Thermal conductivity studies of YBa(_2)Cu(_3)O(_7-δ)

Delap, Martin Richard

Thermal Conductivity Studies of $YBa_2Cu_3O_{7-\delta}$

by

Martin Richard Delap

**A Thesis submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy**

Department of Physics

The University of Durham
1990

To my Parents

Abstract

Apparatus to measure the thermal conductivity of $YBa_2Cu_3O_{7-\delta}$ at temperatures between 20K and 120K has been designed and constructed. The thermal conductivity is measured using a longitudinal steady state heat flow technique. Thermal conductivity measurements have been performed upon a sample of $YBa_2Cu_3O_{7-\delta}$ which has been subjected to a series of heat treatments in order to remove oxygen from the material. The measurements show conclusively that the thermal conductivity of $YBa_2Cu_3O_{7-\delta}$ is very strongly influenced by the oxygen content of the material. A reduction of the oxygen content of the material results in a substantial lowering of the thermal conductivity. To explain this result, a quantitative model has been constructed. The model demonstrates that consideration of the changes in phonon interactions alone cannot account for the differences in the behaviour of the thermal conductivity of $YBa_2Cu_3O_6$ and $YBa_2Cu_3O_7$. In addition, the model shows that there must be a significant carrier contribution to the thermal conductivity in both the normal and superconducting states. A physical process has been proposed which provides the required large carrier contribution below T_c. Further studies have been performed on a series of samples of $YBa_2Cu_3O_{7-\delta}$ which were sintered at slightly different temperatures. Qualitative analysis of the physical properties of these samples has been performed.

Acknowledgements

I am grateful to a large number of people, within the field of Physics and beyond, who have supported me during the course of this work. I would like to thank all of those who have educated me up to this time. Many members of the Durham Physics department have contributed to making this study enjoyable. I am particularly indebted to the following whom I would like to take this oppportunity to thank

Professor A.D.Martin, Professor A.W.Wolfendale for the provision of the facilities of the Physics department. Dr W.D.Corner and Professor B.K.Tanner for enabling me to use the facilities of the Solid State Physics Research Group.

Dr. N.R.Bernhoeft, who supervised this work. I am particularly appreciative of his advice and encouragement. I hope that his infectious enthusiasm for Physics will remain with me.

Cookson Group Plc who have provided the financial support for this work. Dr A.J.Bell, Dr T.P.Beales and Mr J.S.Rogers from Cookson Group Plc for their guidance and assistance, especially with the work contained in chapter VII.

Dr D.K.Thomas and Mrs P.Bransden from the University Industrial Research Laboratories who have overseen and run the Cookson contract.

Mr P.J.Allen, Mr S.Cockerton, Miss S.M.Thompson and Mr P.Warren for their endless cheerful support and advice, and for teaching me in the ways of the VSM and the university computer system. Mr C.I.Gregory for his assistance in obtaining the magnetisation curves.

Dr D.Evans and Dr D.P.Hampshire for useful discussions.

Mr P.Foley and Mr J.Scott for their assistance and technical support in the day-to-day running of the equipment.

Mr D.Jobling and his staff in the Student Workshop, Mr P.Armstrong, Mr G.Teasdale and Mr M.Greener for their hard work and formidable ability at turning my brief sketches into precise pieces of equipment.

Mr T.Jackson and the staff from the Electronics Workshop, Mr C.Mullaney, Mr D.Stockdale, Mr T.Adamson and Mr C.Moore for their seemingly endless assistance, and for producing the sample mounting boards for the resistance measurements.

Mrs P.Russell in the Drawing Office for her advice with diagram design, for the enthusiasm and skill with which she labelled all of the diagrams and for drawing the more complicated diagrams.

Mr M.Lee and Miss V.Greener in the Audio Visual Workshop for the production of the optical micrographs, binding the soft copies of this thesis and, along with Mrs P.Russell, teaching me the finer points of using the photocopier.

Miss J.Outson, Mrs N.Bingham, Mrs C.Webster for typing various letters, helping with all matters secretarial, and placing all manner of orders for stationery and equipment.

Mr R.Hardy from the Geology Department at Durham for his enthusiastic help with the X-ray powder diffraction.

I greatly appreciate the support given to me by my brothers, Stephen and Graeme.

Thanks to Miss J.D.Guilliard for her continual encouragement and advice, I cannot express how much her support has meant.

Finally, I am especially indebted to my parents for all their support, morally and financially - Thank you.

Publications

Thermal conductivity studies of $YBa_2Cu_3O_{7-\delta}$.
M.R.Delap, N.R.Bernhoeft, T.P.Beales, J.Rogers.
Accepted for publication in Chemtronics.

Declaration

I declare that the work contained in this thesis has not been submitted for a degree at this University or any other. All work presented was conducted by the author unless stated otherwise.

Contents

Chapter I

Superconductivity

One should always be a little improbable
— Oscar Wilde.
Phrases and Philosophies for the use of the Young

1.1 Historical

Unlike many other physical properties of materials, the phenomenon of superconductivity was not observed until the early part of the 20^{th} century. In 1908, Kammerling Onnes successfully overcame immense experimental difficulties and managed to liquefy helium. Three years later Onnes discovered that mercury exhibited a dramatic fall in electrical resistance just above the boiling point of liquid helium, such that the resistivity at 4.2K was below the limit of his measuring apparatus (figure 1.1). [Onnes (1911)[1]] Since those early days thousands of materials, ranging from pure elements to alloys and oxides, have been found to undergo a similar phase transition at sufficiently low temperatures. Table 1.1 shows the transition temperatures of the known superconducting elements and some of the highest critical temperature superconducting materials discovered before 1986.

[1] Accounts of the early superconductivity work can be found in Gorter (1964) and Mendelssohn (1964).

Figure 1.1 — First observation of superconductivity.
Onnes' (1911) plot of resistance of a mercury specimen as a function of temperature. Reproduced from Kittel (1976).

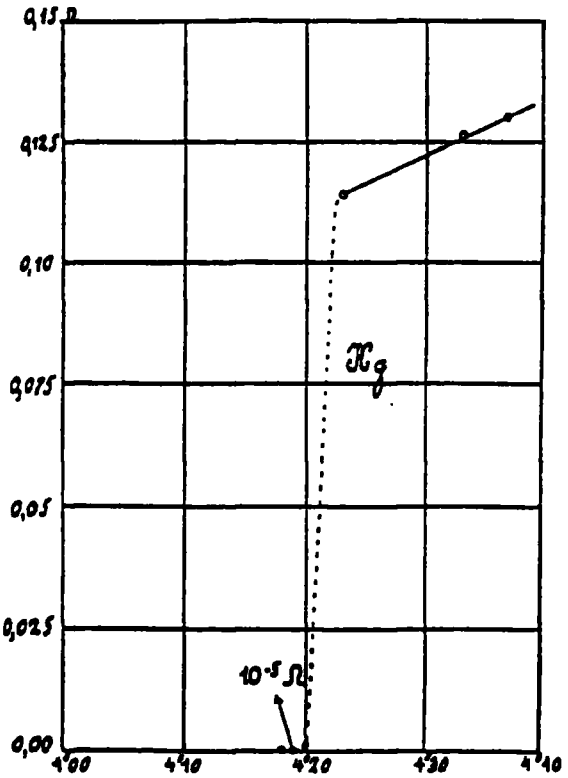


Table 1.1 — Superconducting elements and compounds.
Compiled by Allen (1991). Original data sources Matthias et al (1963), Savitskii et al (1981) and Philips (1989).

Elements			
Element	T_c(K)	Element	T_c(K)
Al	1.14	Re	1.40
Cd	0.56	Ru	0.51
Ga	1.09	Sn	3.72
Hf	0.12	Ta	4.48
Hg	4.15	Tc	7.77
In	3.40	Th	1.36
Ir	0.14	Ti	0.39
La	6.00	Tl	2.39
Lu	0.10	U	0.20
Mo	0.92	V	5.38
Nb	9.50	W	0.48
Os	0.66	Zn	0.88
Pa	1.40	Zr	0.54
Pb	7.49		

Compounds	
Formula	T_c(K)
Nb_3Ge	23.2
Nb_3Ga	21.0
Nb_3Sn	18.1
NbN	17.2
V_3Si	17.2
V_3Ga	16.8
Ta_3Pb	16.0
$Mo_6Pb_{0.9}S_{7.5}$	15.2
Mo_3Re	15.0
MoC	14.3
$LuRh_4B_4$	11.7
La_3In	10.4
ZrV_2	8.8
$LaSn_3$	6.55

1.2 Basic Phenomena

1.2.1 Persistent Currents

The complete disappearance of (dc) electrical resistivity is most elegantly demonstrated by the observation of a persistent current. In such an experiment an electrical current is applied to a ring of superconducting material, the current source is removed and due to the absence of a decay process the current will continue to flow indefinitely. File and Mills (1963) established a lower bound of 100,000 years on the decay time of a supercurrent using a nuclear magnetic resonance technique. Figure 1.2 shows typical data obtained from such experiments.

1.2.2 Meissner–Ochsenfeld Effect

Some years after Onnes' discovery, Meissner and Ochsenfeld (1933) observed that a specimen of superconducting material excludes magnetic flux when it is placed in a magnetic field and cooled below its transition temperature. This effect is not predicted simply by perfect conductivity. In a perfect conductor, current would flow in the material so as to oppose the existing magnetic flux in the material, the flux would be 'trapped' in the material when the field was removed. In the Meissner-Ochsenfeld effect the magnetic flux is actually excluded from the sample.

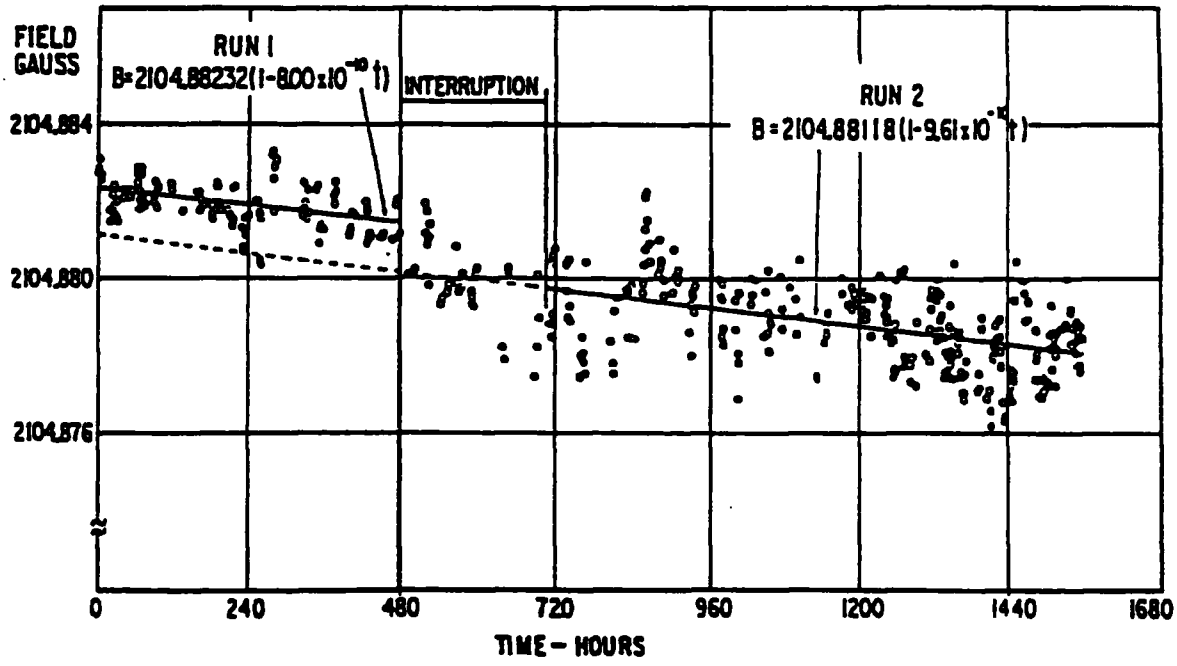
1.2.3 Critical Magnetic Field

The magnetic field applied to a superconductor is opposed by screening currents which are set up to cancel the magnetic field inside the material. If the applied field is sufficiently large a state will be reached where it is energetically favourable for the material to return to the normal state in preference to increasing the screening currents. The field at which this occurs is known as the thermodynamic critical magnetic field, B_c(T), and may be determined by equating the difference in free energy between the normal and superconducting states with the magnetic energy per unit volume.

$$\frac{B_c^2(T)}{2\mu_0} = f_n(T) - f_s(T) \tag{1.1}$$

Figure 1.2 — Decay of persistent currents.

File and Mills' (1963) data showing the decay of a persistent current determined by a nuclear magnetic resonance technique.



1.2.4 Critical Current Density

Maxwell's equations show that a magnetic field is associated with an electrical current. Increasing the electrical current applied to a superconductor eventually results in the magnetic field associated with the current exceeding the critical magnetic field. At this point the material will return to its normal state. The current density at which this occurs is known as the critical current density $J_c(\mathrm{T})$, and the phenomenon known as the Silsbee effect. [Silsbee (1916)]

1.3 Magnetic Penetration Depth

The screening currents, which act to expel magnetic flux, flow within the surface region of the sample to a depth which may be characterised by a length known as the magnetic penetration depth, λ. Hence a real superconductor does not exhibit flux expulsion from the entire sample. Schoenberg (1952) has shown experimentally that in very thin films, of thickness less than the penetration depth, there is appreciable deviation from perfect diamagnetism.

London and London (1935) postulated that the superconducting state can be described by some simple additions to Maxwell's equations.

$$\frac{\partial \mathbf{J}}{\partial t} = \frac{\mathbf{E}}{\mu_0 \lambda^2} \qquad \text{and} \qquad \nabla \times \mathbf{J} = \frac{-\mathbf{B}}{\mu_0 \lambda^2} \tag{1.2}$$

Simple algebraic manipulation of these equations leads to the expression for the penetration of magnetic field inside a superconductor

$$\nabla^2 \mathbf{B} = \frac{\mathbf{B}}{\lambda^2} \tag{1.3}$$

which, for $\mathbf{B}$ parallel to the x axis, has the solution

$$B = Be^{-x/\lambda} \qquad \text{if } B \neq 0 \tag{1.4}$$

These postulates indicate that the magnetic field inside a superconductor falls off exponentially from the surface, and provide a simple mathematical description of the effect, although there is perhaps little physical insight which can be gained from application of the equations.

1.4 Coherence Length

In 1950 Ginzburg and Landau produced a phenomenological theory of superconductivity, the importance of which was not initially widely appreciated. [Ginzburg and Landau (1950), Ginzburg (1955)] They introduced the idea of a pseudo wave function, Ψ, which was chosen as an order parameter for the superconducting electrons such that $|\Psi|^2$ gave the value of the local number density of 'super' electrons, n_s. Ψ must then satisfy the so-called Ginzburg-Landau equation

$$\frac{1}{2m^*}(-i\hbar\nabla - e\mathbf{A})^2\Psi + \beta|\Psi|^2\Psi = -\alpha(T)\Psi \tag{1.5}$$

This equation is very similar in form to the Schrödinger equation, with the addition of a non-linear term in Ψ^3. The Ginzburg-Landau theory leads to the introduction of a length scale which is a characteristic of the spatial distribution over which Ψ can vary without an 'undue' increase in energy, the Ginzburg-Landau coherence length ξ_{GL}.

$$\xi_{GL}(T) = \frac{\hbar}{[2m^*\alpha(T)]^{1/2}} \tag{1.6}$$

Near the superconducting transition the solution of the Ginzburg-Landau equation predicts the same temperature dependence for the magnetic penetration depth and the coherence length, thus it is commonplace to characterise superconducting materials by their temperature independent Ginzburg-Landau parameter

$$\kappa_{GL} = \frac{\lambda_{GL}}{\xi_{GL}} \tag{1.7}$$

The idea of a coherence length was proposed, independently of the Ginzburg-Landau theory, by Pippard (1953). The Pippard coherence length, ξ_0, arises from an appeal to basic physical insight. For a material which undergoes some phenomenon at an onset temperature T_c, only electrons within an energy kT_c of the Fermi energy can play a major role. These electrons will have a typical momentum of $\Delta p \simeq kT_c/v_f$ where v_f is the Fermi velocity. From the uncertainty principle $\Delta x \simeq \hbar v_f/kT_c$. In a pure superconductor well below T_c, $\xi_{GL} = \xi_0$ whereas near T_c, the term $\alpha(T)$ in equation (1.6) tends towards the value of $(T - T_c)$ hence $\xi(T)$ diverges as $1/(T - T_c)^{\frac{1}{2}}$.

1.5 Type I and Type II Superconductors

In fields below B_c, the existence of reversible, perfect diamagnetic behaviour indicates that the material is in a thermodynamically stable state of lowest free energy that can be reached by lowering the temperature and applying a magnetic field. As the applied field is increased it may become energetically favourable to allow magnetic flux to penetrate the material along an ordered lattice of lines of normal material. Such a state where normal and superconducting regions exist together is known as the mixed state and is a characteristic of a type II superconductor. Superconducting materials which do not form this state are classed as type I superconductors.

Type II superconductors have two characteristic magnetic fields, a lower critical magnetic field, B_{c1}(T), up to which a full Meissner–Ochsenfeld effect is seen, and a second critical magnetic field, B_{c2}(T), where the number of regions of normal material has increased to the extent that all of the material is in the normal state. A comparison of the magnetisation curves for 'ideal' type I and type II superconductors is given in figure 1.3. Abrikosov (1957) showed that the Ginzburg-Landau parameter could be used to determine whether a superconducting material was type I or type II; materials with $\kappa_{GL} < 1/\sqrt{2}$ are type I, whereas those with $\kappa_{GL} > 1/\sqrt{2}$ are type II. A further result of Abrikosov's work was that in the mixed state, such that $B_{c1}(\mathrm{T}) < B < B_{c2}(\mathrm{T})$, the magnetic flux is quantised and penetrates the material in a regular array of flux tubes. Each flux tube carries a flux quantum

$$\Phi_0 = \frac{h}{2e} \tag{1.8}$$

Around each flux tube the field decays in an exponential manner as described by the London equation. Magnetic decoration techniques have been used to confirm Abrikosov's prediction of a regular array of flux tubes (figure 1.4).

1.6 Energy Gap

The transition from the normal to superconducting state involves no change in the latent heat of the system, consequently the phase transition is of second order. However, at the superconducting transition there is a discontinuity in the specific heat capacity of the system which is indicative of the superconducting state

Figure 1.3 — Magnetisation curves of ideal superconductors.

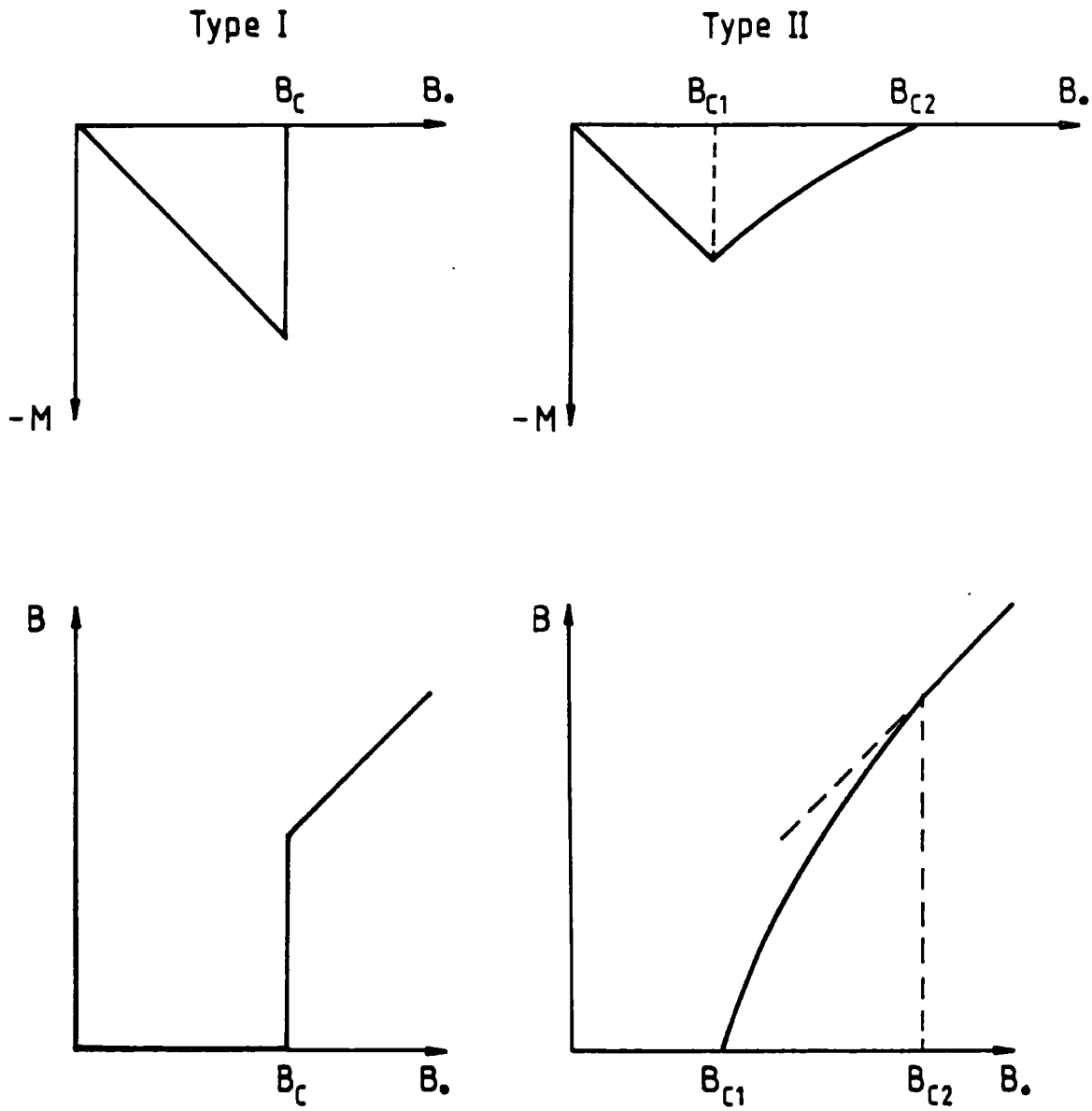
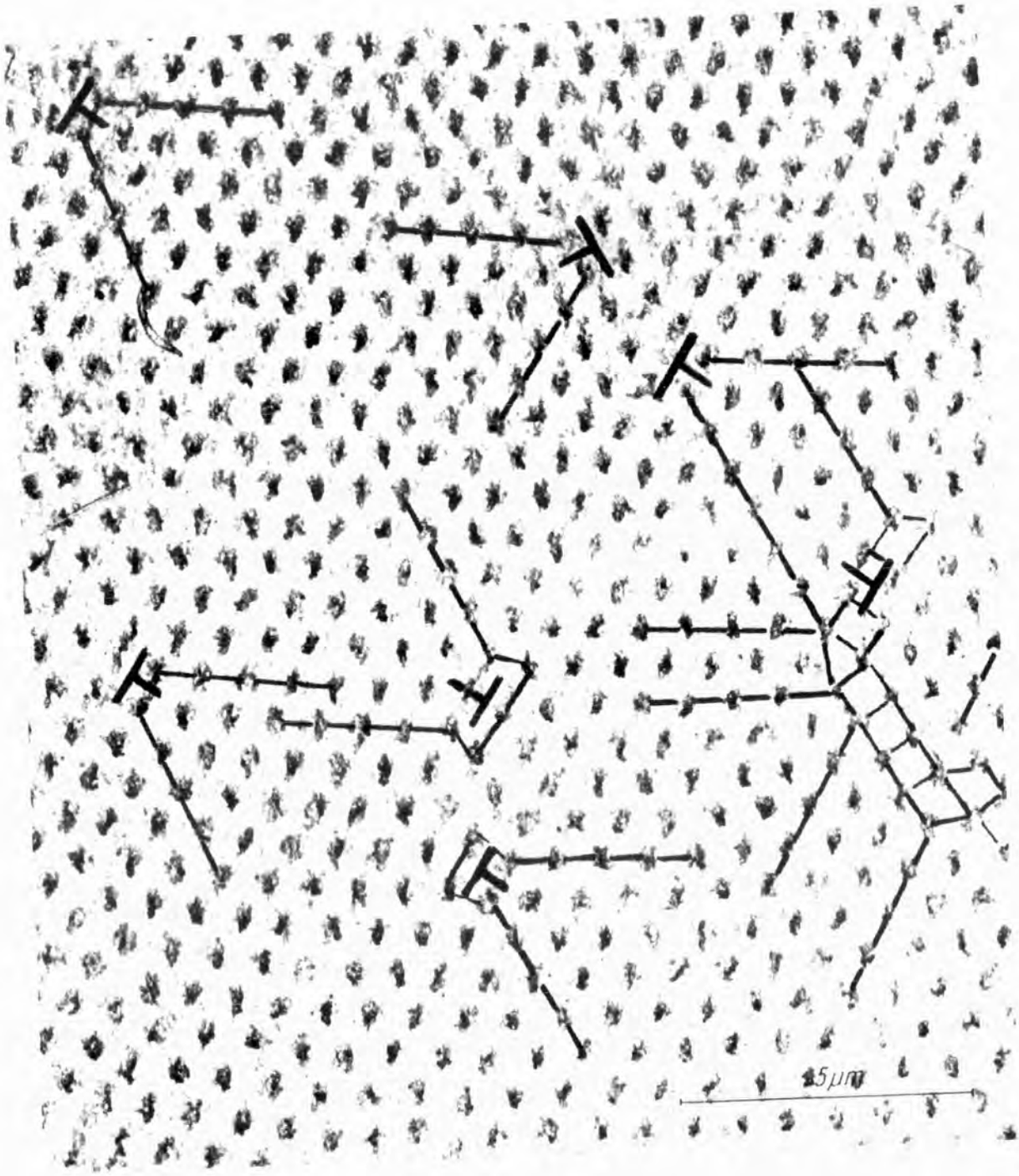


Figure 1.4 — Decoration of the magnetic flux line lattice.
The flux lines form an ordered triangular array. Reproduced from Träuble and Essmann (1968).



having a lower entropy than the normal state. A detailed analysis of the transition shows that the electronic contribution to the specific heat capacity is altered in a manner which is characteristic of a system with excited energy levels separated from the ground state by an energy gap, conventionally denoted 2Δ. The BCS theory of superconductivity proposed by Bardeen, Cooper and Schrieffer (1957) suggests that in the superconducting state the electrons experience an attractive potential and form a paired ground state. Within this framework the electron pair has a ground state energy lower, by an energy 2Δ, than that of two single electron ground states, thus the energy gap is indicative of the strength of the electron pairing.

1.7 Isotope Effect

BCS theory suggests that the electrons in the superconducting state form a paired ground state, the pairing arising from some attractive interaction. If this pairing is caused by an electron-phonon interaction the superconducting transition temperature of the material should be dependent upon the isotopic mass of the atoms comprising the crystal lattice according to the relationship

$$M^{\alpha}T_c = \text{constant} \tag{1.9}$$

For conventional 'low temperature' superconductors such as lead, niobium and mercury, α has the value $\frac{1}{2}$ to within experimental error. [Meservy and Schwartz (1969)] Elements such as zirconium, molybdenum and osmium have a value of α in the range 0.00 to 0.39. It is widely thought [Tinkham and Lobb (1989)] that such departures from $\alpha = \frac{1}{2}$ show only that the simple BCS picture must be generalised to account for the electron-electron interactions as well as the electron-phonon terms.

1.8 Fluctuation Effects

Thermodynamic fluctuations in a material may lead to local breakdown of superconductivity which will cause the existence of an astronomically small, but finite, resistivity in a superconducting material below T_c. Conversely, it is also possible for the fluctuations to reveal superconducting behaviour in a sample at a temperature fractionally above T_c. [Glover (1967)] As the coherence length is a

measure of the distance over which the superconducting order parameter may vary without loss of superconductivity it follows that in materials with a large coherence length, greater than a few interatomic distances, the electrons may interact coherently with each other and result in a sharp transition with little evidence of fluctuation effects. Conversely, materials with a short coherence length are expected to exhibit strong evidence for fluctuation effects.

1.9 Practical Superconductivity

1.9.1 Depairing Critical Current Density

The number density of electrons which carry a supercurrent in a conventional superconductor is constant for a given material at a fixed temperature. Increasing the current applied to a superconductor results in an increase in the electron velocity until the kinetic energy of the electrons is at least equal to the energy gap of the superconductor, at this point excitations may occur which destroy the superconductivity - effectively the superconducting pair is broken. The critical current density at which this phenomenon occurs is known as the depairing critical current density.

1.9.2 Flux Flow Resistivity

When a current is applied to a type II superconductor in the mixed state the flux lines experience a force $\mathbf{J} \times \mathbf{\Phi}$ which will tend to make them move. This flux motion will lead to an induced voltage (Lenz' law) which is in effect a resistive term. Any real material will contain inhomogeneities of many kinds e.g. twin boundaries, grain boundaries, point defects, chemical inhomogeneities. These regions will not be locally superconducting and therefore they may act as centres for flux pinning. In such a case, the flux will flow until it is pinned. As there is no longer flux motion, there will no longer be flux flow resistivity. A direct consequence of this flux pinning is that the superconductor will now be able to sustain a supercurrent up to a critical current density which can be one or more orders of magnitude greater than the defect-free material. In practical applications of superconductors, J_c(T) may be limited by the onset of flux flow resistivity rather than the Silsbee effect or the depairing critical current density. [Das Gupta et al (1978)]

1.9.3 Granular Superconductors

A granular superconductor is a material which contains regions of superconducting material which are separated by some non-superconducting regions in such a manner that there is a weak coupling between the superconducting regions. These boundaries may take many forms, they may be insulating oxide layers or simple defects which are less than or of comparable size to the coherence length of the superconducting material. The superconducting transition of a granular system can be divided into three main regions. In the first region, the resistance of each of the grains of material will fall due to the onset of superconductivity within the grain at some temperature T_c^{onset}, however bulk superconductivity will not be observed as thermal fluctuations will result in loss of coherence of the superconducting wave function between grains. As the temperature is reduced further, a state is reached where the coupling between the grains will overcome the thermal fluctuation effects. Finally, at a sufficiently low temperature, a percolation path will be established and a zero resistance state attained. Thus there may be considerable broadening of the transition due entirely to weak coupling between grains. [Pellan et al (1972), Raboutou et al (1980)]

1.10 Thermal Conductivity of Superconductors

BCS theory affirms that when a material undergoes a phase transition into the superconducting state the electrons which condense into the paired state are unable to scatter phonons and transport entropy. As a consequence of this there is a reduction in the electronic contribution to the thermal conductivity as the material is cooled through T_c. If the dominant contribution to the transport of heat is from the electrons, there will be a significant reduction in the thermal conductivity of the material below T_c. This is the case for many conventional superconductors because the phonon contribution is usually low in the region of T_c (figure 1.5).

If phonon scattering is dominant, for example in a material with a high concentration of defects, there will be a reduction in the electron-phonon scattering resulting in a longer effective phonon mean free path, therefore there will be a rise in the thermal conductivity below T_c. Very few conventional superconductors exhibit this effect because the phonon contribution to the thermal conductivity is

small at low temperatures. Some conventional superconductors which have very high defect concentrations do show this enhancement in the thermal conductivity below T_c, for example lead doped with 30% bismuth (figure 1.6).

Figure 1.5 — The thermal conductivity of pure lead.
• – normal, o – superconducting. Reproduced from Olsen(1952).

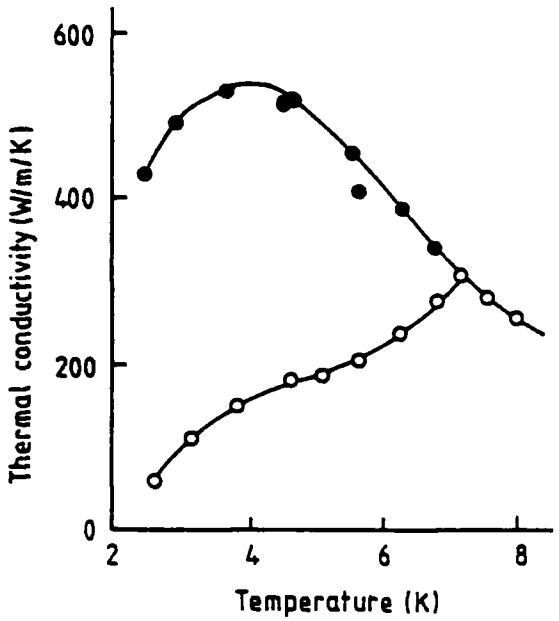
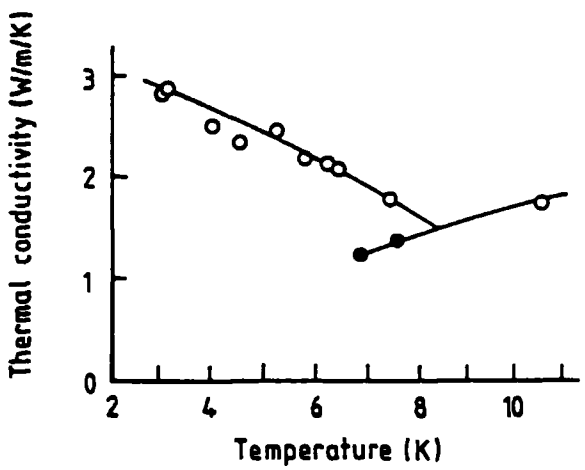


Figure 1.6 — The thermal conductivity of lead-30% bismuth.
• – normal, o – superconducting. Reproduced from Olsen(1952).



Chapter II

High Temperature Superconductivity

'Curiouser and curiouser!' cried Alice
— Lewis Carroll.
Alice in Wonderland

2.1 Introduction

Since Onnes' discovery, progress in finding materials which have higher and higher T_c values has been extremely slow. Before 1986, the highest known value of T_c was around 23.2K for Nb_3Ge. Bednorz and Müller (1986) suggested that a lanthanum based compound has a superconducting transition temperature in excess of the highest T_c for any superconductor then known, this was the first of the so-called 'high temperature' superconductors and for this work they were awarded a Nobel Prize for Physics. Since that time there has been a vast increase in the number of materials which are known to exhibit bulk, 'high temperature' superconductivity, all of these materials are oxides. Table 2.1 compares the critical temperature of some of the higher T_c oxide superconductors known before 1986, with those of the new 'high temperature' oxide superconductors.

Following the discovery of 'high temperature' superconductivity an immense amount of research resource and effort has been devoted to the subject resulting in the publication of tens of thousands of papers. A great many review articles and books have also been written, [see for example Bednorz and Müller (1988), Ginsberg (1989), Philips (1989), Yvon and François (1989), Jorgensen et al (1990)] the reader is initially referred to these for details of material properties not given in

Table 2.1 — Superconducting oxides.

Compiled by Allen (1991). Original data sources: Matthias et al (1963), Savitskii et at (1981) and Philips (1989).

Low T_c oxides	
Formula	T_c(K)
NbO	1.25
SnO	3.81
TiO	0.68
TiReO	5.74
$Pd_{0.285}Zr_{0.61}O_{0.105}$	2.09
$Ti_{0.573}Rh_{0.287}O_{0.14}$	3.37
$Zr_{0.611}Rh_{0.285}O_{0.105}$	11.8
Zr_3V_3O	7.5
$Mo_2PbS_6O_2$	11.7
$Mo_2Cu_2S_6O_2$	9.0
$Tl_{0.3}WO_3$	2.14
$K_{0.33}WO_3$	6.3
$Na_{0.33}WO_3$	3.0
$Ba_{0.13}WO_3$	1.9
$SrTiO_3$	0.39
$Sr_{0.7}Ca_{0.3}TiO_3$	0.6
$(Ba_{0.1}Sr_{0.9})TiO_3$	0.6
$(Ca_{0.31}Sr_{0.69})TiO_3$	0.6
Li_2TiO_4	13.7
$BaPb_{1-x}Bi_xO_{3-y}$	13.2

High T_c oxides	
Formula	T_c(K)
$(Ba_{1-x}K_x)BiO_3$	30
$La_{2-x}Ba_xCuO_4$	35
$La_{2-x}Sr_xCuO_4$	38.5
$La_{2-x}Ca_xCuO_4$	18
$(Nd,Ce,Sr)_2CuO_4$	28
$Nd_{2-x}Ce_xCuO_4$	24
$YBa_2Cu_3O_{7-x}$	93
$YBa_2Cu_4O_{8-x}$	80
$Y_2Ba_4Cu_7O_{15-x}$	40
$Bi_2(Sr_{2-x}Bi_x)CuO_6$	12
$Bi_2Sr_2CaCu_2O_8$	90
$(Bi,Pb)_2(Sr,Ca)_4Cu_3O_8$	110
$Tl_2Ba_2CuO_6$	92
$Tl_2Ba_2CaCu_2O_8$	110
$Tl_2Ba_2Ca_2Cu_3O_{10}$	125

this chapter. An obvious consequence of the immense quantity of published material is that the articles cited in this chapter are not necessarily primary references and do not imply any priority of publication for a given result, although in a few cases such citations are possible and these are indicated in the text.

2.2 Range of High T_c Materials

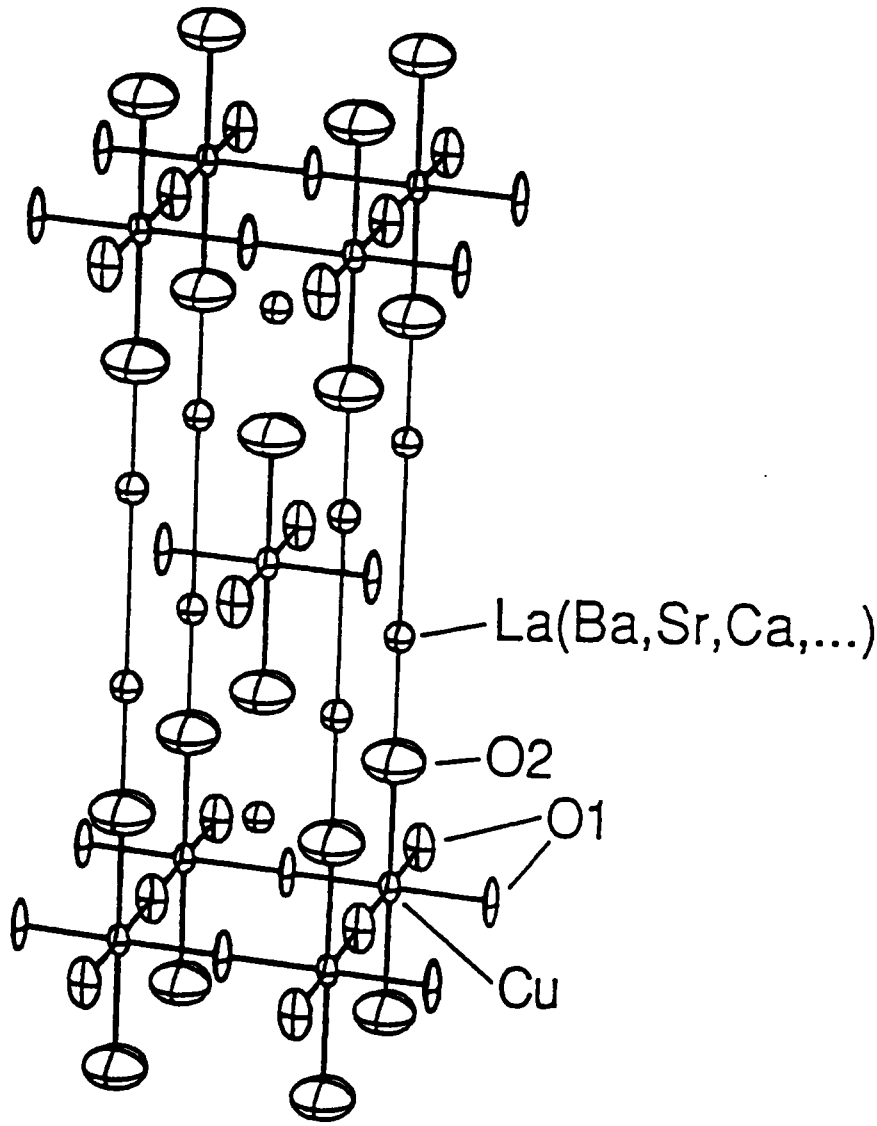
2.2.1 La-Sr-Cu-O

Lanthanum strontium copper oxide was the first 'high temperature' superconductor to be discovered, its structure is identical to that of layered, undoped La_2CuO_4 which consists of a plane of copper atoms surrounded by 4 strongly bonded oxygen neighbours and a further 2 weakly bonded oxygen atoms sited above and below the copper atom. These layers are in turn intercalated by two La-O layers as shown in figure 2.1. Stoichiometric La_2CuO_4 is not itself superconducting, the material must be doped in order to produce a superconducting compound. Considerable work has been carried out by doping the system with a variety of elements, the highest T_c so far discovered for this system is near 39K for $La_{2-x}Sr_xCuO_y$ with $x = 0.15$.

2.2.2 Bi-Sr-Ca-Cu-O

The Bi-Sr-Ca-Cu-O class of materials were discovered by Maeda et al (1988) and Michel et al (1987) after the discovery of the La-Sr-Cu-O and Y-Ba-Cu-O systems. The Bi-Sr-Ca-Cu-O system is a family of superconducting materials which follow the general formula $Bi_mCa_{n-1}Sr_2Cu_nO_x$, each member of the series consists of n Cu-O_2 layers separated by Ca layers containing no oxygen, these layers are intercalated by two Sr-O layers and m Bi-O layers. The structure determination of each of the materials was complicated by the difficulty in producing monophasic samples, caused in part by the large, layered, unit cell (a stacking fault along the c axis may give rise to a second phase) and the fact that the material is quinternary. Although some members of this class of materials are non-superconducting, the transition temperature of the superconducting phases varies from 12K for the $n = 1$ $m = 2$ phase up to 110K for the $n = 3$ $m = 2$ phase.

Figure 2.1 — The structure of $La_{2-x}Sr_xCuO_y$
Reproduced from Schuller and Jorgensen (1989).



2.2.3 Tl-Ca-Ba-Cu-O

The Tl-Ca-Ba-Cu-O system of superconductors was discovered by Sheng and Hermann (1988) and is very similar to the Bi-Sr-Ca-Cu-O system. The generic formula is $Tl_mCa_{n-1}Ba_2Cu_nO_x$ and the materials again consist of n Cu-O layers separated by Ca layers which are in turn intercalated by two Ba-O layers and m Tl-O layers. The precise structures have again proved difficult to determine, for similar reasons to the bismuth based material. The highest transition temperature (125K) for any superconductor found to date belongs to this family for the $n = 3$ $m = 2$ and $n = 4$ $m = 1$ phases. [Parkin et al (1988)] Figure 2.2 depicts schematically the structure of the thallium and bismuth based systems.

2.3 Y-Ba-Cu-O

2.3.1 Structure

Following the discovery of $La_{2-x}Sr_xCuO_y$, a new system of superconducting oxides was discovered by Wu et al (1987), that of $RBa_2Cu_3O_{7-\delta}$ where R is one of the lanthanide elements (except Ce,Pm,Tb). There are 12 known superconducting compounds which have this $RBa_2Cu_3O_{7-\delta}$ composition, [Maple et al (1987)] most of these are metallic in character above T_c and, bar lanthanum, have a superconducting transition temperature of between 90K and 93K (figure 2.3). All have the same layered structure which is closely related to the perovskite structure. [Le Page et al (1987)] A surprising result of investigations into the $RBa_2Cu_3O_{7-\delta}$ systems has revealed that a large number of the superconducting properties such as T_c, B_{c1}, B_{c2} and J_c are roughly independent of the rare earth element.

Various research workers have shown that the Y-Ba-Cu-O material is composed of two basic structures. [Khachaturyan et al (1988), Beno et al (1987), Schuller et al (1987), Van Tendeloo et al (1987), Greedan et al (1987)] The first is an orthorhombic structure based upon the stoichiometric composition $YBa_2Cu_3O_7$ (figure 2.4), whereas the second has a tetragonal structure based upon the composition $YBa_2Cu_3O_6$. The unit cell of orthorhombic $YBa_2Cu_3O_7$ consists of planes of $Cu\text{-}O_2$ separated by a Y layer and intercalated with two Ba-O planes and one plane containing Cu-O. The removal of oxygen from this plane results in the formation of the tetragonal semiconductor $YBa_2Cu_3O_6$. [Jorgensen et al (1990)]

Figure 2.2 — The structure of $Tl_mCa_{n-1}Ba_2Cu_nO_x$

The corresponding bismuth compound has the same structure with thallium replaced by bismuth. Reproduced from Ginsberg (1989).

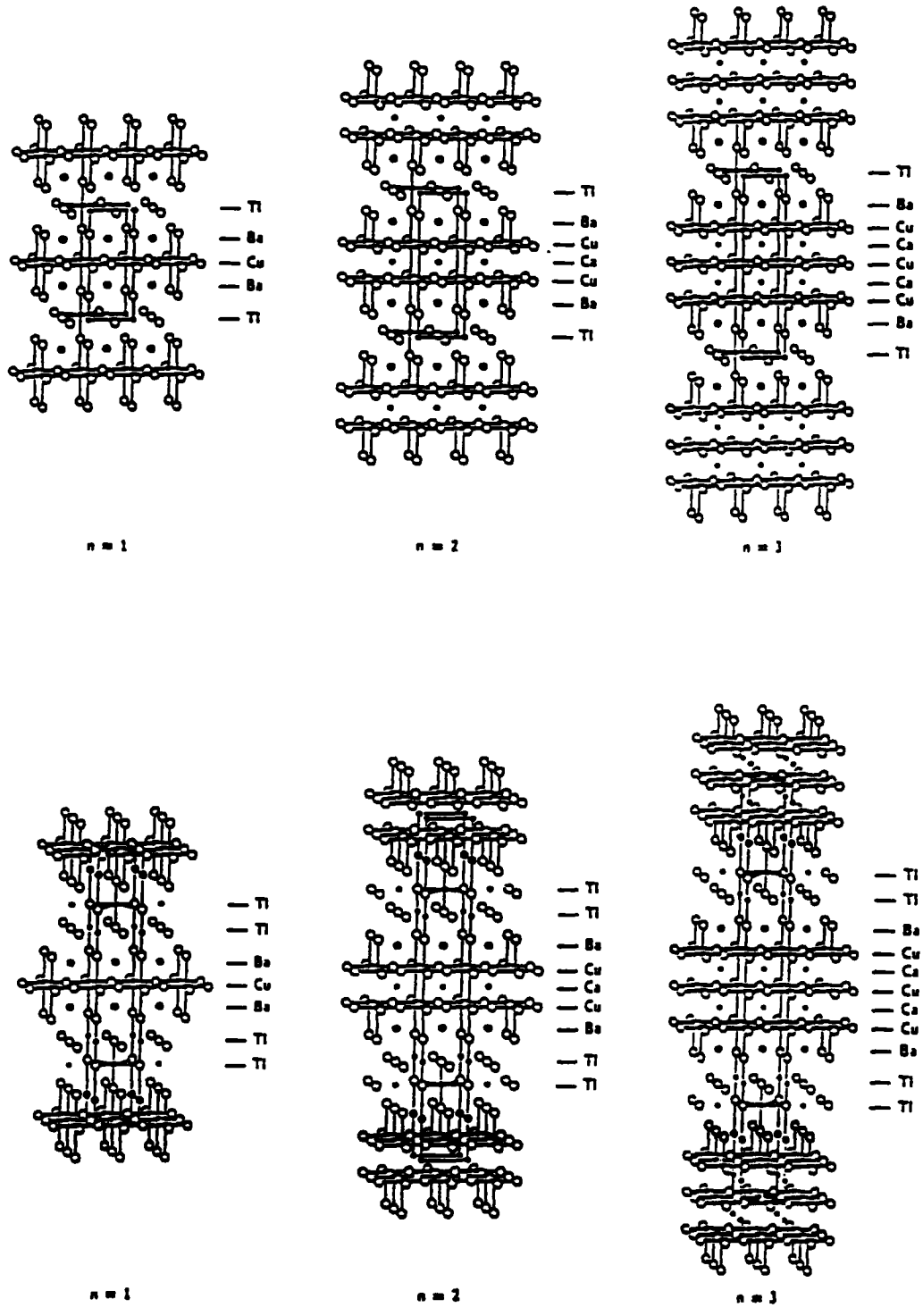


Figure 2.3 — Electrical resistivity of $RBa_2Cu_3O_{7-\delta}$

The electrical resistivity is normalised to its value at 120K (80K for R=La) for $RBa_2Cu_3O_{7-\delta}$ compounds with
(a) R = Y, La, Nd, Sm, Eu and Gd (b) R = Dy, Ho, Er, Tm, Yb and Lu.
Reproduced from Maple et al (1987).

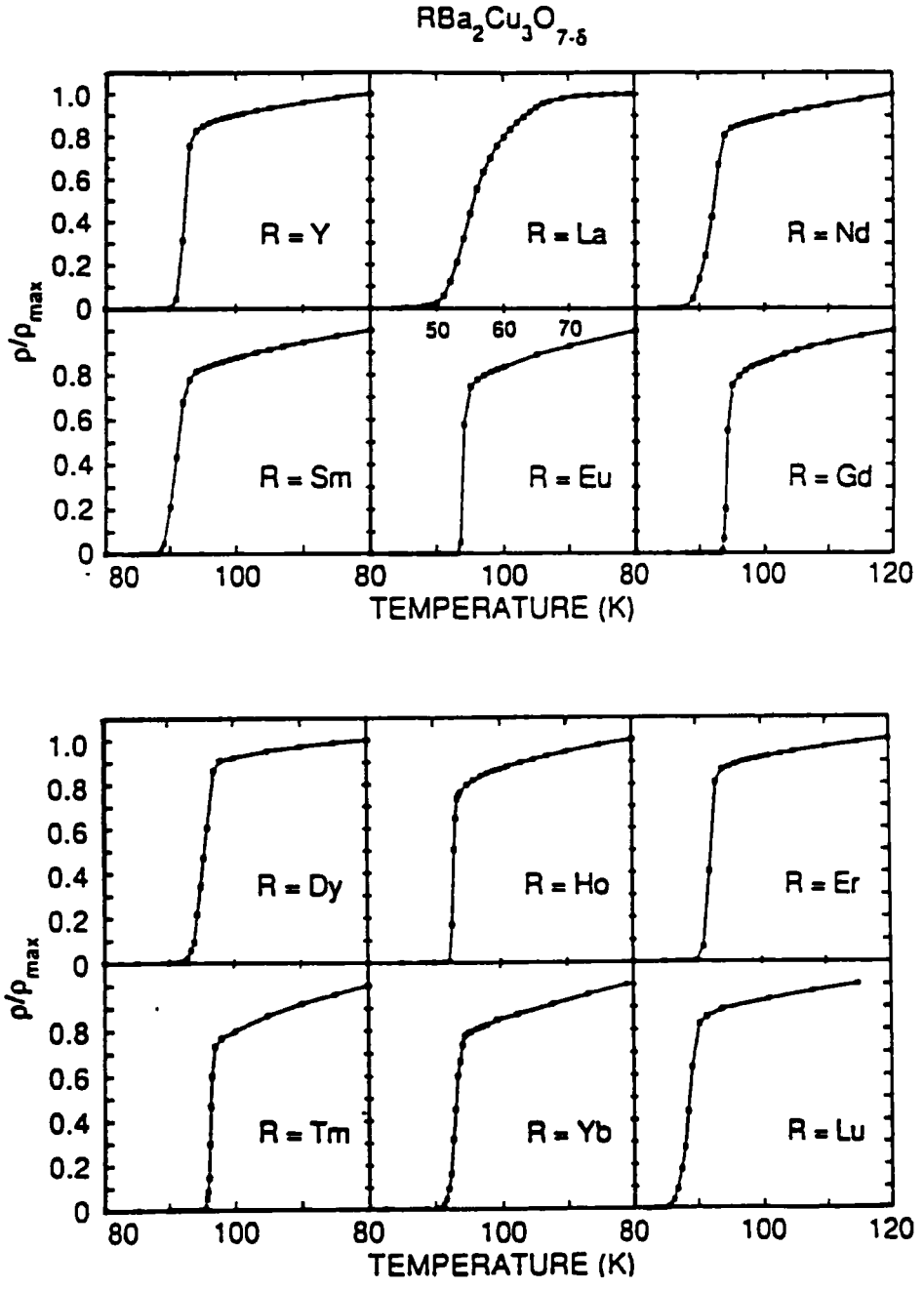
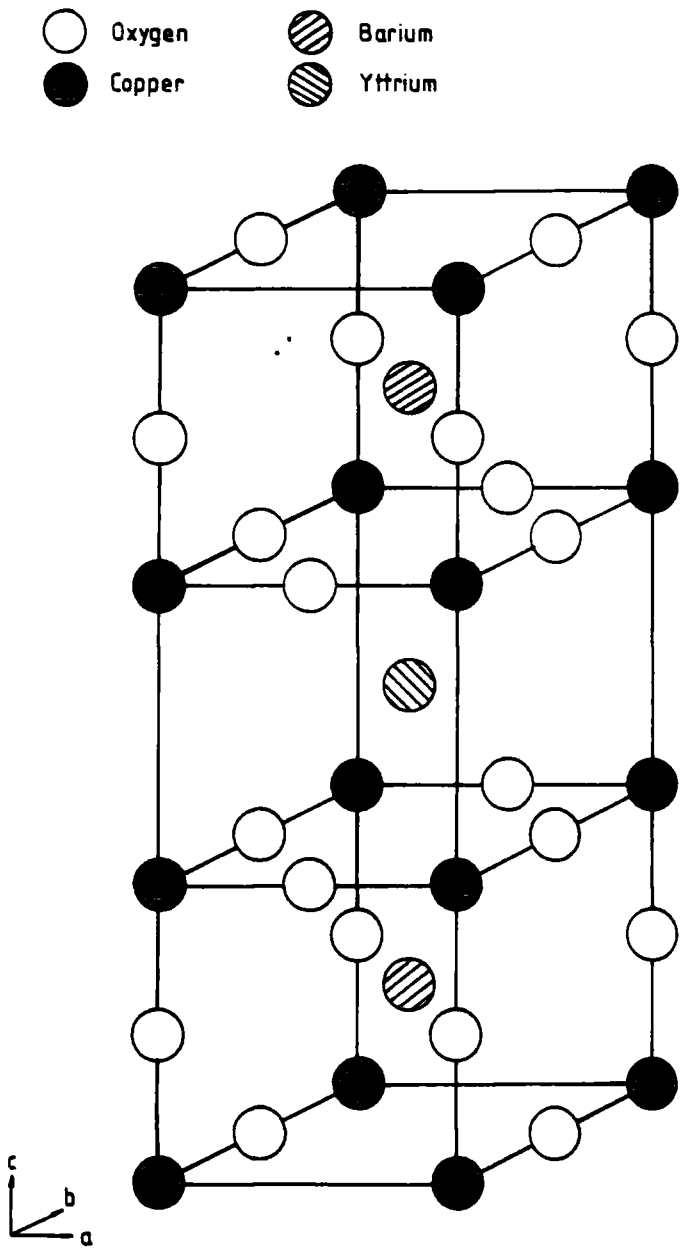


Figure 2.4 — Structure of $YBa_2Cu_3O_7$



2.3.2 Oxygen Content

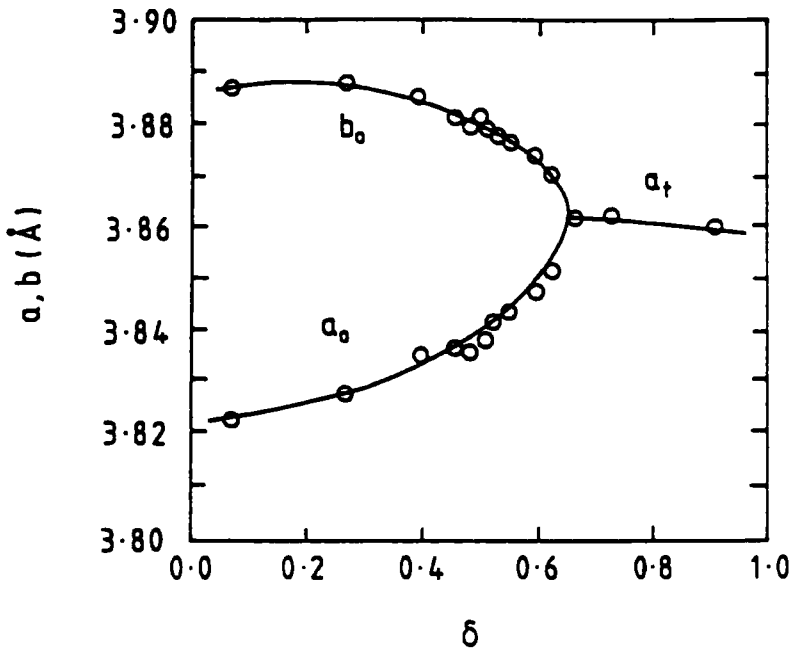
As the orthorhombic phase exhibits superconducting behaviour and the tetragonal phase exhibits semiconducting behaviour, the oxygen vacancy concentration is very important in the $YBa_2Cu_3O_{7-\delta}$ system. The lattice parameters of the unit cell depend upon the oxygen vacancy concentration, with the c axis parameter increasing as the oxygen concentration decreases (figure 2.5). Cava et al (1987) and Beyers et al (1987) have measured the effect of oxygen concentration upon the superconducting transition temperature and have shown that the $\delta = 0$ material has a T_c of $\simeq$ 92K, which decreases by a small amount as δ increases to $\simeq 0.2$. As the oxygen concentration is decreased from $0.2 \leq \delta \leq 0.35$ there is a sharp drop in T_c followed by a plateau region with $T_c \simeq$ 60K for $0.35 < \delta < 0.45$. This is illustrated in figure 2.6. T_c then falls rapidly, reaching 0K at $\delta \simeq 0.65$.

The removal of oxygen from the material is dependent upon the temperature of the material and the partial pressure of oxygen above it. If the oxygen partial pressure is sufficiently high, the tetragonal phase transforms into the orthorhombic phase upon cooling. Jorgensen et al (1990) have shown that on lowering the partial pressure of oxygen from 1013mbar to 0.1mbar the temperature at which the orthorhombic to tetragonal transformation occurs is reduced from 740°C to 500°C. This has serious implications for production of materials for applications which require treatment by conventional semiconductor device processing. Tu et al (1989) have shown that the oxygen diffusion rate is extremely slow, $\simeq 0.035 cm^2 sec^{-1}$, hence it is difficult to ensure that the bulk material has a homogeneous oxygen content.

2.3.3 Twinning

The unit cell of the $YBa_2Cu_3O_{7-\delta}$ system has lattice parameters with the relationship $a \simeq b$ and $c \simeq 3a$. As the $a = b$ symmetry is lost by the tetragonal to orthorhombic transition, some regions form a state where $a > b$, others into states where $a < b$, the change in lattice parameter in each case is only very small resulting in very little energy cost being involved in production of a twinned structure. This can have an appreciable influence upon the superconducting properties of a specimen of $YBa_2Cu_3O_{7-\delta}$.

Figure 2.5 — Lattice parameters of $YBa_2Cu_3O_{7-\delta}$ as a function of oxygen content. Reproduced from Jorgensen et al (1990).



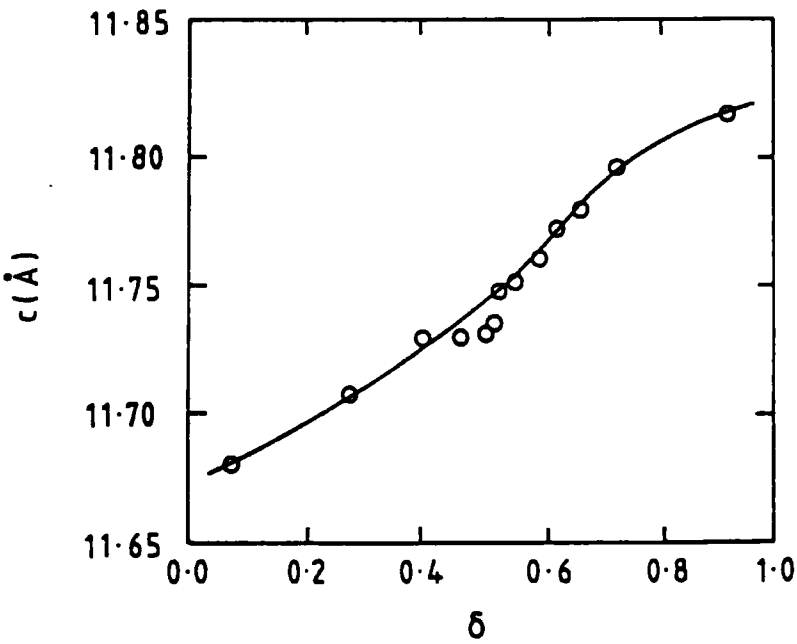
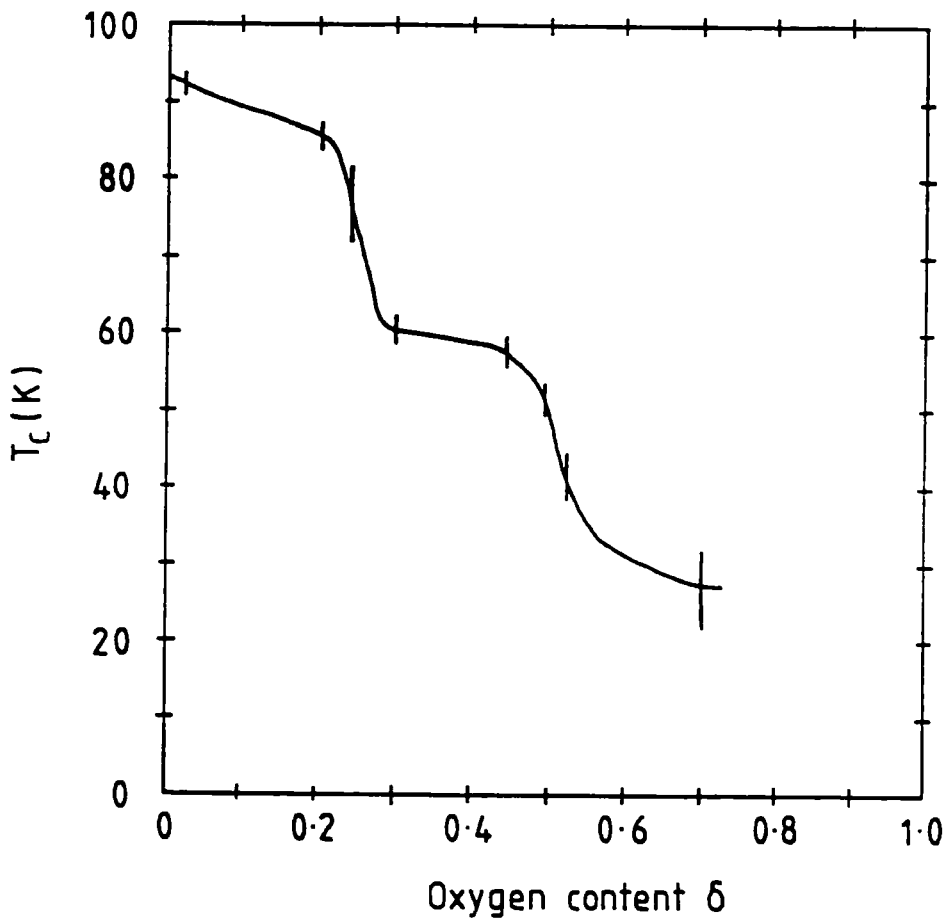


Figure 2.6 — T_c of $YBa_2Cu_3O_{7-\delta}$ as a function of oxygen content. Reproduced from Cava et al (1987).



2.3.4 Preparation of Single Crystal $YBa_2Cu_3O_{7-\delta}$

Single crystals of $YBa_2Cu_3O_{7-\delta}$ have been grown by a variety of methods [Kaiser et al (1987), Müller et al (1988), Takeya et al (1988), Taylor et al (1987), Wanklyn et al (1988)] although to date none of the methods have led to the production of large (10x10x10mm^3) homogeneous single crystals. A popular method is that of crystal growth using a flux such as KCl or CuO. Crystals are produced by this method by heating the mixture of $YBa_2Cu_3O_{7-\delta}$ and the flux to a temperature above the melting point of $YBa_2Cu_3O_{7-\delta}$. The melt is slowly cooled, at typically 3°C/hr, whereupon crystal growth takes place, aided by the flux. The production of crystals with the correct stoichiometry and with low impurity concentrations is dependent upon the initial stoichiometry, quantity and type of flux, and a certain amount of good fortune. Once cool, the crystals must be carefully cut out from the surrounding material.

Even crystals which have the correct $YBa_2Cu_3O_{7-\delta}$ stoichiometry and low impurity concentrations suffer greatly from the problems connected with the oxygen content. The as-grown crystals almost invariably do not have the $YBa_2Cu_3O_7$ composition and require an additional oxygen treatment. Owing to the low oxygen diffusion rate, large crystals of close to theoretical density require an impractically long time for oxygen to diffuse throughout the sample. This problem may be partially overcome by using crystals which have one dimension which is very small, of the order of a few μm. [Salama et al (1989)] Whilst this is satisfactory for some measurements, for a great many experiments, larger crystals are required.

2.4 Preparation of Polycrystalline $YBa_2Cu_3O_{7-\delta}$

Polycrystalline $YBa_2Cu_3O_{7-\delta}$ may be produced by a variety of routes, ranging from precipitation techniques to solid state reaction of precursor compounds of yttrium, barium and copper. As $YBa_2Cu_3O_{7-\delta}$ is a quaternary material the phase diagram is complex, this exacerbates the problem of achieving the correct stoichiometric ratio of the elements throughout the entire material; local regions which contain too little copper result in formation of a tetragonal, green-coloured, semiconducting phase Y_2BaCuO_5; if a region is barium rich or yttrium deficient then phases such as $BaCuO_2$ or Ba_2CuO_3 may form. A close check upon the stoichiometry of the material is highly desirable.

One of the most straightforward preparation routes is that of solid state reaction of precursor compounds of the metallic elements. These are often yttrium oxide Y_2O_3, barium carbonate $BaCO_3$ and copper (II) oxide CuO. The precursor materials are ground together to produce a homogeneous mixture, loaded into a crucible (typically alumina) and calcined. In order to minimise the contamination from the crucible material, the crucible may be filled with some existing $YBa_2Cu_3O_{7-\delta}$ and heated to the melting point of the $YBa_2Cu_3O_{7-\delta}$, thereby leaving a strongly bonded coating of $YBa_2Cu_3O_{7-\delta}$ on the crucible surface. The calcination process is typically performed for 24 hours at a temperature above 900°C but less than 1010°C at which point any $YBa_2Cu_3O_{7-\delta}$ formed will start to undergo incongruent melting. This temperature ensures that the compounds fully react together and that the $BaCO_3$ is fully decomposed to BaO and CO_2.

After calcination the material is powdered and pressed into the desired shape, at this stage it consists of very poorly connected crystallites. In order to improve the links between the individual crystallites, the compacted material must undergo a sintering process. The sintering takes place at a high temperature. At approximately 920°C any $BaCuO_2$ present will melt, this could result in the grains being coated with $BaCuO_2$ which will inhibit the coherence of the superconducting order parameter between grains, as described in §1.9. Once the sintering process is complete, the material will probably be of low oxygen content. The final processing stage is therefore to add oxygen to the system. This is achieved by heating the sintered material to approximately 450°C and passing flowing oxygen over the samples for between 24 and 72 hours.

2.5 Physical Properties of $YBa_2Cu_3O_{7-\delta}$

2.5.1 Isotope Effect

The possible existence of an isotope effect in $YBa_2Cu_3O_{7-\delta}$ has been investigated by Batlogg et al (1987) and Bourne et al (1987). Such experiments, in common with all others made upon the $YBa_2Cu_3O_{7-\delta}$ system, are critically dependent upon the preparation route used for the samples as care must be taken to ensure that the results are not masked simply by differences in oxygen concentration. The samples used by Batlogg et al and Bourne et al were produced by

diffusively replacing the ^{16}O with ^{18}O. Their data suggests that no change in T_c is observed, and that $\alpha = 0 \pm 0.02$. The results of Leary et al (1987) and Morris et al (1988) disagree slightly with this result, suggesting α has a small positive value. From these measurements it follows that if there is a phonon mediated interaction, it is dominated by an electron or other non-phonon interaction and as stated in §1.7 it does not necessarily mean that the idea of a BCS type paired electron state is not applicable in $YBa_2Cu_3O_{7-\delta}$.

2.5.2 Anisotropy

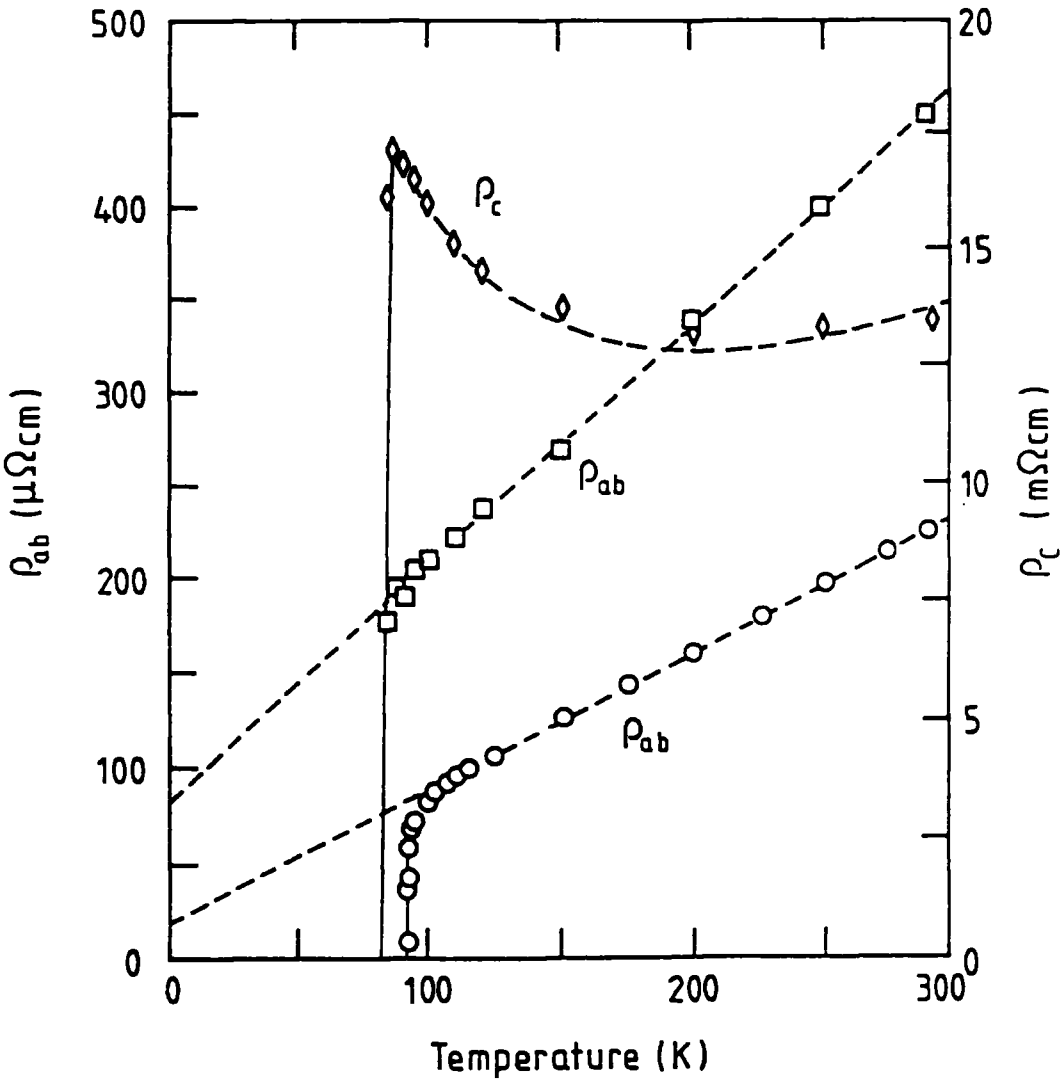
The layered structure of $YBa_2Cu_3O_{7-\delta}$ leads to the material exhibiting highly anisotropic behaviour, the a and b axis properties are similar due to the orthorhombic structure of the unit cell; the c axis, being perpendicular to the Cu-O_2 planes has substantially different properties. Anisotropy in the electrical resistivity, ρ, has been observed by a variety of groups (figure 2.7). [Tozer et al (1987), Konczykowski et al (1988), Vinnikov et al (1988)] The resistivity along the a and b axes shows metallic behaviour down to T_c with a magnitude of approximately 0.20mΩcm which is similar to the electrical resistivity obtained in good polycrystalline samples. Along the c axis, the resistivity is some 2 orders of magnitude higher than ρ_{ab} and shows evidence of semiconducting behaviour - the resistivity increases as the temperature falls, in a manner rather akin to the resistivity behaviour of polycrystalline samples with a poor oxygen content.

2.5.3 B_{c2} and the Coherence Length

With the determination of any physical quantity it is important to be clear which property of the material is being probed by the experiment and how that probe may affect the result. In §1.9 it was stated that the critical current density may be determined in a variety of ways- each of which involves a slightly different definition of J_c; the point at which the quasi-particles are depaired; the onset of flux-flow resistivity and the current at which the self-field exceeds the critical magnetic field. Similarly there are a variety of *practical* definitions of $B_{c2}(T)$ each of which may be obtained by experimental observation.

The value of $B_{c2}(T)$ may be evaluated directly by applying a small probing current to a sample and detecting the onset of a voltage drop across the ma-

Figure 2.7 — Electrical resistivity of single crystal $YBa_2Cu_3O_{7-\delta}$. The two $\rho_{ab}(T)$ curves are from different crystals. The dashed curve through the ρ_c data is a fit to A/T + BT. Reproduced from Penney et al (1988).



terial. In this case the resistance and voltage criterion used will substantially affect the results. This particular problem is compounded further by noting that $YBa_2Cu_3O_{7-\delta}$ is a granular superconductor which results in a broadening of the transition as magnetic field is applied to the sample. B_{c2} may also be defined as the point at which the magnetisation curve exhibits reversible behaviour, corresponding to the onset of flux-flow resistivity, although it should be noted that at this point the material may still exhibit a strong diamagnetic signal. Magnetisation methods again suffer from problem of the broadening of the transition of granular superconductors. For materials with a high T_c, flux creep caused by thermal depinning may also affect the results.

A variety of research workers have determined B_{c2} for $YBa_2Cu_3O_{7-\delta}$ using a range of measurement techniques upon polycrystalline and single crystal specimens. Table 2.2 shows the values of dB_{c2}/dT near T_c. The value of $B_{c2}(0)$ may be obtained from this data by extrapolation of dB_{c2}/dT near T_c using the WHH formula in the 'dirty' ($\lambda/\xi \gg 1$) limit with no paramagnetic limiting. [Wertharmer et al (1966), Hake (1967), Collings (1986)]

$$B_{c2}(0) = -0.68T_c\frac{dB_{c2}}{dT} \tag{2.1}$$

The values of $B_{c2}(0)$ obtained are distributed over a large range of values (table 2.3). The data indicates that $B_{c2}(0)$ is extremely large and shows a marked anisotropy consistent with the layered structure of the material. The physical meaning of the higher values obtained is not at all clear owing to the omission of the effect of the paramagnetic susceptibility which would be significant at very high fields.

Representative values for the coherence length may be determined from $B_{c2}(0)$ through the mathematical relationship

$$B_{c2} = \frac{\Phi_0}{2\pi\xi^2} \tag{2.2}$$

The high values of $B_{c2}(0)$ lead to the result that the coherence length in the $YBa_2Cu_3O_{7-\delta}$ system is particularly small, ~30Å perpendicular to c and ~5Å parallel to the c axis. These values are comparable to the magnitude of the lattice parameters hence twin boundaries, dislocations or small impurities may substantially

Table 2.2 — $-dB_{c2}/dT$ at T_c in Tesla/Kelvin.

Group	Method	$B\perp_c$	$B\|_c$	B_{av}
Ayache (1987)	Magnetic			2
Gallagher (1987)	Magnetic	3.8	0.54	
Iye (1987)	Resistance	3	0.9	
Muto (1990)	Resistance			2.4
Takita (1987)	Resistance	3.8	2.6	
Takabatake (1987)	Resistance			1.27
Welp (1989)	Magnetic	10	1.8	

Table 2.3 — $B_{c2}(0)$ in Tesla.

Group	Method	$B\perp_c$	$B\|_c$	B_{av}
Ayache (1987)	Magnetic			120
Gallagher (1987)	Magnetic	240	34	
Iye (1987)	Resistance	190	56	
Muto (1990)	Resistance			150
Nakao (1988)	Magnetic	110 ± 10	40 ± 5	
Takabatake (1987)	Resistance			80
Takita (1987)	Resistance	240	160	
Welp (1989)	Magnetic	620	110	

affect the bulk superconducting properties of a given specimen of $YBa_2Cu_3O_{7-\delta}$. The coherence length may also be obtained from the observation of fluctuation effects described in §1.8, typically by observation of electronic contribution to the specific heat capacity. The value of $\xi(0)$ estimated in this manner is comparable to the values obtained by magnetic methods. Typical values obtained for $\xi(0)$ are given in table 2.4.

Table 2.4 — $\xi(0)$ in Å.

Group	Method	$\xi \perp_c$	$\xi \parallel_c$	ξ_{av}
Fruchter (1988)	Magnetic	25 ±5	6.4 ±2	
Gallagher (1988)	Magnetic	31	4.3	
Inderhees (1988)	Heat capacity			7 ± 5
Iye (1987)	Resistance	24	7.0	
Oh (1988)	Resistance	16 ± 2	2.2 ± 1	
Muto (1990)	Resistance			15
Umezawa (1988a)	Magnetic	31	4.8	
		22	6.7	

2.5.4 B_{c1} and the Magnetic Penetration Depth

A large number of measurements of B_{c1} for $YBa_2Cu_3O_{7-\delta}$ have been made on polycrystalline specimens and single crystals using a variety of techniques. B_{c1} may be assessed by analysing the initial magnetisation curve and detecting the first deviation from a linear region. This particular method, whilst in principle easy to perform, is particularly susceptible to the problem of determining an appropriate criterion for the point at which the curve deviates from linear. A second method of determining B_{c1} is by analysis of the difference between field-cooled and zero-field-cooled magnetisation data. A precise measurement of B_{c1} may still be difficult to obtain, particularly in samples which are inhomogeneous. Fruchter et al (1988) have used the onset of irreversibility of torque magnetometry signals to define B_{c1}, although this method is prone to the same errors as other magnetic techniques.

Table 2.5 gives representative values of $B_{c1}(0)$ for $YBa_2Cu_3O_{7-\delta}$ which, like $B_{c2}(0)$, show a large spread of values. In contradistinction to the values of B_{c2} the values of B_{c1} are small, ~20mT perpendicular to the c axis and ~100mT parallel to the c axis, indicating that $YBa_2Cu_3O_{7-\delta}$ is an extreme type II superconductor.

Table 2.5 — $B_{c1}(0)$ in milliTesla.

Group	Method	$B\perp_c$	$B\|\|_c$	B_{av}
Fruchter (1988)	Torque curve	24 ± 3	110 ± 10	
Gallagher (1987)	M deviation	< 50	500	
Krusin Elbaum (1989a)	Direct magnetic	18	53	
McGuire (1987)	M deviation	20	400	40
Sridar (1989)	Microwave	25		
Umezawa (1988b)	M deviation	8 ± 1	37.5 ± 0.5	
Yeshurun (1988)	M relaxation	25 ± 5	90 ± 10	

The value of the magnetic penetration depth may be calculated from the values of B_{c1}, B_{c2} and ξ using the relationship

$$B_{c1} = \frac{\Phi_0}{\lambda^2} \ln\left(\frac{\lambda}{\xi}\right) \tag{2.3}$$

From this relationship the magnetic penetration depth of $YBa_2Cu_3O_{7-\delta}$ is found to be of the order of a few thousand angstroms, significantly larger than the coherence length. Direct measurement of the magnetic penetration depth may be made by using the technique of muon spin relaxation. With this technique, the sample is bombarded by a beam of polarised muons. The muons decay by emission of a positron which is emitted preferentially along the polarisation axis. By measurement of the angle of emission of the positron, the local magnetic field may be determined. In practice, a statistical average is taken and the muon depolarisation rate is related to the penetration depth by a theoretical model; it is the validity of this model which determines the validity of the data. Unlike other magnetic measurements, muon spin relaxation has the advantage of probing the local magnetic field of the material, hence voids in the material do not influence the results. Table 2.6 indicates the range of values obtained for $\lambda(0)$.

2.5.5 Fluctuation Effects

A representative value for the coherence length in high T_c materials is about 10Å, it follows that the superconductivity in high T_c materials will be much more

Table 2.6 — $\lambda(0)$ in Å.

Group	Method	$\lambda \perp_c$	$\lambda \parallel_c$	λ_{av}
Forgan (1990)	Muon spin relaxation	12000	2100	5020
Fruchter (1988)	Indirect magnetic	5000	680	
Gallagher (1987)	Indirect magnetic	1800	270	
Harshman (1987)	Muon spin relaxation			1400
Krusin-Elbaum (1989b)	Direct magnetic	4200	1400	
Monod (1988)	Indirect magnetic			1900
Sridar (1989)	Microwave	1400		
Umezawa (1988b)	Indirect magnetic	8400	900	
Uemura (1988)	Muon spin relaxation			1656
Yeshurun (1988)	Indirect magnetic	4600	1000	

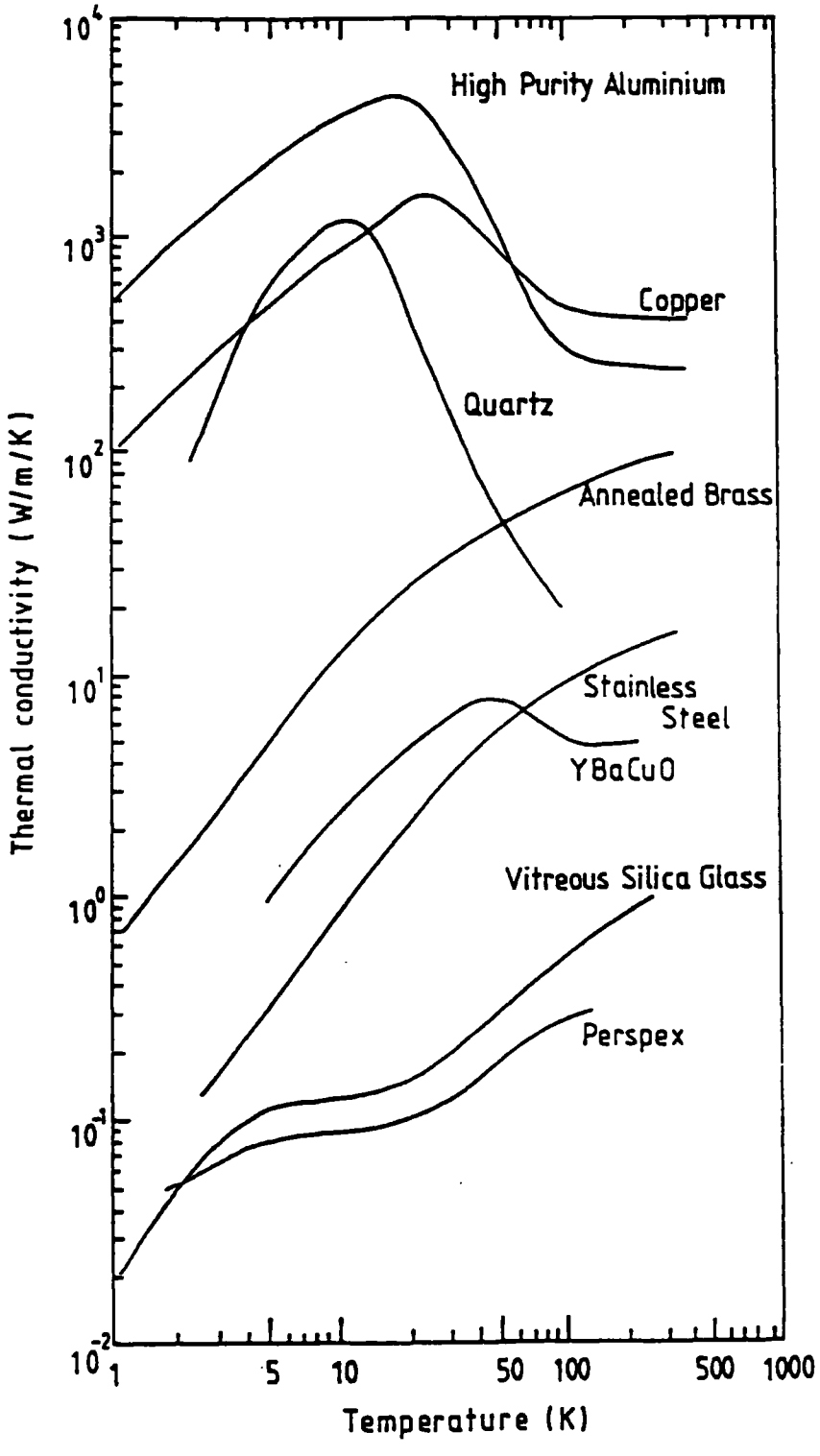
sensitive to small scale structural or chemical imperfections than conventional superconductors which have coherence lengths in the range 10^2Å to 10^4Å. These effects have been observed by Lægreid et al (1989) when performing high resolution heat capacity experiments in the region of T_c. The effects have also been observed in the electrical conductivity of $YBa_2Cu_3O_{7-\delta}$ by Vidal et al (1988), Gordon et al (1989) and Lægreid et al (1987).

2.6 Thermal Conductivity of $YBa_2Cu_3O_{7-\delta}$

In sharp contrast to many other physical parameters the database of the thermal conductivity of $YBa_2Cu_3O_{7-\delta}$ is rather small. This is in part due to the relative complexity of the measurement process and partly the requirement for large samples of well defined geometry. In particular the number of published measurements made upon single crystals is minimal.

Figure 2.8 illustrates the thermal conductivity of $YBa_2Cu_3O_{7-\delta}$ relative to a variety of materials at low temperatures. The magnitude of the thermal conductivity in the normal state is comparable to that of a metal with a poor thermal conductivity such as stainless steel. This is roughly consistent with the picture pre-

Figure 2.8 — The thermal conductivity of a range of materials
Reproduced from White (1979). $YBa_2Cu_3O_{7-\delta}$ data taken from Jeżowski (1989). Perspex data taken from Burgess and Greig (1975).



sented by the electrical resistivity of $YBa_2Cu_3O_{7-\delta}$ which exhibits strong metallic behaviour in its normal state temperature dependence but which has a magnitude which suggests that it is a poor metal.

Early thermal conductivity measurements made upon $YBa_2Cu_3O_{7-\delta}$ disagreed strongly about the absolute magnitude of the thermal conductivity in the normal state, just above T_c (figure 2.9). The data of Jeżowski et al (1987) placed the thermal conductivity at around $5Wm^{-1}K^{-1}$ at 100K, with the thermal conductivity decreasing slightly as the temperature is increased. Uher et al (1987) obtained a thermal conductivity of $3.5Wm^{-1}K^{-1}$ at 100K with the thermal conductivity increasing slightly with temperature. The data of Morelli et al (1987) revealed a significantly smaller thermal conductivity, $0.5Wm^{-1}K^{-1}$ at 100K; remaining roughly constant with increasing temperature! This latter data was obtained upon samples with a high electrical resistivity, $10m\Omega cm$ at 100K, and it is therefore probable that the sample suffered from a high porosity or poor inter-grain coupling.

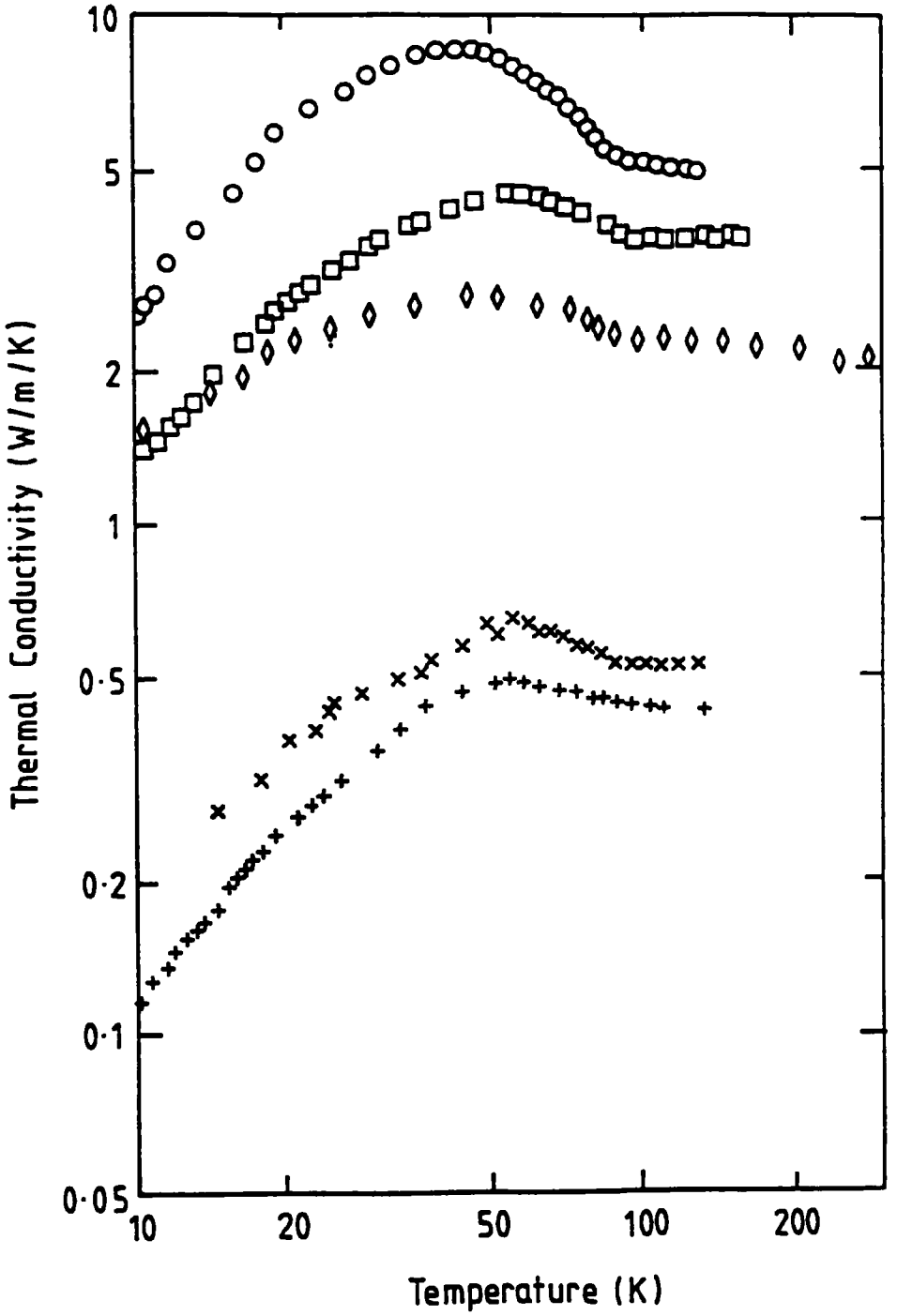
2.6.1 Porosity

The porosity of a sample influences the thermal conductivity in two ways. Firstly the voids in the material act as scattering centres, this enhances the defect-carrier and defect-phonon scattering which tend to lower the thermal conductivity. Secondly, in a material with a high porosity (and hence of low relative density) a large portion of the apparent sample volume will consist of voids which act as barriers to the heat flow. In this instance the thermal conductivity calculated will be smaller than the 'true' value as the cross sectional area of the sample used in the calculation will be too large. This effect will manifest itself at all temperatures and, assuming the volume expansivity is small, will reduce the measured thermal conductivity by a constant proportion throughout the entire temperature range. The data of Heremans et al (1988) places the thermal conductivity of $YBa_2Cu_3O_{7-\delta}$ again at approximately $0.5Wm^{-1}K^{-1}$. However, the density of their sample was only 43% of the theoretical single crystal value, in contrast to the 80% of theoretical density of the samples of Jeżowski et al.

Figure 2.9 — Early thermal conductivity measurements of $YBa_2Cu_3O_{7-\delta}$

o ... Jeżowski (1987) ▫ ... Uher (1987) ◇ ... Bayot (1987)

× ... Morelli (1987) + ... Heremans (1988)



2.6.2 Thermal Conductivity Maximum

All of the thermal conductivity data obtained upon superconducting samples of $YBa_2Cu_3O_{7-\delta}$ exhibits an increase upon cooling through the superconducting phase transition.[1] This increase in thermal conductivity continues as the sample is cooled until the thermal conductivity reaches a maximum at some temperature, T_{max}, before falling away. From figure 2.9, it can be seen that there is no universal value for T_{max}, some materials having the peak at approximately 60K, others at approximately 40K, the precise value depending upon the interplay between the scattering mechanisms.

Jeżowski et al (1989) have measured the thermal conductivity of three separate samples of $YBa_2Cu_3O_{7-\delta}$, each of different oxygen content. Their data suggests that the absolute magnitude of the thermal conductivity in the normal state decreases as the oxygen content of the samples lowered. Unambiguous interpretation of this data is not possible because the data was obtained upon separate samples; the samples may all have an appreciably different microstructure which could influence the results. This problem afflicts all but a small number of the thermal conductivity studies of $YBa_2Cu_3O_{7-\delta}$ to date.

2.6.3 Interpretation of Data

The peak in the thermal conductivity below T_c has been attributed, by a number of research groups [Fischer et al (1988), Uher (1989)] to a decrease in the phonon-carrier scattering below T_c. To understand this picture one needs to understand the various processes which contribute to the thermal conductivity. In a conventional picture there is a contribution to the thermal conductivity from the phonon-phonon interactions, phonon-carrier interactions and carrier-carrier interactions. For the purposes of this discussion phonon-defect scattering is considered as a separate interaction although it is a particular case of phonon-phonon scattering arising from the non-ideal properties of the material. Many research groups have used the Wiedemann-Franz law to estimate the contribution to the thermal

[1] There is one exception to this. Uher et al (1988) drastically increased the number of point defects in their samples by irradiating them with neutrons and found that the peak became 'washed out' at high neutron fluxes.

conductivity from the charge carriers. This treats the carriers as a system of independent particles and relates the electrical resistivity, ρ, to the carrier thermal conductivity, κ_c.

$$\kappa_c = \frac{L_0 T}{\rho} \tag{2.4}$$

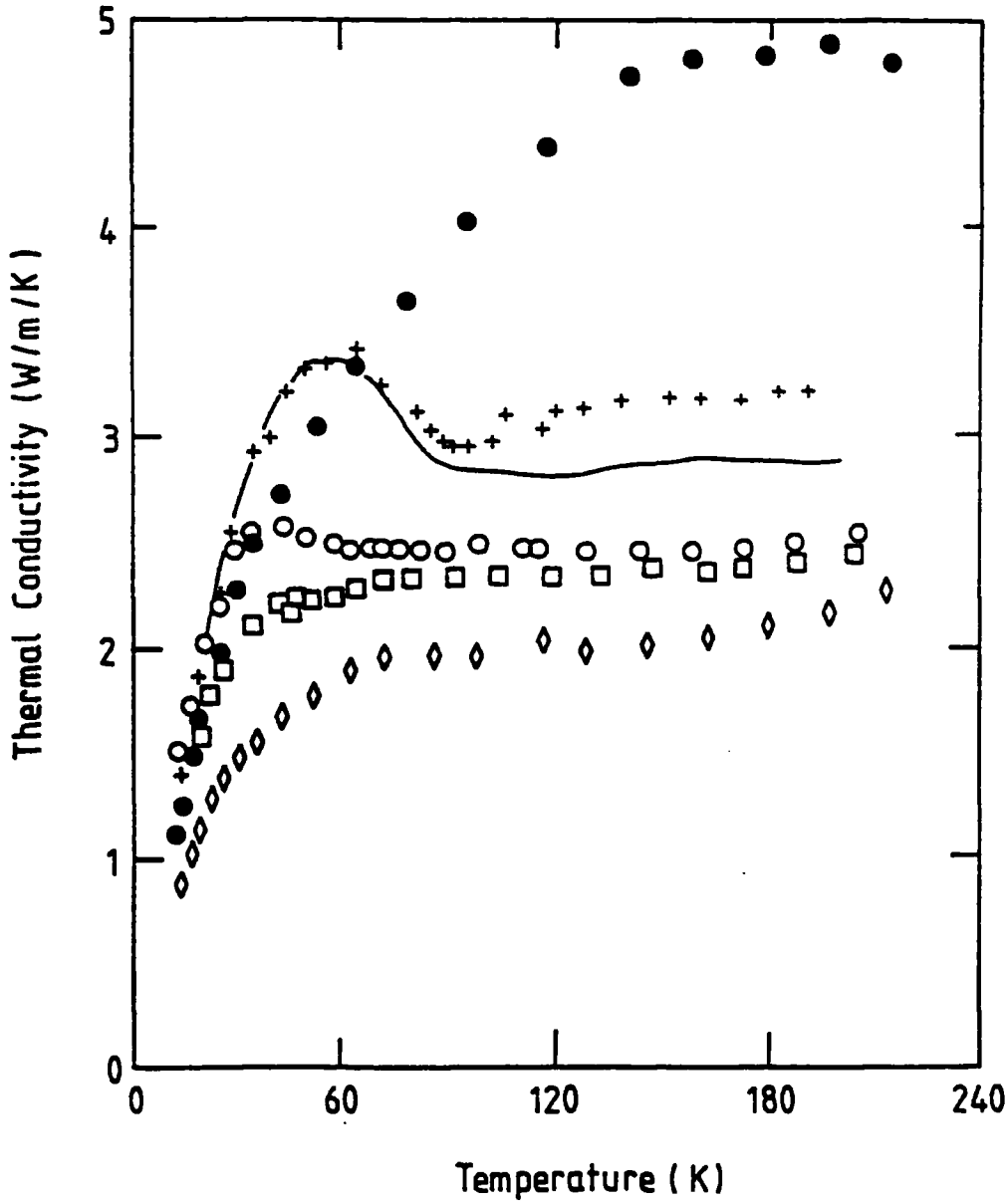
L_0 (1.11×10^{-8} WΩ K^{-2}) is the Lorenz constant. From this relationship, they estimate that the value of κ_c is small and hence that the carriers themselves do not play a major role in the heat conduction mechanism in the normal state. If this is the case, the thermal conductivity will be dominated by the phonon transport mechanisms; the carrier-phonon, phonon-phonon and defect-phonon scattering.

Below T_c it is assumed that the charge carriers are in a condensed state. In this state they carry no entropy and are unable to scatter phonons unless the phonon energy is sufficient to overcome the condensation energy. Thus below the transition temperature the thermal conductivity of the material increases as more and more carriers condense into the superconducting state and the carrier-phonon scattering relaxation time increases. The reduction in carrier-phonon scattering is offset by a reduction in the ability of the phonons to carry heat as the temperature falls. The interplay between the enhancement of the thermal conductivity due to the reduction in carrier-phonon scattering and the reduction in the phonon density as the phonons are 'frozen out' leads to a peak in the thermal conductivity below T_c.

Very few thermal conductivity studies have concentrated upon one particular sample which has undergone a variety of treatments. Zavaritskiĭ et al (1989a, 1989b) have measured the influence of oxygen content upon the same superconducting sample after subjecting it to a number of heat treatments. They found that the normal state thermal conductivity of the sample decreased as the oxygen was removed from the sample, in addition the peak in the thermal conductivity disappeared. However when the sample was heated to 675°C for 24 hours, producing a sample of nominal composition $YBa_2Cu_3O_6$, the normal state thermal conductivity dramatically increased (figure 2.10).

The initial decrease in the thermal conductivity was attributed to an increase in the number of defects in the sample as the oxygen was removed, hence there was

Figure 2.10 — Thermal conductivity as a function of oxygen content.
+ ... $x = 7.00$ ○ ... $x = 6.70$ □ ... $x = 6.53$
◇ ... $x = 6.31$ — ... $x = 6.77$ ● ... $x = 6.00$
Reproduced from Zavaritskiĭ et al (1989a).



a greater increase in the phonon-defect scattering resulting in a lower thermal conductivity. When $YBa_2Cu_3O_6$ was formed, the sample was assumed to have taken on a more ordered form and there is therefore a reduction in the phonon-defect scattering resulting in a higher thermal conductivity. The carrier contribution was assumed to be negligible. However, Zavaritskiĭ et al did not re-oxygenate their sample after forming $YBa_2Cu_3O_6$. It is possible that the high temperature vacuum annealing caused the sample to undergo an irreversible change in its microstructure which led to an increase in the thermal conductivity. This argument is supported by noting that the thermal conductivity of $YBa_2Cu_3O_6$, at low temperatures, is actually lower than that of $YBa_2Cu_3O_7$. As the $YBa_2Cu_3O_7$ material is also well-ordered and at low temperatures is expected to have a low carrier contribution to the thermal conductivity (Cooper pairs carry no entropy) this situation should not arise. The conventional picture of the thermal conductivity of $YBa_2Cu_3O_{7-\delta}$ is therefore left with a considerable difficulty. Part of the work in this thesis attempts to address this issue.

Chapter III

Material Characterisation

Experience is the name everyone gives to their mistakes.
— Oscar Wilde.
'Lady Windermere's Fan'

3.1 X-ray Powder Diffraction

3.1.1 Outline

One method of determining the crystallographic structure and lattice parameters of a crystalline material is the method of powder diffraction. With this method a collimated, monochromatic beam of X-rays, of wavelength λ, is incident upon a finely powdered specimen which has effectively a continuous range of crystallite directions. Peaks in the intensity of the scattered X-rays will occur at the angles which satisfy the Bragg condition.

$$n\lambda = 2d_{hkl}\sin\theta_{hkl} \tag{3.1}$$

The crystallographic structure of $YBa_2Cu_3O_{7-\delta}$ is now well known, consequently by comparing the observed peak positions with the expected peak positions, the peaks may be indexed according to their Miller indices. The lattice parameters of the material may then be calculated to a precision and accuracy which depends critically upon the angular resolution of the detector. An exact determination of the lattice parameters requires the use of much more sophisticated experiments; for example neutron high resolution powder diffraction.

X-ray powder diffraction may also be used to provide some indication of the phase purity of a known material by comparison of the powder diffraction curve with the expected curves for the material and its potential impurity phases. Quantitative analysis of the data may theoretically be possible, although the process is complicated by the influence of the resolution function of the detector, multiple scattering, absorption, statistics and structure factors. In practice, quantitative analysis must be performed by some other method.

3.1.2 Experimental Details

X-ray powder diffraction was performed using the Philips X-ray diffractometer housed in the Geology department at Durham. The experimental arrangement of this instrument is simple in concept and is depicted schematically in figure 3.1. A collimated X-ray beam, produced from a copper target, is incident upon a rotating sample stage whilst a detector rotates around the sample stage and makes an angle of 2θ relative to the incident beam. The output from the detector is amplified and sent to a ratemeter which in turn sends a signal to a chart recorder. The entire process is automated with the detector sweeping through an angle of 1° per minute.

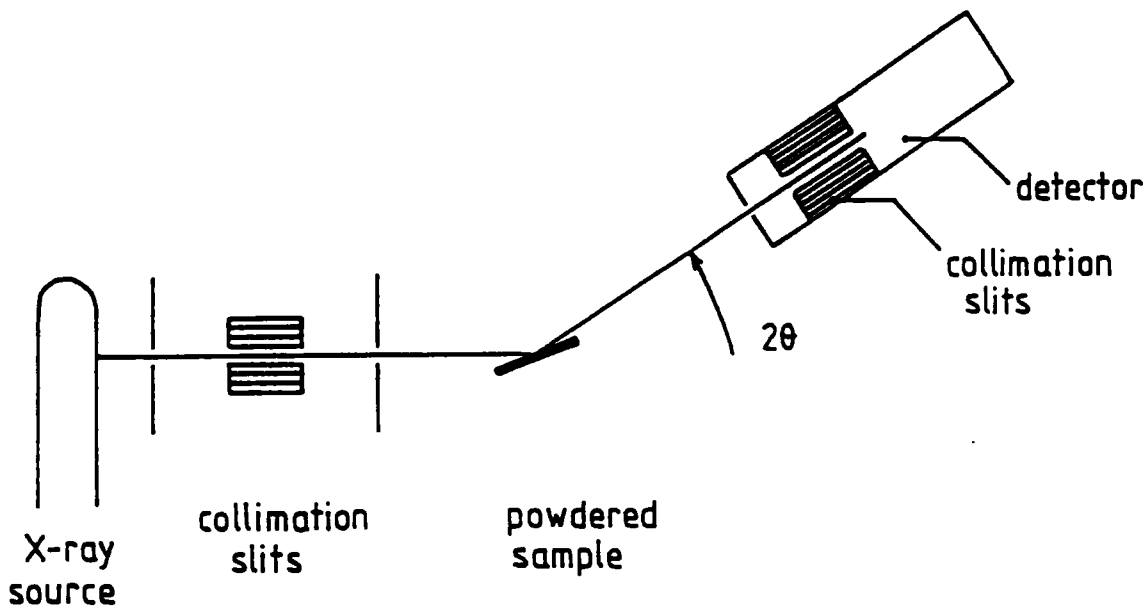
The main diffraction lines for $YBa_2Cu_3O_{7-\delta}$ and the impurity phases $BaCO_3$, Y_2O_3, CuO, Y_2BaCuO_5, Ba_2CuO_3 and $BaCuO_2$ have been calculated by Bell (1990). These values were used to identify the crystallographic reflections corresponding to each diffraction peak. From the observed reflections the lattice parameters of the $YBa_2Cu_3O_{7-\delta}$ phase were calculated using a least squares technique, [Press et al (1988)] the c axis being computed first from the d spacing of the (001) to (007) reflections.

3.2 Electrical Resistivity

3.2.1 Temperature Dependence

The electrical resistivity of a material as a function of temperature is a simple yet powerful characterisation tool. Metallic materials are considered to have electrons in (unfilled) conduction bands even at absolute zero; consequently they tend to have a low electrical resistivity which decreases linearly as the temperature falls. In contrast to this, semiconductors rely upon thermal excitation of charge carriers

Figure 3.1 — X-ray powder diffraction apparatus.



into the conduction band, resulting in an increase in the electrical resistivity as the temperature decreases.

In the previous chapter it was noted that $YBa_2Cu_3O_{7-\delta}$ with a low oxygen content (high value of δ) is a semiconductor. The temperature dependence of the electrical resistivity of randomly oriented, polycrystalline specimens of $YBa_2Cu_3O_{7-\delta}$ serves as a rough screening tool for determining their integrity, any observation of semiconducting behaviour will suggest that a significant quantity of the material is of low oxygen content. The absolute magnitude of the electrical resistivity also serves as an indicator of the connectivity and the oxygen content of the material. A low resistivity may indicate that the grains are well connected or that they have a high overall oxygen content.

3.2.2 Application of a Magnetic Field

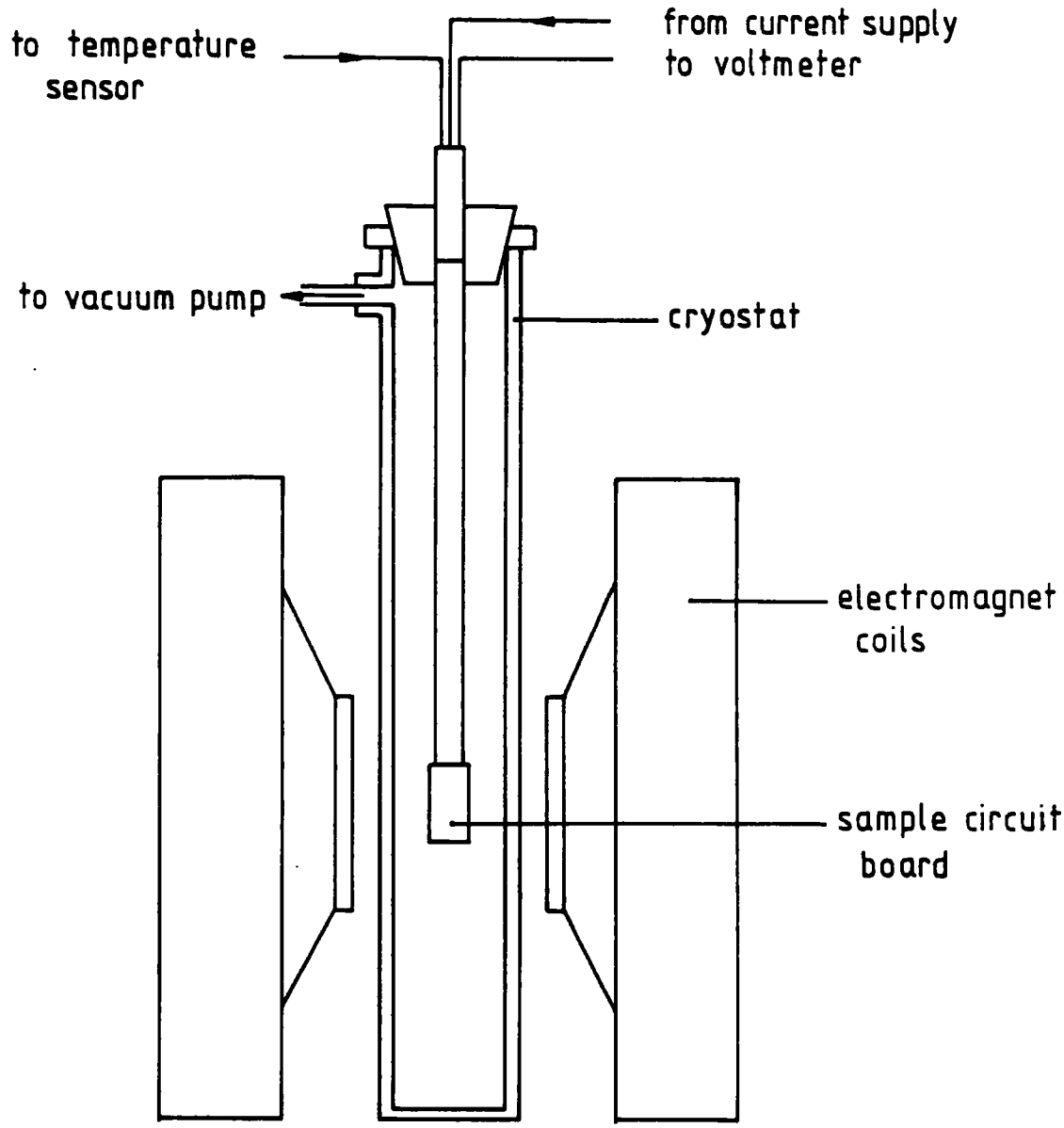
In a polycrystalline sample with very poor inter-grain connections, even very small magnetic fields may exceed the critical magnetic field for the grain boundaries. In this instance the number of inter-granular links which are able to sustain a supercurrent will be reduced as the magnetic field is increased and the observed resistance transition will broaden out. Resistivity measurements as a function of applied magnetic field are therefore a valuable way of determining the strength of the inter-grain coupling.

The zero resistance state is dominated by the percolation threshold – where there is a continuous zero resistance path throughout the sample. In order to counteract the influence of this effect, it is customary to measure the width of the transition by taking the difference between the temperatures at which the resistivity has fallen to 90% and 10% of the normal state value. Similarly T_c is often defined as the temperature at which the resistivity has fallen to 50% of the normal state resistivity.

3.2.3 Experimental Details

The superconducting transition temperature of each sample was determined using a standard four probe resistance technique, thereby eliminating effects due to the contact resistance to the sample. Figure 3.2 shows a schematic of diagram of

Figure 3.2 — Electrical resistivity apparatus.



the experimental arrangement. Bar shaped samples (approximately 3x3x15mm^3) of material were cut from each pellet and mounted onto a small printed circuit board using GE varnish. The circuit board was attached firmly to a small copper block which in turn was mounted upon a stainless steel rod. Electrical contact between the copper tracks on the circuit board and the sample was achieved using electrically conducting silver paint dissolved in heptan-2-one, this gave contact resistances of the order of a few ohms. The current and voltage leads were soldered to the appropriate tracks of the circuit board, shielded and fed continuously through the top plate of the cryostat insert. The temperature of the sample was monitored using a copper-constantan thermocouple. One arm of the thermocouple was mounted adjacent to the sample whilst the second arm was fed through the top plate of the sample insert and immersed in liquid nitrogen. In order to minimise spurious thermal emfs the thermocouple was made from continuous lengths of wire bonded only at the copper-constantan junctions. The thermocouple voltage was read using a Fluke 8840A calibrated microvoltmeter and the sample temperature determined by interpolation of a copper-constantan thermocouple reference table. [Landolt-Börnstein (1984)]

A measuring current of typically 2.8mA was supplied by a constant current power supply (described in §4.3). The current was measured with a calibrated Keithley 175 multimeter and the the voltage drop across the sample, typically a few tens of microvolts, was measured by a Tinsley 6050 nanovoltmeter. Considerable care was taken to ensure that no ground loops existed and that the voltage leads were shielded from electromagnetic interference. No corrections were made to compensate for the small magnetoresistance of $YBa_2Cu_3O_{7-\delta}$. Hall effect voltages were computed out by averaging the signal with the current in both forward and reverse directions. Owing to the limited space available within the cryostat, the magnetic field applied to the sample was always perpendicular the the direction of the current flow. The measuring current was frequently varied to ensure that the voltage drop was directly proportional to the applied current when the sample was in the 'normal' state. The existence of the superconducting state was verified by the absence of a voltage drop across the sample even when the current was increased.

The cryostat was mounted between the pole pieces of an electromagnet and the magnetic field varied by applying a direct current to the elecromagnet coils. A calibrated Bell 610 gaussmeter measured the applied magnetic field. The sample chamber of the cryostat was evacuated with a rotary pump and the sample cooled by radiative heat loss to the cryostat heat exchanger. An Oxford Instruments 3120 temperature controller regulated the temperature of the heat exchanger to ± 0.1K.

A common problem with many experiments which attempt to determine the variation of a physical property as a function of temperature is that there may be thermal hysteresis caused by the sample not being in thermal equilibrium with the temperature sensor. During the course of the electrical resistivity experiments no such hysteresis was observed hence it was assumed that the sample and mounting block were in thermal equilibrium. Figure 3.3 shows a typical set of data obtained close to the superconducting transition. The broadening of the transition as the applied field is increased is clearly visible. Fluctuations in the applied current and voltage drop across the sample were negligible, the largest source of error was the uncertainty in the separation of the voltage wires. The silver paint contacts were approximately 1mm in diameter, this serves as an upper bound for the uncertainty in the voltage wire separation. The uncertainty in the absolute resistivity was approximately 10%.

3.3 Inductance Probe

3.3.1 Outline

When a material undergoes a transition to the superconducting state there is a marked change in the response of the material to an applied magnetic field (See §1.2), this may be used to determine the transition temperature of a superconducting material. Consider a coil of wire which surrounds a specimen of superconducting material above T_c. The magnetic flux, Φ, through the coil is given by the expression

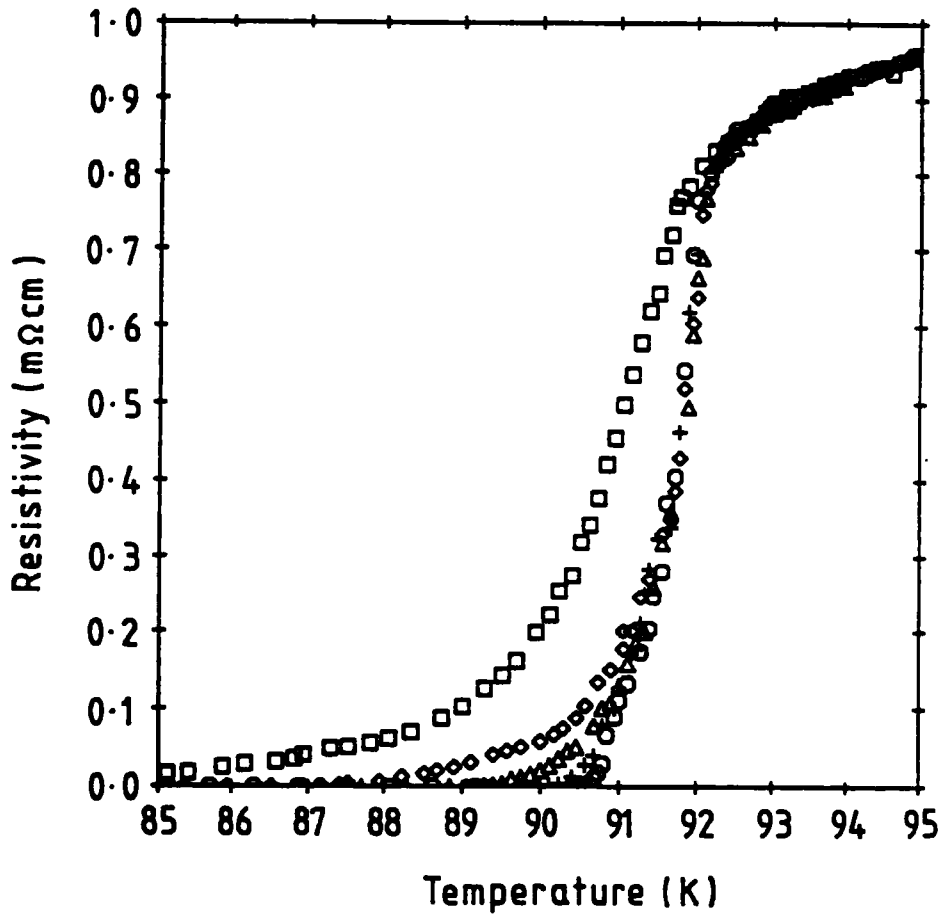
$$\Phi = B.A = L.I \tag{3.2}$$

where B is the applied magnetic field, A is the cross sectional area intersected by the B field, L is the inductance of the coil and I the current applied to the coil. Upon passing into the superconducting state, the material expels magnetic

Figure 3.3 — Electrical resistivity transition of sample D980.

o ... <0.1mT + ... 0.5mT △ ... 5mT

◇ ... 50mT ▫ ... 500mT



flux from the sample and screens it from further applied field, the effective cross-sectional of the coil is reduced and the inductance of the coil decreases. It is important to note that with this technique the signal is dominated by screening currents *not* flux expulsion, so it does not follow that the volume fraction of superconducting material may be determined by this method. The drop in inductance does not, by itself, provide conclusive proof of the existence of a superconducting state – merely that there is some change in the response of the material to applied magnetic fields.

3.3.2 Experimental Details

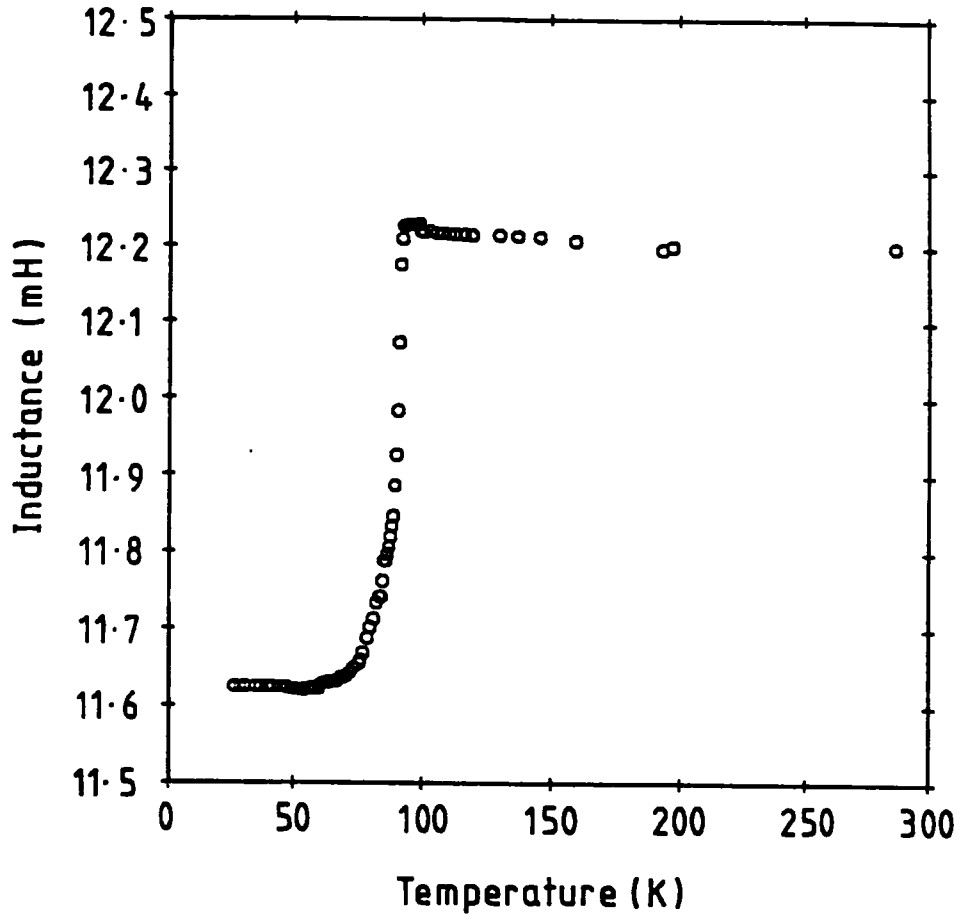
A coil of approximately 2500 turns was wound onto a PTFE former 15mm in diameter and 25mm in length. The coil was attached to a stainless steel tube which was slowly lowered into the neck of an open helium dewar. The temperature of the coil was monitored using a copper-constantan thermocouple referenced to liquid nitrogen and the same experimental procedure for temperature determination followed as for the resistivity transition. The inductance of the coil (typically 12mH) was monitored using a Wayne-Kerr inductance bridge operating at 1kHz. Figure 3.4 shows data obtained for a sample of superconducting material, D980. The transition is broader than that determined from electrical resistivity measurements because the magnetic measurements are not as dramatically affected by the existence of a single percolative path, full shielding occurs when complete surface screening currents are established.

3.4 Magnetisation Curves

3.4.1 Inter-grain Currents

The currents flowing within a polycrystalline superconductor can be considered to consist of two components. The first current is the inter-grain current or transport current which flows through the links between superconducting grains, associated with this current will be an inter-grain critical current density J_c^t which is the current density at which a bulk material appears to lose its superconductivity. J_c^t is often measured by direct methods – applying an increasing current to the sample until a voltage drop across the sample is observed. The critical current density obtained in this manner depends upon the coupling between grains,

Figure 3.4 — Inductance transition of sample D980.



consequently values obtained for simple pressed powders, sintered materials and melt-textured materials will vary considerably.

3.4.2 Intra-grain Currents

The second current flowing within a polycrystalline superconductor is the intra-grain current or local current which flows within the individual grains. Again, this current will have an associated critical current density, J_c^g. The intra-grain currents can only be determined by indirect methods. One of the most popular methods is to perform magnetisation measurements as a function of applied field upon powdered samples and from the magnetic hysteresis attempt to deduce the critical current density using a critical state model.

3.4.3 Bean Critical State Model

An elementary model to calculate the critical current density of a superconductor from hysteresis curves was proposed by Bean. [Bean (1962,1964] This model is frequently used on the grounds that a value of J_c can be rapidly found from the magnetic hysteresis. Bean produced the expression

$$J_c(\mathrm{Acm}^{-2}) = \frac{3 \times 10^4 \Delta M}{d} \tag{3.3}$$

where ΔM is the hysteresis of magnetisation per unit volume ($\mathrm{JT}^{-1}\mathrm{m}^{-3}$) and d is the scaling length (cm). With the application of any critical state model the scaling length used is important. For polycrystalline materials which have only been powder pressed the hysteresis arises as a result of flux pinning within the grains hence the appropriate scaling length will be the average grain size (typically a few μm). For sintered samples the currents flow through the surface of the bulk sample and the appropriate scaling length is the sample dimension.

The Bean model makes a few assumptions, which must be borne in mind when assessing the values of J_c(B) derived from the model. The assumptions Bean made are as follows.

1. The superconductor is capable of sustaining a lossless macroscopic current up to critical current density J_c, which is zero at B_{c2}.

2. The magnetic field is shielded up to the value B_{c1}.

3. The shielding currents flow to the full amount J_c only to the depth required to reduce the magnetic field to B_{c1}.

3.4.4 Vibrating Sample Magnetometer

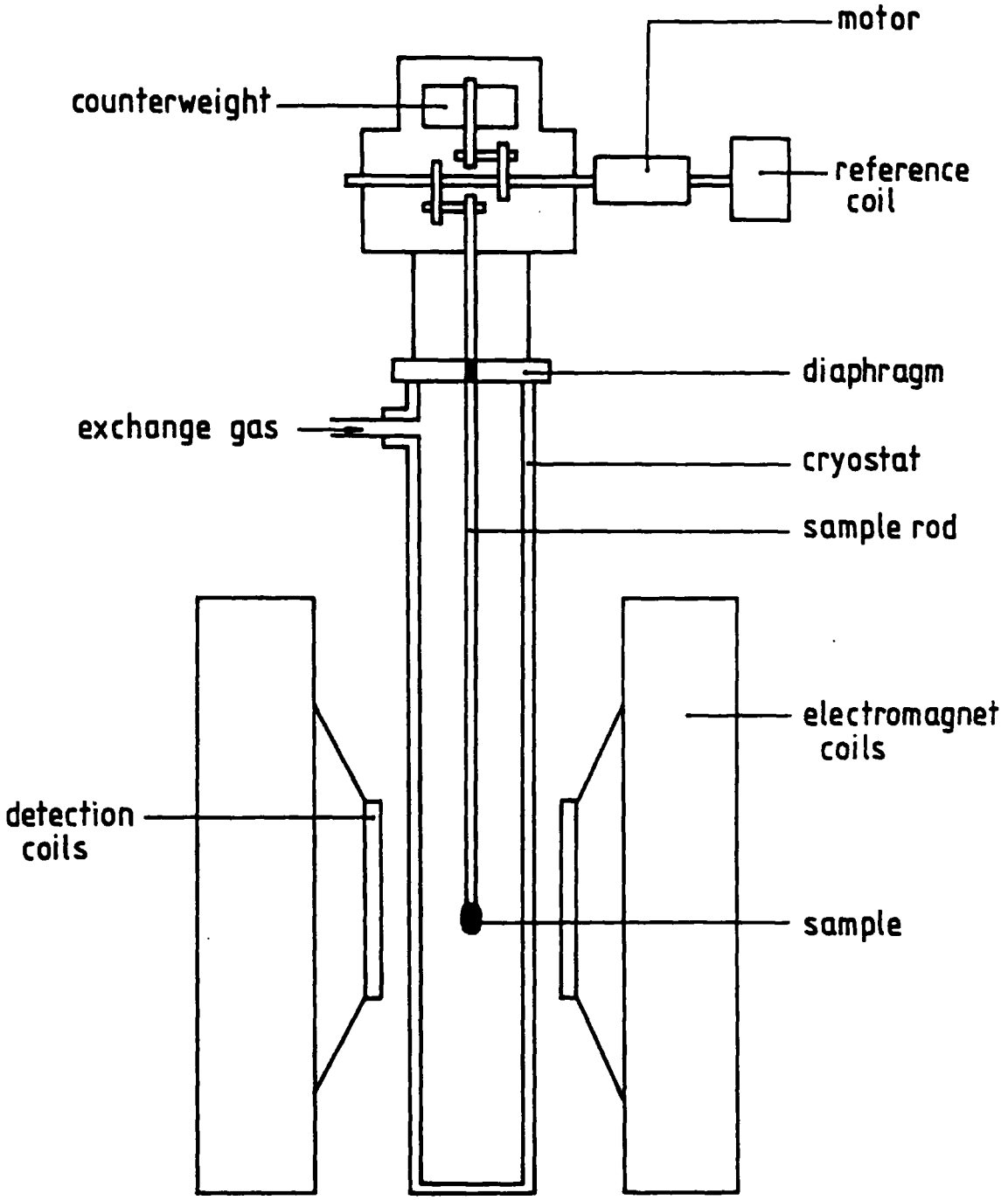
In a vibrating sample magnetometer (VSM) a sample is vibrated in a direction perpendicular to an applied magnetic field which is uniform over the volume of the sample. With the sample fixed, the applied field induces a magnetic moment in the sample which in turn gives rise to a constant magnetic flux through the detection coils. By vibrating the sample, the magnetic flux through the detection coils changes periodically, and this induces a voltage which is proportional to both the magnetisation of the sample and the amplitude of vibration. Attachment of a small permanent magnet to the drive mechanism, with its own pickup coils, provides a reference signal which, in addition to the signal from the sample, can be fed into a phase sensitive detector (PSD). In this way the sample magnetisation can be measured with reasonably high sensitivity. Absolute measurements of magnetisation may be obtained by by calibration against a suitable standard.

A great strength of magnetisation measurements is their high sensitivity to magnetic impurities in a sample, hence a magnetisation curve of $YBa_2Cu_3O_{7-\delta}$ may provide evidence that the material contains a small amount of magnetic impurities. These impurities may have been present in the precursor materials, or may have been introduced during the materials processing stage. As ferromagnetic impurities may affect the superconducting properties, magnetisation measurements may be used to assess the viability of using a particular source of precursor materials or of using a particular preparation route.

3.4.5 Experimental Details

The Durham VSM has been described in detail by Hoon and Willcock (1988), consequently only a bare outline of its operation is provided here. The Durham VSM utilises an electrical motor to drive a 180° double throw crank assembly which in turn vibrates the sample. The head mechanism is pneumatically cushioned from the support mechanism in order to reduce noise. The sample is attached to

Figure 3.5 — The Durham VSM.



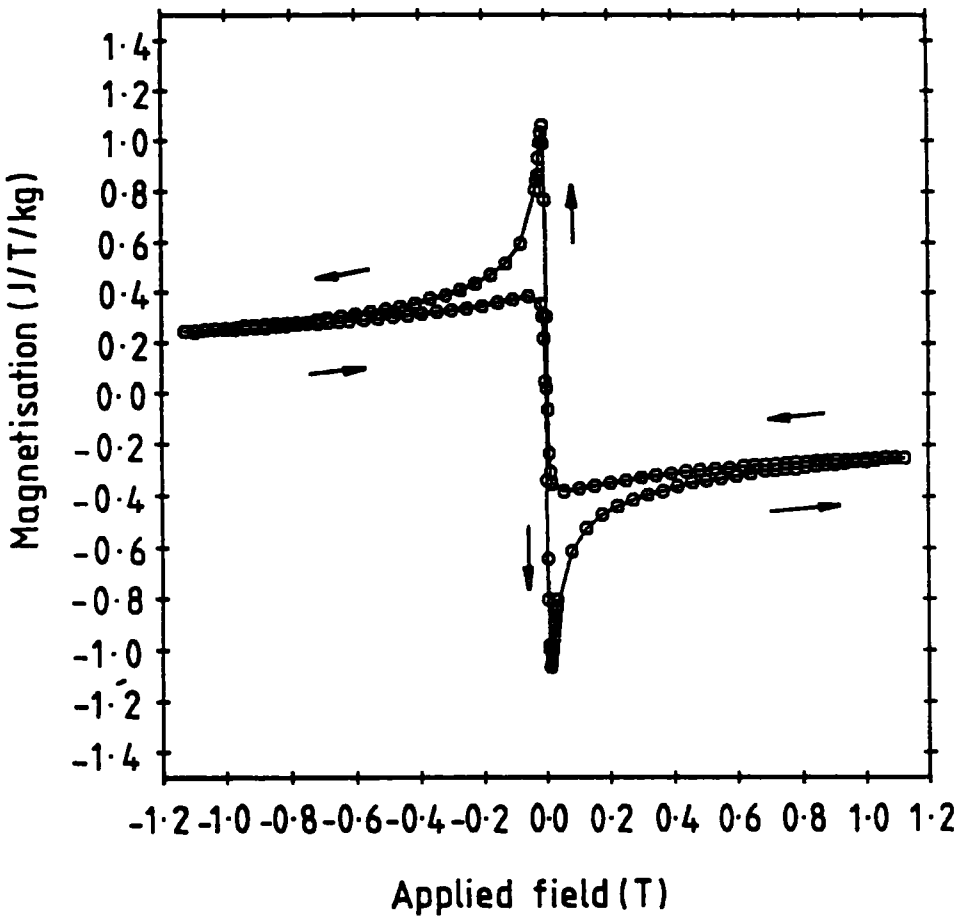
a borosilicate glass rod which is centred, within an Oxford Instruments CF1200 gas flow cryostat, by a PTFE bush. The magnetic field is applied by a Newport Instruments type D electromagnet and monitored with a Hall probe. Figure 3.5 illustrates the experimental arrangement.

In order to produce absolute measurements of the magnetisation, the residual signal from the sample holder and glass rod must first be subtracted from the data. Once this is complete the signal output from the PSD is calibrated using a nickel reference crystal which has a well known magnetisation as a function of temperature and applied field. Such a calibration procedure is only valid if the demagnetisation factor of the nickel reference crystal is taken into account.

An example of a magnetisation curve obtained from a sample of $YBa_2Cu_3O_{7-\delta}$ is given in figure 3.6. At fields below B_{c1} the sample is strongly diamagnetic, but when the applied field is increased above B_{c1} flux lines start to penetrate the sample and the material enters the mixed state. Above a field B_{irr} the magnetisation is reversible showing that there is free flux flow in the sample. The point at which this reversibility occurs corresponds to the onset of flux flow resistivity (discussed in §1.9) and the sample is no longer capable of sustaining a transport current.

Figure 3.6 — Magnetisation curve of sample D970.

Error bars are smaller than the points.



Chapter IV

Thermal Conductivity Experiments

'pon my word, Watson, you are coming along wonderfully. You have really done very well indeed. It is true that you have missed everything of importance, but you have hit upon the method.

— Sir Arthur Conan Doyle.

'A Case of Identity'

4.1 Basic Measurement Methods

The flow of heat in a solid is governed by the equation

$$h_i = -\kappa_{ij}\frac{\partial T}{\partial x_j} \tag{4.1}$$

where κ_{ij} is the thermal conductivity tensor, h_i is the rate of heat flow per unit area perpendicular to direction i and T is the temperature. The values of h_i will usually be independent of the size and shape of the sample although in certain extreme cases (e.g. when the phonon mean free path is of comparable size to the sample) the thermal conductivity will be sample dependent. Such effects are not usually seen until very low temperatures although artificial sapphire has been known to exhibit this effect at $\simeq$ 30K. [Berman et al (1955)]

The measurement of thermal conductivity falls into two main categories - steady state and non-steady state methods.

4.1.1 Non-steady State Methods

Non-steady state methods make use of the idea that an arbitrary temperature gradient imposed upon a sample will change with time at a rate governed by the thermal diffusivity of the sample. If the relaxation time of the system is sufficiently small, a periodic temperature profile may be imposed upon the system and 'lock-in' or data averaging techniques may be used. Non-steady state methods are particularly useful at low temperatures where systems have a low heat capacity and a short relaxation time. At high temperatures the relaxation time may increase to several hours. making accurate determination of the time constant difficult. A review of the various non-steady state methods is given by Touloukhian (1970).

4.1.2 Steady State Methods

Steady state methods rely upon applying a constant rate of heat input to a sample, thereby establishing an equilibrium temperature gradient across the sample. There are a variety of steady state methods which may be used. A detailed discussion of the methods available is given in Touloukhian (1970), Berman (1961) and Berman (1976). The majority of steady state techniques are variants of the longitudinal heat flow and radial heat flow techniques.

4.1.2.1 Longitudinal Heat Flow

With these methods the thermal conductivity is measured in an analogous way to the four probe measurement of electrical resistance. In the thermal case the heat flow corresponds to the electrical current and the temperature difference is analogous to the electrical potential difference. A popular experimental arrangement is to hold one end of a sample rod at a constant temperature whilst a constant rate of heat input ($\dot{Q}$) is applied to the other. At equilibrium this results in a temperature gradient being set up which may be measured using suitable temperature sensors. Longitudinal heat flow methods generally have the advantages of being relatively simple to set up and of allowing direct computation of the thermal conductivity through the mathematical relationship:

$$\kappa = \frac{\dot{Q}l}{A\Delta T} \tag{4.2}$$

where ΔT is the temperature difference measured by the temperature sensors sited a distance l apart. The time taken for such a system to reach 1% of its equilibrium value is given by the expression due to Reese (1966).

$$\tau = \frac{4Cl^2}{\pi^2 \kappa} \tag{4.3}$$

where C is the heat capacity per unit volume, κ is the thermal conductivity and l is the length of the sample. This time constant may become very long for samples which have a very low thermal conductivity or which are particularly long. For a sample of $YBa_2Cu_3O_{7-\delta}$ 20mm long, and using a thermal conductivity of a few $Wm^{-1}K^{-1}$, the time constant will be of the order of a few minutes at 100K.

4.1.2.2 Radial Heat Flow

In order to reduce the effective length of the sample, and thereby decrease the equilibrium time, the sample may be cut into the form of a disc and a radial heat flow technique employed. With this technique the centre of the sample is held at a constant temperature and a heater wrapped uniformly around the circumference of the disc. A constant heat input to the heater will therefore establish a radial temperature gradient throughout the sample. With this geometry the thermal conductivity is given by the expression

$$\kappa = \frac{\dot{Q}\ln(r_1/r_2)}{2\pi l \Delta T} \tag{4.4}$$

where r_1 and r_2 are the radii at which the temperature difference is measured. Radial heat flow methods also have the modest advantage of being dependent upon the ratio of two radii, thereby eliminating errors due to thermal expansion. Use of this method can lead to a reduction in the equilibrium time of an order of magnitude.

4.2 Design Considerations

Any experimental arrangement to measure thermal conductivity is the result of compromise because all connections to the sample incur the penalty of acting as a heat shunt. As a consequence of this the design of a system is strongly

dependent upon the properties of the materials to be measured, hence the system described here represents only one attempt at the solution of the design problems encountered.

Early measurements performed upon $YBa_2Cu_3O_{7-\delta}$ (see §2.6) indicated that the thermal conductivity is comparable to that of a poor metal, ie in the region of 0.1 to 10 $Wm^{-1}K^{-1}$, thus the system was designed to measure thermal conductivities within this range. As this was the first thermal conductivity measurement system to be set up in the Durham Physics department, a detailed description of the consideration of errors and factors influencing the system design is given here.

The samples of superconducting material to be investigated were produced as 20mm diameter pellets. However these suffered from a tendency to crack as they were removed from the die. This made radial heat flow methods impractical. After sintering the pellets were easily cut into bars of approximate dimensions $20x5x3mm^3$, hence the choice of a longitudinal heat flow technique was a natural one.

4.2.1 Temperature Stability

For a steady state technique to be effective, one has to ensure that the temperature gradient across the sample has reached an equilibrium value and that the gradient has not arisen from any temporary drift in temperature. For this reason it is important that the temperature of the cold tip of the cryostat is maintained at a constant value. Fortunately, attachment of the temperature sensor and cold tip heater to an automatic temperature controller makes possible the holding the temperature of the cold tip constant to better than ±10mK.

The temperature of the cold plate of the cryostat study was regulated by a Lakeshore Cryotronics DR91C temperature controller fitted with a Lakeshore Cryotronics GaAlAs diode sensor. The sensor was mounted inside a copper block which was in turn screwed into the central arm of the sample clamp, a good thermal contact between the sensor and the copper block was ensured by thermally anchoring the sensor leads to the block. Calibration of the sensor was performed by the manufacturers according to the American NBS standard and initially enabled absolute temperature to be measured to ±10mK over the entire 20K-120K range.

Unfortunately, during the course of the experiments, the sensor performance degraded, possibly due to the mounting adhesive used by the cryostat manufacturers slowly attacking the sensor housing (the explanation was given by the sensor importers) and a second sensor had to be used. The characteristics of this GaAlAs diode were compared against a calibrated rhodium/iron resistor and the resulting calibration curve fitted to a 12^{th} order chebyshev polynomial using a standard least squares algorithm. [Press et al (1986)] The temperature reproducibility of this second sensor was approximately $\pm$10mK and did not appreciably affect the stability of the cold plate.

With a temperature difference, $T_1 - T_2$, established across the sample, the measured thermal conductivity is an average of the values between T_1 and T_2. Berman (1976) has shown that if the material has a thermal conductivity which varies as sharply as T^3 then the difference in apparent and true conductivities is less than $\frac{1}{4}\%$ even for $(T_1 - T_2)/T_1 \simeq 1/10$. In this study, the temperature gradients used varied between 0.5K and 1.5K depending upon the temperature of the sample.

4.2.2 Sample Heater

In order to limit the errors in the value of the thermal conductivity data obtained, the heat input to the free end of the sample must be known. Ideally, there should be no contact to the sample heater at all; the entire heat loss would then be due to the black body radiation emitted by the heater. Such a system could be achieved by the use of a powerful light source whose output is channelled to the end of the sample by a fibre optic cable. This arrangement is feasible for situations where relative changes in heater power are required and indeed this method has been exploited by Stokka and Fossheim (1988) when performing high resolution 'a.c.' heat capacity measurements. However, absolute measurements of thermal properties by this method are difficult as the efficiency of the transmission cable and emissivity of the sample must be known at all temperatures.

Use of a simple resistance heater requires a compromise to be reached between the necessity to have leads connected to the heater (which shunt heat out of or into the sample) and the resistance of the current leads which dissipate power by ohmic heating. If the wire is too thin or too long the resistance of the leads could be a

significant proportion of the heater resistance, whereas if the leads have too great a cross sectional area, a significant amount of heat may be shunted into or out of the sample, resulting in an inaccurate value for κ. A discussion of this aspect of the design compromise may be found in Berman (1961) and White (1979).

The sample heater used in this study was a standard RS 120Ω strain gauge. This had the twin advantages of being compact, measuring only 8x3x0.25mm^3, and of having a low residual resistance ratio, such that the resistance fell only slightly over the 20K-120K range. Conduction of heat to, or away from, the sample by the heater leads was limited by 20mm lengths of 46swg manganin wire. These wires had the effect of adding approximately 1.5Ω to the heater resistance at room temperature thereby leading to an uncertainty in the heater power of approximately 1.5%. The current was supplied to the heater by 32swg copper wires which were thermally anchored to the central post of the sample clamp. A further two 32swg copper wires attached in the same way allowed direct measurement of the voltage drop across the strain gauge and manganin wire, thus enabling the power developed across the heater to be known accurately.

4.2.3 Residual Gas Conduction

At low pressures, gas molecules have a large mean free path and will tend to travel from one surface to another without colliding into other molecules. The transfer of heat under such conditions is given by the equation [White (1979)]

$$\dot{Q}_{gas} = \alpha \frac{\gamma+1}{\gamma-1} \frac{\sqrt{R}}{\sqrt{8\pi}\sqrt{MT}} P(T_2 - T_1) \tag{4.5}$$

where γ is the ratio C_p/C_v, P is the gas pressure in Pa, R is the universal gas constant, M is the molecular weight of the residual gas and α is an 'accommodation coefficient' which is related to the corresponding coefficients for each surface. Assuming that the residual gas in the system is helium, the surfaces are poor ($\alpha = 1$) and the pressure gauge is at room temperature, the above expression reduces to

$$\dot{Q}_{gas} \simeq 0.03P(T_2 - T_1) \quad \mathrm{Wcm}^{-2} \tag{4.6}$$

For particularly poor surfaces and a temperature difference between the sample and radiation shield of 2K, the heat loss due to residual gas conduction from

a sample 2x0.5x0.3cm^3 and a strain gauge 0.8x0.3x0.025cm^3 is $0.226 \times P$ Watts where P is the residual gas pressure in Pascals. In all of the thermal conductivity experiments discussed in this thesis the sample was held in a vacuum better than 10^{-8}Pa, hence for an applied heater power of 5mW the effect of residual gas conduction was negligible.

4.2.4 Radiation Losses

A black body at a temperature T emits radiant energy at a rate $\dot{Q} = \epsilon\sigma T^4$ per unit area, where σ is Stefan's constant, (5.7×10^{12} Wcm^{-2}K^{-4}) and ϵ is the emissivity of the material. If the sample or heater is at 100K and is surrounded by material at 4.2K, the losses may be as high as 0.6mWcm^{-2}. Such losses can appreciably affect the experimental measurements. In order to keep these losses to a minimum, the sample heater was covered with clean aluminium foil which has an emissivity of much less than 1, reducing the losses by a factor of 50 or so. [White (1979)] The cryostat design was such that the vacuum can was thermally connected to the sample cold plate, ensuring that the sample was always surrounded by a material at a similar temperature.

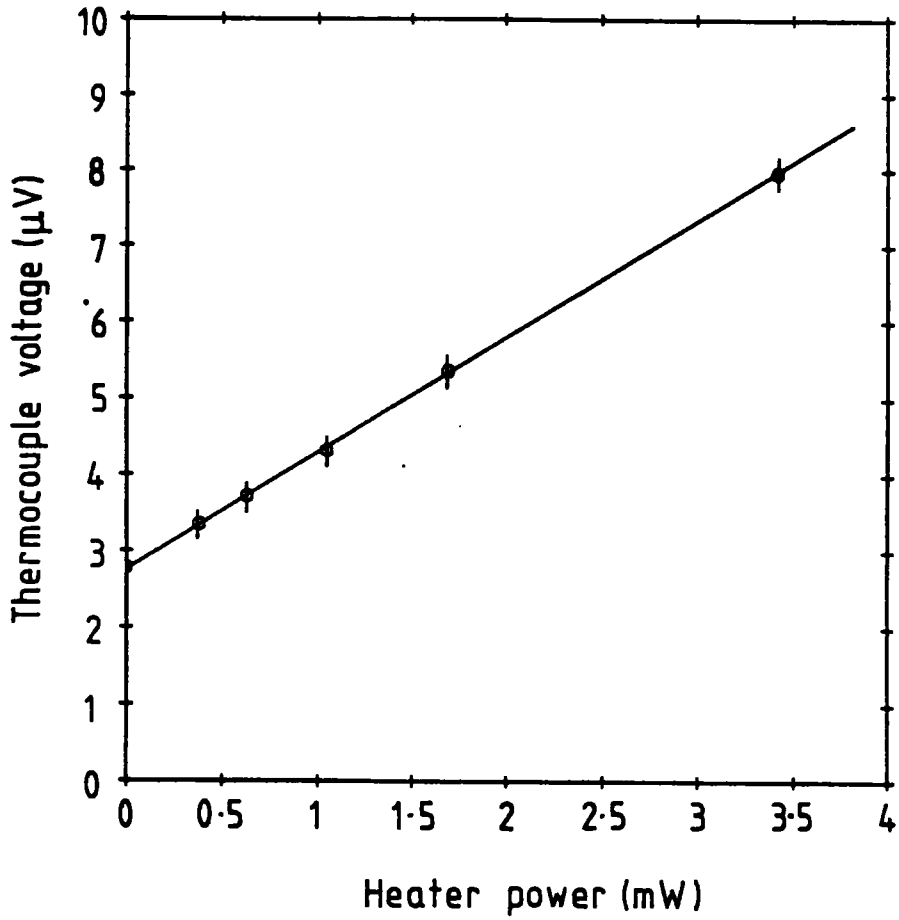
4.2.5 Measurement of Temperature Difference

Accurate measurement of small temperature differences is extremely difficult, requiring the use of either two well-characterised sensors (typically calibrated to better than 5mK) or the use of a direct method of measuring the difference. A thermocouple has obvious potential for the task of direct measurement of a temperature difference, although great care must be taken to ensure that offset currents and spurious emfs are eliminated. A thermocouple made from gold+0.03% iron vs chromel has the advantage of a (roughly) constant and high sensitivity ($\simeq$13μV/K) at temperatures from 4.2K to 300K [Berman et al (1964)] whereas many other thermocouples suffer from loss of sensitivity as the temperature decreases, e.g. copper-constantan has a sensitivity of $\simeq$ 12μV/K at 300K which falls to only $\simeq$ 2.5μV/K at 4.2K. [Berman et al (1965)] The measured thermocouple voltage may be readily converted to a temperature difference by use of standard reference tables and appropriate interpolation algorithms.

A disadvantage to the use of a differential thermocouple is that the absolute temperature of the sample is not measured, it is therefore common practice to use a second thermocouple or other temperature sensor to measure absolute temperature of the sample. With the cryostat used in this study, no fixed reference temperature was available, hence the use of a thermocouple to determine absolute temperature was not possible. A second high resolution temperature sensor with suitable characteristics was not available, however the samples were by necessity short, temperature gradients were kept to a minimum and the control sensor was thermally anchored to the cold tip at a point as close as possible to the sample, thus the absolute temperature of the sample was taken to be the temperature indicated by the temperature sensor. The temperature gradients of 0.5K-1K used for each measurement provides an estimate for the uncertainty in the sample temperature. For accurate work at helium temperatures this state of affairs is most unsatisfactory although it is adequate for the measurements made in this study which were at temperatures above 20K.

The measurement of small temperature differences by use of a direct thermocouple is complicated by the presence of offset currents which arise principally from the finite input impedance of the nanovoltmeter. The effect of these currents is that even when the thermocouple junctions are thermally anchored to the same point, the nanovoltmeter will indicate a finite voltage. During the course of these experiments, the offset currents were nulled by a control on the nanovoltmeter which injected a current in opposition to the input offset current. Offset voltages may also arise from strain, inhomogeneities or local temperature gradients in the thermocouple wires. Such voltages may be compensated for by use of superconducting reversing switch which enables measurement of the thermocouple voltage in a positive and negative sense. Unfortunately, the cryostat design did not allow the use of such a switch, hence the offset voltages were computed by removing the heat input to the free end of the sample and allowing the thermocouple arms to reach equilibrium. Further checks were made by measuring the thermal conductivity for different sample heater powers. This is analogous to measuring the electrical resistivity of a sample by applying various currents to the sample. Figure 4.1 shows a typical set of data for the thermocouple voltage measured across a sample as the heater power is increased, in the presence of an offset voltage on the thermocouple. At high heater powers the function will deviate from linear

Figure 4.1 — Measured thermocouple voltage as a function of heater power.



owing to the influence of radiation losses and the gradual change in the thermal conductivity of the sample along the direction of the temperature gradient.

4.2.6 Thermocouple Wires

A further possible source of error in the experiment is conduction of heat by the thermocouple wires, resulting in an artificially low temperature gradient being established along the sample. As the thermocouples are anchored to the sample clamp and the temperature difference always kept below 1K, this value is used to estimate the conduction 'error' from the thermocouple wire. At 100K, gold has a thermal conductivity of approximately $300\mathrm{Wm}^{-1}\mathrm{K}^{-1}$ and chromel has a thermal conductivity of approximately $20\mathrm{Wm}^{-1}\mathrm{K}^{-1}$. Each wire has a nominal diameter of 80μm hence:

$$\dot{Q}_{AuFe} = \frac{300 \times (40 \times 10^{-6})^2 \times \pi}{l} = \frac{1.5}{l}\mu W \tag{4.7}$$

$$\dot{Q}_{Chromel} = \frac{20 \times (40 \times 10^{-6})^2 \times \pi}{l} = \frac{0.1}{l}\mu W \tag{4.8}$$

For a measuring power of only 1mW the heat conduction down 1 metre of each wire is respectively 0.15% and 0.01% of the total power, in practice the thermocouple wires were approximately 2m in length. In order to fit the wires into the sample space it was necessary to form them into loosely bound coils. This was achieved by carefully winding the each wire onto a stainless steel bar where and lightly holding it with very weak GE varnish. Once the varnish had set, the wires were carefully removed from the rod and small pieces of hair tied around each coil at three points in order to keep the coil in place. The GE varnish was then removed using methanol/toluene solvent.

The thermocouple junctions proved to be extremely difficult to make owing to the very large difference in melting points of the two alloys. They were eventually successfully welded by the following technique: the ends of wires to be joined were stripped of their enamel using a methanol/toluene solvent, the final 1.5cm of the wires were then twisted together and lightly clamped using a pair of fine stainless steel tweezers. A conventional car battery was used to charge two 33mF capacitors in parallel, the battery was disconnected and one terminal of the capacitors connected to the tweezers. The second terminal of the capacitors was connected

to a small, clean copper plate. Spot welding of the wires was achieved by touching the ends of the wires on the copper plate – effectively shorting out the capacitors. Possible oxidation of the wires was impeded by performing the weld in a helium rich atmosphere. A visual check of each weld was made using an optical microscope to ensure that the two wires had fused together, a few attempts were necessary before an acceptable bond was made. Once the junction was made, the wires were carefully untwisted and covered with a thin layer of GE varnish to restore the electrical insulation.

With two thermocouple junctions successfully made, the three coils of wire were supported in the sample space by fine cotton thread to ensure good thermal isolation from the rest of the system. The free ends of the chromel wires were anchored to the clamp centre post and to a binding post situated on the copper cold plate. At this point the chromel wires were soldered to 0.2mm copper wires, coated in GE varnish and bound together. Some problems were encountered with the copper wires, as appreciable emfs ($\simeq 5\mu$V) were generated in some lengths of wire when they were immersed fully in liquid nitrogen and connected directly to a voltmeter. This problem was overcome by using lengths of high purity copper wire which gave emfs of less than 1μV when immersed in liquid nitrogen.

From the chromel-copper junctions the copper wires were taken to a second binding post on the cold plate before being led through a separate stainless steel tube to the top of the sample insert. Here they they were fed continuously through the top plate of the cryostat, the vacuum seal being made with Stycast 2850FT resin set with LV24 catalyst. Outside the cryostat, the wires were separately covered with insulating heat shrink and gently twisted together before being shielded by a copper braid. The shield, nanovoltmeter and cryostat were earthed via the copper water pipes in the laboratory, in this way the possibility of spurious thermal emfs and signals from radio-frequency noise were minimised.

4.3 Experimental Arrangement

4.3.1 Cryostat

The variable temperature insert of the cryostat used for the studies in this thesis is depicted schematically in figure 4.2. The sample and associated equipment

is attached to a copper plate at the end of an insert some 1.3m long and the insert lowered into the cryostat and screwed firmly into a heat exchanger. Radiation losses are reduced by a number of copper heat baffles placed at regular intervals along the tube. This particular cryostat design allows the vacuum seal to be made at room temperature, although it does suffer from the possibility of the sample and associated equipment being stressed as the insert is lowered and screwed into the heat exchanger. The sample is cooled by maintaining the bath of liquid helium at a slight overpressure, thereby forcing liquid helium through the needle valve and into the heat exchanger. A needle valve restricted the gas flow, further temperature regulation was achieved by use of a 25 Watt resistance heater mounted on the cold plate.

4.3.2 Sample Mounting

The samples were held by a simple clamp arrangement constructed from copper (see figure 4.3). The lower screw enables the sample to be loosely clamped, whilst the upper screw may be carefully tightened to ensure a firm grip. In order to insulate electrically the sample from the clamp, the clamp faces were covered with cigarette paper soaked in weak GE varnish. The thermal contact between the sample and copper clamp was aided by coating the clamp faces with a small quantity of vacuum grease. Two 0.3mm diameter holes were drilled into the samples in order to accept the thermocouple junctions, these were filled with vacuum grease in order to improve the thermal contact and allow a good bond length between the thermocouple junctions and the sample. [Kopp and Slack (1971)]

4.3.3 Constant Current Power Supply

In order to facilitate accurate and reliable measurements of both electrical resistance and thermal conductivity as a function of temperature, a constant current power supply was required. The power supply was designed to provide a constant current of 1mA to 10mA to the sample heater. A conventional car battery was used as the basic power source for the supply, thereby virtually eliminating 50Hz noise which is present to some degree in all mains powered supplies. The basic circuit is shown in figure 4.4.

Figure 4.2 — Thermal conductivity cryostat sample chamber.

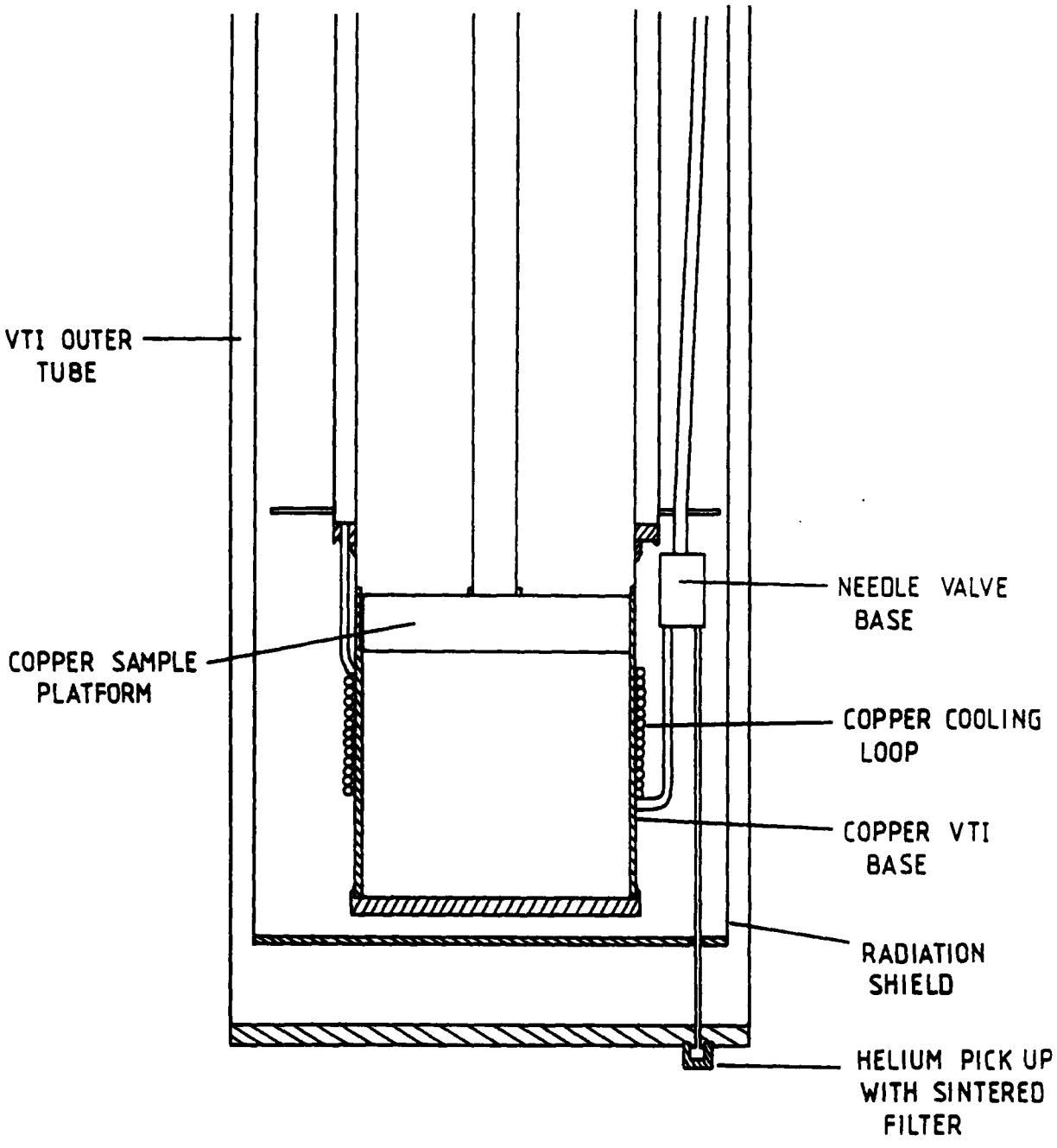


Figure 4.3 — Thermal conductivity cryostat sample stage.

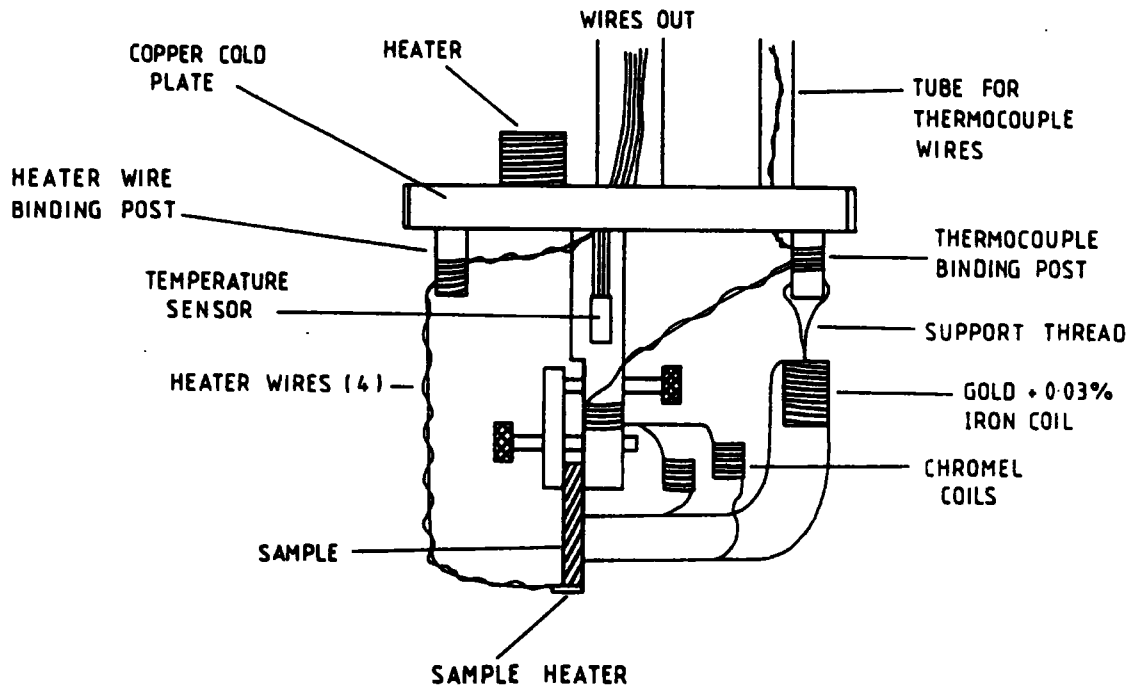
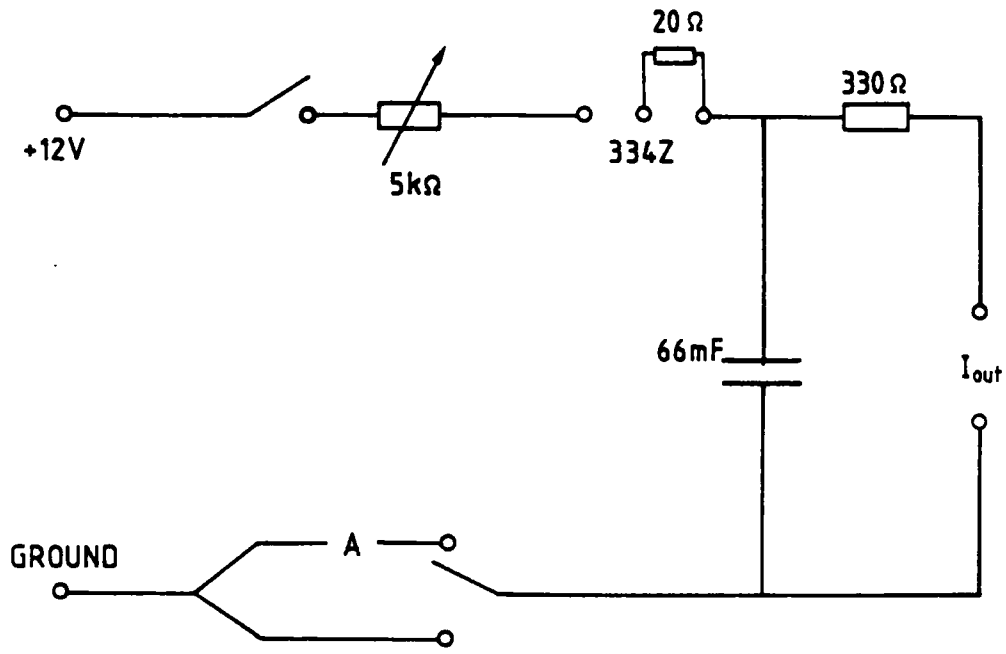


Figure 4.4 — Circuit diagram of the constant current power supply.



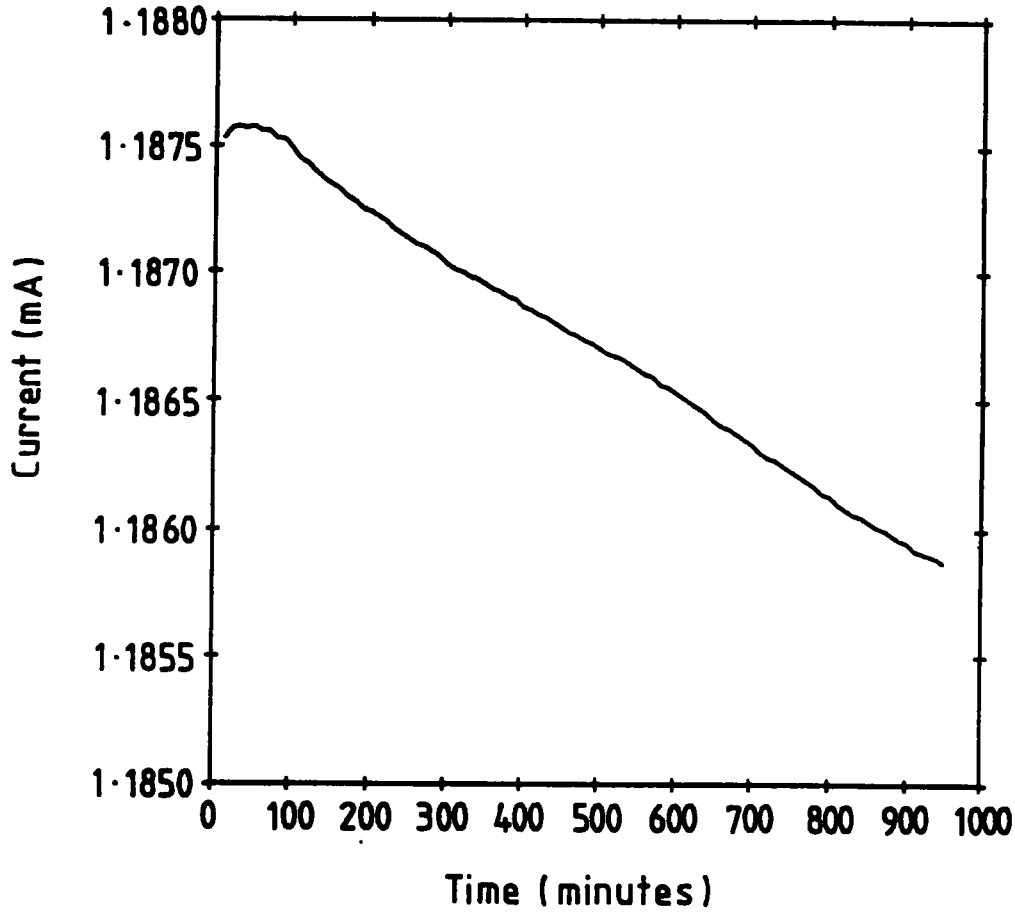
An integrated circuit, a 334Z current stabiliser, provided a basic stabilising circuit. The output of this circuit was smoothed further by use of two 33mF capacitors in parallel. The current output from the supply was selected by altering the resistance input to the 334Z, achieved by use of a rotary switch selecting one from a range of resistors. Two further switches were also incorporated, the first permitted polarity reversal of the current (useful for resistance measurements), whilst the second enabled an ammeter to be included in the circuit. All of the resistors in the construction of the supply were of a metal oxide type which were chosen for their low inherent noise.

Figure 4.5 shows the typical output current of the supply to a strain gauge of 120Ω resistance. The current is seen to be stable over a long period of time, to within 1.5μA over 15 hours, a change of 0.15%. This is of particular importance for the thermal conductivity measurements where the heater power has to be stable in order to establish and maintain an equilibrium temperature gradient. The drop in current was caused by the small drain on the car battery. The stability of the output current from the supply was checked at frequent intervals throughout the time experimental data was accumulated, no change in the stability was observed.

4.3.4 Data Acquisition System

The measurement of thermal conductivity was automated, initially by use of an Acorn BBC microcomputer with IEEE interface, and later by use of an 'IBM PC' compatible Elonex 286 PC-AT fitted with a Scientific Solutions IEEE interface card. All of the measurement instruments (except the nanovoltmeter) were fitted with IEEE cards to enable the two way flow of information from the PC which acted as an overall system controller. The nanovoltmeter could not be fitted with an IEEE card, however an analogue output from a precision amplifier was available and this output was fed into a Keithley 197 microvoltmeter which was fitted with an IEEE bus. The analogue output was found to correspond to the indicated voltage on the analogue display of the nanovoltmeter for ranges of 30μV fsd and lower. As 30μV corresponds to a temperature gradient of approximately 2K this range proved adequate for the measurement. A further limitation imposed by the nanovoltmeter was caused by the nanovoltmeter being unable to drive the analogue output continuously for more than a couple of hours without

Figure 4.5 — Power supply output current.



the internal batteries becoming exhausted. The computer therefore controlled a Bede Scientific Instruments MINICAM fitted with an IEEE interface which fired a relay connecting the nanovoltmeter to the mains power. The nanovoltmeter could then be electrically isolated before each reading and the integrity of the electrical grounding maintained. A schematic diagram of the system connection is given in figure 4.6.

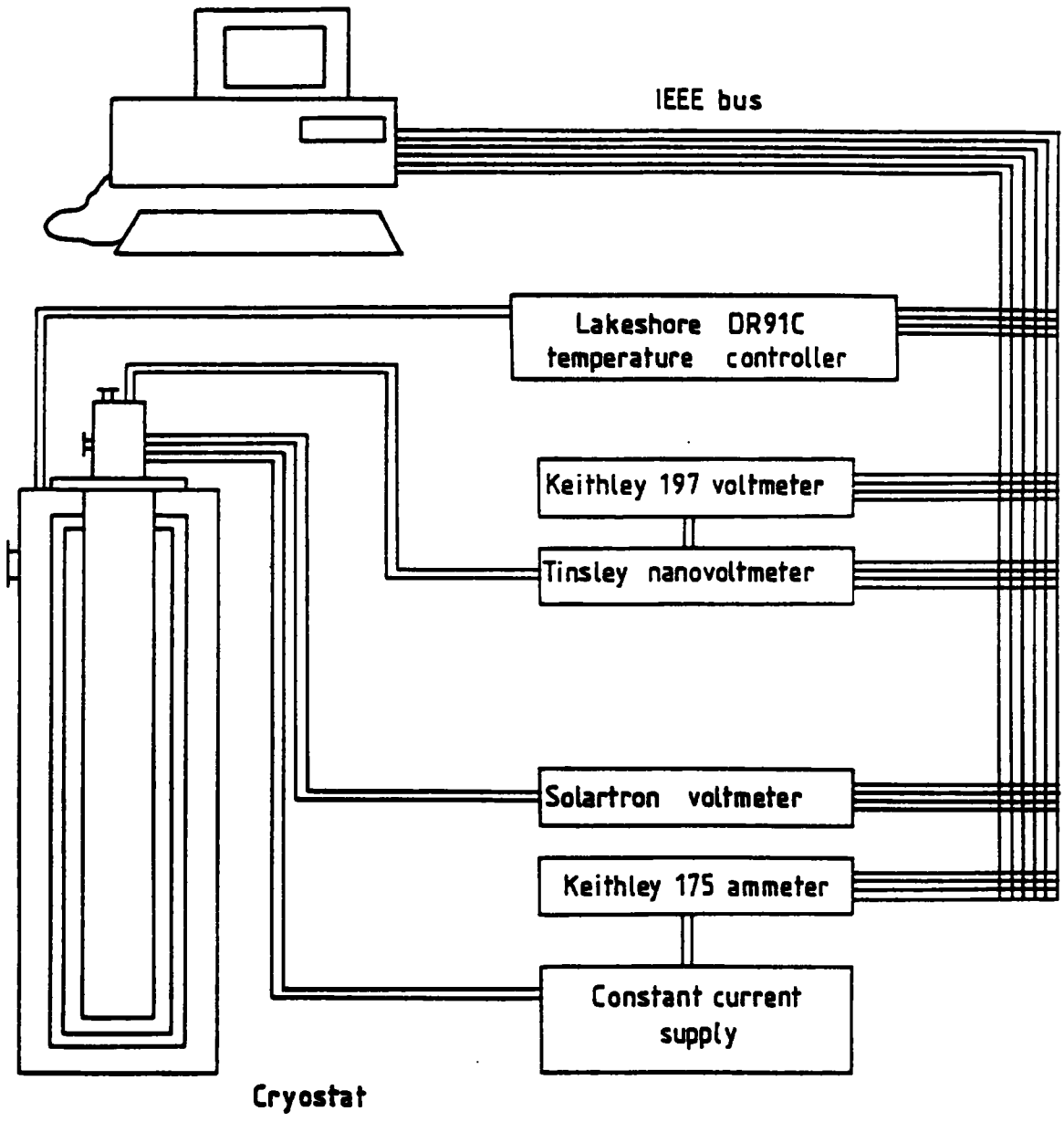
The source code of the program controlling the entire system was written in BASIC for the BBC and in Turbo Pascal for the PC. Briefly, the program allows the operator to program a set of times at which the set temperature is to change. At each temperature the computer changes the PID values on the temperature controller to appropriate values to reflect the change in the response of the system as it cools. Data may be taken at intervals specified by the operator and plotted on screen, enabling an immediate display of the thermal conductivity as the temperature is altered. Automation of the data acquisition has the great advantage of allowing data acquisition to occur overnight, within certain restrictions brought about by the physical logistics of the cryostat (for example it is not possible to control the flow rate of helium gas using the computer). A complete list of the Turbo PASCAL source code of the control software is given in appendix E.

4.4 Test Data

In order to verify that the thermal conductivity measurement system was capable of performing accurate measurements upon a range of materials, the system was initially tested with a sample of perspex. Perspex has a thermal conductivity over an order of magnitude lower than that expected of the $YBa_2Cu_3O_{7-\delta}$ samples. If there were significant sources of error from radiation losses, residual gas conduction, conduction along the thermocouple wires or conduction along the sample heater wires, the measured thermal conductivity of perspex would be significantly higher than those of the literature values. Consequently if the value of a low thermal conductivity material was measured correctly it is reasonable to assume that the results for materials of higher thermal conductivity would be substantially correct.

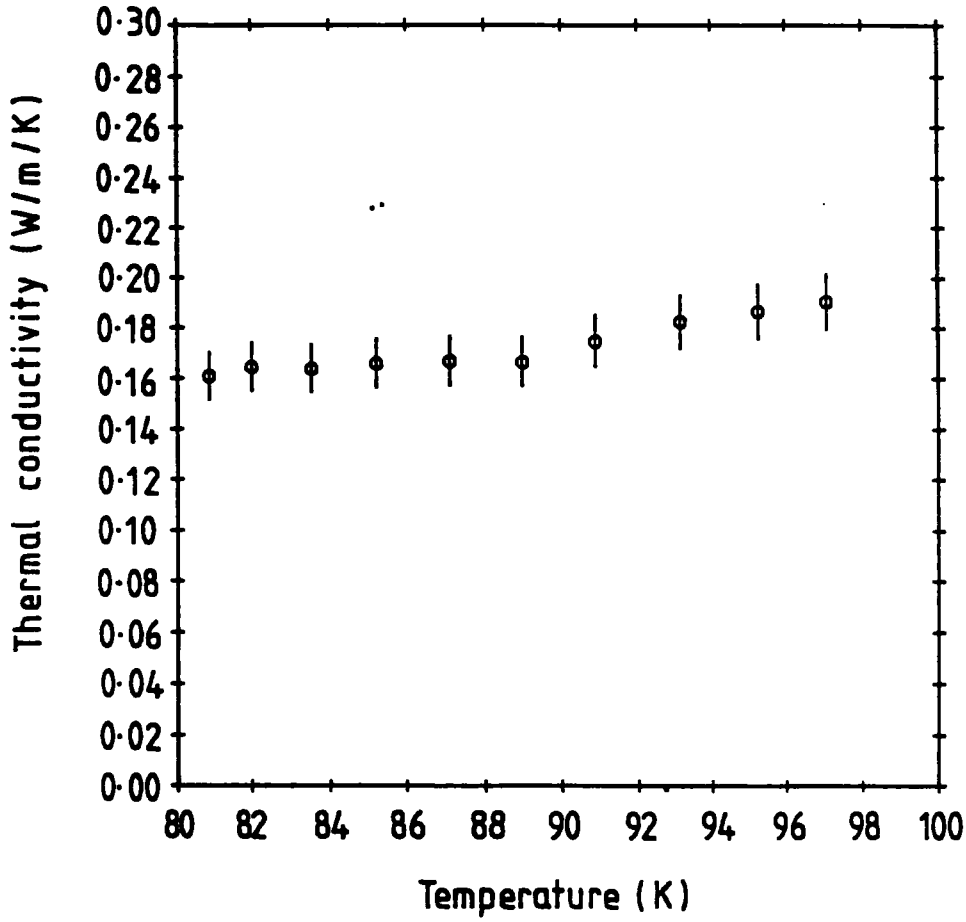
A perspex sample, of dimensions $20x3x3mm^3$ was clamped to the cold tip of the cryostat and the thermal conductivity measured at temperatures just above 77K.

Figure 4.6 — Connection of the data acquisition system.



This served the secondary purpose of determining whether or not there were any anomalies in the thermal conductivity apparatus which might otherwise be falsely attributed to the superconducting nature of $YBa_2Cu_3O_{7-\delta}$. Owing to the long time constant of the system caused by the low thermal conductivity of perspex, the thermal conductivity was measured at each temperature after allowing 24 hours for the sample to reach equilibrium. The results in the region 80K to 100K are depicted in figure 4.7. The data of Burgess and Greig (1974) indicates that the thermal conductivity of of polymethylmethacrylate in this temperature range is $\simeq 0.16 Wm^{-1}K^{-1}$. This is in agreement with the data obtained and therefore, as an initial test, showed that the thermal conductivity apparatus was capable of producing acceptable data.

Figure 4.7 — Measured thermal conductivity of perspex.



Chapter V

Thermal Conductivity Study I
Effect of Oxygen Content

The first moral of this story is just a practical one. Always test your general reasoning against simple models.
— John Bell.
Speakable and Unspeakable in Quantum Mechanics

5.1 Introduction

In §2.6 it was stated that the thermal conductivity of $YBa_2Cu_3O_{7-\delta}$ varies considerably with porosity of the sample. In addition, the detailed microstructure will influence the overall thermal conductivity in the normal and superconducting state. Very few thermal conductivity results have been published in which a property of the same sample of $YBa_2Cu_3O_{7-\delta}$ has been systematically altered. In this study a sample of $YBa_2Cu_3O_{7-\delta}$ was characterised before undergoing vacuum and oxygen annealing to alter the oxygen content of the material. This enabled the influence of oxygen content upon the thermal conductivity to be examined in a systematic way without the influence of variations in porosity, sample size or the detailed microstructure of the material.

5.2 Sample Preparation

The material was prepared by solid state reaction of precursor compounds obtained from Johnson Matthey Materials: 99.9999% yttrium oxide, 99.999% barium carbonate and 99.999% copper (II) oxide. The precursor materials were mixed in

an agate mortar and placed in an alumina crucible. Calcination was performed at 990°C for 21 days using a Carbolite CTF12/65 tube furnace fitted with a Pt vs Pt+10%Rh thermocouple, solid state relay and Eurotherm 818P temperature programmer. The calcined material was ground in an agate mortar, placed in the crucible and heated to 985°C for 48 hours and cooled to room temperature at 40°C/hr. After the calcination process the material was finely ground in a Specamill 8000 agate mill for 30 minutes.

A 20mm diameter pellet of the material was produced by pressing in a Specac stainless steel die at a presssure of 405MPa (4kbar) using a uniaxial bench press. Sintering was performed at 970°C for 24 hours followed by cooling to room temperature at 40°C/hr. Oxygen annealing was performed at 450°C for 48 hours under flowing oxygen followed by a slow cool at 20°C/hr. This resulted in a material which was approximately 85% of the expected density of single crystal $YBa_2Cu_3O_7$. From the pellet, a bar of dimensions 20x5x3mm^3 was cut using a fine bladed saw. After each set of measurements, the bar was subjected to a heat treatment in order to alter the oxygen content. The sequential list of heat treatments is given in table 5.1.

Table 5.1 — Sequential heat treatments of sample D970

Reference	Temperature	Pressure	Time	T_c
D970/1	450°C	1013 mbar	48 hrs	92 K
D970/2	450°C	10^{-2} mbar	24 hrs	68 K
D970/3	450°C	10^{-2} mbar	40 hrs	< 4.2 K
D970/4	500°C	1013 mbar	48 hrs	90 K

5.3 Preliminary Characterisation

5.3.1 X-ray Powder Diffraction

A portion of the 'as-made' sample was powdered and analysed using the X-ray powder diffractometer described in §3.1. The material was found to consist of single phase $YBa_2Cu_3O_{7-\delta}$ with no evidence for the presence of Y_2BaCuO_5 or $BaCuO_2$. The diffraction lines for the (010) and (100) peaks were distinguishable indicating

that a substantial quantity of the material had an orthorhombic structure. The c axis lattice parameter was determined to be 11.70±0.01Å, a rough estimate for the oxygen content of the sample may be obtained by combining this information with the correlation between c axis lattice parameter and oxygen content discussed in §2.3. From this relationship, the oxygen content of the sample was determined to be approximately 6.85 O atoms per Y atom.

5.3.2 Electrical Resistivity Measurements

The electrical resistivity as a function of temperature of a second bar cut from the 'as made' pellet is shown in figure 5.1. Use of a second bar ensured that the integrity of the sample used for the thermal conductivity experiments was maintained. The sample had a low electrical resistivity, 0.4mΩcm, at 100K, and exhibited a sharp transition at 92K in an applied field of less than 0.1mT. As the applied field was increased, the 90%-10% transition width broadened from 1K to 3.5K at 500mT. The low resistivity at 100K, the sharpness of the resistance transition and its resilience to change in relatively high magnetic fields is indicative of the sample having a high overall oxygen content and good inter-grain coupling. No resistivity measurements were performed upon the sample in each of the vacuum annealed states, however the resistance of the sample was measured after it had been re-oxygenated. The material exhibited evidence for semiconducting behaviour. This suggested that the oxygen content of the material had not been restored to its initial level.

5.3.3 Inductance Measurements

The inductance transition of the sample in each state was measured using the apparatus described in §3.3 As the measurements were performed using coils of different inductance, a 'normalised' inductance given by

$$L_{norm}(T) = \frac{L(T) - L(4.2K)}{L(100K)} \tag{5.1}$$

was defined, this allowed a direct comparison of the transitions to be made. The results are plotted in figure 5.3. The sample in the 'as made' condition had a T_c of 92K and a 90%-10% transition width of approximately 10K which is considerably

Figure 5.1 — Resistivity transition of sample in state D970/1.

o ... <0.1mT + ... 0.5mT △ ... 5mT

◇ ... 50mT ▫ ... 500mT

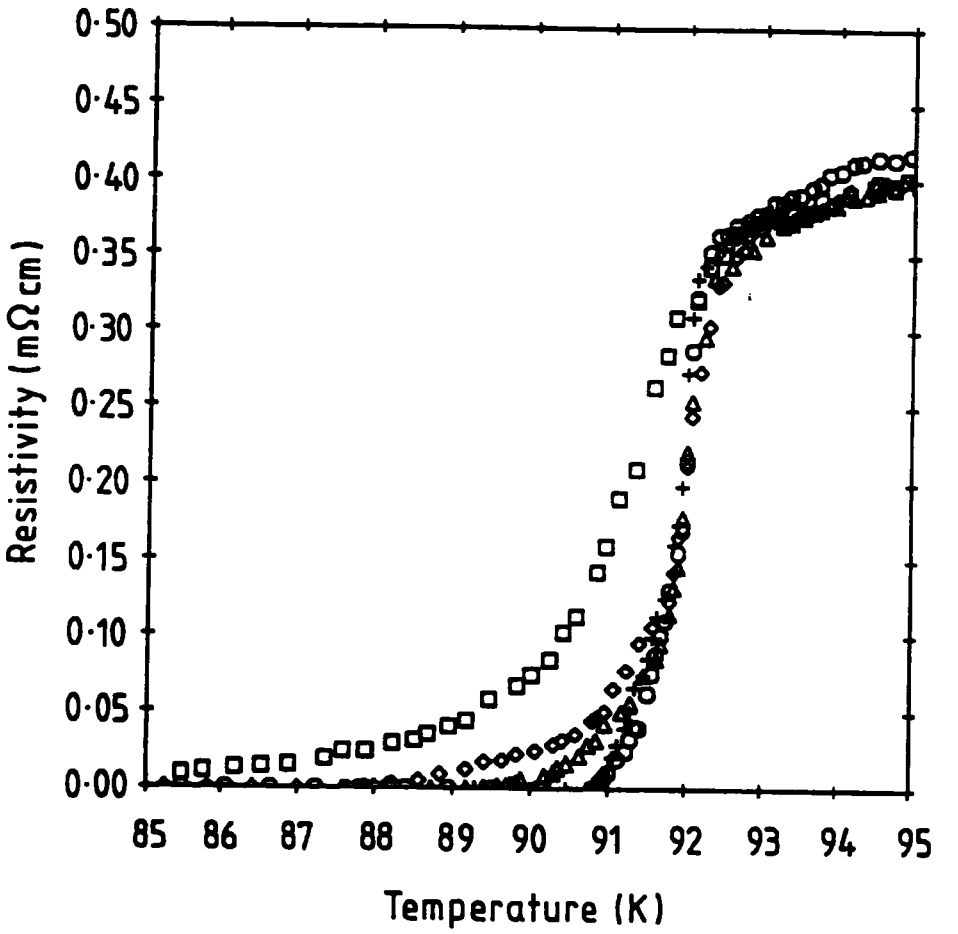
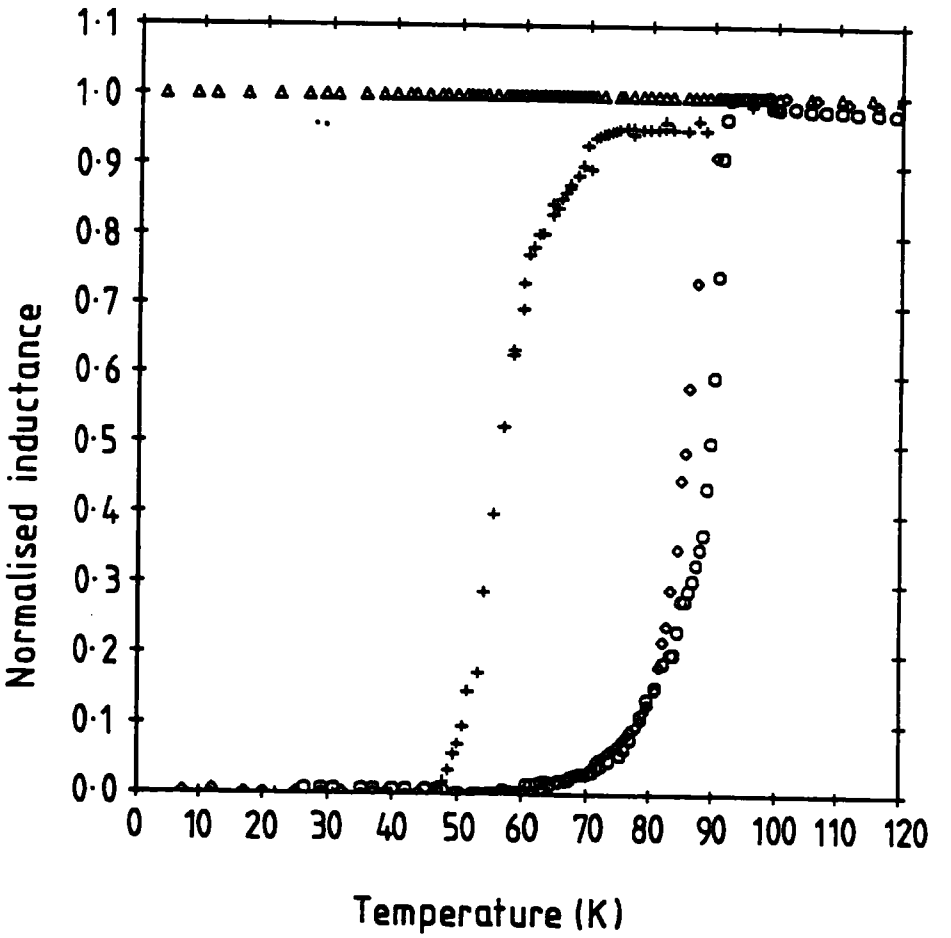


Figure 5.2 — Inductance transition of sample in states D970/1 to D970/4.

o ... D970/1 + ... D970/2 △ ... D970/3 ◇ ... D970/4



broader than the resistivity transition. After the first vacuum anneal, no evidence for superconductivity was seen until approximately 68K when the inductance fell gently, further cooling revealed a sharp transition at 60K. After the second vacuum anneal, the sample did not superconduct even when placed in liquid helium, suggesting that much of the oxygen has been removed from the sample. After oxygen re-annealing, the sample showed a sharp drop in inductance at 90K with a transition width of 12K, slightly broader than that of the 'as made' sample.

5.3.4 Oxygen Content

The preliminary characterisation data may be used to estimate the oxygen content of the sample in each of the different states. This is achieved by comparison of the measured T_c values of the material in each state with the data of T_c as a function of oxygen content obtained from the published literature (see §2.3). The width of the superconducting transition for the sample in each state provides some indication of the spread of T_c values within the material. The estimated ranges of oxygen content of the sample in each annealed state is given in table 5.2.

Table 5.2 — Approximate oxygen content of sample D970.

Treatment	Method	T_c	ΔT_c	O_x
D970/1	Resistance	92K	1K	$x > 6.95$
	Inductance	92K	10K	$x > 6.85$
D970/2	Inductance	68K	10K	$6.5 < x < 6.7$
D970/3	Inductance	< 4.2K	—	$6.0 < x < 6.3$
D970/4	Resistance	90K	12K	$x > 6.8$
	Inductance	90K	12K	$x > 6.85$

5.4 Thermal Conductivity Measurements

The thermal conductivity of the sample as a function of temperature in the range 20K-120K was measured for the sample in each of the four states using the thermal conductivity apparatus described in chapter IV. A preliminary measurement of the thermal conductivity at temperatures down to 77K was made upon

the sample in the 'as made' state using liquid nitrogen as coolant. The sample was then warmed, removed and re-mounted before being cooled again using liquid helium. The measured thermal conductivity at temperatures down to 77K was the same, within experimental error, whether helium or nitrogen was used as coolant. Figure 5.3 shows the thermal conductivity of the sample in each of the four states. All states which exhibited a superconducting transition showed an enhancement of the thermal conductivity upon passing into the superconducting state, consistent with the results of many other research groups (see §2.6).

A second striking feature of the effect of the vacuum annealing treatment is that as the oxygen is removed from the material there is a dramatic decrease in the normal state thermal conductivity of the material. The thermal conductivity of sample D970/1 at 100K being approximately 40% higher than that of D970/3. The mild heat treatments given to the sample would not be expected to alter significantly the microstructure of the sample. When oxygen is partially restored to the material (D970/4), the thermal conductivity in the normal state increases and the enhancement of the thermal conductivity below T_c is again observed. These results provide conclusive evidence that the thermal conductivity of the material in both the normal and superconducting state is very strongly influenced by oxygen content.

5.5 Quantitative Model

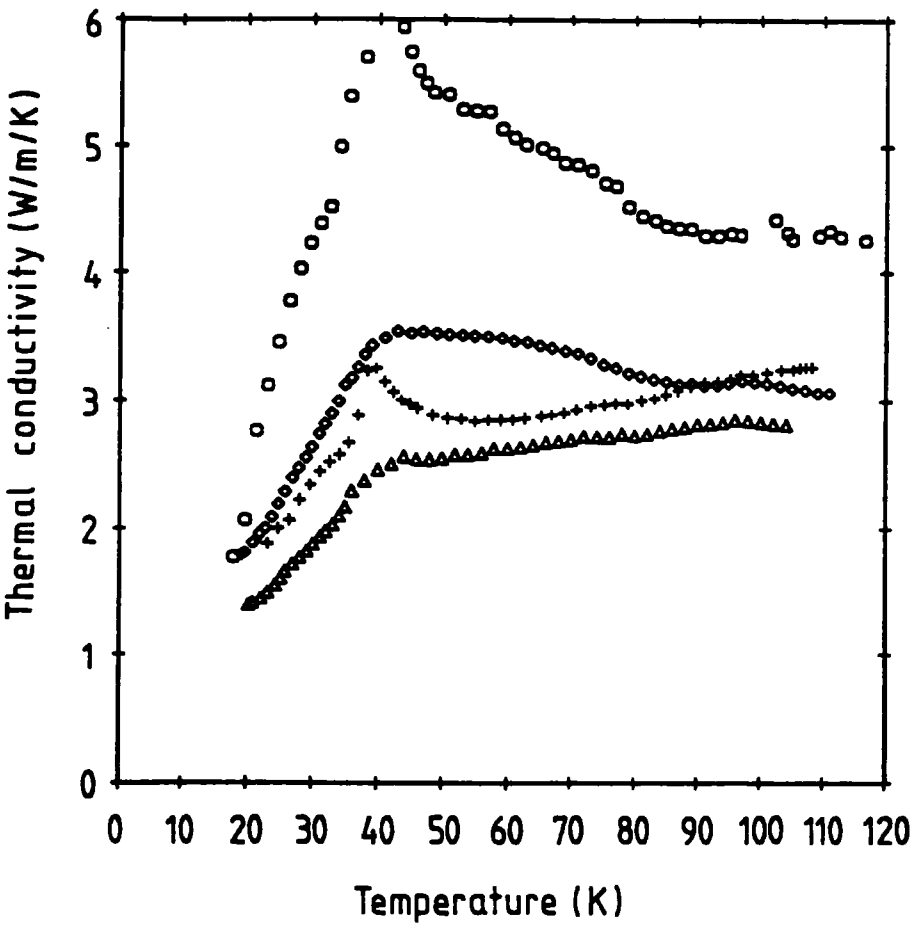
5.5.1 Formulation

As the understanding of the physics of a problem and unambiguous interpretation of experimental results is difficult without a quantitative model with which to test various hypotheses, an elementary quantitative model has been developed. The basis of the model is the construction of the Boltzmann equation for a phonon 'gas' which is solved approximately by utilising the relaxation time approximation. The terminology and mathematical formulation of this approach is given in appendix A, although for clarity a brief outline of the formulation and the assumptions made are given here.

The Boltzmann equation is constructed by considering the 'heat current' which arises from the phonon modes, $\mathbf{q}_i$. In the presence of a temperature gradient, the

Figure 5.3 — Thermal conductivity of sample in states D970/1 to D970/4.

o ... D970/1 + ... D970/2 △ ... D970/3 ◇ ... D970/4



distribution of the phonons will depart from the equilibrium distribution, $N^0(\mathbf{q}_i)$, given by the Bose-Einstein statistical factor

$$N^0(\mathbf{q}_i) = \frac{1}{e^{\hbar\omega_i(\mathbf{q}_i)/k_BT} - 1} \tag{5.2}$$

There will therefore be a 'heat current' which arises as a result of the phonon drift if $N(\mathbf{q}_i) \neq N(-\mathbf{q}_i)$. In the steady state, the overall phonon distribution will be independent of time, consequently the phonon drift must exactly counteract all other processes which alter the phonon distribution – so called 'scattering' processes.

The relaxation time approximation assumes that the scattering processes will restore equilibrium at a rate which is proportional to the departure from equilibrium. This rate is characterised by a relaxation time, τ. By making this approximation, and considering only heat flow which is parallel to the direction of the temperature gradient the thermal conductivity is given by the expression

$$\kappa(T) = \frac{1}{3}k_B \int_\omega \mathbf{v}_g(\omega)^2 \tau(\omega) \left(\frac{\hbar\omega}{k_BT}\right)^2 \frac{e^{\hbar\omega/k_BT}}{(e^{\hbar\omega/k_BT} - 1)^2} D(\omega) d\omega \tag{5.3}$$

The density of states in the above expression involves integrating over all of $\mathbf{q}$ space i.e. averaging over all possible crystal orientations. This is accounted for in equation 5.3 by assuming that that an 'average' phonon group velocity may be found, for each phonon energy, which is capable of reproducing the phonon density of states, $D(\omega)$. A detailed discussion of this point may be found in the text of Grimvall (1986). Since the thermal conductivity measurements were made upon polycrystalline samples, it is reasonable to assume that, experimentally, a similar average has been performed and that the information based upon the phonon density of states is adequate. Partial phonon dispersion curves for $YBa_2Cu_3O_{7-\delta}$ have been measured by Reichardt et al (1989). Their results suggest that upon splitting the measured phonon density of states into appropriate energy intervals it is reasonable to reproduce the empirical phonon density of states using constant (i.e. frequency independent) average group velocities over each interval. With these approximations, equation 5.3 may be used to help elucidate the origin of

the temperature dependence of the thermal conductivity as the oxygen content is altered.

Each mechanism which scatters phonons may be characterised by a relaxation time and it is this relaxation time approximation which provides a mathematical description of the physics of the scattering mechanisms. Unfortunately, it is not possible to calculate exactly the overall relaxation time for the convolution of all of the phonon scattering mechanisms. It is therefore useful, as a first approximation, to consider each of the scattering processes as independent. This implies that the relaxation time may be written in terms of the relaxation times for each of the independent scattering mechanisms. This is an example of Mattheissen's 'rule'. [Mattheissen (1863)]

$$\tau^{-1} = \sum_j \tau_j^{-1} \tag{5.4}$$

A detailed discussion of the validity of the relaxation time approximation has been given by Carruthers (1961). He noted that the major problem is that the distribution function $N(\mathbf{q}_i)$ depends not only upon the occupation of the state being considered but upon all other states as well. Hence the characteristic relaxation time for each scattering mechanism will depend upon how far the system has deviated from equilibrium. For systems which are close to equilibrium, the approximation is likely to be a good one.

5.5.2 Obtaining the Phonon Density of States

In order to use equation 5.3 to compute the thermal conductivity, the 'bare' phonon density of states should be used. Measurement of the 'bare' phonon density of states by inelastic neutron scattering is made difficult due to the neutron data being weighted by a numerical factor which accounts for the 'strength' of neutron scattering by each atomic species. This factor depends upon the scattering cross section and the inverse of the mass of each atomic species. For Y, Ba Cu and O this factor is 0.08449, 0.0252, 0.1178 and 0.2647 barns/atom/amu respectively. As the higher energy phonon modes ($\hbar\omega > 30$meV) may be attributed almost entirely to oxygen vibration states [Chaplot (1990), Reichardt (1989)], then unless these scattering factors are accounted for the measured phonon density of states will be weighted too strongly towards high energies.

5.5.3 Umklapp Processes

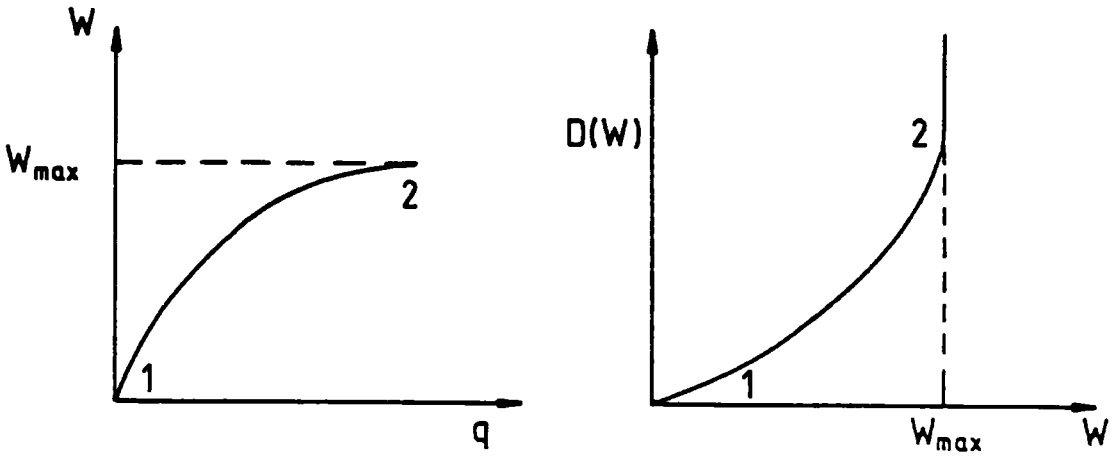
It is relatively straightforward to show that normal scattering processes preserve the crystal momentum and so will not restore an arbitrary phonon distribution to equilibrium, although they will influence the populations of each phonon mode when scattering mechanisms are present. [see, for example, Ashcroft and Mermin (1976)] Umklapp processes occur when two or more phonons scatter and the sum of the wave-vectors lies outside the first Brillouin zone. The onset of a significant contribution to the thermal conductivity from Umklapp processes may be estimated from the first peak in the phonon density of states. Figure 5.4 assists in understanding this idea. The phonon density of states is related to the group velocity $\nabla_{\mathbf{q}}\omega(\mathbf{q}_i)$

$$D(\omega) \propto \sum_i \int_{S_q} \frac{dS}{\nabla_{\mathbf{q}}\omega(\mathbf{q}_i)} \tag{5.5}$$

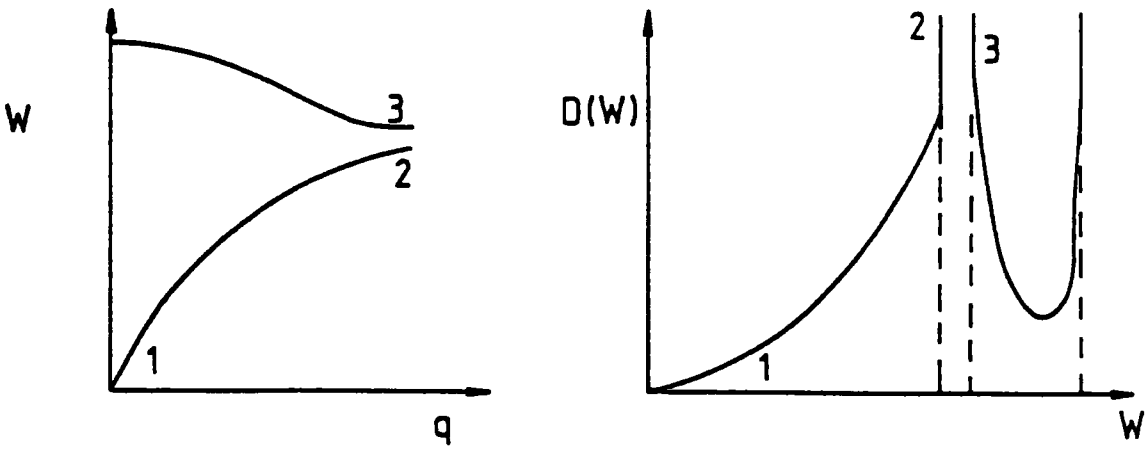
Referring to figure 5.4, in region 1, the phonon modes have a large group velocity and low $\mathbf{q}$, this will lead to a small contribution to the phonon density of states. The periodicity of the lattice requires that the dispersion curve should be zero at the zone boundary. In region 2, the phonon modes have a low group velocity and a high $\mathbf{q}$, giving a large contribution to the phonon density of states. Near the edge of the first Brillouin zone (region 2) the periodic boundary conditions upon the lattice requires the gradient of the dispersion curve to become shallow, hence the phonon density of states will be high. As the energy is increased further (region 3), the modes will have a higher group velocity and a lower $\mathbf{q}$ leading to a fall in the phonon density of states. Hence the first peak in the phonon density of states is indicative of the point at which 2 phonons can just Umklapp scatter outside the first Brillouin zone. From the measured density of states of Strobel et al (1988), Rhyne et al (1987) and the calculated 'bare' density of states of Chaplot (1988), this peak is seen to occur at $\hbar\omega_u \sim$ 16meV and is very nearly independent of the oxygen content of the sample. A characteristic temperature at which Umklapp processes become effective may be determined by assuming that a significant number of Umklapp processes occur when the typical phonon energy is half that at the zone boundary. If it is further assumed that the modes are

Figure 5.4 — Estimation of the characteristic Umklapp temperature

(a) Monatomic lattice



(b) Polyatomic lattice.



populated up to a typical energy $3k_BT$, then ω_u is given by

$$\frac{1}{2}\hbar\omega_u \simeq 3k_BT_u \tag{5.6}$$

As the temperature increases, the number of Umklapp processes occurring will increase and there will be a reduction in the phonon-phonon relaxation time. This fall in phonon relaxation time may be inserted into equation 5.3 by using a temperature dependent relaxation time. At low temperatures, the relaxation time is assumed to have a constant value. As the temperature increases, the relaxation time must decay at some rate which is controlled by a characteristic Umklapp temperature, T_u. A reasonable choice, which is used here, will be to select an exponential decay of the relaxation time τ_{pp} with temperature, thus

$$\tau_{pp} = \tau_{pp1} + d\tau_{pp}(e^{-T/T_u} - 1) \tag{5.7}$$

5.5.4 Effect of Different Phonon Density of States

By utilising the measured phonon density of states from neutron scattering experiments for $YBa_2Cu_3O_6$ and $YBa_2Cu_3O_7$, and using only equation 5.7 as the overall phonon relaxation time, it is possible to estimate the influence of the phonon density of states upon the thermal conductivity. Figure 5.5 shows the result obtained using the phonon density of states measured by Rhyne et al (1988) and the phonon density of states calculated by Chaplot (1988,1990). Chaplot's calculated phonon density of states for $YBa_2Cu_3O_6$ has a large peak at low energies which is not observed in any of the experimental data. This peak has the effect of further increasing the relative population of the low energy modes and increases the calculated thermal conductivity of $YBa_2Cu_3O_6$ relative to that of $YBa_2Cu_3O_7$.

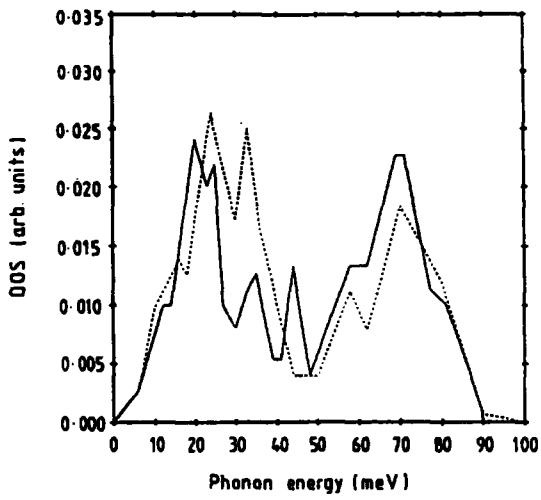
It can be seen from the data that the 'bare' density of states of Chaplot produces a much 'flatter' curve owing to the higher number of low energy phonons. The experimental data of a number of groups [Rhyne et al (1988), Strobel et al (1988), Renker et al (1988)] shows that the phonon density of states for the insulating $YBa_2Cu_3O_6$ material is extremely similar to that of $YBa_2Cu_3O_7$. Rhyne et al observed that the phonon density of states undergoes only a slight renormalisation

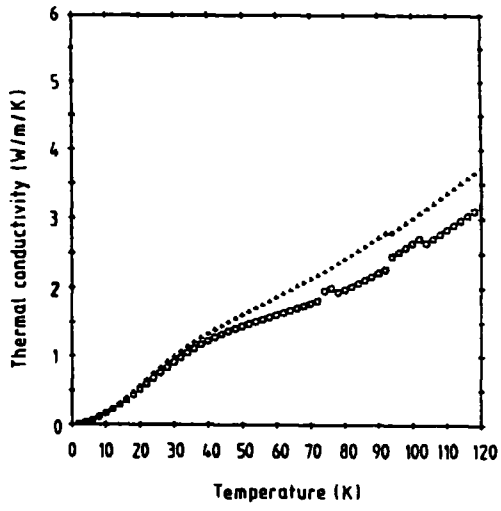
Figure 5.5 — κ computed using only Umklapp scattering.

DOS - - - O_6 — O_7 κ ++++ O_6 o o o o O_7

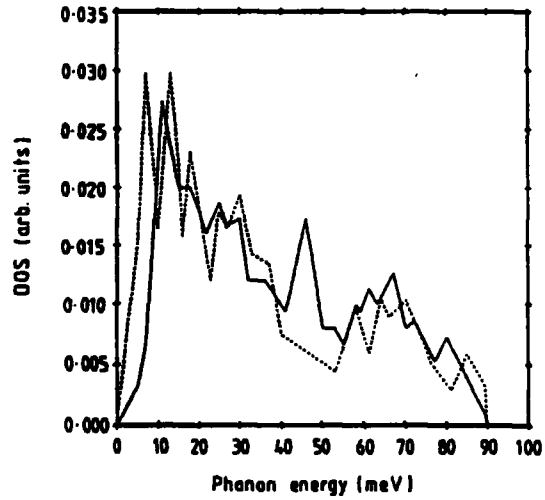
τ_{pp}(low T) = 10ps, τ_{pp}(high T) = 0.5ps, T_u = 20K

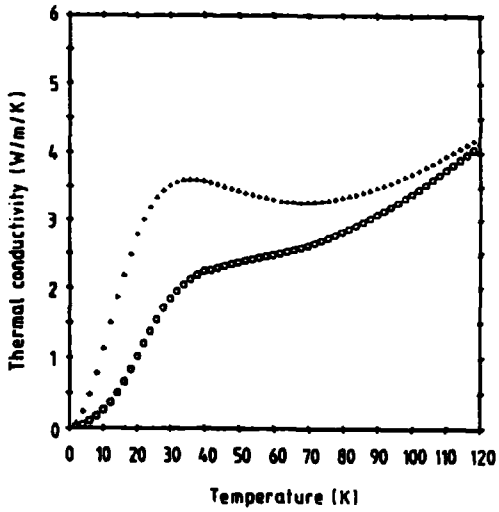
(a) Rhyne et al (1988) 'neutron weighted' phonon DOS.





(b) Chaplot (1990) 'bare' phonon DOS.





as the temperature is decreased. Figure 5.5 shows that the phonon contribution to the thermal conductivity of $YBa_2Cu_3O_6$ is higher than that of $YBa_2Cu_3O_7$. This may be contrasted with the observed thermal conductivity which shows that $YBa_2Cu_3O_6$ has a lower thermal conductivity than $YBa_2Cu_3O_7$. The change in the phonon density of states cannot account for the large change in the thermal conductivity when oxygen is removed from the sample.

5.5.5 Phonon-carrier Scattering

As discussed in chapter II, $YBa_2Cu_3O_{7-\delta}$ which has a sufficiently high oxygen content to undergo a superconducting phase transition exhibits metallic behaviour in the normal state. From this it can be inferred that the material must contain free charge carriers which, in the normal state, will be able to scatter phonons. The phonon-carrier scattering mechanism may therefore be attributed a relaxation time, τ_{pe}. Within the confines of the relaxation time approximation this may be combined with the relaxation time for the Umklapp scattering by use of equation 5.4 thus

$$\tau = \frac{\tau_{pe}\tau_{pp}}{\tau_{pe} + \tau_{pp}} \tag{5.8}$$

As the material passes into the superconducting state, conventional models of superconductivity require that the carriers which condense into the superconducting state are unable to carry entropy and are unable to scatter phonons. If this is accepted, it follows that below T_c there will be an increase in the lifetime of the low energy phonons which are unable to break up the pairs. This process is accounted for in the model by using a τ_{pe} for phonons which have an energy ($\hbar\omega$) above that of the superconducting energy gap which differs from τ_{pe} for phonons which have an energy ($\hbar\omega$) below the superconducting energy gap.

$$\begin{aligned} \hbar\omega < 2\Delta &\rightarrow \tau_{pe} = \tau_{pe1} * d\tau_{pe} \\ \hbar\omega > 2\Delta &\rightarrow \tau_{pe} = \tau_{pe1} \end{aligned} \tag{5.9}$$

The energy gap is temperature dependent and this dependence must be accounted for in the calculation. By measurement of the imaginary part of the conductivity of $YBa_2Cu_3O_{7-\delta}$, Cohen et al (1987) and Porch et al (1988) have

shown that the energy gap in $YBa_2Cu_3O_{7-\delta}$ has a temperature dependence which is close to the form required from BCS theory. The model does not require the BCS theory to be valid, merely that the superconducting mechanism causes the carriers to condense into a state such that the carriers are ineffective phonon scatterers. The temperature dependence of the energy gap may be approximated by the standard 'two fluid' model relation [Schoenberg (1952)]:

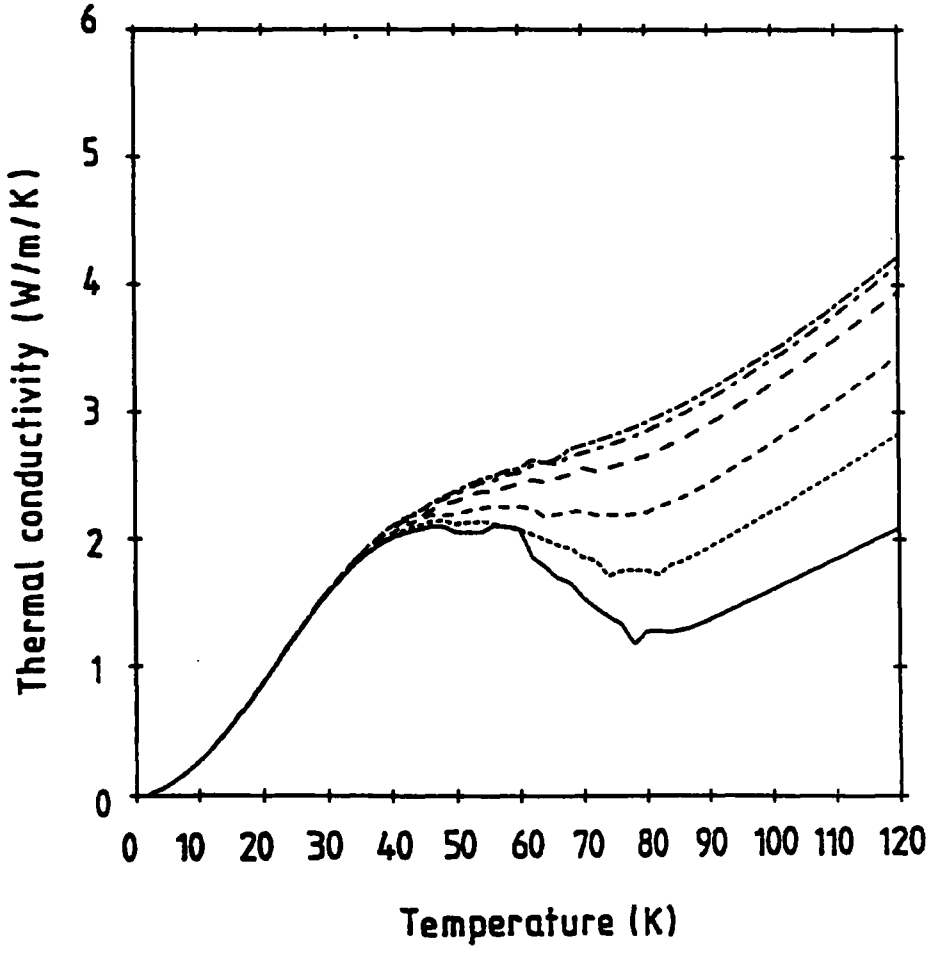
$$2\Delta = 3.52 k_B T_c \left[1 - \left(\frac{T}{T_c}\right)^4\right] \tag{5.10}$$

The calculations presented here assumed that, owing to the absence of carriers in the insulating material, the relaxation time of the phonon-carrier scattering in the insulating material is extremely long. Figure 5.6 shows the effect of adding in the electron-carrier scattering term as detailed above. As expected intuitively, the thermal conductivity of the superconducting material is depressed in the normal state by the carrier-phonon scattering. Upon passing through the superconducting phase transition there is a recovery of the thermal conductivity and it approaches the value of the insulating $YBa_2Cu_3O_6$ phase when the typical phonon energy falls within the superconducting energy gap. This recovery produces a peak in the thermal conductivity as observed in the experimental results. Once again, the thermal conductivity of the superconducting material in the normal state would be expected to be lower than that of the insulator, in sharp disagreement with experiment.

5.5.6 Point Defect Scattering

In chapter II it was stated that oxygen removed from the superconducting material is taken from the Cu-O chains rather than the Cu-O_2 planes, resulting in a tetragonal structure. As the orthorhombic $YBa_2Cu_3O_7$ structure is ordered, removal of oxygen atoms from the Cu-O chain can be considered to create a number of point defects in the material. These point defects will have the effect of acting as scattering centres for phonons and may be accounted for by the addition of a further relaxation time to equation 5.3. Klemens (1955,1969) has shown that the phonon relaxation time for scattering caused by a substitutional atom of different

Figure 5.6 — κ computed using Umklapp and carrier-phonon scattering.
Data obtained using the phonon density of states of Strobel et al (1988).
τ_{pp}(low T) = 10ps, τ_{pp}(high T) = 1.5ps, T_u = 20K
τ_{pe} = 50,20,10,5,2,1ps. The lower curve corresponds to τ_{pe} = 1ps.



mass to a lattice composed of atoms of mass M is

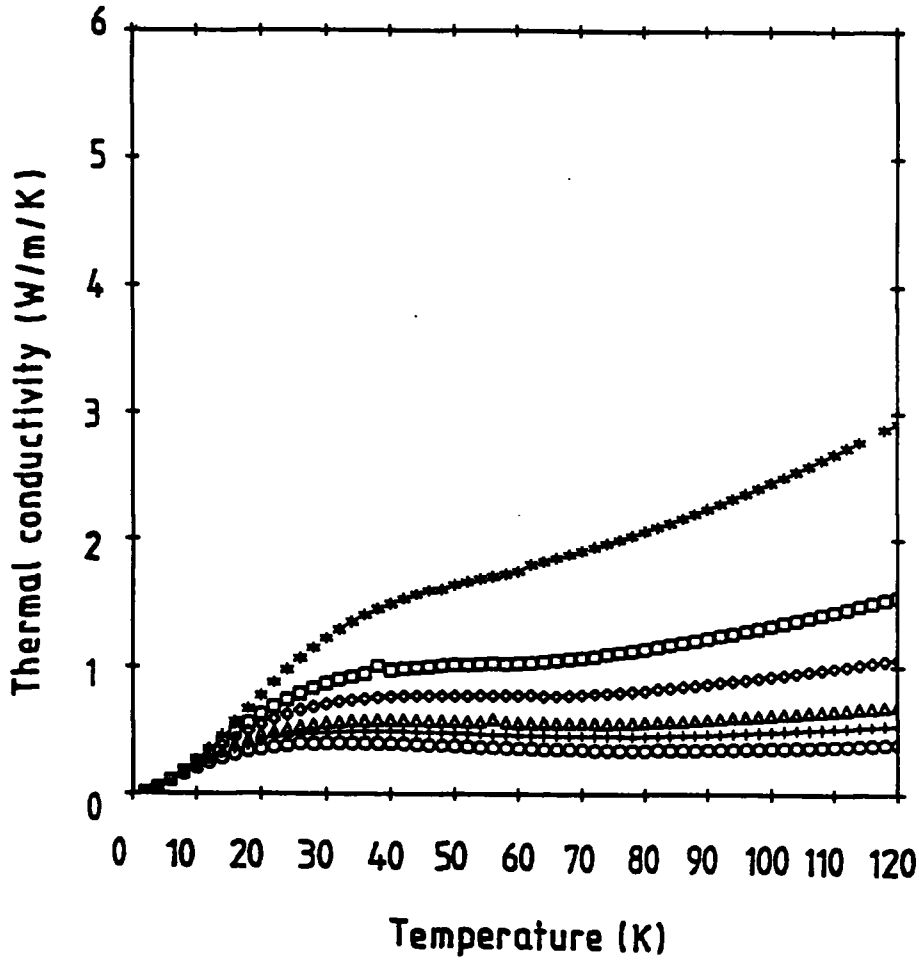
$$\frac{1}{\tau} = \frac{a^3}{4\pi v^3}\left(\frac{\Delta M}{M}\right)^2 \frac{\omega^4}{G} \tag{5.11}$$

where v is the phonon group velocity given by linear dispersion ($\omega = qv$), G is the number unit cells in a volume containing one defect, a is the lattice constant and ΔM is the difference in mass between the solute and solvent atoms. This expression is arrived at by consideration of the perturbation energy of the lattice vibrations of a perfect crystal and may be thought of as simply an expression of Rayleigh scattering. The phonons scatter elastically and independently off point defects which are smaller than the phonon wavelength, leading to the characteristic ω^4 dependence. The expression may be applied to $YBa_2Cu_3O_{7-\delta}$ by treating the oxygen atom as a mass defect sited in an array of atoms of some 'average' mass M, and replacing a^3 by the volume of the unit cell. The various oxygen concentrations may now be treated, to a first approximation, by altering the number of unit cells per defect in equation 5.11. If $YBa_2Cu_3O_7$ is considered to be fully ordered then $G = \infty$, for $O_{6.5}$ G=2, for $O_{6.7}$ G=3.3 etc. $YBa_2Cu_3O_6$ has a fully ordered tetragonal structure, hence $O_{6.3}$ is modelled with G=3.3 as it may be expected to have the same degree of disorder as $O_{6.7}$.

Inclusion of the point defect scattering term in equation 5.3, using equation 5.4, produces the results shown in figure 5.7. The addition of this term reduces significantly the thermal conductivity at higher temperatures. Raising the temperature of the material will result in an increased 'average' energy of the phonon modes. As the collision rate increases strongly with phonon energy, point defect scattering will become more important at high temperatures. The computed curves show that the effect of point defect scattering caused by oxygen vacancies is most effective at depressing the thermal conductivity for an oxygen content of 6.5, yet at this oxygen concentration the sample should still undergo a superconducting transition. The ordered states $YBa_2Cu_3O_6$ and $YBa_2Cu_3O_7$ remain unaffected by the presence of such a scattering mechanism. Hence increasing the concentration of oxygen defects as oxygen is removed from $YBa_2Cu_3O_7$ is unable to provide an adequate explanation of the observed thermal conductivity.

Figure 5.7 — κ computed using Umklapp and point defect scattering.
Data obtained using the phonon density of states of Strobel et al (1988).
τ_{pp}(low T) = 10ps, τ_{pp}(high T) = 1.5ps, T_u = 20K

o ... G = 2	(x= 6.50)	+ ... G = 3.3	(x = 6.70, 6.30)
△ ... G = 5	(x = 6.80, 6.20)	◇ ... G = 10	(x = 6.90, 6.10)
▫ ... G = 20	(x = 6.95, 6.05)	* ... G = 100	(x = 7.00, 6.00)



5.6 Boundary Scattering

Further scattering mechanisms may be present within the material which will affect the thermal conductivity. One possible effect is that of boundary scattering; phonons will scatter at grain boundaries. Such scattering is important at low temperatures when the phonons have a large mean free path. The thermal conductivity limited by grain boundary scattering for the sample used in this study will not change as the oxygen is removed unless a radical re-arrangement of the grains takes place. The vacuum annealing temperatures used were very low, much less than the melting point of $YBa_2Cu_3O_{7-\delta}$ or any impurity phases, hence a drastic change in microstructure is highly unlikely. If the microstructure of the material had altered radically as the oxygen was removed, one must still explain the increase in the thermal conductivity when the sample was re-oxygenated.

It may be argued that a second boundary scattering mechanism exists which could explain the thermal conductivity data – scattering from twin boundaries. As the oxygen content of the material is reduced the material undergoes an orthorhombic to tetragonal phase transition. The occurrence of the phase transition gives rise to the formation of twin boundaries, such that there is 90° re-orientation of the local a direction. The data of Jorgensen et al (1990) indicates that material of oxygen content $\leq$6.4 is tetragonal, consequently for the insulating material no twin boundaries should be present. If twin boundary scattering is important, one may only justifiably introduce this scattering mechanism for materials with δ <0.6 i.e. the materials which undergo a superconducting transition. The addition of a scattering mechanism cannot result in an increase in the phonon relaxation time hence there will be either no change or a reduction in the computed thermal conductivity. Again this can only lead to the situation where the thermal conductivity of the superconducting material would be lower than that of the insulating material. Scattering from twin boundaries cannot explain the reduction in thermal conductivity as the oxygen is removed from the sample.

5.7 Electronic Contribution

One important contribution to the thermal conductivity has been neglected in the model up to this point, the electronic contribution. $YBa_2Cu_3O_6$, being an insulator and therefore containing few free charge carriers will not be expected to

have a significant carrier contribution to the thermal conductivity. This may be contrasted with $YBa_2Cu_3O_7$ which has a low resistivity and therefore has many free charge carriers.

5.7.1 Wiedemann-Franz Law

As discussed briefly in §2.6 the Wiedemann-Franz law [Wiedemann and Franz (1853)] provides one method of estimating the electronic contribution to the thermal conductivity of $YBa_2Cu_3O_{7-\delta}$ in the normal state.

$$\kappa_e = \frac{L_0 T}{\rho} \tag{5.12}$$

For material D970/1 whose normal state resistivity is approximately 0.4mΩcm, the electronic contribution will be approximately 0.28 $Wm^{-1}K^{-1}$. This is less than 10% of the overall thermal conductivity.

It is not clear that the Wiedemann-Franz law should be valid at all for a material which undergoes a transition to a state where the electrons are highly correlated. The Wiedemann-Franz law assumes that the electrons scatter elastically hence both the charge (the carrier of 'electricity') and the energy (the carrier of 'heat') of the electron are conserved and the contribution to the electrical and thermal current possesses the same relaxation time. If the electrons scatter inelastically, charge will be conserved whereas the electron energy state will be altered and the Wiedemann-Franz law will fail.

For electrons which are scattered by phonons, inelastic collisions will occur when the 'average' phonon (determined from the consideration of the thermal energy and the density of states) has a wave-vector close to the Brillouin zone boundary. At very high temperatures (such that $k_B T$ is much greater than the maximum phonon energy $\hbar\omega_{max}$), the electron scattering by phonons is *effectively* elastic and the Wiedemann-Franz law will hold. From the phonon density of states of $YBa_2Cu_3O_7$, $\hbar\omega_{max}/k_B \sim 900$K, thus one would not expect the Wiedemann-Franz law to hold at the temperatures under observation in this study. Hence a different approach must be adopted to compute the carrier contribution to the thermal conductivity.

5.7.2 Model for Electron Scattering

In order to model the electronic contribution to the thermal conductivity it is assumed that the contribution will follow the general form of equation 5.3.

$$\kappa_e = \frac{1}{3} C_v v^2 \tau_e \tag{5.13}$$

where C_v is the electronic specific heat capacity, v is the 'typical' electron velocity which will be of the same order of magnitude as the Fermi velocity, and τ_e is the relaxation time for electron scattering processes. The electronic contribution to the heat capacity may be obtained from the heat capacity of the free electron gas. [see, for example, Ashcroft and Mermin (1976)]

$$C_v = \frac{\pi^2}{3} k_B^2 T g(\epsilon_F) \tag{5.14}$$

where $g(\epsilon_F)$ is the electronic density of states at the Fermi energy. Substituting for the various parameters, it can be shown that the the thermal conductivity is given by

$$\kappa_c = \frac{\pi^2}{3} \frac{k_B^2 T n_c}{m} \tau_e \tag{5.15}$$

where n_c is the number of 'normal state' carriers. At temperatures above T_c, it is assumed that the number of such carriers is constant whereas below T_c the number of such carriers will fall as the electrons condense into the superconducting state.

$$\begin{aligned} n_c &= n_n - n_s(T) \qquad T < T_c \\ &= n_n \qquad\qquad\quad T \geq T_c \end{aligned} \tag{5.16}$$

where $n_s(T)$ is the number of electrons which have condensed into the superconducting state at a temperature T.

As in the case for the phonon contribution, the relaxation time in equation 5.15 provides the mathematical description of the remaining physics of the problem. To a first approximation the relaxation time may be written in terms of the relaxation times of electron-electron and electron-phonon 'collisions'.

$$\tau_e = \frac{\tau_{ee}\tau_{ep}}{\tau_{ee} + \tau_{ep}} \tag{5.17}$$

It is further assumed that the relaxation time τ_{ep} is the same as that appropriate for phonons scattered by electrons τ_{pe}, which increases below T_c according to equation 5.9. An estimate for the functional form of τ_{ee} is now required.

At low temperatures few electrons will be excited to a level above the Fermi energy ϵ_F. If one considers an electron in an energy state ϵ above the Fermi surface, then as effectively only electron states below the Fermi surface are occupied any electron-electron scattering will occur by interaction with an electron which lies within a depth ϵ below the Fermi surface. The Pauli exclusion principle requires that the two electrons scatter only into unoccupied states – which are above the Fermi surface. It can be demonstrated [see, for example, Ashcroft and Mermin (1976)] that this leads to a scattering rate which varies as T^2, the relaxation time for such processes is thus represented in this calculation by the functional form

$$\tau_{ee} = \frac{\alpha}{T^2} \tag{5.18}$$

where α is some constant.

A complete model for the electronic contribution to the thermal conductivity must account for the increase in τ_{ee} and τ_{ep} as the temperature falls. At sufficiently low temperatures the electron-electron scattering relaxation time will be very large and the electron-phonon scattering will dominate. This may be contrasted with the phonon case where the phonon-electron scattering relaxation time becomes large and the phonon-phonon relaxation time is dominant.

The number of electrons which condense into the superconducting state may be described by the functional form of the 'two fluid' model. [Schoenberg (1952)]

$$n_s(T) = n_n\left[1 - \left(\frac{T}{T_c}\right)^4\right] \tag{5.19}$$

As the number of normal state carriers estimated from 5.19 falls rapidly below T_c, estimation of the electronic contribution to the thermal conductivity of $YBa_2Cu_3O_{7-\delta}$ simply by the application of the above equations does not lead to sufficient enhancement of the thermal conductivity to overcome the depression of the thermal conductivity caused by the addition of the electron-phonon scattering.

In the above discussion, the relaxation time for electron-electron scattering is assumed to be independent of the number of electrons which are 'available' for scattering processes. However, this assumption may not be valid. As the electrons condense into the superconducting state there will be a reduction in the population of 'normal' electrons, if a large number of carriers have condensed into the superconducting state the probability of a collision with another 'normal' electron will be reduced and there will be an enhancement of the electron-electron relaxation time. It may be postulated that this physical description is represented mathematically by a simple power law. The relaxation time relevant to electron-electron scattering is then assumed to be (in ps)

$$\begin{aligned} \tau_{ee} &= \frac{\alpha}{T^2} + \beta(T - T_c)^{\rho} \qquad T < T_c \\ &= \frac{\alpha}{T^2} \qquad T \geq T_c \end{aligned} \tag{5.20}$$

The choice of the parameters α, β and ρ is not entirely arbitrary. The parameters must be chosen such that the electron-electron relaxation time is physically reasonable. Typical relaxation times for electron-electron scattering are of the order of 10^{-14} seconds. Thus α, β and ρ must be chosen such that the overall electron-electron relaxation time is of this order.

Utilising only the phonon contribution to the thermal conductivity, the data obtained upon sample D970/3 was fitted using the parameters τ_{pp}(low T) = 35ps, τ_{pp}(high T) = 1.5ps, T_u = 30K, G = 20. Figure 5.8 shows calculated and experimental thermal conductivity curves. Utilising these parameters for the phonon contribution, the electronic contribution was then added to see if the model was capable of fitting the experimental data. Figure 5.9 shows the results obtained by performing the calculation with the addition of equations 5.8 and 5.13. The parameters used to fit the data were $\tau_{pe} = \tau_{ep}$ = 2.5ps, $d\tau_{pe}$ = 1000, α = 60, β = 4x10^{-8}, ρ = 4. The number of carriers per unit volume in the normal state was assumed to be 3x10^{27}m^{-3}, which is comparable to values found in the published literature. [Fischer et al (1988)]

The parameters relating to the electron-electron relaxation time lead to the relaxation time at 100K being 0.006ps. It is important to note that the additional

Figure 5.8 — Comparison of theoretical and experimental data for sample D970/3.
τ_{pp}(low T) = 35ps, τ_{pp}(high T) = 1.5ps, T_u = 30K, G = 20

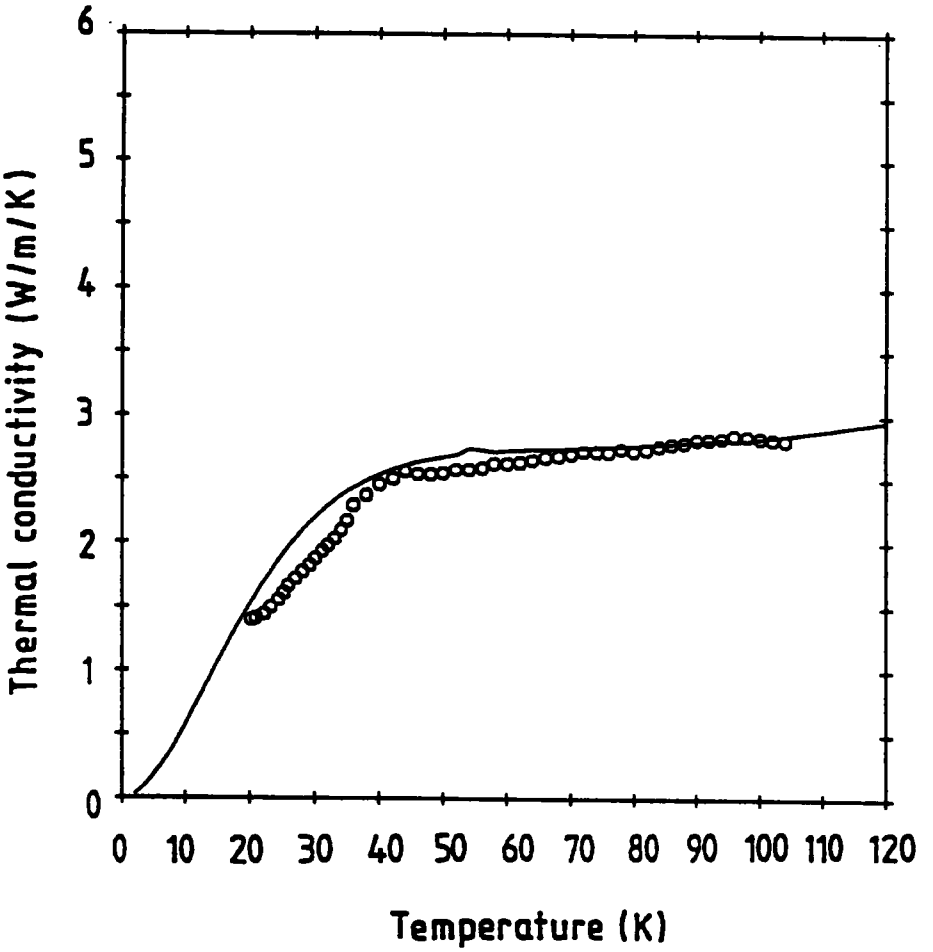
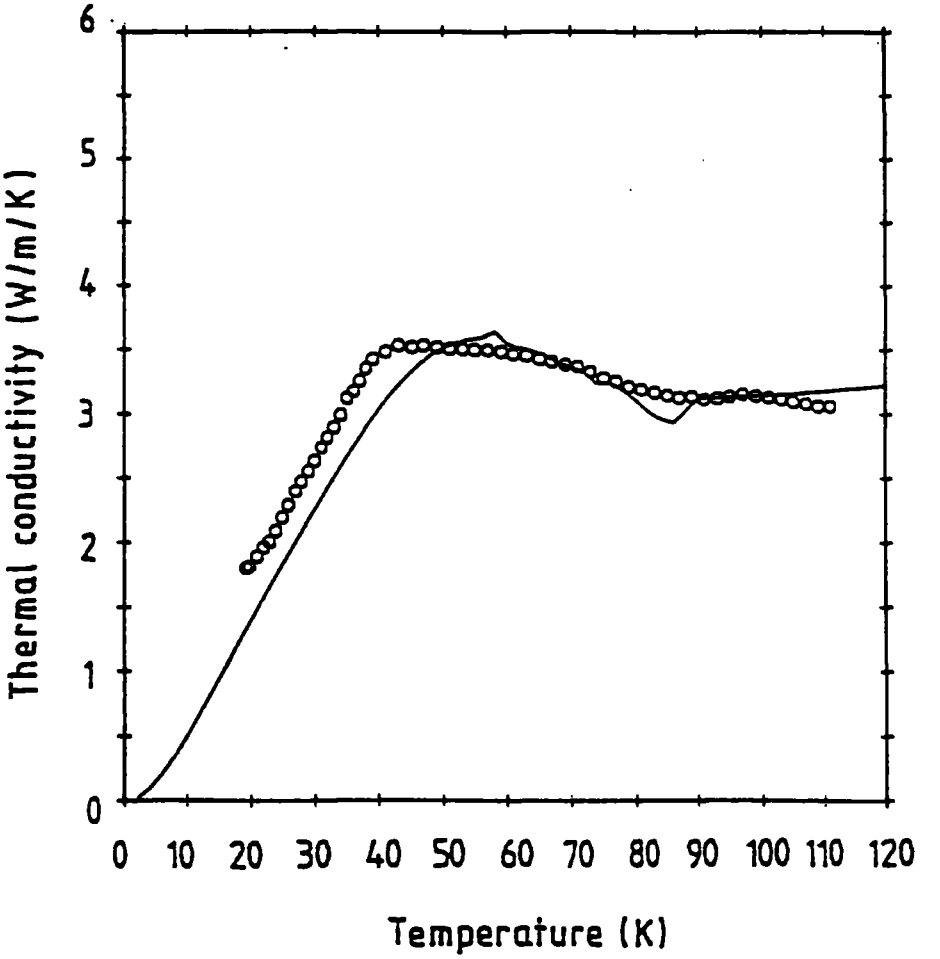


Figure 5.9 — Comparison of theoretical and experimental data for sample D970/4.
τ_{pp}(low T) = 35ps, τ_{pp}(high T) = 1.5ps, T_u = 30K, G = 20
$\tau_{pe} = \tau_{ep}$ = 2.5ps, $d\tau_{pe}$ = 1000, α = 60, $\beta = 4\text{x}10^{-8}$, ρ = 4.



term in the relaxation time is a parameterisation of a physical idea. There are a variety of theoretical ideas which may be able to explain the magnitude of the apparent carrier contribution to the thermal conductivity, some involving the idea that the carriers are bosons, holons or spinons, rather than the fermion approach adopted in this analysis. [see, for example, Mott (1990), Anderson and Ren (1990)]

Thermal conductivity measurements may be able to provide unique information about the nature of the normal state carriers below T_c. It has been suggested that as little as 10% of the normal state carriers eventually condense into the superconducting state. [Mott (1990)] Any theory which espouses this type of idea must therefore take into account the behaviour of the normal state carriers below T_c. Thermal conductivity measurements may be able to assist in the testing of such new theories.

5.8 Conclusions

Thermal conductivity measurements have been performed upon a sample of $YBa_2Cu_3O_{7-\delta}$ which has been subjected to a series of heat treatments in order to remove oxygen from the material. The measurements show that the peak in the thermal conductivity of the sample disappears as oxygen is removed and the sample becomes non-superconducting. Oxygen annealing of the sample restored the superconducting transition.

In order to assist the understanding of this behaviour, a quantitative model based upon the Boltzmann equation has been been constructed. The quantitative model demonstrates that consideration of phonon interactions alone cannot account for the behaviour of the thermal conductivity and that there must be a significant carrier contribution to the thermal conductivity in both the normal and superconducting states.

The electronic contribution estimated from ideas based around the Pauli exclusion principle has been shown to be unable to explain the large carrier contribution to thermal conductivity with the sample in the superconducting state. A physical process has been suggested which may account for the large carrier contribution below T_c. This process has been parameterised in order to produce quantitative

results. The parameters required by the theory are bound by certain physical constraints. Consequently there are no entirely free parameters which can modify the theoretical fit to an arbitrary accuracy. The agreement of experiment and theory is good and it is possible that a more sophisticated refinement of the parameters may lead to an even closer correlation between experiment and theory.

Chapter VI

Thermal Conductivity Study II
Effect of Sintering Temperature

A state without the means of some change is without the means of its conservation.

— Edmund Burke.

Reflections on the Revolution in France

6.1 Sample Preparation

The samples used in this study were produced from the same batch of calcined material used for the production of sample D970 described in §5.2. Equal masses of material were weighed and pressed into six 20mm diameter pellets at a pressure of 405MPa (4kbar). Each pellet was sintered at a slightly different temperature, D950 being sintered at 950°C, D960 sintered at 960°C and so on up to D1000. The furnace temperature was controlled using a Eurotherm 818P temperature controller fitted with a solid state relay which allowed the power input to the heating element to be continuously varied. The stability of the furnace quoted by the manufacturers is ± 2°C. This may be used as an estimate of the error on the sintering temperature. The pellets were annealed under flowing oxygen at a temperature of 450°C for 48 hours and cooled slowly at 20°C/hr. All samples were stored in a desiccator when not in use. The samples were found to be approximately 85% of the expected density of single crystal $YBa_2Cu_3O_7$.

6.2 X-ray Powder Diffraction

X ray powder diffraction was performed upon each of the six samples using the X-ray powder diffractometer described in §3.1. The results show that the materials are all primarily $YBa_2Cu_3O_{7-\delta}$ with possible trace quantities of Y_2BaCuO_5. From the observed reflections the lattice parameters of the $YBa_2Cu_3O_{7-\delta}$ phase were calculated using a standard least squares fitting algorithm, these values are given in table 6.1. The large error bounds upon the a and b lattice parameters reflect the initial uncertainty of the c axis measurements and the low number of strong reflections available from which these parameters may be determined.

Table 6.1 — Structural parameters of samples D950 to D1000

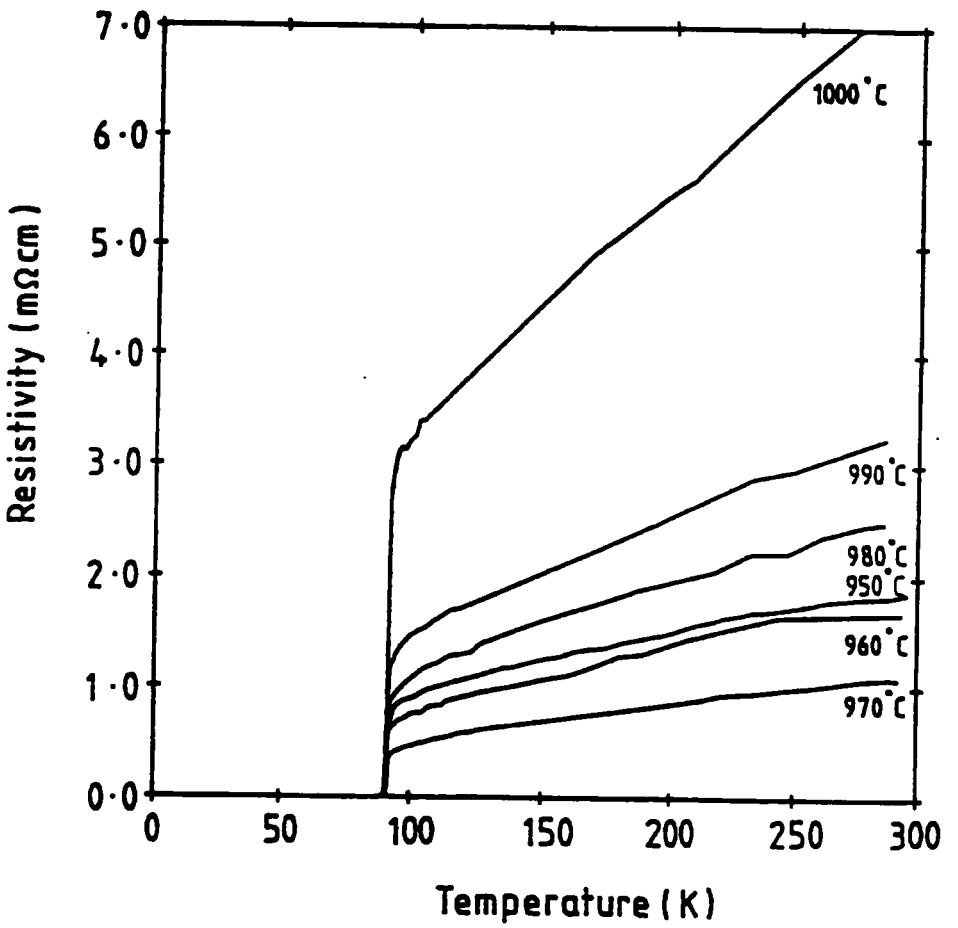
Sample	a (Å)	b (Å)	c (Å)
D950	3.83 ± 0.02	3.88 ± 0.02	11.68 ± 0.01
D960	3.84 ± 0.02	3.86 ± 0.03	11.69 ± 0.01
D970	3.84 ± 0.02	3.87 ± 0.03	11.70 ± 0.01
D980	3.85 ± 0.02	3.85 ± 0.03	11.69 ± 0.01
D990	3.85 ± 0.03	3.86 ± 0.03	11.66 ± 0.01
D1000	3.87 ± 0.03	3.87 ± 0.03	11.68 ± 0.01

6.3 Electrical Resistivity Measurements

The resistivity as a function of temperature and magnetic field was obtained for each of the six materials. The resistivity of the samples in the normal state did not alter as the applied magnetic field was increased. Figure 6.1 shows the temperature dependence of the resistivity (in zero field) for each of the samples. As the sintering temperature was increased above 950°C, the resistivity decreased slightly, reaching a minimum for sample D970. At sintering temperatures above 970°C the resistivity of the material increased considerably. None of the materials exhibited semiconducting behaviour.

The variation in the superconducting transition temperature, $T_c(50\%)$, in magnetic fields up to 500mT, is plotted in figure 6.2. Following the trend of the absolute

Figure 6.1 — Electrical resistivity of samples D950 to D1000.



resistivity, sample D970 had the highest transition temperature in low field, 91.9K. The transition temperature of each sample was found to be approximately independent of applied field in the region 0 to 50mT although T_c(50%) did fall by approximately 0.8K as the field was increased to 500mT. In contrast to the behaviour of the remaining samples, D990 showed a significant depression of T_c(50%) in very low applied fields.

The width of the superconducting transition, ΔT_c(90%-10%), in the various magnetic fields is plotted in figure 6.3. As expected, the transition becomes broader at progressively higher applied fields, reflecting the granular nature of the samples. The variation of ΔT_c(90%-10%) as the sintering temperature is raised is particularly small, showing a slight sharpening of the transition as the sintering temperature is raised.

The resistivity transitions are expected to be affected significantly by two competing effects. The sintering was performed at high temperatures, above and below the initial calcination temperature, consequently it is possible that the samples have undergone some partial decomposition. This would have the effect of producing particularly 'dirty' grain boundaries which have an off-stoichiometric composition. At higher sintering temperatures one would expect the inter-granular connections to become stronger and this would tend to decrease the overall resistivity. The low resistivity at 100K of samples D950, D960 and D970 suggests that these materials have a high overall oxygen content and are likely to have particularly 'clean' grain boundaries. The increase in resistivity for samples D980, D990 and D1000 is indicative of the materials being 'degraded' by the sintering process, resulting in either a lower volume fraction of superconducting material or a decrease in the number of paths of low electrical resistance.

The small changes in T_c and ΔT_c as a magnetic field is applied to each sample is indicative of strong inter-granular coupling throughout each material. The very small changes in T_c and ΔT_c with applied field for all of the samples is curious. If samples D980 to D1000 have suffered from a larger number of impurities at the grain boundaries than samples D950 to D970, one might reasonably expect that the superconducting transition of sample D1000 should be much more broad than that of D950, with the difference increasing with applied field. One must

Figure 6.2 — T_c(50%) of samples D950 to D1000.

○ ... <0.1 mT + ... 0.5mT △ ... 5mT

◇ ... 50mT ▫ ... 500mT

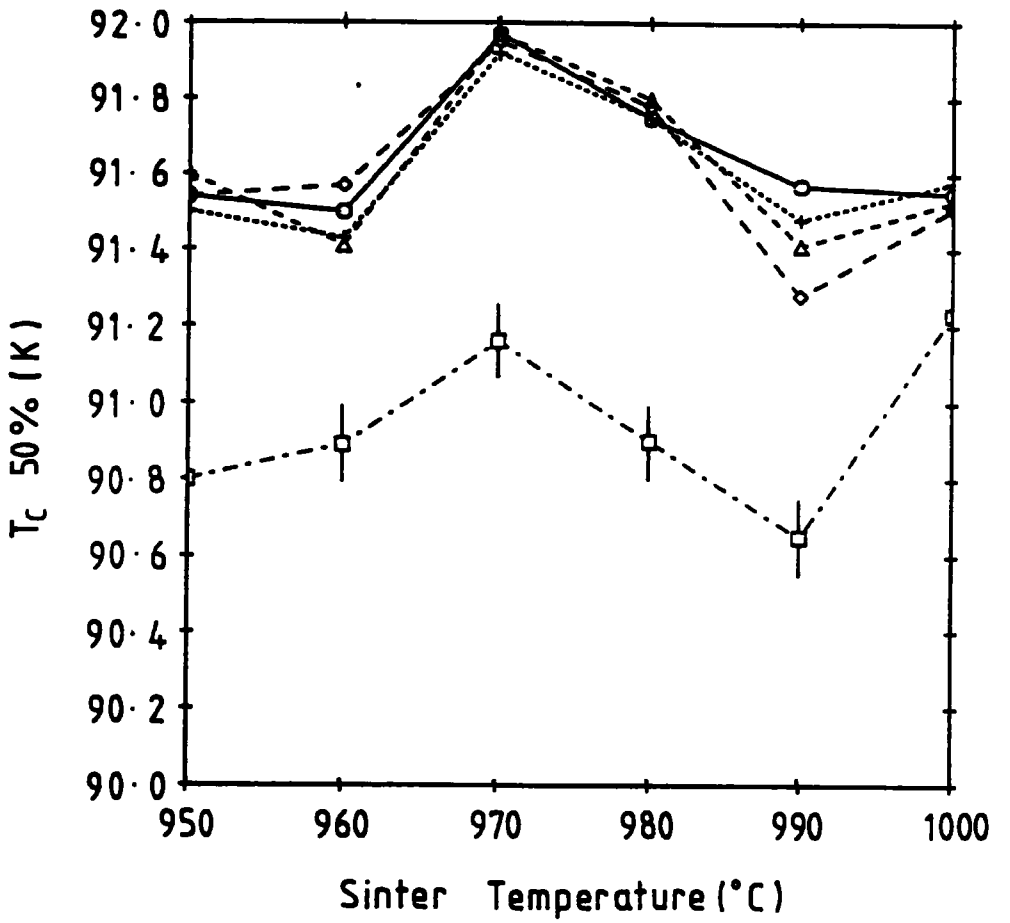
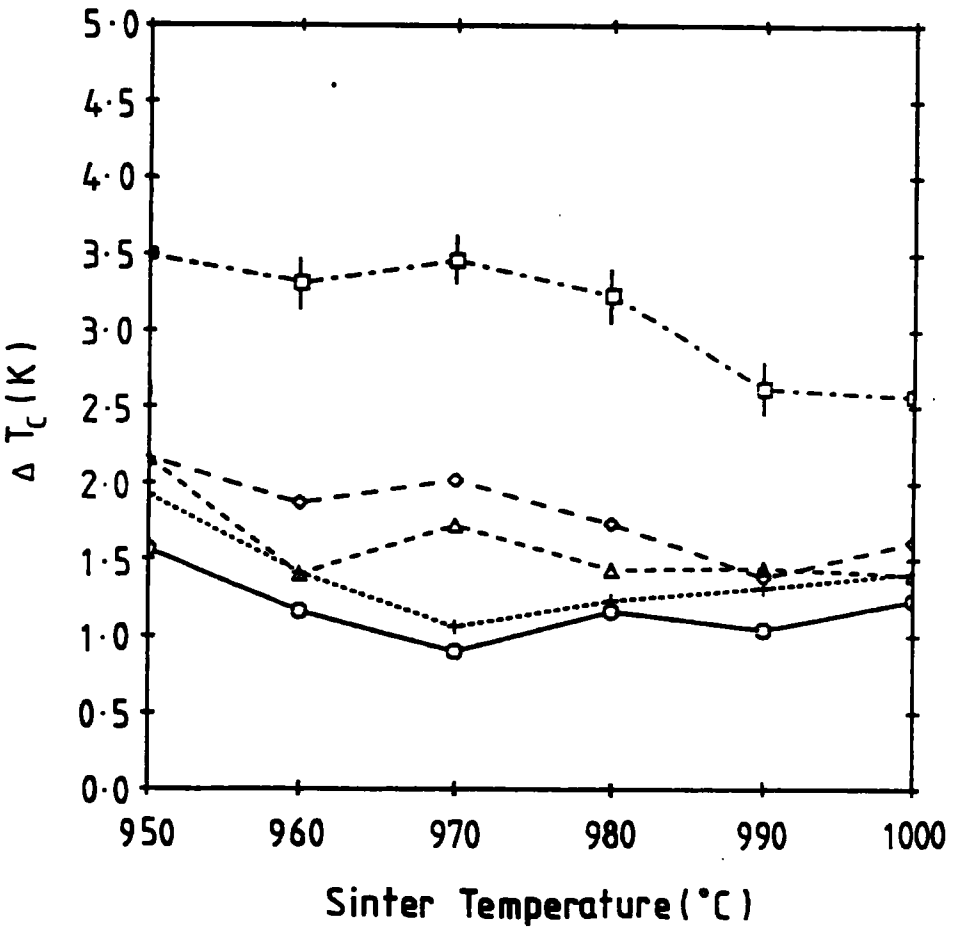


Figure 6.3 — ΔT_c(90%-10%) of samples D950 to D1000.

○ ... <0.1mT + ... 0.5mT △ ... 5mT

◇ ... 50mT □ ... 500mT



note, however, that electrical resistivity in superconducting systems is dominated by the percolation threshold [Shante and Kirkpatrick (1971)] consequently the resistivity provides information only about the strongest set of links which provide a continuous path through the material. Qualitative prediction of the depression of T_c and the broadening of the transition is difficult. For example, although sample D990 shows a lowering of T_c in low applied fields (unlike the other samples), sample D1000 may not show this effect simply because it contains a few very 'strong' superconducting paths – sample D1000 may have fewer percolative paths than sample D990, but each path may be able to support a greater electrical current than those of sample D990.

6.4 Magnetisation Measurements

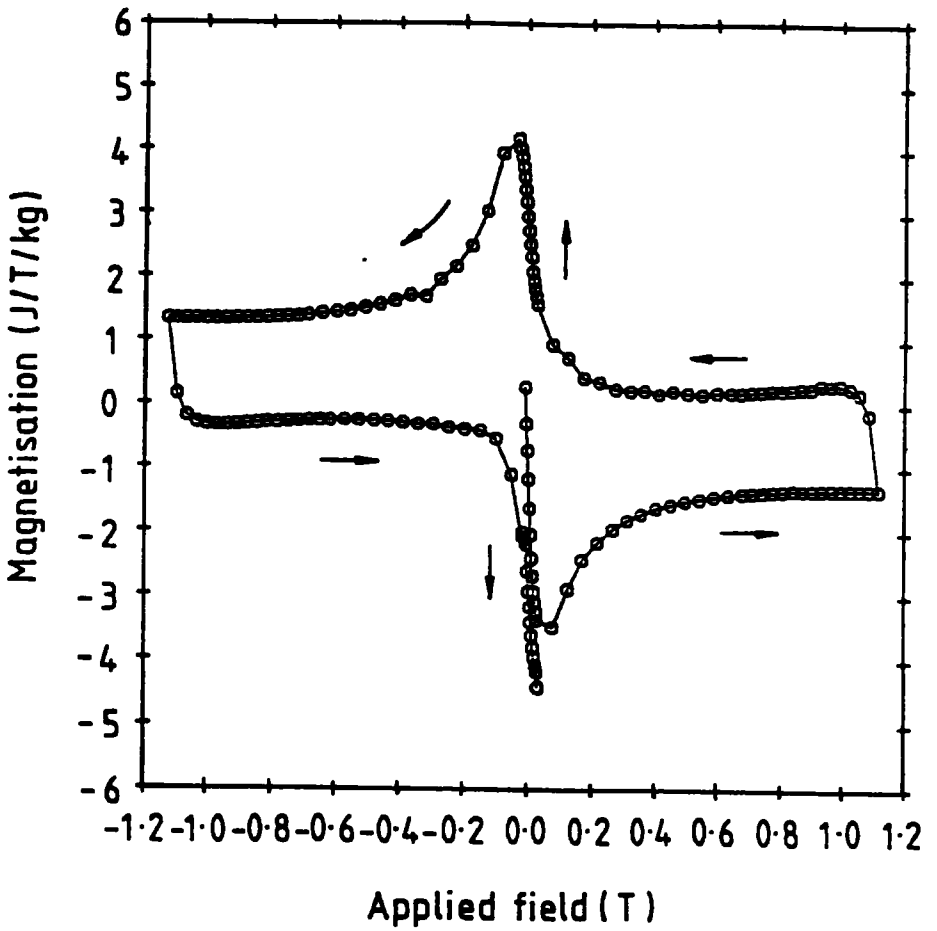
Magnetic hysteresis curves were obtained at 77K and 40K using the experimental apparatus described in §3.4. Figure 6.4 shows the magnetisation curve of sample D980 at 40K. All of the magnetisation curves measured at a given temperature were similar in form, differing only in detail as the sintering temperature was raised. Table 6.2 indicates the main parameters which could be ascertained from each magnetisation curve.

Table 6.2 — Magnetisation curve parameters of samples D950 to D1000.

	Parameter	Sample					
		D950	D960	D970	D980	D990	D1000
77K	B_{irr}(T)	0.89	0.82	0.96	0.86	1.10	0.65
	M (J/T/kg) at 1.15T	0.218	0.216	0.248	0.176	0.203	0.173
	Peak M (J/T/kg)	0.943	1.030	1.054	0.962	1.150	0.920
40K	Peak M (J/T/kg)	4.92	5.28	4.30	4.42	4.38	4.88

All samples exhibited strong diamagnetism at very low fields. At fields above B_{c1} the diamagnetic signal decreased as flux entered the sample. In all cases, at high magnetic fields(>0.6 Tesla) the magnetisation reached a value which was approximately field independent. This is characteristic of an extreme type II superconductor in the regime $B_{c1} \ll B \ll B_{c2}$. At 77K the magnetisation exhibited

Figure 6.4 — Magnetisation curve of sample D980 at 40K.
Error bars are smaller than the points.



irreversible behaviour below a field B_{irr}. The value of B_{irr} for samples D950 to D980 was similar; sample D990 had the highest value of B_{irr} whereas sample D1000 had the lowest value. The magnitude of B_{irr} is indicative of the strength of the screening currents a sample may sustain. An increase in B_{irr} may be achieved by an increase in the strength of the inter-grain coupling ('connectivity') such that the critical current, I_c, of the grain boundaries is increased. This is a possible explanation of the high value of B_{irr} of sample D990. Sample D1000 would be expected also to benefit from an increase in the connectivity, however, as indicated by the resistivity measurements, the effect of increased connectivity may be offset by a corresponding degradation of the sample.

At fields above B_{irr} the sample still exhibits a substantial diamagnetic behaviour, indicating that a significant part of the sample is in the superconducting state. Ando and Akita (1990) have suggested that as flux may freely flow within the sample, the magnetisation above B_{irr} is not dominated by bulk screening currents, hence the value of the magnetisation above B_{irr} provides an indication of the *quantity* of superconducting material. At 77K, the materials showed a slight decrease in the magnetisation at 1.15 Tesla as the sintering temperature was raised, although the value for sample D970 was significantly higher value than for the other materials. This trend is in broad agreement with the hypothesis that the samples degrade at sintering temperatures above 970°C and that sample D970 has a larger volume fraction of superconducting material than the other samples.

From the magnetic hysteresis, the transport critical current density of each material was determined by application of the Bean critical state model using the sample outer dimension as the scaling length (see §3.4). Figure 6.5 illustrates the variation of J_c(B) with sintering temperature at 77K. At 77K, no discernible trend in J_c(B) could be found as the sintering temperature increased, however the values of J_c(B) for sample D990 were significantly higher than for the other samples, reflecting the higher value of B_{irr} found for this sample. Figure 6.6 illustrates the variation of J_c(B) with sintering temperature at 40K. At this temperature J_c(B) decreased as the sintering temperature was raised from 950°C to 970°C, and increased as the sintering temperature rose from 980°C to 1000°C.

Figure 6.5 — Bean model J_c(B) of samples D950 to D1000 at 77K.

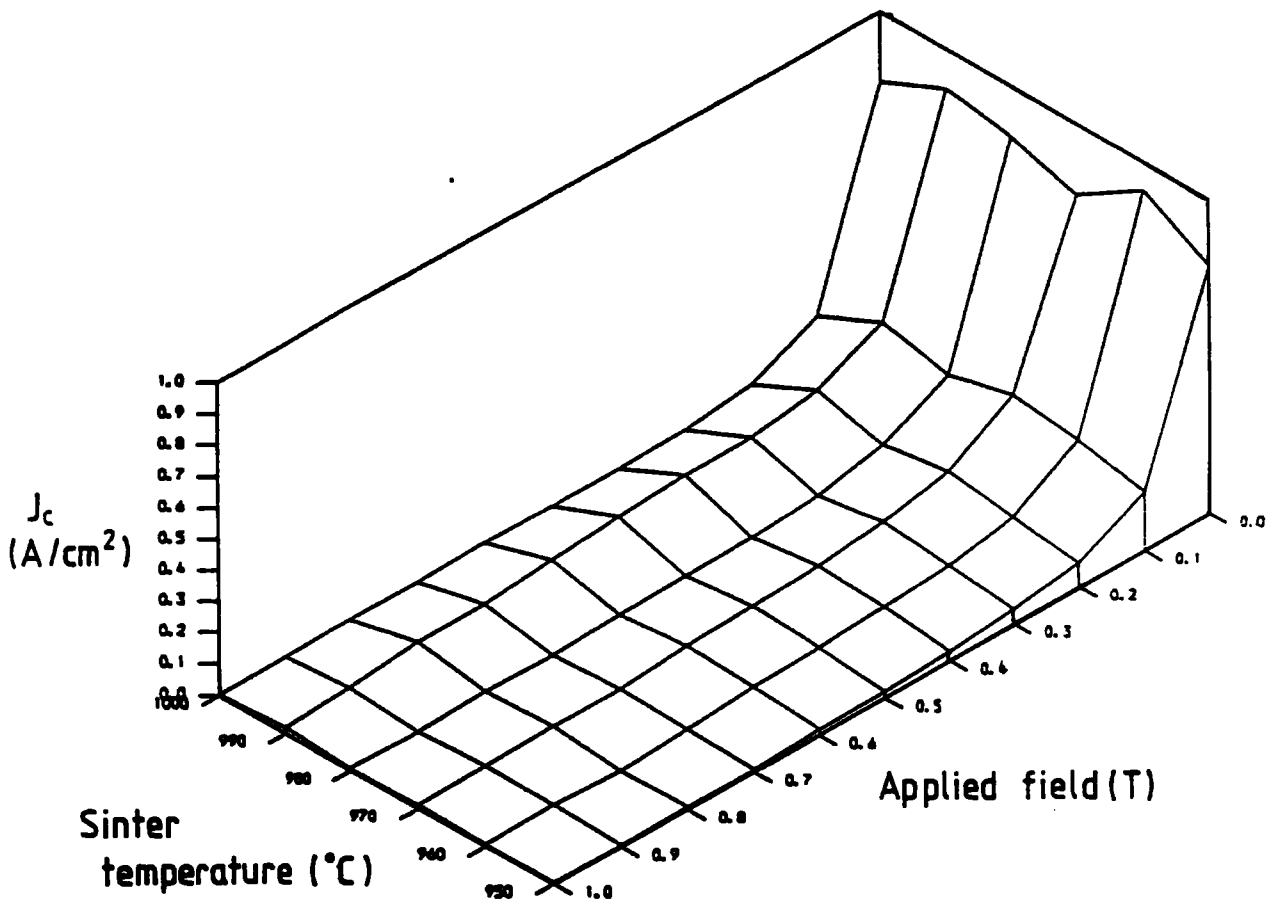
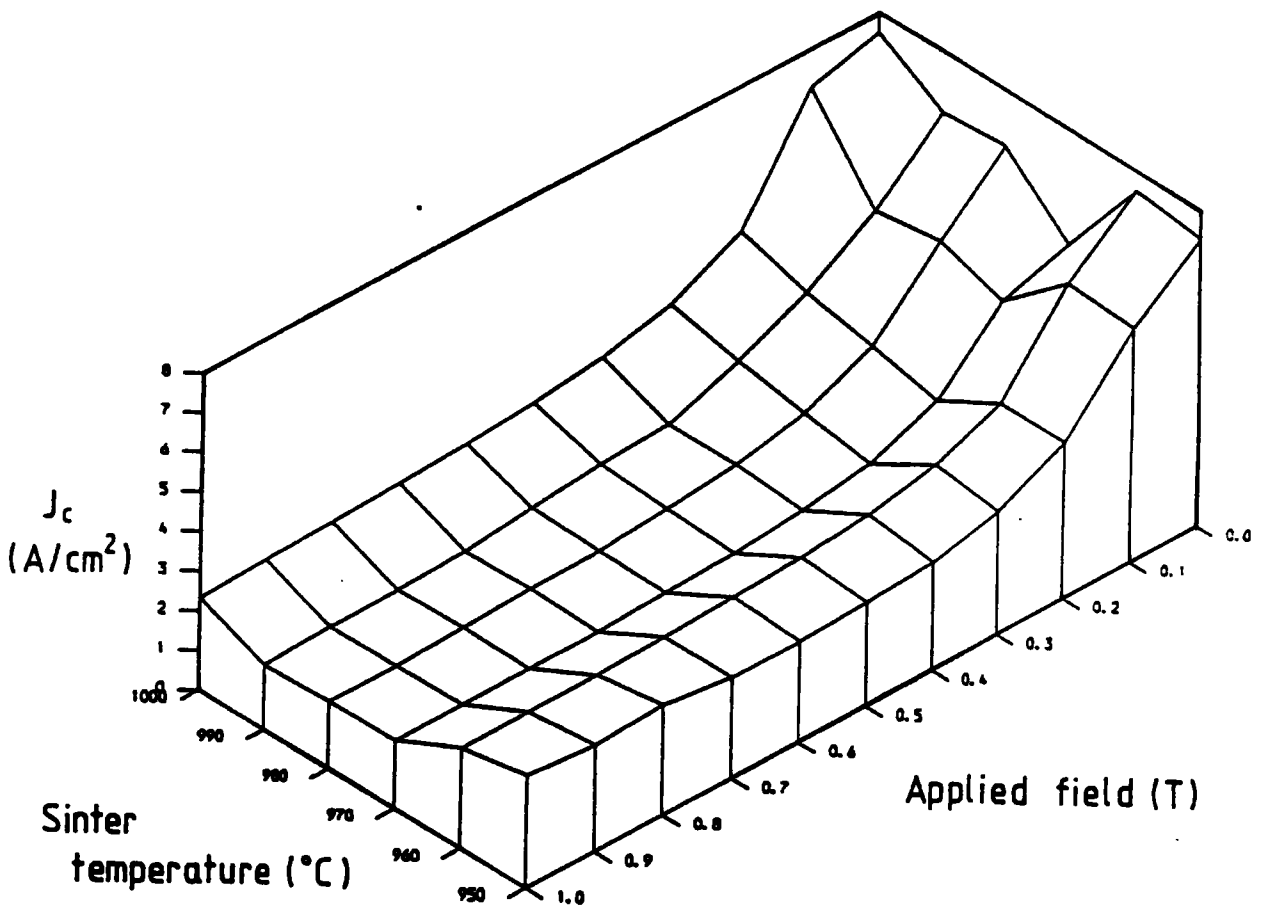


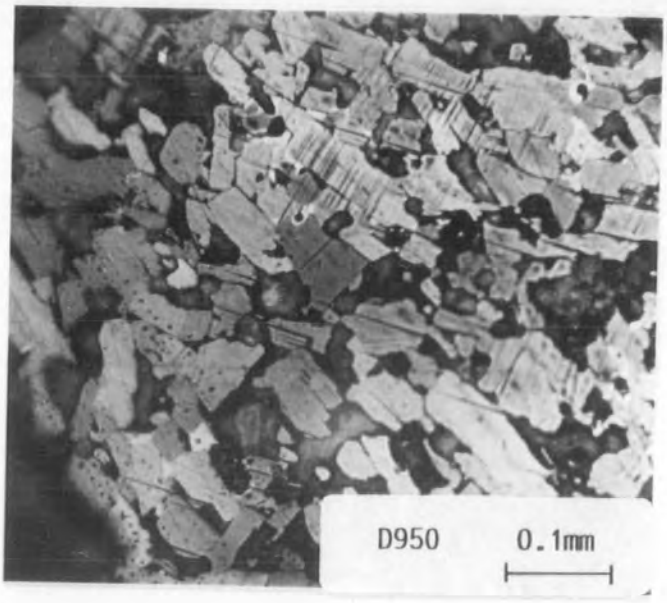
Figure 6.6 — Bean model J_c(B) of samples D950 to D1000 at 40K.

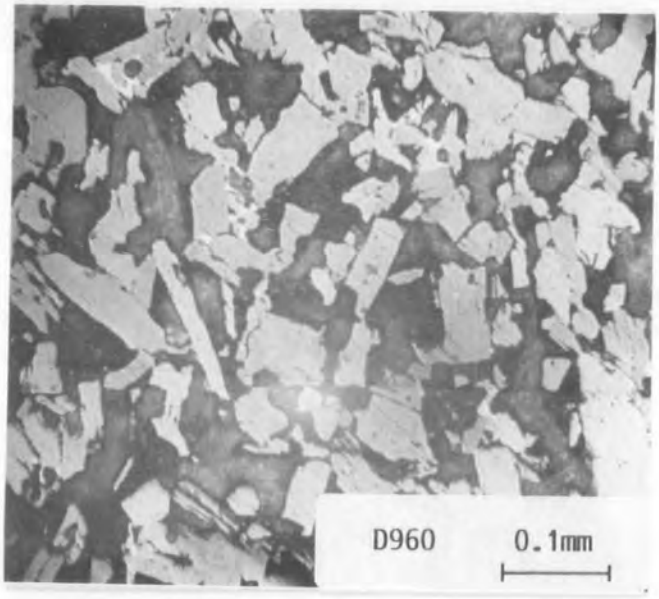


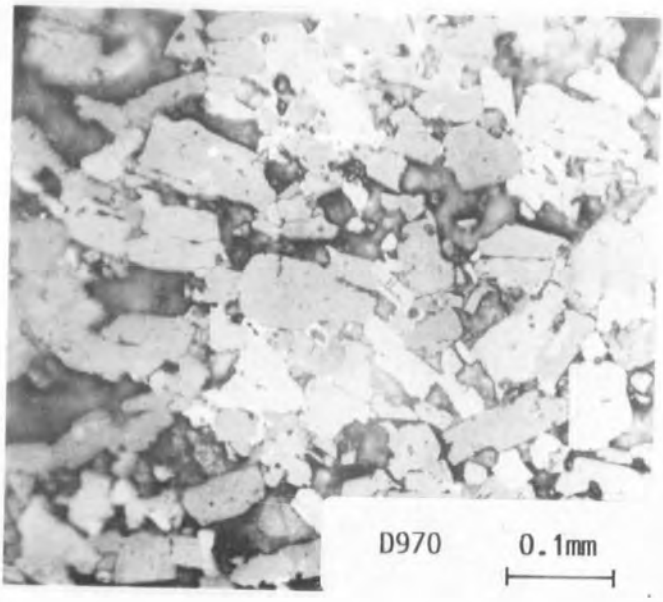
An increase in hysteresis may be explained in two ways. Firstly, if higher J_c(B) material contains a higher quantity of superconducting material, there will be a greater apparent flux expulsion from the material, leading to a higher magnitude of the magnetisation. If, in addition to this, there are are many sites which can strongly pin flux most of the flux lines within the sample will be pinned. Hence upon reducing the applied field the sample will exhibit a higher hysteresis than a material with a lower quantity of superconducting material. A second possible explanation of an increase in the hysteresis is that in the lower J_c(B) material, there are insufficient pinning sites to pin the flux lines. In this latter case, a higher number of pinning sites of sufficient strength will lead to an increase in the observed hysteresis. It is not possible to state unequivocally which of these two explanations is appropriate for the samples in this study.

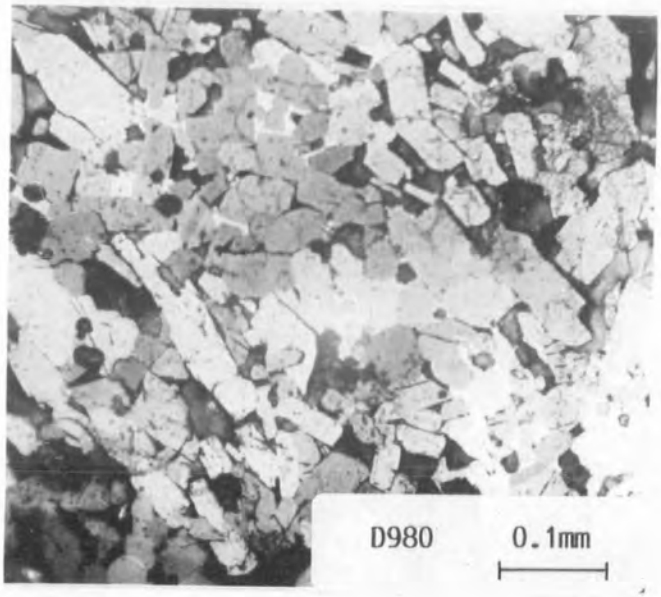
6.5 Optical Microscopy

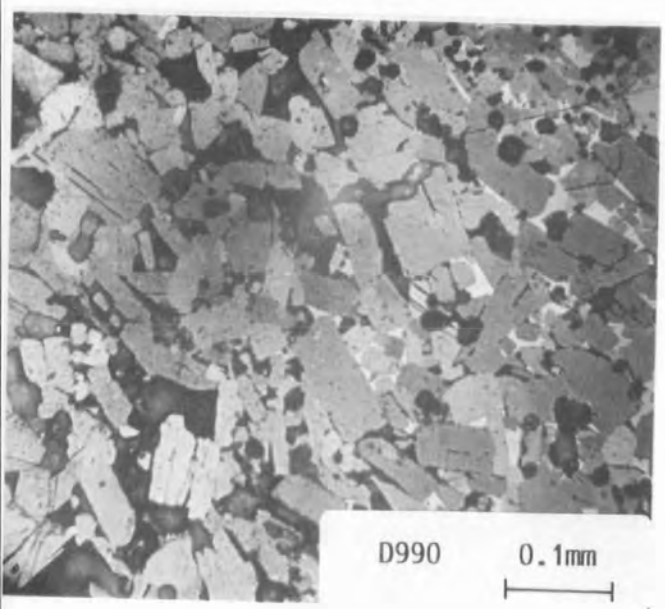
Optical microscopy was performed on each of the specimens used for the resistivity measurements. The sample surfaces were prepared by abrading with fine grade emery paper followed by polishing using successive grades of Metadi diamond lapping compound (9μm, 6μm, 3μm and 1μm). The polished surfaces were washed with methanol. The samples were viewed under an optical microscope using polarised light reflected from the sample. No evidence for the green, Y_2BaCuO_5 phase was found. The optical micrographs obtained are shown in figures 6.7 and 6.8. Moving the sample underfocus verified that the dark, out of focus patches visible in the micrographs were voids in the material. The presence of voids is consistent with the porosity of each sample although some voids may have arisen by grains being pulled out of the material during the preparation of the surface. The small quantity of light coloured material may be attributed to $BaCuO_2$. The grain size varies widely throughout each sample, making the determination of an 'average' grain size difficult. However, it is clear that no significant variation in grain size or distribution occurs as one progresses from sample D950 to D1000. The optical micrographs show that, despite the high sintering temperature, no dramatic change in microstructure takes place.

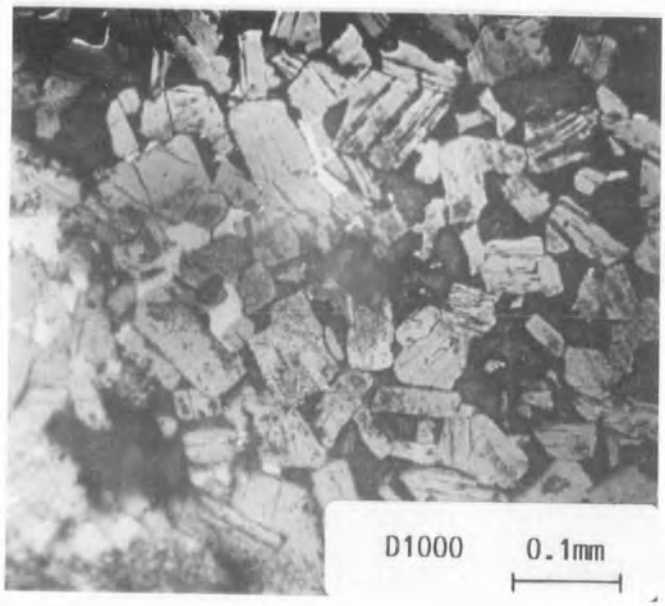

D950
0.1mm

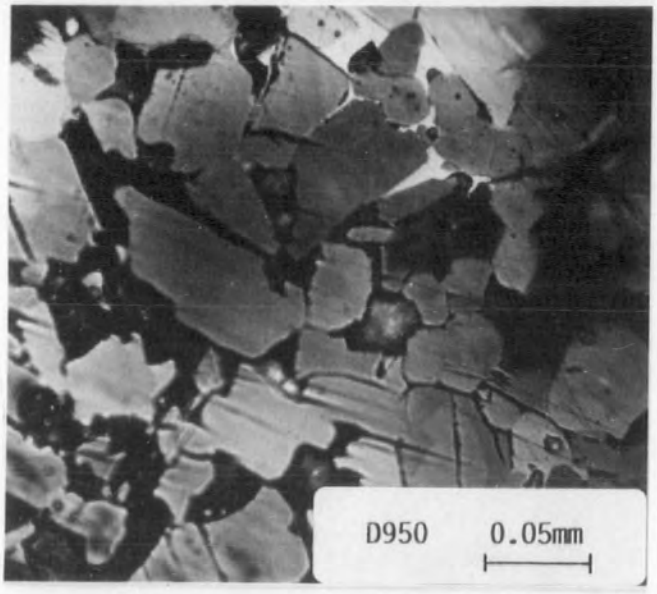

D960
0.1mm

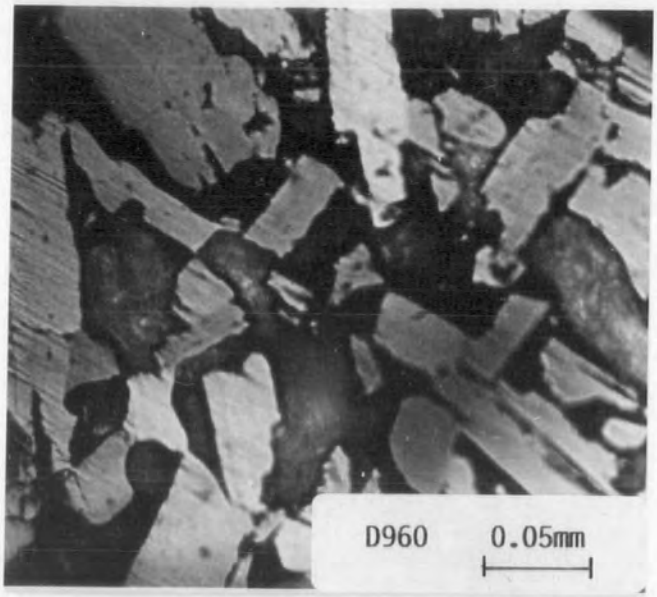

D970
0.1mm

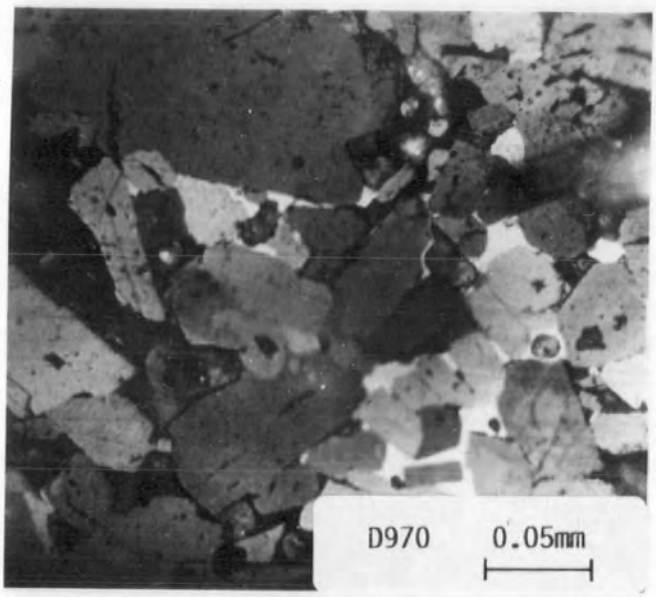

D980
0.1mm

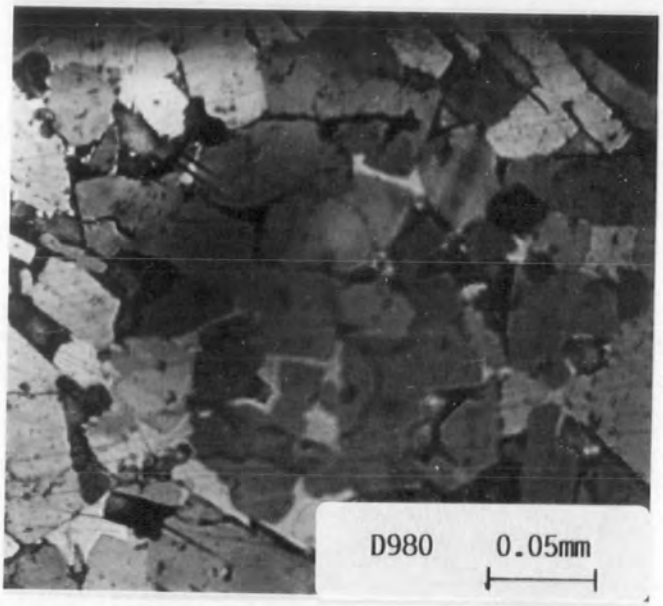

D990
0.1mm

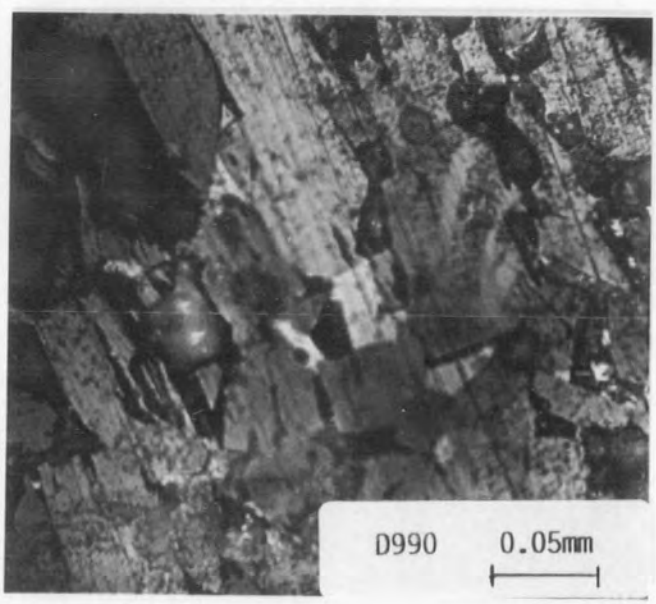

D1000
0.1mm

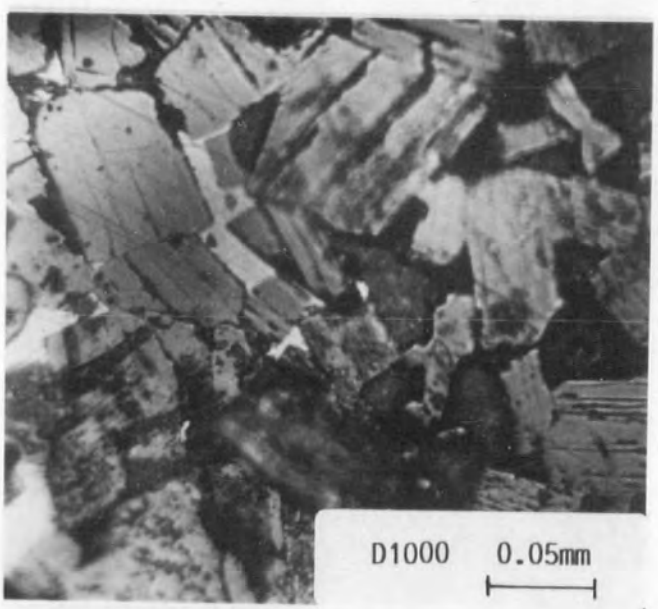

D950
0.05mm


D960
0.05mm


D970
0.05mm


D980
0.05mm


D990
0.05mm


D1000
0.05mm

6.6 Thermal Conductivity Measurements

The thermal conductivity of samples D950 to D1000 was measured using the experimental apparatus described in chapter IV. The most significant source of uncertainty in the values obtained was the accuracy to which the dimensions of each sample could be measured, in particular the separation of the thermocouple wires (approximately 0.3mm in 6mm). For a series of measurements performed upon the same sample, a measurement error of this sort is reflected equally in all measurements and is less important than for measurements made upon different samples. When all of the uncertainties are considered, the thermal conductivity results given in this chapter are estimated to be within 8% of the true value.

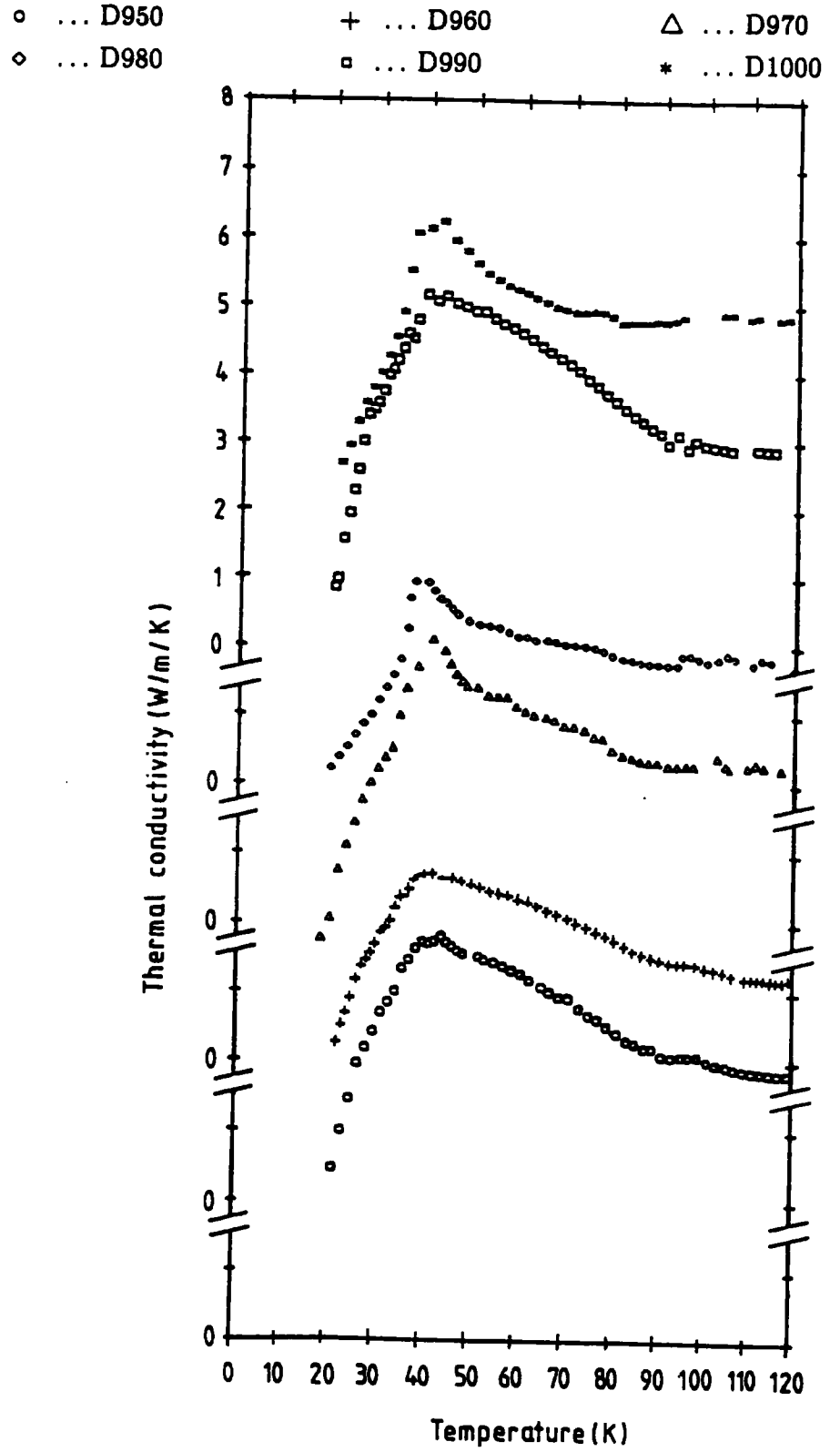
Figure 6.9 illustrates the variation in thermal conductivity as a function of temperature for each of samples D950 to D1000. All of the curves show an increase in the thermal conductivity upon passing through the superconducting transition, reaching a peak at approximately 45K before falling sharply. The samples show a general increase in the normal state thermal conductivity as the sintering temperature is raised (see table 6.3). In contrast to the trend in the normal state, the thermal conductivity of the material at 60K is much more scattered, remaining very roughly constant with increasing sintering temperature, although the value for sample D990 is significantly higher than all of the others. Upon further cooling, the thermal conductivity of each sample reaches a maximum at approximately 45K, again there is a general increase in the value of this 'peak' thermal conductivity– with sample D990 having the highest value.

Table 6.3 — Thermal conductivity of samples D950 to D1000 at selected temperatures.

	D950	D960	D970	D980	D990	D1000
κ(peak)	5.8	4.6	6.1	5.0	7.1	6.2
κ(100K)	4.1	3.2	4.3	3.8	5.0	4.9
κ(60K)	5.3	4.3	5.1	4.1	6.7	5.1

A simple explanation of the thermal conductivity data is made difficult by the effect of a change in sample connectivity. The data of chapter V suggests

Figure 6.9 — Thermal conductivity of samples D950 to D1000.



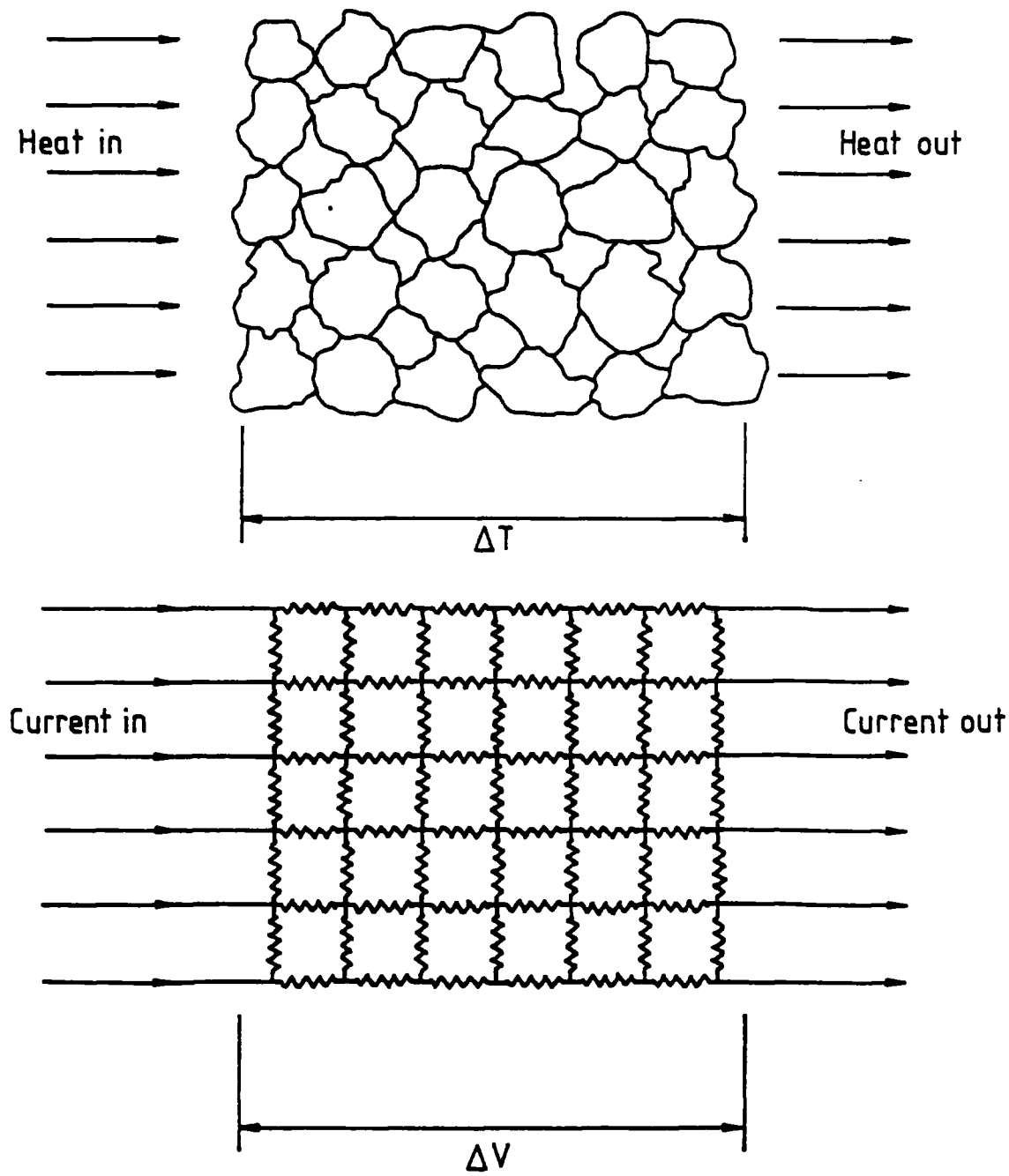
that regions of the sample which exhibit a superconducting transition will have a higher thermal conductivity, above and below T_c, than any regions of the sample which are oxygen deficient. One might expect that an increase in the normal state thermal conductivity, and the presence of an enhancement of the thermal conductivity below T_c, would suggest that a larger quantity of the material has a high oxygen content. However, as discussed previously, the strength of the inter-granular coupling is expected to increase as the sintering temperature is raised. Unlike the electrical and magnetic data, the thermal conductivity will probe both the superconducting and non-superconducting regions of the material and will not be dominated by the percolation threshold. In the absence of a significant increase in the connectivity, analysis of these results using only the information obtained from the measurements of chapter V would suggest that samples D990 and D1000 have a much higher volume fraction of superconducting material than samples D950 to D980. This is in stark contrast to the picture presented by the other characterisation techniques which indicate that sample D1000 has degraded and has a relatively low volume fraction of superconducting material. This information suggests that an explanation of the thermal conductivity data will be incomplete without at least a rough estimate of the effect of increasing the inter-granular coupling.

6.7 Model for Connectivity

An elementary simulation of the effect of increasing the strength of the inter-granular coupling has been produced. The computer simulation is based upon the assumption that a polycrystalline sample may be described as a matrix of individual grains. This matrix of grains is simulated by constructing a network of electrical conductances, one electrical conductance corresponding to the thermal conductance between two grains (see figure 6.10). The temperature of each grain is represented in the simulation by a node voltage and the heat flow between grains by an electrical current.

As the temperature of the sample is cooled below T_c, an increasing number of the inter-granular links will adopt a value appropriate to the superconducting state, this may be simulated by altering the ratio of the number of 'superconducting' links to 'normal' links. At temperatures much less than T_c, all of the links will be in

Figure 6.10 — Construction of the simulation for connectivity.



the superconducting state, whereas above T_c, all links will be in the normal state. Obviously the conductance of the links will vary with temperature and will differ in the superconducting and normal state values. This effect may be modelled, very approximately, from the experimental values of the thermal conductivity in both the normal state and at temperatures well below T_c. In the normal state the links were assigned a conductance which was constant with temperature, whereas the conductance of the superconducting links was described by a simple linear function.

The connectivity of the simulation is determined by eliminating a certain number of conductances, mimicing the effect of weak inter-granular coupling. Thus a connectivity of 0.9 is modelled simply by removing 10% of the total number of links. By keeping the number of node voltages (and therefore grains) constant and altering the number of links available for current flow, the effect of altering the connectivity may be modelled.

Once each network was constructed, the overall conductance was determined by imposing a voltage drop across two opposite sides of the network. Kirchoff's laws for electrical current were applied to each node in turn and the electrical potential of each node altered such that the net current flow into the node was zero. Alteration of the each node voltage was repeated until a self-consistent solution was found. In order to produce a realistic model of the material it is obviously necessary to use a large conductance network. The solution of large matrix equations (up to 50x50x50 nodes) using a reasonable amount of computing time is non-trivial. A detailed discussion of the solution of such equations is given in appendix B.

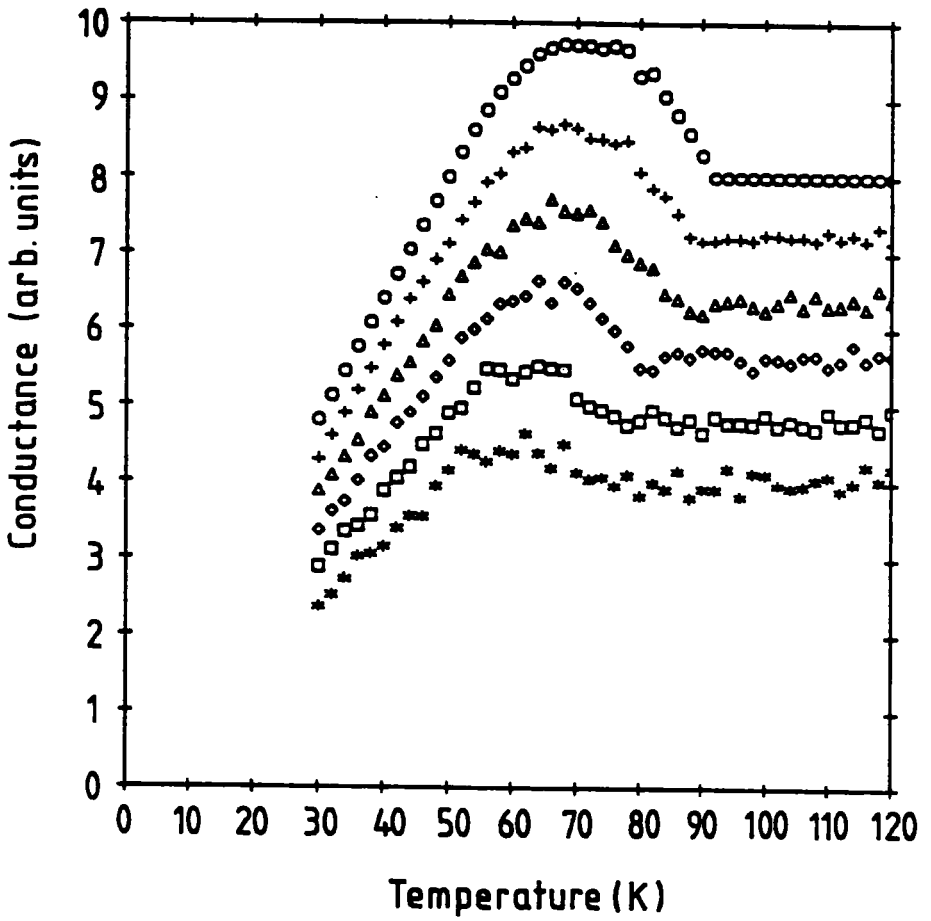
The results of the simulation for different values of the connectivity are shown in figure 6.11. As the connectivity of the system is decreased, the thermal conductance of the network is lowered in the normal and superconducting states and the peak in the thermal conductivity data becomes less distinct. The results of the simulation show that an increase in the connectivity of 20% will cause a corresponding increase in the normal state thermal conductivity, yet the magnitude of the thermal conductivity peak will not appear significantly sharper. The peak in the thermal conductivity only becomes 'washed out' when the connectivity is

Figure 6.11 — Effect of decreasing connectivity.

Connectivity parameters

○ ... 1.00 + ... 0.90 △ ... 0.80

◇ ... 0.70 ▫ ... 0.60 * ... 0.50



reduced dramatically, hence it is reasonable to suppose that the increase in the normal state thermal conductivity of samples D990 and D1000 may be caused in part by the increase in the connectivity of the material, rather than a dramatic increase in the volume fraction of superconducting material.

6.8 Conclusions

A qualitative analysis of the relationship between the electrical resistivity, transport critical current density and thermal conductivity has been performed. Quantitative analysis of the thermal conductivity data is extremely difficult owing to the presently poor state of the understanding of the influence of the various processes which contribute to the thermal conductivity of a polycrystalline sample. However, by utilising the complementary information from the electrical resistivity and magnetic measurements, a reasonably coherent picture begins to emerge from the data. As the sintering temperature increases, there is an increase in the strength of the inter-granular coupling. This effect is most pronounced at sintering temperatures above the initial calcination temperature of the precursor materials. The increase in the connectivity manifests itself most clearly in the thermal conductivity data.

Chapter VII

Thermal Conductivity Study III
Cookson Group Materials

The question which science typically asks is not 'What is it reasonable beforehand to suppose ?' but rather 'What have we evidence to think is actually the case ?'

— John Polkinghorne.
One World

7.1 Introduction

A number of samples prepared by Cookson Group Central Research were made available for study. The purpose of this investigation was to identify whether or not the thermal conductivity of the material would be a useful characterisation tool, and whether it is possible to extract information about the superconducting nature of the material which is unobtainable from other, more conventional characterisation methods. The work presented in this chapter was conducted under the supervision of Dr T.P.Beales at Cookson Central Research, Oxford. Some of the material preparation and characterisation work was performed by Mr J.Rogers and the Cookson Analytical Laboratory.

7.2 Material Preparation

$YBa_2Cu_3O_{7-\delta}$ powder was prepared by Cookson Central Research using a solid state reaction route from precursor oxides. Preliminary analysis of this calcined powder was conducted by Cookson Analytical Services. X-ray powder diffractometry indicated that the material consisted of primarily $YBa_2Cu_3O_{7-\delta}$, with $BaCuO_2$

and Y_2BaCuO_5 present at the 1% to 2% level. X-ray fluorescence analysis revealed that the material deviated slightly from optimum stoichiometry, the ratio Y:Ba:Cu ratio being 1:1.949:2.848 i.e. the material was approximately 2.5% barium deficient and 5% copper deficient. Before sintering, the calcined powder was ground and the particle size determined by sedigraph analysis to be $\simeq 2\mu$m in diameter.

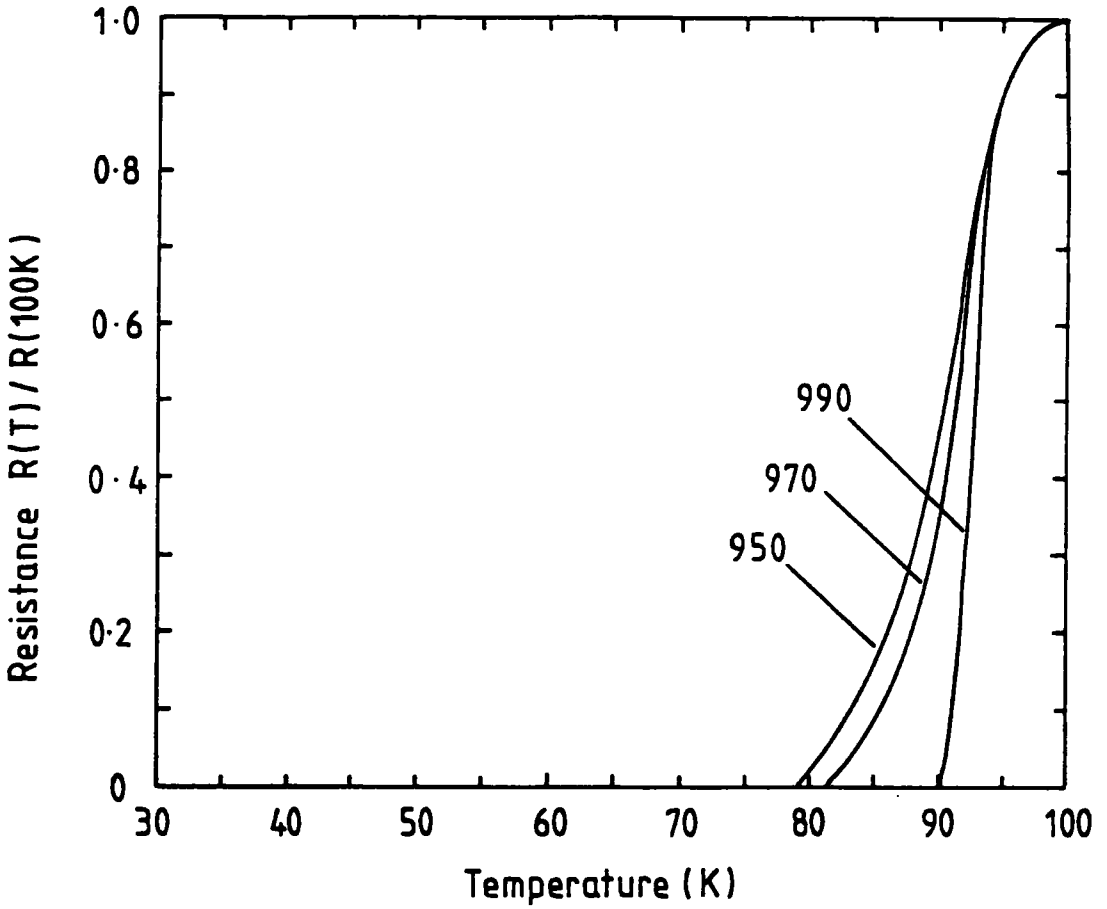
Samples of the calcined powder were weighed out and pressed into 19mm diameter pellets using a stainless steel die and hydrostatic bench press. Three pellets were sintered for 4 hours, in air, at nominal furnace temperatures of 950°C, 970°C and 990°C, followed by a slow cool at 60°C/hr to room temperature, these samples were labelled HM2-950, HM2-970 and HM2-990 respectively. The sintered pellets were heated to 400°C in flowing oxygen for 4 hours and slowly cooled to room temperature. The resulting pellets had a density of 90% of the theoretical density of single crystal $YBa_2Cu_3O_7$. X-ray powder diffraction experiments at Durham indicated that the the high temperature of the sintering process had not significantly altered the composition of any of the pellets, and that they were still primarily $YBa_2Cu_3O_{7-\delta}$.

7.3 Electrical Resistance Measurements

The superconducting transition temperature of each sample was monitored by electrical resistance measurements which were performed by Mr J.Rogers at Cookson Central Research using a closed cycle cryostat. Each sample had approximately the same electrical resistance at room temperature (1.3mΩ). Figure 7.1 shows the normalised electrical resistance as a function of temperature in the vicinity of the superconducting transition. Sample HM2-950 had a very broad transition with an onset temperature of 92K, not reaching the zero resistance state until approximately 81K. As the sintering temperature was increased the superconducting transition sharpened considerably.

As discussed in §6.3, the resistance transition is dominated by percolation effects such that the electrical current will flow through the path with the least electrical resistance. Consequently the sharpening of the resistance transition of the samples as the sintering temperature is increased may be due to *either* an increase in the connectivity of material with the 90K transition temperature *or* an increase in the actual volume of the high transition temperature superconducting material.

Figure 7.1 — Normalised resistance transition of HM2 samples.



The electrical resistance transition alone cannot be explain unambiguously the change in the properties of the material as the sintering temperature is raised.

7.4 Magnetisation Measurements

Magnetic hysteresis curves were obtained for each sample at 72K using the apparatus at Durham described in §3.4. As expected, all of the samples showed partial flux penetration and irreversible behaviour at low fields (see figure 7.2) and no evidence of ferromagnetic impurities. Utilising the Bean critical state model described in §3.4 the intra-grain critical current density was obtained from the magnetic hysteresis and the sample dimensions. The critical current density of the material at 72K, as a function of applied magnetic field up to 0.6 Tesla is shown in figure 7.3.

Sample HM2-950 had the lowest critical current density of the three samples. As the sintering temperature was increased, the magnetic hysteresis and therefore the critical current density increased. In §6.4 it was noted that this behaviour can be explained in two ways: if, as the sintering temperature increases, there is a larger flux expulsion due to a change in the amount or distribution of superconducting material and if a large fraction of the flux lines are pinned, then the measured hysteresis at low fields is dominated by the magnitude of the flux expulsion. As an increase in the number of pinned flux lines also leads to a larger hysteresis, an increase in the pinning site density will also increase the apparent critical current density.

Table 7.1 — Magnetisation curve parameters for HM2 samples.

	HM2-950	HM2-970	HM2-990
Peak M (J/T/kg)	0.58	0.77	1.24
M(J/T/kg) at 1.1 T	0.109	0.140	0.263
B_{irr} (T)	0.82	0.94	1.15

The value of B_{irr} obtained for the samples decreased as the sintering temperature was raised (table 7.1). This trend was followed by the magnetisation at 1.1 Tesla (a field well above B_{irr}). Above B_{irr}, each material still exhibits a strong

Figure 7.2 — Magnetisation curve of HM2-970 at 72K.
Error bars are smaller than the points.

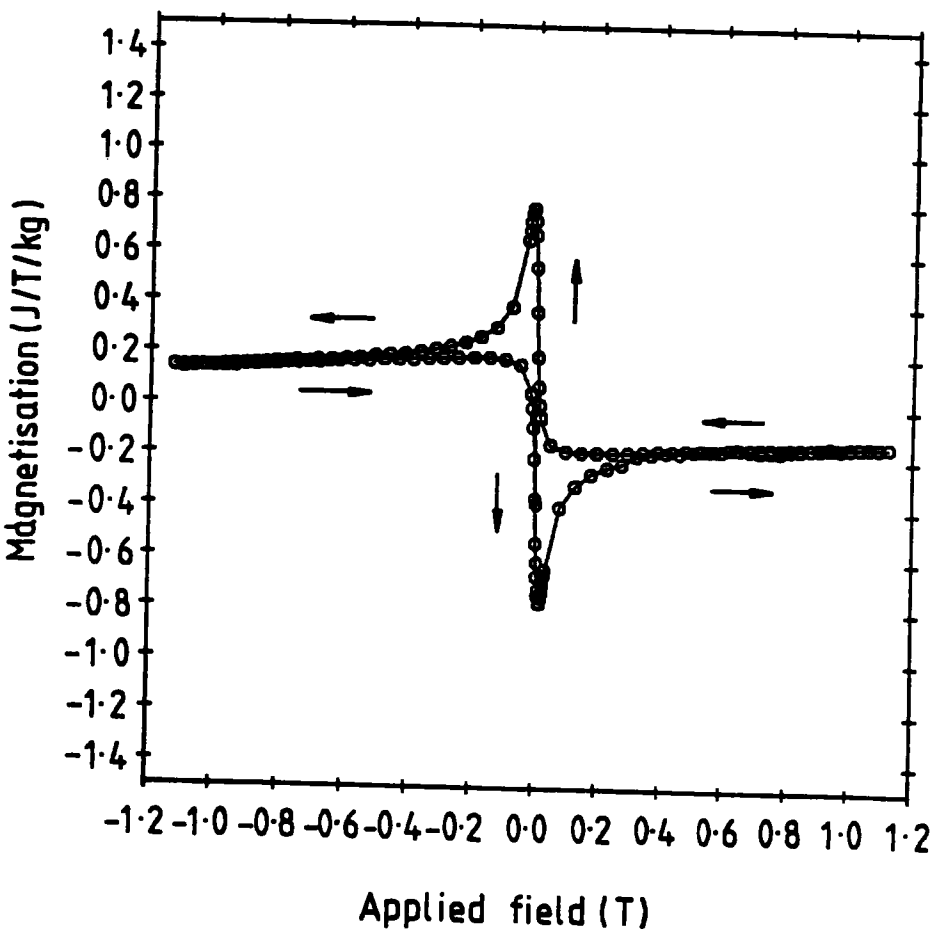
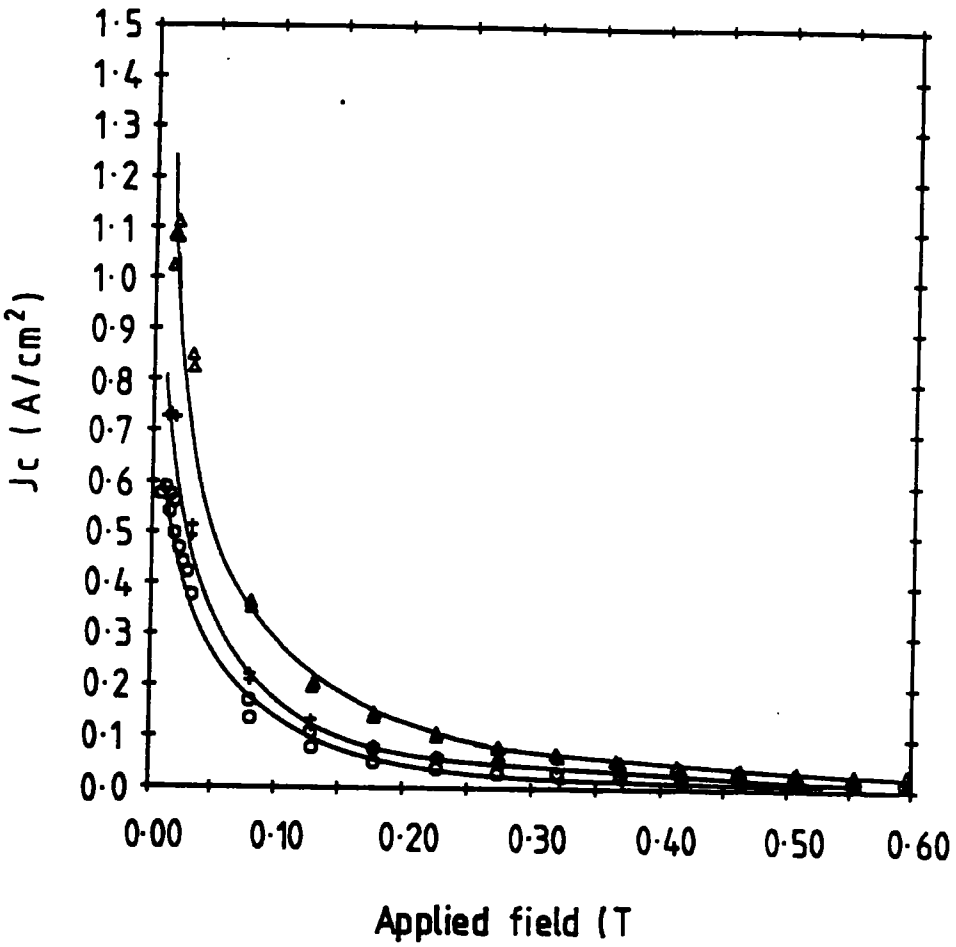


Figure 7.3 — Bean model J_c(B) of HM2 samples.

o ... HM2-950 + ... HM2-970 △ ... HM2-990



diamagnetic signal which suggests that some of the material is in the superconducting state, despite the ability of flux to flow freely through the sample. Ando and Akita (1990) claim that that the magnetisation above B_{irr} is indicative of the quantity of the superconducting material present, although this information relates only to sections of the sample which have B_{c2}(T) above that of B_{irr}(T). Upon lowering the temperature of the sample below T_c, thermal depinning of the flux lines will become less effective (allowing a greater transport current to flow) and B_{c2}(T) of each region of the sample will increase. At temperatures well below T_c, these two effects combine to increase B_{irr} beyond the limits of presently available experimental equipment. Magnetisation data may be able to provide some indication of the volume fraction of superconducting material close to T_c, however such data cannot be obtained from magnetisation data at temperatures well below T_c and in low magnetic fields.

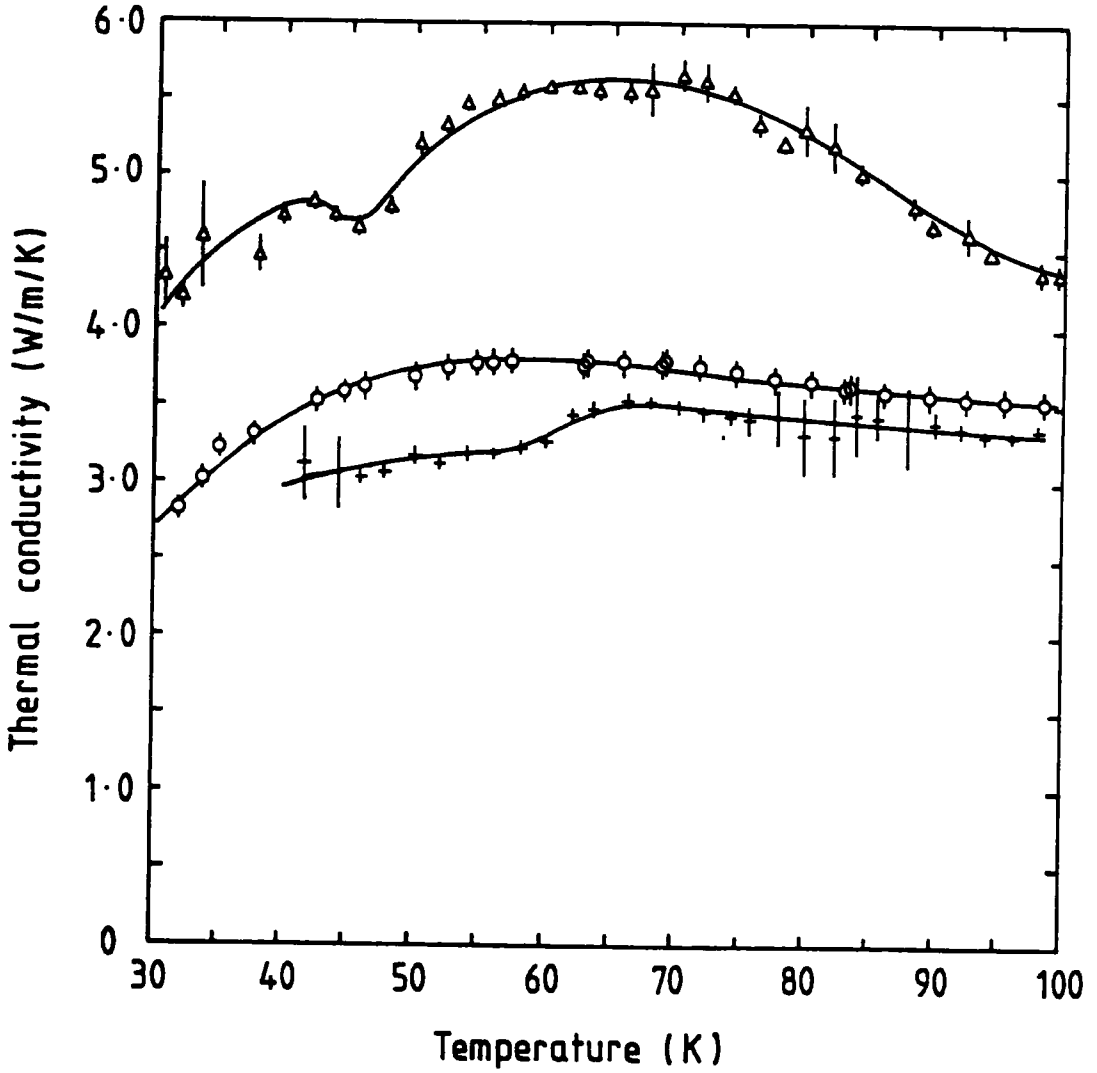
7.5 Thermal Conductivity Measurements

The thermal conductivity of the samples as a function of temperature in the region 30K-100K is shown in figure 7.4. The most striking feature of the curves is that over the full temperature range the thermal conductivity of HM2-990 is considerably higher than that of the samples sintered at lower temperatures.

HM2-970 shows has only a very slight rise in the thermal conductivity at about 80K (corresponding to the zero resistance state). The curve for the HM2-950 is similar to HM2-970, with a shallow peak suggesting that only a small fraction of the material undergoes the transition to the superconducting state. The observed rise in the thermal conductivity is most pronounced in HM2-990, there is also evidence of a small secondary peak at 45K in this sample. Whilst this peak was reproducible on this sample, no evidence for the existence of such a peak has been observed in any of the other materials measured, either supplied by Cookson group, or as prepared at Durham. It is therefore unlikely that such a peak is caused by experimental error. The peak may be intrinsic to the superconducting phase of the material although the absence of such a feature on any of the other superconducting materials measured renders this explanation unlikely. A plausible explanation of the peak is that it is caused by the presence of a second phase of material or a substantial quantity of $YBa_2Cu_3O_{7-\delta}$ which has a low oxygen content. The

Figure 7.4 — Thermal conductivity of HM2 samples.

o ... HM2-950 + ... HM2-970 △ ... HM2-990



presence of low oxygen content $YBa_2Cu_3O_{7-\delta}$ would be detected by neither the electrical resistance nor the magnetisation measurements.

One possible explanation of the behaviour of the thermal conductivity of the samples above and below T_c is that there is a dramatic increase in the connectivity of the material when the sintering temperature is raised from 970°C to 990°C with a concomitant increase in the amount of superconducting phase present. Such an explanation is consistent with the sharpening of the resistance transition of the material as the sintering temperature is raised. This explanation also suggests that the increase in the magnetic hysteresis of the samples as the sintering is increased is caused by the increase in volume fraction of the superconducting material providing a larger initial flux expulsion from the material.

7.6 Conclusions

The electrical resistance transition of the HM2 samples was observed to sharpen as the sintering temperature of the material was raised. From the resistance data alone, it is not possible to determine whether this is caused by an increase in the volume fraction of 90K transition temperature material or by an increase in the strength of the inter-granular connections. In addition, no information about the superconducting material is obtained below T_c. By analysis of the magnetisation data it has been demonstrated that the transport critical current density increases as the sintering temperature is raised. This may be caused by an increase in the volume fraction of superconducting material, although verification of this at low fields and at temperatures well below T_c is not possible utilising magnetisation techniques.

The thermal conductivity data shows that samples HM2-950 and HM2-970 are very similar in their bulk properties and is consistent with the idea that they each sample contains a relatively small amount of superconducting material. HM2-990 has a much higher thermal conductivity than the other two samples, exhibits a much more pronounced peak when cooled below T_c, and provides evidence for a secondary peak at 45K. The thermal conductivity of HM2-990 provides evidence to suggest that the sample has a much higher volume fraction of 90K superconducting material with a higher strength inter-granular coupling. Thermal conductivity

measurements have been used to provide information which complements the resistivity and magnetisation data and has been able to assist in understanding the simple physical properties of the material by providing a qualitative description of the properties of the material at a temperature well below T_c.

Chapter VIII

Conclusions and Suggestions for Further Work

The world is round and the place which may seem like the end may also be only the beginnning
— Ivy Baker Priest.
'Parade'

8.1 Summary

Apparatus to measure the thermal conductivity of $YBa_2Cu_3O_{7-\delta}$ at temperatures between 20K and 120K has been designed and constructed. Measurements are performed on 20mm long bars, held under vacuum, using a longitudinal heat flow technique. A differential thermocouple is used to directly measure the temperature difference across the sample. Equilibrium times in this system are of the order of 15 minutes. The system is computer controlled thereby enabling data acquisition to occur throughout the entire cryostat 'run'.

A series of samples of $YBa_2Cu_3O_{7-\delta}$ have been prepared which have undergone sintering at slightly different temperatures. Preliminary characterisation techniques used have included electrical resistivity in applied magnetic fields, magnetisation measurements and X-ray powder diffraction. Similar measurements have been performed upon samples prepared by Cookson Central Research.

Using these complementary techniques a qualitative analysis of the relationship between the electrical resistivity, transport critical current density and thermal conductivity has been performed. As the sintering temperature increases, there

is an increase in the strength of the inter-granular coupling. This effect is most pronounced at sintering temperatures above the initial calcination temperature of the precursor materials. The increase in the connectivity manifests itself most clearly in the thermal conductivity data. A complete quantitative analysis of the data would require a lengthy and exhaustive study owing to the complexity of the relationship between the oxygen content and the connectivity of the samples.

Thermal conductivity measurements have been performed upon a sample of $YBa_2Cu_3O_{7-\delta}$ which has been subjected to a series of heat treatments in order to remove oxygen from the material. The measurements show conclusively that the thermal conductivity of $YBa_2Cu_3O_{7-\delta}$ is very strongly influenced by the oxygen content of the material. A reduction of the oxygen content of the material results in a substantial lowering of the thermal conductivity.

A quantitative model, based upon the Boltzmann equation, has been constructed. This quantitative model demonstrates that consideration of the changes in phonon interactions alone cannot account for the differences in the behaviour of the thermal conductivity of $YBa_2Cu_3O_6$ and $YBa_2Cu_3O_7$. In addition, the model shows that there must be a significant carrier contribution to the thermal conductivity in both the normal and superconducting states. A physical process has been proposed which provides the required large carrier contribution below T_c. The parameterisation of this process, whilst not arbitrary, is only loosely based. Nevertheless, this model can provide a good fit to the observed thermal conductivity in both the superconducting and non-superconducting material. Improvements in the calculations and a greater understanding of the effect of free carriers below T_c could be achieved if this parameterisation was placed upon a more formal basis.

8.2 Future Work

8.2.1 Models

It is hoped that the model of the thermal conductivity of $YBa_2Cu_3O_{7-\delta}$ developed in this thesis provides some encouragement for the development of further theories which may be able to explain more fully the role of the 'normal' electrons in the transport properties of the superconducting state. Many qualitative

interpretations have been proposed to explain the behaviour of the thermal conductivity of $YBa_2Cu_3O_{7-\delta}$. In contrast to this, the number of quantitative theories is extremely small. It seems difficult to build up a coherent picture of the major heat transport mechanisms without providing some 'hard' numerical evidence for each of the qualitative ideas proposed.

8.2.2 Experiments

The studies conducted in this thesis concentrated upon polycrystalline materials. It was stated in §2.5 that $YBa_2Cu_3O_{7-\delta}$ is highly anisotropic. Very few thermal conductivity measurements have been performed upon single crystal samples, owing largely to problems of obtaining suitable crystals. However large single crystals are becoming much more readily available. Determination of the anisotropy of the thermal conductivity, in particular with regard to the c axis direction, may be able to provide significant insight into the behaviour of the carriers in directions perpendicular and parallel to the oxygen chains.

The application of a magnetic field may strongly influence the behaviour of any 'normal state' carriers which may exist below T_c. Such experiments may be able to provide a reliable method of influencing the normal state heat carriers in a controllable manner, thereby yielding some useful insight into the behaviour of the materials

8.2.3 Improvements to the Experimental Apparatus

The experimental arrangement has a few minor failings with respect to the thermometry of the system. The sample clamping arrangement has proved to be adequate for the measurements obtained to date although more delicate samples may be affected by the pressure which is applied to achieve a good thermal contact. Large samples could be mounted by passing a copper bar through the sample, (in a similar manner to the thermocouple mounting). Rather than directly clamping the samples to the cold tip of the cryostat, the samples could be mounted first in a high thermal conductivity resin (such as Stycast) and firmly clamped to the cold tip. This would minimise the strain induced in the sample.

The siting of the cold plate heater is not ideal. Rewiring the heater such that the heat flow is more evenly distributed over the surface of the cold plate will reduce any errors which may arise from the existence of temperature gradients across the cold plate. A second heater mounted on the central column of the sample clamp might assist the temperature regulation of the system, particularly at very low temperatures.

Radiation losses have been assumed to be small owing to the sample being sited in a can which is thermally linked to the cold plate. The use of a further radiation shield mounted on the cold plate, coupled with appropriate heating and sensing devices, will allow the temperature gradient of the radiation shield to more closely match that of the sample.

The present system of acquisition of data from the analogue nanovoltmeter is adequate but rather cumbersome as the range of the nanovoltmeter cannot be altered remotely. This problem also afflicts the constant current power supply. A programmable constant current supply and a nanovoltmeter fitted with a computer interface would provide an elegant, albeit expensive solution to these irritations.

8.3 Conclusion

A number of theories exist which attempt to elucidate the form of the interactions which give rise to the existence of superconductivity at high temperatures. The development of such theories will require an understanding of the thermal transport mechanisms above and below T_c. Electrical and magnetic measurements provide extremely useful information about the behaviour of the superconducting condensate but they are of limited use in assessing the behaviour of the normal state carriers. Information obtained from electrical and magnetic measurements is complementary to the information obtained from thermal measurements which probe quantitatively the low energy quasi-particle excitations.

Bibliography

All that mankind has done, thought, gained, or been, it is all lying in magic preservation in the pages of books.
— Thomas Carlisle.

Abrikosov A.A. (1957) *Soviet Phys. JETP* **5** pp1174-1182

Allen P.J. (1991) *PhD Thesis, Durham University* To be submitted

Anderson P.W., Ren Y. (1990) *High Temperature Superconductivity: Proceedings of a Symposium at Los Alamos* ed: K.S. Bedell et al Publ: Addison Wesley ISBN 0-201-51249-1 pp3-34

Ando Y., Akita S. (1990) *Jap. Jnl. Appl. Phys.* **29** ppL770-L771

Ashcroft N.W., Mermin N.D. (1976) *Solid State Physics* Publ:Holt, Rhinehart and Wilson ISBN 0-03-049346-3

Ayache C., Barbara B., Bonjour E., Calemczuk R., Couach M., Henry J.H., Rossat-Mignod J. (1987) *Solid State Commun.* **64** pp247-252

Bardeen J., Cooper L.N., Schrieffer J.S. (1957) *Phys. Rev.* **108** pp1175-1204

Batlogg B., Cava R.J., Jayaraman A., van Dover R.B., Kourouklis G.A., Sunshine S., Murphy D.W., Rupp L.W., Chen H.S., White A., Short K.T., Mujsce A.M., Rietman E.A. (1987) *Phys. Rev. Lett.* **58** pp2333-2336

Bayot V., Delannay F., Dewitt C., Erauw J-P., Gonze X., Issi J-P., Jonas A., Kinanay-Alaoui M., Lambricht M., Michenaud J-P., Minet J-P., Piraux L. (1987) *Solid State Commun.* **63** pp983-986

Bednorz J.G., Müller K.A. (1986) *Z. Phys. B* **64** pp189-193

Bednorz J.G., Müller K.A. (1988) *Rev. Mod. Phys.* **60** pp585-600

Bell A.M.T. (1990) *Supercond. Sci. Technol.* **3** pp55-61

Beno M.A., Soderholm L., Capone D.W., Hinks D.G., Jorgensen J.D., Grace J.D., Schuller I.K., Segre C.U., Zhang K. (1987) *Appl. Phys. Lett* **51** pp5759

Berman R., Foster E.L., Ziman J.M. (1955) *Proc. Roy. Soc. (London)* **A231** pp130-144

Berman R. (1961) *Experimental Cryophysics* ed:Hoare, Jackson and Kurti Publ: Butterworths pp327-336

Berman R., Brock J.C.F., Huntley D.J. (1964) *Cryogenics* **4** pp233-239

Berman R., Brock J.C.F., Huntley D.J. (1965) *Advances in Cryogenic Engineering* **10** pp233-238

Berman R. (1976) *Thermal Conduction In Solids* Publ: Clarendon Press, Oxford ISBN 0-19-851430

Beyers R., Ahn B.T., Gorman G., Lee V.Y., Parkin S.S.P., Ramirez M.L., Roche K.P., Vazquez J.E., Gür T.M., Huggins R.A. (1989) *Nature* **340** pp619-621

Bourne L.C., Crommie M.F., Zettl A., zur Loye H.-C., Keller S.W., Leary K.L., Stacy A.M., Chang K.J., Cohen M.L., Morris D.E. (1987) *Phys. Rev. Lett.* **58** pp2337-2339

Burgess S., Greig D. (1974) *J. Phys. D.: Appl. Phys.* **7** pp2051-2057

Carruthers P. (1961) *Rev. Mod. Phys.* **33** pp92-138

Cava R.J., Batlogg B., Chen C.H., Rietman E.A., Zahurak S.M., Werder D. (1987) *Nature* **329** pp423-427

Chaplot S.L. (1988) *Phys. Rev. B* **37** pp7435-7442

Chaplot S.L. (1990) *Phys. Rev. B* **42** pp2149-2154

Collings E.W. (1986) *Applied Superconductivity, Metallurgy and Physics of Titanium Alloys* ISBN 0-306-41690-5 and 0-306-41691-3

Das Gupta A., Koch C.C., Kroeger D.M., Chou Y.T. (1978) *Phil. Mag. B* **38** pp367-380

File J., Mills R.G. (1963) *Phys. Rev. Lett.* **10** pp93-97

Fischer H.E., Watson S.K., Cahill D.G. (1988) *Comments on Condensed Matter Physics* **14** pp65-127

Forgan E.M., Lee S.L., Sutton S., Wellhofer F., Abell J.S., Gough C.E., Cox S.F.J., Scott C.A. (1990) *Supercond. Sci. Technol.* **3** pp217-221

Fruchter L., Giovannella C., Collin G., Campbell I.A. (1988) *Physica C* **156** pp69-72

Gallagher W.J., Worthington T.K., Dinger T.R., Holtzberg F., Kaiser D.L., Sandstrom R.L. (1987) *Physica B* **148** pp228-232

Ginsberg D.M. (1989) *Physical Properties of High Temperature Superconductors* Publ:World Scientific ISBN 9971-50-894-X

Ginzburg V.L., Landau L.D. (1957) *Zh. Eksp. i Teor. Fiz.* **20** 1064
Translation given in: ter Harr D. (1965) *Men of Physics I : L.D.Landau* Publ: Pergamon Press

Ginzburg V.L. (1955) *Il Nuovo Cimento* **10** pp1234-1250

Glover R.E. (1967) *Phys. Lett.* **25A** pp542-544

Gordon J.E., Tan M.L., Fisher R.A., Phillips N.E. (1989)
Solid State Commun. **69** pp625-629

Gorter C.J. (1964) *Rev. Mod. Phys.* **36** pp3-7

Greedan J.E., O'Reilly A.H., Stager C.V. (1987) *Phys. Rev. B.* **35** pp8770-8773

Hake R.R. (1967) *App. Phys. Lett.* **10** pp189-92

Harshman D.R., Aeppli G., Ansaldo E.J., Batlogg B., Brewer J.H., Carolan J.F., Cava R.J., Celio M., Chaklader A.C.D., Hardy W.N., Kreitzman S.R., Luke G.M., Noakes D.R., Senba M. (1987) *Phys. Rev. B* **36** pp2386-2389

Heremans J., Morelli D.T., Smith G.W., Strite S.C. (1988)
Phys. Rev. B **37** pp1604-1610

Hoon S.R., Willcock S.N.M. (1988) *J. Phys. E – Sci. Instrum.* **21** pp772-785

Inderhees S.E., Salamon M.B., Goldenfield N., Rice J.P., Pazol B.G., Ginsberg D.M., Liu J.Z., Crabtree G.W. (1988)
Phys. Rev. Lett **60** pp1178-1180
and erratum *Phys. Rev. Lett.* **60** p2445

Iye Y., Tamegai T., Takeya H., Takei H. (1987) *Physica B* **148** pp224-227

Jeżowski A., Mucha J., Rogacki K., Horyń R., Bukowski Z., Horobiowski M., Rafałowicz J., Stępien-Damm J., Sułkowski C., Trojnar E., Zaleski A., Klamut J. (1987) *Phys. Lett A* **122** pp431-433

Jeżowski A., Klamut J., Horyń R., Rogacki K. (1989)
Supercond. Sci. Technol. **1** pp296-301

Jorgensen J.D., Beno M.A., Hinks D.G., Soderholm L., Volin K.J., Hittermann R.L., Grace J.D., Schuller I.K., Segre C.U., Zhang K., Kleefisch M.S. (1987) *Phys. Rev. B* **36** pp3608-3616

Jorgensen J.D., Veal B.W., Paulikas A.P., Nowicki L.J., Crabtree G.W., Claus H., Kwok W.K. (1990) *Phys. Rev. B* **41** pp1863-1877

Kaiser D.L., Holtzberg F., Scott B.A., McGuire T.R. (1987)
Appl. Phys. Lett. **51** pp1040-1042

Kittel C. (1976) *Introduction to Solid State Physics*
Publ: Wiley & Sons ISBN 0-471-49024-5

Khachaturyan A.G., Semenovskaya S.V., Morris J.W. (1988)
Phys. Rev. B **37** pp2243-2246

Klemens P.G. (1955) *Proc. Roy. Soc (London)* **A68** pp1113-1128

Klemens P.G. (1969) *Thermal Conductivity* **1** pp1-68 Ed:R.P.Tye

Konczykowski M., Rullier-Albenque F., Collin G. (1988)
Physica C **153-155** pp1365-1366

Kopp J., Slack G.A. (1971) *Cryogenics* **11** pp22-25

Krusin-Elbaum L., Greene R.L., Holtzberg F., Malozemoff A.P., Yeshurun Y. (1989a) *Phys. Rev. Lett.* **62** pp217-220

Krusin-Elbaum L., Malozemoff A.P., Yeshurun Y., Cronemeyer D.C., Holtzberg F. (1989b) *Phys. Rev. B* **39** pp2936-2939

Lægreid T., Tuset P., Nes O.-M., Slaski M., Fossheim K. (1989)
Proc. M^2S Conference, Stanford. To be published in Physica C

Lægreid T., Fossheim K., Sandvold E., Julsrad S. (1987) *Nature* **330** pp637-638

Landolt-Börnstein (1984) New Series III/1 Publ: Springer,Berlin

Leary K.J., zur Loye H.-C., Keller S.W., Faltens T.A., Ham W.K., Michaels J.N., Stacy A.M. (1987) *Phys. Rev. Lett.* **59** pp1236-1239

Le Page Y., Siegrist T., Sunshine S.A., Schneemeyer L.F., Murphy D.W., Zahurak S.M., Waszczak J.V., McKinnon W.R., Tarascon J.M., Hull G.W., Greene L.H. (1987) *Phys. Rev. B* **36** pp3617-3621

London F., London H. (1935) *Proc. Roy. Soc. (London)* **A149** pp72-88

Maeda H., Tanaka Y., Fukutomi M., Asano T. (1988)
Jap. Jnl. App. Phys. **27** ppL209-L210

Maple M.B., Dalichaouch Y., Ferreira J.M., Hake R.R., Lee B.W., Neumeier J.J., Torikachvili M.S., Yang K.N., Zhou H., Guertin R.P., Kuric M.V. (1987)
Physica B **148** pp155-162

Matthias B.T., Geballe T.H., Compton V.B. (1963) *Rev. Mod. Phys.* **35** pp1-22

Matthiessen A. (1863)
Reports of the British Association for the Advancement of Science
32 pp125-161

McGuire T.R., Dinger T.R., Freitas P.J.P., Gallagher W.J., Plaskett T.S., Sandstrom R.L., Shaw T.M. (1987) *Phys. Rev. B* **36** pp4032-4035

Meissner W., Ochsenfeld R. (1933) *Naturwissenschaften* **21** pp787-788

Mendelssohn K. (1964) *Rev. Mod. Phys.* **36** pp7-12

Meservy R., Schwartz B.B. (1969) *Superconductivity (Vol.1)*
ed: R.D.Parks, Publ: M.Dekker N.Y. pp117-191

Michel C., Hervieu M., Borel M.M., Grandin A., Deslandes F., Provost J., Raveau B. (1987) *Z. Phys. B* **68** pp421-423

Monod P., Dubois B., Odier P. (1988) *Physica C* **153-155** pp1489-1490

Morelli D.T., Heremans J., Swets D.E. (1987) *Phys. Rev. B* **36** pp3917-3919

Morris D.E., Kuroda R.M., Markelz A.G., Nickel J.H., Wei J.Y.T. (1988)
Phys. Rev. B **37** pp5936-5939

Mott N.F. (1990) *Proc. LT19 Conference on High T_c Superconductors.*
To be published in *Supercond. Sci. Technol.* **4** ppS59-S66

Müller P., Andres K., Groß F., Veith H., Hackl R. (1988)
Physica C **153-155** pp421-422

Muto Y., Kobayashi N., Watanabe K. (1990) *Physica B* **164** pp139-149

Nakao K., Miura N., Tatsuhara K., Takeya H., Takei H. (1988)
Phys. Rev. Lett. **63** pp97-100

Oh B., Char K., Kent A.D., Naito M., Beasley M.R., Geballe T.H., Hammond R.H., Kapitulnik A., Graybeal J.M. (1988)
Phys. Rev. B **37** pp7861-7864

Olsen J.L. (1952) *Proc. Phys. Soc. (London)* **A65** pp518-532

Onnes H.K. (1911) *Akad. van Wetenschapen* **14** 113

Parkin S.S.P., Lee V.Y., Engler E.M., Nazzal A.I., Huang T.C., Gorman G., Savoy R., Beyers R. (1988) *Phys. Rev. Lett.* **60** pp2539-2542

Pellan P., Dousselin G., Cortés H., Rosenblatt J. (1972)
Solid State Commun. **11** pp427-431

Penney T., von Molnár S., Kaiser D., Holtzberg F., Kleinsasser A.W. (1988)
Phys. Rev. B **38** pp2918-2921

Philips J.C. (1989) *Physics of High-T_c Superconductors*
Publ:Academic Press ISBN 0-12-553990-8

Pippard A.B. (1953) *Proc. Roy. Soc. (London)* **A216** pp547-568

Raboutou A., Rosenblatt J., Peyral P. (1980) *Phys. Rev. Lett.* **45** pp1035-1039

Rayleigh, Lord (1911) *Proc. Roy. Soc. (London)* **A84** pp25-46

Reese W. (1966) *J. Appl. Phys.* **37** pp864-868

Reichardt W., Ewert D., Gering E., Gompf F., Pintschovius L., Renker B., Collin G., Dianoux A.J., Mukta H. (1989) *Physica B* **156 & 157** pp897-901

Renker B., Gompff F., Gering E., Roth G., Reichardt W., Ewert D., Reitschel H., Mutka H. (1988) *Z. Phys. B.* **71** pp437-442

Rhyne J.J., Neumann D.A., Gotaas J.A., Beech F., Toth L., Lawrence S., Wolf S., Osofsky M., Gubser D.U. (1987) *Phys. Rev. B* **36** pp2294-2297

Richardson R.C., Smith E.T. (1988) *Experimental Techniques in Low Temperature Physics* Publ: Addison Wesley ISBN 0-201-15002-6

Salama K., Selvamanickam V., Gao L., Sun K. (1989)
Appl. Phys. Lett. **54** pp2352-2354

Sample H.H., Brandt B.L., Rubin L.G., (1982) *Rev. Sci. Instrum.* **53** pp1129-1136

Savitskii E.M., Baron V.V., Efimov Yu.V., Bychkova M.I., Myzenkova L.F. (1981)
Superconducting Materials Publ: Plenum Press

Schoenberg D. (1952) *Superconductivity* Chapter 5 Publ: Cambridge Univ. Press

Schuller I.K., Hinks D.G., Beno M.A., Capone D.W., Soderholm L., Locquet J.-P., Bruynseraede Y., Segre C.U. Zhang K. (1987)
Solid State Commun. **63** pp385-388

Schuller I.K., Jorgensen J.D. (1989) *MRS Bulletin* **Jan 1989** pp27-30

Shante V.K.S., Kirkpatrick S. (1971) *Advances in Physics* **20** pp325-357

Sheng Z.Z., Hermann A.M. (1988) *Nature* **332** pp55-58

Silsbee F.B. (1916) *J. Wash. Acad. Sci.* **6** pp597-602

Sridar S., Wu D.H., Kennedy W. (1989) *Phys. Rev. Lett.* **63** pp1873-1876

Stokka S., Fossheim K. (1982) *J. Phys. E - Sci. Instrum.* **15** pp123-127

Strobel P., Monceau P., Tholence J.L., Currat R., Dianoux A.J., Capponi J.J., Bednorz J.G. (1988) *Physica C* **153-155** pp282-283

Takabatake T., Ishikawa M., Nakazawa Y., Oguro I., Sakakibara T., Goto T. (1987) *Jap. Jnl. App. Phys.* **26** ppL978-L979

Takeya H., Takei H. (1988) *Physica C* **153-155** pp413-414

Takita K., Akinaga H., Katoh H., Uchino T., Ishigaki T., Asano H. (1987)
Jap. Jnl. App. Phys. **26** ppL1323-L1325

Taylor K.N.R., Cook P., Puzzer T., Matthews D.N., Russell G.J., Goodman P. (1988) *Physica C* **153-155** pp411-412

Tinkham M. (1975) *Introduction to Superconductivity* Publ:R.E.Krieger, Malabar, Florida

Tinkham M., Lobb C.J. (1989) *Solid State Physics* **42** pp91-133 Publ: Academic Press ISBN 0-12-607742-8

Touloukhian Y.S., Powell R.W., Ho C.Y., Klemens P.G. (1970) *Thermophysical Properties of Matter* **1** pp3a-39a

Tozer S.W., Kleinsasser A.W., Penney T., Kaiser D., Holtzberg F. (1987) *Phys. Rev. Lett.* **59** 1768

Träuble H., Essmann U. (1968) *Phys. Stat. Sol.* **25** pp373-393

Tu K.N., Yeh N.C., Park S.I., Tsuei C.C. (1989) *Phys. Rev. B* **39** pp304-314

Uemura Y.J., Emery V.J., Moodenbaugh A.R., Suenaga M., Johnston D.C., Jacobson A.J., Lewandowski J.T., Brewer J.H., Kiefl R.F., Kreitzman S.R., Luke G.M., Riseman T., Stronach C.E., Kossler W.J., Kempton J.R., Yu X.H., Opie D., Schone H.E. (1988) *Phys. Rev. B* **38** pp909-912

Uher C., Kaiser A.B. (1987) *Phys. Rev. B* **36** pp5680-5683

Uher C. (1989) *Proc. of 3rd Annual Conference on Superconductivity and Applications* Buffalo, New York

Umezawa A., Crabtree G.W., Liu J.Z (1988a). *Physica C* **153-155** pp1461-1462

Umezawa A., Crabtree G.W., Liu J.Z., Moran T.J., Malik S.K., Nunez L.H., Kwok W.L., Sowers C.H. (1988b) *Phys. Rev. B* **38** pp2843-2846

Van Tendeloo G., Zandbergen H.W., Amelinckx S. (1987) *Solid State Commun.* **63** pp389-393

Varga R.S. (1962) *Matrix Iterative Analysis* Publ:Englewood Cliffs

Vidal F., Veira J.A., Maza J., Miguélez F., Morán E., Alario M.A. (1988) *Solid State Commun.* **66** pp421-425

Vinnikov L.Y., Emelchenko G.A., Kononovich P.A., Ossipyan Y.A., Schegolev I.P., Buravov L.I., Laukhin V.N. (1988) *Physica C* **153-155** pp1359-1360

Wanklyn B.M., Watts B.E., Haycock P., Pratt F., de Groot P.A.J., Rapson G.P. (1988) *Solid State Commun.* **66** pp441-443

Welp U., Grimsditch M., You H., Kwok W.K., Fang M.M., Crabtree G.W., Liu J.Z. (1989) *Physica C* **161** pp1-5

Werthamer N.R., Helfand E., Hohenberg P.C. (1966) *Phys. Rev.* **147** pp295-302

White G.K. (1979) *Experimental Techniques in Low Temperature Physics* Publ:Clarendon Press, Oxford ISBN 0-19-85-1359-3

Wiedemann G., Franz R. (1853) *Annalen der Physik* **89** 497

Wu M.K., Ashburn J.R., Torng C.J., Hor P.H., Meng R.L., Gao L., Huang Z.J., Wang Y.Q., Chu C.W. (1987) *Phys. Rev. Lett.* **58** pp908-910

Yeshurun Y., Malozemoff A.P., Holtzberg F., Dinger T.R. (1988) *Phys. Rev. B* **38** pp11828-11831

Young D.M. (1971) *Iterative Solution of Large Linear Systems* Publ:Academic Press

Yvon K., François M. (1989) *Z. Phys. B* **76** pp413-444

Zavaritskiĭ N.V., Samolĭov A.V., Yurgens A.A. (1988a) *JETP Lett.* **48** pp242-245

Zavaritskiĭ N.V., Samolĭov A.V., Yurgens A.A., Klochko V.S., Makarov V.I. (1988b) *Physica C* **162-164** pp562-563

Appendix A

The Boltzmann Equation

A.1 Basic Formulation

A phonon mode may be described in terms of a wave-vector, $\mathbf{q}$, and its polarisation, i. Every phonon mode $\mathbf{q}_i$ which is excited has an energy $E(\mathbf{q}_i)$ and group velocity $\mathbf{v}_g(\mathbf{q}_i)$ associated with it. The total "heat current" due to a particular phonon mode is given by the product of the energy of the mode and the velocity of propagation. The overall "heat current" due to all modes is then

$$\mathbf{h} = \sum_{\mathbf{q}_i} N(\mathbf{q}_i)\mathbf{v}_g(\mathbf{q}_i)E(\mathbf{q}_i) \qquad (A.1)$$

where $N(\mathbf{q}_i)$ is the number of modes with given $\mathbf{q}_i$ and is a distribution function.

In a state of thermal equilibrium, the distribution function will be given by some $N^0(\mathbf{q}_i)$ which will depend only upon the phonon frequency. In this case, the total heat current must be zero as for each mode $\mathbf{q}_i$ there is a further mode $-\mathbf{q}_i$ which has

$E(\mathbf{q}_i) = E(-\mathbf{q}_i)$

$\mathbf{v}_g(\mathbf{q}_i) = -\mathbf{v}_g(-\mathbf{q}_i)$

The following assumptions are now made:

1. Net phonon transport is in the direction of the temperature gradient.

2. At a time t the number of phonons in mode $\mathbf{q}_i$ is given by $N(\mathbf{q}_i)$

After a time δt, the phonons which were originally in state $\mathbf{q}_i$ will have moved a distance $\mathbf{v}_g(\mathbf{q}_i)\delta t$ and have been replaced by the phonon density appropriate to $\mathbf{v}_g(\mathbf{q}_i)\delta t$. Hence the rate of change of the distribution function, $N(\mathbf{q}_i)$, due to this drift is given by

$$\left.\frac{\partial N(\mathbf{q}_i)}{\partial t}\right|_{drift} = \frac{\partial N(\mathbf{q}_i)}{\partial T}\nabla T.\mathbf{v}_g(\mathbf{q}_i) \qquad (A.2)$$

In steady state this phonon drift must be balanced by other processes which are referred to here as scattering processes.

$$\left.\frac{\partial N(\mathbf{q}_i)}{\partial t}\right|_{drift} + \left.\frac{\partial N(\mathbf{q}_i)}{\partial t}\right|_{scatt} = 0 \qquad (A.3)$$

$$\left.\frac{\partial N(\mathbf{q}_i)}{\partial t}\right|_{scatt} - \frac{\partial N(\mathbf{q}_i)}{\partial T}\nabla T.\mathbf{v}_g(\mathbf{q}_i) = 0 \qquad (A.4)$$

This is a form of the Boltzmann equation.

A.2 Relaxation Time Approximation

The Boltzmann equation may be solved, approximately, by a variety of methods. One of the most common methods is to invoke the relaxation time approximation which makes the following assumptions:

1. The scattering processes restore equilibrium at a rate proportional to the departure from the equilibrium value.
2. The system is close to the equilibrium distribution. ie $\frac{\partial N^0(\mathbf{q}_i)}{\partial t} \simeq \frac{\partial N(\mathbf{q}_i)}{\partial t}$
3. The return to equilibrium is deemed to be characterised by some relaxation time, denoted τ.

Mathematically, these approximations may be expressed as

$$\left.\frac{\partial N(\mathbf{q}_i)}{\partial t}\right|_{scatt} = \frac{N^0(\mathbf{q}_i) - N(\mathbf{q}_i)}{\tau(\mathbf{q}_i)} \qquad (A.5)$$

hence

$$\frac{N^0(\mathbf{q}_i) - N(\mathbf{q}_i)}{\tau} = -\frac{\partial N(\mathbf{q}_i)}{dT}\nabla T.\mathbf{v}_g(\mathbf{q}_i) \qquad (A.6)$$

The heat capacity of a phonon mode $C(\mathbf{q}_i)$ is given by $\partial U(\mathbf{q}_i)/\partial T \equiv E(\mathbf{q}_i).\partial N(\mathbf{q}_i)/\partial T$.

$$N^0(\mathbf{q}_i) - N(\mathbf{q}_i) = -(\mathbf{v}_g(\mathbf{q}_i).\nabla T)\tau(q)\frac{C(\mathbf{q}_i)}{E(\mathbf{q}_i)} \qquad (A.7)$$

From equation A.1

$$\mathbf{h} = \sum_{\mathbf{q}_i} -(\mathbf{v}_g(\mathbf{q}_i).\nabla T)\tau(\mathbf{q}_i)\frac{C(\mathbf{q}_i)}{E(\mathbf{q}_i)}E(\mathbf{q}_i)\mathbf{v}_g(\mathbf{q}_i) \tag{A.8}$$

$$h_z = \sum_{\mathbf{q}_i} -(\mathbf{v}_g(\mathbf{q}_i).\nabla T)\tau(\mathbf{q}_i)C(\mathbf{q}_i)|\mathbf{v}_g(\mathbf{q}_i)|\cos\theta \tag{A.9}$$

Taking the option of the heat flow parallel to ∇T

$$\kappa = \sum_{\mathbf{q}_i} -\mathbf{v}_g(\mathbf{q}_i)^2\tau(\mathbf{q}_i).C(\mathbf{q}_i)\cos^2\theta \tag{A.10}$$

$$= \int_{\omega_i} -\mathbf{v}_g(\omega_i)^2\tau(\omega_i)C(\omega_i)\cos^2\theta d\omega_i \tag{A.11}$$

$$= \int_{\omega} -\mathbf{v}_g(\omega)^2\tau(\omega)\frac{(\hbar\omega)^2}{k_BT^2}\frac{e^{\hbar\omega/k_BT}}{(e^{\hbar\omega/k_BT}-1)^2}D(\omega)\cos^2\theta d\omega \tag{A.12}$$

Provided τ and $\mathbf{v}_g$ are independent of θ, averaging over all available angles results in the replacement of $\cos^2\theta$ by $1/3$.

Appendix B

Iterative Solution of Simultaneous Equations

B.1 Basic Concept

Consider a node at which the potential is v_{ij}. This potential should be set such that Kirchoff's law is obeyed, i.e. the net current flow into the node is equal to the sum of the externally applied currents. Thus where there are no externally applied currents, the net current flow is zero. With an iteration procedure one solves for the elements of the voltage vector by setting up equations of the form:

$$v_{ij}^{n+1} = f(v_{ij}^{n}) \qquad (B.1)$$

There are many ways of re-arranging simultaneous equations into this form, with each one having its own advantages and disadvantages. The more common types are given here, with appropriate modifications to deal with the particular problem which had to be solved.

Consider the matrix equation $\mathbf{G.V = C}$ where $\mathbf{G}$ is the matrix of conductances, $\mathbf{V}$ is the matrix of node potentials (unknown) and $\mathbf{C}$ is the matrix of applied current. This corresponds to a series of linear simultaneous equations:

$$\begin{aligned} g_{11}v_1 + g_{12}v_2 + g_{13}v_3 + g_{14}v_4 &= c_1 \\ g_{21}v_1 + g_{22}v_2 + g_{23}v_3 + g_{24}v_4 &= c_2 \\ g_{31}v_1 + g_{32}v_2 + g_{33}v_3 + g_{34}v_4 &= c_3 \\ g_{41}v_1 + g_{42}v_2 + g_{43}v_3 + g_{44}v_4 &= c_4 \end{aligned} \qquad (B.2)$$

This set of equations may be expressed in matrix form $\mathbf{G.V = I}$, where g_{ij} are the coefficients of the conductance matrix $\mathbf{G}$, v_i are the coefficients of the voltage matrix $\mathbf{V}$ (to be found), and c_i are the coefficients of the current matrix $\mathbf{C}$.

B.2 Gauss Jacobi Method

The matrix **G** may be broken into three matrices, a diagonal matrix (**D**), a lower triangular (**L**) and an upper triangular matrix (**U**).

$$\mathbf{G} = \mathbf{D} + \mathbf{L} + \mathbf{U} \tag{B.3}$$

$$\mathbf{D} = \begin{pmatrix} g_{11} & 0 & 0 & 0 \\ 0 & g_{22} & 0 & 0 \\ 0 & 0 & g_{33} & 0 \\ 0 & 0 & 0 & g_{44} \end{pmatrix} \mathbf{L} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ g_{21} & 0 & 0 & 0 \\ g_{31} & g_{32} & 0 & 0 \\ g_{41} & g_{42} & g_{43} & 0 \end{pmatrix} \mathbf{U} = \begin{pmatrix} 0 & g_{12} & g_{13} & g_{14} \\ 0 & 0 & g_{23} & g_{24} \\ 0 & 0 & 0 & g_{34} \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{B.4}$$

so now

$$\begin{aligned} (\mathbf{D} + \mathbf{L} + \mathbf{U})\mathbf{V} &= \mathbf{C} \\ \mathbf{V} &= \mathbf{D}^{-1}[\mathbf{C} - (\mathbf{L} + \mathbf{U})\mathbf{V}] \\ \mathbf{V} &= \mathbf{V} - \mathbf{D}^{-1}[(\mathbf{L} + \mathbf{U} + \mathbf{D})\mathbf{V} - \mathbf{C}] \end{aligned} \tag{B.5}$$

which suggests the iteration procedure:

$$\mathbf{V}^{n+1} = \mathbf{V}^n - \mathbf{D}^{-1}[(\mathbf{L} + \mathbf{U} + \mathbf{D})\mathbf{V}^n - \mathbf{C}\mathbf{V}^n] \tag{B.6}$$

$\mathbf{D}^{-1}$ is easily invertible as it is a diagonal matrix, so one has a simple iterative procedure for finding the voltage vector **V**. For the example set of equations, they are solved by the procedure:

$$\begin{aligned} v_1^{n+1} &= v_1^n - (g_{11}v_1^n + g_{12}v_2^n + g_{13}v_3^n + g_{14}v_4^n - c_1)/g_{11} \\ v_2^{n+1} &= v_2^n - (g_{21}v_1^n + g_{22}v_2^n + g_{23}v_3^n + g_{24}v_4^n - c_2)/g_{22} \\ v_3^{n+1} &= v_3^n - (g_{31}v_1^n + g_{32}v_2^n + g_{33}v_3^n + g_{34}v_4^n - c_3)/g_{33} \\ v_4^{n+1} &= v_4^n - (g_{41}v_1^n + g_{42}v_2^n + g_{43}v_3^n + g_{44}v_4^n - c_4)/g_{44} \end{aligned} \tag{B.7}$$

This is the Gauss-Jacobi method. Despite its simplicity this method is not often used as it is usually far too slow. A simple modification to this method, known as the Gauss-Siedel method leads to a great increase in speed.

B.3 Gauss Siedel Method

With this method one uses the 'new' values of v_i^n as they are computed. For example, in the equations above when v_2^{n+1} is to be computed the value of v_1^{n+1} would be inserted instead of v_1^n. This produces a speed increase of the order of 50%. For the example set of equations, one therefore solves with the procedure:

$$\begin{aligned}
v_1^{n+1} &= v_1^n - (g_{11}v_1^n + g_{12}v_2^n + g_{13}v_3^n + g_{14}v_4^n - c_1)/g_{11} \\
v_2^{n+1} &= v_2^n - (g_{21}v_1^n{+}1 + g_{22}v_2^n + g_{23}v_3^n + g_{24}v_4^n - c_2)/g_{22} \\
v_3^{n+1} &= v_3^n - (g_{31}v_1^n{+}1 + g_{32}v_2^n{+}1 + g_{33}v_3^n + g_{34}v_4^n - c_3)/g_{33} \\
v_4^{n+1} &= v_4^n - (g_{41}v_1^n{+}1 + g_{42}v_2^n{+}1 + g_{43}v_3^n{+}1 + g_{44}v_4^n - c_4)/g_{44}
\end{aligned} \qquad (B.8)$$

In matrix form this is represented:

$$\mathbf{V}^{n+1} = \mathbf{V}^n - \mathbf{D}^{-1}[\mathbf{L}\mathbf{V}^{n+1} + \mathbf{U}\mathbf{V}^n + \mathbf{D}\mathbf{V}^n - \mathbf{C}\mathbf{V}^n] \qquad (B.9)$$

Thus the 'correction' to the value of $\mathbf{V}^n$ is

$$\mathbf{D}^{-1}[(\mathbf{L}\mathbf{V}^{n+1} + \mathbf{U}\mathbf{V}^n + \mathbf{D}\mathbf{V}^n - \mathbf{C}\mathbf{V}^n] \qquad (B.10)$$

Note that although $\mathbf{V}^{n+1}$ appears on both sides of the equation, the fact that on the right hand side it is multiplied by $\mathbf{L}$, ensures that the equation is consistent.

B.4 Simultaneous Over Relaxation

The method of simultaneous over relaxation (SOR) follows the same principles as the Gauss-Siedel method. However in this case an attempt is made to accelerate towards the solution. Effectively an 'over-correction' in the iterative procedure is made by multiplying the correction term by an acceleration parameter, ω. The solution procedure is now

$$\mathbf{V} = \mathbf{V}^n - \omega\mathbf{D}^{-1}[\mathbf{L}\mathbf{V}^{n+1} + \mathbf{U}\mathbf{V}^n + \mathbf{D}\mathbf{V}^n - \mathbf{C}] \qquad (B.11)$$

Varga (1962) has shown that the SOR method is only convergent for $0 < \omega < 2$. In order to obtain the optimum value of ω for a particular system, the solution

must already be known. As the boundary conditions of the problem determine the value of ω , once an optimum ω has been found for a particular system, this value will be the same for all similar problems.

B.5 Chebyshev Acceleration

Chebyshev acceleration takes the SOR method one step further. This system makes use of the idea that although there is one particular value of ω which is optimal when close to the solution, this may not be the best value to choose when far away from the solution. To understand how this works one must look at the rate of convergence of the Gauss-Siedel method. In fact it is easier to consider the Gauss-Jacobi method and follow on from there.

In the Gauss-Jacobi method, the new vector $\mathbf{V}$ is found by multiplying the old vector by $\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})$ and then adding on a constant. This matrix has eigenvalues which indicate by what factor the modulus of the eigenvalue of the residual vector is suppressed. Clearly these eigenvalues must be less than 1 otherwise the residual will increase. The rate of convergence is therefore controlled by the slowest decaying eigenmode (the largest eigenvalue of $\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})$). This value is called the spectral radius ρ_s, in general it will increase as the matrix size increases. Varga (1962) has shown that the spectral radius for the Gauss-Siedel iterative process is simply the square of ρ_s for the Gauss-Jacobi method. He also demonstrated that the optimal choice of ω is given by

$$\omega = \frac{2}{1+\sqrt{(1-\rho^2)}} \qquad (B.12)$$

and that the spectral radius for SOR is

$$\rho_{SOR} = \left(\frac{1-\rho}{1+\sqrt{1-\rho^2}}\right)^2 \qquad (B.13)$$

However the value of ρ must still be known before the formula is applied, this involves knowing the solution! As ω is related to ρ, the value of the parameter ρ (not ω) was varied and an optimal one found, this was then used for all calculations utilizing that particular matrix size.

Chebyshev acceleration begins the iteration procedure with a particular value of ω (usually 1) and then update this value after each iteration. The particular

implementation used here is due to Young (1971). The matrix G is subdivided into both 'black' (B) and 'white' (W) points, rather like a chess board. The 'black' points are updated first, followed by the 'white' points. In the implementation used here, the matrix G is sub-divided as follows:

$$\begin{pmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \\ g_{41} & g_{42} & g_{43} & g_{44} \end{pmatrix} = \begin{pmatrix} W & B & W & B \\ B & W & B & W \\ W & B & W & B \\ B & W & B & W \end{pmatrix} \qquad (B.14)$$

With the matrix sub-divided, one now uses one acceleration parameter, ω_n, for the 'black' points, and a different one, ω'_n, for the 'white' points. With Chebyshev acceleration these parameters are allowed to vary, according to the formulae:

$$\begin{aligned} \omega_1 &= 1 \\ \omega_n &= (1 - \tfrac{1}{4}\omega'_{n-1}\rho^2)^{-1} \\ \omega'_1 &= (1 - \tfrac{1}{2}\rho^2)^{-1} \\ \omega'_n &= (1 - \tfrac{1}{4}\omega_n\rho^2)^{-1} \end{aligned} \qquad (B.14)$$

Thus as $n \rightarrow \infty$ the ω tend towards the value

$$\omega_\infty = \frac{2}{1 + \sqrt{1 - \rho^2}} \qquad (B.15)$$

In the final implementation of the computer program, the optimum value of ρ was found empirically. The optimum values for ρ used were

No of nodes	matrix order	no of iterations	ρ
10x10x10	1000 x 1000	28	0.984
20x20x20	8000 x 8000	38	0.9954
30x30x30	27000 x 27000	70	0.9982
40x40x40	64000 x 64000	90	0.9993

Appendix C

Program to Model Thermal Conductivity

The program is written in FORTRAN 77 and conforms to the ANSI standard. Functions and subprograms are utilised wherever they may assist with the clarity of structure. Global data is passed to each subroutine or function by use of COMMON blocks which are by necessity large. Meaningful variable names are used throughout the program wherever possible. The routines to perform the numerical integration utilise Simpson's rule and are adapted from the set of routines to be found in Press et al (1988).

NO PAGINATION

After P. 160 to End.

```
C
C ***************************************************************
C
      PROGRAM THERM
C
C ***************************************************************
C
C     List of parameter names and meanings
C
C     W = phonon frequency
C     WMIN = phonon frequency at start of integration
C     WMAX = phonon frequency at end of integration
C
C     TAUPP  = phonon phonon collision time in 10e-12 seconds
C     TAUPP1 = effective phonon phonon collision time at low T
C     TAUPP2 = effective phonon phonon collision time at high T
C     DTPP   = difference between TAUPP1 and TAUPP2
C
C     TAUEP  = electron phonon collision time in 10e-12 seconds
C     TAUEP1 = electron phonon collision time for unpaired electrons
C     DTEP   = change in electron phonon time below energy gap
C
C     RLV = pi / lattice parameter
C     V   = phonon group velocity
C
C ***************************************************************
C
      DIMENSION WLIM(30), WHT(30), WPOW(30)
      DIMENSION ALIM(30), AHT(30), APOW(30)
      DIMENSION BLIM(30), BHT(30), BPOW(30)
      DIMENSION WLAD(10), V(10)
      DIMENSION A(30), B(30)
      DIMENSION TPPLO(10), TPPHI(10), TTU(10), TEP1(10), DTTEP(10)
      DIMENSION ALPHA(10), BETA(10), RHO(10)
      CHARACTER*12 FILOUT, DOSFIL
      CHARACTER*1 ANS, ELTERM
      CHARACTER*12 TIMER
      CHARACTER*16 DATE
      COMMON /BLOCK/ WLIM, WHT, WPOW, NLIMS
      COMMON /PREF/ A, B
      COMMON /PARS/ T, TC, HBAR, RKB, RLV
      COMMON /TAUS/ TAUPP1, DTPP, TAUEP1, DTEP, TU
      COMMON /WS/ WMIN, WMAX, WSTEP
      COMMON /DEFECT/ HW, G, DM, ACUBED
      COMMON /LADDER/ WLAD, V, NLAD
C
      HBAR = 6.25E-34 / (2*3.141592654)
      RKB = 1.38E-23
      RLV = 3.141592654 / 3.85
      ACUBED = 3.85E-10 * 3.85E-10 * 11.67E-10
      DM = 0.312
      CALL SETDOS
      CALL LADD
C
C ***************************************************************
C  Calculate the prefactors for the DOS
C  Evaluate the integrals of DOS and normalise them
C  to 3 (no degrees of freedom)
C  Then write to disc if required
C ***************************************************************
C
      CALL PREFAC
      WRITE (6,*)'Calculated DOS prefactors '
      RVAL = 0.
      CALL DSIMP(WMIN, WMAX, RVAL)
      PRINT *, 'Done numerical normalisation.  RVAL = ', RVAL
      RNORM = 3. / RVAL
C
      WRITE (6,*)'Do you wish to save the density of states ?'
      READ (5,10) ANS
   10 FORMAT (1A)
      IF ((ANS .EQ. 'Y') .OR. (ANS .EQ. 'y')) THEN
       WRITE (6,*)'Enter the filename to save DOS to ? '
       READ (5,80) DOSFIL
   20  FORMAT (1A)
       OPEN (8,FILE=DOSFIL)
       WRITE (8,*)'Density of states for'
       IF (TC .GT. 0) THEN
        WRITE (8,*)'Oxygen 7 (superconducting)'
       ELSE
        WRITE (8,*)'Oxygen 6 (semiconducting)'
       END IF
       DO 30 LOOP = 1, 7
        WRITE (8,*)'---------------'
   30  CONTINUE
       WRITE (8,*)'Phonon energy (meV)'
       WRITE (8,*)'DOS (arbitrary units)'
       WSTEP = 0.5
       DO 40 RL = WMIN, WMAX, WSTEP
        WRITE (8,*) RL, DOS(RL) / RVAL
   40  CONTINUE
       WRITE (6,*)'Done DOS'
      END IF
C
      WRITE (6,*)'Do you wish to use the electron terms Y/N? '
      READ (5,50) ELTERM
   50 FORMAT (1A)
C
C ***************************************************************
C    Enter the required relaxation times, etc
C ***************************************************************
C
      WRITE (6,*)'Enter the data as requested.'
      WRITE (6,*)'You will be prompted for data so that you may'
      WRITE (6,*)'overlay up to 6 graphs.'
      WRITE (6,*)'If you do not wish to enter any more data'
      WRITE (6,*)'enter a negative relaxation time'
      WRITE (6,*)'Enter 0 to use the previous value.'
C
      NUMBER = 1
   60 CONTINUE
      WRITE (6,*)'Enter the phonon-phonon relaxation time',
     1      ' at low T ?'
      READ (5,*) TPPLO(NUMBER)
      IF (TPPLO(NUMBER) .EQ. 0) TPPLO(NUMBER) = TPPLO(NUMBER - 1)
      IF (TPPLO(NUMBER) .LE. 0) GO TO 70

      WRITE (6,*)'Enter the phonon-phonon relaxation time',
     1      ' at high T ?'
      READ (5,*) TPPHI(NUMBER)
```

```
      IF (TPPHI(NUMBER) .EQ. 0) TPPHI(NUMBER) = TPPHI(NUMBER - 1)
      IF (TPPHI(NUMBER) .LE. 0) GO TO 70
      WRITE (6,*)'Enter the temperature at which Umlapp processes',
     1       ' are effectively switched on ?'
      READ (5,*) TTU(NUMBER)
      IF (TTU(NUMBER) .EQ. 0) TTU(NUMBER) = TTU(NUMBER - 1)
      IF (TTU(NUMBER) .LE. 0) GO TO 70
      WRITE (6,*)' Enter the electron-phonon relaxation time for',
     1       ' unpaired electrons? '
      READ (5,*) TEP1(NUMBER)
      IF (TEP1(NUMBER) .EQ. 0) TEP1(NUMBER) = TEP1(NUMBER - 1)
      IF (TEP1(NUMBER) .LE. 0) GO TO 70
      WRITE (6,*)'Enter the change in e-ph relaxation time'
      WRITE (6,*)'for phonons of energy less than the energy gap ? '
      READ (5,*) DTTEP(NUMBER)
      IF (DTTEP(NUMBER) .EQ. 0) DTTEP(NUMBER) = DTTEP(NUMBER - 1)
      IF (DTTEP(NUMBER) .LE. 0) GO TO 70
      WRITE (6,*)' Enter coefficient for a/T^2  '
      READ (5,*) ALPHA(NUMBER)
      IF (ALPHA(NUMBER) .EQ. 0) ALPHA(NUMBER) = ALPHA(NUMBER - 1)
      IF (ALPHA(NUMBER) .LT. 0) GO TO 70
      WRITE (6,*)' Enter coefficient for b/(Tc-T) '
      READ (5,*) BETA(NUMBER)
      IF (BETA(NUMBER) .EQ. 0) BETA(NUMBER) = BETA(NUMBER - 1)
      IF (BETA(NUMBER) .LT. 0) GO TO 70
      WRITE (6,*)' Enter coefficient for (Tc-T)^n '
      READ (5,*) RHO(NUMBER)
      IF (RHO(NUMBER) .EQ. 0) RHO(NUMBER) = RHO(NUMBER - 1)
      IF (RHO(NUMBER) .LT. 0) GO TO 70
C
      NUMBER = NUMBER + 1
      IF (NUMBER .LE. 6) GO TO 60
   70 CONTINUE
      IF (NUMBER .EQ. 1) GO TO 60
      NUMBER = NUMBER - 1
C
C*********************************************************************
C Write header information to disc
C*********************************************************************
C
      WRITE (6,*)'Enter the filename to save data to ? '
      READ (5,80) FILOUT
   80 FORMAT (1A)
      OPEN (7,FILE=FILOUT)
C
      TMIN = 2
      TMAX = 150
      TSTEP = 2
C
      CALL TIME(21, 0, DATE)
      CALL TIME(4, 0, TIMER)
      WRITE (7,*) DATE, TIMER
      WRITE (7,*)' Tc = ', TC
      WRITE (7,*)' Taupp (Low T)  = ', TPPLO(1), ' to ', TPPLO(NUMBER)
      WRITE (7,*)' Taupp (High T) = ', TPPHI(1), ' to ', TPPHI(NUMBER)
      WRITE (7,*)' E-ph   Tau     = ', TEP1(1), ' to ', TEP1(NUMBER)
      WRITE (7,*)' d(E-ph Tau)    = ', DTTEP(1), ' to ', DTTEP(NUMBER)
      WRITE (7,*)' T (umklapp)    = ', TTU(1), ' to ', TTU(NUMBER)
      WRITE (7,*)' Alpha = ', ALPHA(1), ' to ', ALPHA(NUMBER)
      WRITE (7,*)' Beta  = ', BETA(1), ' to ', BETA(NUMBER)
      WRITE (7,*)' Temperature (K)'
      WRITE (7,*)' Thermal Conductivity)'
C
C*********************************************************************
C Finally get around to doing the calculation
C************** tau in 10e-12 s *************
C**************    T in meV     *************
C*********************************************************************
C
      WRITE (6,*)'Enter the defect coefficient G'
      WRITE (6,*)'G=2   for x=6.5'
      WRITE (6,*)'G=2.5 for x=6.4,6.6'
      WRITE (6,*)'G=3.3 for x=6.3,6.7 (approx)'
      WRITE (6,*)'G=10000 for x=6,7'
      READ (5,*) G
      TC = TC * 0.08614
      DO 100 LOOP = 1, NUMBER
C
       TAUPP1 = TPPLO(LOOP)
       TAUPP2 = TPPHI(LOOP)
       DTPP = TAUPP1 - TAUPP2
       TAUEP1 = TEP1(LOOP)
       DTEP = DTTEP(LOOP)
       TU = TTU(LOOP)
       TU = TU * 0.08614
       ALPH = ALPHA(LOOP)
       BET = BETA(LOOP)
       RH = RHO(LOOP)
C
       RNE = 3E27
       RME = 9.1E-31
       EPREF = (RNE*3.14*3.14*RKB*RKB*1E-12) / (3*RME)
       RKMAX = 0
       TCK = TC / 0.08614
C
      DO 90 TEM = TMIN, TMAX, TSTEP
       T = TEM * 0.08614
C
C  Compute electronic contribution
C    if requested
C
        IF (ELTERM .EQ. 'Y') THEN

         IF (T .GE. TC) THEN
          TAUEE = ALPH / (TEM*TEM)
          EPRF2 = EPREF * TEM
         ELSE
          TAUEE = ALPH / (TEM*TEM)
          TAUEE = TAUEE + BET * ((TCK - TEM)**RH)
          EPRF2 = EPREF * TEM * ((T/TC)**4)
         END IF
C
         TAUE = TAUEE * TAUEP1 / (TAUEE + TAUEP1)
         RKE = EPRF2 * TAUE
C
        ELSE
         RKE = 0
        END IF
C
        RKVAL = 0.
```

```
        CALL TSIMP(WMIN, WMAX, RKVAL)
        RKVAL = RKVAL * 1E-12 * RKB * 8E28 * RNORM / 3.
        RKVAL = RKVAL + RKE
        WRITE (7,*) TEM, RKVAL
        IF (RKVAL .GT. RKMAX) RKMAX = RKVAL
C        WRITE (6,*)'T = ', TEM, '  K = ', RKE
C        WRITE (6,*)'TAUEE = ', TAUEE, '  TAUEP = ', TAUEP1
C        WRITE (6,*)'EPREP = ', EPREF, '  EPRF2 = ', EPRF2
C        WRITE (6,*)'TAUE = ', TAUE
   90  CONTINUE
       WRITE (6,*)'Kmax = ', RKMAX
       WRITE (7,*) - 1., -1.
  100 CONTINUE
      CLOSE (7)
      STOP
      END
C
C
C*********************************************************************
      FUNCTION TAU(W)
C     Evaluates the relaxation time
C*********************************************************************
C
      COMMON /PARS/ T, TC, HBAR, RKB, RLV
      COMMON /TAUS/ TAUPP1, DTPP, TAUEP1, DTEP, TU
      COMMON /WS/ WMIN, WMAX, WSTEP
      COMMON /DEFECT/ HW, G, DM, ACUBED
C
      VV = GROUPV(W)
      HW = W * 1.602E-19 * 1E-3 / HBAR
      TAUPP = TAUPP1 + DTPP * (EXP(-1.*T/TU) - 1)
C
      IF (T .GT. TC) THEN
       TAUEP = TAUEP1
      ELSE
       GAP = TC * 3.52 * (1 - ((T/TC)**4))
       IF (W .LT. GAP) THEN
        TAUEP = TAUEP1 * DTEP
       ELSE
        TAUEP = TAUEP1
       END IF
      END IF
C
      TAU0 = TAUEP * TAUPP / (TAUEP + TAUPP)
C
      RNUMER = G * 4 * 3.1416 * VV * VV * VV
      DENOM = ACUBED * HW * HW * HW * HW * DM * DM
      T2 = 1E12 * RNUMER / DENOM
      TAU = TAU0 * T2 / (TAU0 + T2)
      HW = T2
C
      END
C
C
C*********************************************************************
      FUNCTION RINTEG(X)
C     Evaluates the integral function.
C     Has to be a function in order to call from integration routines.
C     Which are taken from NUMERICAL recipes.
C*********************************************************************
C
      DIMENSION WLIM(30), WHT(30), WPOW(30)
      COMMON /BLOCK/ WLIM, WHT, WPOW, NLIMS
      COMMON /PARS/ T, TC, HBAR, RKB, RLV
      COMMON /WS/ WMIN, WMAX, WSTEP
C
      VV = GROUPV(X)
      XT = X / T
      IF ((XT .GE. 30) .OR. (X .GT. WMAX)) THEN
       R = 0.
      ELSE
       BOL = EXP(XT)
C
C********** x is already * hbar/1.602e-19 **********
C********** RLV in inverse Angstroms      **********
C********** WU converted to meV           **********
C
       R = VV * VV * XT * XT * BOL / ((BOL - 1)*(BOL - 1))
       R = R * TAU(X) * DOS(X)
      END IF
      RINTEG = R
      END
C
C
C*********************************************************************
      FUNCTION GROUPV(X)
C     Calculates the group velocity from the ladder
C     for phonons of energy hx.
C*********************************************************************
C
      DIMENSION WLAD(10), V(10)
      COMMON /LADDER/ WLAD, V, NLAD
C
      LOOP = 1
      OK = 0
      VV = 0
   10 CONTINUE
      IF ((X .GT. WLAD(LOOP)) .AND. (X .LE. WLAD(LOOP + 1)))
     1      THEN
       VV = V(LOOP + 1)
       OK = 1
      ELSE
       LOOP = LOOP + 1
      END IF
      IF ((LOOP .LT. NLAD) .AND. (OK .NE. 1)) GO TO 10
C      PRINT *,'X = ',X,' WLAD = ',WLAD(LOOP),' VV = ',VV
      GROUPV = VV
C
      RETURN
      END
C
C
C*********************************************************************
      SUBROUTINE LADD
C      Enters phonon 'ladder' for group velocities
C*********************************************************************
C
      DIMENSION WLAD(10), V(10), RLAD(10)
      COMMON /PARS/ T, TC, HBAR, RKB, RLV
      COMMON /LADDER/ WLAD, V, NLAD
```

```
C
      DATA RLAD /0., 16, 32, 64, 80, 96, 112, 0, 0, 0/
      NLAD = 7
C
      DO 10 LOOP = 1, 10
       WLAD(LOOP) = RLAD(LOOP)
C       PRINT *,LOOP,WLAD(LOOP)
   10 CONTINUE
C
      DO 20 LOOP = 2, 10
       V(LOOP) = (WLAD(LOOP) - WLAD(LOOP - 1)) * 1.602E-19 / (1E3*HBAR)
       V(LOOP) = V(LOOP) / (RLV*1E10)
   20 CONTINUE
C
      END
C
C
C ********************************************************************
      FUNCTION DOS(W)
C     Evaluates the density of states for a given omega
C ********************************************************************
C
      DIMENSION WLIM(30), WHT(30), WPOW(30)
      DIMENSION A(30), B(30)
      COMMON /BLOCK/ WLIM, WHT, WPOW, NLIMS
      COMMON /PREF/ A, B
C
      RDOS = 0
      DO 10 LOOP = 1, (NLIMS - 1)
       IF ((W .GT. WLIM(LOOP)) .AND. (W .LT. WLIM(LOOP + 1)) .AND. (
     1       WPOW(LOOP) .NE. 0)) THEN
         IF (WHT(LOOP) .LT. WHT(LOOP + 1)) THEN
          RDOS = A(LOOP) * ((W - WLIM(LOOP))**WPOW(LOOP)) + B(LOOP)
         ELSE
          RDOS = A(LOOP) * ((WLIM(LOOP + 1) - W)**WPOW(LOOP)) + B(LOOP)
         END IF
        END IF
        IF ((W .EQ. WLIM(LOOP)) .AND. (WPOW(LOOP) .NE. 0)) THEN
         RDOS = B(LOOP)
        END IF
   10 CONTINUE
      DOS = RDOS
      END
C
C
C ********************************************************************
      SUBROUTINE PREFAC
C     Calculates prefactors for density of states
C ********************************************************************
C
      DIMENSION WLIM(30), WHT(30), WPOW(30)
      DIMENSION A(30), B(30)
      COMMON /BLOCK/ WLIM, WHT, WPOW, NLIMS
      COMMON /PREF/ A, B
      COMMON /PARS/ T, TC, HBAR, RKB, RLV
C
      DO 20 LOOP = 1, (NLIMS - 1)
       IF ((WLIM(LOOP) .NE. WLIM(LOOP + 1)) .AND. (WPOW(LOOP) .NE. 0.))
     1       THEN
        IF (WHT(LOOP) .LE. WHT(LOOP + 1)) THEN
         A(LOOP) = (WHT(LOOP + 1) - WHT(LOOP)) / ((WLIM(LOOP + 1) -
     1   WLIM(LOOP))**WPOW(LOOP))
         B(LOOP) = WHT(LOOP)
        ELSE
         A(LOOP) = (WHT(LOOP) - WHT(LOOP + 1)) / ((WLIM(LOOP + 1) -
     1   WLIM(LOOP))**WPOW(LOOP))
         B(LOOP) = WHT(LOOP + 1)
        END IF
   10   CONTINUE
       ELSE
        A(LOOP) = 0.
        B(LOOP) = 0.
       END IF
   20 CONTINUE
      RETURN
      END
C
C
C ********************************************************************
      SUBROUTINE SETDOS
C ********************************************************************
C     The density of states is modelled by specifying the
C     intervals over which it changes.
C     With WLIM = 0, 30, 50
C          WHT = 0,  1,  0
C         WPOW = 2,  2,  2
C     The DOS follows an w^2 dependence from W=0 to W=30meV
C     with the value at W=30 being 1.
C     From W=30 to W=50meV the DOS falls from 1 to 0 with the
C     dependence also being w^2.
C ********************************************************************
C
      DIMENSION WLIM(30), WHT(30), WPOW(30)
      DIMENSION RSCLIM(30), RSCHT(30), RSCPOW(30)
      DIMENSION RNLIM(30), RNHT(30), RNPOW(30)
      DIMENSION SSCLIM(30), SSCHT(30), SSCPOW(30)
      DIMENSION SNLIM(30), SNHT(30), SNPOW(30)
      DIMENSION CSCLIM(30), CSCHT(30), CSCPOW(30)
      DIMENSION CNLIM(30), CNHT(30), CNPOW(30)
      CHARACTER*1 ANS, DSET
      COMMON /BLOCK/ WLIM, WHT, WPOW, NLIMS
      COMMON /PARS/ T, TC, HBAR, RKB, RLV
      COMMON /WS/ WMIN, WMAX, WSTEP
C
C ********************************************************************
C  Data of Rhyne et al - neutron weighted phonon DOS
C ********************************************************************
C  Data for superconducting phonon DOS in meV
C ********************************************************************
C
      DATA RSCLIM /00., 06., 12., 14., 20., 23., 25., 27., 30., 33.,
     1     35., 39., 41., 44., 48., 58., 62., 69., 71., 77., 81., 90.,
     2     0., 0., 0., 0., 0., 0., 0., 0./
      DATA RSCHT /00., 04., 15., 15., 36., 30., 33., 15., 12., 17., 19.,
     1     08., 08., 20., 06., 20., 20., 34., 34., 17., 15., 01., 0.,
     2     0., 0., 0., 0., 0., 0., 0./
      DATA RSCPOW /1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
     1     1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 0., 0., 0.,
     2     0., 0./
      RSCL = 22
```

```
C
C ********** Data for normal phonon DOS in meV **********
C
      DATA RNLIM /00., 06., 10., 16., 18., 24., 30., 33., 36., 44., 50.,
     1     58., 62., 70., 80., 90., 100., 0., 0., 0., 0., 0., 0., 0.,
     2     0., 0., 0., 0., 0., 0./
      DATA RNHT /00., 04., 15., 21., 19., 40., 26., 38., 25., 06., 06.,
     1     17., 12., 28., 18., 1., 0., 0., 0., 0., 0., 0., 0., 0.,
     2     0., 0., 0., 0., 0./
      DATA RNPOW /1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
     1     1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 0., 0., 0., 0.,
     2     0., 0./
      RNL = 17
C
C
C ********************************************************************
C Data of Strobel et al
C ********************************************************************
C Data for superconducting phonon DOS in meV
C ********************************************************************
C
      DATA SSCLIM /00., 05., 11., 12., 14., 15., 18., 20., 22., 24.,
     1     25., 27., 30., 35., 40., 45., 50., 55., 60., 65., 70., 75.,
     2     80., 85., 90., 95., 100., 110., 0., 0./
      DATA SSCHT /00., 7., 30., 42., 39., 50., 88., 84., 74., 83., 71.,
     1     67., 70., 57., 70., 60., 58., 43., 40., 36., 30., 22., 16.,
     2     16., 13., 11., 10., 01., 0., 0./
      DATA SSCPOW /1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
     1     1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
     2     1., 1./
      SSCL = 28
C
C ********** Data for normal phonon DOS in meV **********
C
      DATA SNLIM /00., 05., 11., 12., 14., 15., 17., 18., 20., 22., 24.,
     1     25., 29., 30., 35., 37., 40., 45., 47., 50., 55., 60., 65.,
     2     70., 75., 80., 85., 90., 95., 100./
      DATA SNHT /00., 07., 30., 42., 39., 65., 60., 66., 60., 58., 69.,
     1     69., 62., 66., 52., 50., 55., 40., 35., 35., 33., 31., 32.,
     2     32., 27., 20., 15., 12., 11., 9./
      DATA SNPOW /1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
     1     1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
     2     1., 1./
      SNL = 30
C
C
C ********************************************************************
C Data of Chaplot et al  'bare' phonon DOS
C ********************************************************************
C Data for superconducting phonon DOS in meV
C ********************************************************************
C
C      DATA CSCLIM /0., 07., 10., 13., 16., 24., 26., 28., 32., 36.,
C     1     38., 41., 44., 45., 50., 54., 57., 60., 66., 67., 73., 75.,
C     2     87., 90., 0., 0., 0., 0., 0., 0./
C       DATA CSCHT /00., 21., 50., 40., 44., 37., 36., 27., 21., 21.,
C 15.,
C     1     18., 12., 13., 12., 18., 12., 12., 16., 12., 08., 09., 01.,
C     2     01., 0., 0., 0., 0., 0., 0./
      DATA CSCLIM /0., 5., 7., 11., 15., 18., 22., 25., 27., 30., 32.,
     1     36., 41., 46., 50., 53., 55., 58., 59., 61., 63., 67., 70.,
     2     72., 77., 80., 90., 91., 100., 0./
      DATA CSCHT /00., 05., 10., 41., 30., 30., 24., 28., 25., 26., 18.,
     1     18., 14., 26., 12., 12., 10., 15., 14., 17., 15., 19., 12.,
     2     13., 8., 11., 1., 0., 0., 0./
      DATA CSCPOW /1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
     1     1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
     2     1., 1./
C       CSCL = 23
      CSCL = 27
C
C ********** Data for normal phonon DOS in meV **********
C
C ************************************************
C
      DATA CNLIM /00., 05., 07., 10., 13., 15., 16., 18., 23., 25., 27.,
     1     30., 33., 37., 40., 53., 58., 61., 64., 66., 70., 76., 81.,
     2     85., 90., 93., 95., 100., 0., 0./
      DATA CNHT /00., 20., 40., 22., 40., 30., 21., 31., 16., 24., 22.,
     1     26., 19., 18., 10., 06., 13., 08., 14., 12., 14., 07., 04.,
     2     08., 04., 0., 0., 0., 0., 0./
      DATA CNPOW /1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
     1     1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
     2     1., 1./
      CNL = 25
C
      DATA RMIN, RMAX /0.01, 100./
      WMIN = RMIN
      WMAX = RMAX
C
   10 CONTINUE
      WRITE (6,*)'Select which DOS to use '
      WRITE (6,*)'R.... Rhyne et al (neutron weighted)'
      WRITE (6,*)'S.... Strobel et al'
      WRITE (6,*)'C.... Chaplot (bare, calculated)'
      WRITE (6,*)
      READ (5,30) DSET
      IF ((DSET .NE. 'R') .AND. (DSET .NE. 'S') .AND. (DSET .NE. 'C'))
     1   GO TO 10
C
   20 CONTINUE
      WRITE (6,*)'Calculate K for superconducting material Y/N ?'
      READ (5,30) ANS
   30 FORMAT (1A)
      IF ((ANS .NE. 'Y') .AND. (ANS .NE. 'y') .AND. (ANS .NE. 'N') .AND.
     1   (ANS .NE. 'n')) GO TO 20
C
      IF ((ANS .EQ. 'Y') .OR. (ANS .EQ. 'y')) THEN
       SC = 1.
       IF (DSET .EQ. 'R') THEN
        NLIMS = RSCL
       END IF
       IF (DSET .EQ. 'S') THEN
        NLIMS = SSCL
       END IF
       IF (DSET .EQ. 'C') THEN
        NLIMS = CSCL
       END IF
      ELSE
       SC = 0.
```

```
      IF (DSET .EQ. 'R') THEN
       NLIMS = RNL
      END IF
      IF (DSET .EQ. 'S') THEN
       NLIMS = SNL
      END IF
      IF (DSET .EQ. 'C') THEN
       NLIMS = CNL
      END IF
     END IF
C
     DO 40 LOOP = 1, NLIMS
      IF (SC .EQ. 1) THEN
       IF (DSET .EQ. 'R') THEN
        WLIM(LOOP) = RSCLIM(LOOP)
        WHT(LOOP) = RSCHT(LOOP)
        WPOW(LOOP) = RSCPOW(LOOP)
       END IF
       IF (DSET .EQ. 'S') THEN
        WLIM(LOOP) = SSCLIM(LOOP)
        WHT(LOOP) = SSCHT(LOOP)
        WPOW(LOOP) = SSCPOW(LOOP)
       END IF
       IF (DSET .EQ. 'C') THEN
        WLIM(LOOP) = CSCLIM(LOOP)
        WHT(LOOP) = CSCHT(LOOP)
        WPOW(LOOP) = CSCPOW(LOOP)
       END IF
      ELSE
       IF (DSET .EQ. 'R') THEN
        WLIM(LOOP) = RNLIM(LOOP)
        WHT(LOOP) = RNHT(LOOP)
        WPOW(LOOP) = RNPOW(LOOP)
       END IF
       IF (DSET .EQ. 'S') THEN
        WLIM(LOOP) = SNLIM(LOOP)
        WHT(LOOP) = SNHT(LOOP)
        WPOW(LOOP) = SNPOW(LOOP)
       END IF
       IF (DSET .EQ. 'C') THEN
        WLIM(LOOP) = CNLIM(LOOP)
        WHT(LOOP) = CNHT(LOOP)
        WPOW(LOOP) = CNPOW(LOOP)
       END IF
      END IF
   40 CONTINUE
     IF (SC .EQ. 1) THEN
      TC = 90
     ELSE
      TC = 0
     END IF
C
     END
C
C
C ***********************************************************************
     SUBROUTINE DTRAPZ(A, B, S, N)
C    Integration routine for Density Of States
C ***********************************************************************
C
     IF (N .EQ. 1) THEN
      S = 0.5 * (B - A) * (DOS(A) + DOS(B))
      IT = 1
     ELSE
      TNM = IT
      DEL = (B - A) / TNM
      X = A + 0.5 * DEL
      SUM = 0.
      DO 10 J = 1, IT
       SUM = SUM + DOS(X)
       X = X + DEL
  10  CONTINUE
      S = 0.5 * (S + (B - A)*SUM/TNM)
      IT = 2 * IT
     END IF
     RETURN
     END
C
C
     SUBROUTINE DSIMP(A, B, S)
     PARAMETER (EPS=1E-2,JMAX=14)
     OST = -1.E30
     OS = -1.E30
     DO 10 J = 1, JMAX
      CALL DTRAPZ(A, B, ST, J)
      S = (4.*ST - OST) / 3.
      IF (ABS(S - OS) .LT. EPS*ABS(OS)) RETURN
      OS = S
      OST = ST
  10 CONTINUE
     PAUSE 'Too many steps when calculating DOS.'
     END
C
C
C ***********************************************************************
     SUBROUTINE TTRAPZ(A, B, S, N)
C    Integration routine for Thermal conductivity
C ***********************************************************************
C
     IF (N .EQ. 1) THEN
      S = 0.5 * (B - A) * (RINTEG(A) + RINTEG(B))
      IT = 1
     ELSE
      TNM = IT
      DEL = (B - A) / TNM
      X = A + 0.5 * DEL
      SUM = 0.
      DO 10 J = 1, IT
       SUM = SUM + RINTEG(X)
       X = X + DEL
  10  CONTINUE
      S = 0.5 * (S + (B - A)*SUM/TNM)
      IT = 2 * IT
     END IF
     RETURN
     END
C
C
     SUBROUTINE TSIMP(A, B, S)
     PARAMETER (EPS=1E-2,JMAX=16)
```

```
      OST = -1.E30
      OS = -1.E30
      DO 10 J = 1, JMAX
       CALL TTRAPZ(A, B, ST, J)
       S = (4.*ST - OST) / 3.
       IF (ABS(S - OS) .LT. EPS*ABS(OS)) RETURN
       OS = S
       OST = ST
   10 CONTINUE
      PAUSE 'Too many steps when calculating K.'
      END
```

Appendix D

Program to Model Connectivity

The program is written in FORTRAN 77 and conforms to the ANSI standard. Functions and subprograms are utilised wherever they may assist with the clarity of structure. Global data is passed to each subroutine or function by use of COMMON blocks which are by necessity large. The program requires the use of the NAG subroutine library to generate a sequence of random numbers. Routine G05CCF is a subroutine which resets the seed for the random number generator to a non-repeatable value. Function G05CAF returns a psuedo-random number of type REAL in the range 0 to 1. These NAG routines be replaced by an appropriate set of routines which generate pseudo random numbers. Meaningful variable name are used throughout the program wherever possible.

The overall conductance of the network is computed by the iterative procedure to be found in routine CHEBY. A description of the method of solution and a table of appropriate acceleration parameters may be found in appendix B.

```
C
C  ************************************************************
C   Program to calculate the effect of a matrix of
C   resistors and capacitors.   This attempts to simulate
C   the effect of thermal conductivity  V fixed, find I
C  ************************************************************
C
      IMPLICIT REAL(A - H,O - Z)
      IMPLICIT INTEGER(I - N)
      INTEGER FLAG
      CHARACTER*12 FIL
      DIMENSION V(50,50,50), S(50,50,50,3)
      DIMENSION SS2(4)
      COMMON /MATRIX/ V, S
      COMMON /VARS/ S1, S2, P1, CONN
      COMMON /VAR2/ N, M, L, NPTS, MN, RHO, ICIN, ICOUT, VTOT
      COMMON /FILE/ FIL
      COMMON /THRESH/ THRESH
      COMMON /LIMIT/ START, END, STEP
C
C  ************************************************************
C  Set up the required arrays and variables
C  S1,S2 and S3 are the possible conductance values
C  P1 is the probability of S1
C  PF is P(S2)/P(S3)
C  ISNUM=3 chooses the first 3 values of SS2
C  G05CBF sets the random number seed to a repeatable value
C  IMAXIT specifies max no of iterations before OMEGA=1
C  ************************************************************
C
      S1 = 1.
      VTOT = 5.
      PF = 0.5
      CALL HEADER
      ISNUM = 3
      ICOUNT = 0
      IMAXIT = 300
      NPTS = M * N * L
      MN = N * M
      ICIN = 1
      ICOUT = M
      CALL G05CCF(10)
C
C  ************************************************************
C  Go around loops, choosing a new matrix each time.
C  Start off the voltage condition as zero, after that use the
C  old solution, this speeds things up a bit. For testing the
C  program, reset the voltage condition each time.
C  ************************************************************
C
      WRITE (6,*)'Enter Tc '
      READ (5,*) TC
      WRITE (6,*)'Enter dTc'
      READ (5,*) DTC
      DO 10 T = 30, 120, 2
       S1 = 0.05
       S2 = T * 0.001
       IF (T .GE. TC) THEN
        P1 = 1
       END IF
       IF (T .LE. (TC - DTC)) THEN
        P1 = 0
       END IF
       IF ((T .GT. (TC - DTC)) .AND. (T .LT. TC)) THEN
        P1 = 1 + (T/DTC) - (TC/DTC)
       END IF
C
       CALL SETUP
C
       ITER = 0
       SOLN = CHEBY(IMAXIT,ITER)
       ICOUNT = ICOUNT + ITER
       IF (SOLN .EQ. - 1.) THEN
        WRITE (8,*) T, SOLN, CONN, RHO, ICOUNT, ITER,
     1        'Failed - too many iterations'
       ELSE
        WRITE (8,*) T, SOLN, CONN, ITER
        WRITE (6,*) 1. - P1, SOLN, ITER
C        CALL VPRINT
C        CALL SPRINT
       END IF
C
   10 CONTINUE
C
      WRITE (8,*) ICOUNT, -1
C
      STOP
      END
C
C
C  ************************************************************
C  Subroutine to pick a value of the conductance
C  The three values are passed in A,B and C.
C  The probability of A is passed into PP1,
C  the relative probabilities of B and C is passed into PP2
C  NOTE that the routine requires use of the NAG subroutine library
C  for the choice of a random number
C  ************************************************************
C
      REAL FUNCTION SVAL(A,B,PP1,CONN)
C
      RAN = G05CAF(RAN)
      RAN2 = G05CAF(RAN2)
      IF (RAN .LE. CONN) THEN
       IF (RAN .LT. PP1) THEN
        SVAL = A
       ELSE
        SVAL = B
       END IF
      ELSE
       SVAL = 0
      END IF
C
      RETURN
      END
C
C
C  ************************************************************
C  Read in the matrix size required, and the output file name
C  then open the file ready for output and put in the data heading
```

```
C  ************************************************************
C
      SUBROUTINE HEADER
      CHARACTER*12 FIL
      CHARACTER*16 DATE
      CHARACTER*8 TIMER
      COMMON /VARS/ S1, S2, P1, CONN
      COMMON /VAR2/ N, M, L, NPTS, MN, RHO, ICIN, ICOUT, VTOT
      COMMON /FILE/ FIL
      COMMON /THRESH/ THRESH
      COMMON /LIMIT/ START, END, STEP
C
      WRITE (6,*)'Enter the file name to store the data'
      READ (5,10) FIL
   10 FORMAT (1A)
      OPEN (8,FILE=FIL,STATUS='NEW')
      WRITE (6,*)'Enter number of rows, columns and planes (M,N,L)'
      READ (5,*) M, N, L
      WRITE (6,*)'Enter the threshold value'
      READ (5,*) THRESH
      WRITE (6,*)'Enter the value of Rho for this matrix size'
      READ (5,*) RHO
      WRITE (6,*)'Enter the Connectivity from 0 to 1'
      READ (5,*) CONN
C
C  Print out options so can be seen on batch output
C  Print out headings onto data file for GPLOT
C
      CALL TIME(21, 0, DATE)
      CALL TIME(4, 0, TIMER)
C
      WRITE (6,20) TIMER, DATE
      WRITE (6,30)
      WRITE (6,40) FIL
      WRITE (6,50) M, N, L
      WRITE (6,60) RHO
      WRITE (6,70) THRESH
      WRITE (6,80) CONN
      WRITE (6,90)
      WRITE (6,100)
C
      WRITE (8,20) TIMER, DATE
      WRITE (8,30)
      WRITE (8,40) FIL
      WRITE (8,50) M, N, L
      WRITE (8,60) RHO
      WRITE (8,70) THRESH
      WRITE (8,80) CONN
      WRITE (8,90)
      WRITE (8,100)
C
   20 FORMAT (1X, 1A, 1X, 1A)
   30 FORMAT (' Program     : MATRIX.V')
   40 FORMAT (' File        : ', 1A)
   50 FORMAT (' Matrix      :', 3I3)
   60 FORMAT (' Rho         :', F7.4)
   70 FORMAT (' Threshold   :', E8.1)
   80 FORMAT (' Connectivity:', F6.3)
   90 FORMAT (' S2 varies  S2=S3')
  100 FORMAT (' ..........')
C
      WRITE (8,*)'Temperature (K)'
      WRITE (8,*)'Conductivity'
      RETURN
      END
C
C
C  ************************************************************
C  This procedure uses a simultaneous over-relaxation method
C  using a Gauss-Siedel type method and Chebyshev acceleration!
C  Needs a value of RHO to be defined before use,
C  ************************************************************
C
      FUNCTION CHEBY(IMAXIT, ITER)
      REAL OMEGA
      INTEGER FLAG
      DIMENSION V(50,50,50), S(50,50,50,3)
      COMMON /MATRIX/ V, S
      COMMON /VAR2/ N, M, L, NPTS, MN, RHO, ICIN, ICOUT, VTOT
      COMMON /THRESH/ THRESH
C
C
C  IMAX1 is the max no of iterations before OMEGA=1
C  IMAX2 is the max no of iterations before giving up!
C
      IMAX1 = IMAXIT * 2
      IMAX2 = INT(FLOAT(IMAX1)*1.4)
      OMEGA = 1.
C
   10 CONTINUE
      FLAG = 0
C
C  Do Chebyshev with odd/even ordering
C
      DO 50 J = 0, 1
       ITER = ITER + 1

       DO 40 IAX = 1, M
        DO 30 IAY = 1, N
         DO 20 IAZ = 1, L
C
          I = IAX + (IAY - 1) * M + (IAZ - 1) * MN
          IF ((MOD(I + J,2) .EQ. MOD(ITER,2)) .AND. (I .NE. NPTS))
     1          THEN
           SUM = 0.
           SUM2 = 0.
C
C     WORK OUT CO-ORDS OF V IN THE EQUATION
C     CHECKING FOR OUT OF RANGE POINTS AT SAME TIME
C
           IX = IAX + J
           IY = IAY
           IZ = IAZ
           IF (IX .GT. M) THEN
            IX = 1
            IY = IY + 1
            IF (IY .GT. N) THEN
             IY = 1
             IZ = IZ + 1
            END IF
```

```
            END IF
C
C  V=0 at IX=ICIN and V=VTOT at IX=ICOUT
C
            IF ((IX .EQ. ICIN) .OR. (IX .EQ. ICOUT)) GO TO 20
C
            SUM = SUM + S(IX,IY,IZ,1) * (V(IX + 1,IY,IZ) - V(IX,IY,IZ))
            SUM2 = SUM2 + S(IX,IY,IZ,1)
            SUM = SUM + S(IX - 1,IY,IZ,1) * (V(IX - 1,IY,IZ) - V(IX,IY,
     1      IZ))
            SUM2 = SUM2 + S(IX - 1,IY,IZ,1)
C
            IF (IY .LT. N) THEN
             SUM = SUM + S(IX,IY,IZ,2) * (V(IX,IY + 1,IZ) - V(IX,IY,IZ))
             SUM2 = SUM2 + S(IX,IY,IZ,2)
            END IF
            IF (IY .GT. 1) THEN
             SUM = SUM + S(IX,IY - 1,IZ,2) * (V(IX,IY - 1,IZ) - V(IX,IY,
     1       IZ))
             SUM2 = SUM2 + S(IX,IY - 1,IZ,2)
            END IF
            IF (IZ .LT. N) THEN
             SUM = SUM + S(IX,IY,IZ,3) * (V(IX,IY,IZ + 1) - V(IX,IY,IZ))
             SUM2 = SUM2 + S(IX,IY,IZ,3)
            END IF
            IF (IZ .GT. 1) THEN
             SUM = SUM + S(IX,IY,IZ - 1,3) * (V(IX,IY,IZ - 1) - V(IX,IY,
     1       IZ))
             SUM2 = SUM2 + S(IX,IY,IZ - 1,3)
            END IF
C
            IF (SUM2 .NE. 0) THEN
             SUM = SUM / SUM2
            ELSE
             SUM = 0
            END IF
C
C     SO NOW STORE OLD VALUE, AND NEW VALUE
C
            CORR = SUM * OMEGA
            IF (ABS(CORR) .GT. THRESH) THEN
             FLAG = 1
            END IF
            V(IX,IY,IZ) = V(IX,IY,IZ) + CORR
           END IF
   20     CONTINUE
   30    CONTINUE
   40   CONTINUE
   50 CONTINUE
C
C  Use Chebyshev acceleration to find new omega
C
      IF (ITER .EQ. 1) THEN
       OMEGA = 1. / (1. - .5*RHO*RHO)
      ELSE
       IF (ITER .GE. IMAX1) THEN
        OMEGA = 1.
       ELSE
        OMEGA = 1. / (1. - .25*OMEGA*RHO*RHO)
       END IF
      END IF
C
C  If all points are within THRESH of the last iteration, stop
C  Stop calculating if too many iterations done
C
      IF (ITER .GT. IMAX2) THEN
       ITER = IMAXIT
       CHEBY = -1.
       RETURN
      END IF
      IF (FLAG .EQ. 1) GO TO 10
C
C  Calculate the overall conductance, with different
C  current vector this will be calculated differently
C
      CURR = 0.
      DO 70 IY = 1, N
       DO 60 IZ = 1, L
        DV = VTOT - V(M - 1,IY,IZ)
        CURR = CURR + (DV*S(M - 1,IY,IZ,1))
   60  CONTINUE
   70 CONTINUE

C
      CHEBY = CURR / VTOT
      ITER = ITER / 2
C
      RETURN
      END
C
C
C  *****************************************************************
C  Now set up the initial voltage condition
C  Each iteration will use the last one as a starting point
C  this may help to speed things up
C  *****************************************************************
C
      SUBROUTINE RESETV
C
      DIMENSION V(50,50,50), S(50,50,50,3)
      COMMON /MATRIX/ V, S
      COMMON /VARS/ S1, S2, P1, CONN
      COMMON /VAR2/ N, M, L, NPTS, MN, RHO, ICIN, ICOUT, VTOT
C
      DO 30 J = 1, N
       DO 20 K = 1, L
        DO 10 I = 2, M - 1
         V(I,J,K) = VTOT / NPTS
   10   CONTINUE
        V(1,J,K) = 0.
        V(M,J,K) = VTOT
   20  CONTINUE
   30 CONTINUE
C
      RETURN
      END
C
C  *****************************************************************
C  Set up the conductance array and store it
C  *****************************************************************
```

```
C
      SUBROUTINE SETUP
C
      DIMENSION V(50,50,50), S(50,50,50,3)
      COMMON /MATRIX/ V, S
      COMMON /VARS/ S1, S2, P1, CONN
      COMMON /VAR2/ N, M, L, NPTS, MN, RHO, ICIN, ICOUT, VTOT
C
      DO 40 IX = 1, M
       DO 30 IY = 1, N
        DO 20 IZ = 1, L
         DO 10 IC = 1, 3
          S(IX,IY,IZ,IC) = SVAL(S1,S2,P1,CONN)
   10    CONTINUE
   20   CONTINUE
   30  CONTINUE
   40 CONTINUE
      DO 60 IY = 1, N
       DO 50 IZ = 1, L
        S(1,IY,IZ,2) = 0.
        S(1,IY,IZ,3) = 0.
        S(M,IY,IZ,2) = 0.
        S(M,IY,IZ,3) = 0.
   50  CONTINUE
   60 CONTINUE
C
      RETURN
      END

C
      SUBROUTINE VPRINT
C
      DIMENSION V(50,50,50), S(50,50,50,3)
      COMMON /MATRIX/ V, S
      COMMON /VARS/ S1, S2, P1, CONN
      COMMON /VAR2/ N, M, L, NPTS, MN, RHO, ICIN, ICOUT, VTOT
C
C
      DO 30 IX = 1, M
       DO 20 IY = 1, N
        WRITE (6,10) (V(IX,IY,J),J=1,L)
   10   FORMAT (8F12.8)
   20  CONTINUE
   30 CONTINUE
      RETURN
      END
C
      SUBROUTINE SPRINT
C
      DIMENSION V(50,50,50), S(50,50,50,3)
      COMMON /MATRIX/ V, S
      COMMON /VARS/ S1, S2, P1, CONN
      COMMON /VAR2/ N, M, L, NPTS, MN, RHO, ICIN, ICOUT, VTOT
C
C
      PRINT *, 'Horizontal'
      DO 30 IZ = 1, L
       DO 20 IY = 1, N
        WRITE (6,10) (S(J,IY,IZ,1),J=1,M - 1)
   10   FORMAT (8F12.8)
   20  CONTINUE
   30 CONTINUE
C
      PRINT *, 'Vertical'
      DO 60 IX = 1, M
       DO 50 IZ = 1, L
        WRITE (6,40) (S(IX,J,IZ,2),J=1,N - 1)
   40   FORMAT (8F12.8)
   50  CONTINUE
   60 CONTINUE
C
      PRINT *, 'Lateral'
      DO 90 IX = 1, M
       DO 80 IY = 1, N
        WRITE (6,70) (S(IX,IY,J,3),J=1,L - 1)
   70   FORMAT (8F12.8)
   80  CONTINUE
   90 CONTINUE
C
      RETURN
      END
```

Appendix E

Data Acquisition Program

The data acquistion program was written in Borland's Turbo Pascal V for an 'IBM PC' compatible computer. The IEEE routines referred to throughout the program were supplied with the Scientific Solutions IEEE card and were converted from the original BASIC to Turbo Pascal by Dr.D.B.Lambrick.

The program is divided up into several units which perform the following functions

THERMAL - main program
MRDGLOB - set up global variables.
MRDUTILS - general purpose input/output utility routines.
DEFAULTS - sets up the default settings for the IEEE devices and program variables.
DEVICES - IEEE device routines for inputting and outputting data to various devices attached to the IEEE bus.

Functions and subprograms are utilised wherever they may assist with the structure. Meaningful variable, procedure and function names are used throughout the program. The data is displayed graphically as it is acquired. The program was written to run on an 'IBM PC' compatible 286 AT machine with 640K RAM, and colour VGA graphics board, fitted with a Scientific Solutions IEEE bus. The program should run on non-AT machines and machines without colour VGA graphics although not all configurations have been tested.

```
{$M 30000,0,65000}

{*************************************************************************}
{*  Program THERMAL.PAS                                                  *}
{*  obtains data for thermal conductivity experiments                    *}
{*  M.R.Delap 1989                                                       *}
{*************************************************************************}

program thermal;

uses defaults,mrdglob,mrdutils,ieee,devices,crt,graph,dos;

const
   version = 2.07;
   version_date='20/09/90';

var dummy : real;

 {*****************THERMAL. PAS*********************************************}
{ These routine convert between the sensor diode voltage and temperature
  using the   fitted chebyshev polynomial coefficients.
  If the DR91C controller is set to read temperature,
  these diode voltage conversions should be omitted.}

function convert_Volt(voltage:real):real;
var
 coeff : Array[1..12] of real;
 x,x1,x2,sum : real;
 loop: integer;
  dum : real;
begin
 if (voltage=0) then sum:=400
               else begin
 coeff[1]:=-2380.58083;  coeff[2]:=-269.16639;  coeff[3]:=6915.02096;
 coeff[4]:=-4890.49141;  coeff[5]:=685.19201;   coeff[6]:=-75.26313;
 coeff[7]:=260.1993;     coeff[8]:=111.8136;    coeff[9]:=-135.50468;
 coeff[10]:=-20.49711;   coeff[11]:=33.84063;   coeff[12]:=-6.34534;
 x1:=1;
 x2:=voltage;
 sum:=x1*coeff[1]+x2*coeff[2];
 for loop:=3 to 12 do begin
     x:=2*voltage*x2-x1;
     sum:=sum+coeff[loop]*x;
     x1:=x2; x2:=x;
 end;
 end;
 if sum>400 then sum:=400;
 if sum<0 then sum:=0;
 convert_volt:=sum;
end;


function convert_Temp(temperature:real):real;
var
 coeff : Array[1..12] of real;
 x,x1,x2,sum : real;
 loop: integer;
begin
 {Coefficients taken from fitten 7th order chebyshev polynomial}
 coeff[1]:=1.4154967767;     coeff[2]:=-9.3711538784e-4;
 coeff[3]:=1.1101416234e-5;  coeff[4]:=-1.2055536247e-7;
 coeff[5]:=4.3326955522e-10; coeff[6]:=-7.0226932006e-13;
 coeff[7]:=4.2746418359e-16;
 x1:=1;
 x2:=temperature;
 sum:=x1*coeff[1]+x2*coeff[2];
 for loop:=3 to 7 do begin
     x:=2*temperature*x2-x1;
     sum:=sum+coeff[loop]*x;
     x1:=x2; x2:=x;
 end;
 convert_temp:=sum;
end;


procedure hold(instring:str255);
begin
  RestoreCrtMode;
  writeln(instring);
  space;
  SetGraphMode(GraphMode);
end;


procedure header;
begin
  Window(1;1,80,5);
  if colour then TextBackground(Red);
  ClrScr;
  if colour then TextColor(White);
  writeln;
  writeln('                    Thermal Conductivity data collection program');
  writeln('                            Version ',version:5:2,' ',version_date);
  writeln('                             by M.R.Delap');
  if colour then TextBackground(LightGray);
  if colour then TextColor(Black);
  Window(1,6,80,25);
  ClrScr;
end; {header}


procedure title;
begin
  header;
  writeln;
  writeln('Ensure that the apparatus is set up correctly');
  writeln('Check ');
  writeln('1. The sample is in place');
  writeln('2. The thermocouple wires are ok');
  writeln('3. The sample space is evacuated');
  writeln('4. All meters are connected to the IEEE port');
  writeln('5. The MINICAM has been reset');
  writeln;
  space;
  header;
end; {title}


 function auto_min(max,min:real): real;
var
```

```
 temp,dummy : real;
 newd       : integer;
 sign,dum2  : real;
begin
  if min<>0 then sign:=min/Abs(min)
            else sign:=-1;
  if sign>0 then begin
                   temp:=log(Abs(min));
                   dummy:=Frac(temp);
                   newd:=trunc(power_of(10,dummy));
                   dum2:=newd*power_of(10,(Int(temp)));
                 end
                 else
                   dum2:=min;
  if (dum2=max) then dum2:=max-1;
auto_min:=dum2;
end; {auto_min}


function auto_step(max,min:real): real;
var
 step,temp,dummy,dvs : real;
 newd                : integer;
 ans                 : real;
begin
  step:=(max-min)/8;
  temp:=log(step);
  dummy:=Frac(temp);
  newd:=trunc(power_of(10,dummy));
  if (newd=0) then dvs:=1;
  if (newd=1) then dvs:=1;
  if (newd=2) then dvs:=2;
  if (newd=3) then dvs:=2.5;
  if ((newd>=4) and (newd<=7)) then dvs:=5;
  if (newd>=8) then dvs:=10;
  ans:=dvs*power_of(10,(Int(temp)));
  if (min+2*ans)>max then ans:=ans/10;
  if (min+3*ans)>max then ans:=ans/5;
  if (min+4*ans)>max then ans:=ans/2;
  auto_step:=ans;
end; {auto_step}


procedure find_max_and_min;
var
 loop : integer;
begin
 if point>2 then begin
                   Xmin:=1E34; Xmax:=-1E34;
                   Ymin:=1E34; Ymax:=-1E34;
                   for loop:=1 to point do
                   begin
                     if temp[loop]<Xmin then Xmin:=temp[loop];
                     if temp[loop]>Xmax then Xmax:=temp[loop];
                     if kval[loop]<Ymin then Ymin:=kval[loop];
                     if kval[loop]>Ymax then Ymax:=kval[loop];
                   end; {loop}
                 end; {if}
end; {find_max_and_min}
```

```
{*****************THERMAL. PAS*********************************************}


procedure display_devices;
var
 dummy : integer;
begin
  for dummy:=1 to no_of_devices do
    writeln(dummy,'.....',device_name[dummy]);
end;  {display_devices}


procedure display_meters;
begin
  writeln;
  writeln(' Meter              device no      device name     IEEE channel');
  writeln;
  writeln('Heater current          ',Imeter.device,'       ',
    device_name[Imeter.device],'   ',Imeter.chan);
  writeln('Heater voltage          ',Vmeter.device,'       ',
    device_name[Vmeter.device],'   ',Vmeter.chan);
  writeln('Nanovoltmeter           ',nVmeter.device,'      ',
    device_name[nVmeter.device],'   ',nVmeter.chan);
  writeln('Temperature controller  ',Tcontrol.device,'      '
    ,device_name[Tcontrol.device],'   ',Tcontrol.chan);
  writeln('nV power controller     ',nVcontrol.device,'      ',
    device_name[nVcontrol.device],'   ',nVcontrol.chan);
  writeln;
end;


procedure device_init_display(device: integer);
begin;
  RestoreCRTmode;
  Cls;
  writeln('At device initialisation');
  writeln('Device number is ',device);
  writeln('Device name is ',device_name[device]);
  writeln('Device channel is ',device_default_chan[device]);
  space;
end; {device_init_display}


procedure device_init(device:integer);
begin
  {device_init_display(device)}
  if (device=k197) then init_k197;
  if (device=k175) then init_k175;
  if (device=dr91c) then init_dr91c;
  if (device=solar) then init_solar;
  if (device=fluke8860) then init_fluke8860;
  if (device=weston) then init_weston;
  {device_init_display}
 end; {device_init}


procedure initialise_devices;
begin
  init_ieee;
```

```
  device_init(Imeter.device);
  device_init(Vmeter.device);
  device_init(nVmeter.device);
  device_init(Tcontrol.device);
end;{initialise_devices}


procedure device_defaults;
var
  dummy : integer;
 begin
  zero_add;
  meter_defaults;

  {Note that meter defaults is external so that the may be easily
  changed without hunting  out this section}

  for dummy:=1 to no_of_devices do
  begin
    devla[dummy].prim:=device_default_chan[dummy];
    devta[dummy].prim:=device_default_chan[dummy];
  end;
  Imeter.chan:=device_default_chan[Imeter.device];
  Vmeter.chan:=device_default_chan[Vmeter.device];
  nVmeter.chan:=device_default_chan[nVmeter.device];
  Tcontrol.chan:=device_default_chan[Tcontrol.device];
  nVcontrol.chan:=device_default_chan[nVcontrol.device];
  Imeter.name:=device_name[Imeter.device];
  Vmeter.name:=device_name[Vmeter.device];
  nVmeter.name:=device_name[nVmeter.device];
  Tcontrol.name:=device_name[Tcontrol.device];
  nVcontrol.name:=device_name[nVcontrol.device];
 end; {device_defaults}


function devices_ok : boolean;
var dummy : boolean;
begin
  dummy:=true;
  if (Imeter.chan=Vmeter.chan) or (Imeter.chan=nVmeter.chan)
     or (Imeter.chan=Tcontrol.chan) then dummy:=false;
  if (Vmeter.chan=nVmeter.chan) or (Vmeter.chan=Tcontrol.chan)
      or (nVmeter.chan=Tcontrol.chan) then dummy:=false;
  if (Imeter.device=Vmeter.device) or (Imeter.device=nVmeter.device)
     or (Imeter.device=Tcontrol.device) then dummy:=false;
  if (Vmeter.device=nVmeter.device) or (Vmeter.device=Tcontrol.device)
      or (nVmeter.device=Tcontrol.device) then dummy:=false;
  if (nVcontrol.chan=Imeter.chan) or (nVcontrol.chan=Vmeter.chan)
     or (nVcontrol.chan=nVmeter.chan) or (Imeter.chan=Tcontrol.chan)
      then dummy:=false;
  if ((Imeter.chan=pc_ieee) or (Vmeter.chan=pc_ieee) or
      (Tcontrol.chan=pc_ieee) or (nVmeter.chan=pc_ieee)) then dummy:=false;
  devices_ok:=dummy;
end;  {meters_ok}


function device_setup_ok:boolean;
begin;
  header;
  display_meters;
  if devices_ok then
             begin
               write('Are these meters and IEEE channels correct ? ');
               device_setup_ok:=yes_or_no(true);
             end
             else
             begin
               device_setup_ok:=false;
               writeln('These devices are incorrectly set up');
               space;
             end;
  header;
 end;


procedure change_device_parameters;
var  ok : boolean;
begin
  while (NOT device_setup_ok) do
  begin
    header;
    display_meters;
    display_devices;
    write('Enter the heater ammeter: 1 to ',no_of_devices:2,' ? ');
    Imeter.device:=input_integer(Imeter.device,1,no_of_devices,false);
    Imeter.name:=device_name[Imeter.device];
    Imeter.chan:=device_default_chan[Imeter.device];
    header;
    display_meters;
    display_devices;
    writeln('Heater ammeter...',device_name[Imeter.device]);
    writeln('IEEE channel......',Imeter.chan);
    writeln;
    write('Is the IEEE channel ok ? ');
    ok:=yes_or_no(true);
    if (not ok) then begin
   write('Enter the ieee channel number: 1 to ',ieee_no_of_channels:2,' ? ');
           Imeter.chan:=input_integer(Imeter.chan,1,
                                          ieee_no_of_channels,false);
                    end;
    header;
    display_meters;
    display_devices;
    write('Enter the heater voltmeter: 1 to ',no_of_devices:2,' ? ');
    Vmeter.device:=input_integer(Vmeter.device,1,no_of_devices,false);
    Vmeter.name:=device_name[Vmeter.device];
    Vmeter.chan:=device_default_chan[Vmeter.device];
    header;
    display_meters;
    writeln('Heater voltmeter...',device_name[Vmeter.device]);
    writeln('IEEE channel......',Vmeter.chan);
    writeln;
    write('Is the IEEE channel ok ? ');
    ok:=yes_or_no(true);
    if (not ok) then begin
  write('Enter the ieee channel number: 1 to ',ieee_no_of_channels:2,' ? ');
                   Vmeter.chan:=input_integer(Vmeter.chan,1,
                            ieee_no_of_channels,false);
                   end;
    header;
```

```
    display_meters;
    display_devices;
    write('Enter the nanovoltmeter output device: 1 to ',no_of_devices:2,' ? ');
    nVmeter.device:=input_integer(nVmeter.device,1,no_of_devices,false);
    nVmeter.name:=device_name[nVmeter.device];
    nVmeter.chan:=device_default_chan[nVmeter.device];
    header;
    display_meters;
    writeln('nVmeter device...',device_name[nVmeter.device]);
    writeln('IEEE channel......',nVmeter.chan);
    writeln;
    write('Is the IEEE channel ok ? ');
    ok:=yes_or_no(true);
    if (not ok) then begin
  write('Enter the ieee channel number: 1 to ',ieee_no_of_channels:2,' ? ');
              nVmeter.chan:=input_integer(nVmeter.chan,1,
                                          ieee_no_of_channels,false);
                       end;
    header;
    display_meters;
    display_devices;
  write('Enter the temperature control device: 1 to ',no_of_devices:2,' ? ');
    Tcontrol.device:=input_integer(Tcontrol.device,1,no_of_devices,false);
    Tcontrol.name:=device_name[Tcontrol.device];
    Tcontrol.chan:=device_default_chan[Tcontrol.device];
    header; display_meters;
    writeln('Temperature controller...',device_name[Tcontrol.device]);
    writeln('IEEE channel......',Tcontrol.chan);
    write('Is the IEEE channel ok ? ');
    ok:=yes_or_no(true);
    if (not ok) then begin
  write('Enter the ieee channel number: 1 to ',ieee_no_of_channels:2,' ? ');
               Tcontrol.chan:=input_integer(Tcontrol.chan,1,
                                            ieee_no_of_channels,false);
                       end;
    header;
    display_meters;
    display_devices;
    write('Enter the nV meter power switching device: 1 to ',
                                                 no_of_devices:2,' ? ');
    nVcontrol.device:=input_integer(nVcontrol.device,1,no_of_devices,false);
    nVcontrol.name:=device_name[nVcontrol.device];
    nVcontrol.chan:=device_default_chan[nVcontrol.device];
    header; display_meters;
    writeln('nV power switching...',device_name[Tcontrol.device]);
    writeln('IEEE channel......',nVcontrol.chan);
    write('Is the IEEE channel ok ? ');
    ok:=yes_or_no(true);
    if (not ok) then begin
  write('Enter the ieee channel number: 1 to ',ieee_no_of_channels:2,' ? ');
               nVcontrol.chan:=input_integer(nVcontrol.chan,1,
                                            ieee_no_of_channels,false);
                       end;
  end;

{Again need to set device listen and talk addresses for IEEE routines, as
they require a device number to be passed rather than listen and
talk addresses.}
  devla[Imeter.device].prim:=Imeter.chan;
  devta[Imeter.device].prim:=Imeter.chan;
  devla[Vmeter.device].prim:=Vmeter.chan;
  devta[Vmeter.device].prim:=Vmeter.chan;
  devla[nVmeter.device].prim:=nVmeter.chan;
  devta[nVmeter.device].prim:=nVmeter.chan;
  devla[Tcontrol.device].prim:=Tcontrol.chan;
  devta[Tcontrol.device].prim:=Tcontrol.chan;
  devla[nVcontrol.device].prim:=nVcontrol.chan;
  devta[nVcontrol.device].prim:=nVcontrol.chan;
end; {alter_defaults}


{*****************THERMAL. PAS*******************************************}


procedure variable_defaults;
var
 dumtime : time;
 dumstr  : str20;
begin
  path:=default_path;
  filename:=file_version(default_filename,path,'.DAT');
  with min_time do begin
       hour:=0;minute:=0;second:=1;day:=1;month:=1;year:=0;
       end;
  with max_time do begin
       hour:=23;minute:=59;second:=59;day:=31;month:=12;year:=99;
       end;
  pause:=min_time;
  next_reading_time:=min_time;
  end_time:=max_time;
  timeout:=false;
  system_clock(dumtime);
  compress_clock(dumtime,dumstr);
  old_time:=min_time;
  next_reading_time:=min_time;
  next_setpoint_time:=min_time;
  free_time:=min_time;
  nV_ontime:=min_time;
  point:=1;
  no_of_setpoints:=0;
  setpoint_index:=0;
  setpoint_time[0]:=min_time;
  setpoint_temperature[0]:=1;
  no_more_setpoint_changes:=false;
  setpoints_sorted:=false;
  nV_old_status:=true;
  thermocouple_file:=thermocouple_default_file;
  var_defaults;
  {Note that VAR_DEFAULTS is in the unit DEFAULTS so that they may
   be easily aletered without finding this routine!!!}
end; {variable_defaults}


procedure display_variable_parameters;
begin
  header;
  writeln;
  writeln('Path for filename      ',path);
  writeln('Filename to store data ',filename);writeln;
  writeln('Sample width            = ',width:6:2,' mm');
```

```
  writeln('Sample thickness       = ',thick:6:2,' mm');
  writeln('Thermocouple separation = ',llength:6:2,' mm');writeln;
  writeln('Scale on nV meter in uV    = ',nVscale:5:0,' uV');
  writeln('nV meter offset in uV      = ',manual_offset:8:3,' uV');
  writeln('Voltmeter fsd on nV output = ',Vfsd:8:6,' Volts');
 end; {display_variable_parameters}


function variable_parameters_ok:boolean;
begin
  display_variable_parameters;
  writeln;
  write('Are these parameters OK ? ');
  variable_parameters_ok:=yes_or_no(true);
end; {variable_parameters_ok}


procedure change_variable_parameters(start:boolean);
begin
  while (NOT variable_parameters_ok) do
  begin;
  display_variable_parameters;
  writeln;
  path:=get_pathname(path);
  display_variable_parameters;
  writeln;
  filename:=get_filename(filename,path,'.DAT',start);
  display_variable_parameters;
  writeln;
  write('Enter the width of the sample: 0 to 50mm ? ');
  width:=input_real(width,0,50,false);
  display_variable_parameters;
  writeln;
  write('Enter the thickness of the sample: 0 to 50mm ? ');
  thick:=input_real(thick,0,50,false);
  display_variable_parameters;
  writeln;
 write('Enter the distance between the thermocouple wires: 0 to 50mm ? ');
  llength:=input_real(llength,0,50,false);
  writeln;
  area:=width*thick;
  display_variable_parameters;
  writeln;
  write('Enter the scale on the nanovoltmeter: 0 to 1000 uV ? ');
  nVscale:=input_real(nVscale,0,1000,false);
  display_variable_parameters;
  writeln;
write('Enter the manual offset on the nanovoltmeter: -1000 to 1000 uV ? ');
  manual_offset:=input_real(manual_offset,-1000,1000,false);
  display_variable_parameters;
  writeln;
  writeln('Enter the voltage for fsd from the nVmeter analogue output');
  write('From 0 to 10 volts ? ');
  Vfsd:=input_real(Vfsd,0,10,false);
  header;
  end; {while}
end; {change_variable_paramters}


procedure enter_intervals;
var
 dumstr  : str20;
 default : time;
begin
  header;
  writeln;
  writeln('The system clock will run throughout the experiment');
  writeln('Readings will be taken at constant intervals.');
  writeln;
  writeln('Enter the interval between readings in the form HH:MM:SS');
  dumstr:=default_interval+null_date;
  expand_clock(dumstr,default);
  input_time(default,interval,false);
  writeln;
  writeln('Enter the delay after changing set point in the form HH:MM:SS');
  dumstr:=default_pause+null_date;
  expand_clock(dumstr,default);
  input_time(default,pause,false);
  header;
  writeln;
  system_clock(start_time);
  compress_clock(start_time,start_str);
  writeln('The start time of this program was ',start_str);
  repeat
    writeln('Enter the time at which data collection is to end');
    writeln('Enter the time in the form HH:MM:SS use >24hrs if necessary');
    dumstr:=copy(start_str,9,9);
    dumstr:=default_end_time+dumstr;
    expand_clock(dumstr,default);
    input_time(default,end_time,false);
    compress_clock(end_time,dumstr);
    writeln('The program will finish at ',dumstr);
  until compare_time(end_time,start_time);
  space;
end; {enter_intervals}


{*****************THERMAL. PAS*****************************************}


procedure display_graphics_variables;
begin
  header;
  writeln;
  writeln('Auto rescale ',auto_scale);
  writeln('Minimum temperature value plotted ',Xmin:6:3,' K');
  writeln('Maximum temperature value plotted ',Xmax:6:3,' K');
  writeln;
  writeln('Minimum K value plotted ',Ymin:6:3,' W/m/K');
  writeln('Minimum K value plotted ',Ymax:6:3,' W/m/K');
  writeln;
end;


function graphics_variables_ok : boolean;
begin
 display_graphics_variables;
 writeln;
 if ((Xmax<Xmin) or (Ymax<Ymin)) then
                 begin
```

```
                graphics_variables_ok:=false;
              end
              else
              begin
                write('Are these variables ok ? ');
                graphics_variables_ok:=yes_or_no(true);
              end;
 end; {graphics_variables_ok}


procedure change_graphics_variables;
var
 dummy : integer;
begin
  while (NOT graphics_variables_ok) do
  begin
    writeln;
    write('Do you wish to have auto_scaling ? ');
    auto_scale:=yes_or_no(auto_scale);
      if (NOT auto_scale) then
      begin
        display_graphics_variables;
        write('Enter the minimum display temperature: 0 to 300K ? ');
        Xmin:=input_real(Xmin,0,300,false);
        display_graphics_variables;
   write('Enter the maximum display temperature: ',Xmin:5:2,' to 300K ? ');
        Xmax:=input_real(Xmax,Xmin,300,false);
        display_graphics_variables;
        write('Enter the minimum display value of K in W/m/K ? ');
        Ymin:=input_real(Ymin,0,1000,false);
        display_graphics_variables;
        write('Enter the maximum display value of K in W/m/K ? ');
        Ymax:=input_real(Ymax,Ymin,1000,false);
      end; {if}
    end; {while}
  if ((Xmin-Xmax)=0) then
                       begin
                         Xmax:=Xmin+1;
                         Xstep:=screen_X_axis/auto_step(Xmax,Xmin);
                       end
                     else
                       begin
                         Xmax:=Xmin+1;
                         Xstep:=screen_X_axis/(Xmin-Xmax);
                       end;
  if ((Ymin-Ymax)=0) then
                         Ystep:=screen_Y_axis/auto_step(Ymax,Ymin)
                       else
                     Ystep:=screen_Y_axis/(Ymin-Ymax);
end; {change_graphics_variables}


procedure initialise_graphics_variables;
begin
  auto_scale:=false;
  Xmin:=0; Xmax:=300;
  Ymin:=0; Ymax:=5;
  Xstep:=screen_X_axis/(Xmin-Xmax);
  Ystep:=screen_Y_axis/(Ymin-Ymax);
  change_graphics_variables;
end; {initialise_graphics_variables}


{*****************THERMAL. PAS*****************************************}


procedure list_setpoints;
var
 dummy  : integer;
 setstr : str20;
begin
 header; writeln;
 writeln('Point no     Set time       Set temp');
  for dummy:=1 to no_of_setpoints do begin
    if ((dummy mod no_of_lines)=0) then begin
                                     space;
                                     header; writeln;
                           writeln('Point no     Set time       Set temp');
                                end; {if}
    compress_clock(setpoint_time[dummy],setstr);
    writeln('  ',dummy,'      ',setstr,'  ',
                            setpoint_temperature[dummy]:7:3);
  end; {dummy}
  if ((no_of_setpoints mod no_of_lines)<>0) then space;
  cls;
end; {list_setpoints}


procedure sort_setpoints;
var
 loop1,loop2  : integer;
 dummy        : str20;
 dummyreal    : real;
 dumtime      : time;
{Noddy bubble sort - this can be made more efficient!}
begin
writeln;
writeln('Sorting the setpoints into order.. this may take a while..');
  for loop1:=1 to no_of_setpoints do
  begin
    for loop2:=1 to no_of_setpoints do
    begin
      if compare_time(setpoint_time[loop2],setpoint_time[loop1]) then
  begin
          dumtime:=setpoint_time[loop1];
          setpoint_time[loop1]:=setpoint_time[loop2];
          setpoint_time[loop2]:=dumtime;
          dummyreal:=setpoint_temperature[loop1];
          setpoint_temperature[loop1]:=setpoint_temperature[loop2];
          setpoint_temperature[loop2]:=dummyreal;
        end;
    end; {loop2}
  end; {loop1}
  setpoints_sorted:=true;
end; {sort_setpoints}


procedure get_time(index : integer;var ok : boolean);
var
```

```
 dumstr              : str20;
 default,answer      : time;
 def_setpoint_change : time;
begin
  header;
  writeln;
  writeln('Set point change number :',index);
  writeln('Enter the time in HH:MM:SS or -1 to end');
  normalise_clock(start_time);
  compress_clock(start_time,dumstr);
  clock1:=copy(dumstr,1,8);
  clock2:=copy(dumstr,10,8);
  writeln('The starting time was ',clock1);
  writeln('The starting date was ',clock2);
  writeln;
  with def_setpoint_change do begin
       hour:=default_setpoint_hour;
       minute:=default_setpoint_min;
       second:=default_setpoint_second;
       day:=0; month:=0; year:=0;
  end;
  add_time(start_time,def_setpoint_change,default);
  input_time(default,answer,true);
  if answer.hour=255 then ok:=false;
  if ok then begin
     normalise_clock(answer);
     setpoint_time[index]:=answer;
     compress_clock(setpoint_time[index],dumstr);
     writeln(dumstr);
  end; {if}
end; {get_time}


procedure get_temperature(index : integer;var ok : boolean;default:real);
var
 dumstr : str20;
begin
  header;
  writeln;
  writeln('Setpoint change number ',index);
  compress_clock(setpoint_time[index],dumstr);
  writeln('Setpoint change time   ',dumstr);
  writeln;
  write('Enter the setpoint at this time : 0 to 300K ? ');
  setpoint_temperature[index]:=input_real(default,0,300,true);
  if (setpoint_temperature[index]=-1) then ok:=false;
end; {get_temperature}


procedure enter_setpoints;
var
 interval_start_index         : integer;
 interval_time                : integer;
 dummy                        : time;
 answer                       : str8;
 ok, constant_time,started    : boolean;
 clockstr                     : str20;
 interval_start_time          : time;
 clock1,clock2                : str12;
 start_index                  : integer;
 loop_start_temp,loop_temp_interval : real;
 dummy2                       : real;
 dumstr                       : str20;
begin
  header;
  writeln;
  writeln('The temperature controller can be programmed to change the setpoint');
  writeln('at particular times.  You must specify the times at which the');
  writeln('controller is to change the setpoint');
  writeln;
  writeln('Enter the times in the form HH:MM:SS, if you wish to go');
  writeln('over to the next day, use hours greater than 24.');
  writeln('For example,');
  writeln('11.30pm is 23:30:00, 1.30am on the following day is 25:30:00');
  writeln;
  writeln('You may program the computer to change setpoints at regular');
  writeln('intervals, this eases typing.');
  writeln;
write('Do you wish to enter the setpoint time at constant time intervals ? ');
  constant_time:=yes_or_no(constant_time);
  header;
  writeln;
  started:=false;
  ok:=true;
  start_index:=no_of_setpoints;
  while (((no_of_setpoints+1)<max_setpoints) and ok) do
  begin
    no_of_setpoints:=no_of_setpoints+1;
    if (NOT constant_time) then
        begin
          get_time(no_of_setpoints,ok);
          if ok then get_temperature(no_of_setpoints,ok,100);
        end;
    if (constant_time and started) then
      begin
        with dummy do begin
          hour:=0; second:=0; day:=0; month:=0; year:=0;
        end;
        dummy.minute:=(no_of_setpoints-interval_start_index)*interval_time;
        add_time(interval_start_time,dummy,setpoint_time[no_of_setpoints]);
        normalise_clock(setpoint_time[no_of_setpoints]);
        compress_clock(setpoint_time[no_of_setpoints],dumstr);
        writeln('Next setpoint change is at ',dumstr);
        dummy2:=((no_of_setpoints-interval_start_index)
                                        *loop_temp_interval)+loop_start_temp;
        get_temperature(no_of_setpoints,ok,dummy2);
      end;
    if (constant_time and (NOT started)) then
       begin
    write('Enter the interval (in minutes) between setpoint changes ? ');
          interval_time:=input_integer(default_setpoint_interval_time,
                                                        1,10000,false);
          writeln;
          writeln('Enter the time for the interval loop to',
          ' start in HH:MM:SS');
          get_time(no_of_setpoints,ok);
          interval_start_index:=no_of_setpoints;
          interval_start_time:=setpoint_time[no_of_setpoints];
          no_of_setpoints:=no_of_setpoints-1;
```

```
            header; writeln;
            writeln('Loop start point is ',interval_start_index);
            compress_clock(interval_start_time,dumstr);
            writeln('Loop start time is  ',dumstr);
            writeln;
            write('Enter the start temperature of the loop: 0 to 325K ? ');
            loop_start_temp:=input_real(100,0,325,false);
            writeln;
            write('Enter the temperature step of the loop: -325 to 325 K  ');
            loop_temp_interval:=input_real(-1,-325,325,false);
          end; {if}
          header;
    end; {while}
    no_of_setpoints:=no_of_setpoints-1;
    header;
    if (start_index<>no_of_setpoints) then sort_setpoints;
    if (NOT setpoints_sorted) then sort_setpoints;
    no_more_setpoint_changes:=false;
  end; {enter_setpoints}


procedure alter_setpoints;
const
  delete = 1;
  add    = 2;
  change = 3;
  list   = 4;
var
 answer, dummy,loop : integer;
 no_of_choices      : integer;
 ok,flag            : boolean;
 dumstr             : str20;
begin
  no_of_choices:=5;
  repeat
    header; writeln;
    writeln('Setpoint options list');
    writeln;
    writeln('1... delete setpoint');
    writeln('2... add setpoints');
    writeln('3... change setpoint');
    writeln('4... list setpoints');
    writeln('5... end this section');writeln;
    write('Choose the option you require: 1 to 5 ? ');
    answer:=input_integer(5,1,5,false);
    header;
    writeln;
    if ((answer=delete) and (no_of_setpoints>=1)) then
      begin
        write('Enter the point to delete: 1 to ',no_of_setpoints:3,' ? ');
 dummy:=input_integer(0,1,no_of_setpoints,false);
              no_of_setpoints:=no_of_setpoints-1;
 for loop:=dummy to no_of_setpoints do
 begin
     setpoint_time[loop]:=setpoint_time[loop+1];
     setpoint_temperature[loop]:=setpoint_temperature[loop+1];
 end; {loop}
      end;
    if (answer=add) then enter_setpoints;
    if (answer=change) then
              begin
        write('Enter the point to change: 1 to ',no_of_setpoints:3,' ? ');
 dummy:=input_integer(0,1,no_of_setpoints,false);
              writeln;
              compress_clock(start_time,dumstr);
              dumstr:=copy(dumstr,10,8);
              writeln('Start date ',dumstr);
              compress_clock(setpoint_time[dummy],dumstr);
              write('point no = ',dummy,'  Time = ',dumstr,
                    '  setpoint ',setpoint_temperature[dummy]:7:3,' ? ');
              flag:=yes_or_no(true);
              if ((dummy<=no_of_setpoints) and flag) then
                                    begin
                                     setpoints_sorted:=false;
                                get_time(dummy,ok);
                                     get_temperature(dummy,ok,100);
                                    end; {if}
            sort_setpoints;
            end; {if}
    if (answer=list) then list_setpoints;
  until (answer=no_of_choices);
  if (NOT setpoints_sorted) then sort_setpoints;
end; {alter_setpoints}


{*****************THERMAL. PAS*****************************************}


procedure input_gain_variation(index:integer;var ok : boolean);
begin
 header; writeln;
 writeln('Gain variation number ',index);
 write('Enter the lower temperature of the band: 0 to 325K ? ');
 temp_variation_lower[index]:=input_real(10,0,325,true);
 if (temp_variation_lower[index]=-1) then ok:=false;
 if ok then begin
   write('Enter the upper temperature of the band: 0 to 325K ? ');
   temp_variation_upper[index]:=input_real(temp_variation_lower[index]+0.01,
                               0,325,false);
   writeln('Enter the gain variation for this band');
   write('From ',min_gain:5:2,' to ',max_gain:5:2,' ? ');
   gain_variation[index]:=input_real(1.5,min_gain,max_gain,false);
   writeln('Enter the rate variation for this band');
   write('From ',min_rate:5:2,' to ',max_rate:5:2,' ? ');
   rate_variation[index]:=input_real(0.0,min_rate,max_rate,false);
   writeln('Enter the reset variation for this band');
   write('From ',min_reset:5:2,' to ',max_reset:5:2,' ? ');
   reset_variation[index]:=input_real(0.0,min_reset,max_reset,false);
   write('Enter the heater range for this band: ',
                min_heater_power:2,' to ',max_heater_power:2,' ? ');
   heater_range_variation[index]:=input_integer(4,min_heater_power,
                     max_heater_power,false);
   end; {if}
end; {input_gain_variation}


procedure list_gain_variations;
var
 loop  : integer;
```

```
  header; writeln;
  writeln('Temperature controller gain settings'); writeln;
writeln('Band   Min temp  Max temp   Gain    Rate   Reset    Heater power');
  for loop:=1 to no_of_gain_variations do begin
    if ((loop mod no_of_lines)=0) then begin
                          space;
                          header; writeln;
                          writeln('Temperature controller gain settings');
                          writeln;
writeln('Band   Min temp  Max temp   Gain    Rate   Reset    Heater power');
                        end; {if}
    writeln(' ',loop,'     ',temp_variation_lower[loop]:7:3,'    ',
          temp_variation_upper[loop]:7:3,'   ',
          gain_variation[loop]:5:2,'  ',rate_variation[loop]:5:2,'  ',
          reset_variation[loop]:5:2,'     ',heater_range_variation[loop]);
  end; {loop}
  if ((no_of_gain_variations mod no_of_lines)<>0) then space;
  cls;
end; {list_gain_variations}


procedure enter_gain_variations;
var
 ok          : boolean;
 start_index : integer;
begin
  header;
  writeln;
  writeln('At the prompt, enter the value you wish to use.');
  writeln('Enter -1 for the lower temperature limit to end.');
  writeln;
  start_index:=no_of_gain_variations;
  while (((no_of_gain_variations+1)<max_gain_variations) and ok) do
  begin
    header;
    no_of_gain_variations:=no_of_gain_variations+1;
    input_gain_variation(no_of_gain_variations,ok);
     end; {while}
  no_of_gain_variations:=no_of_gain_variations-1;
  header;
 end; {enter_gain_variations}


procedure alter_gain_variations;
const
  delete = 1;
  add    = 2;
  change = 3;
  list   = 4;
var
 answer, dummy,loop : integer;
 no_of_choices      : integer;
 ok,flag            : boolean;
begin
  no_of_choices:=5;
  ok:=true;
  header;
  writeln;
  writeln('The temperature controller can be programmed to change the');
  writeln('gain, rate, reset and heater power depending upon');
  writeln('the temperature band within which it is operating.');
  writeln;
  writeln('This allows you to make allowance for the change in');
  writeln('time constant, sensor sensitivity and cooling power');
  writeln('as the temperature falls. For example');
  writeln;
  writeln('Min temp  Max temp  gain   rate  reset  heater range');
  writeln(' 20.000    30.000   1.30   3.40   4.50    5');
  writeln;
  writeln('Note that there are defaults already programmed');
  space;
  repeat
    header;
    writeln;
    writeln('Temperature band selection');
    writeln;
    writeln('1... delete a temperature band');
    writeln('2... add a temperature band');
    writeln('3... change a temperature band');
    writeln('4... list temperature bands');
    writeln('5... end this section'); writeln;
    write('Choose the option you require: 1 to 5 ? ');
    answer:=input_integer(5,1,5,false);
    header;
    writeln;
    if ((answer=delete) and (no_of_gain_variations>=1)) then
         begin
    write('Enter the point to delete: 1 to ',no_of_gain_variations:3,' ? ');
    dummy:=input_integer(0,1,no_of_gain_variations,false);
          if (dummy<>0) then begin
           no_of_gain_variations:=no_of_gain_variations-1;
           for loop:=dummy to no_of_gain_variations do
           begin
             temp_variation_lower[loop]:=temp_variation_lower[loop+1];
      temp_variation_upper[loop]:=temp_variation_upper[loop+1];
             gain_variation[loop]:=gain_variation[loop+1];
             rate_variation[loop]:=rate_variation[loop+1];
             reset_variation[loop]:=reset_variation[loop+1];
             heater_range_variation[loop]:=heater_range_variation[loop+1];
           end; {loop}
         end; {if}
  end;
    if (answer=add) then enter_gain_variations;
    if (answer=change) then
             begin
    write('Enter the point to change: 1 to ',no_of_gain_variations:3,' ? ');
dummy:=input_integer(0,1,no_of_gain_variations,false);
             if (dummy<>0) then begin
               writeln;
  writeln(dummy,'  ',temp_variation_lower[dummy]:7:3,temp_variation_upper[dummy]:7:
             gain_variation[dummy]:5:2,rate_variation[dummy]:5:2,
             reset_variation[dummy]:5:2,heater_range_variation[dummy]);
             flag:=yes_or_no(true);
             ok:=true;
             if ((dummy<=no_of_gain_variations) and flag) then
                                      input_gain_variation(dummy,ok);
           end; {if}
           end; {if}
    if (answer=list) then list_gain_variations;
```

```
end; {alter_gain_variations}


procedure set_gain_etc(temperature:real);
var
  loop : integer;
begin
  for loop:=1 to no_of_gain_variations do begin
    if ((temperature>temp_variation_lower[loop])
            and (temperature<=temp_variation_upper[loop])) then begin
              write_gain(Tcontrol.device,gain_variation[loop]);
              delay(100);
              write_rate(Tcontrol.device,rate_variation[loop]);
              delay(10);
              write_reset(Tcontrol.device,reset_variation[loop]);
              delay(10);
              write_heater_range(Tcontrol.device,heater_range_variation[loop]);
              delay(100);
            end;
    end; {loop}
  end;


{*****************THERMAL. PAS*********************************************}


{Following procedure looks up thermocouple table to convert dV to dT}

procedure load_thermocouple;
var
 f       : text;
 counter : integer;
begin
  Assign(f,thermocouple_file);
  Reset(f);
  counter:=1;
  repeat
    read(f,aufe[counter],aufe_temperature[counter]);
    counter:=counter+1;
  until Eof(f);
  Close(f);
  end; {load_thermocouple}


function convert_nV(temperature,value:real):real;
var
 counter     : integer;
 volt1,volt2 : real;
 temp1,temp2 : real;
 sensitivity : real;
begin
  counter:=1;
  while (temperature<aufe_temperature[counter]) do
    counter:=counter+1;
  volt1:=aufe[counter];
  volt2:=aufe[counter-1];
  temp1:=aufe_temperature[counter];
  temp2:=aufe_temperature[counter-1];
  sensitivity:=(volt1-volt2)/(temp1-temp2);
  if (sensitivity=0) or (sensitivity<-100) or (sensitivity>100)
  then sensitivity:=1;
  convert_nV:=value/sensitivity;
end; {convert_nV}


function calculate_k(index:integer): real;
begin
if (dT[index]=0) then
                      calculate_k:=0
                 else calculate_k:=current[index]*V[index]*1length*
                                                    1E-3/(area*1E-6*dT[index]);
end; {calculate_k}


procedure open_file;
var
 f    : text;
begin
  outfile:=path+'\'+filename;
  Assign(f,outfile);
  Rewrite(f);
  writeln(f,'Data acquired : ',start_str);
  writeln(f,'Data file     : ',filename);
  writeln(f,'Path          : ',path);
  writeln(f,'No radiation corrections');
  writeln(f,'nV offset uV  : ',manual_offset:6:3);
  writeln(f,'nV scale  uV  : ',nVscale:7:3);
  writeln(f,'Vfsd on meter : ',Vfsd:7:6);
  writeln(f,'Sample size (mm)  ');
  writeln(f,width:6:2,' ',thick:6:2,' ',1length:6:2);
  writeln(f,'Temperature (K)');
  writeln(f,'Thermal Conductivity (W/m/K)');
  close(f);
end;


procedure store_points(index:integer);
var
 f       : text;
 dummy,loop : integer;
begin
  assign(f,outfile);
  append(f);
  dummy:=index-no_points_in_block+1;
  for loop:=dummy to index do
    writeln(f,temp[loop]:9:4,' ',kval[loop]:9:5,' ',current[loop]*1E3:7:5,
                        ' ',V[loop]:9:6,' ',dV[loop]:9:6,' ',dT[loop]:8:4,
                        ' ',setpt[loop]:8:3,' ',power[loop]:3:1,' ',
                        gain[point]:5:2,' ',rate[point]:5:2,' ',
                        rreset[point]:5:2);
 close(f);
end; {store_points}

procedure plot_point(xval,yval:real);
var
 X,Y   : integer;
 dummy : integer;
begin
  dummy:=round((xval-Xmin)*Xstep);
```

```
  Y:=round((yval-Ymin)*Ystep);
  Y:=Y+yborder_bottom;
  if colour then SetColor(Green);
  MoveTo(X-3,Y); LineTo(X+3,Y);
  MoveTo(X,Y-3); LineTo(X,Y+3);
end; {plot_point}


procedure get_setpoint_change_time;
var
 clock : time;
 flagged : boolean;
begin
setpoint_index:=0;
system_clock(clock);
flagged:=false;
{keep going until time now>set(index) but < (index+1)}
repeat
  begin
  if (setpoint_index<no_of_setpoints) then
       begin
         next_setpoint_temperature:=setpoint_temperature[setpoint_index+1];
         next_setpoint_time:=setpoint_time[setpoint_index+1];
       end {if}
     else
       begin
         setpoint_index:=no_of_setpoints;
         flagged:=true;
         no_more_setpoint_changes:=true;
         with next_setpoint_time do begin
           hour:=255;minute:=255;second:=255;day:=255;month:=255;year:=255;
         end;
       end; {else}
  end;
  if ((compare_time(clock,setpoint_time[setpoint_index])) and
       (NOT compare_time(clock,setpoint_time[setpoint_index+1]))) then
                                                        flagged:=true;
  setpoint_index:=setpoint_index+1;
  until flagged;
end; {get_setpoint_change_time}


procedure change_setpoint;
begin
  if (next_setpoint_temperature>0.0) then begin
                last_setpoint:=next_setpoint_temperature;
                write_setpoint(Tcontrol.device,next_setpoint_temperature);
                delay(100);
                if auto_gain then set_gain_etc(next_setpoint_temperature);
              end;
  if (NOT no_more_setpoint_changes) then
                          begin
                            pause_flag:=true;
                            add_time(next_setpoint_time,pause,free_time);
                          end;
  get_setpoint_change_time;
end; {change_setpoint}
```

```
procedure get_next_reading_time;
var
 dumtime : time;
 clock   : time;
begin
  system_clock(clock);
  add_time(clock,interval,dumtime);
  normalise_clock(dumtime);
  if ((compare_time(dumtime,next_setpoint_time)) and
         (NOT no_more_setpoint_changes)) then
                                  add_time(next_setpoint_time,pause,dumtime);
  normalise_clock(dumtime);
  if compare_time(dumtime,end_time) then dumtime:=max_time;
  normalise_clock(dumtime);
  next_reading_time:=dumtime;
  dumtime.minute:=dumtime.minute-nV_minutes_on;
  dumtime.second:=dumtime.second-nV_seconds_on;
  normalise_clock(dumtime);
  nV_ontime:=dumtime;
end; {get_next_reading_time}


procedure display_text(X,Y:integer;instring:str255);
var
 X1,Y1,X2,Y2 : integer;
 high,long   : integer;
begin
  SetTextStyle(DefaultFont,HorizDir,1);
  high:=TextHeight(instring);
  long:=TextWidth(instring);
  X1:=X; Y1:=Y-high;
  X2:=X+long; Y2:=Y;
  if X1<Xleft then X1:=Xleft;
  if X2>Xright then X2:=Xright;
  if Y1<Ytop then Y1:=Ytop;
  if Y2>Ybottom then Y2:=Ybottom;
  SetViewPort(X1,Y1,X2,Y2,ClipOn);
  ClearViewPort;
  SetViewPort(0,0,GetMaxX,GetMaxY,ClipOn);
  OutTextXY(X,Y,instring);
end; {display_text}


procedure set_nV_status(status,write_text:boolean);
var
 dumstr : str255;
 X,Y    : integer;
 begin
  if write_text and colour then SetColor(White);
  if status then begin
                   dumstr:=dal(minicam_address,minicam_on);
                   dumstr:='On ';
                 end
            else begin
                   dumstr:=dal(minicam_address,minicam_off);
                   dumstr:='Off';
                 end;
  if write_text then begin
                       X:=xleft+(xborder_left div 4);
                       Y:=data_line3;
```

```
                    end;
 nV_old_status:=status;
{The minicam is called by function dal, stored in the DEVICES.PAS unit.
 It can be switched 'on' or 'off' directly.  The minicam device number,
 and switch number are passed as STRINGS.
 Note that when the minicam switch is OFF then there is power to the
 nV meter and you shouldn't take a reading}
end; {set_nV_status}


procedure display_data(index:integer);
var
 X,Y           : integer;
 outstr,dumstr : str80;
 dumstr2       : str6;
 dummy         : real;
begin
  X:=xleft+(xborder_left div 4);
  Y:=data_line2;
  {******Diode voltage conversion*************************************************
  dummy:=convert_Volt(temp[index]);
  str(dummy:7:3,dumstr);
  {****************************************************************}
  outstr:=dumstr;
  str(kval[index]:8:5,dumstr);
  outstr:=outstr+' '+dumstr;
  str((current[index]*1E3):7:5,dumstr);
  outstr:=outstr+' '+dumstr;
  str(V[index]:8:6,dumstr);
  outstr:=outstr+' '+dumstr;
  str(dV[index]:7:3,dumstr);
  outstr:=outstr+' '+dumstr;
  str(dT[index]:7:4,dumstr);
  outstr:=outstr+' '+dumstr;
  {******Diode voltage conversion*************************************************
  dummy:=convert_Volt(setpt[index]);
  str(dummy:6:2,dumstr2);
  outstr:=outstr+' '+dumstr2;
  {****************************************************************}
  str(power[index]:3:0,dumstr2);
  outstr:=outstr+'  '+dumstr2;
  str(gain[index]:4:1,dumstr2);
  outstr:=outstr+' '+dumstr2;
  str(rate[index]:4:1,dumstr2);
  outstr:=outstr+' '+dumstr2;
  str(rreset[index]:4:1,dumstr2);
  outstr:=outstr+' '+dumstr2;
  if colour then SetColor(LightRed);
  display_text(X,Y,outstr);
  if colour then SetColor(White);
end;


 procedure display_time;
var
 dumstr,outstr  : str255;
 dumstr2        : str255;
 X,Y            : integer;
 clock          : time;
 timestr,next_set_str,next_reading_str : str20;
begin
  system_clock(clock);
  if compare_time(clock,old_time) then
    begin
      old_time:=clock;
      X:=xleft+xborder_left div 4;
      Y:=data_line1;
      str(next_setpoint_temperature:7:3,dumstr2);
      compress_clock(clock,timestr);
      compress_clock(next_reading_time,next_reading_str);
      compress_clock(next_setpoint_time,next_set_str);
      if no_more_setpoint_changes then begin
                                         next_set_str:='No more changes';
                                         dumstr2:='No more changes';
                                       end;
      outstr:=timestr+'   '+next_reading_str+'    '+
                                           next_set_str+'   '+dumstr2;
      if colour then SetColor(LightRed);
      display_text(X,Y,outstr);
      if colour then SetColor(White);
     end; {if}
 end; {display_time}


procedure reading;
var
  dummy,loop              : integer;
  tem,setpnt,curr,volt,nV : real;
  dumreal                 : real;
begin
  tem:=0.0;
  setpnt:=0.0;
  curr:=0.0;
  volt:=0.0;
  nV:=0.0;
  for loop:=1 to average_readings do
  begin
    tem:=tem+read_temp(Tcontrol.device);
    curr:=curr+read_current(Imeter.device);
    volt:=volt+read_volt(Vmeter.device);
    nV:=nV+read_nV(nVmeter.device);
  end; {loop}
  power[point]:=read_heater_power(Tcontrol.device);
  gain[point]:=read_gain(Tcontrol.device);
  rate[point]:=read_rate(Tcontrol.device);
  rreset[point]:=read_reset(Tcontrol.device);
  setpt[point]:=read_setpoint(Tcontrol.device);
  temp[point]:=tem/average_readings;
  current[point]:=curr/average_readings;
  V[point]:=volt/average_readings;
  nV:=nV/average_readings;
  dV[point]:=(nV*nVscale/Vfsd)+manual_offset;
  {******Diode voltage conversion*************************************************
  dumreal:=convert_Volt(temp[point]);
  dT[point]:=convert_nV(dumreal,dV[point]);
{****************************************************************}
  kval[point]:=calculate_k(point);
end; {reading}
```

```
procedure take_reading(plot_data:boolean);
var
 saveblock,dummy : integer;
 dumreal : real;
begin
reading;
if plot_data then begin
                    display_data(point);
  {******Diode voltage conversion*************************************************************
                    dumreal:=convert_Volt(temp[point]);
                    if plot_data then plot_point(dumreal,Abs(kval[point]));
{*****************************************************************}
                  end;
 saveblock:=(point mod no_points_in_block);
  if (saveblock=0) then store_points(point);
  reading_taken:=true;
  nVpower:=false;
  set_nV_status(nVpower,plot_data);
  point:=point+1;
  if (point>=max_points) then
                         begin
                           point:=point-1;
                           dummy:=no_points_in_block;
                           no_points_in_block:=point mod no_points_in_block;
                           store_points(point);
                           no_points_in_block:=dummy;
                           point:=1;
                         end;
end; {take_reading}



procedure draw_scales;
var
 X_axis_step,Y_axis_step : real;
 dumstr                  : str8;
 X,Y                     : real;
 X1,Y1                   : integer;
begin
  if colour then TextColor(White);
  X_axis_step:=auto_step(Xmax,Xmin);
  X_axis_step:=X_axis_step*2;
  Y_axis_step:=auto_step(Ymax,Ymin);
  rectangle(0,Max_Y_value,Max_X_value,0);
  SetTextStyle(SmallFont,Horizdir,4);
  SetTextJustify(CenterText,CenterText);
  X:=Xmin;
  while (X<=Xmax) do
  begin
    X1:=round((X-Xmin)*Xstep)+xborder_left;
    if colour then SetColor(White);
    line(X1,yborder_top,X1,yborder_bottom);
    str(X:6:2,dumstr);
    if colour then SetColor(Yellow);
    OutTextXY(X1,yborder_bottom-(yborder_bottom_thickness div 3),dumstr);
    X:=X+X_axis_step;
  end; {while}
  Y:=Ymin;
  while (Y<=Ymax) do
    Y1:=round((Y-Ymin)*Ystep)+yborder_bottom;
    if colour then SetColor(White);
    line(xborder_right,Y1,xborder_left,Y1);
    str(Y:6:2,dumstr);
    if colour then SetColor(Yellow);
    OutTextXY(xleft+((xborder_left_thickness*2)div 3),Y1,dumstr);
    Y:=Y+Y_axis_step;
  end; {while}
  if colour then SetColor(Cyan);
  OutTextXY(xcentre,
    yborder_bottom-((yborder_bottom_thickness*2) div 3),'Temperature (K)');
  SetTextStyle(DefaultFont,VertDir,1);
  OutTextXY((xleft+xborder_left_thickness div 3),
                                      ycentre,'Thermal Conductivity W/m/K') ;
  if colour then SetColor(White);
  SetLineStyle(SolidLn,0,NormWidth);
  SetTextStyle(DefaultFont,HorizDir,1);
  SetTextJustify(LeftText,BottomText);
  X1:=xleft+(xborder_left div 4);
  Y1:=title_line1;
  SetViewPort(xleft,ytop,xright,ybottom,ClipOn);
  if colour then SetColor(LightGray);
  display_text(X1,Y1,'      Time            Next data at        '+
                           'Set change at      Set to K');
  X1:=xleft+(xborder_left div 4);
  Y1:=title_line2;
  display_text(X1,Y1,' Temp.    K     I(mA)      V     '+
                          '  dV(uV)  dT(K)  Set(K)  P  Gain Rate Reset');
  if colour then SetColor(White);
end; {draw_scales}



procedure graph_mode;
var
  GraphDriver,ErrorCode : integer;
begin
  GraphDriver:=Detect;
  InitGraph(GraphDriver,GraphMode,graph_path);
  ErrorCode:=GraphResult;
  if ErrorCode<>grOk then
    begin
      RestoreCrtMode;
      header;
      writeln('Graphics Error: ',GraphErrorMsg(ErrorCode));
      writeln('Program aborted...');
      Halt(1);
    end;
  colour:=(GetPaletteSize=16);
  graphset:=true;
end; {graph_mode}


procedure initialise_graph(flag:integer);
var
  loop  : integer;
  dummy : real;
begin
  if NOT graphset then graph_mode
                  else SetGraphMode(GraphMode);
```

```
  max_X_value:=GetMaxX;
  max_Y_value:=GetMaxY;
  if (max_Y_value<300) then top_extra_lines:=0
                       else top_extra_lines:=1;

  {Border thicknesses are calculated right edge minus left edge,
   and top edge minus bottom edge. This way the fact that 0,0 is
   top left does not matter}

  total_lines:=top_no_of_lines+top_extra_lines;
  xleft:=0;
  xright:=max_X_value;
  ytop:=0;
  ybottom:=max_Y_value;
  yborder_top_thickness:=-1*(pixel_line_height*total_lines);
  xborder_left_thickness:=round((xright-xleft)*1.5/10);
  xborder_right_thickness:=round((xright-xleft)/10);
  xborder_left:=xleft+xborder_left_thickness;
  xborder_right:=xright-xborder_left_thickness;
  yborder_bottom_thickness:=round((ytop-ybottom)/10);
  yborder_top:=ytop-yborder_top_thickness;
  yborder_bottom:=ybottom+yborder_bottom_thickness;
  yheight:=yborder_top-yborder_bottom;
  Ystep:=yheight/(Ymax-Ymin);
  xlength:=xborder_right-xborder_left;
  Xstep:=xlength/(Xmax-Xmin);
  ycentre:=yborder_bottom+(yheight div 2);
  xcentre:=xborder_left+(xlength div 2);
  title_line1:=round(yborder_top+(yborder_top_thickness*(total_lines-1))
                                                        /total_lines);
  data_line1:=round(yborder_top+(yborder_top_thickness*(total_lines-2))
                                                        /total_lines);
  title_line2:=round(yborder_top+(yborder_top_thickness*
                          (total_lines-3-top_extra_lines))/total_lines);
  data_line2:=round(yborder_top+(yborder_top_thickness*
                          (total_lines-4-top_extra_lines))/total_lines);
  title_line3:=round(yborder_top+(yborder_top_thickness*
                          (total_lines-5-top_extra_lines))/total_lines);
  data_line3:=round(yborder_top+(yborder_top_thickness*
                          (total_lines-6-top_extra_lines))/total_lines);
  draw_scales;
  if (flag=1) then
            begin
              for loop:=1 to point-1 do
  {******Diode voltage conversion************************************************
               begin
               dummy:=convert_Volt(temp[loop]);
               plot_point(dummy,Abs(kval[loop]));
               end;
{*****************************************************************************}
               display_data(point-1);
            end;

end; {initialise_graph}


{*****************THERMAL. PAS*********************************************}
```

```
procedure alter_controller_manually;
var
  dummy,loop              : integer;
  setpt,newsetpt          : real;
  gain,rate,rreset        : real;
  newgain,newrate,newrreset : real;
begin
  header;
  gain:=read_gain(Tcontrol.device);
  rate:=read_rate(Tcontrol.device);
  rreset:=read_reset(Tcontrol.device);
  setpt:=read_setpoint(Tcontrol.device);
  {******Diode voltage conversion************************************************
  setpt:=convert_Volt(setpt);
  writeln('Set point  = ',setpt:6:2);
  {*****************************************************************************}
  write('Enter the new setpoint: 0 to 325K ? ');
  newsetpt:=input_real(setpt,0,325,true);
  header;writeln;
  writeln('Gain       = ',gain:6:2);
   write('Enter the new gain from ',min_gain:5:2,' to ',max_gain:5:2,' ? ');
  newgain:=input_real(gain,min_gain,max_gain,true);
  header;writeln;
  writeln('Rate       = ',rate:6:2);
   write('Enter the new rate from ',min_rate:5:2,' to ',max_rate:5:2,' ? ');
  newrate:=input_real(rate,min_rate,max_rate,true);
  header;writeln;
  writeln('Reset      = ',rreset:6:2);
write('Enter the new reset from ',min_reset:5:2,' to ',max_reset:5:2,' ? ');
  newrreset:=input_real(rreset,min_reset,max_reset,false);
  if (setpt<>newsetpt) then write_setpoint(Tcontrol.device,newsetpt);
  delay(100);
  if (rate<>newrate) then write_rate(Tcontrol.device,newrate);
  delay(100);
  if (gain<>newgain) then write_gain(Tcontrol.device,newgain);
  delay(10);
  if (rreset<>newrreset) then write_reset(Tcontrol.device,rreset);
  delay(10);
end; {reading}


procedure break_into_program;
var
 ans : integer;
 look : char;
 maxoptions : integer;
begin
 RestoreCRTMode;
 maxoptions:=9;
 ans:=9;
 look:=ReadKey;
 while ((look='i') or (look='I')) or (ans<>maxoptions) do
 begin
   look:='x';
   header;
   writeln;
   writeln('1... alter devices used (change keithley for fluke, etc)');
   writeln('2... alter variable parameters (nV offsets etc)');
   writeln('3... alter setpoints');
   writeln('4... alter graphics data');
```

```
    writeln('5... alter the gain data');
    writeln('6... take a data point');
    writeln('7... alter the controller data');
    writeln('8... end the program');
    writeln('9... go back to the graph and start collecting data again');writeln;
    write('Choose the option you require: 1 to 9 ? ');
    ans:=input_integer(maxoptions,1,maxoptions,false);
    if (ans=1) then change_device_parameters;
    if (ans=2) then change_variable_parameters(false);
    if (ans=3) then begin
                      alter_setpoints;
                      get_setpoint_change_time;
                    end;
    if (ans=4) then change_graphics_variables;
    if (ans=5) then alter_gain_variations;
    if (ans=6) then take_reading(false);
    if (ans=7) then alter_controller_manually;
    if (ans=8) then begin
                      header;writeln;writeln('End program now ? ');
                      if (yes_or_no(false)) then begin
                                                   Window(1,1,80,25);
                                                   ClrScr;
                                                   halt;
                                                 end;
                    end;
  end; {while}
  initialise_graph(1);
 end; {break_into_program}


{*****************THERMAL. PAS*****************************************}
{* main program                                                        *}
{*********************************************************************}

begin
   system_clock(start_time);
   graph_mode;
   RestoreCRTMode;
   title;
   device_defaults;
   change_device_parameters;
   variable_defaults;
   change_variable_parameters(true);
   gain_variation_defaults;
   alter_gain_variations;
   enter_setpoints;
   alter_setpoints;
   enter_intervals;
   initialise_graphics_variables;
   initialise_graph(0);
   initialise_devices;
   nVpower:=false;
   set_nV_status(nVpower,true);
   load_thermocouple;
   open_file;
   {******Diode voltage conversion*************************************************
   dummy:=10.;
   write_setpoint(Tcontrol.device,dummy);
   last_setpoint:=1;
   {*********************************************************************}
```

```
  if auto_gain then set_gain_etc(last_setpoint);
  get_next_reading_time;
  get_setpoint_change_time;
  repeat
    system_clock(now_time);


{*****************THERMAL. PAS*****************************************}


    if ((compare_time(now_time,next_reading_time))
          and (NOT pause_flag)) then begin
                                       take_reading(true);
                                       get_next_reading_time;
                                     end;

    if ((compare_time(now_time,next_setpoint_time))
          and (NOT no_more_setpoint_changes)) then change_setpoint;

    if (compare_time(now_time,free_time)) then pause_flag:=false;

    if ((compare_time(now_time,nV_ontime))
          and (NOT nVpower)) then
                               begin
                                 nVpower:=true;
                                 set_nV_status(nVpower,true);
                               end;
    if (KeyPressed) then break_into_program;
   display_time;
  until compare_time(now_time,end_time);
  RestoreCRTMode;
  Window(1,1,80,25);
  ClrScr;
end.
```

```
(*                                                                          *)
(*   UNIT mrdglob - global definitions for programs                         *)
(*   3/8/90   M.R.Delap                                                     *)
(*                                                                          *)
(**************************************************************************)

unit mrdglob;

{--------------------------------------------------------------------------}

interface

 type
  time =
  record
      year   : integer;
      month  : integer;
      day    : integer;
      hour   : integer;
      minute : integer;
      second : integer;
    end;

    namfil = string[12];
    str2   = string[2];
    str3   = string[3];
    str4   = string[4];
    str5   = string[5];
    str6   = string[6];
    str8   = string[8];
    str10  = string[10];
    str12  = string[12];
    str14  = string[14];
    str16  = string[16];
    str20  = string[20];
    str40  = string[40];
    str80  = string[80];
    str255 = string[255];
    double   = real;
    string10 = string[10];

  var
     GraphMode : integer;

{--------------------------------------------------------------------------}

implementation
{--------------------------------------------------------------------------}
end.
```

```
{*************************************************************************}
{*  Unit MRDUTILS.PAS                                                    *}
{*  obtains data for thermal conductivity experiments                    *}
{*  M.R.Delap 1 November 1989                                            *}
{*************************************************************************}

unit MRDUTILS;

interface

uses mrdglob,crt,dos,graph;

{*************************************************************************}
function zero(instring:str12):str12;
procedure normalise_clock(var intime:time);
procedure compress_clock(intime:time;var result:str20);
function input_string: str255;
procedure input_time(default:time;var result:time;minus_one:boolean);
function input_integer(default,min,max:integer;minus_one:boolean):integer;
function input_real(default,min,max:real;minus_one:boolean):real;
function log(value:real):real;
function power_of(a,b:real):real; {calculates a^b}
procedure find_char(st:str255;ch:char;var pointer:integer);
procedure space;
procedure cls;
procedure dashed_line;
function yes_or_no (default : boolean) : boolean;
procedure expand_clock(intime:str20;var result:time);
function compare(num1,num2:integer):integer;
function compare_time(time1,time2:time): boolean;
procedure system_clock(var result: time);
procedure add_time(intime1,intime2:time;var result:time);
function file_exists(filename,path:str255):boolean;
function get_pathname(oldpath:str255):str255;
function file_version(name:str12;path:str255;extension:str4): str12;
function get_filename(oldfile:str12;var path:str255;extension:str4;new_file:boolean)
procedure hold(instring:str255);
procedure display_directory(path:str255);
{*************************************************************************}

implementation

function zero(instring:str12):str12;
var
 dummy : integer;
begin
  dummy:=Length(instring);
if dummy=0 then zero:='00';
if dummy=1 then zero:='0'+instring;
if dummy>1 then zero:=instring;
end; {zero}


procedure normalise_clock(var intime:time);
var
 flag,ok                    : boolean;
begin
  with intime do begin
  if (second>59) then begin
                      second:=second mod 60;
                     end;
  if (minute>59) then begin
                      hour:=hour+(minute div 60);
                      minute:=minute mod 60;
                     end;
  if (hour>23) then begin
                      day:=day+(hour div 24);
                      hour:=hour mod 24;
                     end;
  repeat begin
    ok:=true;
    flag:=false;
    if ((month=4) or (month=6) or (month=9) or (month=11)) then
     begin
       flag:=true;
       if (day>30) then begin
                          month:=month+1;
                          day:=day-30;
                          ok:=false;
                         end;
    end;
    if (month=2) then begin
    flag:=true;
       if (day>28) then begin
                          month:=month+1;
                          day:=day-28;
                          ok:=false
                         end;
    end;
    if (flag=false) then
    begin
      if (day>31) then begin
                          month:=month+1;
                          day:=day-31;
                          ok:=false;
                         end;
    end; {if}
  if (month>12) then begin
                      while month>12 do
                      begin
                        year:=year+1;
                        month:=month-12;
                      end;
                     end;
    end; {repeat}
    until ok;
 end;
end;


procedure compress_clock(intime:time;var result:str20);
var
 outstr,dumstr : str20;
 dummy         : integer;
begin
{  normalise_clock(intime);}
  with intime do begin
    result:='';
```

```
    str(minute,dumstr);  result:=result+zero(dumstr)+':';
    str(second,dumstr);  result:=result+zero(dumstr)+'-';
    str(day,dumstr);  result:=result+zero(dumstr)+'/';
    str(month,dumstr);result:=result+zero(dumstr)+'/';
    str(year,dumstr);
    dummy:=Length(dumstr);
    if dummy=1 then dumstr:='0'+dumstr
               else dumstr:=copy(dumstr,(dummy-1),2);
    result:=result+dumstr;
  end;
end;


procedure expand_clock(intime:str20;var result:time);
{input HH:MM:SS-DD/MM/YY}
var
 dumstr                : str20;
 dummy,start,stop,len  : integer;
 flag,ok               : boolean;
begin
  with result do begin
    stop:=0; len:=0;
    start:=1; find_char(intime,':',stop);len:=stop-start;
    dumstr:=copy(intime,start,len);
    val(dumstr,hour,dummy);
    start:=stop+1; find_char(intime,':',stop);len:=stop-start;
    dumstr:=copy(intime,start,len);
    val(dumstr,minute,dummy);
    start:=stop+1; find_char(intime,'-',stop);len:=stop-start;
    dumstr:=copy(intime,start,len);
    val(dumstr,second,dummy);
    start:=stop+1; find_char(intime,'/',stop);len:=stop-start;
    dumstr:=copy(intime,start,len);
    val(dumstr,day,dummy);
    start:=stop+1; find_char(intime,'/',stop);len:=stop-start;
    dumstr:=copy(intime,start,len);
    val(dumstr,month,dummy);
    start:=stop+1;
    dumstr:=copy(intime,start,2);
    val(dumstr,year,dummy);
  end;
end;


function input_string: str255;
var
 counter,dummy,colon : integer;
 dumstr              : str255;
 keystr              : char;
 accept              : boolean;
begin
  dumstr:='';
  counter:=0;
  repeat
    keystr:=ReadKey;
    accept:=true;
    if ((keystr=chr(8)) and (counter>=1)) then
                          begin
                            dumstr:=copy(dumstr,1,counter-1);
                            counter:=counter-1;
                            accept:=false;
                            write(keystr);
                          end;
    if accept then begin
                     dumstr:=dumstr+keystr;
                     counter:=counter+1;
                     write(keystr);
                   end;
  until ((keystr=chr(13)) or (counter>=253));
  if (counter=254) then dumstr:=dumstr+chr(13);
  writeln;
  input_string:=dumstr;
end;


procedure input_time(default:time;var result:time;minus_one:boolean);

{NB the value returned is the time only - the date returned is the
  same as that for default.  Also, if -1 is true on entry then if
  -1 is entered then result.minute=255 on exit}

var
 loop,dummy,colon,len   : integer;
 dumstr,dumstr2         : str255;
 letter                 : string[1];
 accept,valid,lastcolon : boolean;
 dumtime                : time;
 dumtime2,dumdate       : str20;
 dumstr3                : str20;
begin
  valid:=false;
  compress_clock(default,dumtime2);
  dumstr3:=copy(dumtime2,1,8);
  while (NOT valid) do begin
    writeln('The default time is ',dumstr3);
    dumstr:=input_string;
    if (dumstr=chr(13)) then valid:=true
                        else valid:=false;
    lastcolon:=true;
    colon:=0;
    dumstr2:='';
    len:=Length(dumstr);
    for loop:=1 to len do begin
      accept:=false;
      letter:=copy(dumstr,loop,1);
      if ((letter='0') or (letter='1') or (letter='2') or (letter='3') or
         (letter='4') or (letter='5') or (letter='6') or (letter='7') or
         (letter='8') or (letter='9')) then accept:=true;
      if ((letter='-') and lastcolon) then accept:=true;
      if accept then lastcolon:=false;
      if ((letter=':') and (NOT lastcolon)) then begin
                                                   colon:=colon+1;
                                                   accept:=true;
                                                   lastcolon:=true;
                                                 end;
      if accept then dumstr2:=dumstr2+letter;
    end; {loop}
    if (((minus_one and ((dumstr<>'-1'+chr(13)) and (colon<>2))) or
        ((NOT minus_one) and (colon<>2))) and
```

```
                    writeln;
                    writeln('That input time is invalid - try again');
                    valid:=false; writeln;
                  end
                  else
                  valid:=true;
  end; {while}
 compress_clock(default,dumdate);
 dumdate:=copy(dumdate,9,10);
 dumstr2:=dumstr2+dumdate;
 if (dumstr=('-1'+chr(13))) then
                          begin
                            result:=default;
                            result.hour:=255;
                          end
             else begin
             if (dumstr=chr(13)) then begin
                                        result:=default;
                                        normalise_clock(result);
                                      end
                                 else begin
                                        expand_clock(dumstr2,result);
                                        normalise_clock(result);
                                      end;
             end;
end;


function input_integer(default,min,max:integer;minus_one:boolean):integer;
var
 ans,error : integer;
 dumstr    : str255;
 x,y       : byte;
begin
  x:=WhereX;
  y:=WhereY;
  repeat
    GotoXY(x,y);ClrEol;
    GotoXY(x,y);write(default:6);
    GotoXY(x,y);
    readln(dumstr);
    val(dumstr,ans,error);
    if dumstr='' then ans:=default;
  until ((ans>=min) and (ans<=max)) or ((ans=-1) and minus_one);
  input_integer:=ans;
end;


function input_real(default,min,max:real;minus_one:boolean):real;
var
 ans    : real;
 error  : integer;
 dumstr : str255;
 x,y    : byte;
begin
  x:=WhereX;
  y:=WhereY;
  repeat
    GotoXY(x,y);ClrEol;
    GotoXY(x,y);
    readln(dumstr);
    val(dumstr,ans,error);
    if dumstr='' then ans:=default;
  until ((ans>=min) and (ans<=max)) or ((ans=-1) and minus_one);
  input_real:=ans;
end;


function log(value:real):real;
begin
  if value<=0 then log:=0
              else log:=Ln(value)/Ln(10);
end; {log}


function power_of(a,b:real):real; {calculates a^b}
begin
  power_of:=exp(b*Ln(a));
end; {power_of}


procedure find_char(st:str255;ch:char;var pointer:integer);
var
 len : integer;
begin
len:=Length(st);
 repeat
   pointer:=pointer+1;
 until (st[pointer]=ch) or (pointer>=len);
 if ((pointer>=len) and (st[len]<>ch)) then pointer:=0;
 end;


procedure space;
var dummy : char;
begin
  writeln ('Press SPACE to continue ....');
  repeat
    dummy:=ReadKey;
  until (dummy=' ');
end; {space}

procedure cls;
begin
  ClrScr;
end; {cls}

procedure dashed_line;
begin
writeln('------------------------------------------------------------')
end;


function yes_or_no (default:boolean): boolean;
var
 ans,defn : char;
 x,y      : byte;
begin
```

```
  y:=WhereY;
    if default then defn:='Y'
               else defn:='N';
    GotoXY(x,y);ClrEol;
    GotoXY(x,y);write(defn);
    GotoXY(x,y);
  repeat
    ans:=ReadKey;
  until (ans='Y') or (ans='y') or (ans='N') or (ans='n') or (ans=chr(13));
  if (ans='Y') or (ans='y') then  yes_or_no:=true;
  if (ans=chr(13))          then  yes_or_no:=default;
  if (ans='N') or (ans='n') then  yes_or_no:=false;
end; {yes_or_no}

{***************UNIT MRDUTILS************************************************}

function Int_to_str(i:integer):string;
var
 s : str12;
begin
 str(i,s);
 Int_to_str:=s;
end;


 function compare(num1,num2:integer):integer;
 begin
   if (num1>num2) then compare:=0;
   if (num1=num2) then compare:=1;
   if (num1<num2) then compare:=2;
 end;


 function compare_time(time1,time2:time): boolean; {  sees if time1 is  }
 var                                               {  later than time2  }
  ans                              : boolean;
  dummy                            : integer;
 begin
  ans:=false;
   dummy:=compare(time1.year,time2.year);
   if (dummy=0) then ans:=true;
   if (dummy=1) then begin
      dummy:=compare(time1.month,time2.month);
      if (dummy=0) then ans:=true;
      if (dummy=1) then begin
         dummy:=compare(time1.day,time2.day);
         if (dummy=0) then ans:=true;
         if (dummy=1) then begin
             dummy:=compare(time1.hour,time2.hour);
             if (dummy=0) then ans:=true;
             if (dummy=1) then begin
                dummy:=compare(time1.minute,time2.minute);
                if (dummy=0) then ans:=true;
                if (dummy=1) then begin
                   dummy:=compare(time1.second,time2.second);
                   if (dummy=0) then ans:=true;
                end;
             end;
         end;
```

```
   end;
  if (time1.hour=time2.hour) and (time1.minute=time2.minute)
              and (time1.second=time2.second) and (time1.day=time2.day)
              and (time1.month=time2.month) and (time1.year=time2.year)
              then ans:=false;
  compare_time:=ans;
end;



procedure system_clock(var result:time);
var
 hr,min,sec,sec100  : word;
 yr,mnth,dy,dayofwk : word;
begin
  GetTime(hr,min,sec,sec100);
  GetDate(yr,mnth,dy,dayofwk);
  result.hour:=hr;
  result.minute:=min;
  result.second:=sec;
  result.day:=dy;
  result.month:=mnth;
  result.year:=yr-1900;
  normalise_clock(result);
end;


procedure add_time(intime1,intime2:time;var result:time);
var
 dumstr                           : str255;
begin
  result.hour:=intime1.hour+intime2.hour;
  result.minute:=intime1.minute+intime2.minute;
  result.second:=intime1.second+intime2.second;
  result.day:=intime1.day+intime2.day;
  result.month:=intime1.month+intime2.month;
  result.year:=intime1.year+intime2.year;
end;


function file_exists(filename,path:str255):boolean;
var
 fil : str255;
 f   : file;
begin
  fil:=path+'\'+filename;
  {$I-}
  Assign(f,fil);
  Reset(f);
  Close(f);
  {$I+}
  File_exists:=((IOResult=0) and (Filename<>''));
end; {file_exists}


function get_pathname(oldpath:str255):str255;
var
 answer,dumstr              : str255;
 letter                     : string[1];
```

```
 startpt,len,loop,counter    : integer;
begin
  ok:=false;
  while (NOT ok) do begin
    writeln('Enter the name of the directory path (including drive name)');
    writeln('The old path was ',oldpath);
    writeln('Press RETURN to use this path');
    answer:=input_string;
    if (answer=chr(13))  then ok:=true;
    dumstr:='';
    startpt:=0;
    started:=false;
    accept:=true;
    counter:=0;
    had_colon:=false;
    len:=Length(answer);
    ok:=true;
    for loop:=1 to len do begin
      accept:=true;
      letter:=copy(answer,loop,1);
      if ((letter=' ') or (letter=chr(13))) then accept:=false;
      if (letter=':') and (counter<>startpt+1) then accept:=false;
      if ((letter=':') and accept and (NOT had_colon))
                                              then had_colon:=true;
      if (accept and (NOT started)) then begin
                                    startpt:=counter;
                                    started:=true;
                                  end;
      if accept then begin
                 dumstr:=dumstr+letter;
                 counter:=counter+1;
                 end;
     end; {loop}
     if ((NOT had_colon) and (answer<>chr(13))) then begin
                           ok:=false;
                           writeln('You must specify a drive!');
                           had_colon:=true;
                         end;
   end;{while}
   if (answer=chr(13)) then get_pathname:=oldpath
                       else get_pathname:=dumstr;
end; {get_pathname}


function file_version(name:str12;path:str255;extension:str4): str12;
var                                  {Takes the first 6 letters of filename}
 dumstr     : str2;                  {Adds the version number and the }
 dumstr2    : str6;                  {extension}
 filenumber : integer;
 failed     : boolean;
 fname      : str12;
 dash       : integer;
 len        : integer;
 dot        : integer;
begin
  dot:=0;
  find_char(name,'.',dot);
  if (dot<>0) then name:=copy(name,1,dot-1)
              else begin
                   len:=Length(name);
                   if (len>8) then name:=copy(name,1,8);
                  end;
  filenumber:=0;
  repeat
    dash:=0;
    find_char(name,'-',dash);
    dumstr:='';
    dumstr2:='       ';
    filenumber:=filenumber+1;
    str(filenumber,dumstr);
    if (dash<>0) then dumstr2:=copy(name,1,dash-1)
                 else begin
                        len:=Length(name);
                        if (len<=6) then dumstr2:=name
                                    else dumstr2:=copy(name,1,6)
                      end;
    fname:=dumstr2+'-'+dumstr+extension;
    failed:=file_exists(fname,path);
    if (filenumber=99) then failed:=false;
  until (NOT failed);
  file_version:=fname;
end; {file_version}


function get_filename(oldfile:str12;var path:str255;extension:str4;new_file:boolean)
var
 answer,dumstr   : str255;
 len             : integer;
 dumstr2,dumstr3 : str12;
 dumchar         : string[1];
 loop            : integer;
 ok,ok2          : boolean;
 dot             : integer;
begin
  repeat
  repeat;
    ok:=true;
    ClrScr;
    display_directory(path);
    writeln('Enter the name of the file you wish to use');
    writeln('Enter RETURN to use the old name, DIR to change directory');
    writeln('Old file was ',oldfile);
    writeln('The ',extension,' extension is automatically added');
    answer:=input_string;
    if (answer<>chr(13)) then answer:=copy(answer,1,Length(answer)-1);
    if (answer=chr(13)) then answer:=oldfile;
    if (answer='DIR') or (answer='DIr') or (answer='Dir') or (answer='dir')
          or (answer='diR') or (answer='dIR') or (answer='DiR') or
               (answer='dIr') then begin
                                     ok:=false;
                                     path:=get_pathname(path);
                                   end;
  until ok;
  len:=Length(answer);
  if (len>12) then answer:=copy(answer,1,8);
  dot:=0;
  find_char(answer,'.',dot);
  dumstr:='';
  dumstr2:='       ';
```

```
            else begin
                   len:=Length(answer);
                   if (len>8) then answer:=copy(answer,1,8);
                  end;
  dumstr2:=answer+extension;
  ok2:=true;
  if new_file then get_filename:=file_version(dumstr2,path,extension)
              else begin
               if NOT file_exists(dumstr2,path) then ok2:=false
                                                else get_filename:=dumstr2
                   end
  until ok2;
end;

{***************UNIT MRDUTILS*************************************************}

  procedure hold(instring:str255);
  begin
  RestoreCrtMode;
  writeln;
  writeln(instring);
  writeln;
  writeln('If using THERMAL CONDUCTIVITY program, change the'+
            ' meter to the correct scale.');
  writeln('Then press SPACE');
  writeln('The graphics display will then need to be re-drawn');
  writeln('To do this, press i and then RETURN');
  space;
  SetGraphMode(GraphMode);
  end;


  procedure display_directory(path:str255);  {Stolen from Roger!! 11/01/90}
  var
   test_key, fnkey : char;
   x_block, y_block, rec_num, last_rec, last_row, last_blk_colmn : byte;
   prev_rec        : byte;
   x_string        : string[2];
   i,j             : integer;
   dir_search      : SearchRec;
   dir_rec         : array[1..225] of SearchRec;
   ok              : boolean;
   dir             : DirStr;
   filename        : NameStr;
   ext             : ExtStr;
   dumstr          : str255;
   filename_length : integer;
   screen_width    : integer;
   no_of_files_per_line : integer;
  begin
    filename_length:=13;
    screen_width:=80;
    no_of_files_per_line:=(screen_width div filename_length);
    i := 1;
    dumstr:=path+'\*.*';
    FindFirst(dumstr, Archive + Directory, dir_search);
    dir_rec[i] := dir_search;
    while DosError = 0 do
    begin
      FindNext(dir_search);
      dir_rec[i] := dir_search;
    end;
    last_rec := i-1;
    last_row := last_rec div no_of_files_per_line + 1;
    last_blk_colmn := last_rec - (last_row-1) * no_of_files_per_line;
                  {last block column of last row}
    i := 1;
    while i <> last_row do
    begin
      for j := 0 to no_of_files_per_line-1 do begin
                GotoXY(filename_length*j+2,i);
                FSplit(dir_rec[(i-1)*no_of_files_per_line+j+1].name,dir,filename,ext)
                write(filename,ext);
               end;
      i := i + 1;
    end;
   for j := 0 to last_blk_colmn-1 do
      begin
        i := last_row;
        GotoXY(filename_length*j+2,i);
        FSplit(dir_rec[(i-1)*no_of_files_per_line+j+1].name,dir,filename,ext);
        write(filename,ext);
      end;
   writeln;
   end;


end.
```

```
{*************************************************************************}
{*        Sets up defaults for thermal conductivity programs             *}
{*        3/8/90   M.R.Delap                                             *}
{*************************************************************************}

unit defaults;

interface

 uses mrdglob;

 const
  max_setpoints         = 100;    {Sets max number of setpoint changes}
  max_devices           = 10;     {Sets max number of IEEE devices     }
                                  {Note 10 is max allowed by IEEE routines}
                                  {Nothing to do with me! }
  max_points            = 100;    {Sets max number of points stored in RAM}
  max_gain_variations   = 20;     {Sets number of gain changes       }
  min_gain              = 0.1;
  max_gain              = 99.;     {Max and minimum permitted values}
  min_rate              = 0.0;
  max_rate              = 99.;
  min_reset             = 0.0;
  max_reset             = 99.;
  min_heater_power      = 0;       {Set max and min heater ranges}
  max_heater_power      = 5;
  nV_minutes_on         = 1;       {Set how long nV meter is switched on}
  nV_seconds_on         = 0;       {before a reading is taken           }
  ieee_no_of_channels   = 31;
  no_of_lines           = 17;
  more_than             = 0;        {Codes returned from compare routines}
  equal                 = 1;
  less_than             = 2;
  pc_ieee               = 1;                {IEEE address of PC}
  default_interval      = '00:04:59';       {Default time between readings}
  default_pause         = '00:24:59';  {Default pause after set point change}
  default_path          = 'C:\MARTIN\THERMAL';
  default_filename      = 'DATA';
  graph_path            = 'C:\TP5\GRAPH';
  default_end_time      = '96:00:01';
  null_date             = '-00/00/00';
  minicam_on            = '128';
  minicam_off           = '0';
  top_no_of_lines       = 7;
  pixel_line_height     = 10;
  default_setpoint_hour = 0;
  default_setpoint_min  = 1;
  default_setpoint_second = 0;
  default_setpoint_interval_time = 1;
  thermocouple_default_file    = 'C:\MARTIN\AUFE.CAL';
  dr91cinitstr1  = 'Z0M1T0'; {EOI status, remote mode, terminator = LF CR}
{  dr91cinitstr2  = 'F0KF1KF3A3';} {units = Kelvin, resolution = xx.xxx}
 dr91cinitstr2 = 'F1SF3A4';    {units = Volts  resolution = x.xxxx}
  k197initstr    = 'G0K0D0T5';  {Volts, 0-2V }
  k175initstr    = 'G0K0D0T4';  {Amps, 0-20mA}
  solarinitstr   = 'U4R0N1';    {Volts, 0-2V }

 type
```

```
 record
  device : integer;
  name   : str8;
  chan   : integer;
 end;

var
  bdeal,fluke8860,k197      : integer;
  k175,solar,weston,dr91c   : integer;
  minicam                   : integer;
  GraphMode                 : integer;
  area                      : real;
  aufe                      : array [1..150] of real;
  aufe_temperature          : array [1..150] of real;
  auto_gain                 : boolean;
  auto_scale                : boolean;
  average_readings          : integer;
  colour                    : boolean;
  current                   : array [1..max_points] of real;
  datafile                  : text;
  data_line1,data_line2     : integer;
  data_line3                : integer;
  data_time                 : array [1..max_points] of str12;
  device_name               : array [1..ieee_no_of_channels] of str20;
  device_default_chan       : array [1..ieee_no_of_channels] of integer;
  dV,dT     .               : array [1..max_points] of real;
  end_time                  : time;
  filename                  : str12;
  free_time                 : time;
  gain_variation            : array [1..max_gain_variations] of real;
  graph_X_limit             : integer;
  graph_Y_limit             : integer;
  graphset                  : boolean;
  heater_range_variation    : array [1..max_gain_variations] of integer;
  Imeter                    : meters;
  interval                  : time;
  kval                      : array [1..max_points] of real;
  last_setpoint             : real;
  llength                   : real;
  manual_offset             : real;
  max_time,min_time         : time;
  max_X_value,max_Y_value   : integer;
  minicam_address           : str5;
  next_reading_time         : time;
  now_time                  : time;
  no_more_setpoint_changes  : boolean;
  no_of_devices             : integer;
  no_of_gain_variations     : integer;
  no_of_setpoints           : integer;
  no_points_in_block        : integer;
  nVmeter,nVcontrol         : meters;
  nV_ontime                 : time;
  nV_power,nV_old_status    : boolean;
  nVpower                   : boolean;
  nVscale                   : real;
  next_setpoint_temperature : real;
  next_setpoint_time        : time;
  outfile,path              : str255;
  old_time                  : time;
```

```
  pause_flag                : boolean;
  point                     : integer;
  power                     : array [1..max_points] of real;
  rate_variation            : array [1..max_gain_variations] of real;
  reading_taken             : boolean;
  reset_variation           : array [1..max_gain_variations] of real;
  screen_X_axis             : integer;
  screen_Y_axis             : integer;
  setpoint_index            : integer;
  setpoints_sorted          : boolean;
  setpoint_time             : array [0..max_setpoints] of time;
  setpoint_temperature      : array [0..max_setpoints] of real;
  start_time                : time;
  start_str                 : str20;
  setpt                     : array [1..max_points] of real;
  temp                      : array [1..max_points] of real;
  temp_variation_lower      : array [1..max_gain_variations] of real;
  temp_variation_upper      : array [1..max_gain_variations] of real;
  thick                     : real;
  title_line1,title_line2   : integer;
  title_line3               : integer;
  total_lines               : integer;
  top_extra_lines           : integer;
  gain,rate,rreset          : array [1..max_points] of real;
  Tcontrol                  : meters;
  timeout                   : boolean;
  thermocouple_file         : str255;
  V                         : array [1..max_points] of real;
  Vfsd                      : real;
  Vmeter                    : meters;
  width                     : real;
  xborder_left              : integer;
  xborder_right             : integer;
  xborder_left_thickness    : integer;
  xborder_right_thickness   : integer;
  xleft,xright,ytop,ybottom : integer;
  Xmin,Xmax,Ymin,Ymax       : real;
  Xstep,Ystep               : real;
  yborder_top               : integer;
  yborder_bottom            : integer;
  yborder_top_thickness     : integer;
  yborder_bottom_thickness  : integer;
  yheight,xlength           : integer;
  ylabel_position           : integer;
  xlabel_position           : integer;
  ycentre,xcentre           : integer;

procedure meter_defaults;
procedure gain_variation_defaults;
procedure var_defaults;


implementation


procedure meter_defaults;

{This procedure sets up the default meter configurations
 and is called from DEVICE_DEFAULTS in the main program
 Extra special beware- DO NOT have any IEEE device number 1 as the
 computer IEEE routines assume that the computer is address 1}

begin

{ The following are 'physical' device numbers note that the parameter
 names need to be defined in the VAR statement at the top of this unit}

  fluke8860:=1;
  k197:=2;
  k175:=3;
  dr91c:=4;
  solar:=5;
  weston:=6;
  minicam:=7;
  no_of_devices:=7;

  {default_chan gives the IEEE address of each particular device, they
  may be altered by the main program if the user wishes                }

  minicam_address:='42';
  device_name[fluke8860]:='Fluke 8860          ';
  device_default_chan[fluke8860]:=16;
  device_name[k197]:='Keithley 197        ';
  device_default_chan[k197]:=2;
  device_name[k175]:='Keithley 175        ';
  device_default_chan[k175]:=3;
  device_name[dr91c]:='DR91C controller    ';
  device_default_chan[dr91c]:=4;
  device_name[solar]:='Solartron           ';
  device_default_chan[solar]:=7;
  device_name[weston]:='Weston              ';
  device_default_chan[weston]:=9;
  device_name[minicam]:='Minicam board       ';
  device_default_chan[minicam]:=6;

  Imeter.device:=k175;
  Vmeter.device:=solar;
  nVmeter.device:=k197;
  Tcontrol.device:=dr91c;
  nVcontrol.device:=minicam;
 end; {meter_defaults}


procedure gain_variation_defaults;

{At different temperatures, the PID settings will need to vary
this sets the default configurations
If you add any more, make sure that you change
no_gain variations at the end of this procedure}

begin
  temp_variation_lower[1]:=10.;
  temp_variation_upper[1]:=15.;
  gain_variation[1]:=0.2;
  rate_variation[1]:=0.;
  reset_variation[1]:=0.;
  heater_range_variation[1]:=3;

  temp_variation_lower[2]:=15.;
```

```
    gain_variation[2]:=0.3;
    rate_variation[2]:=0.;
    reset_variation[2]:=0.;
    heater_range_variation[2]:=4;
    temp_variation_lower[3]:=20.;
    temp_variation_upper[3]:=30.;
    gain_variation[3]:=0.5;
    rate_variation[3]:=0.0;
    reset_variation[3]:=0.0;
    heater_range_variation[3]:=4;

    temp_variation_lower[4]:=30.;
    temp_variation_upper[4]:=40.;
    gain_variation[4]:=0.5;
    rate_variation[4]:=0.0;
    reset_variation[4]:=0.0;
    heater_range_variation[4]:=4;

    temp_variation_lower[5]:=40.;
    temp_variation_upper[5]:=50.;
    gain_variation[5]:=0.5;
    rate_variation[5]:=0.0;
    reset_variation[5]:=0.0;
    heater_range_variation[5]:=4;

    temp_variation_lower[6]:=50.;
    temp_variation_upper[6]:=60.;
    gain_variation[6]:=10;
    rate_variation[6]:=0.0;
    reset_variation[6]:=0.0;
    heater_range_variation[6]:=5;

    temp_variation_lower[7]:=60.;
    temp_variation_upper[7]:=80.;
    gain_variation[7]:=20.;
    rate_variation[7]:=0.0;
    reset_variation[7]:=0.0;
    heater_range_variation[7]:=5;

    temp_variation_lower[8]:=80.;
    temp_variation_upper[8]:=300.;
    gain_variation[8]:=20.;
    rate_variation[8]:=0.0;
    reset_variation[8]:=0.0;
    heater_range_variation[8]:=5;
    no_of_gain_variations:=8;
  end;


  procedure var_defaults;

  {This procedure sets up some of the sample and meter configurations
   and is called from VARIABLE_DEFAULTS in the main program}

  begin
    width:=3;
    thick:=3;       {sizes in millimetres}
    llength:=6;
    area:=width*thick;
```

```
    nVscale:=30;
    Vfsd:=0.98;
    nVpower:=true;
    manual_offset:=0;
    average_readings:=1;
    no_points_in_block:=1;
    auto_gain:=true;
  end;


end.
```

```
{ UNIT DEVICES                                                              }
{ Contains the routines for reading/writing to the DR91C,Keithley 197&175 }
{ Schlumberger Solartron and Fluke 8840 DMMs                                }
{ M.R.Delap 1 November 1989                                                 }
{*************************************************************************** }


 unit devices;

{*************************************************************************** }
{ Note that the set temperature is in volts and that the conversion         }
{ routine for the new sensor is still here                                  }
{*************************************************************************** }

 interface

 uses defaults,mrdglob,ieee,mrdutils,crt;

 procedure init_ieee;
 function dr91c_com(comstr:str80; rep:char): str80;
 function dr91c_com_write(filename:namfil;comstr:str80;rep:char):str80;
 procedure init_dr91c;
 function dr91c_read_sensor: real;
 function dr91c_read_setpoint: real;
 function convert_temp(temp:real):real;
 procedure dr91c_write_setpoint(temp:real);
 function dr91c_read_gain: real;
 procedure dr91c_write_gain(value:real);
 function dr91c_read_reset: real;
 procedure dr91c_write_reset(value:real);
 function dr91c_read_rate: real;
 procedure dr91c_write_rate(value:real);
 function dr91c_read_heater_power: real;
 procedure dr91c_write_heater_range(value:integer);
 function decipher_keithley(instr:str80; digits:integer): real;
 function k197_com(comstr:str80; rep:char; var range:str3): real;
 procedure init_k197;
 function k197_range_check(intended:str3): boolean;
 function k175_com(comstr:str80; rep:char; var range:str3): real;
 function k175_range_check(intended:str3): boolean;
 procedure init_k175;
 function solar_com(comstr:str80; rep:char): real;
 procedure init_solar;
 function weston_com(comstr:str80;rep:char):real;{*** not written **********}
 procedure init_weston;{************************ not written **********}
 function read_temp(device:integer):real;
 function read_setpoint(device:integer):real;
 procedure write_setpoint(device:integer; value:real);
 function read_gain(device:integer): real;
 procedure write_gain(device:integer; value:real);
 function read_rate(device:integer): real;
 procedure write_rate(device:integer;value:real);
 function read_reset(device:integer): real;
 procedure write_reset(device:integer; value:real);
 function read_heater_power(device:integer): real;
 procedure write_heater_range(device:integer; value:integer);
 function read_current(device:integer): real;
 function read_volt(device:integer): real;
 function read_nV(device:integer): real;
 procedure minicam_com( comstr : str20 );
 function dal(madd,n : str5) : str255;
 procedure fluke8860_com(comstr:str12);
 procedure cr_eoi;
 procedure init_fluke8860;
 function read_fluke8860 : double;
 function ave_fluke8860 : double;

{*************UNIT DEVICES************************************************** }

 implementation


 procedure init_ieee;
  begin
   cntrllr:=true;
   my_flag:=false;
   my_addr:=1;
   int1stat:=0;
   init;
 end; {init_ieee}

{*********DR91C CONTROLLER ROUTINES************************************** }

 function dr91c_com(comstr:str80;rep:char):str80;
 var
  dumstr : str5;
  dummy  : real;
 begin
   init_ieee;
   eois:=chr(10);
   iors:='i';
   datastring:=comstr+chr(13);
   wr_str(dr91c);
   if (rep='R') then begin
         read_str(dr91c);
         dr91c_com:=datastring;
        end
   else dr91c_com:=' ';
  end;


 function dr91c_com_write(filename:namfil;comstr:str80;rep:char):str80;
 var
  dumstr : str5;
  dummy  : real;
 begin
   init_ieee;
   eois:=chr(10);
   iors:='i';
   datastring:=comstr+chr(13);
   wr_str(dr91c);
   if (rep='R') then begin
         read_to_file(filename,dr91c);
         dr91c_com_write:=datastring;
        end
   else dr91c_com_write:=' ';
  end;
```

```
 var  dumstr : str80;
 begin
   dumstr:=dr91c_com(dr91cinitstr1,'N');
   dumstr:=dr91c_com(dr91cinitstr2,'N');
 end;


 function dr91c_read_sensor : real;
 var
   dumstr : str80;
   dummy  : integer;
   dummy2 : real;
 begin
   dumstr:=dr91c_com('W0','R');
   dumstr:=copy(dumstr,1,7);
   val(dumstr,dummy2,dummy);
   dr91c_read_sensor:=dummy2;
 end;


 function dr91c_read_setpoint: real;
 var
   dumstr : str80;
   dummy  : integer;
   dummy2 : real;
 begin
   dumstr:=dr91c_com('W0','R');
   dumstr:=copy(dumstr,19,6);
   val(dumstr,dummy2,dummy);
   dr91c_read_setpoint:=dummy2;
 end;

{*************UNIT DEVICES*************************************************** }

 function convert_temp(temp:real):real;
 var
  coeff : Array[1..12] of real;
  x,x1,x2,sum : real;
  loop: integer;
 begin
  {Coefficients taken from fitten 7th order chebyshev polynomial}
  coeff[1]:=1.4154967767;      coeff[2]:=-9.3711538784e-4;
  coeff[3]:=1.1101416234e-5;   coeff[4]:=-1.2055536247e-7;
  coeff[5]:=4.3326955522e-10;  coeff[6]:=-7.0226932006e-13;
  coeff[7]:=4.2746418359e-16;
  x1:=1;
  x2:=temp;
  sum:=x1*coeff[1]+x2*coeff[2];
  for loop:=3 to 7 do begin
      x:=2*temp*x2-x1;
      sum:=sum+coeff[loop]*x;
      x1:=x2; x2:=x;
  end;
  convert_temp:=sum;
 end;

{*************UNIT DEVICES*************************************************** }

 procedure dr91c_write_setpoint(temp:real);
 var
  dumstr : str80;
  volt   : real;
 begin
   volt:=convert_Temp(temp);
{   str(temp:5:2,dumstr);             }
 str(volt:6:4,dumstr);
   dumstr:=dr91c_com('S'+dumstr,'N')

 end;


 function dr91c_read_gain : real;
 var
  dumstr : str80;
  dummy  : real;
  error  : integer;
 begin
   dumstr:=dr91c_com('W3','R');
   dumstr:=copy(dumstr,1,3);
   val(dumstr,dummy,error);
   dr91c_read_gain:=dummy;
 end;


 procedure dr91c_write_gain(value:real);
 var
  dumstr : str80;
 begin
   str(value:4:1,dumstr);
   dumstr:=dr91c_com('P'+dumstr,'N');
 end;


 function dr91c_read_reset : real;
 var
  dumstr : str80;
  dummy  : real;
  error  : integer;
 begin
   dumstr:=dr91c_com('W3','R');
   dumstr:=copy(dumstr,9,3);
   val(dumstr,dummy,error);
   dr91c_read_reset:=dummy;
 end;


 procedure dr91c_write_reset(value:real);
 var dumstr : str80;
 begin
   str(value:5,dumstr);
   dumstr:=dr91c_com('I'+dumstr,'N');
 end;


 function dr91c_read_rate : real;
 var
  dumstr : str80;
  dummy  : real;
  error  : integer;
```

```
  dumstr:=dr91c_com('W3','R');
  dumstr:=copy(dumstr,5,3);
  val(dumstr,dummy,error);
  dr91c_read_rate:=dummy;
end;


procedure dr91c_write_rate(value:real);
var
 dumstr : str80;
begin
  str(value:5,dumstr);
  dumstr:=dr91c_com('D'+dumstr,'N');
end;


function dr91c_read_heater_power : real;
var
  dumstr : str80;
  dummy  : real;
  error  : integer;
begin
  dumstr:=dr91c_com('W3','R');
  dumstr:=copy(dumstr,15,3);
  val(dumstr,dummy,error);
  dr91c_read_heater_power:=dummy;
end;


procedure dr91c_write_heater_range(value:integer);
{ value = 0 ... heater off
          1.....heater off
          2.....10e-3
          3.....10e-2
          4.....10e-1
          5.....max     }
var
 dumstr  : string[1];
 dumstr2 : str20;
begin
  str(value,dumstr);
  dumstr2:=dr91c_com('R'+dumstr,'N');
end;


{*********Keithley 197 & 175 routines*********************************** }

function decipher_keithley(instr:str80; digits:integer) : real;
var
  dummy  : real;
  dummy2 : integer;
  dumstr : str80;
begin
  dummy2:=digits+6;
  dumstr:=copy(instr,5,dummy2);
  val(dumstr,dummy,dummy2);
  decipher_keithley:=dummy;
end;
```

```
function k197_com(comstr:str80; rep:char; var range:str3) : real;
begin
  init_ieee;
  eois:=chr(10);
  iors:='i';
  datastring:=comstr+'X'+chr(13);
  wr_str(k197);
  k197_com:=0.0;
  range:='   ';
  if (rep='R') then begin
                      read_str(k197);
                      k197_com:=decipher_keithley(datastring,5);
                      range:=copy(datastring,2,3);
                    end;
  end;


procedure init_k197;
var
  dummy  : real;
  dumstr : str3;
begin
  sel_dev_rem(k197);
  dummy:=k197_com(k197initstr,'N',dumstr);
end;


function k197_range_check(intended:str3) : boolean;
var
  dummy  : real;
  dumstr : str3;
begin
  dummy:=k197_com('G0','R',dumstr);
  if (dumstr<>intended) then k197_range_check:=false
                        else k197_range_check:=true;
end;


function k175_com(comstr:str80; rep:char; var range:str3) : real;
begin
  init_ieee;
  eois:=chr(10);
  iors:='i';
  datastring:=comstr+'X'+chr(13);
  wr_str(k175);
  k175_com:=0.0;
  range:='   ';
  if (rep='R') then begin
                      read_str(k175);
                      k175_com:=decipher_keithley(datastring,4);
                      range:=copy(datastring,2,3);
                    end;
 end;


function k175_range_check(intended:str3) : boolean;
var
  dummy  : real;
  dumstr : str3;
```

```
  dummy:=k175_com('','R',dumstr);
  if (dumstr<>intended) then k175_range_check:=false
                        else k175_range_check:=true;
end;


procedure init_k175;
var
  dummy  : real;
  dumstr : str3;
begin
  sel_dev_rem(k175);
  dummy:=k175_com(k175initstr,'N',dumstr);
end;

{*********Solartron routines************************************* }

function solar_com(comstr:str80; rep:char) : real;
var
  dummy  : real;
  dummy2 : integer;
  dumstr : str12;
begin
  init;
  eois:=chr(13);
  iors:='i';
  datastring:=comstr;
  wr_str(solar);
  solar_com:=0.0;
  if (rep='R') then begin
                      read_str(solar);
                      dumstr:=copy(datastring,1,8);
                      val(dumstr,dummy,dummy2);
                      str(dummy,datastring);
                      solar_com:=dummy;
                    end;
end;


procedure init_solar;
var
  dummy  : real;
begin
  sel_dev_rem(solar);
  dummy:=solar_com(solarinitstr,'N');
end;

{*********Weston routines************************************* }

function weston_com(comstr:str80; rep:char):real;
begin
writeln ('This routine is not written yet');
end;


procedure init_weston;
begin
  sel_dev_rem(weston);
writeln('init_weston not written yet');
```

```
{*****************************************************************************
 * If more meters are added then the following routines should be altered *
 * to point to the new devices                                            *
 * where possible, the meters are asked to confirm the display units      *
 * Some meters (ie Solartron can't do this so the program assumes that    *
 * all is OK                                                              *
 *****************************************************************************}


function read_temp(device:integer) : real;
begin
  if (device=dr91c) then read_temp:=dr91c_read_sensor
                    else read_temp:=-9999.
end;


function read_setpoint(device:integer) : real;
begin
  if (device=dr91c) then read_setpoint:=dr91c_read_setpoint
                    else read_setpoint:=-999.;
end;


procedure write_setpoint(device:integer; value:real);
var dumstr : str20;
begin
  if (device=dr91c) then dr91c_write_setpoint(value);
end;


function read_gain(device:integer) : real;
var
 dummy : real;
begin
  if (device=dr91c) then dummy:=dr91c_read_gain;

  read_gain:=dummy;
end;


procedure write_gain(device:integer;value:real);
begin
  if (device=dr91c) then dr91c_write_gain(value);

end;


function read_rate(device:integer) : real;
var
 dummy : real;
begin
  if (device=dr91c) then dummy:=dr91c_read_rate;

  read_rate:=dummy;
end;
```

```
begin
  if (device=dr91c) then dr91c_write_rate(value);

end;


function read_reset(device:integer) : real;
var
 dummy : real;
begin
  if (device=dr91c) then dummy:=dr91c_read_reset;

  read_reset:=dummy;
end;


procedure write_reset(device:integer;value:real);
begin
  if (device=dr91c) then dr91c_write_reset(value);

end;


function read_heater_power(device:integer) : real;
var
 dummy : real;
begin
  if (device=dr91c) then dummy:=dr91c_read_heater_power;

  read_heater_power:=dummy;
end;

procedure write_heater_range(device:integer; value:integer);
begin
 if (device=dr91c) then dr91c_write_heater_range(value);
end;


function read_current(device:integer) : real;
var
 range_ok : boolean;
 dumstr   : str3;
begin
  repeat
  begin
    range_ok:=true;
    if (device=k197) then begin
                           read_current:=k197_com('','R',dumstr);
                           range_ok:=k197_range_check('DCA');
                          end;
    if (device=k175) then begin
                           read_current:=k175_com('','R',dumstr);
                           range_ok:=k175_range_check('DCA');
                          end;
    if (device=solar) then read_current:=solar_com('T0','R');
    if (device=weston) then read_current:=weston_com('U0','R');
    if (NOT range_ok) then
                        hold('The current meter is not correctly set. '+
                             'Presently it is, '+dumstr);
  until range_ok;
end;


function read_volt(device:integer) : real;
var
 range_ok : boolean;
 dumstr   : str3;
begin
  repeat
  begin
    range_ok:=true;
    if (device=k197) then begin
                           read_volt:=k197_com('','R',dumstr);
                           range_ok:=k197_range_check('DCV');
                          end;
    if (device=k175) then begin
                           read_volt:=k175_com('','R',dumstr);
                           range_ok:=k175_range_check('DCV');
                          end;
    if (device=solar) then read_volt:=solar_com('T0','R');
    if (device=weston) then read_volt:=weston_com('U0','R');
    if (NOT range_ok) then
                        hold('The voltage meter is not correctly set. '+
                        'Presently it is '+dumstr);
  end;
  until range_ok;
 end;


function read_nV(device:integer) : real;
var
 range_ok : boolean;
 dumstr   : str3;
begin
  repeat
  begin
    range_ok:=true;
    if (device=k197) then begin
                           read_nV:=k197_com('','R',dumstr);
                           range_ok:=k197_range_check('DCV');
                          end;
    if (device=k175) then begin
                           read_nV:=k175_com('','R',dumstr);
                           range_ok:=k175_range_check('DCV');
                          end;
    if (device=solar) then read_nV:=solar_com('T0','R');
    if (device=weston) then read_nV:=weston_com('U0','R');
    if (NOT range_ok) then
                    hold('The nanovoltmeter meter is not correctly set.'+
                         ' Presently it is '+dumstr);
  end;
  until range_ok;
 end;


{*************************************************************************}
{*                                                                       *}
{*             Minicam Routines    -   Version One                       *}
```

```
{*                    D.B.Bamboick  August 1987                              *}
{*  Altered by M.R.Delap for use with thermal conductivity experiment        *}
{*                            24 November 1989                               *}
{*                                                                           *}
{*****************************************************************************}

procedure minicam_com(comstr : str20 );
begin
 cntrllr:=true;
 my_flag:=false;
 my_addr:=1;
 intlstat:=0;
 eois:=chr(13);
 iors:='s';
 datastring:=comstr+chr(13);
 init;
 wr_str(minicam);
 read_str(minicam);
end;


function dal(madd,n : str5) : str255;
var comstr : str20;
begin
 comstr:='DA1,'+madd+','+n;
 minicam_com(comstr);
 dal:=datastring;
end;


procedure fluke8860_com(comstr:str12);
begin
  eois:=chr(10);
  iors:='i';
  datastring:=comstr+chr(13)+chr(10);
  init;
  wr_str(fluke8860);
end;

procedure cr_eoi;
begin
  fluke8860_com('W2');
end;

procedure init_fluke8860;
begin
  eois:=chr(10);
  iors:='i';
  gtsby;
  tcsy;
  tci($f8);
  wr_byte($30);
  gtsby;
  tcsy;
  wr_byte($3f);
  wr_byte($5f);
  wr_byte($30);
  gtsby;
  wr_byte($2a);
  wr_byte($54);
  wr_byte($30);
  wr_byte($0d);
  set_eoi;
  last:=true;
  wr_byte($0a);
  last:=false;
  tcsy;
end;

function read_fluke8860;
var dum : double;
    ec  : integer;
begin
  eois:=chr(10);
  iors:='i';
  gtsby;
  tcsy;
  read_str(fluke8860);
  datastring:=copy(datastring,1,length(datastring)-2);
  val(datastring,dum,ec);
  read_fluke8860:=dum;
end;

function ave_fluke8860;
var sum : double;
    i   : integer;
begin
  sum:=0.0;
  for i:=1 to 10 do
    begin
      sum:=sum+read_fluke8860;
      delay(500);
    end;
  ave_fluke8860:=sum/10.0;
end;


end.
```

```
(*                                                                          *)
(*  UNIT ieee - Pascal routines for Scientific Solutions RevD board         *)
(*  12/04/89 D.B.Lambrick                                                   *)
(*                                                                          *)
(*  Rd_Byte Amended 7/7/88                                                  *)
(*  Conversion to Turbo 5.0 21/3/89 - Unit conv. 12/04/89                   *)
(*                                                                          *)
(**************************************************************************)

unit ieee;

{------------------------------------------------------------------------}

 interface

 uses mrdglob,mrdutils;

  const Bdadd : integer = $310;

  type
    laddr  = record
               prim : integer;
                sec : array[1..10] of integer;
    end;
    devadd = array[1..10] of laddr;
    stray  = array[1..10] of string[255];

  var
      int1stat,my_addr,lastint1,dat,poll_resp,ieeestatus,srq : integer;
      bit,sense,i,n1 : integer;
      eois,iors : char;
      cntrl1r,last,my_flag,first1n,statusactive: boolean;
      datastring,petfile : str255;
      filnam : namfil;
      sar : stray;
      devla,devta : devadd;
      filvar : text;
      pcfile : str14;

  procedure ibclear;
  procedure wait12;
  procedure tci(comm:integer);
  procedure fetch_la(devnum:integer);
  procedure out_ta(devnum:integer);
  procedure init;
  procedure gtsby;
  procedure go_idle;
  procedure gtscnt;
  procedure abrt;
  procedure abort;
  procedure devclr;
  procedure getrigg(devnum:integer);
  procedure devloc(devnum:integer);
  procedure llo;
  procedure my_rsp(devnum:integer);
  procedure read_setup(devnum:integer);
  procedure read_end;
  procedure read_str(devnum:integer);
  procedure read_to_file(filnam:namfil;devnum:integer);
  procedure wr_str(devnum:integer);
  procedure wr_from_file(filnam:namfil;devnum:integer);
  procedure rd_byte;
  procedure wr_byte(dat:integer);
  procedure par_poll;
  procedure unconfig;
  procedure disab_poll(devnum:integer);
  procedure read_bus_stat;
  procedure sel_dev_clr(devnum:integer);
  procedure sel_dev_rem(devnum:integer);
  procedure ser_poll(devnum:integer);
  procedure pp_en(devnum:integer);
  procedure all_loc;
  procedure all_rem;
  procedure trnsfr(devnum:integer);
  procedure my_la;
  procedure my_ta;
  procedure listen(devnum:integer);
  procedure talk(devnum:integer);
  procedure unlisten;
  procedure untalk;
  procedure tcsy;
  procedure tasy;
  procedure rcv_cntrl;
  procedure pass_cntrl(devnum:integer);
  procedure set_eoi;
  procedure rdsup(devnum:integer);
  procedure rd_arry(sar:stray;sep,lastchar:char;devnum:integer);
  procedure zero_add;
{------------------------------------------------------------------------}

implementation

 procedure ibclear;                { Waits for the input buffer to clear }
 var   dummy : integer;            { i.e. 8292 buffer}
 begin
  repeat
   dummy:=Port[Bdadd+9]
  until (dummy and 2)<>2;
 end;

 procedure wait12;
 var w12dum : integer;
 begin
  repeat
   w12dum:=Port[Bdadd+1];
  until (w12dum and 2) = 2;
 end;

 procedure tci(comm:integer);
 var   dummy : integer;
 begin
  ibclear;
  Port[Bdadd+10]:=0;
  Port[Bdadd+9]:=comm;
  repeat
   dummy:=Port[Bdadd+12]
  until (dummy and 1) <> 0;
 end;
```

```
procedure fetch_la;
var j :integer;
begin
  if devla[devnum].prim = my_addr then my_flag:=true
   else begin
          Port[Bdadd+0]:=devla[devnum].prim+$20;
          j:=1;
          while (devla[devnum].sec[j] <> 0) (or (j=1)) do
           begin
            Port[Bdadd+0]:=devla[devnum].sec[j]+$60;
            j:=j+1;
           end;
         end;
end;

procedure out_ta;
var j : integer;
begin
 wait12;
 Port[Bdadd+0]:=devta[devnum].prim+$40;
 j:=1;
 while (devta[devnum].sec[j] <> 0)  do
  begin
   wait12;
   Port[Bdadd+0]:=devta[devnum].sec[j]+$60;
   j:=j+1;
  end;
end;

procedure init;
var   dummy : integer;
begin
 dummy:=Port[bdadd];       { Clear IBF }
 ibclear;                  { Wait for IBF low }
 Port[Bdadd+9]:=$f2;       { Reset 8292 }
 Port[Bdadd+5]:=$2;        { Reset 8291A }
 Port[Bdadd+5]:=$25;       { Init T1 delay }
 Port[Bdadd+1]:=$93;       { 8291A intrr enble 1 }
 Port[Bdadd+2]:=$0;        { 8291A intrr enble 2 }
 Port[Bdadd+3]:=$0;        { 8291A serial poll mode }
 Port[Bdadd+5]:=$80;       { 8291A Aux mode A }
 Port[Bdadd+5]:=$a1;       { 8291A Aux mode B, enable cpt }
 Port[Bdadd+6]:=$1;        { 8291A addr mode - bd is 1 }
 Port[Bdadd+6]:=$e0;       { Disable addr 1 }
 ibclear;
 Port[Bdadd+8]:=$0;        { Clr 8292 err mask }
 ibclear;
 Port[Bdadd+8]:=$a0;       { Enable tci }
 tci($e6);                 { Read 8292 cont status reg }
 repeat
  dummy:=Port[Bdadd+9]
 until (dummy and 1) <> 0;
 Port[Bdadd+4]:=$80;
 Port[Bdadd+5]:=$0;
end;

procedure gtsby;
begin
 tci($f6);
end;

procedure go_idle;
begin
 tci($f1);
end;

procedure gtscnt;
begin
 tci($f4);
end;

procedure abrt;
begin
 ibclear;
 Port[Bdadd+9]:=$f9;
end;

procedure abort;
begin
 if cntrllr then abrt else
  begin
   tci($f9);
   Port[Bdadd+4]:=$80;
   Port[Bdadd+5]:=$0;
   cntrllr:=true;
  end;
end;

procedure devclr;
begin
 wait12;
 Port[Bdadd+0]:=$14;
end;

procedure getrigg;
begin
 wait12;
 Port[Bdadd+0]:=$3f;
 fetch_la(devnum);
 wait12;
 Port[Bdadd+0]:=$8;
end;

procedure devloc;
begin
 wait12;
 Port[Bdadd+0]:=$3f;
 fetch_la(devnum);
 wait12;
 Port[Bdadd+0]:=1;
end;

procedure llo;
begin
 wait12;
 Port[Bdadd+0]:=$11;
end;

procedure my_rsp;
```

```
 out_ta(devnum);
 wait12;
 Port[Bdadd+0]:=my_addr+$20;
 wait12;
 Port[Bdadd+4]:=$40;
 Port[Bdadd+5]:=0;
 tci($f6);
end;

procedure read_setup;
var dummy : integer;
begin
 if eois='' then dummy:=13 else dummy:=ord(eois);
 Port[Bdadd+7]:=dummy;
 if iors='s' then Port[Bdadd+5]:=$86 else Port[Bdadd+5]:=$82;
 if my_flag then my_rsp(devnum);
 if cntrllr then begin
                  wait12;
                  Port[Bdadd+0]:=$3f;
                  my_rsp(devnum);
                 end;
 if not (my_flag or cntrllr) then repeat
                                   dummy:=Port[Bdadd+4];
                                  until (dummy and 4)=0;
end;

procedure read_end;
var dummy : integer;
begin
 if cntrllr then begin
                  tci($fd);
                  Port[Bdadd+4]:=$80;
                  Port[Bdadd+5]:=$3;
                  Port[Bdadd+5]:=$80;
                  Port[Bdadd+5]:=$0;
                 end
            else begin
                  Port[Bdadd+5]:=$80;
                  lastint1:=int1stat;
                  Port[Bdadd+5]:=$3;
                  dummy:=Port[Bdadd+0];
                 end;
end;

procedure read_str;
var iobyt : integer;
    achar : string[1];
begin
 read_setup(devnum);
 datastring:='';
 repeat
  int1stat:=port[Bdadd+1];
  if ((int1stat and 1)<>0) then
    begin
     iobyt:=Port[Bdadd+0];
     achar:=chr(iobyt);
     datastring:=datastring+achar;
{    write(datafile,achar);}
    end;
 read_end;
end;

procedure read_to_file;
var datout : text;
    iobyt : integer;
    achar : string[1];
    loop : integer;
begin
 loop:=0;
 assign(datout,filnam);
 rewrite(datout);
 read_setup(devnum);
 writeln('Done setup');
 space;
 int1stat:=Port[Bdadd+1];
 while ((int1stat and 1)=1) and ((int1stat and $11)<>$11) do
  begin
   loop:=loop+1;
   iobyt:=Port[Bdadd+0];
   achar:=chr(iobyt);
   write(datout,achar);
   write(achar);
   if loop>=80 then begin
                     loop:=1;
                     write(datout,chr(13));
                    end;
   int1stat:=Port[Bdadd+1];
  end;
 read_end;
 close(datout);
 end;

procedure write_setup;
var dummy : integer;
begin
 if cntrllr then begin
                  wait12;
                  gtsby;
                  tcsy;
                  Port[Bdadd+0]:=my_addr+$40;
                  wait12;
                  Port[Bdadd+0]:=$3f;
                  fetch_la(devnum);
                  wait12;
                  gtsby;
                 end
            else begin
                  repeat
                   dummy:=Port[Bdadd+4];
                  until ((dummy and 2)=0);
                  if not ((lastint1 and 2)=2) then wait12;
                 end;
end;

procedure wr_str;
var iobyt,strcnt : integer;
    schar : string[1];
    achar : char;
```

```
begin
 write_setup(devnum);
 for strcnt:=1 to length(datastring)-1 do
  begin
   schar:=copy(datastring,strcnt,1);
   achar:=schar[1];
   iobyt:=ord(achar);
   Port[Bdadd+0]:=iobyt;
   wait12;
  end;
 schar:=copy(datastring,length(datastring),1);
 achar:=schar[1];
 iobyt:=ord(achar);
 if iors='i' then begin
                   Port[Bdadd+5]:=$6;
                   Port[Bdadd+0]:=iobyt;
                  end
             else begin
                   if achar <> eois then begin
                                          Port[Bdadd+0]:=iobyt;
                                          str(iobyt,dumstr);
                                          hold(dumstr);
                                          wait12;
                                         end;
                   Port[Bdadd+0]:=ord(eois);
              {     writeln(ord(eois));}
                  end;
 lastint1:=int1stat;
 if cntrllr then tcsy;
end;

procedure wr_from_file;
var datin : file of char;
    schar : string[1];
    achar : char;
begin
 assign(datin,filnam);
 reset(datin);
 write_setup(devnum);
 repeat
  read(datin,achar);
  while not eof(datin) do
   begin
    Port[Bdadd+0]:=ord(achar);
    wait12;
   end;
 until eof(datin);
 if iors='i' then begin
                   Port[Bdadd+5]:=$6;
                   Port[Bdadd+0]:=ord(achar);
                  end
             else begin
                   if achar <> eois then begin
                                          Port[Bdadd+0]:=ord(achar);
                                          wait12;
                                         end;
                   Port[Bdadd+0]:=ord(eois);
                  end;
 close(datin);
 if cntrllr then tci($fd);
end;

procedure rd_byte;
begin
 repeat
  int1stat:=Port[Bdadd+1];
 until ((int1stat and 1)=1) or ((int1stat and $10)=$10);
 if (int1stat and $10)=$10 then begin
                    if (int1stat and $1)=$1 then begin
                                                   dat:=Port[Bdadd+0];
                                                   last:=true;
                                                  end
                                            else
                                                  begin
                                                   dat:=256;
                                                   last:=true;
                                                  end;
                   end
                else dat:=Port[Bdadd+0];

end;

procedure wr_byte;
begin
 Port[Bdadd+0]:=dat;
 if not last then wait12;
 if not last then write(chr(dat));
 if last then hold(chr(dat));
end;

procedure par_poll;
var dummy : integer;
begin
 Port[Bdadd+4]:=$40;
 Port[Bdadd+5]:=0;
 repeat
  dummy:=Port[Bdadd+9];
 until ((dummy and 2)<>2);
 Port[Bdadd+9]:=$f5;
 repeat
  dummy:=Port[Bdadd+1];
 until ((dummy and 1)<>0);
 poll_resp:=Port[Bdadd+0];
 Port[Bdadd+4]:=$80;
 Port[Bdadd+5]:=$0;
end;

procedure unconfig;
begin
 wait12;
 Port[Bdadd+0]:=$15;
end;

procedure disab_poll;
begin
 wait12;
 Port[Bdadd+0]:=$3f;
 fetch_la(devnum);
```

```
 Port[Bdadd+0]:=$5;
 wait12;
 Port[Bdadd+0]:=$70;
 unconfig;
end;

procedure read_bus_stat;
var dummy : integer;
begin
 tci($e7);
 Port[Bdadd+9]:=$e7;
 repeat
  dummy:=Port[Bdadd+9];
 until not ((dummy and 1)=0);
 ieeestatus:=Port[Bdadd+8];
 srq:=ieeestatus and 1;
end;

procedure sel_dev_clr;
begin
 wait12;
 Port[Bdadd+0]:=$3f;
 fetch_la(devnum);
 wait12;
 Port[Bdadd+0]:=$4;
end;

procedure sel_dev_rem;
begin
 wait12;
 gtsby;
 tcsy;
{ Port[Bdadd+0]:=$3f;}
 tci($f8);
 fetch_la(devnum);
{  gtsby; }
end;

procedure ser_poll;
var dummy : integer;
begin
 wait12;
 Port[Bdadd+0]:=$3f;
 wait12;
 Port[Bdadd+0]:=$18;
 out_ta(devnum);
 wait12;
 Port[Bdadd+4]:=$40;
 Port[Bdadd+5]:=0;
 gtsby;
 repeat
  dummy:=Port[Bdadd+1];
 until not ((dummy and 1)=0);
 poll_resp:=Port[Bdadd+0];
 Port[Bdadd+4]:=$80;
 Port[Bdadd+5]:=$40;
 tci($fd);
 wait12;
 Port[Bdadd+0]:=$19;
```

```
end;

procedure pp_en;
begin
 wait12;
 Port[Bdadd+0]:=$3f;
 fetch_la(devnum);
 wait12;
 Port[Bdadd+0]:=$5;
 wait12;
 Port[Bdadd+0]:=$60+bit+sense;
end;

procedure all_loc;
begin
 tci($f7);
end;

procedure all_rem;
begin
 tci($f8);
end;

procedure trnsfr;
var dummy : integer;
begin
 wait12;
 Port[Bdadd+0]:=$3f;
 fetch_la(devnum);
 if not my_flag then
  begin
   out_ta(devnum);
   if eois='' then dummy:=13 else dummy:=ord(eois);
   Port[Bdadd+7]:=dummy;
   if iors='s' then Port[Bdadd+5]:=$87 else Port[Bdadd+5]:=$83;
   Port[Bdadd+4]:=$40;
   Port[Bdadd+5]:=$0;
   tci($f6);
   repeat
    int1stat:=Port[Bdadd+1];
   until not ((int1stat and $10)=0);
   tci($fd);
   Port[Bdadd+5]:=$80;
   Port[Bdadd+4]:=$3;
   Port[Bdadd+5]:=$0;
  end;
end;

procedure my_la;
begin
{  wait12;
 Port[Bdadd+0]:= my_addr + $20;
 wait12; }
 Port[Bdadd+5]:= $82;
 Port[Bdadd+4]:= $40;
 Port[Bdadd+5]:= $0;
 last:=false;
end;

procedure my_ta;
```

```
 wait12;
 Port[Bdadd+0]:= my_addr + $40;
 wait12;
 last:=false;
end;

procedure listen;
begin
 fetch_la(devnum);
end;

procedure talk;
begin
 out_ta(devnum);
end;

procedure unlisten;
begin
 tcsy;
 wait12;
 Port[Bdadd+0]:=$3f;
 gtsby;
end;

procedure untalk;
begin
 tcsy;
 wait12;
 Port[Bdadd+0]:=$5f;
 gtsby;
end;

procedure tcsy;
begin
 tci($fd);
 Port[Bdadd+5]:=$80;
 Port[Bdadd+5]:=$3;
 Port[Bdadd+4]:=$80;
 Port[Bdadd+5]:=$0;
end;

procedure tasy;
begin
 tci($fc);
 Port[Bdadd+5]:=$80;
 Port[Bdadd+5]:=$3;
 Port[Bdadd+4]:=$80;
 Port[Bdadd+5]:=$0;
end;

procedure rcv_cntrl;
var dummy,comm : integer;
begin
 if not cntrllr then
  begin
   if not ((lastint1 and 128)=128) then repeat
                                         intlstat:=Port[Bdadd+1];
                                        until not ((intlstat and 128)=0);
   repeat
   until not ((dummy and 2)=0);
   comm:=Port[Bdadd+5];
   Port[Bdadd+5]:=$f;
   comm:=comm and 127;
   if comm =9 then begin
                    Port[Bdadd+4]:=$80;
                    Port[Bdadd+5]:=$0;
                    cntrllr:=true;
                    tci($fa);
                   end
              else begin
                    hold('Error - undefined command');
                   end;
  end;
end;

procedure pass_cntrl;
begin
 if not cntrllr then begin
                      out_ta(devnum);
                      wait12;
                      Port[Bdadd+0]:=$9;
                      wait12;
                      Port[Bdadd+4]:=$1;
                      Port[Bdadd+5]:=$a1;
                      Port[Bdadd+5]:=$0;
                      lastint1:=0;
                      cntrllr:=false;
                      tci($f1);
                     end;
end;

procedure set_eoi;
begin
 Port[Bdadd+5]:=$6;
 last:=true;
end;

procedure rdsup;
var dummy : integer;
begin
 if my_flag then my_rsp(devnum);
 if cntrllr then begin
                  wait12;
                  Port[Bdadd+0]:=$3f;
                  my_rsp(devnum);
                 end;
 if not (my_flag or cntrllr) then repeat
                                   dummy:=Port[Bdadd+4];
                                  until (dummy and 4)=0;
end;

procedure rd_arry;
var dummy,iobyt,i : integer;
    achar         : string[1];
begin
 Port[Bdadd+5]:=$81;
 rdsup(devnum);
 repeat
```

```
   sar[i]:='';
   repeat
    repeat
     dummy:=Port[Bdadd+1];
    until not ((dummy and 1)=0);
    iobyt:=Port[Bdadd+0];
    achar:=chr(iobyt);
    sar[i]:=sar[i]+achar;
    if (achar <> lastchar) then Port[Bdadd+5]:=$3;
   until (achar=sep) or (achar=lastchar);
  until (achar=lastchar);
 end;

 procedure zero_add;
 var i : integer;
 begin
  for i:=1 to 10 do
   begin
    with devla[i] do begin
                      prim:=0;
                      sec[1]:=0;
                      sec[2]:=0;
                      sec[3]:=0;
                      sec[4]:=0;
                      sec[5]:=0;
                      sec[6]:=0;
                      sec[7]:=0;
                      sec[8]:=0;
                      sec[9]:=0;
                      sec[10]:=0;
                     end;
    with devta[i] do begin
                      prim:=0;
                      sec[1]:=0;
                      sec[2]:=0;
                      sec[3]:=0;
                      sec[4]:=0;
                      sec[5]:=0;
                      sec[6]:=0;
                      sec[7]:=0;
                      sec[8]:=0;
                      sec[9]:=0;
                      sec[10]:=0;
                     end;
   end;
 end;
{-------------------------------------------------------------------------}
end.
```