atomic Arrangements Studied By X-Ray Diffraction. Metallurgical Society Conferences Vol 36.

FOXON, C.T. and JOYCE, B.A. (1981) Current Topics In Materials Science Vol. 7, Edited by E. Kaldis (Amsterdam: North Holland).

FUKUHARA, A. and TAKANO, Y. (1977) Acta Cryst. **A33**, 137-142.

FUKUHARA, A. and TAKANO, Y. (1977) J. Appl. Cryst. **10**, 387-391.

GOETZ, K.-H., BIMBERG, D., JURGENSEN, H., SELDERS, J., SOLOMONOV, A.V., GLINSKII, G.F. and RAZEGHI, M. (1983) J. Appl. Phys. **54**(6) 4543-4552.

GOURLEY, P.L. and BIELIELD, R.M. (1984) Appl. Phys. Lett. **45**(7), 749-751.

HALLIWELL, M.A.G., LYONS, M.H., TANNER, B.K. and ILCZYSZYN, P. (1983) J. Cryst. Growth, **65**, 672-678.

HALLIWELL, M.A.G., JULER, J. and NORMAN, A.G. (1983) Microscopy of Semiconductor Materials, Oxford, 1983, Inst. Phys. Conf. Ser. No.67 (Inst. Phys., London-Bristol, 1983), p.365.

HARGREAVES, M.E. (1951) Acta Cryst. **4**, 301-309.

HART, M. (1980) Characterization Of Crystal Growth Defects By X-Ray Methods, edited by B.K. Tanner and D.K. Bowen (New York: Plenum) pp 421-432.

HARTWIG, J. (1977) Phys. Stat. Sol. (a) **42**, 495-500.

HARTWIG, J. (1978) Exper. Technik. Physik **26**, 5, 447-456.

HORNSTRA, J. and BARTELS, W.J. (1978) J. Cryst. Growth, **44**, 513-517.

HSEIH, J.J., FINN, M.C. and ROSSI, J.A. (1977) Gallium Arsenide and Related Compounds (St Louis, 1976), Inst. Phys. Conf. Ser. No.33b, pp 37-44.

ISHERWOOD, B.J., BROWN, B.R. and HALLIWELL, M.A.G. (1981) J. Cryst. Growth, **44**, 449-460.

INTERNATIONAL TABLES FOR X-RAY CRYSTALLOGRAPHY (1974) Vol. IV, Birmingham: Kynoch Press (Present Distributor D. Reidal: Dordrecht).

JAGER, H. (1965) Z. Angew. Phys. **20**, 73-79.

JAGER, H. (1966) Z. Angew. Phys. **21**, 543-548.

JAMES, R.W. (1948) The Optical Principles Of The Diffraction Of X-Rays: The Crystalline State Vol. II. edited by L. Bragg (London: Bell).

JESSER, W.A. and KUHLMAN-WILSDORF, D. (1967) Phys. Stat. Sol. **19**, 95-105.

JUNG, H., DOHLER, G.H., KUNZEL, H., PLOOG, K., RUDDEN, P. and STOLTZ, H.J. (1982) Solid State Commun. **43**, 291-294.

JUNG, H., DOHLER, G.H., GOBEL, E.O. and PLOOG, K. (1983) Appl. Phys. Lett. **43**,

KAWAMURA, Y. and OKAMOTO, H. (1979) J. Appl. Phys. **50**(6), 4457-4458.

KERVAREC, J., BAUDET, M., CUALET, J., AUVRAY, P. EMERY, J.Y. and REGRENY, A. (1984) J. Appl. Cryst. **17**, 196-205.

KISHINO, S., OGIRIMA, M. and KUHRATA, K. (1972) J. Electrochem. Soc. **119**(5), 617-622.

KLITZING, K.Von, OBLOH, H., EBERT, G., KNECHT, J. and PLOOG, K. (1981) Natn. Bur. Stand. Spec. Publ. No. 617, 382.

KLAR, B. and RUSTICHELLI, F. (1973) Il Nuovo Cimento **13B**(2), 249-269.

KOCHENDORFER, A. (1939) Z. Kristallogr. **101**, 149-155.

KOMENOU, K., HIRAI, I., ASAMA, K. and SAKAI, M. (1978) J. Appl. Phys. **49**(12), 5846-5822.

KUNZEL, H., DOHLER, G.H., RUDEN, P. and PLOOG, K. (1982) Appl. Phys. Lett. **41**, 852-854.

LARSON, B.C. and BARHORST, J.F. (1980) J. Appl. Phys. **51**(6), 3181-3185.

LAUE, M.Von, (1931) Z. Physik **72**, 472-477.

LAUE, M.Von, (1960) Rontengenstrahl Interferenzen, Akademische (Frankfurt: Verlag).

LUDOWISE, M.J., DIETZE, W.T., LEWIS, C.R., HOLONYAK Jr., N., HESS, K., CAMRAS, M.D. and NIXON, M.A. (1983) Appl. Phys. Lett. **42**, 257-259.

MAAN, J.C., ENGLERT, Th., UIHLEIN, Ch., KUNZEL, H., PLOOG, K. and FISCHER, A. (1983) J. Vac. Sci. Technol. B1, 289.

MACRANDER, A.T., CHU, S.N.G., STREGE, K.E., BLOEMEKE, A.F. and JOHNSTON Jr., W.D. (1984) Appl. Phys. Lett. **44**, 615-617.

MATSUI, J., ONABE, K., KAMEJIMA, T. and HIYASHI, I. (1979) J. Electrochem. Soc. **126**, 664-667.

MATSUI, J., WATANBE, H. and SEKI, Y. (1979) J. Cryst. Growth **46**, 563-568.

MATTHEWS, J.W. and BLAKESLEE, A.E. (1977) J. Vac. Sci. Technol. **14**, 989.

MERIAM ABDUL GANI, S. (1982) PhD Thesis, University of Durham.

MEYER, K.E., FLETCHER, G.P., SINHA, S.K. and SCHULLER, I.K. (1981) J. Appl. Phys. **52**(11), 6608-6610.

MOSS, R.H. and SPURDENS, P.C. (1984) Electronics Lett. **20**, 978-979.

MOTOROLA MCU/MPU APPLICATIONS MANUAL VOL. 2 (1985), (Motorola).

MOTOROLA MCU/MPU INTERFACE COMPONENTS (1984), (Motorola).

NAHORY, R.E., POLLACK, M.A., JOHNSTON Jr., W.D. and BARNES, R.L. (1978) Appl. Phys. Lett. **33**, 659-662.

NEWMAN, D. (1982) New Electronics, May 18.

OE, K., SHINODA, Y. and SUGIYAMA, K. (1978) Appl. Phys. Lett. **33**(11), 962-964.

OLSEN, G.H. and ETTENBERG, M. (1977) J. Appl. Phys. **48**, 2543-2547.

OSBOURN, G.C. (1982) J. Appl. Phys. **53**(3), 1586-1589.

OSBOURN, G.C., BIEFIELD, R.M. and GOURLEY, P.L. (1982) Appl. Phys. Lett. **41**(2), 172-174.

PETROFF, P.M. (1977) J. Vac. Sci. Technol. **14**, 973-981.

PETROFF, P.M., GOSSARD, A.C., WEIGMANN, W. and SAVAGE, A. (1978) J. Cryst. Growth, **44**, 5-13.

PINSKER, Z.G. (1978) Dynamical Scattering Of X-Rays In Crystals (Berlin: Springer-Verlag).

PLOOG, K. (1980) Crystals: Growth Properties and Applications Vol. 3, edited by H.C. Freyhardt (Berlin, Heidelberg: Springer-Verlag).

PLOOG, K. (1981) A Rev. Mat. Sci. **11**, 171-179.

PLOOG, K. (1982) A Rev. Mat. Sci. **12**, 123-132.

PLOOG, K. and DOHLER, G.H. (1983) Advances In Physics, **32**(3), 285-359.

REZEK, E.A., VOJAK, B.A., CHIN, R. and HOLONYAK Jr., N. (1980) Appl. Phys. Lett. **36**, 744-746.

ROZGONYI, G.A. and CIESIELKA, T.J. (1973) Rev. Sci. Instrum. **44**(8), 1053-1057.

SAI-HALASZ, G.A., TSU, R. and ESAKI, L. (1977) Appl. Phys. Lett. **30**, 651-653.

SAI-HALASZ, G.A., ESAKI, L. and HARRISON, W.A. (1978) Phys. Rev. B **18**, 2812-2818.

SAKAKI, H., CHANG, L.L., LUDEKE, R., CHANG, C.A., SAI-HALASZ, G.A. and ESAKI, L. (1977) Appl. Phys. Lett. **31**, 211-213.

SAUL, R.H. (1969) J. Appl. Phys. **46**(8), 3273-3279.

SCHWARZSCHILD, M.M. (1928) Phys. Rev. **32**, 162-171.

SEGMULLER, A. (1973) Thin Solid Films **18**, 287-294.

SEGMULLER, A. and BLAKESLEE, A.E. (1973) J. Appl. Cryst. **6**, 19-25.

SEGMULLER, A., KRISHNA, P. and ESAKI, L. (1977) J. Appl. Cryst. **10**, 1-6.

SKIER, K. (1981) Beyond Games: System Software For Your 6502 Personal Computer (BYTE, McGraw-Hill: Peterborough, New Hampshire).

SPERIOSU, V.S., GLASS, H.L. and KOBAYASHI, T. (1979) Appl. Phys. Lett. **34**(9), 539-542.

SPERIOSU, V.S. (1981) J. Appl. Phys. **52**(10), 6094-6103.

SPERIOSU, V.S., PAINE, B.M., NICOLET, M.A. and GLASS, H. (1982) Appl. Phys. Lett. **40**(7), 604-606.

SPERIOSU, V.S. and WILTS, C.H. (1983) J. Appl. Phys. **54**(6), 3025-3343.

SPERIOSU, V.S. and VREELAND Jr., T. (1984) J. Appl. Phys. **56**(6), 1591-1600.

SPERIOSU, V.S., NICOLET, M.-A., PICRAUX, S.T. and BIEFIELD, R.M. (1984) Appl. Phys. Lett. **45**(3), 223-225.

SPERIOSU, V.S., NICOLET, M.-A., TANDON, J.L. and YEH, Y.C.M. (1985) J. Appl. Phys. **57**(4), 1377-1379.

STRINGFELLOW, G.B. (1972) J. Appl. Phys. **43**, 3455-3460.

TAKAGI, S. (1962) Acta Cryst. **15**, 1311-1312.

TAKAGI, S. (1969) J. Phys. Soc. Jpn. **26**, 1239-1254.

TAKEUCHI, T., OHATA, N., SUGITA, Y. and FUKUHARA, A. (1983) J. Appl. Phys. **54**(2), 715-720.

TANNER, B.K., BARNETT, S.J. and HILL, M.J. (1985) Microscopy of Semiconductor Materials, Oxford 1985. (In Press).

TANNER, B.K. and HILL, M.J. (1985) Proc. Denver Conf. Applications of X-ray Analysis, Denver, Colorado. To be published in: Advances in X-ray Analysis Vol. 29 (1986).

TASHIMA, M.M., COOK, L.W. and STILLMAN, G.E. (1981) J. Cryst. Growth **54**, 132-137.

TAUPIN, D. (1964) Bull. Soc. Franc. Miner. Cryst. **88**, 469-511.

TERAUCHI, H., SEKIMOTO, S., SANO, N., KATO, H. and NAKAYAMA, M. (1984) Appl. Phys. Lett. **44**(10), 971-973.

TIMOSHENKO, S. (1925) J. Opt. Soc. **11**, 233-248.

TSUI, D.C., STORMER, H.L. and GOSSARD, A.C. (1982) Phys. Rev. B **25**, 1405-1407.

TSUI, D.C. STORMER, H.L. and GOSSARD, A.C. (1982) Phys. Rev. Lett. **48**, 1559-1561.

VARDANYAN, D.M. and MANOUKYAN, H.M. (1982) Phys. Stat. Sol. (a) **69**, 475-482.

VARDANYAN, D.M., MANOUKYAN, H.M. and PETROSYAN, H.M. (1985) Acta Cryst. **A41**, 212-217.

VARDANYAN, D.M., MANOUKYAN, H.M. and PETROSYAN, H.M. (1985) Acta Cryst. **A41**, 218-222.

VILMS, J. and KERPS, D. (1982) J. Appl. Phys. **53**, 1536-1537.

VOEGLI, O., CALHOUN, B.A., ROSIER, L.L., and SLONCZEWSKI, J.C. (1974) AIP Conf. Proc. **24**, 617-623.

YOSHIMURA, J. (1984) J. Appl. Cryst. **17**, 426-434.

ZACHARIASEN, W.H. (1945) Theory of X-ray Diffraction in Crystals (New York: John Wiley).

ZELLER, C.H., VINTER, B., ABSTREITER, G. and PLOOK, K. (1982) Phys. Rev. B**26**, 2124-2132.