An investigation of anisotropic magnetic properties of rare earth iron boron alloys

Hawton, M. J

An Investigation of Anisotropic Magnetic Properties of Rare Earth Iron Boron Alloys.

M.J.Hawton

Physics Department

Durham University

Thesis submitted for the Degree of

Doctor of Philosophy

May 1987

The work contained in this thesis was carried out by the author between 1983 and 1987 while a student in the Physics Department at Durham University. Work carried out in collaboration with other parties is clearly acknowledged at the appropriate point.

It is a most remarkable thing. I sat down with the full intention of writing something clever and original; but for the life of me I can't think of anything clever and original - at least not at this moment.

Jerome K Jerome(1943).

Jerome J.K *"Idle Thoughts of an Idle Fellow"* (Arowsmith,London 1943).

Abstract

In 1983 Sumitomo in Japan and General Motors in the USA both announced the production of new 'super magnets' based on a rare earth iron boron mixture. The Sumitomo magnets are based on an intermetallic compound with a composition $Nd_2Fe_{14}B$. This thesis describes an investigation towards understanding the reason why permanent magnets made from this material have such favourable magnetic properties.

A description is given of a new type of torque magnetometer designed to operate within the bore of a superconducting solenoid to make measurements at a field high enough for this type of material. The construction and testing of the instrument are described.

Results are presented of uniaxial magnetocrystalline anisotropy values calculated from torque measurements and also of magnetisation measurements for $R_2Fe_{14}B$, R=Nd,Ho,Gd and Dy from 4.2K to room temperature. The basal plane anisotropy calculated from torque measurements for $Gd_2Fe_{14}B$ from 100K to 300K is also reported. These anisotropy results are related qualitatively to the crystal structure and electronic structure of the rare earth ions.

Observations of regions of uniform magnetization, domains, within the material are also described, and the movement of the boundaries between such domains, domain walls, under varying magnetic fields. These movements charactarise the material and help in understanding the processes involved in making a magnet resistant to demagnetisation.
From the observation of equilibrium domain wall patterns domain wall energies and domain wall thicknesses have been calculated.

Aknowledgements

Space doesn't permit me to mention all by name but ....

I would like to thank the members of the Physics Department for allowing me the use of the Departmental facilities.
I would like particularly to thank W.D.Corner for his patient supervision and encouragement, and all other members of the Solid State Group (past and present) for all help given and time taken in discussion..
The help of all the technical staff in the department is gratefully acknowledged, particularly that of John Scott for his assistance with the superconducting magnet.

The help of Rex Harris and Stewart Abell and the gang at Birmingham must also be mentioned, for use of crystal preparation facitities and provision of samples.

The work was carried out during an SERC studentship, which is gratefully acknowledged. Helium for the magnet and funds for travelling and an additional studenship came from the EEC through the Concerted European Action on Magnets (CEAM) - without this financial support the work would have been much more restricted. Discussion with other collaborators within CEAM was invaluable and Dominic Givord was also very kind in providing single crystal samples grown at the Lab. Louis Néel in Grenoble.

But finally and most importantly my thanks must go to my wife Dorothy without whose patience(sometimes), understanding(at times) and love(at all times) I would never have had a chance to start, let alone finish, this work.

Contents

Figures

Tables.

Chapter 1: Introduction to magnetism.

1.1 Magnetism and Units.

Although magnetism is an old branch of science, having not only been investigated around 3000 years ago in Greece, but also having technological importance from the time the first piece of magnetite was suspended to show the direction of North, it is still an area where much research is still being done to fully understand the processes involved, both on an atomic and macroscopic level.

Magnetism is now generally accepted to be the additional interaction between two charges both moving with respect to the observer, above the electostatic force experienced when the charges are at rest. This force can be considered in terms of *magnetic fields* originating from moving charges, which can be superimposed and affect the motion of any other moving charge. This field was long considered to be due to *magnetic poles*, the existence of which was postulated long before the existence of electric charges. The search for the magnetic monopole has however yet to yield proof of the existence of such a pole.

Maxwell's equations, which described the interaction of electric and magnetic forces are now over 120 years old, and form a solid basis for a general understanding of magnetism. However understanding of the interactions which take place within solids, although completely described by Maxwell, are not simply analysable.

The development of quantum mechanics at the beginning of the century shed new light on the interactions which give rise to co-operative magnetism (see section 1.2.3).

Throughout this thesis S.I. units are used, and the Sommerfeld(1948) convention is used. That is the induction **B** in a medium is given as:

$$\mathbf{B} = \mu_o (\mathbf{H} + \mathbf{M}) = \mathbf{B}_o + \mu_o \mathbf{M} \ (T) \qquad (1.1)$$

the volume susceptibility is given as the ratio of the magnetization over the field which produced it:

$$\mathbf{k} = \mathbf{M} / \mathbf{B}_o \ (J\ T^{-2}\ m^{-3}) \qquad (1.2)$$

and the mass susceptibility as:

$$\chi = \mathbf{k} / \rho \ (J\ T^{-2}\ kg^{-1}) \qquad (1.3)$$

both of which are tensor properties.

The torque on a dipole in a field is given as: **Torque = moment × field**, i.e.:

$$\mathbf{T} = V\ \mathbf{M} \times \mathbf{B}_o \ (N\ m) \qquad (1.4)$$

1.2. Solid State Micro-Magnetism

1.2.1. Diamagnetism

Diamagnetism is the reduction of the induced field below that of free space by the presence of a medium. The diamagnetic susceptibility is generally small ($\mathbf{k} \simeq -10\ J\ T^{-2}\ m^{-3}$) and independent of temperature. Although all materials can be said to exhibit diamagnetism, it is swamped in materials which are paramagnetic or exhibit co-operative magnetism. This means that

it is observed best in inert gasses, or other materials with a large gap between occupied full orbitals and empty ones.

It can be considered to be due to precession of the electronic orbitals around the nuclei. This can be calculated classically using Larmor precession theory (Kittel 1976) or quantum mechanically (van Vleck 1932), the result in both cases is the same:

$$k = - \frac{N Z e^2 \langle r^2 \rangle}{6 m} \qquad (1.5)$$

where N is the number of atoms per unit vol, Z the number of electrons per atom, e the electron charge, $\langle r^2 \rangle$ the mean square distance of an electron from the nucleus and m the mass of an electron. As the factors in (1.5) are all positive ,the diamagnetic susceptibility will always be negative.

A special form of diamagnetism is superconductivity where the field is totally expelled from a material, i.e $\mathbf{M} = -\mathbf{H}$.

1.2.2. Paramagnetism.

For low fields or high tempratures many materials have been observed to obey the Curie Law:

$$k = C / T \qquad (1.6)$$

where C is known as the Curie constant, or the Curie-Weiss Law:

$$k = C / (T - \theta_p) \qquad (1.7)$$

where θ_p is the paramagnetic Curie temperature, and may be either positive or negative.

The Curie law can be deduced either classically (Langevin 1905) or quantum mechanically (Brillouin 1931). In the classical treatment the material is considered to consist of an array of non-interacting dipoles, which are aligned by the external field, but are randomized by the thermal energy. Application of Maxwell-Boltzmann statistics gives the result:

$$M = M_o \left(\coth \left(\mu B_o / kT \right) - 1 / \left(\mu B_o / kT \right) \right)$$
$$= M_o \mathcal{L} \left(\mu B_o / kT \right) \qquad (1.8)$$

where μ is the moment of each dipole, M_o is the total moment of all the dipoles if aligned, T is the temperature, k is Boltzmann's constant and $\mathcal{L}(x)$ is known as the Langevin function.
For B_o/T small this approximates to the Curie Law form (1.6):

$$M = M_o \mu B_o / 3 kT \qquad (1.9a)$$

or
$$k = M_o \mu / 3 k T \qquad (1.9b)$$

The classical result (1.8) has been shown to apply to superparamagnetism, where small ferromagnetic particles have an effectively continuous range of dipole orientations.
Weiss (1907) suggested inclusion of interaction between dipoles, which could be considered as an internal field within the material, known as the Weiss field. This leads to a result in the form of the Curie-Weiss Law(1.7).

A quantum mechanical treatment of the same phenomena has to consider quantization of the component of total atomic angular momentum, J along the aligning field direction. The loss of degeneracy imposed by the external field results in a splitting of the energy levels from the free atom model.

The splitting for a field of about 1 Tesla is usually much smaller than both the thermal energy at room temperature and the gap between states of different J.

Brillouin's treatment yields the expression:

$$M = N J g \mu_b B(J, y) \qquad (1.10)$$

where g is the Landé g factor:

$$g = 1 + \frac{J(J + 1) + S(S + 1) - L(L + 1)}{2 J(J + 1)} \qquad (1.11)$$

μ_b is the Bohr magnetron:

$$\mu_b = e h / 4 \pi m \qquad (1.12)$$

(h = Planck's constant) and B(J, y) is known as the Brillouin function:

$$B(J, y) = (1 + 1/2J)\coth[y(1 + 1/2J)] - (1/2J)\coth(y/2J) \qquad (1.13)$$

For small fields or high temperatures (1.10) approximates to:

$$M = \frac{N g^2 \mu_b^2 J(J+1) B_0}{3 kT} \qquad (1.14)$$

which has the Curie Law form. At low temperatures and high field the full shape of the Brillouin function has been observed (Henry 1952).As with the classical derivation, the inclusion of interactions between atoms is required to produce an equation of the form of (1.7).

1.2.3 Cooperative magnetism.

1.2.3.1 Exchange Interaction.

As discussed in section 1.2.2 inclusion of an internal field or magnetic interactions between atoms can give rise to the Curie-Weiss law. It can also give a description of *spontaneous magnetization* in ferromagnetic materials where the moments can be considered to arise from localized moments. However the size of the internal field required to produce a Curie temperature of the same magnitude as those found in ferromagnets is about 1000 times the value of B observed in actual ferromagnets, in other words the magnetic interaction between atoms is about 1000 times smaller than is required by the theory.

It appears therefore that the interaction must originate elsewhere and Heisenberg(1928) was the first to give an explanation in terms of the so called exchange interaction. This originates in the overlapping of regions of charge density when two wave functions are considered together. As electrons are fermions this means that the combined wave function of two electrons must always be anti-symmetric. This leads to an effective coupling between the spatial and spin wave functions, if one is symmetric the other must be anti-symmetric and vice-versa. The Coulomb interaction between electrons leads to an energy difference between symmetric and non-symmetric spatial wave functions due to their relative compactness and this in turn leads to the so called exchange interaction between the spins.

The energy of this exchange interaction is given by the Heisenberg expression,

$$\Delta E = -2 \Sigma J_{ij} S_i.S_j \qquad (1.15)$$

where J_{ij} is the exchange constant, given by

$$J_{ij} = \langle i\ j \mid H \mid j\ i \rangle \qquad (1.16)$$

where H is the Hamiltonian of the electron-electron and electron-nuclei interaction.

1.2.3.2 Ferromagnetism

A ferromagnetic material is usually divided up into volumes which are spontaneously magnetized. The magnitude of this magnetization decreases as the Curie temperature is approached. In a ferromagnet at T=0K all the spins are aligned in one direction (see fig 1.1). The moment can arise either from localized electrons, as in the rare earth metals where 4f electrons are unpaired, or conduction electrons, as in the transition elements where the 3d electrons are delocalized. Both types of ferromagnetism can be treated as though there was a large internal field, which produces an energy difference between states of differing spin(e.g. Crangle 1977).

As discussed above (1.2.3.1) the fields required to produce a large enough splitting could not arise purely from magnetic interaction, and the exchange interaction is thought to be the origin of this energy gap. For ferromagnetic ordering the sign of J_{ij} (1.16) must be positive.

Calculations of J_{ij} in particular cases show that this also is often not large enough to account for the ferromagnetic ordering. In these cases a more complex interaction than direct exchange interaction is involved. In insulating ferromagnets *superexchange*(Kramer 1934) is involved. This is the interaction between two local moments separated by a non magnetic ion. Spins on the magnetic ions are coupled to the non-magnetic ion. This gives a preferredspatial distribution of spin on the intervening non-magnetic ion which, although not giving rise to a net moment on the ion itself, interacts

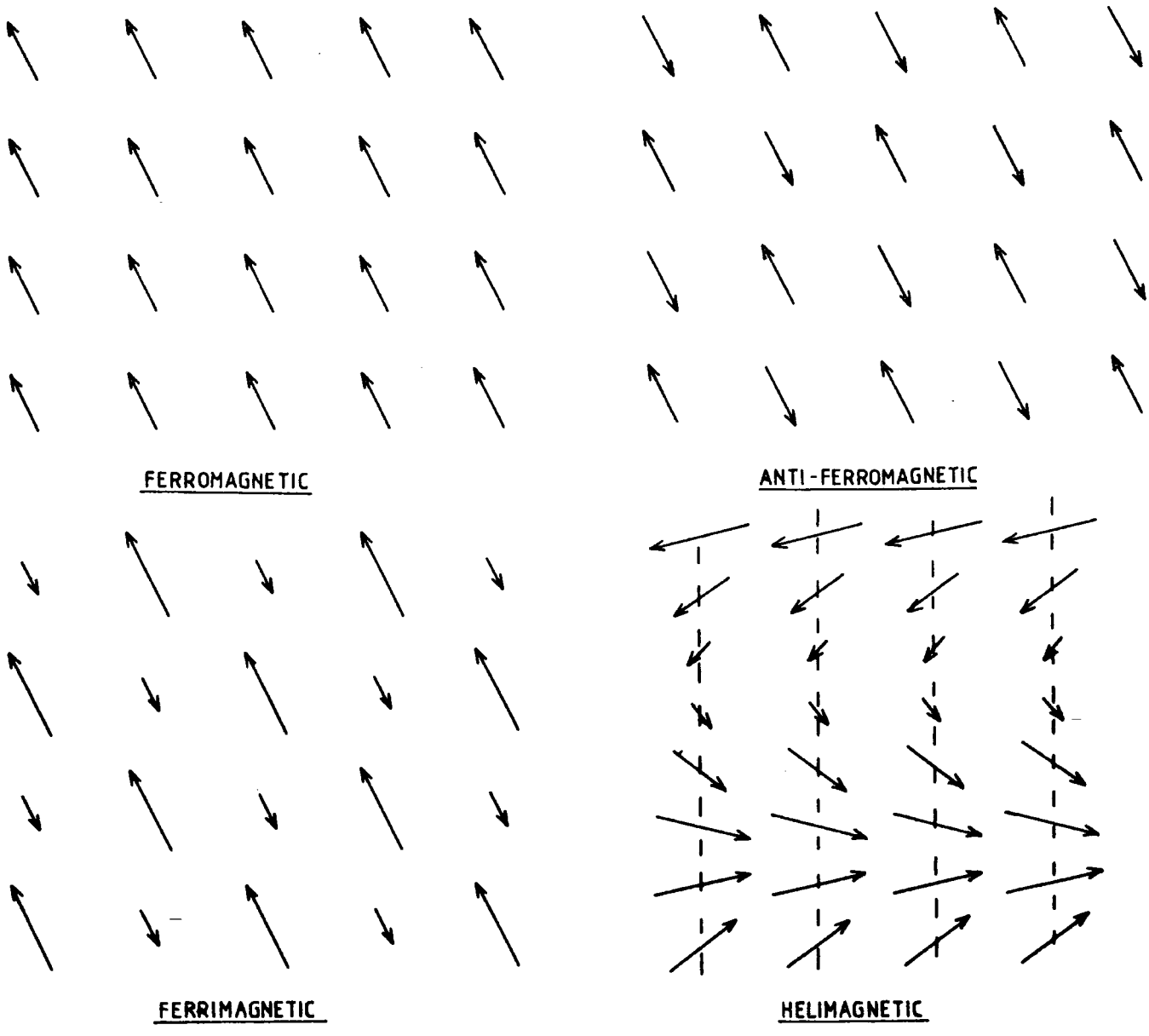


Fig 1.1 Different forms of magnetic ordering.

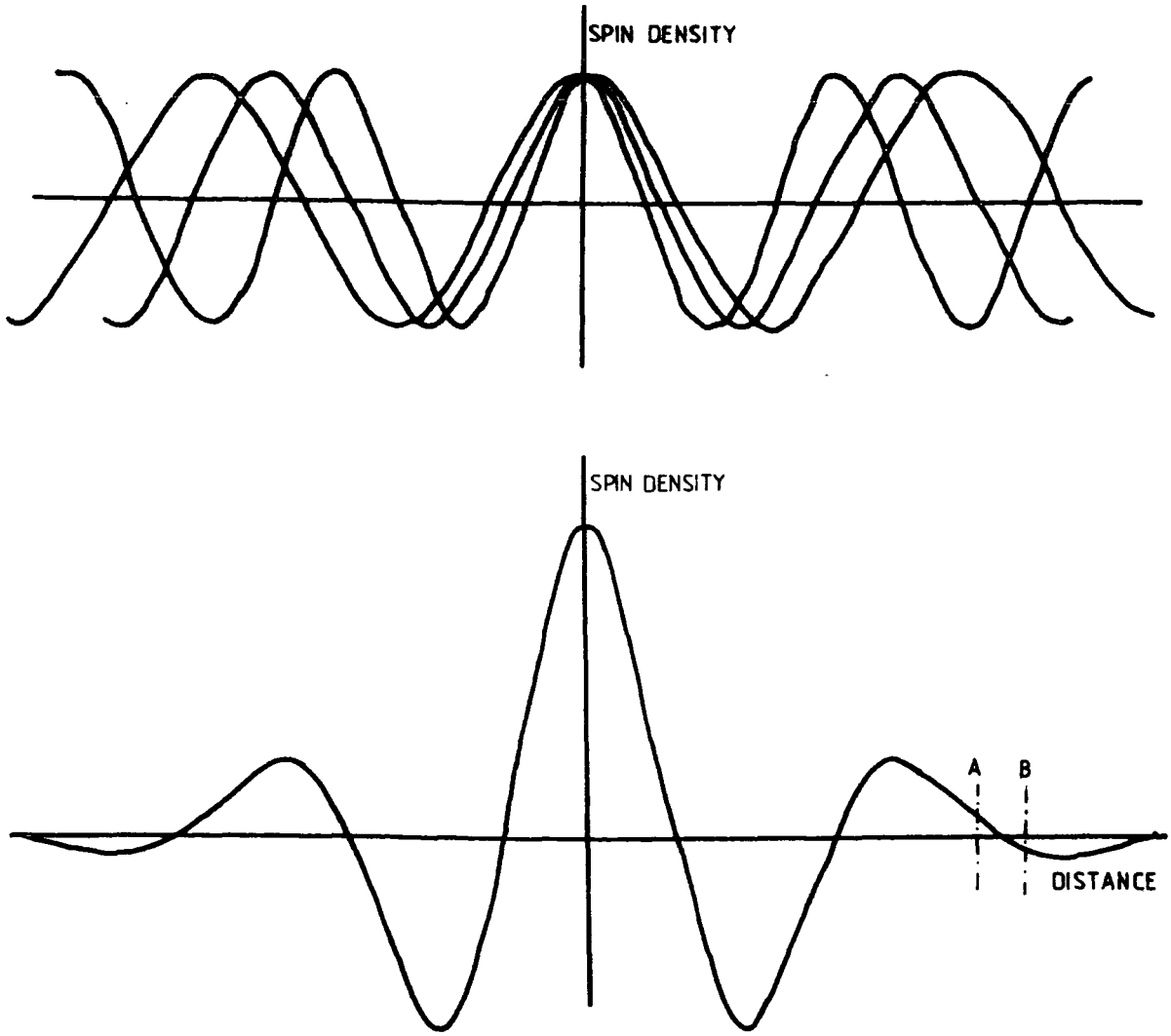
with the magnetic ion on the other side to produce a prefered spin orientation. In conducting magnets where unpaired states exist below the Fermi level which are localized within the ions and little direct exchange is possible the RKKY process (after Ruderman and Kittel 1954, Kasuya 1956 and Yosida 1957) is the accepted explanation. In this model the coupling with conduction electrons near the Fermi energy produces a preferred positioning of the wave functions of electrons with one spin direction around an ion, and a equally 'unpreferred' positioning of electron with opposite spin. The addition of these wave functions leads to a spatial spin distribution spreading out from the magnetic ion (see Fig 1.2). This type of interaction is relatively long range and highly dependent on the separation of the magnetic ions. The resultant indirect coupling between the localized moments may give rise to ferromagnetism.

1.2.3.3 Anti-ferromagnetism.

If the sign of the Heisenberg exchange integral (1.16) is negative then coupling between adjacent ions gives rise to alternate spin directions. (see Fig 1.1). The result is no bulk magnetisation. Alternatively there may be strong positive coupling within sub lattices, but weak coupling between sublattices, leading to two sub-lattices within a crystal with opposing moments. Anti-ferromagnetic ordering was proposed by Néel(1936) and neutron diffraction confirmed its existence in 1950. Anti-ferromagnetism may revert to ferromagnetism at high field, or low temperature. The temperature at which the anti-ferromagnetic ordering ceases is known as the Néel temperature.

1.2.3.4. Ferrimagnetism and Helimagnetism.

Anti-ferromagnetism can be seen as a special case of ferrimagnetism when the moments on the sub-lattices are equal. If they are not then this gives



a) Conduction electrons near the Fermi surface are positioned at preferred position around a magnetic ion

b) The wave functions in a) summed to give the spin density distribution around a magnetic ion.

Fig 1.2 Origin of the RKKY interaction.

(after Crangle 1977)

rise to a total resultant magnetisation, equal to the difference in the moments. The dependence of spontaneous magnetization on temperature can take a variety of forms depending on the Curie temperatures of the two sub lattices, and may have compensation points where the magnetic moments on the two sites are equal and cancel (see fig 1.3).

The coupling between atoms may also result in more complex magnetic structures where moments align at an angle to each other. The result may be an overall magnetic moment or not. An example is Helimagnetism (see fig 1.1) where the moments lie in a helix. Many other complex forms of ordering have been observed, and these tend to occur in rare-earth alloys. This is thought to be due to the influence of separation on the RKKY exchange interaction (see sect 1.2.3.2)

1.2.4. Magnetocrystalline Anisotropy.

In ferro- and ferrimagnetic material the directional dependence of the orbital overlap gives rise to preferred directions along which the moments will tend to align. These minima in the angular dependence of magnetisation are known as easy directions. To force the moments to another direction requires additional energy. The angular distribution of this energy has the same symmetry as the crystal structure and is generally described by phenomenological constants based on power series of the major spherical harmonics. These series are dependent on the crystallographic point group(Doring 1958). Analogous expressions are also possible using a different basis. Fourier or harmonic analysis is possibly a better model, allowing better comparison with theoretical predictions(Birss and Keeler 1974) and Legendre polynomials are another possibility. Conventions are still being suggested (Volkov 1981) and as there is no consensus as to the best, the expressions of Doring are used in this thesis.

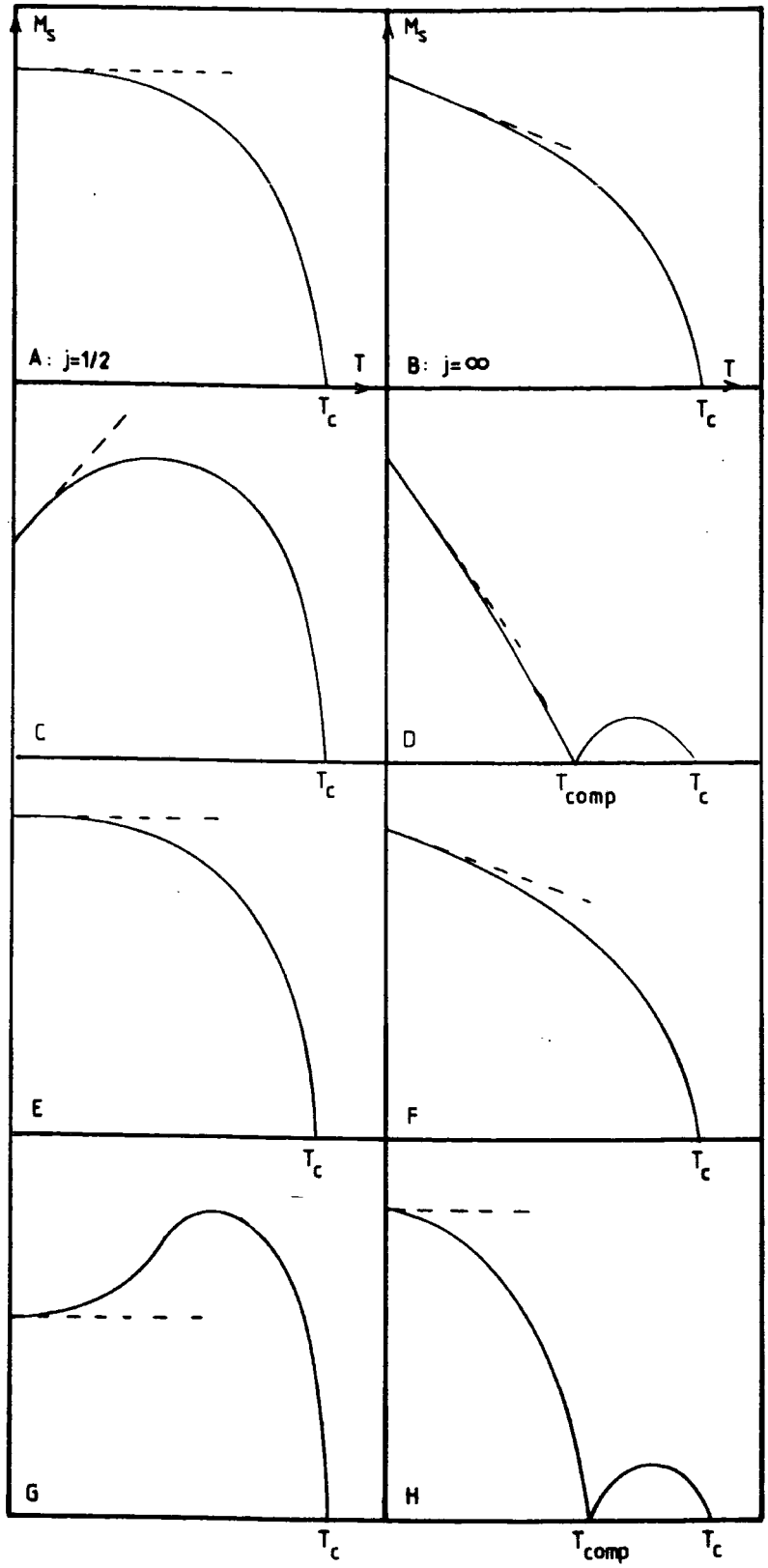


Fig 1.3 Magnetization in Ferro(A,B) and Ferri(C-H) magnets.

Various models have been proposed for the origin of the magnetocrystalline anisotropy which have been shown to apply to a greater or lesser extent to different materials.

1.2.4.1 Single Ion Model.

In this model, otherwise known as the crystal field model, the crystalline structure forces a prefered orientation on the electronic charge distribution around a single magnetic ion. As this charge distribution is governed by the spatial component of the wave function, this in turn is coupled to the crystalline lattice. The spin-orbit interaction discussed in section 1.2.3.1 means that the spin, that is the moment of the ion, will be coupled to the crystal lattice.

This model is probably dominant in materials where the moments are localised and the ions are well separated, for example in magnetic insulators and in rare earths, where the 4f orbitals are deep within the ion and therefore overlap very little.

1.2.4.2 Two Ion Model.

In the same way that the dipole coupling between moments is anisotropic, the exchange interaction was also shown to be anisotropic by van Vleck(1937). This anisotropy can be compared with the classic dipole-dipole interaction, and is found to drop off much more rapidly with separation of the dipoles. This model can be important in 4f magnetism as a contribution to the overall anisotropy.

1.2.4.3 Conduction electron models.

In a material where the magnetisation arises from conduction electrons, then the coupling of the wavefunctions with the crystal have to be considered. An analogy to the single ion model in localized magnetization is

possible by considering the influence of the crystal potential on the free-electron wave function. This potential gives rise to a band structure which is orientation dependent. The spin-orbit coupling for these Bloch states then introduces a directional dependence to the magnetization.
Models of this type have been applied qualitatively to 3d magnetisation (e.g.Mori et al 1974).

1.2.5. Magnetostriction

The change in direction of magnetisation within a material causes a change in crystal lattice constants, this effect is known as magnetostriction. It is usually described in terms of constants related to direction cosines with respect to the major crystallographic axes. This change in lattice constant and the elastic constants of the material contribute to the magnetoelastic energy.

1.3 Solid State Macro-Magnetism.

1.3.1 B-H loop

Most magnetic materials possess no overall moment until placed in a field. As the field is increased (OA in Fig 1.4) the moment increases until it reaches a saturated value (M_S), further increases in the applied field result in an increase in the induction but not in moment. On decreasing the field (AB in fig 1.4) the moment will also decrease but when the field is zero a positive moment will remain, called the remnance (M_R), or remanent induction (B_R). On increasing the applied field in an opposite direction (BC in Fig 1.4) the induction will be reduced to zero at a applied field H_{CB} and the moment at a field H_{CM}, known as the coercive field or coercivity. Further increasing the applied field (CD in Fig 1.4) results in the moment again increasing to the saturated moment. Reducing the field and increasing it again in the initial direction (DEFA in Fig 1.4) returns the material to it's initially magnetized state. The outside loop (ABCDEFA) is known as the hysteresis loop. This loop is symmetrical about the origin and usually highly reproducible after a few cycles. Each time the loop is traversed an amount of energy proportional to the area of the loop is absorbed by the material.

Materials in which the loop area is small are called soft. The three important magnetic characteristics of soft materials are their susceptibility, saturation magnetisation and the area of the hysteresis loop, either given by quoting the remanence or the coercivity.

Materials in which the loop area is large are called hard. The two important magnetic characteristics of a hard material are the coercivity and the 'energy product', $(BH)_{MAX}$ (fig 1.4B). The values of B and H corresponding to $(BH)_{MAX}$ define the optimum working point for a permanent magnet.

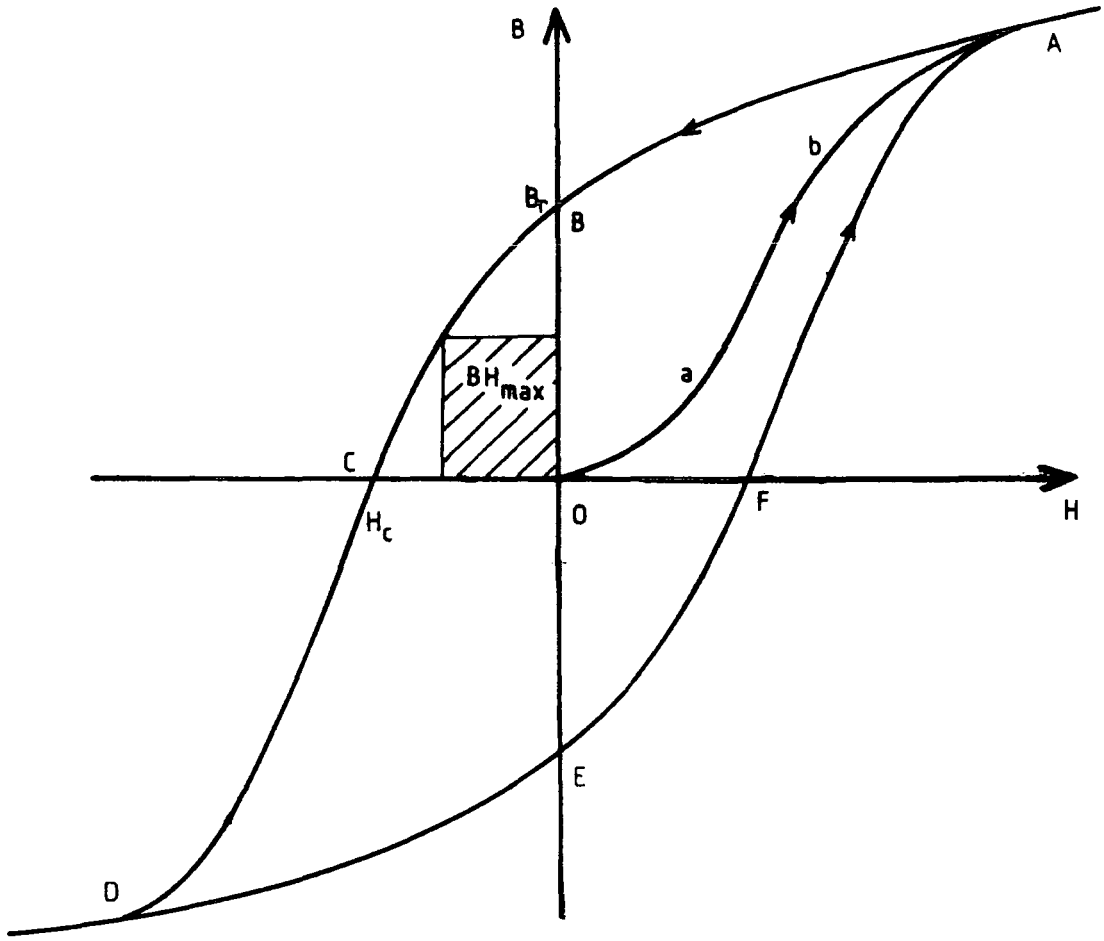


Fig 1.4 Typical Hysteresis loop.

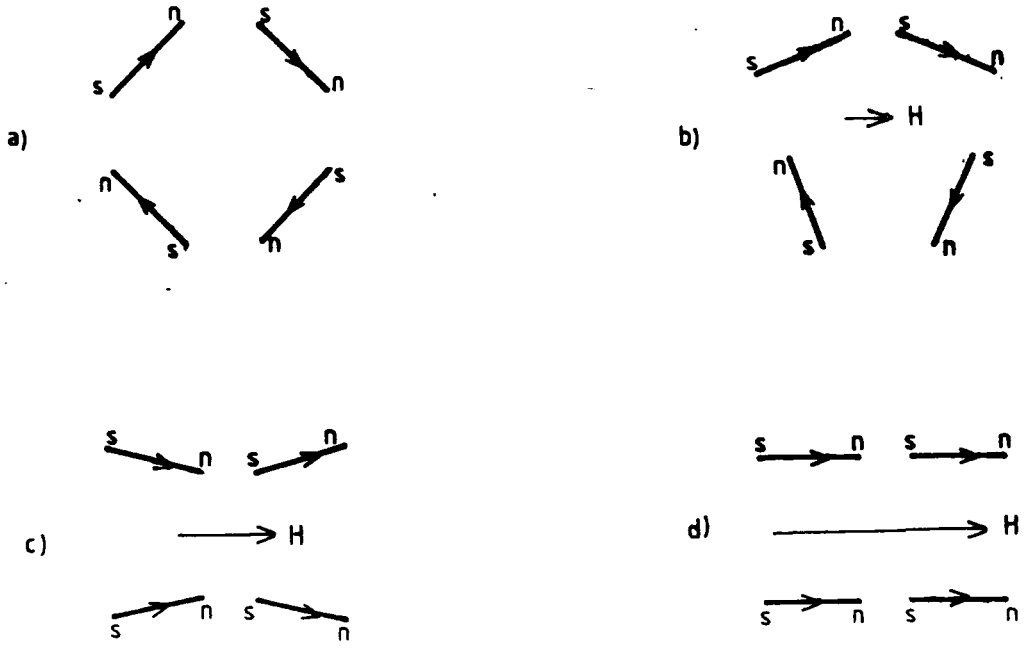


Fig 1.5 Movement of four moments under an increasing field.

Other properties such as temperature and chemical stability, mechanical hardness, etc. obviously play a part in full characterisation and comparison of magnetic materials for commercial purposes.

1.3.2. Domains

The shape of the hysteresis loop provides a puzzle. It has long been known that magnetic moments were sub microscopic in origin. In this case why do they not all align to produce a saturated moment when a small external field is applied to the sample? Early investigators proposed a form of molecular friction which prevented moments from turning readily. Ewing(1890) showed qualitively that the shape of the hysteresis loop could be explained without introducing the idea of friction, in terms of the effect of moments on their neighbours. For example a group of four moments would react to increasing field as shown in Fig 1.5. In the unmagnetised state (a) the N and S poles are close to each other to produce no external field. As the field is increased the moments first rotate so as to retain the structure of the group (b), this corresponds to the initial reversible part of the magnetization curve (Oa in Fig 1.4). On further increasing H two of the moments swing round to face the opposite direction (c), giving a large increase in M for small H (ab in Fig 1.4). This part of the curve is not reversible and, in the model, reducing the field does not initially let the moments swing back. Finally increasing the field further pulls the moments completely into line (bA in Fig 1.4). In practise a much larger number of moments is involved and the situation a lot more complex.

Weiss(1907) suggested that ferromagnetic materials are divided into regions which are each saturated by the internal field within the material and have a uniform magnetisation within each region. These regions are called domains. In the demagnetised state the domains are so arranged to give zero

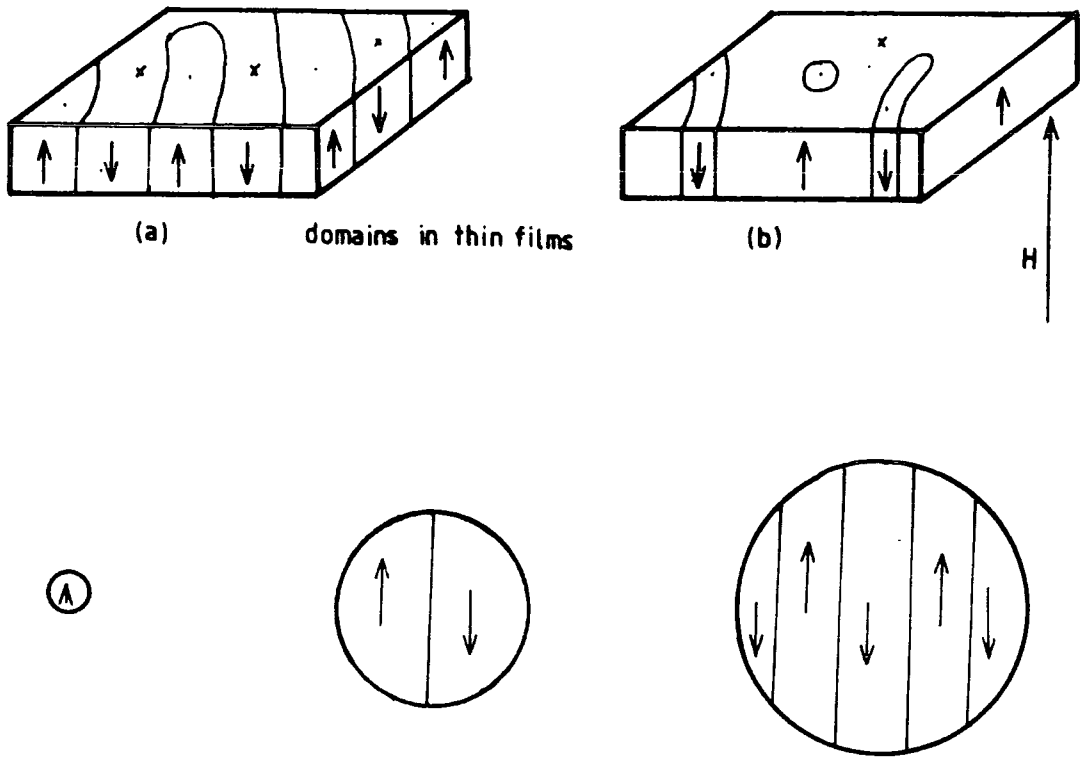


Fig 1.6 Examples of domain patterns.

net magnetization. This has since been validated by observation of the domains (e.g. Williams, Bozorth and Shockley 1949).

Lifshitz and Landau(1935) considered domains in terms of minimisation of magnetostatic energy and Néel(1944) developed the idea further.

The domain structure can best be understood in terms of the total magnetic energy of the material. This can be expressed in terms of the Gibbs function for the material:

$$G_T = G_E + G_A + G_M + G_\sigma + G_\omega + G_B + G_o \qquad (1.17)$$

where G_E : Exchange energy;

G_A : Anisotropy energy;

G_M : Internal magnetostatic energy;

G_σ : Magnetoelastic energy;

G_ω : Domain wall energy;

G_B : External Magnetostatic energy;

G_o : Other energy terms.

The first four terms in the equation (1.17) are neglecting the material within the domain wall which is considered separately as the fourth term. Solving this equation to find the minimum of the Gibbs free energy is not usually possible in any but the very simplest case, but it is possible to use the equation to show that an observed or postulated domain structure is a local minimum and to compare it with other structures. In this way it can be shown that domain formation is often energetically favoured. The equation also helps to understand the type of structure expected in a qualitqtive manner.

In a ferromagnet the exchange term will tend to align the moments within a domain, and produce a uniform magnetisation. The anisotropy term means that, except in high field, this magnetisation will tend to lie along an

easy direction. A single domain can be thought of as a volume with a free pole density at the surface. The magnetostatic term is large if there is any surface or domain wall where there is a large free pole density. This leads to closure domain structures (see Fig 1.6d) and, as the magnetisation is restricted to a limited number of easy directions, allows only certain domain wall orientations.

The magnetoelastic term has an influence on the type of domain wall in highly magnetostrictive materials. The origin of the domain wall term is discussed further in the next section. It is considered to have an energy per unit area. The ratio of the domain wall term to the magnetostatic term in general influences the size of the domains. The external magnetostatic term is the interaction with an applied field, and accounts for the domain wall movements.

1.3.2.1. Domain Wall Energy

The structure of a domain wall can be very complex, especially in materials with more than one type of magnetic atom and more than one lattice site.

Let us consider, however, the simplest model involving only one type of magnetic ion in one crystal site in a magnetically uniaxial crystal (see Fig 1.7). The change in magnetisation direction generally takes place gradually through the *domain wall thickness* with the spins rotating within the plane of the wall as required to minimise magnetostatic energy. This thickness is determined by the balance between the anisotropy energy and the exchange energy. If two spins are misaligned by an angle $\delta\phi$ then the increase in exchange energy, following from (1.15), is

$$\delta G_E = J S^2 \delta\phi^2 \qquad (1.18)$$

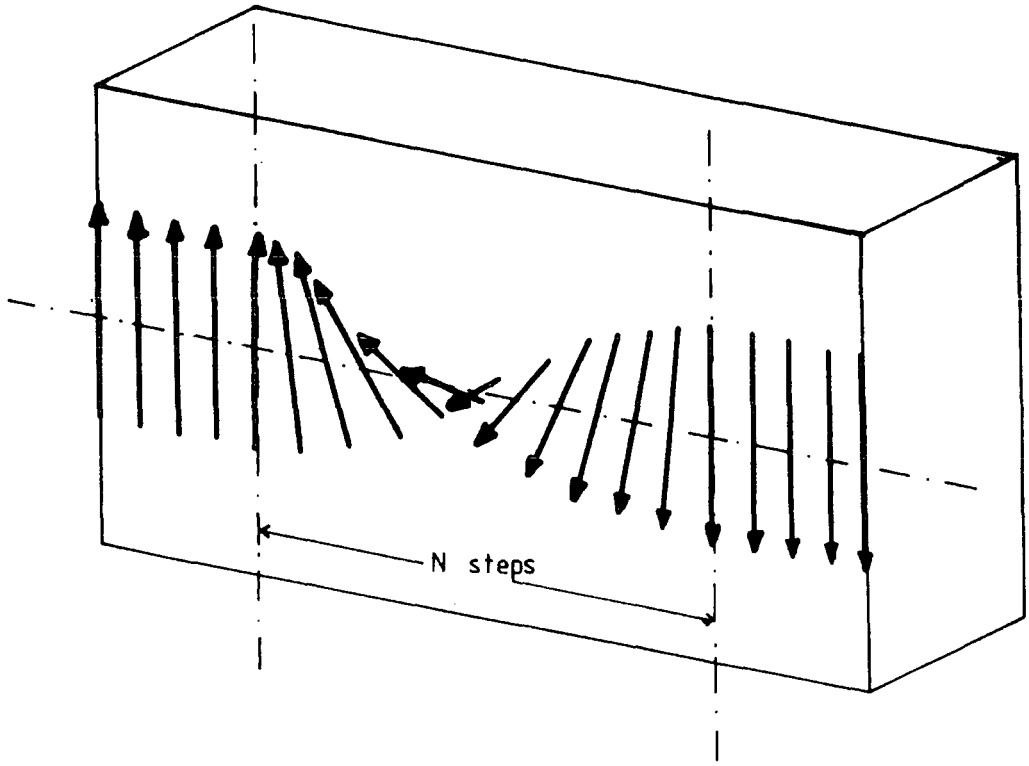


Fig 1.7 Domain Boundary Wall (Kittel 1949)

If we assume that $\delta\phi$ is a constant through the wall, and that the rotation takes place over N steps the total increase in exchange energy is given by

$$\Delta G_E = N\ J\ S^2\ \delta\phi^2 = J\ S^2\ \pi^2\ /\ N \qquad (1.19)$$

as the total rotation in a uniaxial wall must be π.

Considering only the first term in the anisotropy energy (A6.1), the total anisotropy energy in the wall is given by

$$\Delta G_A = \sum_{n=0}^{N} k_1 \sin^2(n*\pi/N) \qquad (1.20)$$

where k_1 is the microscopic anisotropy constant. If the wall is thick enough to be considered uniform then this can be approximated to an integral and solved to give:

$$\Delta G_A = N\ k_1\ /\ 2 \qquad (1.21)$$

Summing (1.19) and (1.21) and minimising the free energy gives:

$$N = S \pi (2 J / k_1)^{1/2} \qquad (1.22a)$$

and

$$\Delta G = S \pi (2 J k_1)^{1/2} \qquad (1.22b)$$

where ΔG is the increase in Gibbs free energy due to the wall. It is interesting to note that the anisotropy and exchange contributions to the total domain wall energy are the same.

1.3.2.2. Domain Structure in Uniaxial Systems.

In uniaxial systems there are only two allowed orientations for the magnetisation within a domain. In a thin film with the easy axis perpendicular to the film the zero field domain structure is a series of band domains of equal thickness(1.6a). Minimising the surface free pole magnetostatic energy and domain wall energy(γ) gives a relationship between the thickness of the bands (W) and the film thickness(D) (Kittel 1976):

$$W = (D \gamma / 0.135 \mu_o M_s^2)^{1/2} \qquad (1.23)$$

Under an applied field the bands with magnetisation along the direction of the field broaden at the expense of the opposing domains, which thin and break leaving *bubble domains* (1.6b).

In the case of a 'thick' film or a slab this thin film model, with only flat 180° walls is not valid as the energy is reduced by sloped walls which allow the free poles to be distributed through the sample. This model (Fig 1.6e) gives a domain width at the surface independent of the thickness of the block(Bodenberger and Hubert 1977):

$$W_1 = \beta 4 \pi \gamma 10^7 / M_s^2 \qquad (1.24)$$

The factor β is a 'correction factor' which depends on the geometric structure and was found to equal 0.31 ± 0.02 for $SmCo_5$.

1.3.2.3. Domain Movement

The form of the hysteresis curve is determined by the domain structure within a material. In general the changes in magnetisation can be understood as resulting from either the movement of domain walls so that one domain grows at the expense of another, the creation or destruction of domains, or the rotation of the magnetisation of a domain. The latter is generally important only at high fields, and the first two determine the magnetic properties of the material for most purposes.

Domain walls move in steps rather than continuously, although the size of the steps are so small that sensitive readings are required to separate them (Barkhausen 1919). Movement of one domain wall will cause a change in the environment of the surounding domains and may cause a 'domino' effect, so the measurement of Barkhausen jumps places a maximum on the amount of material which is reversed in one wall movement. The domain wall may stop at a number of different types of 'pinning sites', for example small holes, inclusions, dislocations, grain boundaries etc.

Creation of a domain requires overcoming an energy barrier, as a small domain will have a large domain wall energy for the reduction in magnetostatic energy, conversly a domain will disappear rapidly when it gets below a certain critical size.

1.3.2.4. Domain Observation

A number of techniques have now been developed for observing domains, but until comparatively recently the method suggested by Bitter(1931) of allowing small magnetic particles to move freely over the surface and observe where they came to rest was the only method commonly used. In the standard method a magnetic colloid, or *ferrofluid* is spread over the surface. Where domain walls reach the surface the leaked magnetic field is large and the particles are attracted to these domain walls. If the

attraction is large enough to overcome the Brownian scattering of the particles a pattern develops. The pattern can be directly observed, or the fluid dried off and the pattern observed under a scanning electron microscope.

An extension to this wet Bitter method is the dry Bitter method (Hutchinson et al 1965), where a ferromagnetic material is evaporated onto the sample surface in the presence on a inert gas. The particles produced by the evaporation depend primarily on the gas pressure, and the patterns so produced can be examined in the same way as for the wet Bitter method. Recently(Szewczyk et al 1983) particles for the dry Bitter method have been produced by cooling a mixture of Oxygen and Helium. The solid oxygen produced is paramagnetic and Bitter patterns are observed.

If the particles in the ferrofluid are small enough and the field gradients above the surface not too large, then the wet Bitter method may be used for dynamic observation of domain wall movements. If the particles are to large then the fluid becomes unstable in the field gradients and the colloid settles out at the domain wall boundary, not moving with the domain wall.

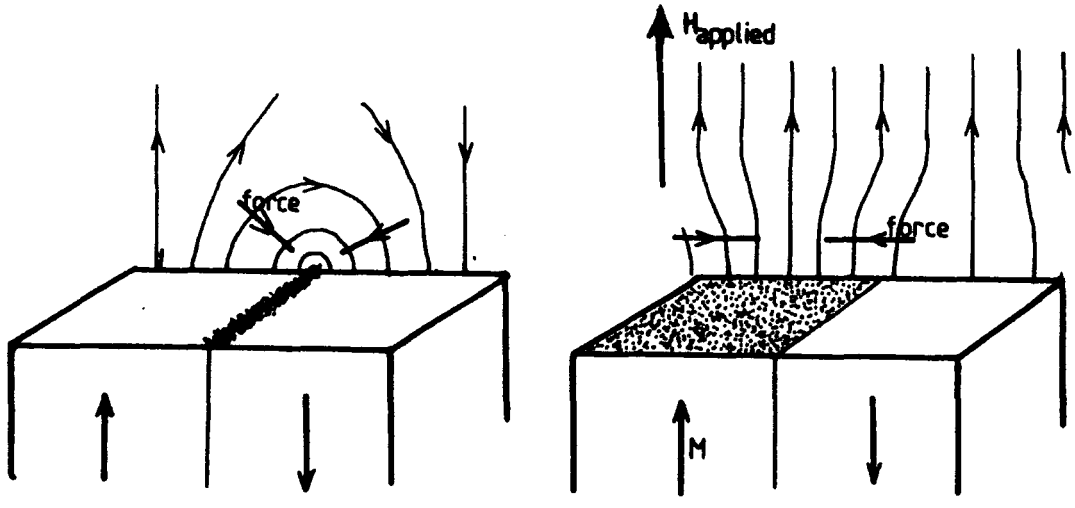


Fig 1.8 Bitter pattern formation.

With an applied field domain contrast can be obtained in the Bitter patterns, as the field above one domain will differ in magnitude from that above another(Fig 1.8b).

Direct optical methods using polarised light can also be used for domain observation, a beam of light reflecting from the surface of a material will have its polarisation changed dependent on the relationship between the magnetization and the direction of polarisation of the light. This is termed the Kerr effect. Unlike the Bitter technique where generally domain wall contrast is used, in the Kerr effect only domain contrast is obtained.

In transparent magnetic materials, e.g.garnets, a similar method can be used to observe domains, as the polarisation of the transmitted light will be affected in the same way although in this case by the internal field and not the stray field. This is called the Faraday effect.

Magnetostrictive effects can also be used to observe domain patterns. In very highly magnetostrictive materials the pattern can be observed optically by the wrinkling of the surface. In other materials X-ray topography can be used to image the domain pattern(e.g. Tanner 1976), relying on the sensitivity of topography to detect the small changes in lattice constant.

Electron Microscopy can also be used directly, as electrons have a magnetic moment. Transmission electron microscopy gives direct observation of domains, the non-scattered electrons being deflected by the internal magnetic fields in the materials. Scanning electron microscopy can also be used, where either the depth of penetration of an angled beam depends on the magnetisation of the domain as the electrons are bent in the field, giving contrast, or the effect of the stray field on the secondary electron emission is used to image the domain walls.

1.3.3. Bulk Anisotropy

Ideally a single crystal sphere should be used for measuring anisotropy. However this is not always possible, and allowances should be made for poor alignment of powdered or other multicrystalline samples and anisotropy due to the shape of a sample.

1.3.3.1 Shape Anisotropy.

A magnetised material in an external field will itself produce a field which should be considered when the field applied to the magnet is calculated.
If the object is not spherical then the amount of magnetostatic energy in this demagnetising field will depend in the direction of magnetisation. This shape anisotropy can be calculated either by integrating the total energy of the magnetostatic field over all space or integrating the energy released when the object is assembled, already magnetised, from infinity.

Chapter 2: Introduction to rare earths.

2.1 Introduction.

Rare Earth metals are so called due to the resemblance of their oxides to other metal oxides, known as the 'common earths'. The group, which appears in the third column of the periodic table, consists of yttrium(at. no. 21), scandium(39), lanthanum(57) and the lanthanides(58-71). Despite the name the rare earths are not especially rare, the rare earth content of most rocks being from 1 part in 10^4 to 1 part in 10^7. Exploitable ores yield rare earth mixtures with local variations in the relative abundances of the constituents, scandium being the only rare earth which is found on its own.

2.2 Physical Properties of the Metals.

The group is notable in its uniformity in chemical behaviour, rare earths replacing each other in crystal structures with little strain to the lattice. This is due to the similarity in the external electronic structure of the elements. After lanthanum the extra electrons start filling the 4f shell, which is compact and shielded from forming any chemical bonds by the more extensive 6s and 5d levels. The extra nuclear charge as the series is traversed is only partially shielded by the extra 4f electrons, leading to a reduction in metallic radius, the so called lanthanide contraction (see table 2.1).

With non metallic elements and in solution the rare earths form primarily trivalent compounds, although cerium will donate its single 4f electron to form tetravalent compounds and europium and ytterbium will form divalent compounds due to their nearly half complete and nearly complete 4f shell respectively.

Rare Earth Metal	Crystal struct.	Lattice Constants. (°A) a_o	c_o	Radius/°A CN=12	Metallic Density kgm^{-3}
Sc	hcp(a)	3.3088	5.2680	1.6406	2989
Y	hcp	3.6482	5.7318	1.8012	4469
La	dhcp(c)	3.7740	12.171	1.8791	6146
Ce	fcc(b)	5.1610	-	1.8247	6770
Pr	dhcp	3.6721	11.8326	1.8279	6773
Nd	dhcp	3.6582	11.7996	1.8214	7008
Pm	dhcp	3.65	11.65	1.811	7264
Sm	rhomb(d)	3.6290	26.207	1.804	7520
Eu	bcc	4.5827	-	2.0418	5244
Gd	hcp	3.6336	5.7810	1.8013	7901
Tb	hcp	3.6055	5.6966	1.7833	8230
Dy	hcp	3.5925	5.6501	1.7740	8551
Ho	hcp	3.5778	5.6178	1.7661	8795
Er	hcp	3.5592	5.5850	1.7566	9066
Tm	hcp	3.5375	5.5540	1.7462	9321
Yb	fcc	5.4848	-	1.9392	6966
Lu	hcp	3.5052	5.5494	1.7349	9841

Table 2.1 Crystal structure of rare earth metals

extrapolated to 0K

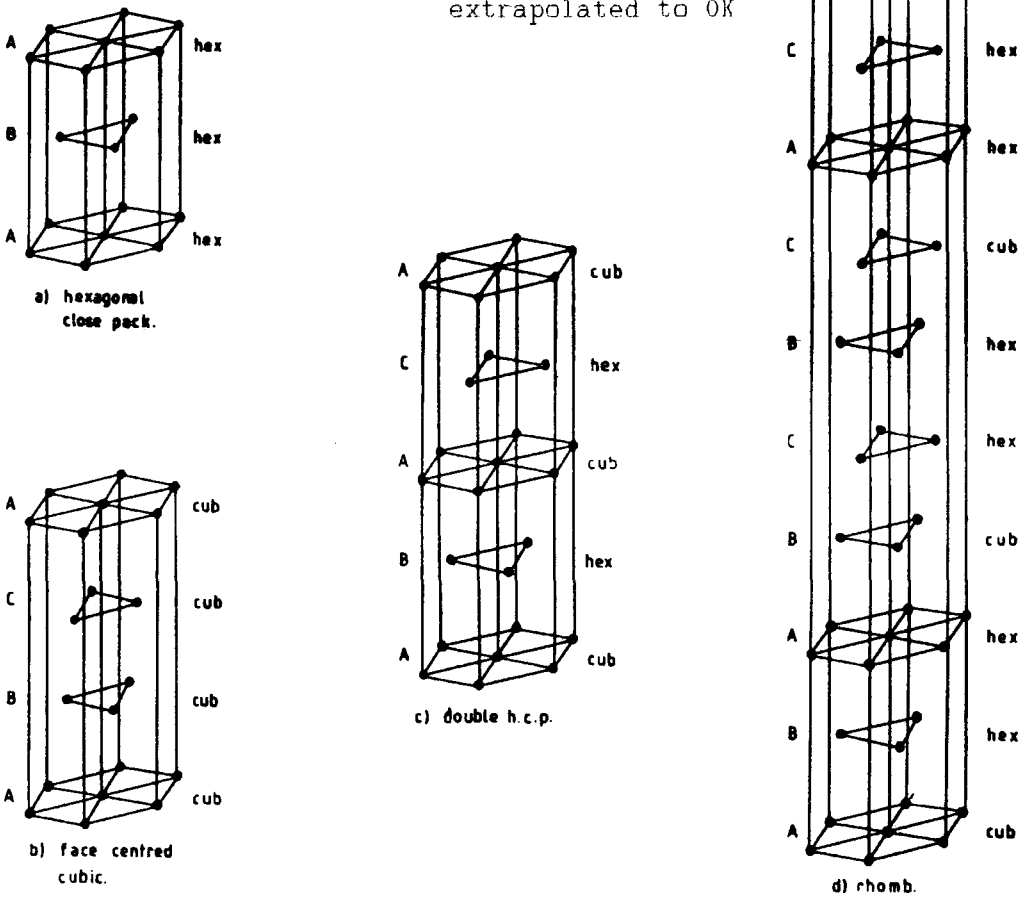


Fig 2.1 Close packed structures in rare earth metals.

2.2.1. Magnetic Structures.

The rare earth metals show a legion of different magnetic structures, due to the long range oscillatory nature of the RKKY interaction and interplay with the crystal field and magnetoelastic energies. These magnetic structures are summarised in Table 2.2 and Fig 2.2. The Curie temperatures are typically an order of magnitude lower than those of the 3d itinerant electron ferromagnets, showing the comparative weakness of the indirect exchange interaction.

The magnetic anisotropy of the rare earth ion is due mainly to the crystal field interaction with the 4f orbitals (see section 1.2.4.1). This is dependent on the multipole moment of the ground state rare earth ion, and Stevens(1952) demonstrated how operator equivalents could be derived from appropriate angular momentum operators with the correct symmetry by means of multiplicative factors. These Stevens Factors give a qualitive tool to understanding the type of magnetic ordering each rare earth ion will have in a given environment.

The light rare earths tend to align antiferromagnetically, the exception being promethium where there is some evidence of ferromagnetism (Koehler et al. 1972). The double hexagonal close-packed structure gives two inequivalent sites, with local hexagonal and cubic symmetry.

In neodymium and praseodymium the moments in the hexagonal layers align ferromagnetically with the moments lying in the layers, the Stevens' α_J Factor is negative, while moments in alternate layers align in opposite directions. At a lower temperature the moments in the cubic layers align antiferromagnetically, the moment being again within the layer.

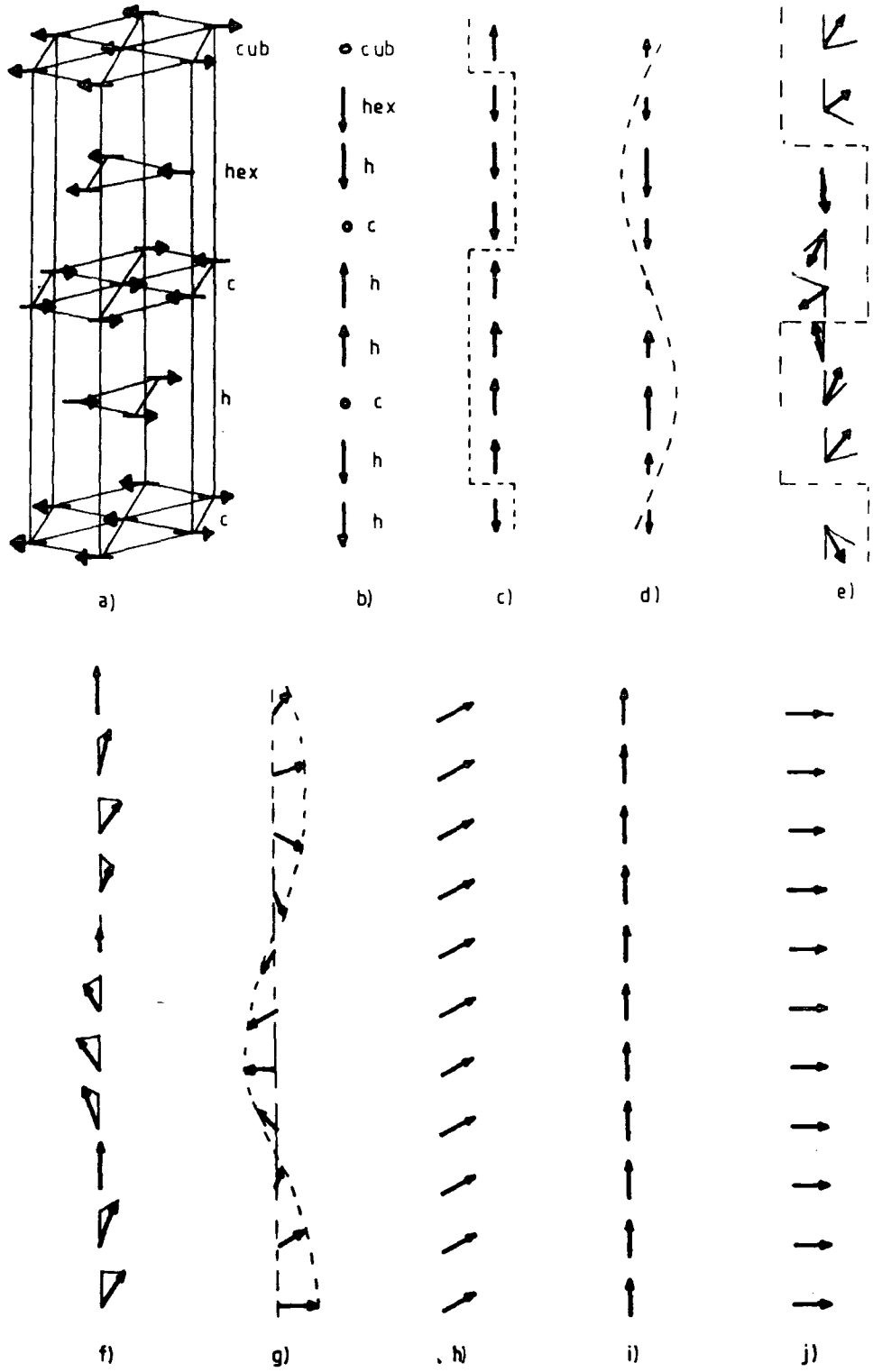
In samarium, with a positive Stevens' α_J factor, the moments align along the c axis. Again moments in the hexagonal layers order ferromagnetically within the layers. The rhombohedral crystal structure gives a stacking sequence HHCHHCHHC... and the pairs of hexagonal layers align together. Moments in alternate pairs oppose each other.. At a lower temperature moments in the cubic layers align antiferromagnetically within the layers, the moments also lying along the c-axis.

Rare Earth Metal	Crystal struct.	Magnetic Ordering Structure*	Néel/ Curie Temp.	Stevens' Factor. α_J	β_J	γ_J
La	dhcp	Supercond.	$T_c \simeq 5K$	0	0	0
Ce	fcc	none		-5.71	63.5	0
"	dhcp	AFM [1]	12.5K	"	"	"
Pr	dhcp	AFM [2] a	25K	-2.10	-7.35	61.0
Nd	dhcp	AFM [3] a	7.5/19K	-0.643	-2.91	-38.0
Pm	dhcp	FM [4]	98K	0.771	4.08	60.8
Sm	rhomb	AFM [4] b	13.8/106K	4.13	25.0	0
Eu	bcc	helix [5] g	90.5K	0	0	0
Gd	hcp	FM c [6] i	293K	0	0	0
"	"	FM cone [6] h	232K	"	"	"
Tb	hcp	helix [7] g	230K	-1.01	1.22	-1.12
"	"	FM b [7] j	≃220K	"	"	"
Dy	hcp	helix [1] g	176K	-0.635	-0.592	1.03
"	"	FM b [1] j	≃87K	"	"	"
Ho	hcp	helix [8] g	133K	-0.222	-0.333	-1.30
"	"	cone [8] f	≃20K	"	"	"
Er	hcp	sin CAM [9] d	80K	0.254	0.444	2.07
"	"	hel/CAM [9] e	52K	"	"	"
"	"	cone [9] f	20K	"	"	"
Tm	hcp	sin CAM' [10] d	56K	1.01	1.63	-5.60
"	"	FI sq CAMc	32-40K	"	"	"
Yb	fcc	?	?	3.17	-17.3	148.0
Lu	hcp			0	0	0

* FM≡ferromagnetic, AFM≡antiFM, FI≡ferrimagnetic, CAM≡c axis modulated. superfixes: letters refer to Fig 2.2, numbers to references below.

References: Cooper(1972), Sinha(1978), [1]Wilkinson et al. 1961, [2]Cable et al.(1964), [3]Johansson et al.(1970). [4]Koehler et al.(1972). [5]Millhouse and McEwan(1973), [6]Cable and Wollan(1968). [7]Koehler(1967). [8]Koehler et al.(1967), [9]Habenschuss et al.(1974), Brun et al. (1970).

Table 2.2 Magnetic Structures and Stevens' factors of rare earth metals.



a) Nd, Pr. b) Sm. c) square CAM, Tm. d) sine CAM, Er, Tm.

e) Helical plus square CAM, Er. f) Helical cone, Ho, Er.

g) Helical, Tb, Dy, Ho. h) FM cone, Gd. i) FM c-axis, Gd.

j) FM basal plane, Tb, Dy.

Fig 2.2 Magnetic Structures in rare earth metals.

Europium has a 2+ ground state, due to the extra stability of the half filled 4f shell. This larger ion then packs with a lower packing fraction and has a body centred cubic crystal structure. It orders magnetically with a helical structure, with the axis of the helix along a cubic axis.
Gadolinium orders ferromagnetically with the easy direction moving from the c axis at higher temperature down towards the basal plane and then back towards the c axis as the temperature decreases. The anisotropy is weak, reflecting the symmetry of the half full 4f orbitals, and this behaviour is due to higher order exchange anisotropy terms.

Terbium, dysprosium and holmium form helical phases, with the moments within the basal plane, reflecting the negative sign of α_J. At lower temperatures they order ferromagnetically, the moments remaining in the basal plane, except for holmium where the ferromagnetic transition does not take place. The α_J of this ion is much smaller than the other two. Instead the moments tilt slightly out of the basal plane giving a ferrimagnetic structure. Application of a moderate field (Koehler et al. 1967) in the basal plane induces the ferromagnetic phase.

Erbium orders with the moment along the c axis, α_J is positive, with a c axis modulated (CAM) structure. The modulation is sinusoidal at high temperatures, but squares up as the temperature decreases. At the same time a component of the moment in the basal plane appears (the magnitude of α_J is only a little larger than holmium) which orders helically. At lower temperatures still the c axis modulation disappears and the moments order ferrimagnetically in a cone structure.
Thulium has the same CAM structure at high temperature and this squares up to give a ferrimagnetic ordering (4 one way, 3 the other) at low temperature.

2.3 Rare earth - Group VIII transition metal intermetallics

The rare earths form an increasing number of intermetallics with the 3d metals as this series is traversed. The crystal structures and magnetic ordering found are set out briefly in this section. For more detail see Wallace(1973) or Kirchmayr(1979).

2.3.1 Rare earth nickel intermetallic compounds.

As Table 2.3 shows there are no binary rare-earth nickel intermetallic compounds with ferro or ferrimagnetic ordering at room temperature.
The moment of the strongly electopositive nickel is reduced to zero by the filling of the 3d band by electrons provided by the rare earth. The 4f electrons on the rare earths then order at a lower temperature than in the pure metal

Rare Earth	R_3Ni ortho-rhombic	R_7Ni_3	RNi ortho-rhombic	RNi_2 cubic	RNi_3 hex-rhomb.	R_2Ni_7 rhomb/ hexag.*	RNi_5 hexag.	R_2Ni_{17} hexag.
La			NM	NM		ʰ	NM	
Ce		NM	NM	NM		ʰ48 FI	NM	48 FI
Pr	2 AF	NM	22 F	NM	20 F	ᵐ85 FI	NM	85 FI
Nd	15 AF		28 F	16 F	27 F	ᵐ87 FI	13 F	87 FI
Sm			45 F	21 F	85 F	ᵐ	25 F	186 F
Eu								
Gd	100 AF		71 F	85 F	116 FI	ᵐ118 FI	33 F	205 FI
Tb	62 AF		52 AF	45 F	98 FI	ᵐ101 FI	27 F	178 FI
Dy	35 AF	R_5Ni_7	62 F	30 F	69 FI	ᵐ81 FI	15 F	168 FI
Ho	20 AF		37 F	22 F	66 FI	ʳ70 FI	23 F	162 FI
Er	7 AF	13 F	13 F	21 F	64 FI	ʳ70 FI	13 F	166 FI
Tm	12 F		8 F	NM	43 FI		22 F	152 FI
Yb								<20 FI
Lu			NM	NM			NM	

Table 2.3 Curie and Néel temperatures for rare earth nickel intermetallic compounds.

* ʰ - hexagonal, ʳ - rhomb., ᵐ - mixed.

2.3.2 Rare earth cobalt intermetallic compounds.

As can be seen from Table 2.4, the light rare earth cobalt intermetallics tend to have ferromagnetic ordering whereas the heavy rare earths have ferrimagnetic ordering. The ordering temperatures increase with increasing cobalt content, and the last two in the series, RCo_5 and R_2Co_{17} have Curie temperatures approaching that of cobalt (>1350K). In addition these compounds have hexagonal structure, and this gives the possibility of strong uniaxial anisotropy. $SmCo_5$ and Sm_2Co_{17} are the basic components in cobalt based rare earth magnets.

Rare Earth	R_3Co ortho-rhombic	$R4Co_3$ hexag.	RCo_2 Cubic	RCo_3 hex-rhomb.	R_2Co_7 rhomb/ hexag.	RCo_5 hexag.	R_2Co_{17} rhomb/ hexag.
La					m	840 F	
Ce				78 F	h	737 F	m1080 F
Pr	7 F		50 F	349 F	h	912 F	r1170 F
Nd	14 FI		120 F	395 F	h	910 F	r1155 F
Sm			240 FI		m	1020 F	m1190 FI
Eu							
Gd	127 FI	230 FI	408 FI	612 FI	r775 FI	1008 FI	m1210 FI
Tb	82 FI	55 FI	250 FI	506 FI	r717 FI	980 FI	m1185 FI
Dy	45 FI		155 FI	450 FI	r	966 FI	m1165 FI
Ho	24 FI	44 FI	95 AF	418 FI	r670 FI	1000 FI	h1180 FI
Er	7 F	25 FI	40 FI	401 FI	r	986 FI	h1180 FI
Tm	5 F		25 FI	370 FI	r	1020 FI	h1185 FI
Yb							—
Lu							h1210 F

Table 2.4 Curie and Néel temperatures for rare earth cobalt intermetallic compounds

* h - hexagonal, r - rhomb., m - mixed.

2.3.3. Rare earth iron intermetallic compounds.

Binary intermetallics with iron are fewer than with cobalt. Only four structures are observed. The important RCo_5 has no iron analogy, and the Curie temperatures of the light rare earth R_2Fe_{17} are only up to 395K.

Rare Earth	RFe_2 cubic	RFe_3 hex-rhomb.	R_6Fe_{23}	R_2Fe_{17} rhomb/ hexag.
La				
Ce	235 F			m70 F
Pr				r282
Nd			492	r327
Sm	700 FI	651		r395
Eu				
Gd	782 FI	728 FI	468	m472 FI
Tb	705 FI	648 FI	574	m408 FI
Dy	638 FI	600 FI	524	h363 FI
Ho	614 FI	567 FI	501	h325 FI
Er	596 FI	553 FI	493	h310 FI
Tm	610 FI	539 FI	475	h271 FI
Yb				
Lu	610 F	529	471	h100 F

Table 2.5 Curie and Néel temperatures for rare earth iron intermetallic compounds

* h - hexagonal, r - rhomb., m - mixed.

2.4 Preparation and Purification of Rare Earths.

Rare earths occur in nature as ore containing a mixture of rare earths. Cerium is the most abundant, followed by lanthanum, neodymium and then praseodymium. Preparation is either by reduction of a salt and purification of the metal, or by purification of the salt and then reduction to produce the pure metal. In practice highest purities are obtained by a combination of salt and metal purification. Impurities can greatly affect the physical properties, in particular transport properties, and many older magnetic measurements show effects of impurities.

Rare earths can be purified by fractional distillation and zone refining, to reduce the lattice impurities, and electrotransport to reduce the interstitial impurities.(Beaudry and Gchneidner 1978)

Large single crystals have been grown by "strain annealing" to allow recrystallization, zoning, Czochralski and Bridgman methods. Crystals with dimensions of 1 to 2 cm have been grown for most rare earths.

Chapter 3: Rare Earth based Permanent Magnets.

3.1 Introduction.

Magnet development began around the century, but although extensively used in motors and generators, permanent magnet performance has only just begun to compete with the power of electromagnets. Magnets were first made from carbon steel, addition of cobalt improved the performance somewhat, but increased the price substantially. The development of an alloy of iron, aluminium, nickel and cobalt and subsequent manufacturing processes to obtain grain orientation and precipitation hardening meant that by the mid 1950's the microstructure had been essentially optimised and little further progress was possible with these weakly magnetically anisotropic materials.

Strnat and co-workers(1967) first showed the promising properties of RCo_5 intermetallic compounds for permanent magnets, and over the next ten years a lot of research went into producing magnets based on $SmCo_5$ and Sm_2Co_{17}.

Sagawa et al.(1984) and Croat et. al.(1984) both announced the stabilisation of a rare earth iron intermetallic phase suitable for permanent magnet production by the addition of boron.

3.2 Rare earth cobalt magnets

As shown in section 2.3.2., there exist intermetallic compounds at the cobalt end of the iron-cobalt system which have Curie temperatures in excess of 1000K and hexagonal crystal structure. Compounds containing the light rare earths order ferromagnetically, whereas the rare earth and cobalt sub lattices oppose each other in compounds containing the heavy rare

earths. The basic structure is the RCo_5, and the R_2Co_{17} as well as the R_2Co_7 cells can be derived from the RCo_5 unit cell by alteration in stacking order and replacement.

The hexagonal crystal structure gives the possibility of a uniaxial anisotropy, but unfortunately the crystal field parameters give only samarium with a positive magnetocrystalline anisotropy amongst the ferromagnetic light rare earths. Er, Tm and Yb have reduced saturation moments due to ferrimagnetic ordering. Samarium is not the commonest rare earth, but it is available in sufficient quantities for the present magnet demand.

$SmCo_5$ magnets are therefore the simplest rare earth transition metal magnets and account for the bulk of such magnets presently produced. The normal production process consists of producing a powder, either by milling an ingot or co-reducing, and then sintering the powder to produce the magnet. Increasing the cobalt content and a suitable heat treatment produces "precipitation hardened" magnets, a fine platelet structure of $SmCo_5$ and Sm_2Co_{17}. The larger cobalt content of Sm_2Co_{17} should give a higher saturation magnetisation than the $SmCo_5$, but the cobalt sublattice has a negative anisotropy which competes with the rare earth sub lattice leading to ferrimagnetic ordering. However the platelet structure gives an improvement in the permanent magnet properties over $SmCo_5$.
Traces of other materials are also included in commercial magnet processing.

3.3 Rare earth iron magnets

3.3.1. Introduction

Cobalt is expensive and also limited in the areas of the world where it can be mined. Samarium is also one of the more expensive rare earths.

However the iron analogue to RCo_5 does not exist and even if it were to be produced as a metastable phase the Curie temperature would probably be too low. R_2Fe_{17} intermetallics do exist, but they have low Curie temperatures and the rare earth anisotropy is not strong enough to overcome an unfavourable 3d anisotropy.

It was therefore realised that any magnet must be based on a ternary rare earth iron compound. Such a compound should have a high uniaxial anisotropy, i.e. it can not be cubic, and a high Curie temperature. So far the only compounds to fit the requirements have been those of the type $R_2Fe_{14}B$. This structure is described in section 3.2.2.2., and a large part of the experimental work described in this thesis has been in measuring magnetocrystalline anisotropy in this material. The structure of $R_{1+\epsilon}Fe_4B_4$ is also described in section 3.2.2.1., as this occurs in conjuction with the main phase and may play an important role in the coercivity mechanism.

3.3.2. Crystal Structure

3.3.2.1. $R_{1+\epsilon}Fe_4B_4$

This interesting structure was first described by Braun and co-workers(1982) and then later investigated by Givord et al(1985) It consists of a series of columns constructed of iron tetrahedrons linked along opposite edges. These columns are connected together by boron atoms to form a square 'honeycomb' (Fig 3.1). The cells are filled by stacks of rare earth ions, the repeat being only weakly linked to the repeat distance in the iron lattice. This is described in terms of two tetragonal lattices with different c parameters. For example for neodymium the iron sublattice has

a=7.117 'A, c=3.502 'A and the neodymium sublattice has a=7.117 'A and c= 3.897 'A. This leads to a repeat every 8 neodymium ions and a 'superlattice' with c=57'A (see fig 3.2).

Different rare earth ions with other radii give different c parameters for the rare earth sublattice and the weakness of the linkage between the lattices leads to a different number of rare earth ions before the structure repeats. This gives different superlattice sizes and also to a different value for ϵ, which defines the rare earth : iron ratio.

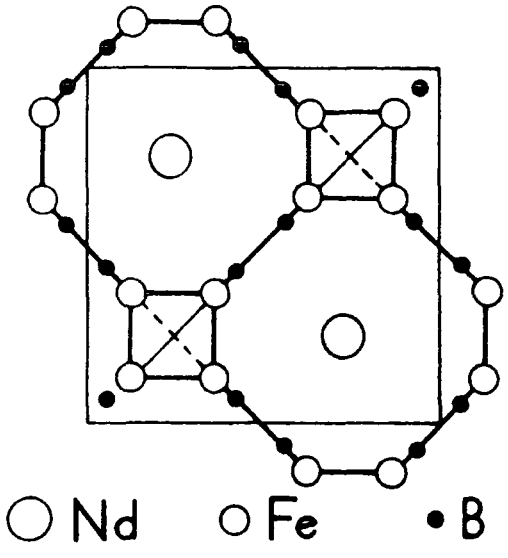


Fig 3.1 $R_{1+\epsilon}Fe_4B_4$ looking down the c - axis (after Givord et al 1985)

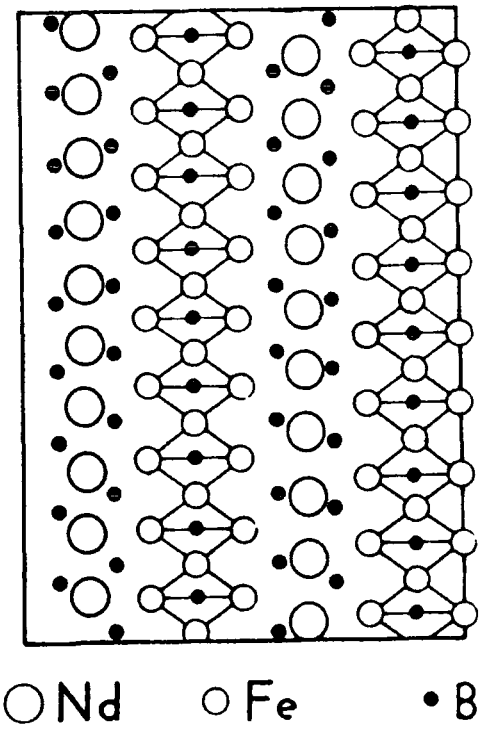


Fig 3.2 $R_{1+\epsilon}Fe_4B_4$ from (110) (after Givord et al 1985)

3.2.2.2. $R_2Fe_{14}B$

The structure of $R_2Fe_{14}B$(see Fig 3.3) was first solved by Herbst et al.(1984) and the results have been confirmed by other investigators (Givord et al. 1984, Shoemaker et al 1984). The tetragonal unit cell contains 68 atoms, with 2 different rare earth sites, and six different iron

sites. The local symmetry was investigated using the program included in Appendix 2 and some pictures generated are included in Fig 3.4. These are based on the positional parameters of Shoemaker et al. for $Nd_2Fe_{14}B$. The numbering of Shoemaker is also adopted for the discussion that follows.

The structure consists of an approximately triangular network of Nd, Fe, and B ions on the z=0 and z=0.5 planes. Above and below these planes are buckled planes of Fe ions, made up of nets of hexagons and triangles. The hexagons sit over the Nd ions, so that both rare earth sites have a CN of 20 with the six surrounding ions in the z=0 plane, six in each hexagon above and below and an Fe(4) ions on each side, positioned about half way between the rare earth planes. The Nd(1) and Nd(2) sites differ only in their neighbours in the z=0 plane, vis a B, two Fe and three Nd for Nd(1)(fig 3.4a) and two B, two Fe and two Nd for Nd(2)(fig 3.4b).

The boron ion sits at the centre of triangles in the planes above and below, and therefore in a triangular prism. This is contorted by the closeness of a Nd(1) ion, which opens up one side of the prism(fig 3.4c).

The Fe(4) ion mentioned above has a CN of 14, with the two Fe hexagons above and below it, where it is completing the close pack layer for both layers!, and the rare earth ions also centred in these(fig 3.4d).

The Fe(1) ion in the z=0 plane has a CN of 12+2. It is in the centre of four Nd ions in the z=0 plane, with two B ions completing the hexagon, but really too far away to be considered touching. Above and below are two pairs of triangles from the Fe nets.(fig 3.4e).The Fe ions within the networks take on a variety of variations on an icosahedron, depending on their relationship to the ions in the z=0 plane(fig 3.4f,g,h,i).

Nd(1)		B	2.9194
	4	Fe(2)	3.0704
	2	Fe(4)	3.1393
	4	Fe(3)	3.2815
	4	Fe(5)	3.2948
	2	Fe(1)	3.3853
	1	Nd(1)	3.5538
	2	Nd(2)	3.7993
Nd(2)	2	Fe(4)	3.0531
	4	Fe(3)	3.0707
	2	Fe(2)	3.0727
	2	Fe(1)	3.1211
	2	Fe(6)	3.1878
	2	B	3.2456
	2	Fe(5)	3.2852
	2	Nd(1)	3.7993
Fe(2)		B	2.1139
		Fe(3)	2.461
		Fe(6)	2.5039
		Fe(3)	2.5172
		Fe(3)	2.531
		Fe(1)	2.5809
		Fe(5)	2.5916
		Fe(2)	2.5921
		Fe(4)	2.7145
		Fe(4)	2.7545
		Nd(1)	3.0704
		Nd(2)	3.0727
		(Fe(2)	3.1089)
Fe(1)	4	Fe(3)	2.5
	4	Fe(2)	2.5809
	2	Nd(2)	3.1211
	2	Nd(1)	3.3853
	(2	B	3.4917)
Fe(3)		Fe(5)	2.3891
		Fe(2)	2.461
		Fe(1)	2.5
		Fe(2)	2.5172
		Fe(2)	2.531
	2	Fe(3)	2.5493
		Fe(3)	2.5561
		Fe(4)	2.6343
		Fe(4)	2.6599
		Nd(2)	3.0707
		Nd(1)	3.2815
B	2	Fe(6)	2.0705
	4	Fe(2)	2.1139
	1	Nd(1)	2.9194

Table 3.1 Bond lengths in $Nd_2Fe_{14}B$. (calculated from Shoemaker et al 1984)

Fe(5)	2	Fe(3)	2.3891
		Fe(5)	2.4367
		Fe(6)	2.5176
	2	Fe(2)	2.5916
	2	Fe(4)	2.6466
		Fe(4)	2.7804
		Nd(1)	3.2852
	2	Nd(2)	3.2948
Fe(4)	2	Fe(3)	2.6343
	2	Fe(5)	2.6466
	2	Fe(3)	2.6599
	2	Fe(2)	2.7145
	2	Fe(2)	2.7545
		Fe(6)	2.7742
		Fe(5)	2.7804
		Nd(2)	3.0531
		Nd(1)	3.1393
Fe(6)	2	B	2.0705
	4	Fe(2)	2.5039
	2	Fe(5)	2.5176
	2	Fe(4)	2.7742
	2	Fe(6)	2.7983
	2	Nd(2)	3.1878

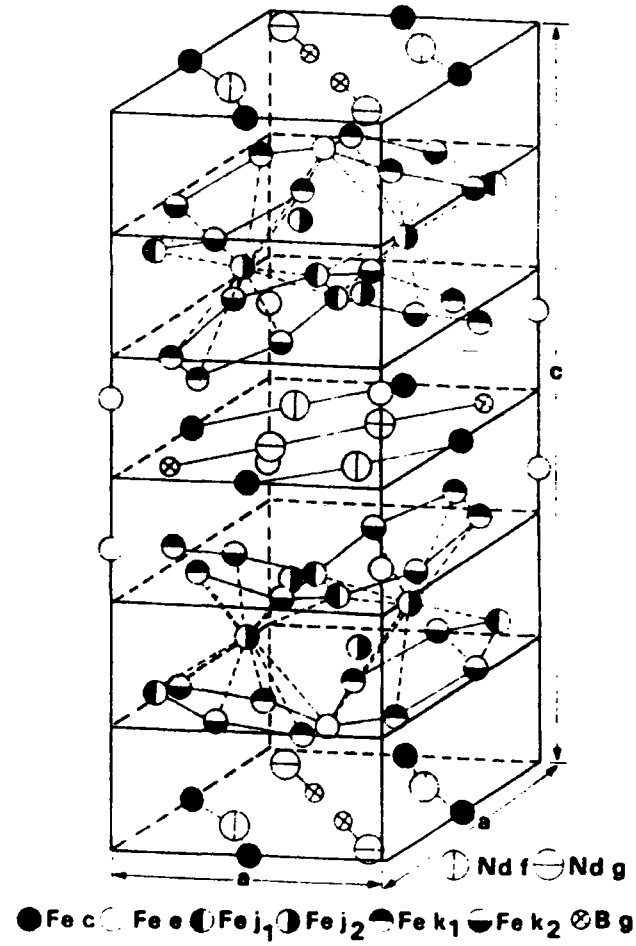


Fig 3.3 Structure of $Nd_2Fe_{14}B$ (after Herbst et al 1984).

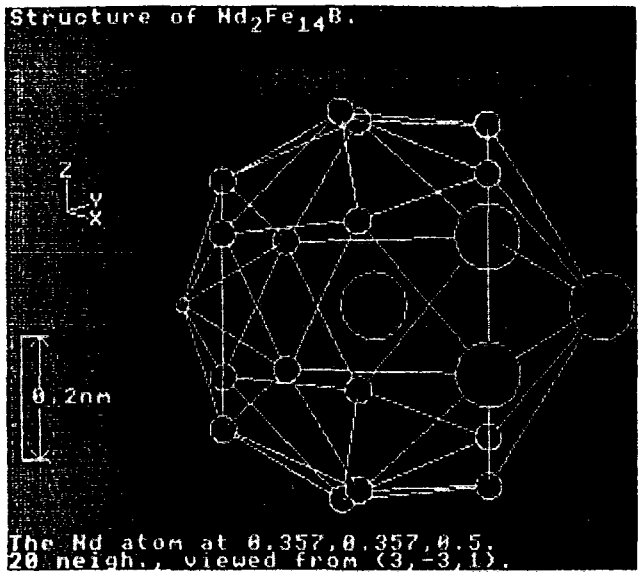


Fig 3.4a Nd(1)

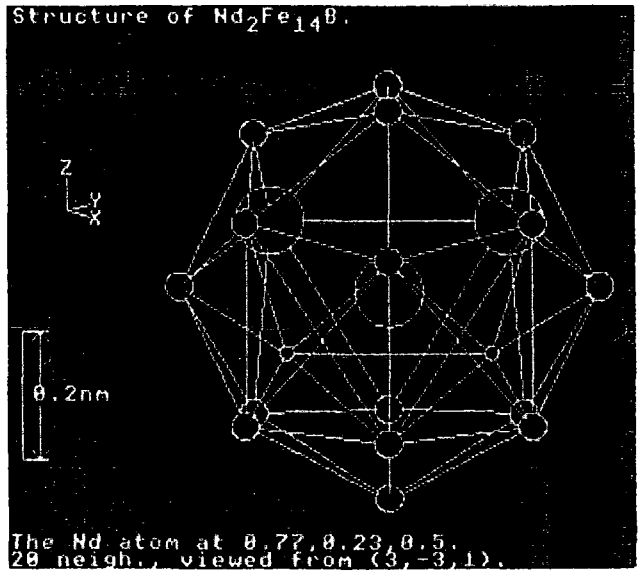


Fig 3.4b Nd(2)

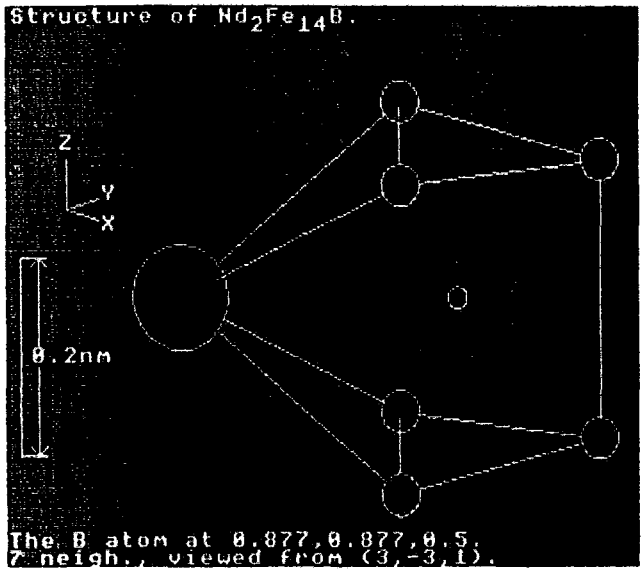


Fig 3.4c B

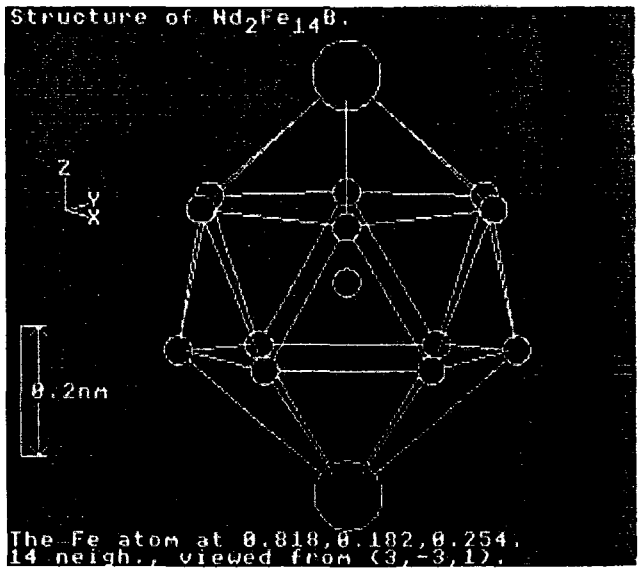


Fig 3.4d Fe(4)

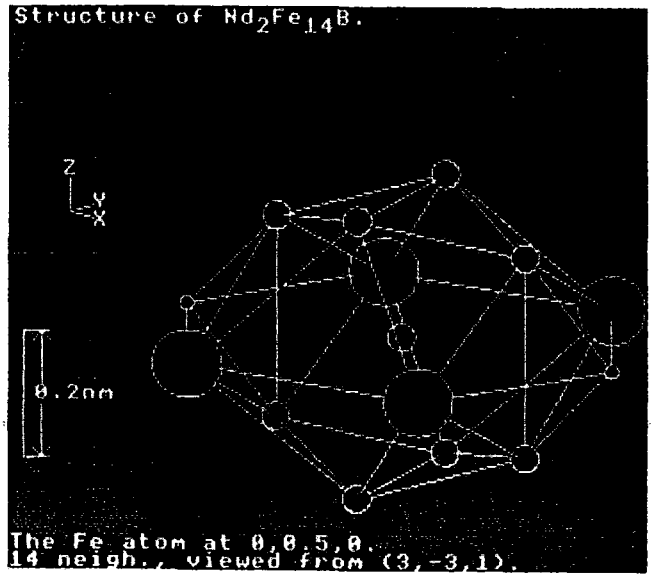


Fig 3.4e Fe(1)

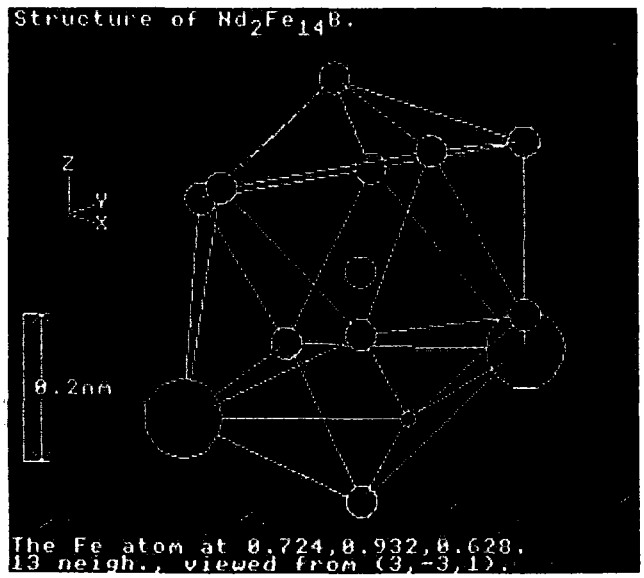


Fig 3.4f Fe(2)

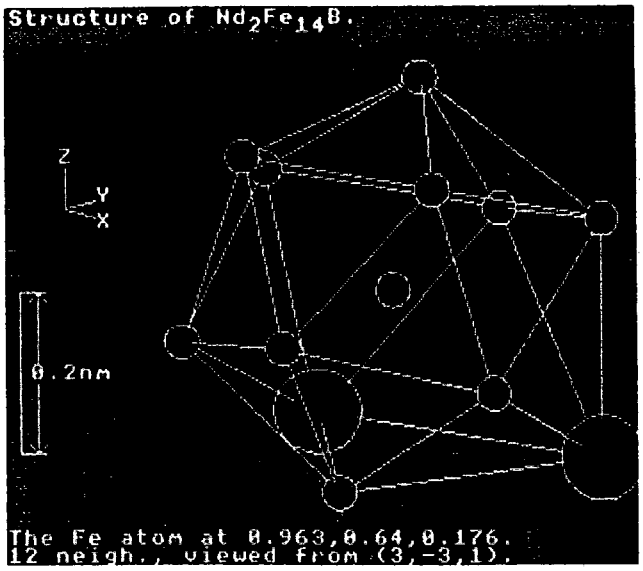


Fig 3.4g Fe(3)

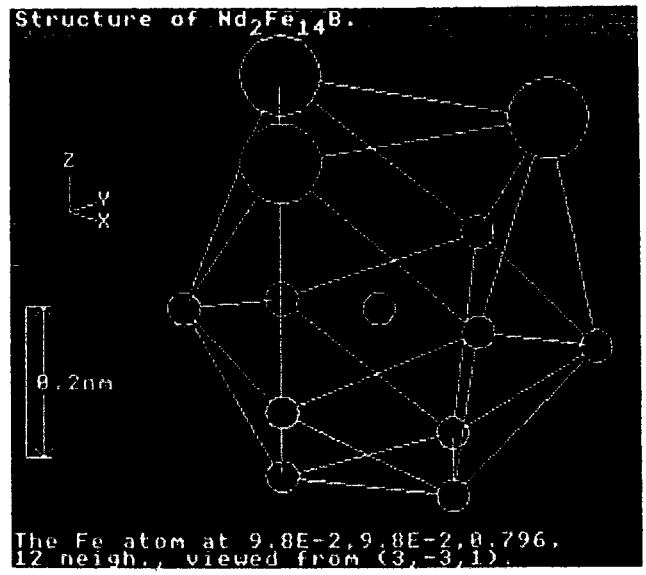


Fig 3.4h Fe(5)

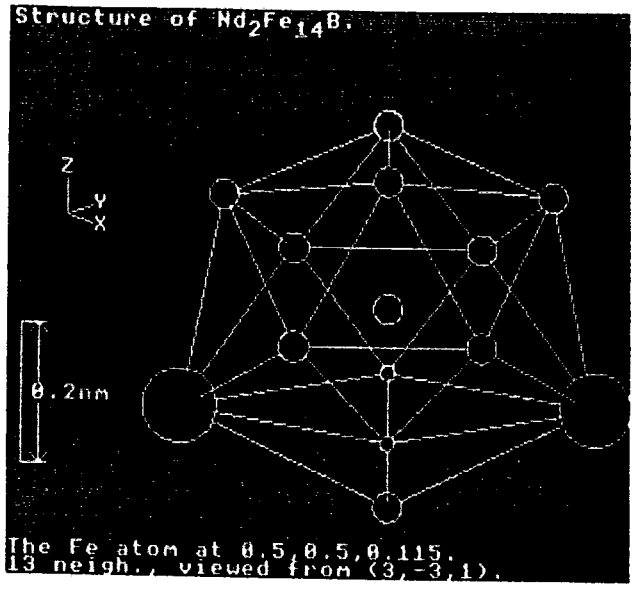


Fig 3.4i Fe(6)

Fig 3.4 Local environments of atoms within the $Nd_2Fe_{14}B$ structure

An interesting point of the structure is that despite some local similarity of environment with the rare earth environment in the $CaCu_5$ related structures, the rare earth anisotropy is primarily opposite in sign to the anisotropy of the RCo_5 series. Point charge calculations made by Cadogan and Coey(1984) showed that this behaviour was explicable in terms of a single ion model for anisotropy. However the spin reorientation observed in

$R_2Fe_{14}B$, R=Nd, (Yamada et al 1986),Tm and Er(Hirosawa et al 1985), Ho and Pr (Grossinger et al. 1986) and Yb(Meyer (to be pub.)) can only be explained in terms of exchange anisotropy(Sankar and Narasimhan 1986) or higher order crystal field terms.

3.3.3. Intrinsic Magnetic Properties

3.3.3.1. Spontaneous Magnetisation

Magnetisation measurements have been reported for members of the $R_2Fe_{14}B$ series by many authors, using both vibrating sample magnetometer and Faraday balance methods. Table 3.2 contains this information compiled by Buschow(1986) for single crystal samples where available and for aligned powders where not. Older reports tended to underestimate the magnetisation due to the presence of non magnetic phases.

The Curie temperatures are all in the range 515-660K with most lying close to 600K. These are some 200K higher than the equivalent R_2Fe_{17}. The ferrimagnetic ordering of the heavy rare earths can be seen in the lower Js and also in an increase in Js between 4.2K and room temperature. This effect (see sect 1.2.3.4) is due to the localised rare earth moment falling off more rapidly than the iron moment. As the iron has the larger overall moment this results in a rise in Js and there exists no compensation point (e.g.Fig 1.3C)

3.3.3.2. Magnetocrystalline Anisotropy

Values for the anisotropy constants for most $R_2Fe_{14}B$ compounds are given in Table 3.2. These are average values from the literature. They have been derived from magnetization measurements using either from the difference in area between curves along an easy or hard direction, by measuring the anisotropy field where the easy and hard curves meet and calculating the

anisotropy or by using a singular point detection method. For $Nd_{14}Fe_{14}B$ some values from torque measurements have also been included. The spin reorientation mentioned in 3.2.2.2. is seen in the change of sign of K_1 between 4.2K and room temperature in $Ho_2Fe_{14}B$ and $Nd_2Fe_{14}B$.

			Room Temperature			At liquid Helium		
	Density $10^3 kgm^{-3}$	Tc K	Js T	K_1 MJm^{-3}	K_2 MJm^{-3}	Js T	K_1 MJm^{-3}	K_2 MJm^{-3}
Y	6.98	571[F]	1.36	1.06	0	1.55	0.80	-
La	7.39	516[F]	1.38	-	-	1.48	-	-
Ce	7.78	533[F]	1.16	1.7	-	1.49	1.8	-
Pr	7.45	565[F]	1.41	5.6	-	1.84	24	-
Nd	7.59	588[F]	1.59	5.0	0.66	1.86	-16	28
Sm	7.76	618[F]	1.49	-12	0.29	1.66	-26	-
Gd	7.88	660[FI]	0.84	0.67	-	0.90	0.65	-
Tb	7.92	629[FI]	0.62	5.9	-	0.66	6.9	-
Dy	8.05	593[FI]	0.67	4.5	-	0.57	3.8	-
Ho	8.09	574[FI]	0.85	2.5	-	0.57	-1.1	4.4
Er	8.23	557[FI]	0.95	-0.03	-	0.66	-1.4	-
Tm	8.20	540[FI]	1.10	-0.03	-	0.94	-3.6	-
Lu	8.41	539	1.17	-	-	1.45	-	-

Table 3.2 Intrinsic magnetic properties of $R_2Fe_{14}B$ (after Buschow 1986).

3.3.3.3. Magnetostriction

Few measurements on magnetostriction have been reported in this class of material. Ibarra et. al.(1986) made measurements on polycrystalline samples of $R_2Fe_{14}B$ (R=Nd,Ho,Dy and Y) and concluded that the rare earth ion dominated the magnetostrictive behaviour. They were also able to observe an anomaly at about 150K in $Nd_{14}Fe_{14}B$ and 100K in $Ho_{14}Fe_{14}B$ corresponding to the spin reorientation transition.

3.3.4. Extrinsic Properties.

The coercivities of magnets of this class vary enormously, this is probably due to the complex structure of the intergranular material, with a legion of phases. These range from a liquid sintering phase, the $R_{1+\epsilon}Fe_4B_4$ phase, a so-called Nd-rich phase, Nd oxide and α iron. How these phases interact

with the $R_2Fe_{14}B$ phase, isolating the exchange interaction from one grain to the next and possibly locally influencing the intrinsic magnetic properties of the matrix phase has not been very well studied. Microstructure studies, combined with magnet charactarisation and local domain observation are required to complete the picture.

3.3.4.1 Domain Studies.

Many studies have been made of the domain structures in sintered magnets in zero field (e.g. Zhau T. et al 1986, Livingston J.D. 1985) and under applied field (Li and Strnat 1985, Durst 1987) and also on thin slices with Lorentz TEM.(Suzuki T. and Hiraga K. 1986).

On pole faces closure spikes and a large number of multidomain grains visible even at remnance after saturation.

On side faces during demagnetization (ABC in Fig 1.4) spike domains are observed to start at grain boundaries and grow into the grains, this growth being smooth and reversible. Durst noted that there was a tendancy for these spikes to appear in conjunction with relatively large areas of non magnetic phases, in particular Nd oxide and the $R_{1+\epsilon}Fe_4B_4$ phase. He concludes that the demagnetising field of the magnetic void is a major factor in influencing the coercivity. Li and Strnat also observed that some grains completely reversed their magnetisation at larger demagnetising fields.

Suzuki and Hiraga observed a smooth movement and bowing of domain walls. They also noted that a domain wall in their thin sample was pinned in the region of a triple junction of grain boundaries.

There is some controversy as to whether processes observed on the surface and in thin films represent a true picture of the bulk magnetisation processes. This is particularly the case at pole faces. On side faces the observations should be more representative.

Chapter 4: Sample Preparation.

4.1 Starting Materials.

Samples used for measurement came from a variety of sources.

Powder of composition $Nd_{15}Fe_{77}B_8$, prepared by Rare Earth Products, from Roger Coe at Lucas, Solihull. This was aligned and used for anisotropy measurements (samples N1,N2).

Commercial cast ingot of the same composition as the powder, also cast by Rare Earth Products, was obtained from Dr. R. Harris at Birmingham University, Department of Metallurgy and Material Science. This was used for a lot of the domain studies (samples N3,N4,N5).

Sintered magnets were also supplied by Dr. R. Harris and by Dr. A. Clegg from Sunderland Polytechnic Magnets Centre. These were used for torque measurements in collaboration with Mr. I. Coulson of Sunderland Polytechnic (samples S1,S2,S3).

Single crystals of $Gd_2Fe_{14}B$ and $Dy_2Fe_{14}B$, grown from high purity starting materials were supplied by Dr. D. Givord at the Laboratoire Louis Néel, Grenoble. These were used for domain studies and for torque measurements (samples D1,D2,D3 and G1-G4).

Finger ingots of $Nd_{1.7}Fe_{14}B$ and $Ho_2Fe_{14}B$ were made from high purity starting materials (Rare earth 99.99%, Iron 99.99% and Boron 99.8%) using an induction furnace. They were melted in a copper cold boat under a purified argon atmosphere. The ingots were turned and remelted to improve homogeneity. They were then wrapped in tantalum foil, sealed in a silica glass bomb under an argon pressure of 250mmHg at room temperature and annealed for 45 days at 1000°C to induce grain growth and try and ensure a single phase.

The $Ho_2Fe_{14}B$ ingot produced consisted of a single phase ingot, with a grain size of less than 150μm in all parts of the ingot.

It was intended to make an ingot of composition $Nd_2Fe_{14}B$ but an arithmetic error resulted in a composition of $Nd_{1.7}Fe_{14}B$. This finger was multiphase, with a large amount of α iron in the lower half of the ingot. This was verified by microprobe analysis. It is assumed that the iron had not dissolvedin the melt during the ingot production. No other phases were observed in the sections.

These ingots were used for domain studies and for anisotropy measurements (samples H1-H5,H5B and N6,N7,N8).

A reference nickel sample was cut from some a high purity nickel button which had been melted in an induction furnace and quenched by switching the furnace off. This gave a fine grained sample (sample N1).

4.2 Cutting and polishing.

Single crystals were aligned using back reflection Laue photography, the pictures obtained (e.g. Fig 4.1) were compared with computer simulations generated on a BBC micro (e.g. Fig 4.2). The program (see app. 2) calculates the position of spots but makes no allowances for intensity. It was sufficient to check the symmetry of reflections and then study individual spots to ensure the correct orientation had been found.

The crystals were mounted on a goniometer using a conducting glue made from perspexdissolved in acetone mixed with graphite powder. Silver loaded conducting paint was applied to the joint. When the correct crystallographic direction had been found the sample could be transfered onto a specially constructed table in a spark erosion machine and a wire knife used to cut slices perpendicular to the x-ray beam. These slices were polished (see below) before disks were cut from them by spark erosion. The size of the disk cut depended on the size of the crystal.(see Table 4.1).

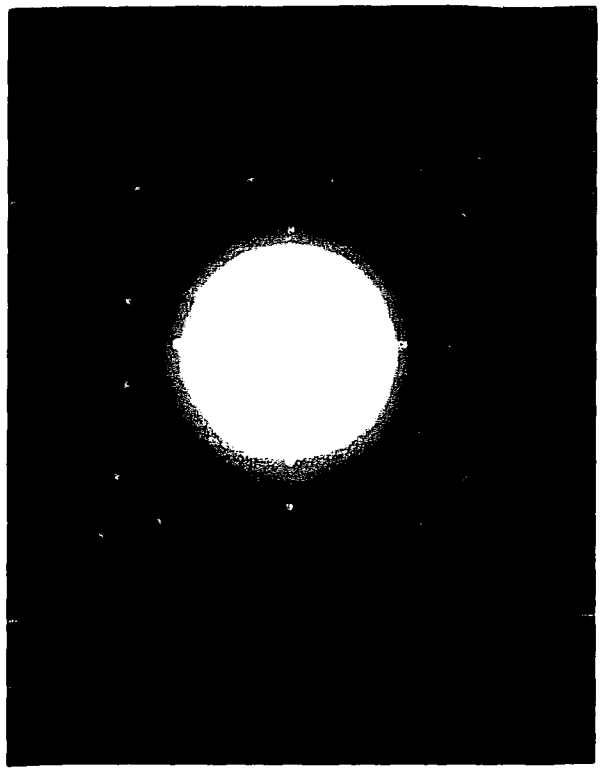


Fig 4.1 Laue photograph of $Gd_2Fe_{14}B$ along a axis (@3)

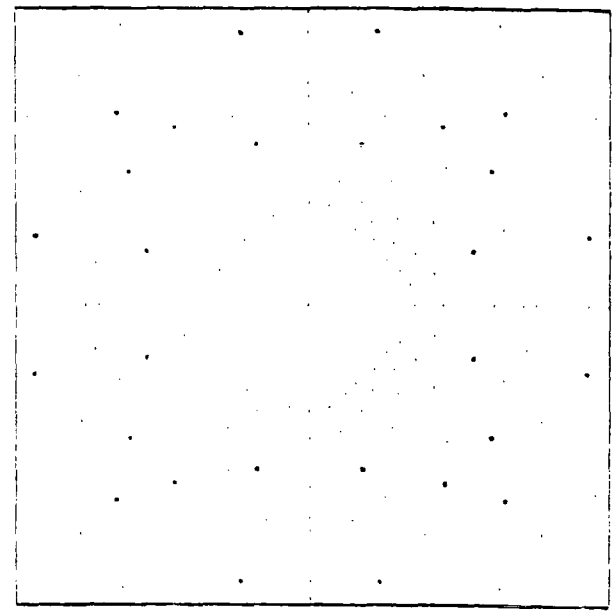


Fig 4.2 Simulated Laue pattern (spots corresponding to those on Fig 4.1 have been darkened)

In this way disks of $Gd_2Fe_{14}B$(@3) and $Dy_2Fe_{14}B$(@3) with the a axis perpendicular to the plane of the disk and a disk of $Gd_2Fe_{14}B$(@4) with the c axis perpendicular to the plane of the disk were cut.

Sintered samples were cut in the same manner($1,$2,$3), the alignment being done by reference to the die shape and the known field direction during aligning. Disks were cut of a size to give reasonable torques.

The $Ho_2Fe_{14}B$ finger was found not to contain any grains of a large enough size to allow a single crystal to be cut. However the preferred growth direction was found to be along the c axis which meant that parts of the

ingot were well aligned as far as this axis went. A section was chosen by polishing cut surfaces and examining the magnetic domain structure and a disk was cut with the c axis in the plane of the disk(H3).

Similarly a disk was cut from the $Nd_2Fe_{14}B$ ingot.(N8).

The nickel sample(NI) was cut at a randon orientation.

4.2.1 Polishing

Samples for domain observation were polished using diamond paste. They were first flattened with emery paper, finishing with 1200 grade. Then they were polished with 6 micron, 1 micron and finally 1/4 micron diamond paste. It was found that no further polishing was required and a sample polished with Syton (a suspension of alumina in a mild chemical etch) showed the same domain pattern as when finished with 1/4 micron diamond polish.

Care had to be taken to polish for a sufficiently long time (about an hour) at each grade, as the emery paper and the coarser paste pulled out the softer intergranular phases.

Etching a polished sample with 50% nitric acid in ethanoic acid indicated that there was little surface strain as the polished surface took much longer to etch that the ingot surface and the etching started along the grain boundaries.

4.3 Setting and aligning.

For the compositions where no single crystal material was available aligned powder samples were made.

Samples were made from the Rare Earth Products powder, a sample produced by filing the $Ho_2Fe_{14}B$ ingot and one from filing the top half of the $Nd_{1.7}Fe_{14}B$ finger, where no free iron was observed in a section.

These were set in fibre-glass resin using moulds made from polytetrafluoroethane (PTFE). Two moulds were made, one to allow setting in an electromagnet(Fig 4.3)(H4,H5,H5B and N6,N7) and the other to allow setting in the superconducting solenoid(Fig 4.4)(N1,N2).

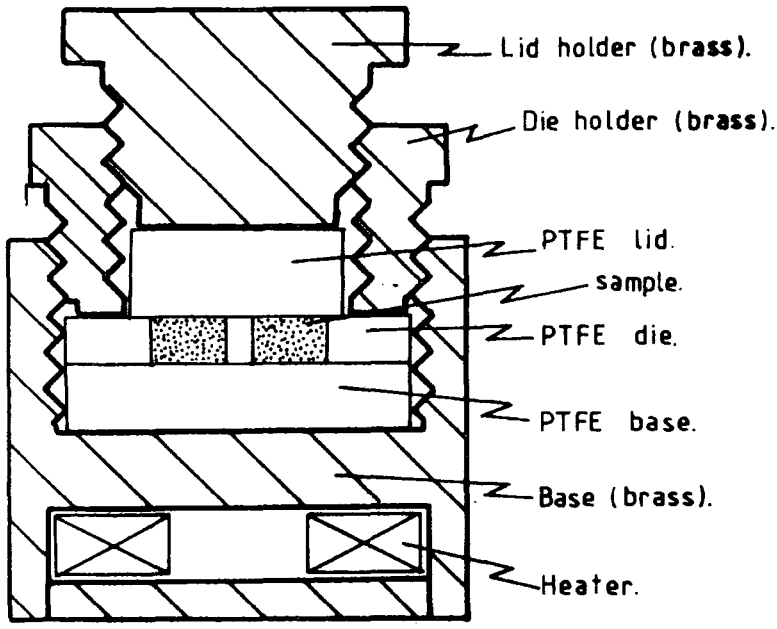


Fig 4.3 Mould for setting powders in electromagnet.

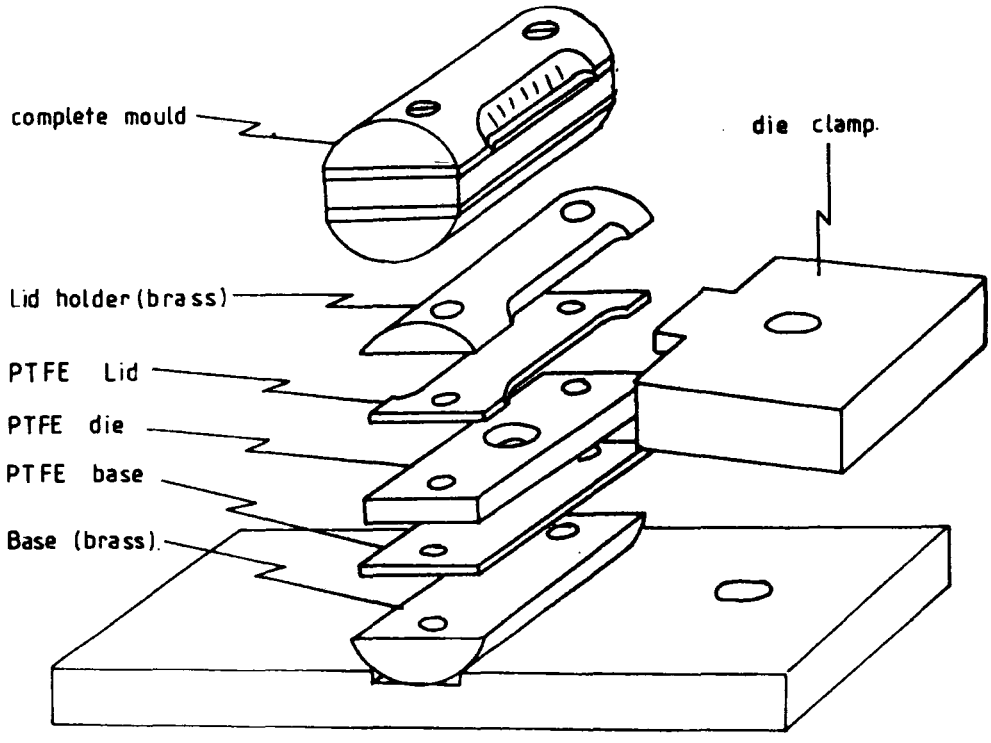


Fig 4.4 Mould for setting powders in solenoid.

The principle was the same for both, that is a slice of PTFE with holes reamed in it was sandwiched firmly between two other pieces of PTFE. While filling the slice could be held firmly against the lower piece and then the upper piece could be placed on top as a lid and secured in place. The natural self lubrication of the PTFE meant that the disk shaped samples produced could be easily pushed out of the dies.

The electromagnet mould was made of two disks of PTFE with a central disk with holes reamed in it as a die. These were held in a large screwing clamp, with two concentric screw threads.

The solenoid mould was made of a brass cylinder split down the centre with PTFE slices sandwiched in it. While filling the lower half could be held in a pair of clamps and then when filled was held together with two screws. It could be lowered into the centre of the magnet at the end of a stainless steel tube in a simple holder.

4.4 Hydrogen Decrepitation

4.4.1 Single crystal separation?

Following discussion with I.R.Harris in the University of Birmingham and with reference to his work on the samarium cobalt system (e.g. Kiavanash and Harris 1984,1985) he suggested that there was a possibility of breaking up large grain ingots of the neodymium iron boride to produce single crystals for intrinsic magnetic property measurements. An attempt was made to produce single crystals of $Nd_2Fe_{14}B$ large enough to be used for torque measurements by hydrogen decrepitation. This was the preliminary study carried out at Birmingham, which was later followed up by a more extensive study of hydrogen decrepitaton of the ingot (Harris et al 1985) and magnet production (McGuiness et al 1986).The base material for this was the cast ingot material of commercial composition (N3). This had a variation of grain size from an average grain width of 20μm near the surface to 50μm through

the material(Fig 4.5), but there was no conclusive evidence of grain growth. On exposure to hydrogen at 1.0 bar for 16 hours the material was cracked but retained its integrity. It was left in hydrogen for 100 hours and on handling broke into a coase powder with some lumps up to 5mm in size. These lumps were however badly cracked and no use as single crystals.

It was concluded that the heat treatment did not produce very good crystals due to the inclusions of a secondary phase. It is also suggested that the hydrogen decrepitation proceeds most rapidly within this secondary phase and therefore was able to attack the as cast ingot rapidly due to the interconnected network structure but took longer for the hydrogen to migrate through the matrix phase in the annealed sample.

Hydrogen decrepitation may possibly be a route for separating single crystals in some materials but in this material the damage produced by the hydrogen attacking inclusions in the crystals rendered the samples useless.

4.4.2 Powder production

In view of the spectacular disintegration of the commercial ingot and attempt was made to produce a powder from the high purity fingers of $Nd_{1.7}Fe_{14}B$ and $Ho_2Fe_{14}B$. These were exposed to hydrogen at 2 bar for 48hrs but showed no signs of cracking. This is taken as further evidence to suggest that the hydrogen attacks the rare earth rich phase in the commercial ingot. It was concluded that there was little or none of this phase in the high purity ingots.

A powder was therefore produced with a mechanical filing with a clean file. Half of the powder was set immediately(H4,N6) while half was annealed at 1000°C for 2 hours to remove any strain introduced by the filing process before setting(H5,H5B,N7). The powder produced had a particle size of 10-20μm. These particles are magnetically attracted to form agregates of about 500μm and it was not possible to measure the particle size very accurately.

	compostion	form	mass grams	%sample %weight	setting field/T	shape	diameter mm	thickness mm	orientation (disk plane)
D1	$Dy_2Fe_{14}B$	multigrain	-	-	-	polished surface			
D2	$Dy_2Fe_{14}B$	multigrain	-	-	-	polished surface			perp (001)
D3	$Dy_2Fe_{14}B$	single crystal	0,01474	-	-	disk	1,8	0,76	perp (100)
G1	$Gd_2Fe_{14}B$	single crystal	-	-	-	polished surface			
G2	$Gd_2Fe_{14}B$	single crystal	-	-	-	polished surface			perp (001)
G3	$Gd_2Fe_{14}B$	single crystal	0,03035	-	-	disk	2,20	1,1	perp (100)
G4	$Gd_2Fe_{14}B$	single crystal	0,02364	-	-	disk	2,13	0,85	perp (001)
H1	$Ho_2Fe_{14}B$	annealed ingot	-	-	-	polished surface			
H2	$Ho_2Fe_{14}B$	annealed ingot	-	-	-	polished surface			perp (001)
H3	$Ho_2Fe_{14}B$	aligned ann. ingot	0,06314	-	-	disk	3,6	0,81	
H4	$Ho_2Fe_{14}B$	powder in resin	0,10039	30,0	0,7	disk	±5	±3	
H5	$Ho_2Fe_{14}B$	annealed powder/resin	0,11365	42,53	0,7	disk	±5	±3	
H5B	$Ho_2Fe_{14}B$	annealed powder/resin	0,11377	42,53	0,7	disk	±5	±3	
N1	99,99% Ni	multigrain	,07823	-	-	disk	3,6	0,93	
N1	$Nd_{15}Fe_{77}B_8$	powder in resin	0,065	24,3	5,07	disk	±5	±2	
N2	$Nd_{15}Fe_{77}B_8$	powder in resin	0,068	24,9	10	disk	±5	±2	
N3	$Nd_{15}Fe_{77}B_8$	cast ingot	-	-	-	polished surface			
N4	$Nd_{15}Fe_{77}B_8$	cast ingot	-	-	-	polished surface			perp (001)
N5	$Nd_{15}Fe_{77}B_8$	annealed cast ingot	-	-	-	polished surface			
N6	$Nd_{1.7}Fe_{14}B$	powder in resin	0,10267	30,81	0,7	disk	±5	±3	
N7	$Nd_{1.7}Fe_{14}B$	annealed powder/resin	0,11827	44,38	0,7	disk	±5	±3	
N8	$Nd_{1.7}Fe_{14}B$	annealed ingot	0,08315	-	-	disk	4	0,95	
S1	High coercivity comm. magnet		0,06689	-	-	disk	3,7	0,91	perp easy.
S2	5% cobalt sintered magnet		0,02229	-	-	disk	2,18	0,91	perp easy.
S3	10% cobalt sintered magnet		0,02177	-	-	disk	2,20	0,85	perp easy.
T1	terbium	single crystal	0,08412	-	-	sphere	2,7	-	perp (1000)

Table 4.1 Samples used for this work.

Chapter 5: Equipment used

5.1 Torque Magnetometer.

5.1.1 Introduction

Torque magnetometry is a powerful technique for analysing the magnetocrystalline anisotropy of ferromagnetic materials. If the magnetic energy is anisotropic then choice of a suitable crystallographic direction means that the magnetic energy it can be expressed as:

$$E = E_I + E_A(\theta,\phi) \qquad (5.1)$$

where E_I is the isotropic term and $E_A(\theta,\phi)$ is generally expressed in terms of phenomenological constants (sect 1.2.4), θ and ϕ are the usual spherical harmonic coordinates. If the sample is placed in an external magnetic field then it will experience a torque which depends on E_A.

If the sample is free to rotate around the x axis then the torque will be given by

$$T = \delta (E_A) / \delta\theta \qquad (5.2)$$

evaluated in the direction that the moment lies. This direction can be calculated by using equation 1.4 to give the angle between the moment and the external field.

Similar, if more complicated, expressions to 5.2 can be written down if the sample is free to rotate in other directions and from these the full shape of E_A can be calculated.

In order to measure the magnetocrystalline anisotropy of rare earth alloys high fields are required to move the moment appreciably away from the easy axis and explore more of the magnetic energy suface. The Durham University Central High Field Superconducting Solenoid (DUCHESS) was used to provide a field for these measurements. In order to use a solenoid for torque measurements a new type of instrument had to be developed, and this is described in this section and in less detail in a paper by Hawton and Corner(1987). During the investigation modifications of a general nature were made to the solenoid temperature control, field ramping and field measurement systems. Details of these are included in Appendix 1.

5.1.2. The Magnet (Duchess).

The Duchess is a magnet wound of Niobium-Titanium and Niobium-Tin and built by Thor Cryogenics. It provides a steady field of up to 13 Tesla when immersed in liquid Helium at atmospheric pressure, with a field uniformity of 0.5×10^{-4} over a cylindrical region of length 30mm by 20mm diameter. The power for the magnet is provided by a Thor 6010 power supply, which is controlled by a Thor 2020 electronic programmer. The magnet is wound in 8 sections, each of which is protected by having a low resistance short connected in series with it against damage should the magnet become normal at a high current. The reversion of the magnet from a superconducting to a normal state causes a rapid loss of field and boil off of helium and is termed *quenching*. Equipment operating in the solenoid, particularly if containing fine windings, must be protected in the event of such a quench.

The maximum allowable rate of current change in the magnet (ramp rate) is determined by the heat dissipated in the normal part of the windings which are 'Type II' superconductor and hence consist of regions of superconducting and normal material. The field/temperature stability of the superconductor determines the heat dissipation rate for any field and for the Duchess this

means a rate of 0.25 Tesla per min. up to 5 Tesla and 0.06 Tesla per min. from 5 Tesla to 13 Tesla. This gives a minimum time of 2½ hours to ramp up to full field.

The bore of the magnet is 45mm, and within this fit a set of VSM coils, immersed in the liquid helium, and a gas flow cryostat which is able to control the sample temperature between 4.2K and 400K at the bottom of a sample tube with an internal diameter of 24.7mm and a length of 1.5m. The cryostat takes liquid from the helium bath within which the magnet is immersed and this helium flows through a brass sintered plug set in a copper block. The block has a carbon-glass resistance thermometer embedded in it and a heater attached to warm the helium flow. A Thor 3030 controller is used to keep this block at a constant temperature. The warmed Helium then flows up through the sample tube at slightly below atmospheric pressure, a vacuum pump sucking through a needle valve and another needle valve in the liquid tube control the rate of helium flow. The quality of the temperature control depends on the thermal mass of the instrument in the tube. With the torque magnetometer insert described below good control below 25K is obtained in about 10 mins and in about ½hour at room temperature. This difference is probably due to the variation in density of the helium, and therefore its usefulness as a heat transfer medium.

5.1.3. Description of the magnetometer

The present design bears some similarities to previous designs used to measure induced torques (Verge et al. 1977) and de Haas van Alphen oscillations(Vanderkooy 1969). A much more rugged construction was required for the high torques to be measured as well as the ability to measure over a rotation of at least 180 degrees. At the time of construction it was believed to be the only instrument designed to measure such torques in a

solenoid but since its construction an instrument has been described which was designed with the same aim although the solution adopted is completely different and consists of a pulley system to transfer the torque to an external sensor situated on top of the solenoid.(Otani et al 1987).

The Duchess is solenoidal, with access and field along the same axis. This geometry is not compatible with conventional types of torque magnetometer consisting of a sample suspended between the poles of an electromagnet with either a mechanical or an electromagnetic system to produce a counter-torque. In these systems either the measuring assembly or the magnet is rotated to measure the dependence of the torque on angle(see Pearson 1979).

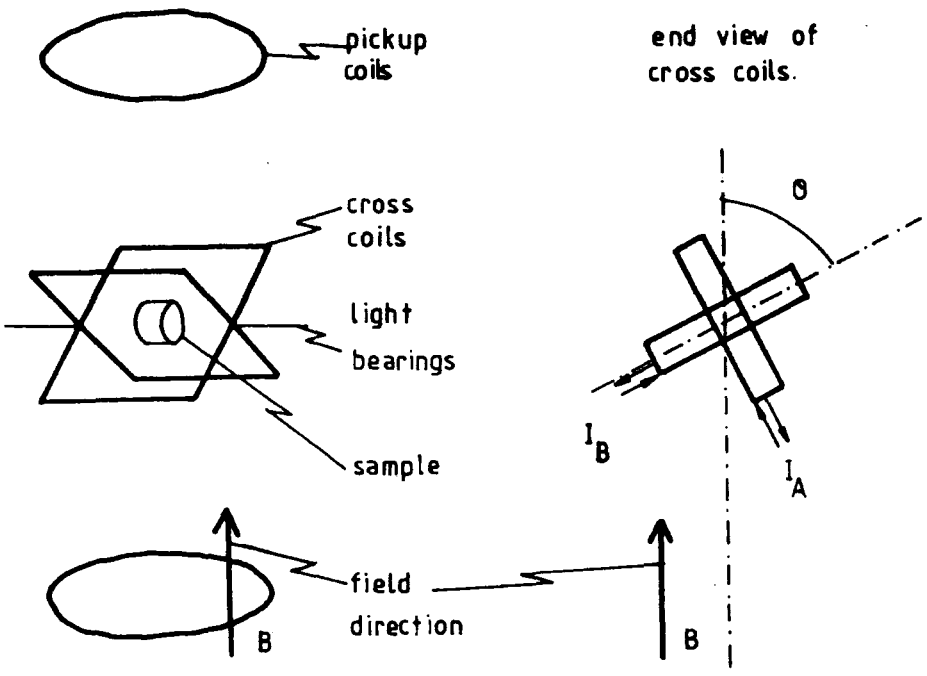


Fig 5.1 Configuration of magnetometer coils.

The Duchess torque instrument was designed with the constraints imposed by the sample space in mind. There are also no mechanical linkages, all connections being electrical. The measuring part of the instrument consists of a pair of coils wound on one former with the planes of the coils at right angles to each other. The sample is fixed in the centre of these

cross coils. This assemblage is free to rotate around an axis through the intersections of the coils (fig. 5.1). Two coils which are wound concurrently are located above and below these cross coils. An alternating current is passed through these outside coils and the alternating e.m.f.'s picked up by the two cross coils are analysed to give a measurement of the angular position of the sample.

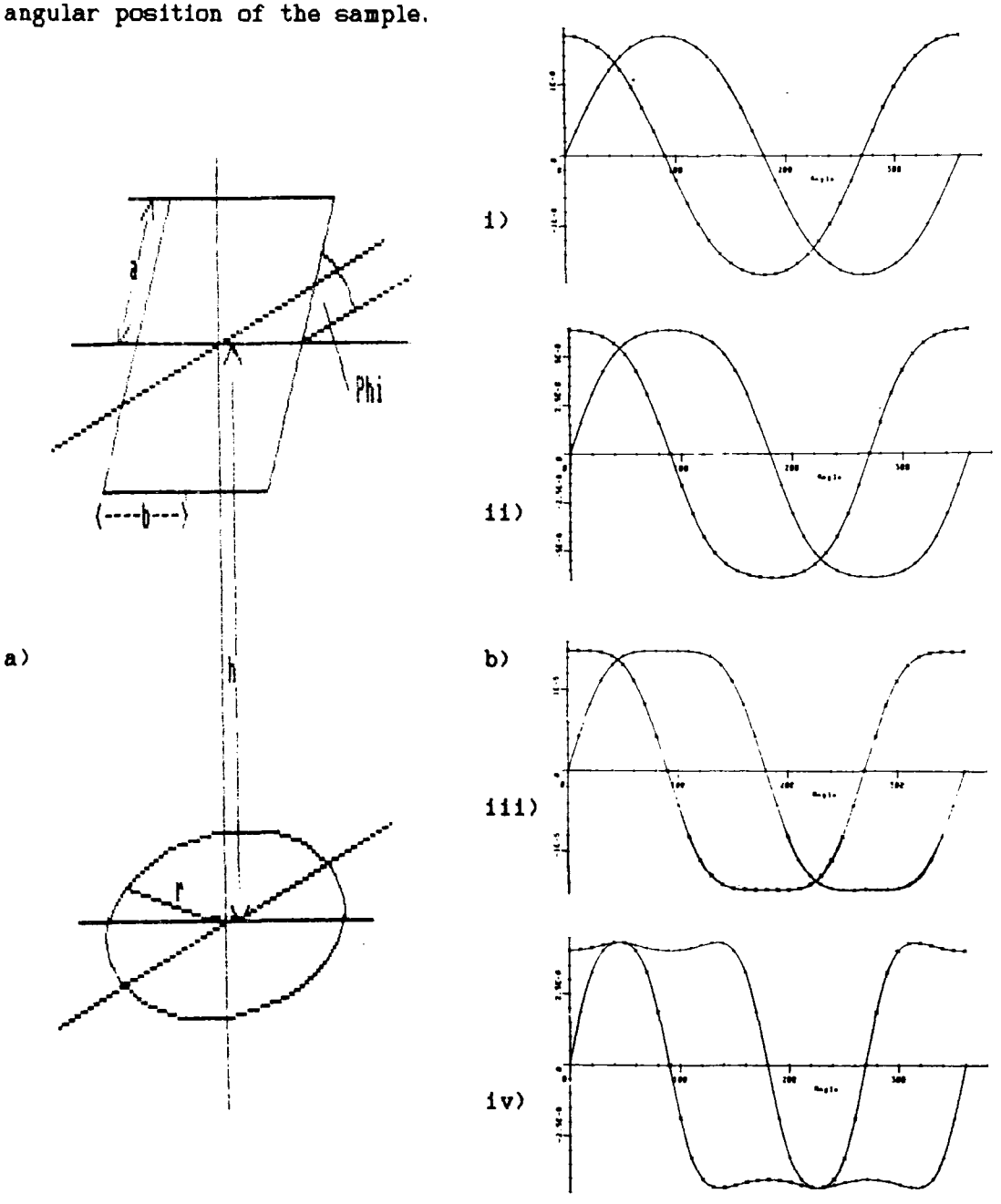


Fig 5.2 Simulation of magnetometer pickup.

a) configuration for calculation. b) typical results:

i) a=8.7,b=5.4,r=11.1,h=42, ii) h=25, iii) h=17, iv)h=17,r=4.9

The separation of the outside pair of coils is chosen to be small enough to ensure adequate pick-up by the cross coils while being large enough for the dependance of pick-up on angular position to be nearly sinusoidal and therefore to permit unambiguous measurement of angle.

Analysis of the pick-up in free space for this geometry was done using numerical integration on a micro (see App. 2) to determine the optimum spacing for the coils and typical results of these are given in fig 5.2. On the basis of these it was concluded that the coils could be positioned close to the cross coils without any loss of resolution. However there is a large difference between the pick-up in free space and in the solenoid (Fig5.3). The solenoid consists of a series of concentric conducting tubes culminating in a large superconducting shield, and it was found in practice that the separation had to be larger than for free space.

The final choice of separation was based on experiment as the effect of the large number of coupled inductances was difficult to calculate.

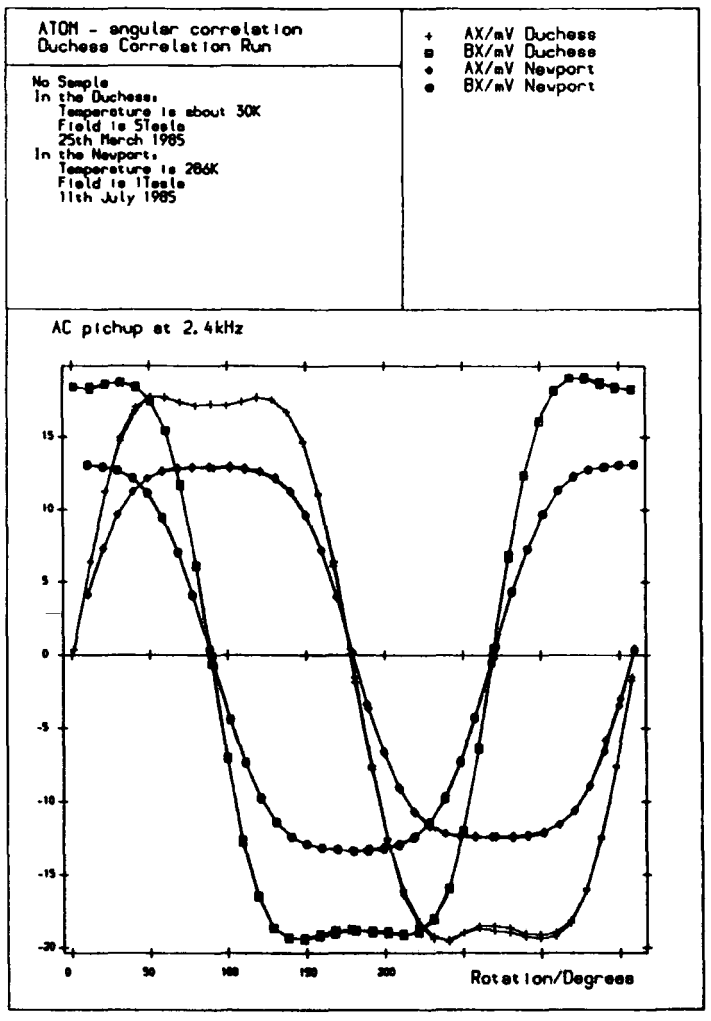


Fig 5.3 Pickup in Duchess and Newport electromagnet

Direct currents are passed simultaneously through the two cross coils and varied under microcomputer control. The interaction of this current in these coils with the field of the superconducting magnet provides the counter-torque to balance the magnetic torque the sample experiences as the moment is pulled away from the easy direction. The torque is calculated as

$$\text{Torque} = B \left(I_A N_A \text{Area}_A \sin(\theta) - I_B N_B \text{Area}_B \cos(\theta) \right) \qquad (5.3)$$

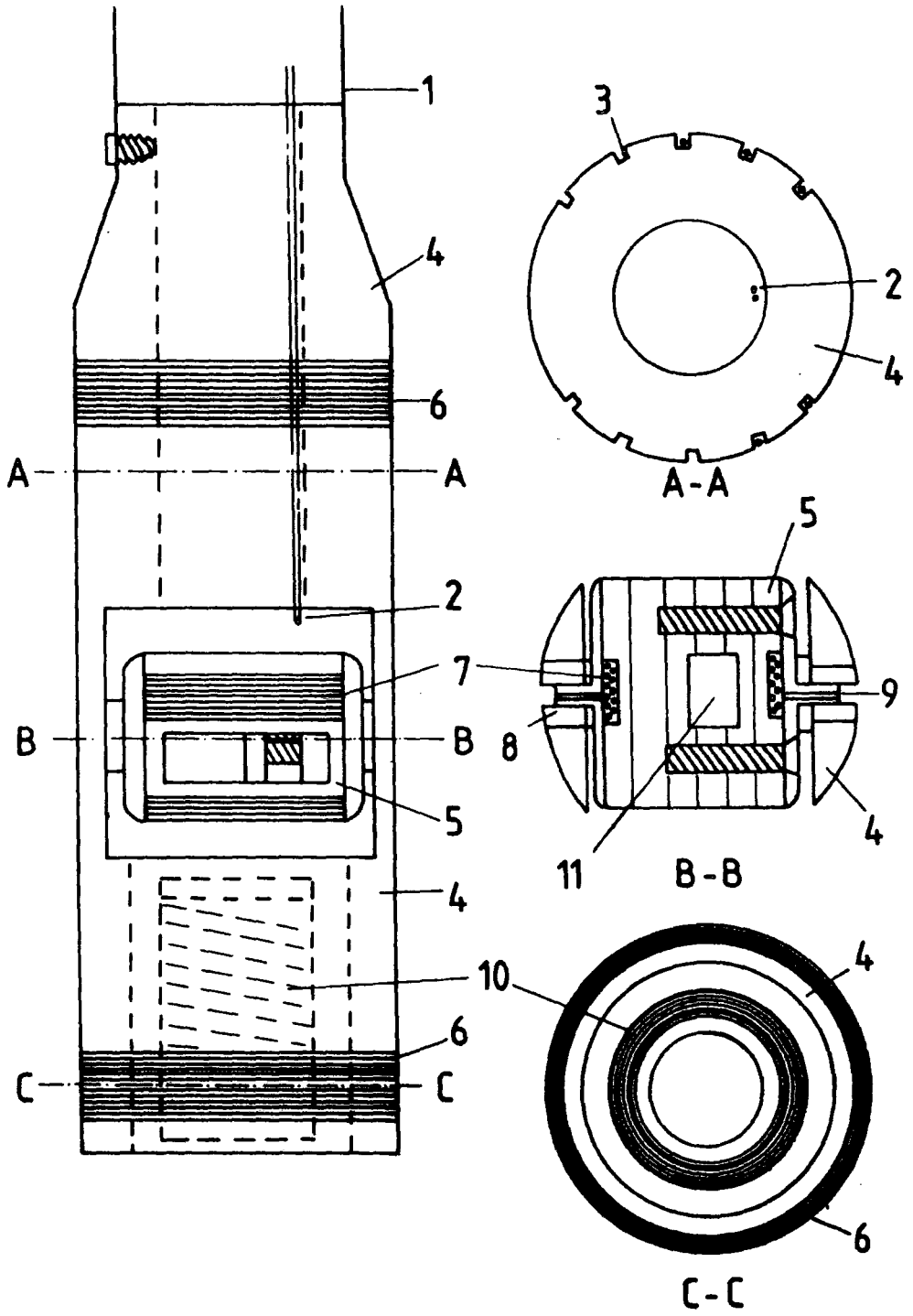
where N_i is the number of turns on coil i, I_i is the current flowing in coil i and Area_i is the cross sectional area of coil i.

The cross coils were made as large as possible within the limited space available in order to reduce the ohmic heating to a minimum for a given counter-torque. The use of two coils makes the instrument usable over a full rotation and allows the total heat dissipated by the coils to be kept constant while the counter torque is varied.

5.1.4. Construction of the apparatus.

The instrument described here (see fig 5.4 and 5.5) allows torques of up to 0.3Nm to be measured with a minimum resolution of 10^{-6}Nm. This range is determined by the space available in the cryostat and the mechanical construction of the magnetometer head. Larger torques would be difficult to obtain in the restriced space but use of finer wiring and more sophisticated bearings could lower the minimum resolution.

The cross coils consist of 140 turns each of 38 SWG enamelled copper wire, wound in slots in a cylindrical cloth Tufnol former, 19mm in diameter and 13mm thick. The coils were wound at the same time with alternate layers interleaving to keep them the same size and allow good thermal contact.



1) Thin walled tube
2) Thermocouple
3) Slots for wires
4) Tufnol housing
5) Tufnol former
6) A.C. coils
7) Cross coil
8) PTFE bearing
9) Hole for wiring
10) Heater
11) Sample

Fig 5.4 Details of magnetometer head.

The AC coils are wound on slots in the Tufnol housing, at a distance of 415mm from the centre of the cross coils. They consist of 600 turns each of 42 SWG copper wire. Small slots are also milled down the length of the housing for wiring to the coils to prevent rubbing on the sample tube during insertion. The centre of the housing is drilled out up the whole length to allow a smooth flow of helium.

A Au(0.07%) Fe/chromel thermocouple mounted just above the sample measures the sample temperature, which differs by up to 5K from the cyostat set point due to the heat dissipated in the cross coils. An auxilliary heater is mounted in the bottom of the housing in addition to the cryostat heater. This is used to speed up sample changing when working at low temperatures and to reduce the heat leakage to the helium bath when working at elevated temperatures.

The complete magnetometer head, see fig 5.4 and Fig 5.5, is mounted on the bottom of a 19mm diameter thin walled (.3mm) stainless steel tube, which gives a rigid, easily removable construction to facilitate sample changing. The top of the tube has a brass coupling with a 10 pin metal in glass seal with the wiring to the heater and the coils on one side and a pass through glass to metal seal for the thermocouple leads. A 25mm KF flange on the top allows connection for the helium flow.

5.1.5. Electronics and Control.

The AC coils are driven by a Feedback VPO602 oscillator through a Quad 50E power amplifier. The frequency used is about 2.4kHz. The oscillator also provides the reference signal for the lock-in amplifier.

The DC for the cross coils is provided by a small dual power supply which is voltage programmable. This is based on a switching regulator design and details are given in Appendix 3. The programming voltage comes from a digital to analogue (D to A) converter and the current is reversed using micro controlled relays. The D to A converters and the relays are contained in a Minicam interface. Details of this interface and also the design of the relay board are given in Appendix 4. The DC current passes through a choke to reduce the amount of high frequency interference from the switched mode power supply reaching the lock-in signal amplifier and to avoid the effective short circuiting of the AC e.m.f. picked up in the cross coils by smoothing capacitors in the final stage of the power supply. These chokes are driven at a fraction of their maximum rating to prevent variations in the susceptibility of their cores with core flux from affecting their impedance and therefore causing the AC voltage, as measured by the lock-in amplifier, to be dependant on the DC current.

The DC current flowing in each coil is measured using a Weston 6400 digital voltmeter to measure the voltage across a shunt. The shunt value can be selected for the required torque range, the same switch also switches load resistors in series and in parallel with the DC power supply to reduce the maximum current to the cross coils. Shunts, load resistors and effective current ranges are given in Table 5.1. The use of external shunts means that one voltmeter can be used to measure currents in both coils without disturbing the system.

The AC is measured by connecting it, through a capacitor of 8 μF, to an EG&G 5206 lock-in amplifier, which has an input impedance of 10MΩ. The capacitor prevents the lock-in amplifier from being subjected to the full output voltage of the D.C. power supply. The lock-in amplifier is set to

extract that part of the induced AC e.m.f. which is out of phase with a reference signal from the oscillator.

Range Selector	Shunt Resist.	Series Resist.	Load Resist.	Maximum Current
High	0.075Ω	0.0Ω	0.0Ω	1.0A
Medium	1Ω	150Ω	100Ω	100mA
Low	10Ω	1,8kΩ	100Ω	10mA

Table 5.1 Range resistances.

5.1.5.1 Computer interface

The instruments are all interfaced with a a BBC model B micro computer. Ideally the interfacing should be as shown in Fig 5.6, that is with all the instruments connected to an IEEE488 bus and then an RS232 connection for the further transfer of data. Unfortunately the instruments available did not all have IEEE488 interface and in some that did there was not a full range of facilities so that they could better be controlled through an RS232 link. In order to do this an RS232 multiplexor was designed and built (Appendix 5). The lock-ins, a Thurlby 1905 voltmeter connected to the thermocouple, and a Clearway node are connected to the multiplexor. The Clearway ring is a local area network installed in the Solid State laboratories. During the experiment this network connects the BBC to an IEEE488 controller to which are connected the Minicam interface and Weston voltmeter. The choice of communication system was based primarily on the available instruments.

Programmes have been written to allow the magnetometer to take a fully automated torque curve. The field and temperature have to be set manually and the range of torque chosen by the experimenter. A full 360 degree rotation in both directions with readings every 10 degrees takes about 20 minutes.

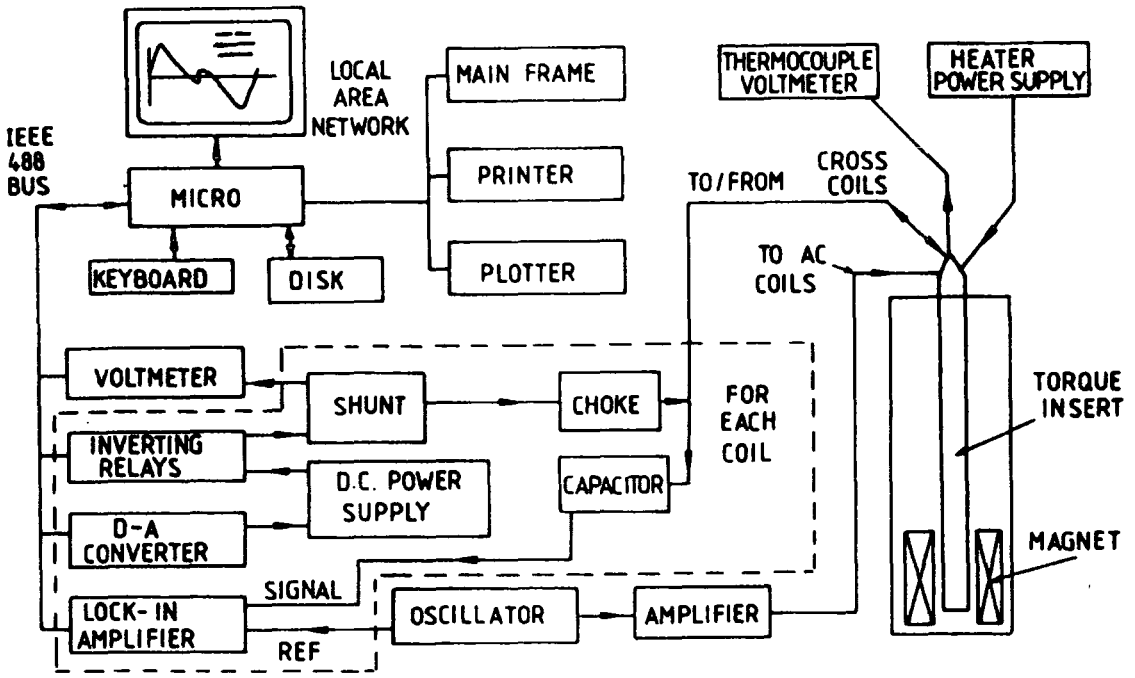


Fig 5.6 Block diagram of axial torque magnetometer.

5.1.6. Measurement Procedure.

To calibrate the angular measurement a complete 360 degree run is done first with no sample in the instrument. As the only torques are then those of the cross coils, the angular position can be calculated and related to the AC e.m.f. picked up in the coils (see Fig 5.7). The ratio of these e.m.f.s is used for this correlation in order to give a good correlation for all angles (see fig 5.8) and partly to correct for any changes in the resistance of the coils during runs at different temperatures. Resistance changes will affect the AC current flowing in the AC coils, but as the impedence of the measuring instrumentation is large the change in resistance of the cross coils has little effect. However there is some slight dependence of the pickup on temperature, probably due to resistance changes in the sample tube and copper shield, which results in errors of up

to 0.5° in the angular correlation at extremes of temperature. To prevent this for more accurate measurement a series of correlation runs must be done, for example every 50K. These correlations are used during subsequent runs to calculate the angle from the e.m.f's picked up.

A sample is then placed in the instrument and a choice is made of the amplitude of rotation each way, around which angular position, and how many readings are to be taken. The range selected and the current required must also be chosen and this determines the maximum torque that can be exerted and the resolution. If too high a current or range is chosen then the errors in the angular correlation are amplified whereas too low a range may result in the instrument being unable to turn the sample against the torque exerted by the sample. The measurement is then done automatically. The torque is calculated during the run and can therefore be plotted

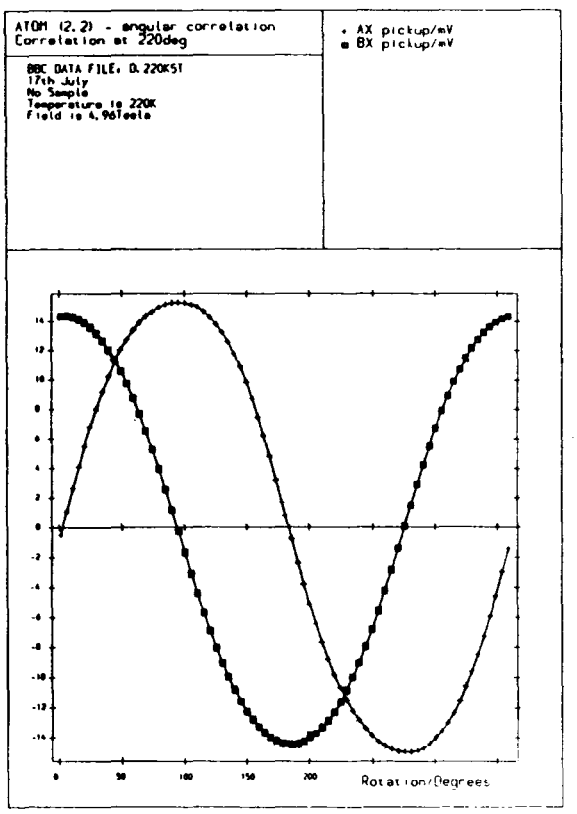


Fig 5.7 Pickup in the cross coils

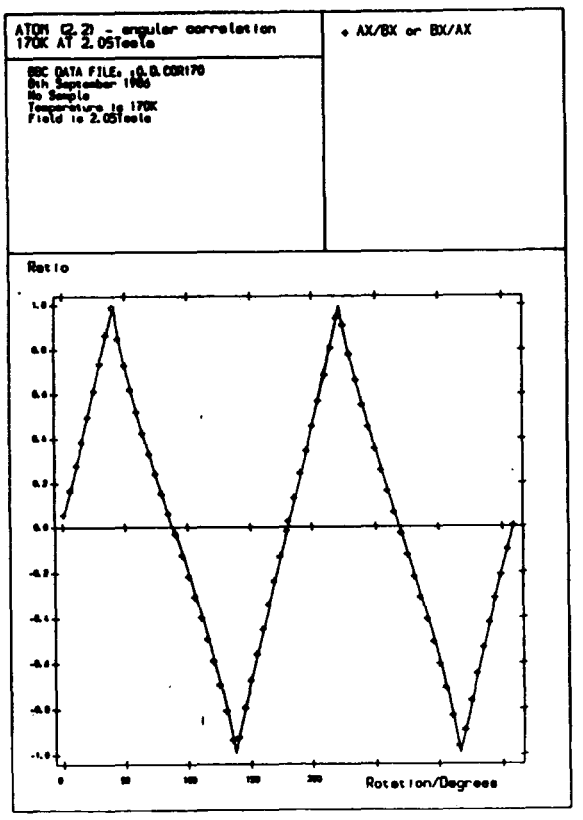


Fig 5.8 Ratio used for correlation

straight out, while both the calculated torque and the 'raw' measurements are stored on disc so that subsequent recalculation of the torque is possible with other correlation data. The programs for this are included in Appendix 2.

As mentioned above the BBC is connected to a Clearway ring through one port of the RS232 multiplexor. This gives it access to the plotter and a printer as well as to the University mainframe. The mainframe allows rapid analysis of the data collected using standard routines. These were used to extract the normal anisotropy constants. A least squares fitting routine was written using the minimization algorithm E04FDF from the Numerical Algorithms Group (NAG) suite of subroutines to fit a Fourier series to the data. From these Fourier coefficients the anisotropy constants can be calculated(Appendix 6). The orthogonality of the Fourier coefficients gives a much more sturdy program than direct fitting of anisotropy constants. The programs used are included in Appendix 2.

5.1.7. Calibration.

The sample needs no calibration against a standard sample, as the torque is measured directly as the interaction of known current passing round a coil in a known uniform field (sect 5.1.3). The cross section of the coils was measured using an integrating fluxmeter and a Bell gaussmeter. This was also compared with a 20 turn coil wound carefully on a cylindrical former so that the area was known to better than 0.5%. The difference from the value given by the fluxmeter was about 2% and as this was within the expected accuracy of the analogue integrating fluxmeter the value obtained using the 20 turn coil as a standard was used for the torque calculations. Linearity of the integrating fluxmeter was also checked and found to be better than 0.5%

5.1.8.Accuracy

5.1.8.1. Field measurement and stability

The field is measured by measuring the current in the leads to and from the magnet and relating this to the field by a factor given by the manufacturer and checked by M. A. Chaudri (1983). Measurement of the voltage across the solenoid connectors allows the current flowing in the leads to be accurately matched to that in the solenoid to better than 1 part in 10^{-4}. The uniformity of the field is described in section 5.1.2 and the decay is less than 1 part in in 10^{-4} per hour. All measurements were made in persistant mode when there is no fluctuation in the field intensity

5.1.8.2. Temperature measurement and stability

The temperature is measured at a position above the sample. This thermocouple is measuring the temperature of the helium after it passed the sample. A test was done to check that this temperature was indeed the same as the sample temperature. A series of measurements with a differential thermocouple, one junction strapped to the sample and the other on the normal thermocouple was performed. Over the region investigated, between 15 and 165K, the error in the measured temperature was up to 2.5K. However the average magnitude of the error was only around 0.5K and there was no systematic deviation in the temperature measured(see fig 5.9), the average error being 0.12K±0.12K.

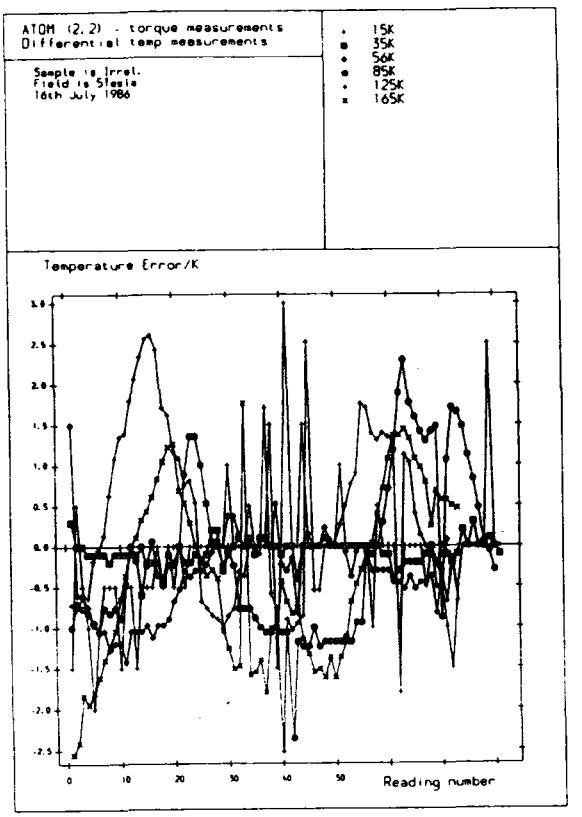


Fig 5.9 Temperature Error

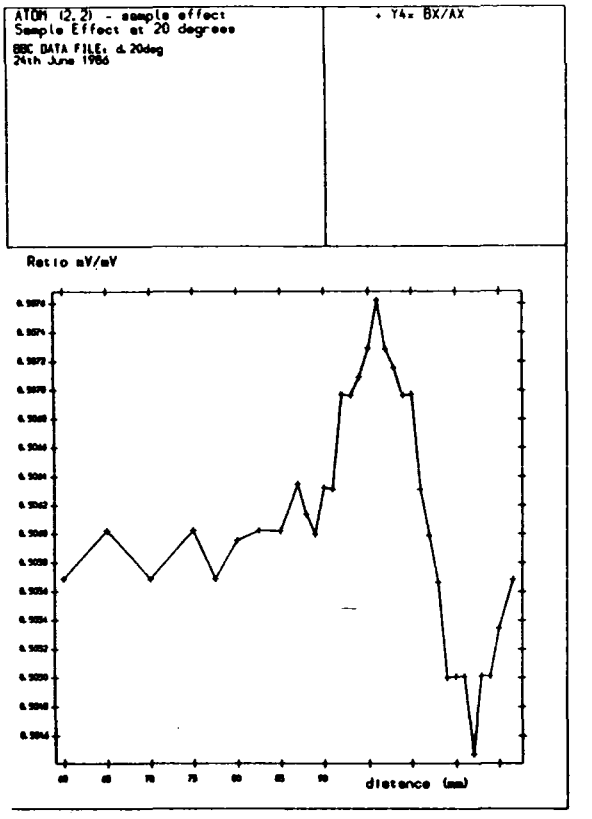


Fig 5.10 Effect of 0.93g soft $Nd_{15}Fe_{77}B_8$ sample on pickup ratio.

5.1.8.3. Angular Measurement.

The accuracy of the angle measured determines the accuracy of the torque measured. Checks were made on the independance of the AC pickup on other factors.

As pointed out in section 5.1.6 the temperature of the rig has a small effect on the calculated angle. This was allowed for by taking a series of correlation runs through the temperature range.

The sample may be assumed to have some effect on the mutual inductance of the coils, acting as a core for the 'transformer'. The significance of this effect was tested by inserting a large 0.93g soft $Nd_{15}Fe_{77}B_8$ ingot sample into the rig while this was held at a constant angle and watching the effect on the pickup. Although the pickup varied by up to 10%, the ratio did

not vary by more than 2%(e.g. Fig 5.10). This gave an error of up to 1° in the angular measurement. This was in itself not very satisfactory but the ingot sample used for these tests had ten times the mass of any sample used for the measurements. Moreover the ingot sample had a large initial susceptibility and the measurements were carried out on a lab bench, i.e. in zero field; in the solenoid at high field the suceptibility of the samples measured would be at least an order of magnitude less. These two arguments suggest that errors from the sample would be expected to be less that 1/100th degree.

The effect of the dc current on the pickup was also investigated and was found to be up to 0.6%, but more normally around 0.2%. This is probably the largest error in the angular determination and gives an error of up to 0.3°, but more normally 0.1°. It is probably due to high frequency interference penetrating the choke from the power supply and shows no systematic nature(see Fig 5.11).

5.1.8.4 Torque Values.

The accuracy of the torque values depends on the angular correlation and on the accuracy of measuring the current. The angular measurement may be up to 0.3° out, which means that if the sample has caused a deviation of, for example, 10° then the calculated torque may be up to 3% in error. However this is a random scatter and not systematic.

The measurement of the current passing in the coils is prone to an error of typically 0.1% due to the accuracy of the voltmeter, however the resultent counter-torque is calculated as a difference of two torques(sect 5.1.3) and this can typically multiply the error by 10. Therefore errors of the order of 1% are to be expected. If the current is very low for the range setting then the voltage measured by the Weston voltmeter will be correspondingly low and the error in the measured current will increase.

5.1.9. Performance

The final check on performance must come from the measurement of the anisotropy on a known sample. For this a Terbium single crystal sphere was used. This sphere had been grown by D. Fort at the University of Birmingham and had been used by Corner et al.(1985) for measurements on conflicing Rare Earth anisotropies using the high field facilities in the Laboratoire Louis Neél , Grenoble. The sample was very strongly anisotropic and could not be turned more than about 20° from the easy direction without the glue becoming unstuck. A typical intercept of the torque curve with the easy axis is given in Fig 5.12, along with a straight line fit. From the slope of the torque curve K_1 can be calculated and this was compared with the values obtained by Corner et al.(1985)(Fig 5.13). It was concluded that the values were in reasonable agreement below about 150K. Discussion of the difference at higher temperature can be found in chapter 7..

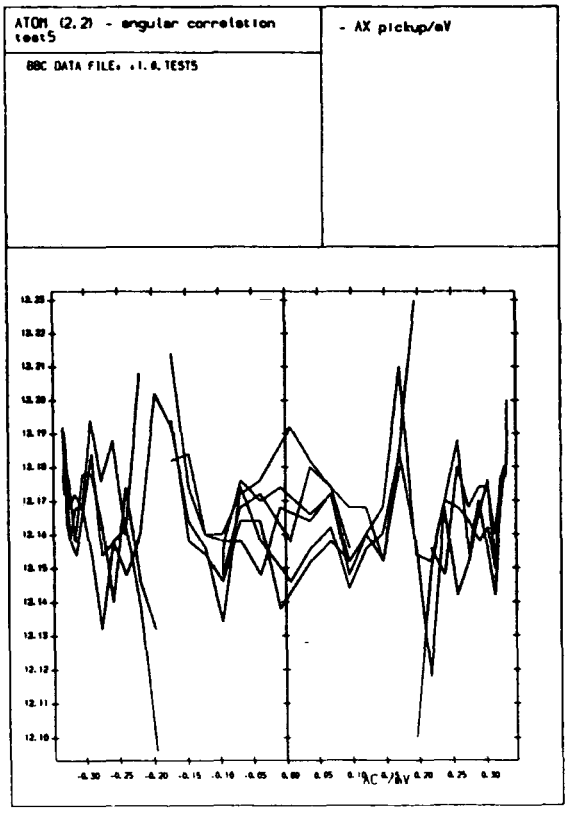


Fig 5.11 Dependance of pickup on DC current in coil

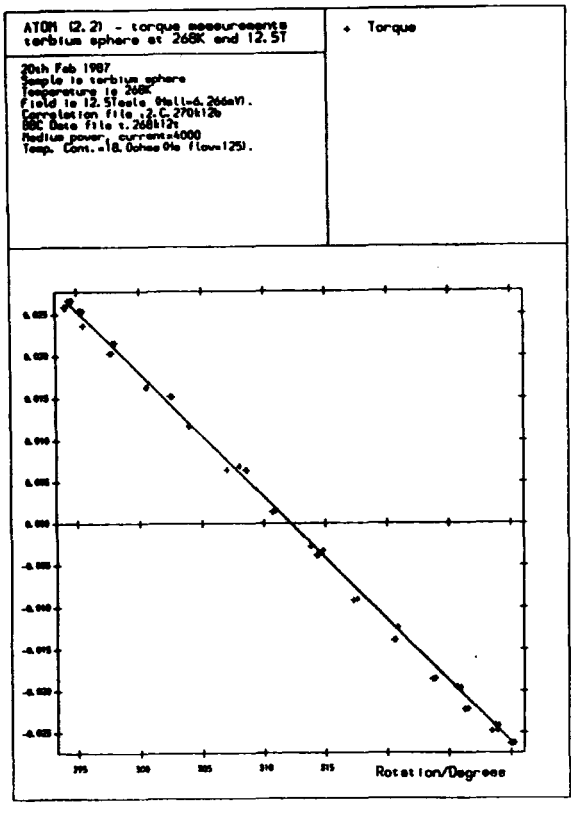


Fig 5.12 Typical Tb torque curve.

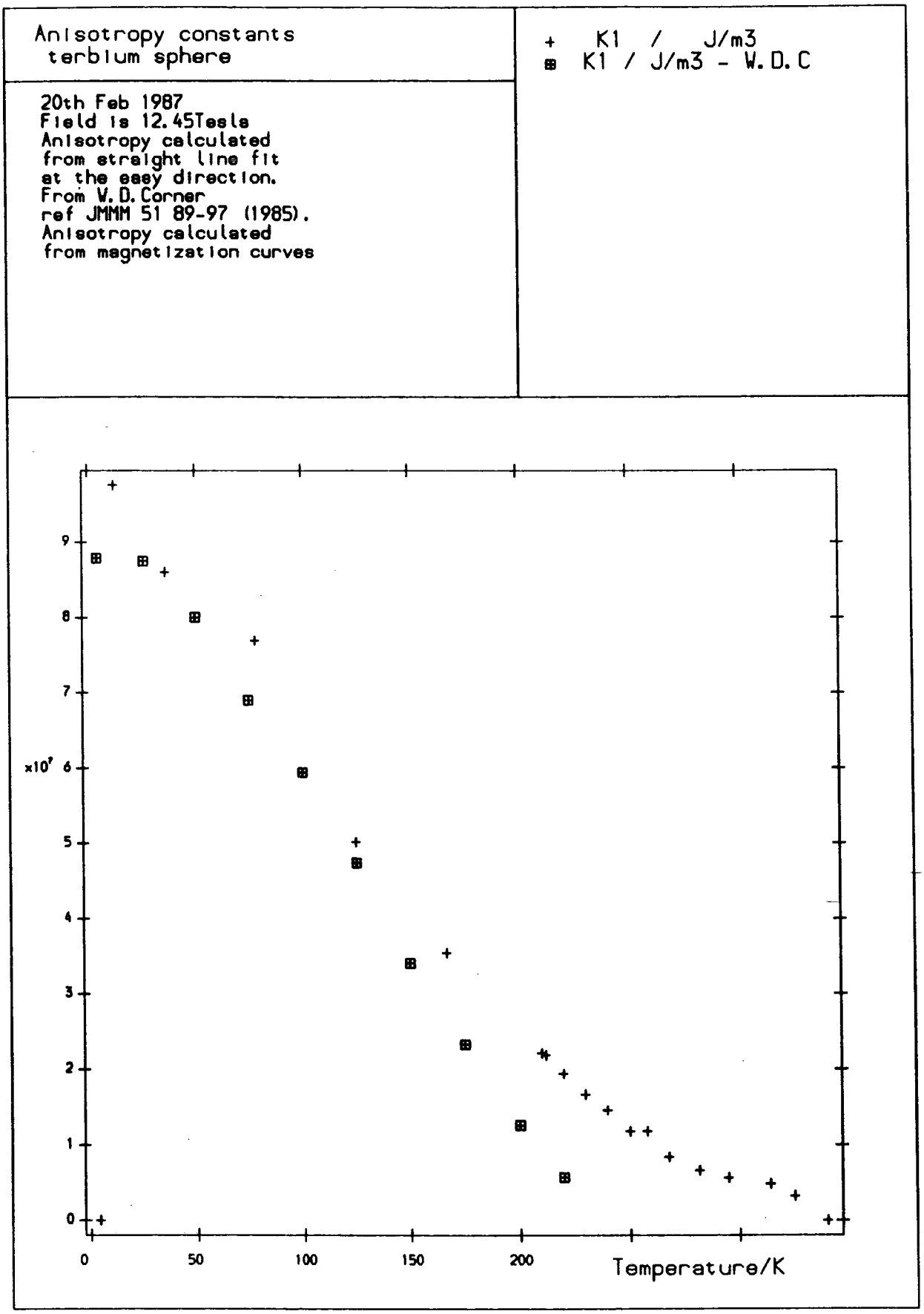


Fig 5.12 Temperature dependance of K_1 in Tb.

5.2 Optical Domain Observation.

5.2.1 Introduction

In order to study the magnetisation processes in $R_2Fe_{14}B$ it was decided to study the surface domain patterns using a ferrofluid to reveal the domains, as describe in section 1.3.2.4, and an optical microscope to observe them. This had the advantage of simplicity of sample preparation, and allowed rapid observation of moving domains, as the contrast obtained with the ferrofluid is far greater than with the Kerr effect allowing short exposure times and even recording on video film.

5.2.2 The Microscope

A Vickers metallurgical microscope was used. This was adapted to hold either a low intensity video camera or a 120 film cassette for still photography. A green filter(M173584) was used and a range of objective lenses from an X10 air lens with a N.A. of 0.25 to a X80 oil immersion lens with a N.A. of 1.32.

A microscope stage was constructed (see Fig 5.14) which allowed two translational movements and one rotational. This replaced the standard stage, but allowed the focus knob of the microscope to be used. It was made of non-magnetic brass and designed to fit between the pole pieces of an Oxford Instruments 1⅛" water cooled magnet. Special pole pieces were constructed for this magnet with parts of the pole tip removed to allow the objective to get close to the sample while this remained in the region of maximum field. This introduced a large inhomogeneity into the field, but meant that continuous applied fields of up to 1.25Tesla could be obtained.

5.2.3 Experimental Detail.

The sample was first polished(section 4.2.1) and then mounted on the stage with a strong adhesive. A ferrofluid was made of fine particle magnetite in petroleum ether with Sonsperse 3000 as surfactant. A drop of this fluid was dropped onto the sample and then this was covered with a small 5mm disk made from microscope cover slide. The domain pattern was then either observed directly with an air lens, or an oil immersion lens was used, the cover slide keeping the oil and ferrofluid separate(Fig 5.15).

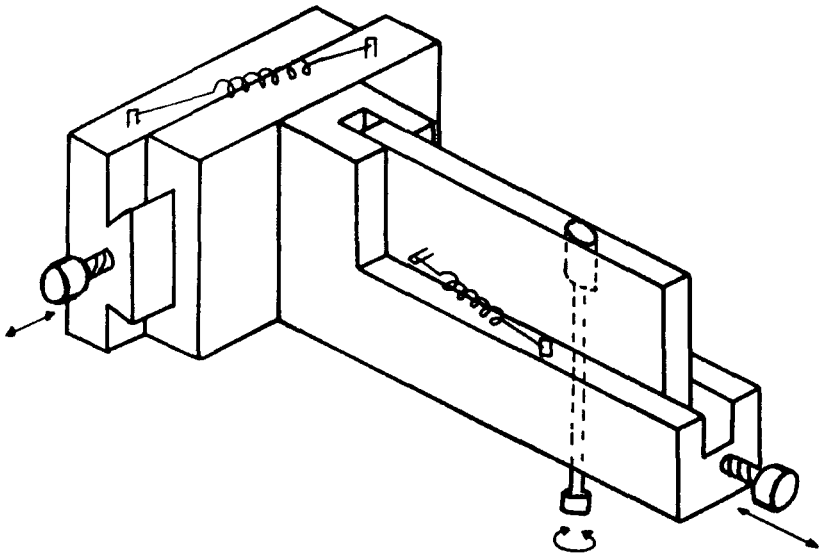


Fig 5.14 Microscope stage for magnet.

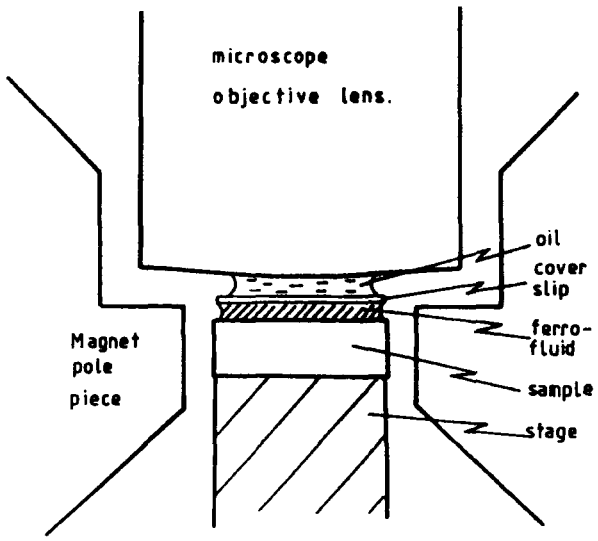


Fig 5.15 Sample set-up for domain observation with ferrofluid.

5.3 Vibrating Sample Magnetometer

A vibrating sample magnetometer (VSM) was used to measure the moment of the samples in order to apply the shear correction(Appendix 8) to convert the measured field angle to the angle of the moment from the easy axis.

This VSM was designed by S.R.Hoon to operate both in an electromagnet split pair and in the DUCHESS. The instrument and its automation is described by Willcock(1985) and also by Lambrick(1986). The signal induced by a vibrating nickel reference sample is compared with that induced by the sample to be measured vibrating at the same frequency and in the same position with respect to the pick-up coils and the moment is calculated from the ratio. Corrections are needed for the diamagnetic moment of the sample holder, and values obtained by Lambrick and Hoon(1987) were used.

The temperature is measured with a 0.07%Au-Fe / Chromel thermocouple, using liquid nitrogen as a reference. Values measured by Lambrick and Hoon(1987) were fitted to give a straight line fit above 77K and a quadratic below 77K. This gives:

$$\text{Temperature/K} = 77.4 + 48.075 * \text{Voltage/mV} \quad \text{... above 77.4K} \qquad (5.4)$$

and:

$$\text{Temperature/K} = 77.4 + 48.075 * \text{Voltage/mV} - 9.3926 * \text{Voltage/mV}^2 \quad \text{... below 77.4K.} \qquad (5.5)$$

6.1. Magnetisation Measurements.

These were done with a vibrating sample magnetometer (see section 5.3). The coils mounted in the sample space were used in preference to those in the helium bath.

The signal derived from a reference sample(Ni) was measured first to give a calibration for the instrument when measuring samples of this size.

A calibration factor was calculated at three temperatures, using the diamagnetic signal of the sample holder measured by Lambrick and Hoon (1987) of $-3.82\mu VT^{-1}$ and values for magnetisation of Nickel from Crangle (1977)(see Table 6.1).

The difference in the values for different temperatures was within the experimental error and the error in the value used for the magnetisation of nickel, so a calibration factor of 0.000812 J/T/mV was taken, independent of temperature. This implies that any effects due to changing resistance of tubes and radiation baffles must be small. The magnetisation of nickel as a function of temperature calculated using this factor is compared with the three calibration points in fig 6.1.

Temperature. Kelvin	Signal mV	Corrected Signal mV	Magnetisation Nickel * $Jkg^{-1}T^{-1}$	Calibration. Factor $JT^{-1}mV^{-1}$
4.2	5.59	5.638	58.6	0.0008131
77	5.567	5.624	58.4	0.0008123
293	5.262	5.31	55.1	0.0008118

Table 6.1 Calculation of calibration factor for Duchess VSM using coils in sample space and nickel reference disk(Ni) of 0.07823g(see Table 4.1).

* Crangle 1977.

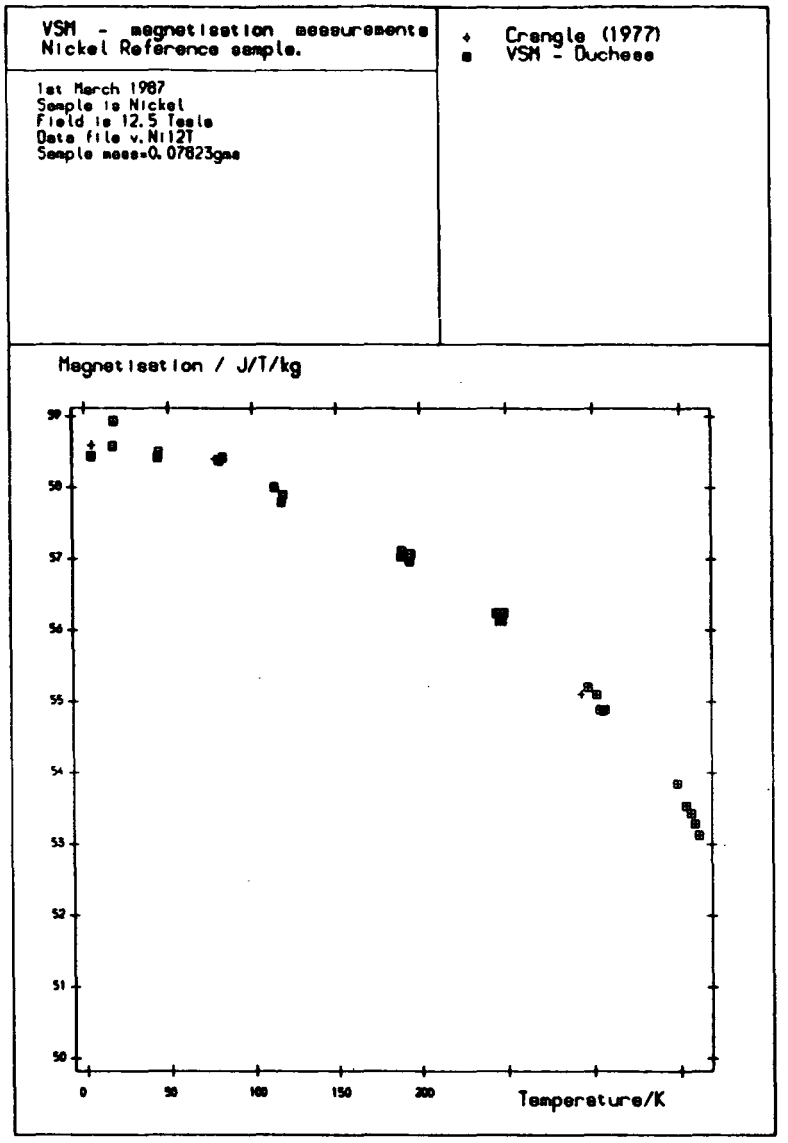


Fig 6.1 Magnetisation of nickel(NI) at 12.5T.

6.1.1. $R_2Fe_{14}B$ crystals and aligned powders.

Magnetisation of single crystals of $R_2Fe_{14}B$, R=Dy,Gd (D3,G3,G4), aligned powders of $Ho_2Fe_{14}B$(H4,H5,H5B) and $Nd_{1.7}Fe_{14}B$(N6,N7) and multigrain samples of $Ho_2Fe_{14}B$(H3) and $Nd_{1.7}Fe_{14}B$(N8) were measured at 12.5 Tesla. The results are given in Fig 6.2.

The measurements are all with the easiest direction within the plane of the disk aligned with the field direction

For the $Nd_{1.7}Fe_{14}B$ samples the results are compared with other workers in Fig 6.2a. The three sets of readings obtained are essentially of the same shape but have different absolute magnitudes due to the uncertainty in the

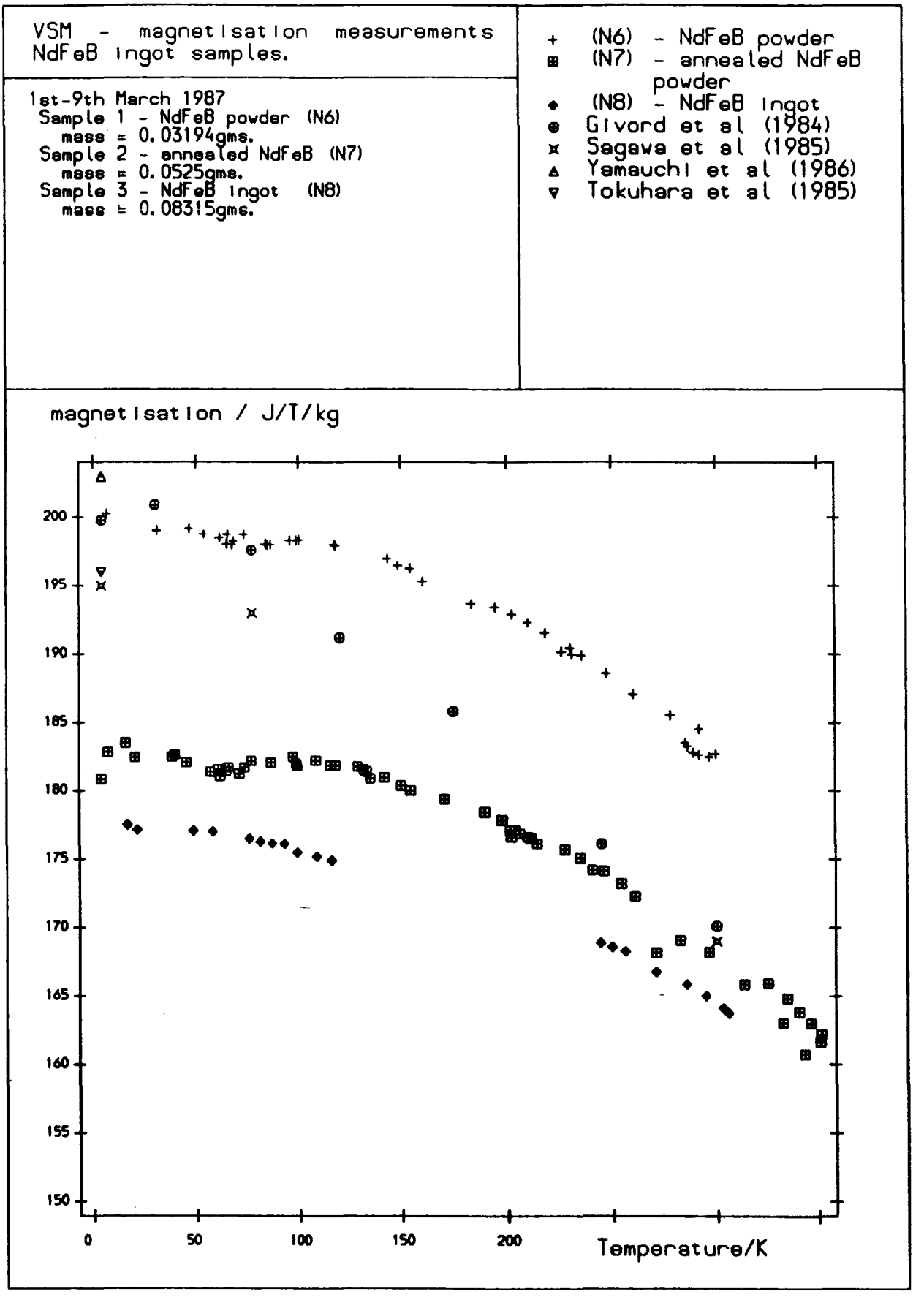


Fig 6.2a $Nd_{1.7}Fe_{14}B$ magnetisation

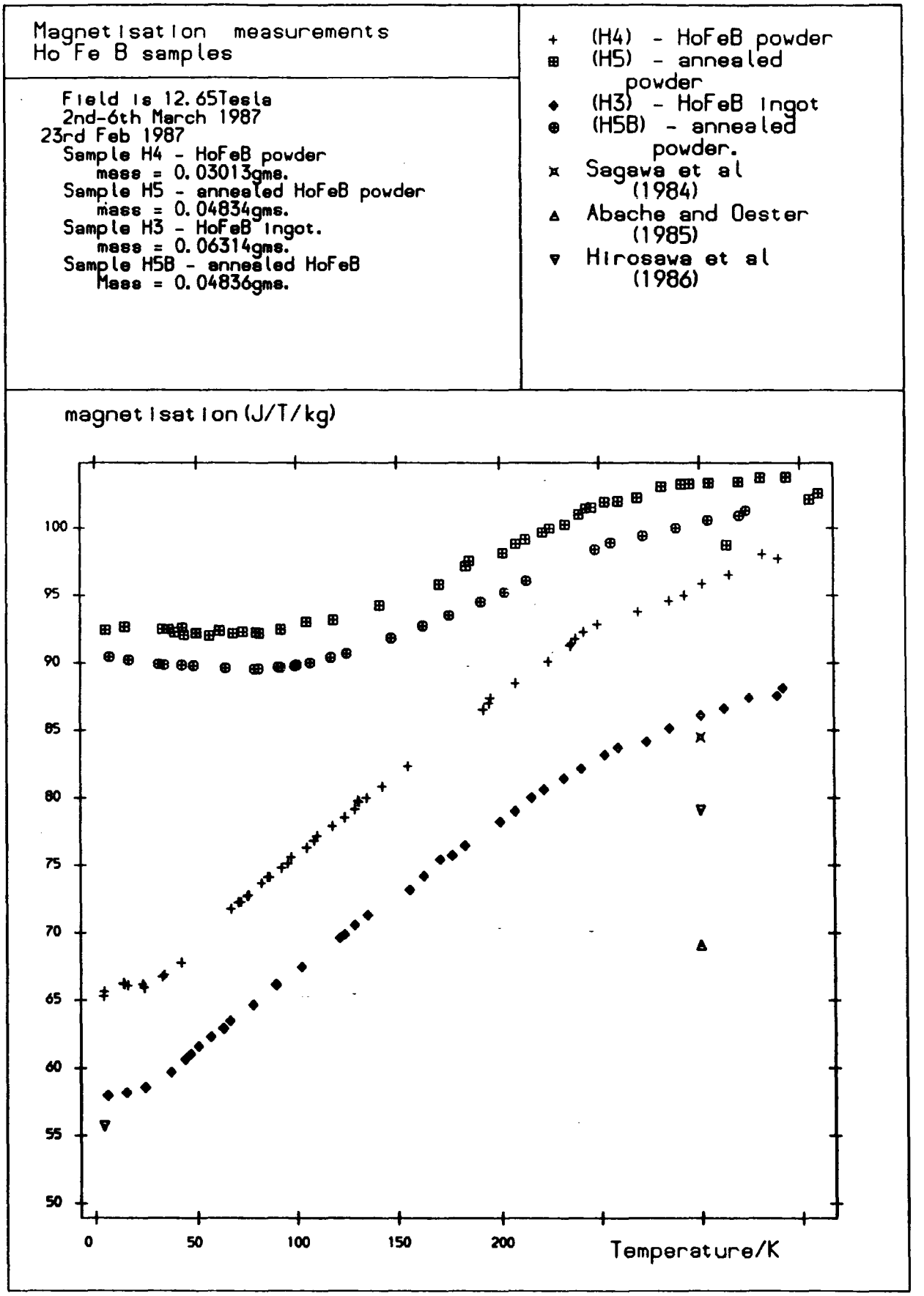


Fig 6.2b $Ho_2Fe_{14}B$ magnetisation.

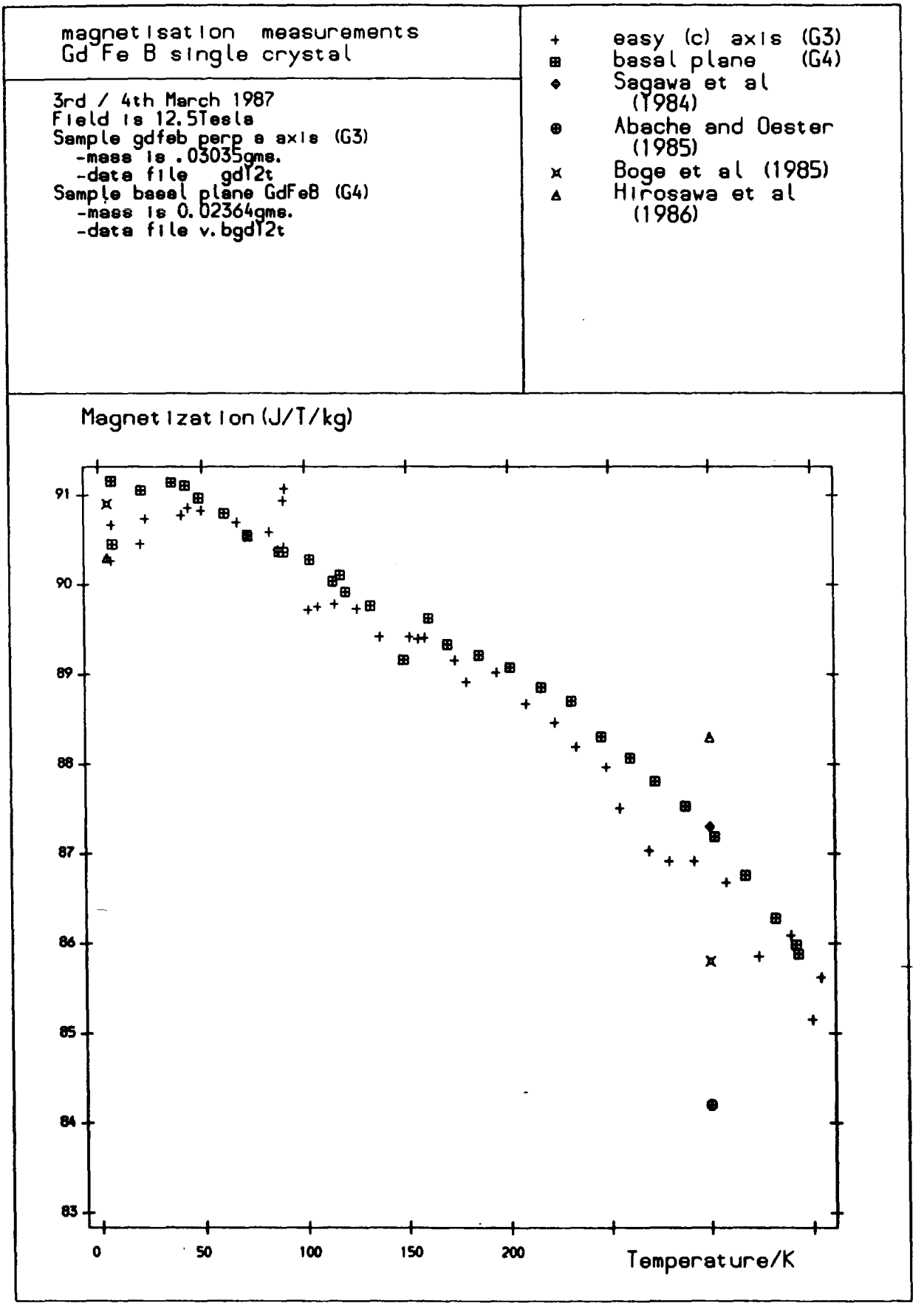


Fig 6.2c $Gd_2Fe_{14}B$ magnetisation

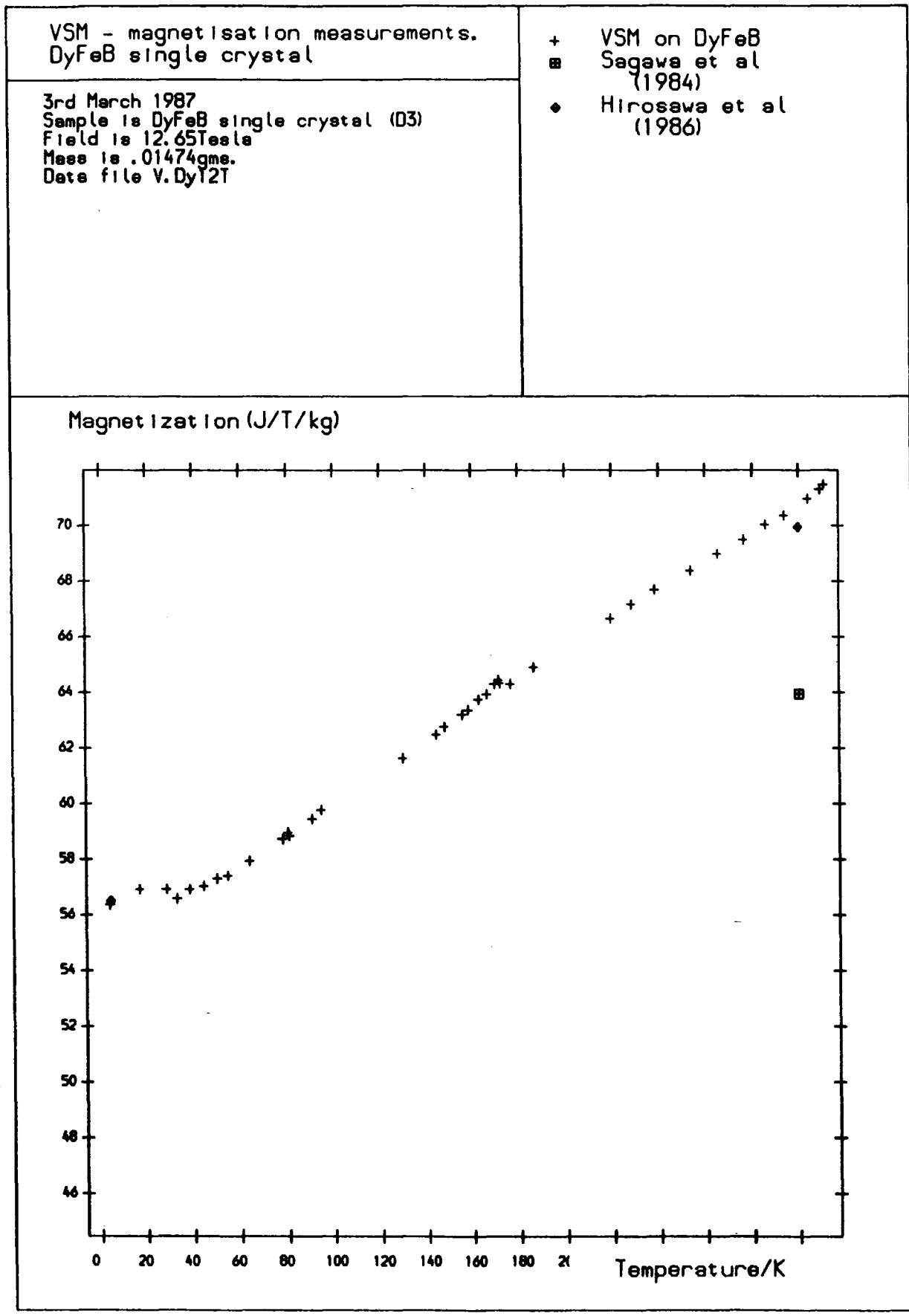


Fig 6.2d $Dy_2Fe_{14}B$ magnetisation

amount of material set within any one sample. Powder was mixed in a known ratio with epoxy and this was then divided between the sample moulds, leading to a large uncertainty in the amount of powder in each sample. Comparison with the magnetisation of the ingot was used to recalculate the amount of material in each specimen for anisotropy measurements.

Comparison of the magnetisation with other workers shows a lower temperature coefficient, although the actual values are quite similar. This is probably due to the presence of a significant quantity of free iron, which has a magnetisation of 221.7 $JT^{-1}kg^{-1}$ at 4.2K, which drops by 2% to 217.6 $JT^{-1}kg^{-1}$ at room temperature. This is much less than the 15% fall observed by Givord et al(1984b) and the 13% observed by Sagawa et al(1985) for $Nd_2Fe_{14}B$. In the samples measured here the magnetisation fell by 8%between 4.2K and room temperature.

For the $Ho_2Fe_{14}B$ samples the results are compared with other workers in Fig 6.2b.

The ingot (H3) compares well with the results of Hirosawa et al(1986) at low temperatures. The magnetisation at room temperature is closer to that observed by Sagawa et al(1985) and 7% larger than that observed by Hirosawa et al(1986). The value observed by Abache and Oesterreicher(1985) measured on aligned powder is some 20% lower than the value we observe.

The powdered sample (H4) follows a similar temperature dependence to the ingot, but 11% larger. This is again due to the uncertainty in the amount of powder in the sample.

The annealed powder samples(H5, H5B) show a larger magnetisation at low temperature which falls off slightly up to about 80K and then rises becoming almost flat again by room temperature. Both samples behaved similarly. It is clear that some reaction took place during the annealing

process. The sample had already been annealed in ingot form for 45 days at the same temperature, so the short 2 hour anneal of the powder was not expected to have any effect.

The shape suggests that two other magnetic phases may be present, one with a Curie or Neel temperature of about 120K and one with a large Curie temperature (>> 300K). It is however difficult to suggest what these may be, as all known holmium-iron intermetallics order ferrimagnetically with Curie temperatures 325-614K (Table 2.5). Another possibility is a subtle phase change quenching the Ho moment at low temperature. Further investigation would be required to explain these curves.

The $Gd_2Fe_{14}B$ crystals(G3, G4) showed a slight decrease in magnetisation with increasing temperature, fig 6.2c. The results for both samples are essentially identical, reflecting the weak anisotropy. The low temperature value compares well with that of Bogé et al(1985) and Hirosawa et al(1986), all the values being within 1% of each other. The room temperature value was similar to that of Sagawa et al(1984) and lay half way between those of Bogé and Hirosawa but only 1.5% different from each. The value of Abache and Oesterreicher(1985) was another 2% lower than that of Bogé et al.

The magnetisation measured on the $Dy_2Fe_{14}B$ crystal(D3) (see fig 6.2d) increased in temperature in a form similar to the holmium samples. The results are similar to those of Hirosawa et al(1986). The room temperature result of Sagawa et al(1984) is some 10% lower.

6.1.2. Rare earth iron magnets.

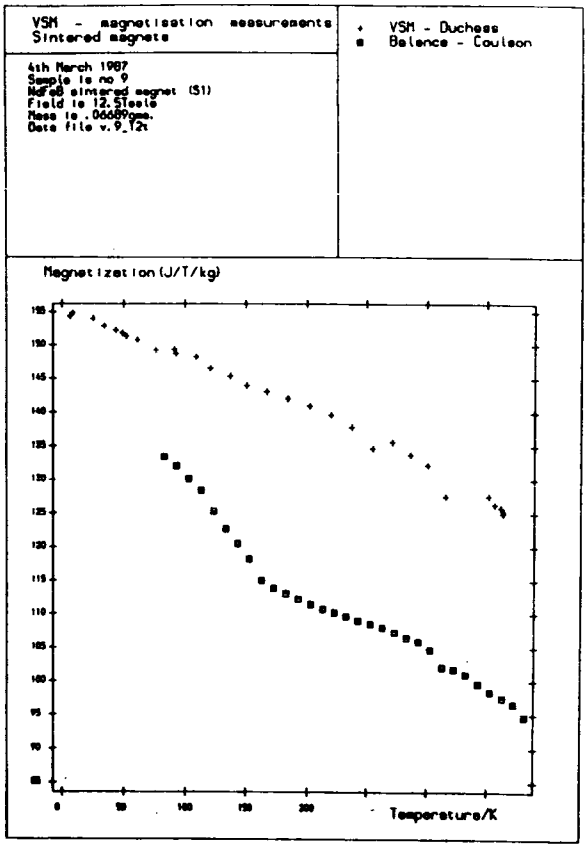


Fig 6.3a Magnetisation $Nd_{15}Fe_{77}B_8$ magnet.

Three sintered samples (S1,S2,S3) with increasing cobalt content were measured at 12.5 Tesla and the results are compared with results from a balance at 1.75Tesla(Coulson 1987) in Fig 6.3. Both sets of readings were taken with the aligning field along the easy direction.The magnetisation at 12.5T is significantly higher than those measured at a lower field. This may be due either to a poor alignment of the grains within the

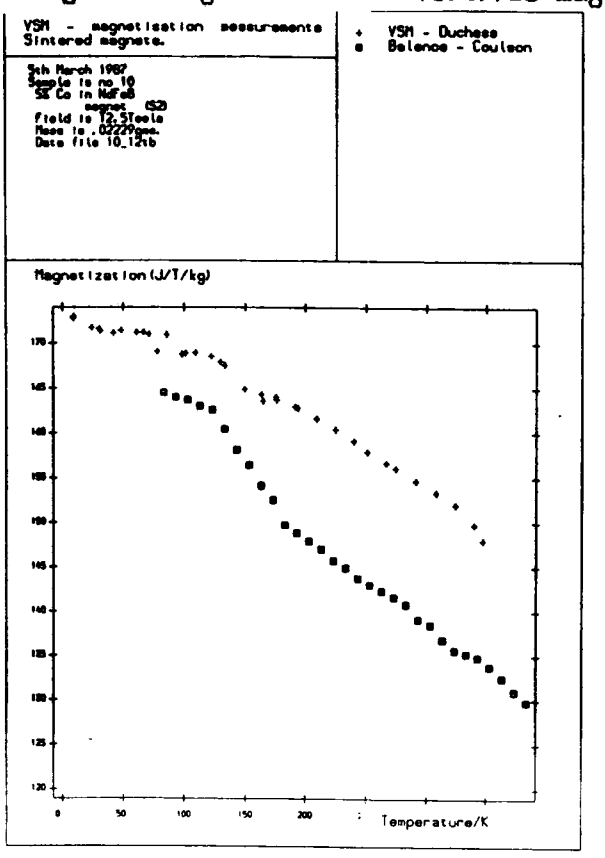


Fig 6.3b Mag. 5% cobalt magnet.

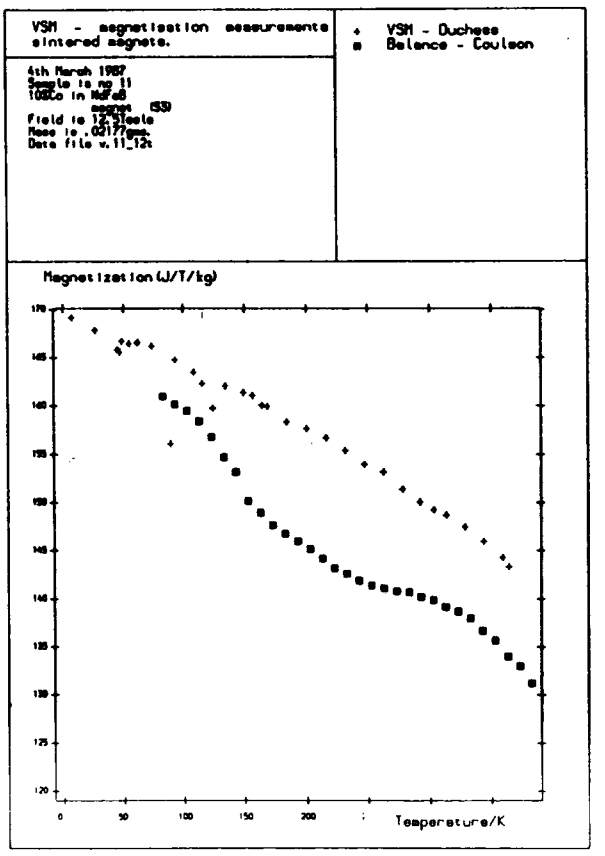


Fig 6.3c Mag. 10% cobalt magnet.

magnet, meaning that the resultant moment is lowered at a low field, whereas the larger field is able to pull the moment away from the easy direction within a misaligned grain, or due to hysteresis in the magnetisation loop meaning that saturation is not being reached by 1.75T.

6.1.3. Terbium.

The magnetisation of a terbium sphere(T1) was measured at 12.5 Tesla and compared with the spontaneous magnetisation measured by Corner(1985). The results are given in Fig 6.4.

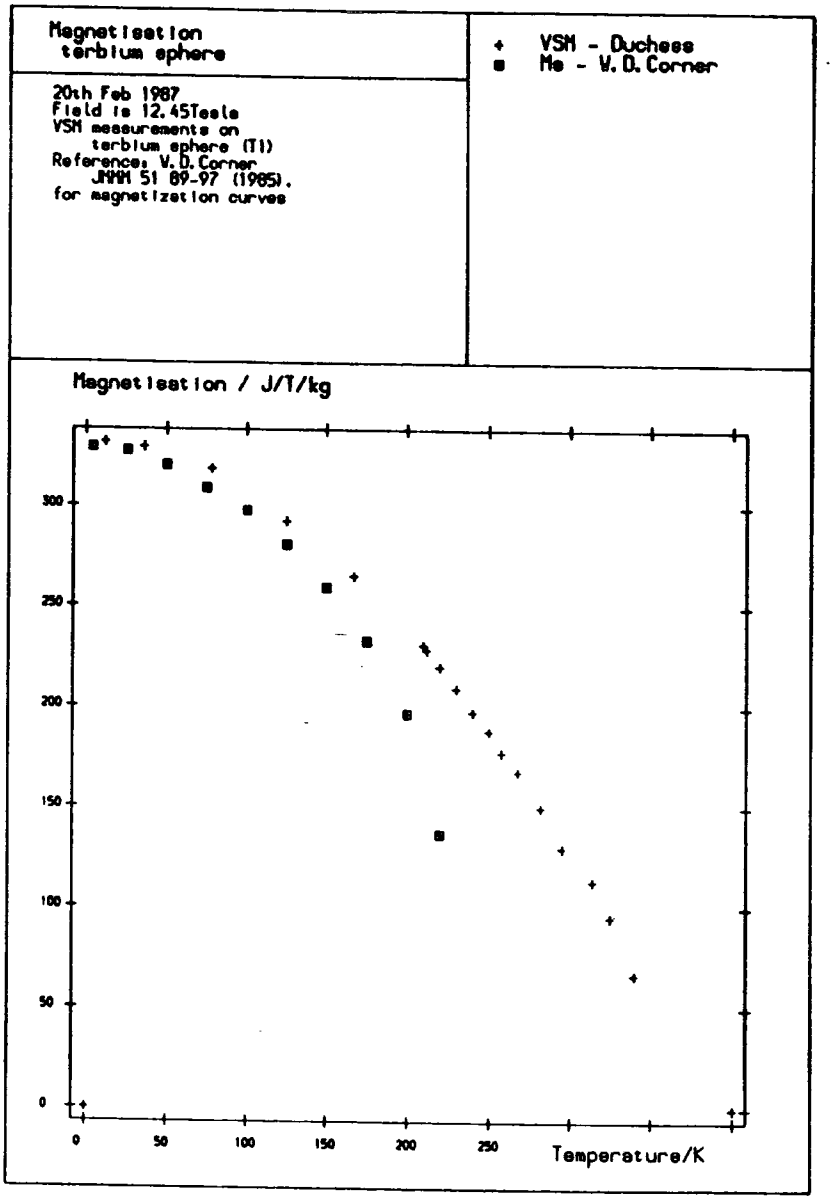


Fig 6.4 Magnetisation of terbium

A simple theory in terms of equating the magnetic energy associated with the applied field with the kinetic energy at each temperature suggests that the Curie temperature should be shifted by a temperature given by:

$$k.\Delta T = \mu_J \, . \, B \qquad (6.1)$$

or

$$\Delta T = g \, \mu_B \, [\, J \, (\, J + 1 \,) \,]^{\frac{1}{2}} \, . \, B \, / \, k \qquad (6.1b)$$

where μ_J is the magnetic moment along the J direction, J is the angular momentum quantum number, g is the Landé g factor, μ_B the Bohr magneton and k is Boltzmann's constant.

For terbium L=3, S=3, J=6 giving g=3/2 and therefore ΔT=82K at 12.5T. This is the right order of magnitude, as the data suggests an increase of about 100K in the observed Curie temperature. At temperatures below the Curie temperature the alignment is better and the number of effectively available states decreases so that ΔT reduces to zero at 0K.

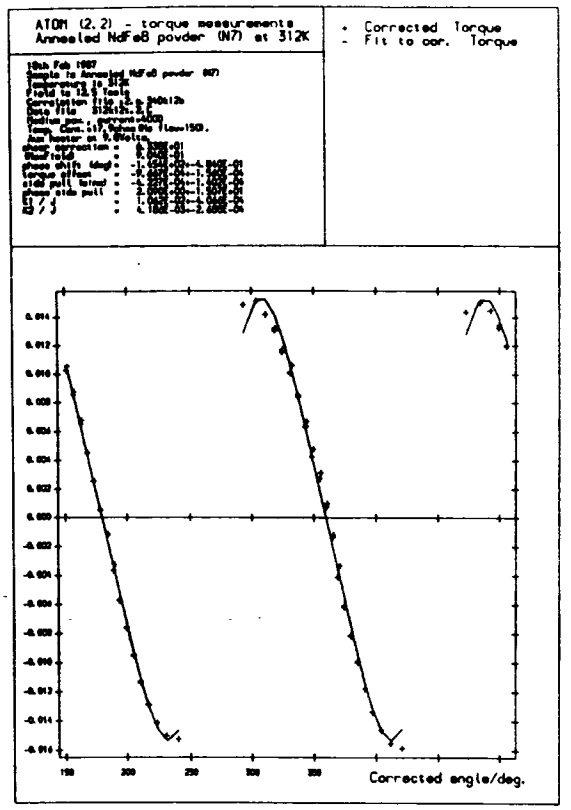
6.2. Anisotropy Measurements.

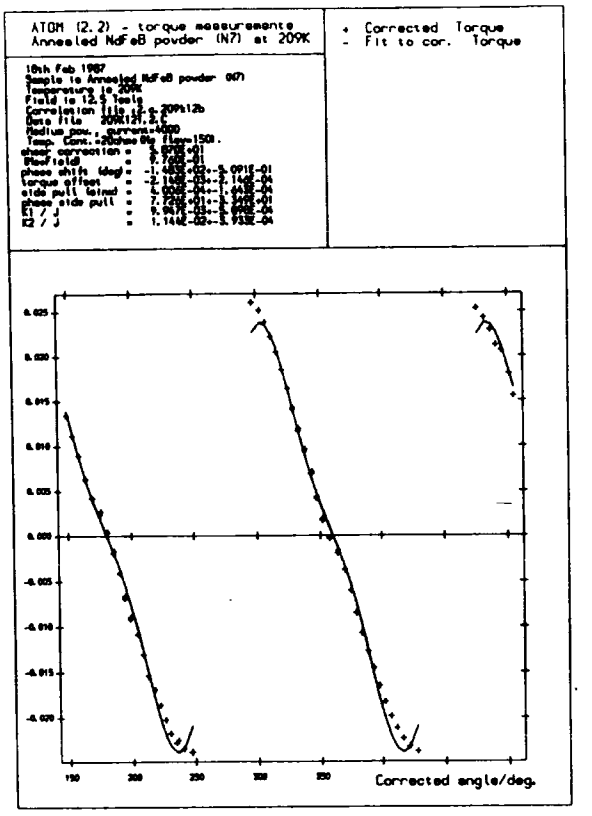
Torque curves were measured on samples using the instrument described in section 5.1. The information obtained was transfered to the University Amdahl mainframe. It was corrected for the shear correction(Appendix 8) and a sin(θ) term which is similar to the side pull term in a conventional torque magnetometer and arises due to the sample not being positioned exactly on the rotation axis of the instrument. These corrections were done while fourier coefficients were fitted and anisotropy constants were calculated from these fits. The values obtained depended on the number of anisotropy constants fitted, and for different samples a number of constants was chosen to give a good fit while not including spurious data. The criterion used was the smoothness of the temperature dependence of the constants.

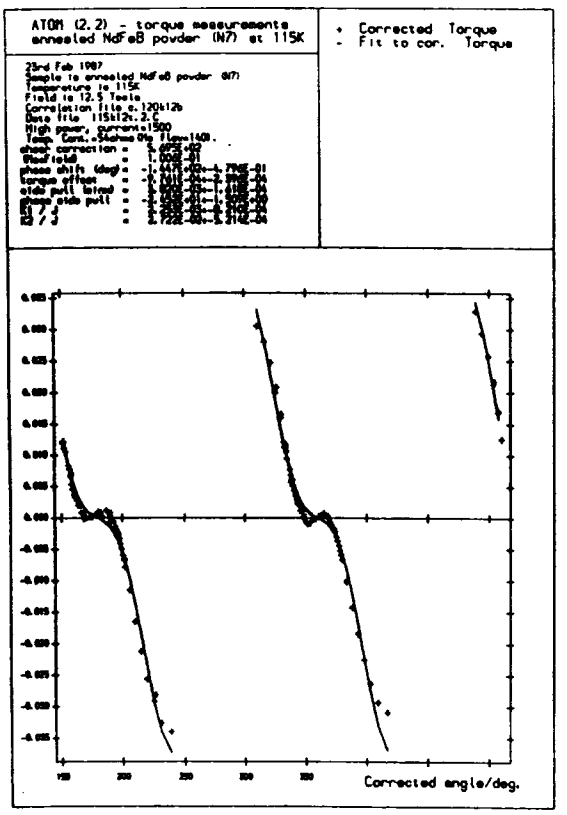
6.2.1. $R_2Fe_{14}B$ crystals and aligned powders.

6.2.1.1 $Nd_{1.7}Fe_{14}B$ ingot and powders.

Torque curves were measured on samples of powdered NdFeB and a least squares fit was made to extract the anisotropy constants. Examples of these torque curves are shown in fig 6.5. They have been corrected for side pull and plotted against the angle of the moment within the sample, the latter being calculated using the magnetisation measurements reported in section 6.1 on the assumption that the torque is due to the total moment measured. At high temperatures, above about 250K, there is no observable deviation from a simple uniaxial anisotropy, with a $\sin^2(\theta)$ torque curve. Below this temperature a kink begins to develop around the easy direction(c axis) and this kink continues to increase in size until at about 130K the kink has a stationary point and then below this a maximum and minimum. This means that there are two easy directions, one on either side of the c axis,







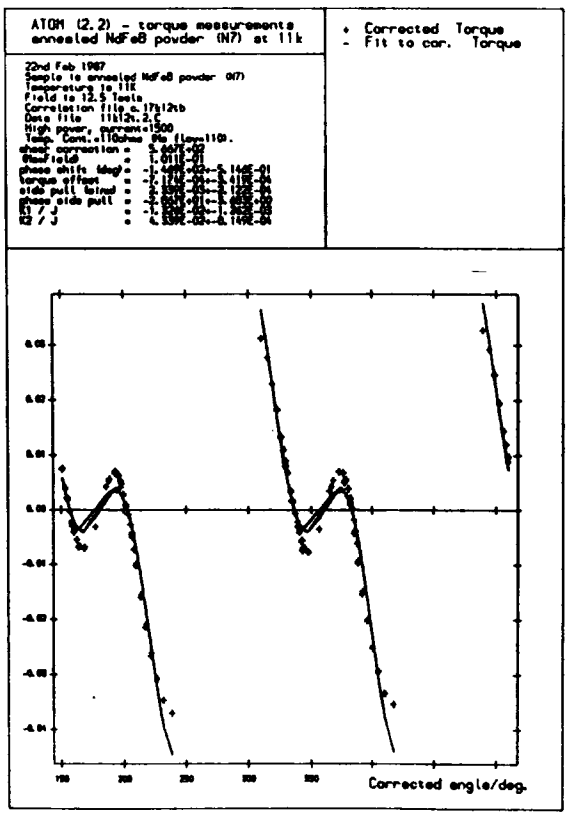


Fig 6.5 Torque curve of $Nd_{1.7}Fe_{14}B$

corresponding to an easy cone system. The angle of the cone can be easily read off and is plotted in figure 6.6.

The curves were fitted using the least square method to an expression including two anisotropy constants the result being indicated by the full lines in Fig 6.5. As it can be seen the fit is not very good at low temperature, the sharpness of the kink in the torque curve which develops below about 130K is not fittable with the restriction to two fourier coefficients. Addition of higher coefficients improves the fit, but the quality of the rest of the data results in a larger scatter of calculated anisotropy coefficients.

The first anisotropy constant can also be derived from the slope of the torque curve at the $\theta=0$ direction or, in the case of an easy cone system, the first two anisotropy constants can be derived from the slope of the torque curve at the easy direction and the cone angle (see appendix 7). This is on the assumption that any further anisotropy constants are small and insignificant.

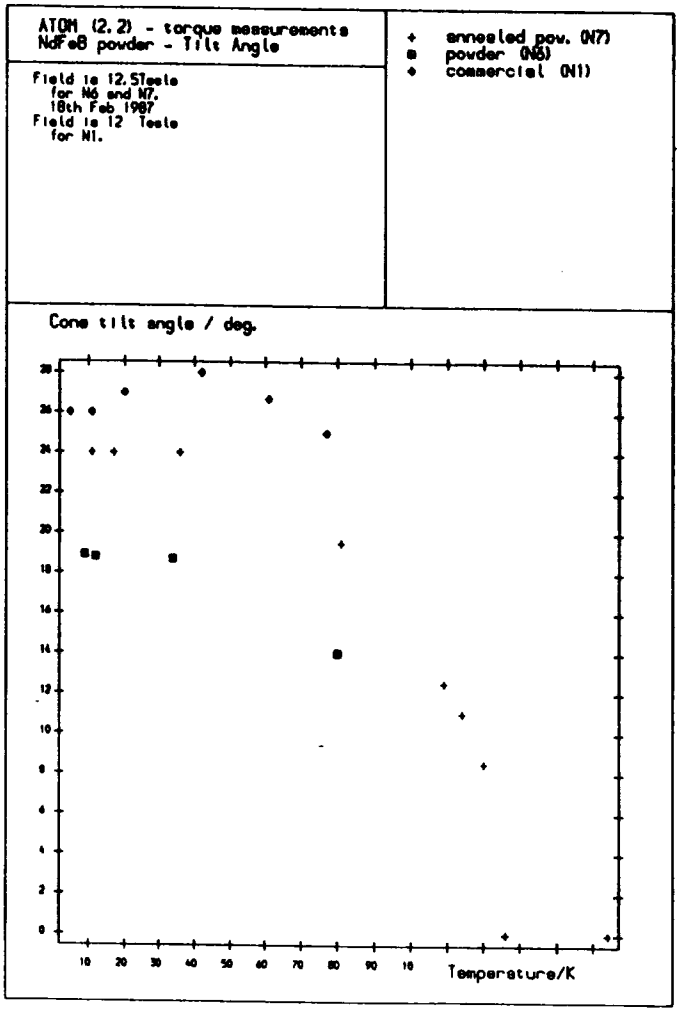


Fig 6.6 Tilt angle measured on torque curve of $Nd_{1.7}Fe_{14}B$

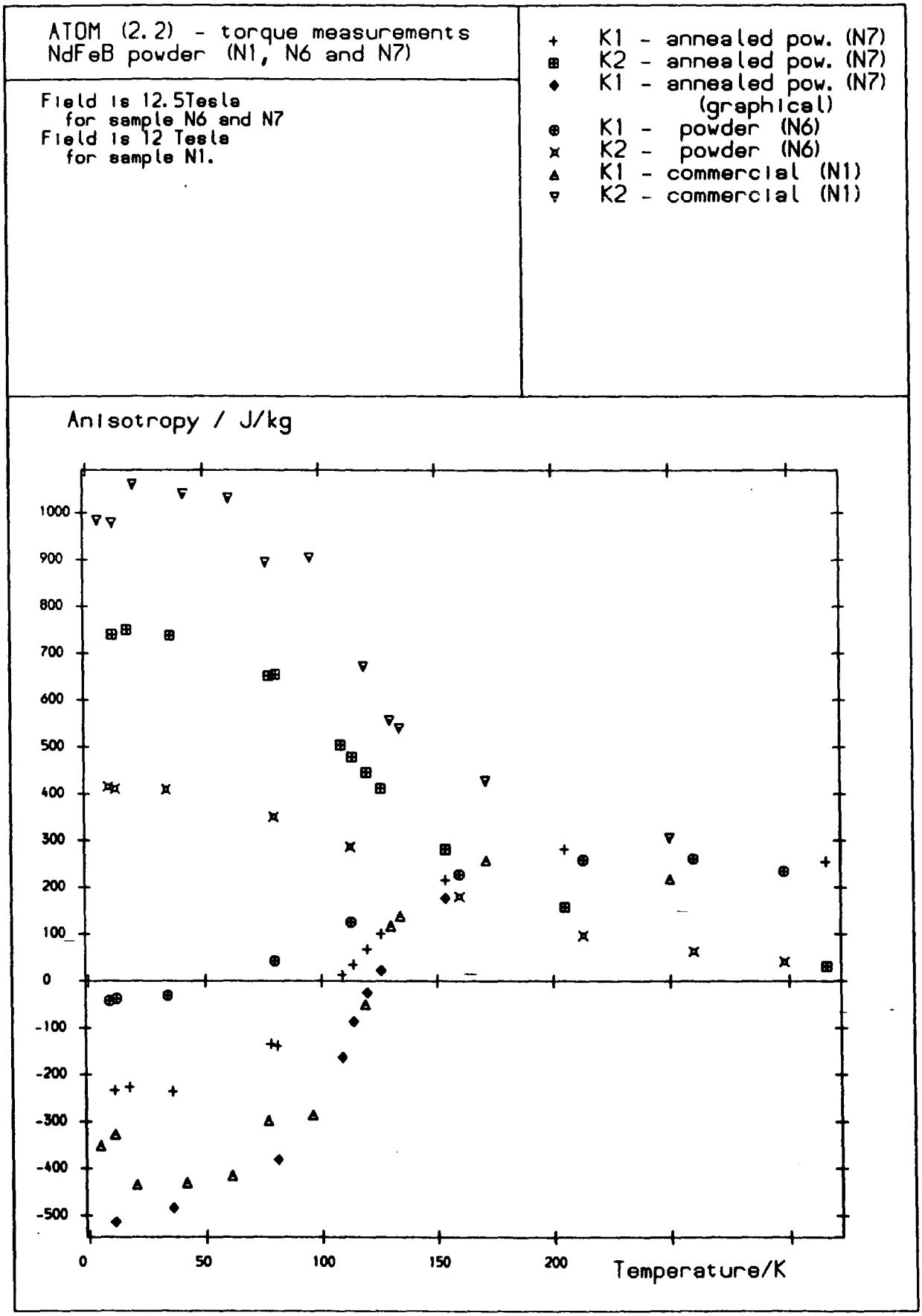


Fig 6.7 Anisotropy constants from torque curve of $Nd_{1.7}Fe_{14}B$

This graphical method was used for the torque curves measured on sample N6. Results for anisotropy constants derived from torque curves measured on the samples N1, N6 and N7 (see Table 4.1) are given in Fig 6.7. The shape of the curves are similar, but they vary considerably in magnitude. This is due mainly to the quality of the samples.

The errors originate in a number of ways:

a) None of the samples are single phase which means that the shear correction for the angle, based on the total magnetic moment, is incorrect. In this material at around 10 Tesla the angle between the measured angle and the moment is about the same as that between the moment and the easy direction. This means that any error in the moment contributes directly to the error in the anisotropy. Any other phases present also contribute to the error due to the error in the actual amount of the $Nd_2Fe_{14}B$ phase. The other phases are estimated to be up to 20% of the material giving an error of up to 40%.

b) The powder is not single phase, and the grains were very small in the material, this means that the powder probably contains particles with more than one grain within a particle. The annealed powder is more likely to contain single grain particles.

c) The powder may not be completely aligned, due in part to the shape of the powder grains which were very angular and therefore not very free to rotate against each other.

d) Within a set aligned powder sample there is texture, adding a shape anisotropy which it was not possible to calculate and allow for.

e) The non orthogonality of the anisotropy constants means that a relatively small error may result in a larger reduction in one constant and an increase in the other

With these errors in mind these results can only be taken as an approximation to the values, showing the trends but not the absolute values.

6.2.1.2 $Ho_2Fe_{14}B$ ingot and powders.

Torque curves on samples H3 and H4 were analysed to extract anisotropy constants. H5 and H5B were not included due to the uncertainty in the results of the annealing(see section 6.1.1).

The curves observed for HoFeB were essentially the same as those observed for NdFeB; a kink developed below about 70K and this became acute enough to cause an easy cone below about 42K. An example of a torque curve is given in Fig 6.8 and the angle of the cone measured from the torque curves is given in Fig 6.9. This only rises to a maximum cone angle of about 20°, somewhat less than the angle of 27° observed for NdFeB.

The curves were analysed by the least squares method and also graphically, all the results are included in Fig 6.10. At low temperatures the least squares method didn't fit the curves near the easy axis very well and the graphical result are more accurate. The values for the powder sample(H4) agree well with the ingot sample.

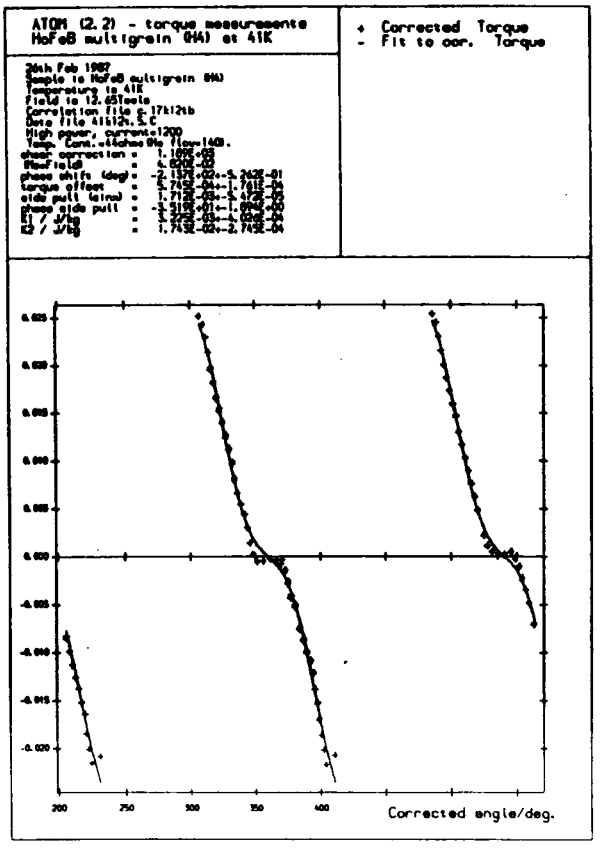


Fig 6.8 Torque curve of $Ho_2Fe_{14}B$

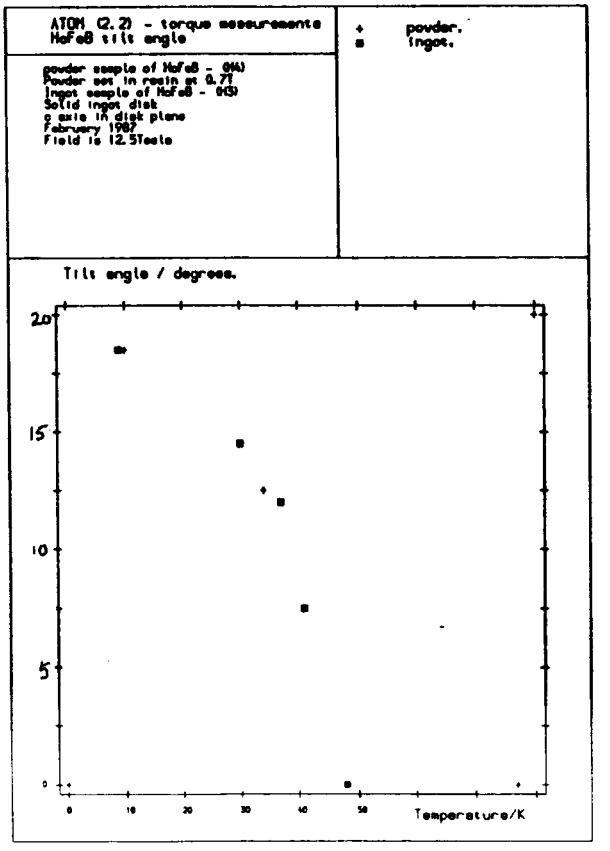


Fig 6.9 Tilt angle of $Ho_2Fe_{14}B$

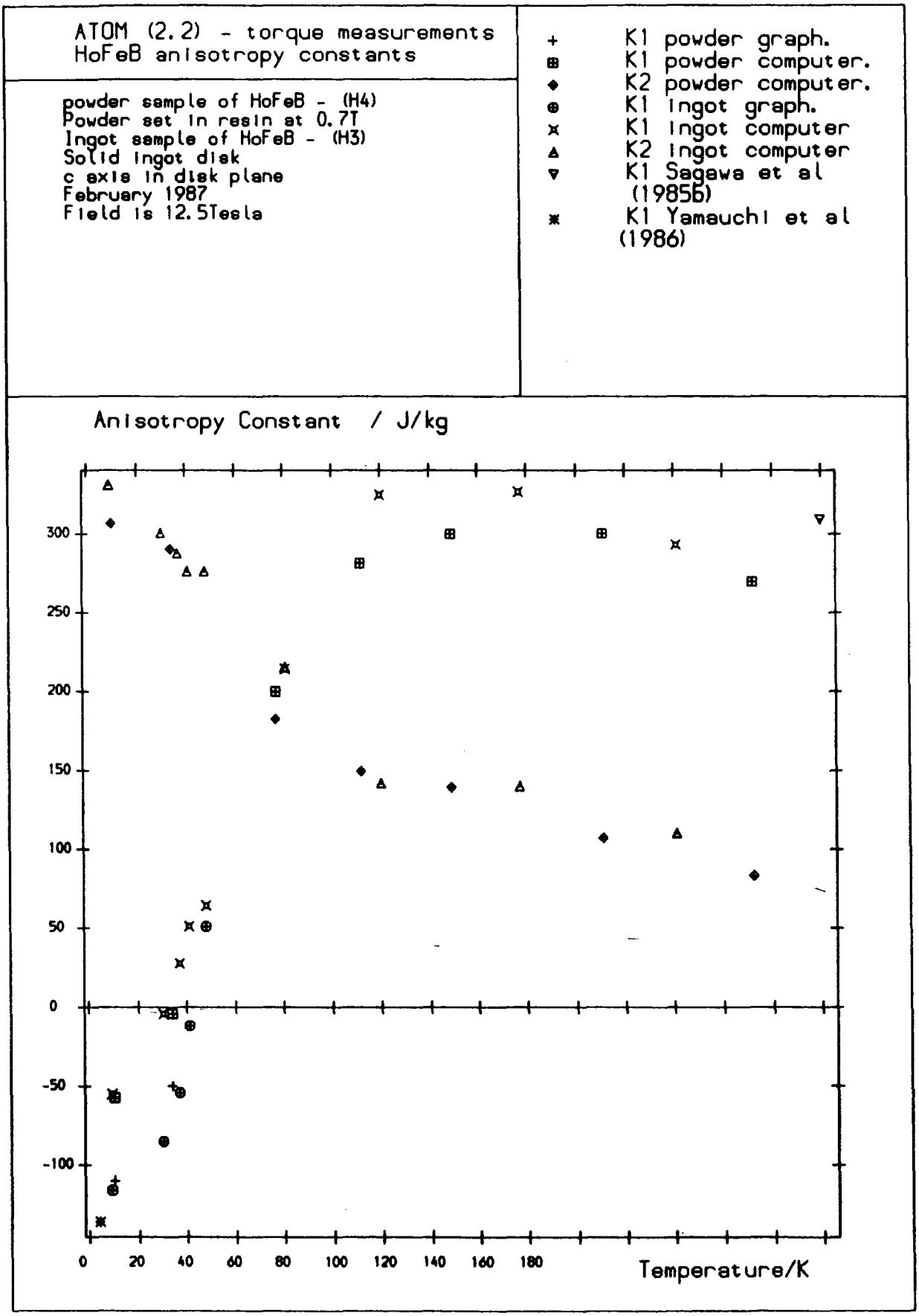


Fig 6.10 Anisotropy constants from the torque curves of $Ho_2Fe_{14}B$.

K_1 is almost constant from room temperature down to about 100K, below this it reduces rapidly and below 42K becomes negative, ending up with a negative value at 5K of about 40% of its room temperature value.

K_2 is about 25% of K_1 at room temperature but increases with decreasing temperature. This differs slightly from the behaviour in the case of NdFeB where the value of K_2 at room temperature is very much less than K_1

The good agreement between powder and ingot gives some confidence in the values. The same types of errors as for the NdFeB sample are possible but with the provision that the powder was single phase, no other phase was observed under the microscope, and as the ingot had much larger grains than the NdFeB sample most particles may be expected to be single grains.

The room temperature results obtained by Sagawa(1985b) is some 15% larger than those measured here and the value of Yamauchi et al(1986) at 4.2K is also about 15% larger in magnitude than the graphically calculated values(see Fig 6.10). These results were obtained from magnetisation curves on single crystals.

6.2.1.3 $Gd_2Fe_{14}B$ single crystal.

6.2.1.3.1. Uniaxial Anisotropy.

Measurements on sample (G3) were made with the field at 5 Tesla and at 12 Tesla. The torque curves were all smooth with no sign of higher order terms. when plotted against the moment angle they appear sinusoidal (e.g. Fig 6.12) with measurements being taken over about 180°, even near the hard direction. These were fitted with one anisotropy constant and with two, the fit appeared good in both cases and the two constants are plotted in Fig 6.11.

K_1 increases with temperature, up to 300K the highest measured. At higher field there is an upturn in K_1 at temperatures below 80K. This may be due

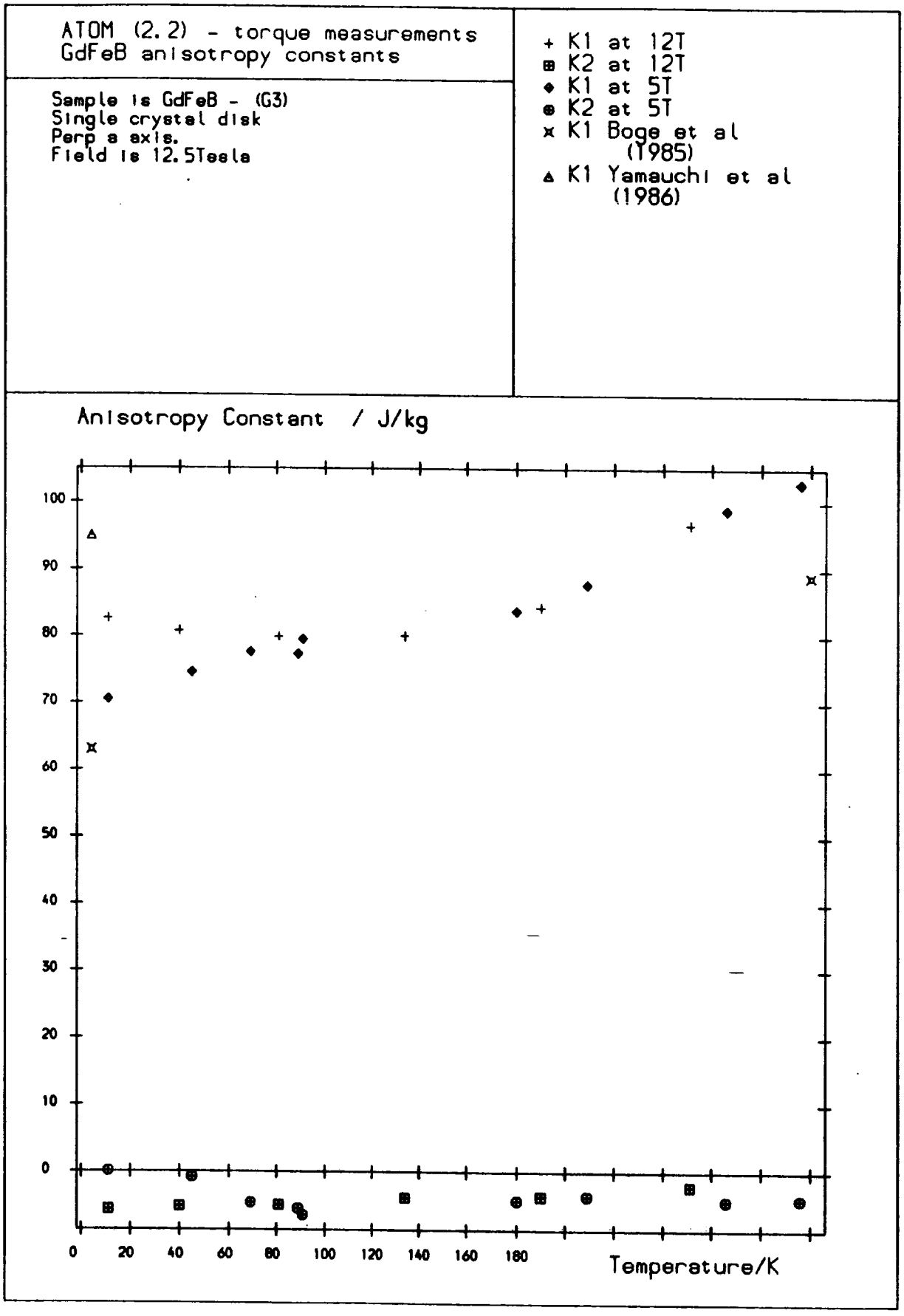


Fig 6.11 Uniaxial anisotropy constants from torque curve of $Gd_2Fe_{14}B$

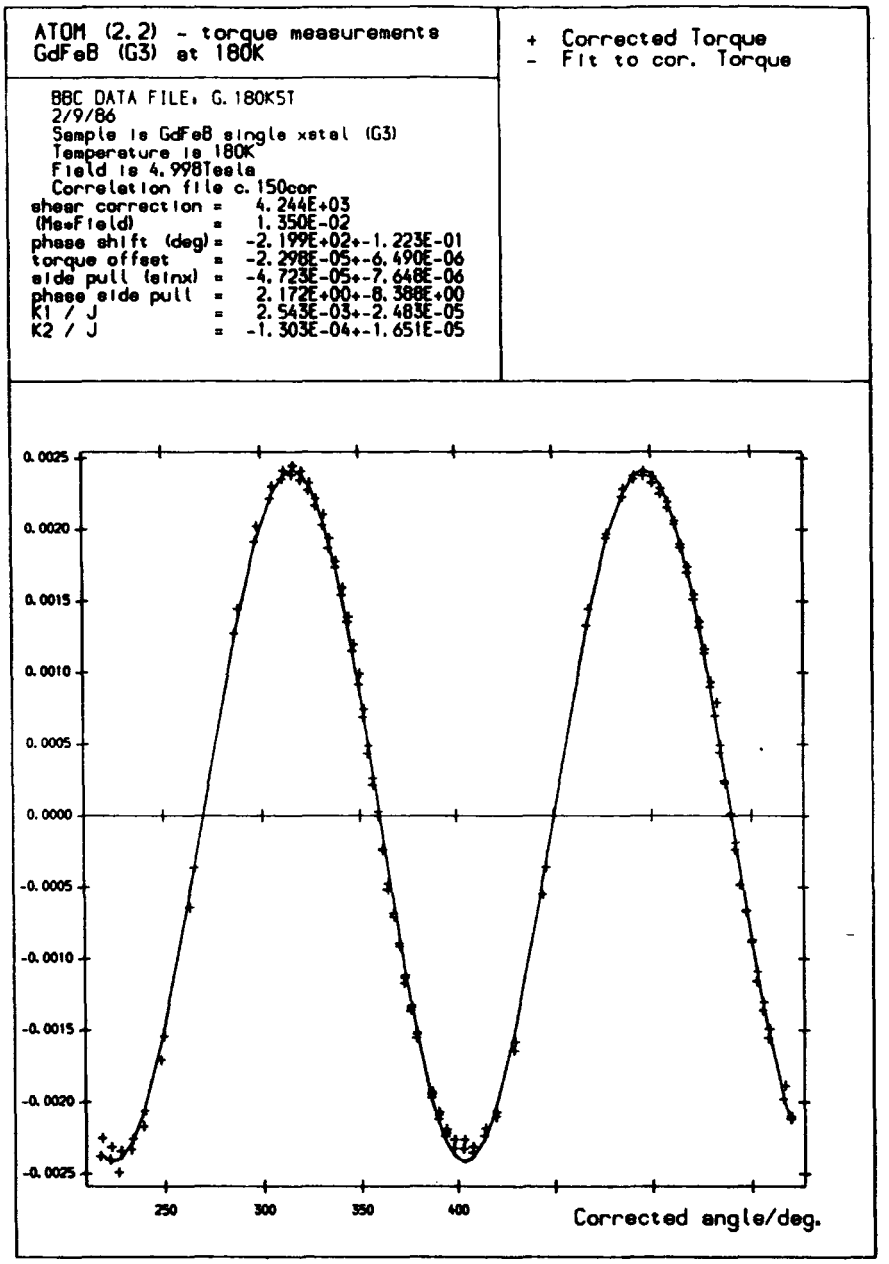


Fig 6.12 Torque curve of $Gd_2Fe_{14}B$ perp a axis

to the separation of the coupling as observed by Franse et al(1987) when studying ferrimagnetically aligning compounds of the same structure in fields up to 40 Tesla. With this in mind the low field values (5 Tesla) have been taken as the more accurate.

K2 is relatively small, around -5 J/kg, and appears to be constant over the range studied.

Bogé et al (1985) noted a similar rise in K_1 between 4.2K and room temperature, although they measured a value some 10% lower than measured by the torque curves. Yamauchi et al(1986) measured a value some 20% larger than ours at low temperature.

6.2.1.3.2. Basal plane Anisotropy of $Gd_2Fe_{14}B$

From a single crystal sample a disk was cut at right angles to the c axis to allow measurement of the basal plane anisotropy. No other crystals were obtained from other materials to allow comparison. As the basal plane anisotropy is much smaller than the uniaxial anisotropy the torque curves measured had more serious correction neccessary before the basal plane anisotropy could be extracted.
The instrument was also run at a much lower current to measure the much smaller torques involved.

The correction terms allowed for were:
a) The side pull term in the form of $A\sin(\theta+B)$

b) The effect of uniaxial anisotropy due to the not quite perfect cutting of the crystal disk (the value measured suggested an error of about 0.4° from the c-axis). This was just in the form of $A_2\sin^2(\theta+B_2)$. The low value of K_2 measured and reported in sect. 6.2.1.3.1. justifies this simple correction for the off perfect cutting.

c) The hysteresis of the instrument, probably due to the springiness in the wires. In Fig 6.13a this is clear as the two curves lie one above the other. Comparison with measurements made without a sample, but rotating through the same angle showed that this 'springiness' was a constant once the instrument had rotated more than a certain amount. In practice this was allowed for by discarding the data from about 100° to about 200° and splitting the measurements into two sets of data, one for the forward direction and one for the reverse direction. These were then fitted independently and the results plotted together.

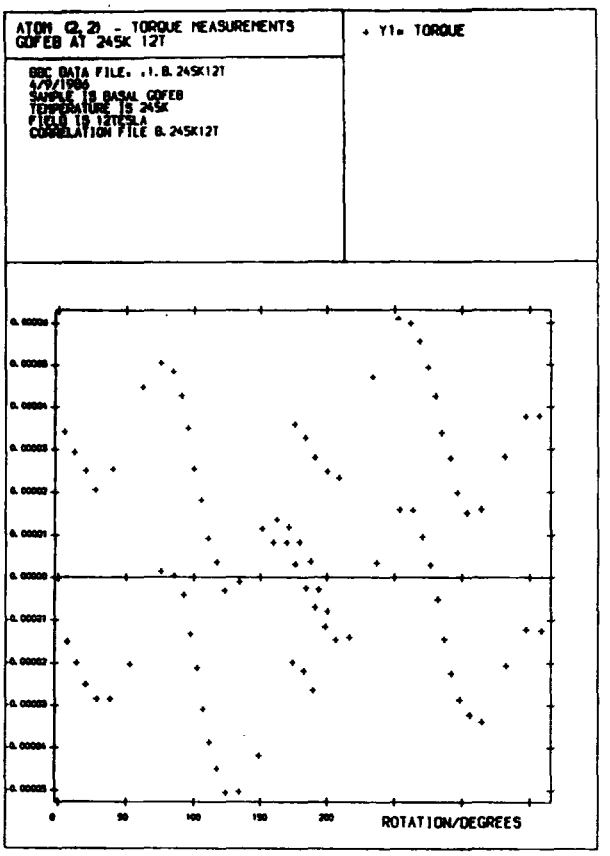


Fig 6.13a Torque curve of $Gd_2Fe_{14}B$ perp c axis

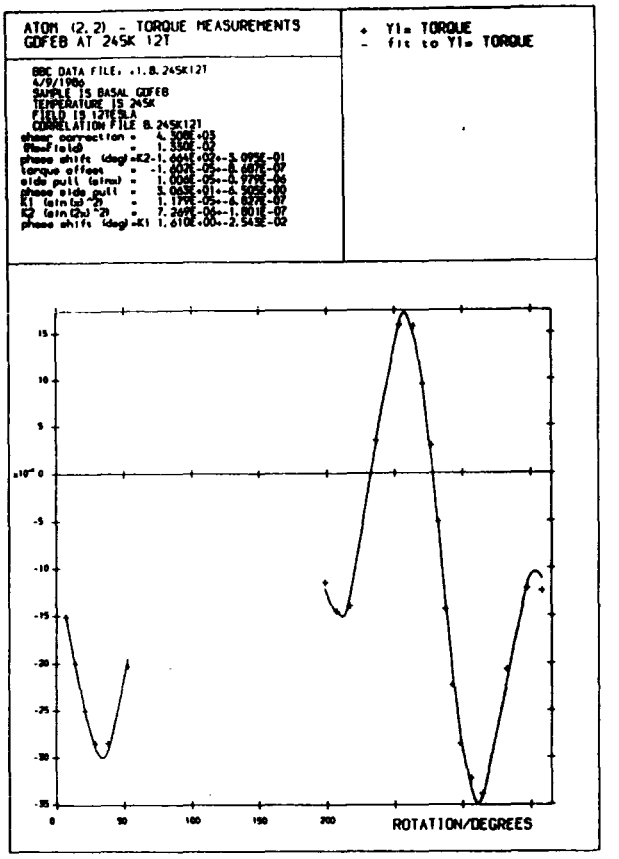


Fig 6.13b Torque curve of $Gd_2Fe_{14}B$ with fitted curve

d) The usual shear correction (Appendix 8) to the angle of the moment inside the sample, although in this case the correction was small. It should be noted that as the basal plane is hard and the measurements were made at 5 Tesla, the moment was canted at about 10 degrees to the basal plane while the measurements were made. This sounds quite a long way but as K_3 is of the form:

$$E= \quad .... + \quad K_3 \cos(4\phi) \sin^2(\theta) \quad + \quad ... \qquad (6.1)$$

this gives an error of less than 4% in the measured anisotropy.

The values for K_3 are plotted in figure 6.14 below.

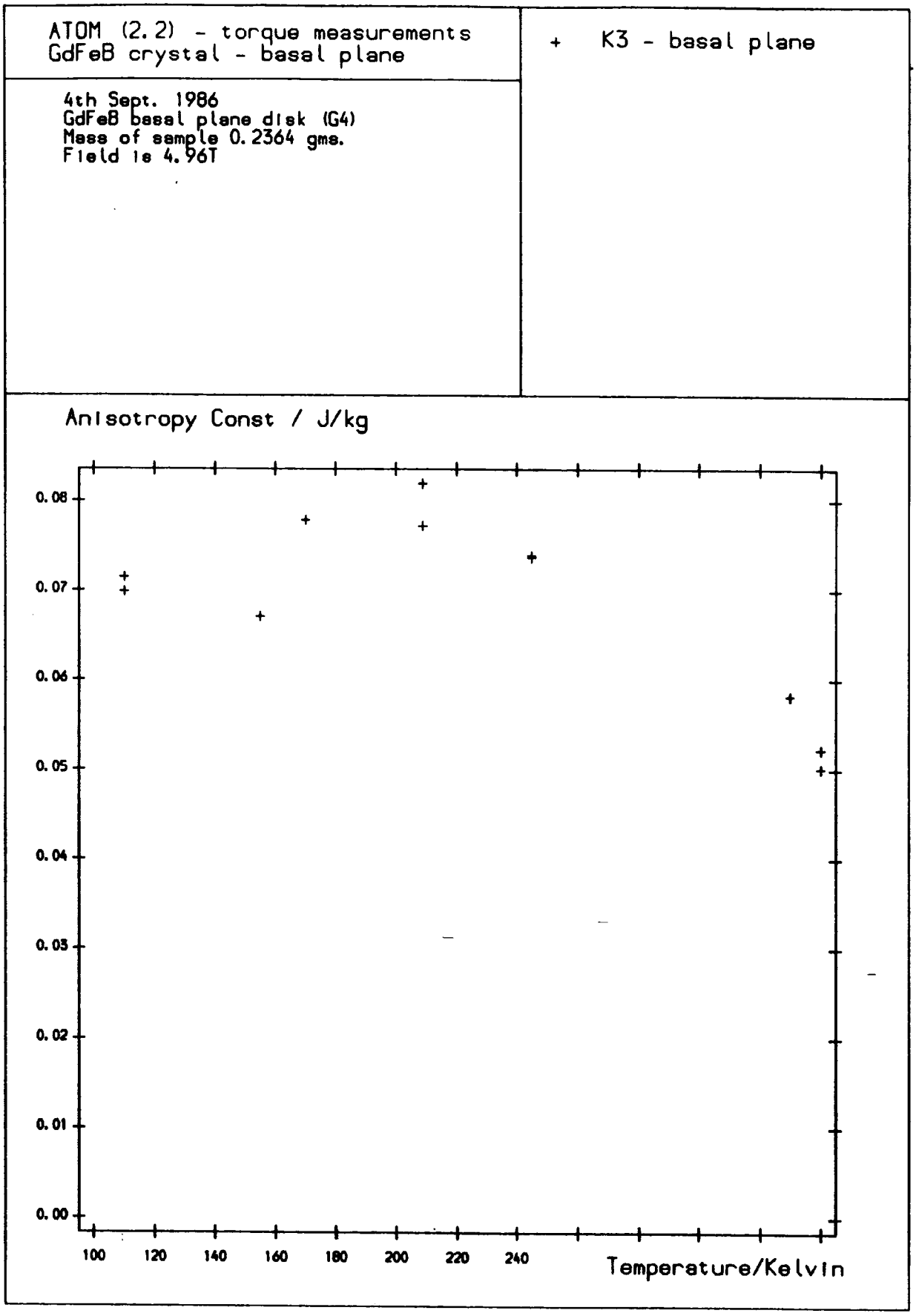


Fig 6.14 Basal plane anisotropy constant from torque curve of $Gd_2Fe_{14}B$

Measurements could only be made down to about 100K, below this the noise on the data became unacceptable, it is not clear whether this was due to an increase in the stiffness of the wiring or due to freezing of some oil or grease on the bearings. Whatever the reason the results below this temperature were not of a good enough quality to be confidently fitted, especially in view of the large number of corrections neccessary. There appears to be a reduction in K_3 above 200K, It is not clear whether there is a peak at this temperature as the data is obviously subject to scatter of the order of 20%.

6.2.1.4 $Dy_2Fe_{14}B$ single crystal.

The torque curves measured on sample (D3) were very different from those measured on the $Gd_2Fe_{14}B$ sample due to a much larger anisotropy and a smaller moment. These combined to make it difficult to pull the moment far out of the easy direction, even at 12 Tesla. The moment was pulled about 30 to 35 degrees from the easy direction. The curves were analysed on the assumption that the total moment of the sample, measured with the VSM, behaved as a single moment and only one anisotropy constant was fitted. Although a second anisotropy constant improved the fit (see figure 6.15) the value of the second constant cannot be relied on, due to the high anisotropy, meaning that the sample is unlikely to remain single domain a long way from the easy direction, neither are the two sublattices likely to remain antiparallel.

The results of the fits are given in figure 6.16, these are also compared with values from Sagawa(1985b) and Yamauchi et al(1986). These values are slightly lower than the values measured here.

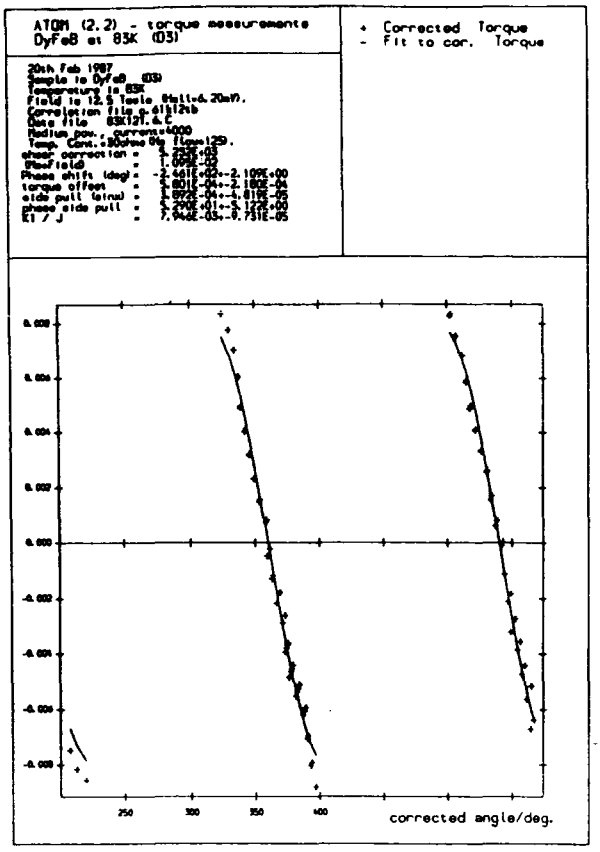


Fig 6.15a Torque curve of $Dy_2Fe_{14}B$. fitted with one const.

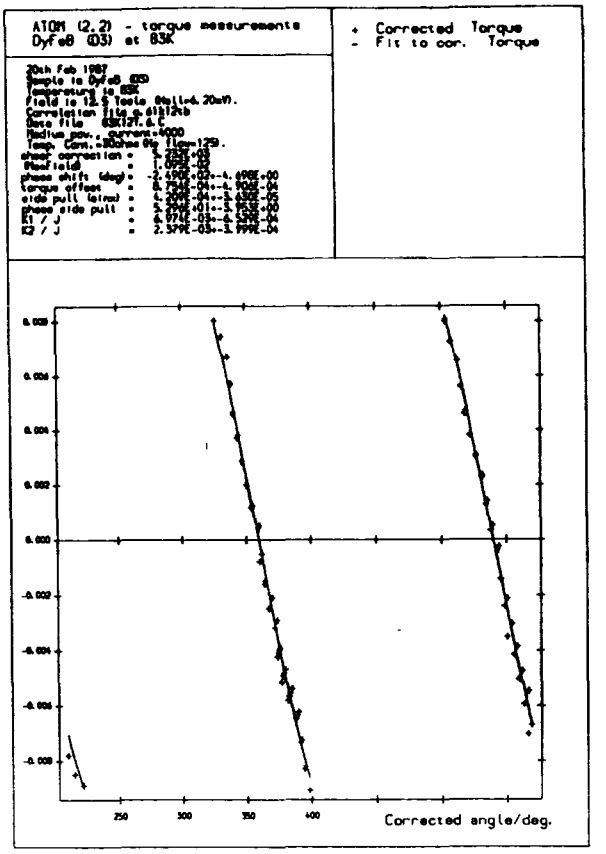
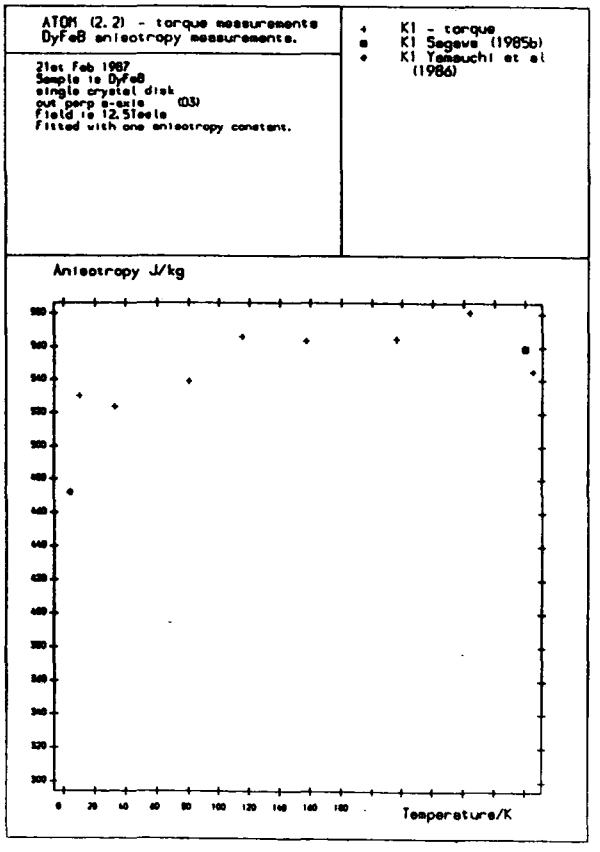


Fig 6.15b Torque curve of $Dy_2Fe_{14}B$. fitted with two const.



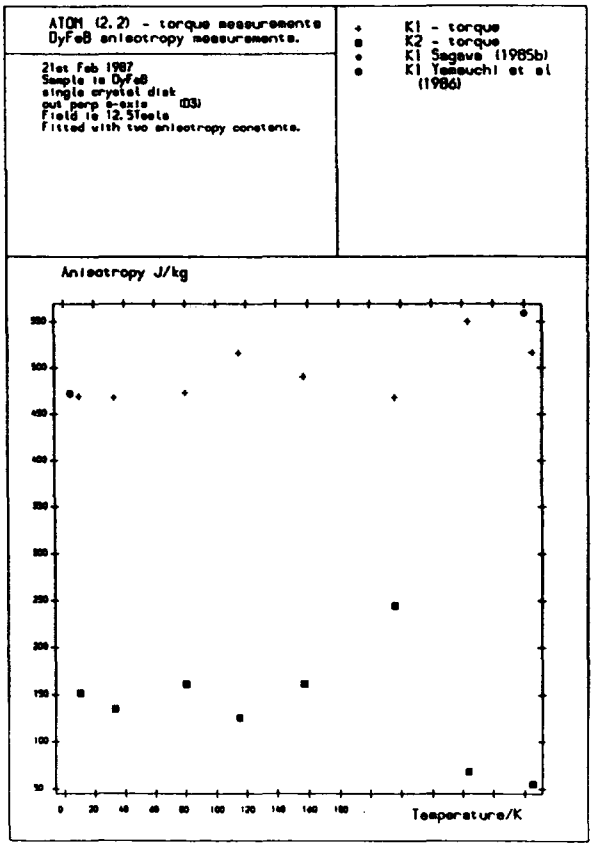


Fig 6.16 Anisotropy constants from torque curve of $Dy_2Fe_{14}B$

6.2.2. Rare earth iron magnets.

Torque curves were measured on a series of magnets with varying cobalt content. The samples were in the form of disks cut from sintered magnets with the aligning direction within the plane of the disk. These were measured from 10K to 370K. For analysis and more details see the PhD Thesis of I.Coulson, Sunderland Polytechnic (1987/88).

An example of the torque curves measured is given in Figure 6.17 and the dependence of anisotropy on cobalt concentration in figure 6.18.

There is an initial increase in the anisotropy with cobalt addition, possibly due to the increased Curie temperature then a decrease with increasing cobalt concentration.

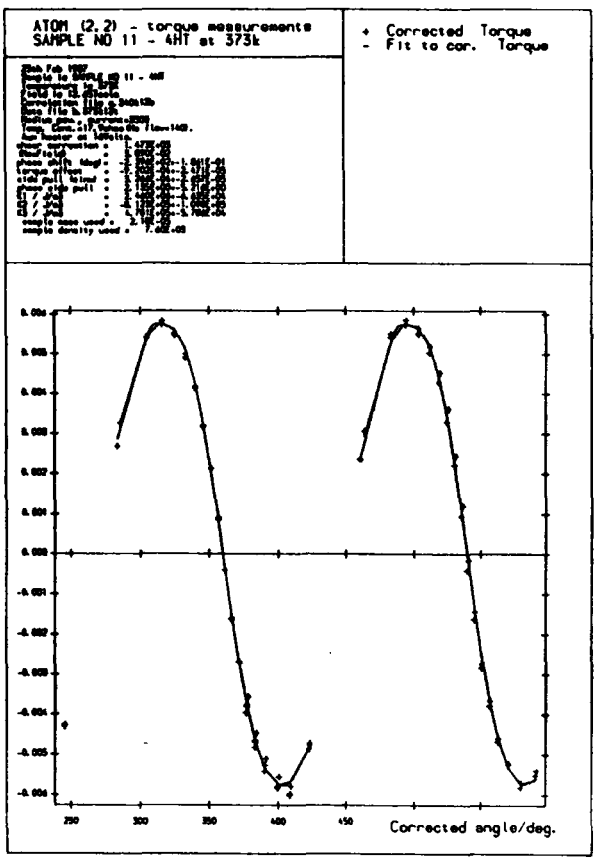


Fig 6.17 Torque of a sintered magnet.

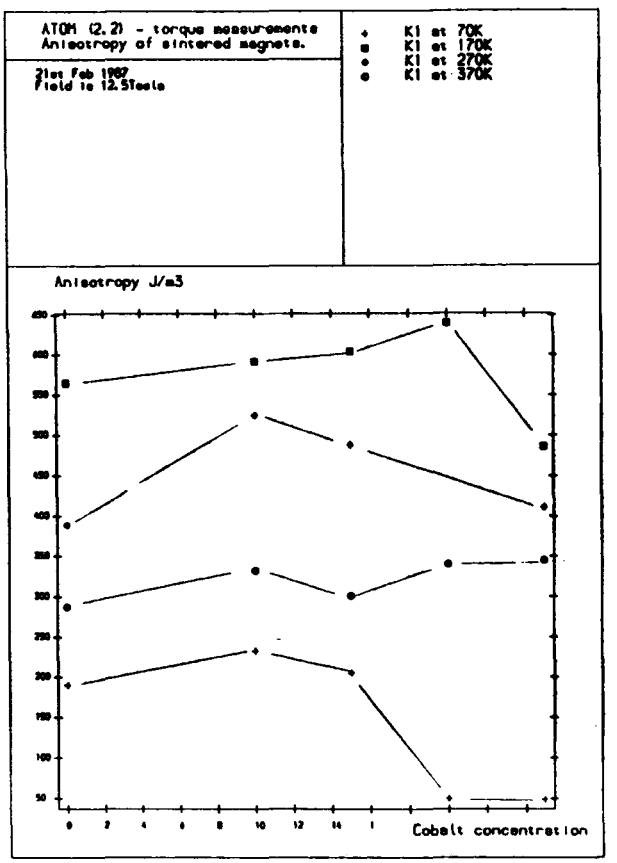


Fig 6.18 Dependance of anisotropy on Cobalt conc.

6.2.3. Terbium.

An example of the torque near the easy direction on a pure terbium sphere is given in figure 5.12, the temperature dependence of the anisotropy is given in figure 5.13, compared with values obtained by Corner(1985) from magnetisation measurements.

The values agree well between 50K and 150K.

Below 50K there is a difference in the way the curves approach the T=0K axis. The difference may be due to error in the values. Those of Corner at low temperature are obtained by extrapolating magnetisation measurements to higher field. Those from the torque measurements also have an angular correction amounting to about 30% of the signal, so any error in the mass will be increased to 1.5X the error and any error in the moment by X0.5. The slope was measued to better than 4% so the total error in the anisotropy should be about 6%.

Above 150K the anisotropy measured by Corner rapidly drops off while there is a larger 'tail' on the curve measured by the torque magnetometer.This is due to the increased moment at high field(see section 6.1.3) which causes an increase in the anisotropy. This results in the zero field anisotropy measured by Corner dropping to zero at 240K while at 12.5 Tesla the anisotropy does not fall to zero until about 350K.

Using the anisotropy measured in sect 6.2. values for δ are calculated and included in Table 6.2.

Rare Earth	Domain Spacing W_1 / μm	Saturat. Magn. $\sigma_s/JT^{-1}kg^{-1}$	Density $\rho_{m/kgm^{-3}}$	Wall Energy. γ / Jm^{-2}	Anisot. Const. K_1/Jkg^{-1}	Wall Thick. δ / nm.
Nd	0.78	169	7590	0.033	630*	5.4
Dy	4.35	70	8050	0.035	570	6.1
Gd	2.64	87	7880	0.032	102	31
Ho	1.50	86	8090	0.019	280	6.5

Table 6.2 Wall energy and wall thickness calculated from domain spacings and magnetisation and anisotropy measurements. * Taken from Yamada et al(1986).

Except for the gadolinium compound where a small anisotropy gives a larger wall thickness, the domain wall thicknessess calculated are small, of the order of 10 cell widths, so that a simple model of domain wall with smoothly rotating moments is not very good.

6.3.2. Domain patterns under applied field.

Using commercial ferrofluids, e.g. Ferrofluidics 809, it was possible to observe static domain walls. There was a tendency however for the particles to settle at the domain wall position and not move with the domain wall, leaving 'ghosts'. Using a more stable fine grain ferrofluid it was possible to watch domains moving under applied fields. This fluid was also made using a high boiling petroleum ether as solvent and this meant that the fluid evaporated very slowly and observations could be made over a long time period.

Within grains the domains moved smoothly until pinned by inclusions or grain boundary imperfections. As the field was increased main domain walls disappeared but spike domains remained which shrunk with increasing field

finally dissappearing into the inclusion or grain boundary. On decreasing the field domains reappeared spontaneously, the position not necessarily being the same as that at which the last domain disappeared. The behaviour for all the samples was similar, with only variations in the magnitude of the field necessary to remove the domain pattern.

An example is given in figure 6.25, on a sample of commercially cast ingot. In the field of view are two regions, one with the easy direction nearly in the plane of the surface on the right and one with the easy direction more nearly perpendicular to the surface on the left.

In figure 6.25a the applied field is low, the domains are nearly evenly spaced and the ferrofluid settles on the domain walls giving domain wall contrast. Between 6.25a and 6.25c the increasing field causes the domain walls to move within the grains, but the intersection with the grain boundary does not move. As the field increases the non uniformity of the field results in a vertical component to the field which gives a domain contrast, both types of contrast are visible in fig 6.25c. On further increasing the field (6.25d) domains of reversed magnetisation from the bulk begin to disappear, at this point it is also clear that there is some coupling between the grains, even through a large grain boundary layer, as domains tend to remain at the same point on both sides of a such a boundary layer. The domains in the left hand half of the sample have not moved up to this field. On increasing the field these domains begin to move(6.25e to 6.25h), but even with the maximum field which can be applied by the electromagnet there still remains a large number of domain walls observable within this region.

On decreasing the field the domains in the left hand region move smoothly back(fig 6.25i, 6.25j) and at a lower field than that at which they disappeared the domains in the right hand half begin to reappear (Fig 6.26k, 6.25l). It is not possible to suggest any preferable sites for reappearing domains from our observations, the only point to note is the difference in the shape from the last domains to disappear. The reappearing domains are rounder in shape, showing that they move rapidly from their initial appearance to occupy a large volume. The disappearing domains are long and thin, occupying as little volume as possible while maintaining their 'pinning' sites at the grain boundaries.

Reversing the field (Fig 6.25m - 6.25p) gives very similar domain patterns (c.f. fig6.25a - 6.25d) and suggests again that there is some influence of the grain boundary on the domain pattern.

Chapter 7: Conclusions and Suggestions for Further Work.

7.1 Instrumentation

A novel torque magnetometer designed to operate within the confines of a solenoid has been described in chapter 5. This instrument has been built and it has been shown that the problems inherent in this type of instrument can be overcome. The viability of such an instrument for measuring torques within a solenoid has been shown by performing measurements on terbium where the anisotropy had been previously measured. The instrument is a direct reading instrument and doesn't rely on comparison with standards. The standardisation is on voltmeters and ammeters which can be easily calibrated.

In constructing such an instrument care has to be taken to ensure independance of the alternating e.m.f picked up and used for the determination of angle from the current providing the counter torque. The correlation between the e.m.f. picked up and the angle has to be determined with the instrument in position in the solenoid and at the measuring temperature to prevent unacceptable errors from image effects. Care is also required in the measurement of the temperature of the sample at high torques as heat dissipation in the counter-torque producing coils cannot be ignored.

The actual instrument described was tailored to make measurements on the anisotropy of rare earth based materials with correspondingly large torques. Being designed to operate in the Durham University Superconducting Solenoid it is able to make measurements in fields of up to 13Tesla and at temperatures of 4.2K to 400K. The instrument is able to measure torques of

up to 1Nm at full field and also down to a resolution of 10^{-6} Nm. The maximum dimension of the sample is 7mm in any direction. By variations in the detailed design instruments could be made on the same principle to cover a larger range of torques. The lower limit of sensitivity is determined by the type of bearings used and the maximum torque measurable by the heat dissipated in the counter-torque producing coils. The present instrument approaches the maximum torque measurable, but at a price of resolution at high sensitivities. An instrument with a number of ranges could be built by constructing interchangable heads, the basic electronics remaining the same. As calibration is only on the current measurement there would be no problem of recalibration on changing heads.

7.2 Rare Earth Iron Boron Anisotropy

Some measurements have been made of the anisotropy of $R_2Fe_{14}B$,R=Nd,Ho,Gd,Dy alloys using high field torque magnetometry. The results confirm the anisotropy measured using magnetisation curve analysis by other workers
At room temperature neodymium, the only light rare earth studied, showed ferromagnetic ordering while the heavy rare earths, gadolinium, dysprosium and holmium showed ferrimagnetic ordering. This fits into the pattern found with all rare-earth iron intermetallic compounds.

The gadolinium compound showed the weakest anisotropy, about 20% of that shown by the other compounds. This therefore sets a maximum on the contribution from the iron anisotropy. The anisotropy of the gadolinium ion must be a lot weaker than that of the other ions. As the gadolinium ion has a spherically symmetrical unperturbed ground state and therefore has no possible contribution from single ion exchange terms as the ion is

isotropic if not perturbed, it also suggests that the majority of the anisotropy in the other compounds originates from crystal field effects on the rare earth ions.

As the two ion anistropy will be present in all the compounds it is difficult to distinguish this from iron lattice anisotropy. The increasing anisotropy between 4.2K and room temperature suggests that there is some two-ion contribution from the gadolinium at 4.2K which reduces the total anisotropy, and that this falls off more rapidly than the anisotropy due to the 3d magnetism and therefore causes an increase in the overall anisotropy The easy axis of magnetic anisotropy at room temperature was the c axis in all four cases. As the other three rare earths, excluding gadolinium, have negative Stevens' α_J factors (Table 2.2) it is not surprising that they should have the same easy axis.

The neodymium and holmium compounds both show a spin reorientation from an easy c-axis ferromagnetic state at high temperature to an easy cone ferromagnetic state at low temperature. The study of the anisotropy shows that this reorientation is a second order effect due to the weakening of the first anisotropy constant, K_1, as the temperature is decreased. This weakening begins to be observable some 50K above the spin reorientation temperature.

The origin of this reorientation, whether due to the dominance of higher order crystal fields at lower temperature or due to the difference between the temperature dependence of the crystal field anisotropy and the exchange anisotropy is not clear. The local situation of the rare earth ion in the structure(see Fig3.4a and b) is not simple, with two different sites, both with 20 nearest neighbours, consisting of six neighbours in the same plane,

a tilted hexagon of iron ions above and below and one interplanar iron ion (Fe(4)) in the centre of each hexagon. At low temperature the higher order crystal field terms will play a more important role than at higher temperatures as they fall off more rapidly with temperature than the first order term. These higher order terms are also short range and therefore determined more by the local environment than the extended crystal field. This means that these terms are therefore the most affected by the complexity of the local rare earth environments and most difficult to estimate by point charge models.

It would require more detailed anisotropy measurements, particularly of K_2 and higher anisotropy constants as well as a theoretical study of the expected behaviour of this structure to determine the real cause of this spin reorientation. The samples used in this study were not of a good enough quality to enable these measurements to be made, and highly perfect single crystal samples would be required of all the materials. In particular it would be interesting to look at the basal plane anisotropy which it has only been possible to measure for the gadolinium compound so far as no other suitable single crystals were available.

It would also be interesting to look at materials with a positive α_J Steven's Factor. It would then be possible to investigate the competition of the expected negative rare earth anisotropy with the positive iron anisotropy. The erbium compound is a particularly interesting possibility where, if the spin reorientation observed in neodymium and holmium compounds is due to higher order crystal terms, it may be possible to observe a tendency to weaken the negative anisotropy at low temperatures.

7.3 Magnetisation Processes in Rare Earth Iron Boron Magnets

The study of domains in large grained and single crystal samples has shown classical uniaxial domain patterns. Similar structures were observed in the four different intermetallics studied, although the scale was related to the anisotropy Gadolinium with the weakest anisotropy had domain structures on a larger scale than the other compounds.

The structures observed under zero field conditions were equilibrium structures. A fine grained ferrofluid in a non volatile solvent makes dynamic observations of domain walls possible. Under an applied field the domain walls moved freely within the bulk of the grains although some interaction with grain boundaries was observed.

On saturating the material, in the case of NdFeB ingot with a field of around 400kAm^{-1} quite considerably less than the coercivity of up to 1000kAm^{-1} obtained with magnets produced from this material, the reverse domains were squeezed to extinction. When the field was removed new domains appeared at places other than those from where the last domain disappeared.

The high coercivity possible with this class of material must therefore be due to the microstructure induced by powdering and sintering the material. The coercivity depends on the magnetic isolation of grains and the suppression of nucleation sites, or areas of grain boundary where reverse domains can escape the grain boundary walls, as any domain wall once free in the grain will rapidly traverse the whole grain and reverse its magnetisation.

Calculation of domain wall energies and widths showed narrow walls, of the order of 7 to 8 cell units. These walls might therefore be easily pinned at any small non magnetic precipitate which could be induced within the material, or at any inhomogeneities near the grain boundary layers. This fuels the debate as to whether the magnetisation is determined by true nucleation processes or by trappped domains in the grain boundary region. It might be of interest to see if any correlation is possible between the domain wall width and the coercivity to help understand this mechanism.

Further samples could be studied in single crystal form, although the processes observed in all those studied were so similar that no additional information should be expected for magnetisation processes. Energies and widths for domain walls could be obtained for the other compounds in the series.

Domain studies using other methods would be more appropriate for studying sintered magnets. The Bitter patterns are not very good on sintered material as the fluid particles tend to migrate to the pores or grain boundaries after a couple of sweeps of the field. The Kerr effect would be more appropriate. Transmission electron microscopy, and x-ray topography if good enough crystals can be grown, could be used to investigate the interaction of domains with imperfections and boundaries.

References

Abache C. and Oesterreicher H. *J.Appl. Phys.* 57 4112 (1985)

Barkhausen H. *Phys. Z.* **20** 401 (1919)

Beaudry B.J.and Gschneidner K.A., Handbook on the Physics and Chemistry of the Rare Earths, ed. Gschneidner K.A. and Eyring L. (North Holland 1979)

Birss R.R. and Keeler G.J. *Phys. Stat. Sol. (B)* **64**(1) 357-366 (1974)

Bitter F. *Phys. Rev.* **38** 1903 (1931)

Bodenberger R. and Hubert A. *Phys. Stat. Sol. (A)* **44** K7-11 (1977)

Bogé M., Coey J.M.D., Czjzek G., Givord D., Jeandry C., Li H.S., and Odden J.L. *Solid State Commun.* **55** 295 (1985)

Braun H.F., Pelizonne M., and Yvon K. *Proc. of the Int. Conf. on Transition Metal Compounds*, **IIB** 11 (Grenoble 1982).

Brillouin L. *"Die Quanten Statistik"* (Springer, Berlin 1931)

Brun T.O., Sinha S.K., Wakalayashi N., Lander G.H., Edwards L.R., Spedding F.H. *Phys. Rev.* B1 1251 (1970).

Buschow K.H.J. *Mat. Sci. Reports* **1** 1-64 (1986)

Cable J.W., Moon R.M., Koehler W.C. and Wollan E.O. *Phys. Rev. Letter* **12** 553 (1964).

Cable J.W. and Wollan E.O. *Phys. Rev.* **165** 733 (1968).

Cadogan J.M. and Coey J.M.D. *Phys. Rev.* **B30** 7326 (1984).

Chaudri M.A. *PhD Thesis* (Durham University 1983)

Cooper B.R. in *Magnetic Properties of Rare Earth Metal, ed R.J.Elliot* (Plenum, London, 1972)

Corner W.D., Paige D.M., Hawkins R.D., Fort D. and Jones D.W. *J. Magn. Magn. Mat.* **51** 89 (1985)

Coulson I. *private communication* (1987).

Craik D.J. and Tabble R.S. *"Ferromagnetism and Ferromagnetic Domains"*.(1965)

Crangle J. *"The Magnetic Properties of Solids"* (Edward Arnold, London, 1977)

Croat J.J., Herbst J.F., Lee R.W. and Pinkerton F.E.

J. Appl. Phys. **55** 2078 (1984)

Doring W. *Ann. Physik* **7** 102 (1958)

Durst. K.D. *presented at CEAM meeting, Dublin.* (1987).

Ewing J.A. *"Magnetic induction in iron and other metals"*

Franse J.J.M. *presented at CEAM meeting, Dublin.* (1987).

Givord D., Li H.S., and Moreau J.M. *Solid State Commun.* **50** 497 (1984)

Givord D., Li H.S. and Perrier de la Bâthie R. *Sol. State Comm.***51** 857 (1984b)

Givord D., Moreau J.M., and Tenaud P. *Solid State Commun.* **55** 303 (1985).

Grössinger R., Sun X.K., Eibler R., Buschow K.H.J. and Kirchmayer H.R.

J. Mag. Mag. Mat. **58** 55 (1986).

Habenschuss M., Stassis C., Sinha S.K., Spedding F.H.

Phys. Rev. **B10** 1020 (1974)

Harris I.R. Noble C. and Bailey T. *J. Less Common Metals* **106** L1 (1985)

Hawton M.J. and Corner W.D. *J.Phys E:Sci Inst.* **20** 406 (1987).

Heisenberg W. *Z. Phys.* **49** 619 (1928).

Henry *Phys. Rev.* **88** 559 (1952)

Herbst J.F., Croat J.J., Pinkerton F.E. and Yellon W.B.

Phys. Rev. **B29** 4176 (1984)

Hirosawa S. and Sagawa M. *Solid State Commun.* **54** 335 (1985).

Hirosawa S., Matsuura Y., Yamamoto H., Fujimura S., Sagawa H. and Yamauchi H.

J. Appl. Phys. **59** 873 (1986).

Hutchinson R.I. *J. Sci. Inst.* **42** 885 (1965)

Ibarra M.R., Algarabel P.A., Alberdi A., Bartolomé J. and del Moral A.

presented at MMM conference Baltimore (1986).

Johansson J., Lebech B., Nielsen. M., Bjerrum-Møller H., Mackintosh A.R.

Phys. Rev. Lett. **25** 524 (1970)

Kasuya T. *Prog. Theor. Phys.* **16** 45 (1956).

Kianvash A. and Harris I.R. *J. Mat. Sci.* **19** 353 (1984)

Kianvash A. and Harris I.R. *J. Mat. Sci.* **20** 682 (1985)

Kirchmayr H.R. and Poldy C.A. *in Handbook on the Physics and Chemistry of the Rare Earths, ed. Gschneidner K.A. and Eyring L.* (North Holland 1979)

Kittel C. 1949 *Rev. Mod. Phy.* **21** 541

Kittel C. *"Introduction to Solid State Physics" 5th Edition* (Wiley, New York 1976)

Koehler W.C. Cable J.W., Wilkinson M.W. and Wollan E.O. *Phys Rev* **151** 414 (1961)

Koehler W.C. in *Trans. Am. Cryst. Ass. ed. H.G.Smith* **3** 53 (Polycrystal Book Service, Pitsburgh, 1967)

Koehler W.C., Cable J.W., Child H.R., Wilkinson M.W., and Wollan E.O. *Phys. Rev.* **158** 450 (1967)

Koehler W.C., Moon R.M., Child H.R. *AIP Conf Proc* **10** 1319 (1972)

Kramer M. *Physica* **1**, 182-192 (1934)

Lambrick D.B. *PhD Thesis* (Durham University 1986).

Lambrick D.B. and Hoon S.R. *personal communication* (1987).

Langevin P. *Annls. Chim. Phys.* **5** 70 (1905)

Li. D. and Strnat K.J. *J.Appl. Phys.* **57** 4143 (1985).

Lifshitz E. and Landau L. *Phys. Zeits. d Sowjetunion* **82** 153 (1935)

Livingston J.D. *J.Appl. Phys.* **57** 4137 (1985).

McGuiness P.J., Harris I.R., Rozendaal E., Ormerod J. and Ward M. *J. Mat. Sci.* **21** 4107 (1986)

Millouse A.H. and Mc.Ewan K.A. *Sol. St. Comm.* **13** 339 (1973)

Moriya T. and Kawabata A. *J. Phys. Soc. Jap.* **84** 639 (1973)

Néel L. *Ann. Phys.* **5** 232 (1936)

Néel L. *J de Phys. et rad.* **5** 241-251, 265-276 (1944)

Otani Y., Miyajima H. and Chikazumi S. *Jap. J. Appl. Phys.* **26** 623 (1987).

Pearson R.F. in "*Experimental Magnetism*" *ed. Kalvius and Tebble*
(John Wiley and Sons, New York 1979)

Ruderman M.A. and Kittel C. *Phys. Rev.* **96** 99 (1954)

Sagawa M., Fujimura S., Togawa M. and Matsuura Y.
J. Appl. Phys. **55** 2083 (1984)

Sagawa M., Fujimura S., Yamamoto H., Matsuura Y. and Hiraga K.
IEEE Trans Magn. **MAG-20** 1584 (1984b).

Sagawa M., Fujimura S., Yamamoto H., Matsuura Y. and Hirosawa S.
J. Appl. Phys. **57** 4094 (1985)

Sagawa M. *J. Magn. Soc. Japan.* **9** 25 (1985b).

Sankar S.G. and Narasimh K.S. *J. Magn. Magn Mat.* **54-57** 530 (1986)

Shoemaker C.B., Shoemaker D.P. and Fruchart R. *Acta Cryst.* **C40** 1665 (1984)

Sinha S.K. in *Handbook on the physics and chemistry of Rare Earths,*
ed Gschneidner K.A. and Eyring L.R. (North Holland, 1978)

Sommerfeld *"Elektrodynamik"* (1948) - translated as
"Electrodynamics" (Academic Press, New York 1952)

Stevens K.W.H. *Proc. Phys. Soc.* **A65** 209 (1952)

Strnat K., Hoffer G., Olson J., Ostertag W. and Becker J.J.
J.Appl.Phys. **38** 1001(1967).

Suzuki T. and Hiraga K. *J. Mag. Mag. Mat.* **54-57** 527 (1986).

Szewczyk A., Piotrowski K. and Szymczak R. *J. Phys. D.* **16** 687 (1983)

Tanner B.K. *"X-ray Diffraction Topography"* (Pergammon Press 1976).

van Vleck J.H. *"The Theory of Electronic and Magnetic Susceptibilities"*
(Oxford University Press 1932)

van Vleck J.H. *Phy. Rev.* **52** 1178. (1937)

Vanderkooy J. *J.Phys. E:Sci Inst* **2** 718 (1969).

Verge C., Altounian Z. and Datars W.R. *J.Phys. E:Sci Inst* **10** 16 (1977).

Volkov V.M., Kustov E.F., Makietov T.K. and Steczko G.

Phys. Stat. Sol. (B) **104** 649 (1981)

Wallace W.E. *Rare Earth Intermetallics* (Academic Press, London, 1973).

Weiss P. *J. de Phys.* **6** 661-690 (1907)

Wilkinson M.K., Child H.R., McHargue C.J., Koehler W.C. and E.O. Woolan,

Phys. Rev. **122** 1409.(1961).

Willcock S.N.M. *PhD Thesis* (Durham University 1985)

Williams H.J., Bozorth B.M. and Shockley W. *Phys. Rev.* **75** 155 (1949)

Yamada O., Tokuhara H., Ono F., Sagawa M. and Matsuura Y.

J. Mag. Mag. Mat. **54-57** 585 (1986).

Yamauchi H., Yamada M., Yamaguchi Y. and Yamamoto H.

J. Magn. Magn. Mat. **54-57** 575 (1986).

Yoshida K. *Physics Review* **106** 893 (1957)

Zhao T, Jin H., Deng J., Wu F., Song J. and Shi Y.

J. Mag. Mag. Mat. **54-57** 573 (1986).

Appendix 1:Modifications to Duchess

A1.1 Alterations to the cryostat and temperature controller.

During the investigations a leak in the central sample tube of the Duchess meant that this had to be replaced. The opportunity was taken to replace the copper block on the bottom and improve the temperature control. The original design is shown roughly in Figure A1.1, and the present design is shown in figure A1.2. The original design suffered two drawbacks.

i) After a period of operation of the cryostat for about three days there was a tendancy for the flow of Helium to drop, and eventually for it to be impossible to get any flow.

ii) The heater coil burnt out on a number of occasions, and replacement required not only dismantling the croystatic vessels to get access to the sample tube, but also winding a new heater in a groove with the Helium inlet tube across it. With the old heater 300K was the maximum that could be achieved.

These were solved in the present design:

i) The removal of the loop below the bottom of the copper block removed a condensation trap. The flow of helium to this point meant that this was always very cold. Any air or other vapour which got into the system and condensed was trapped and frozen at this point plugging the pipe. It was then almost impossible to remove. Any appreciable helium flow induces in the opposite direction to try and warm this tube went was straight back towards the liquid helium reservoir and the offending plug would be frozen further up the tube.

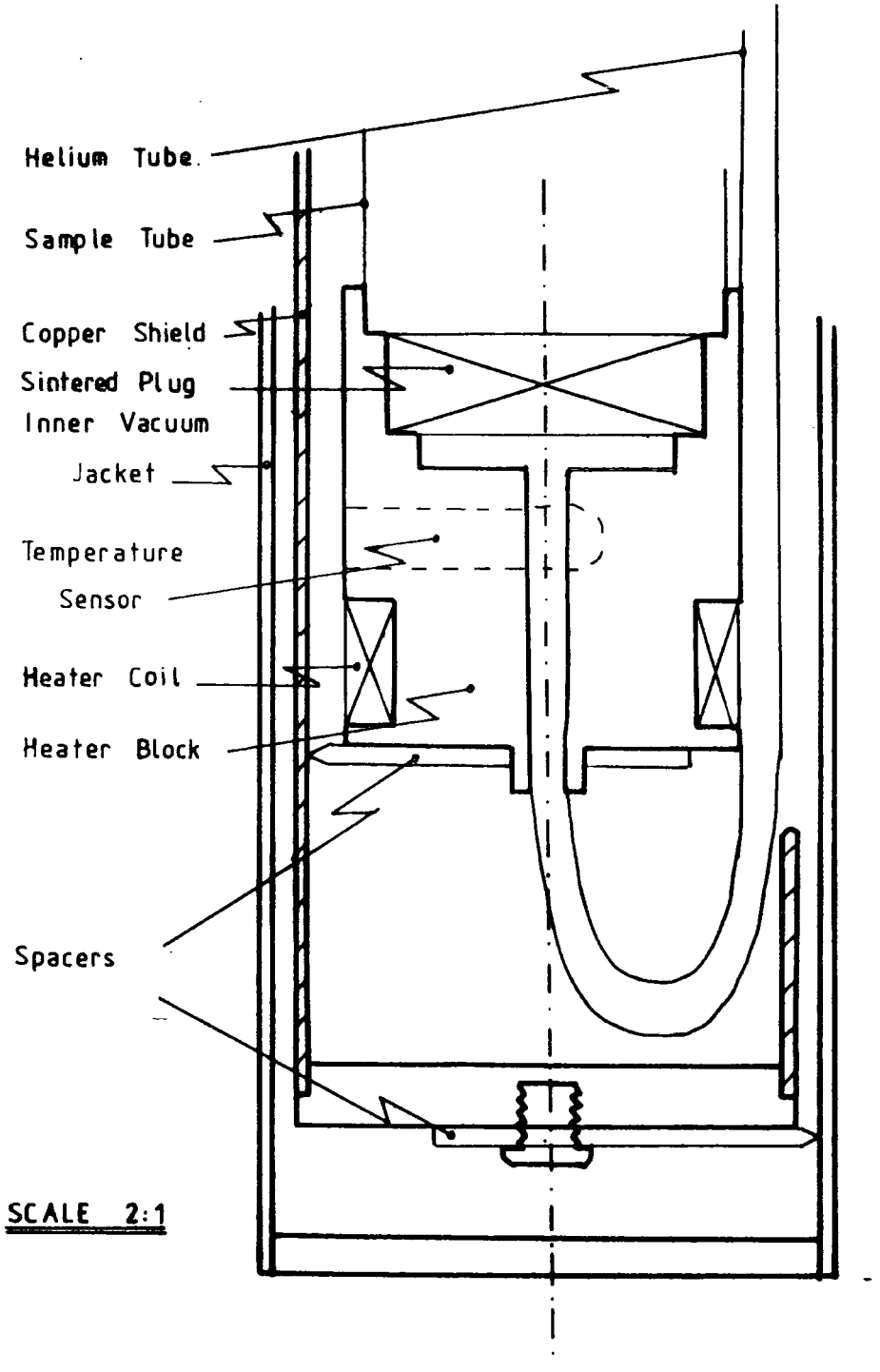
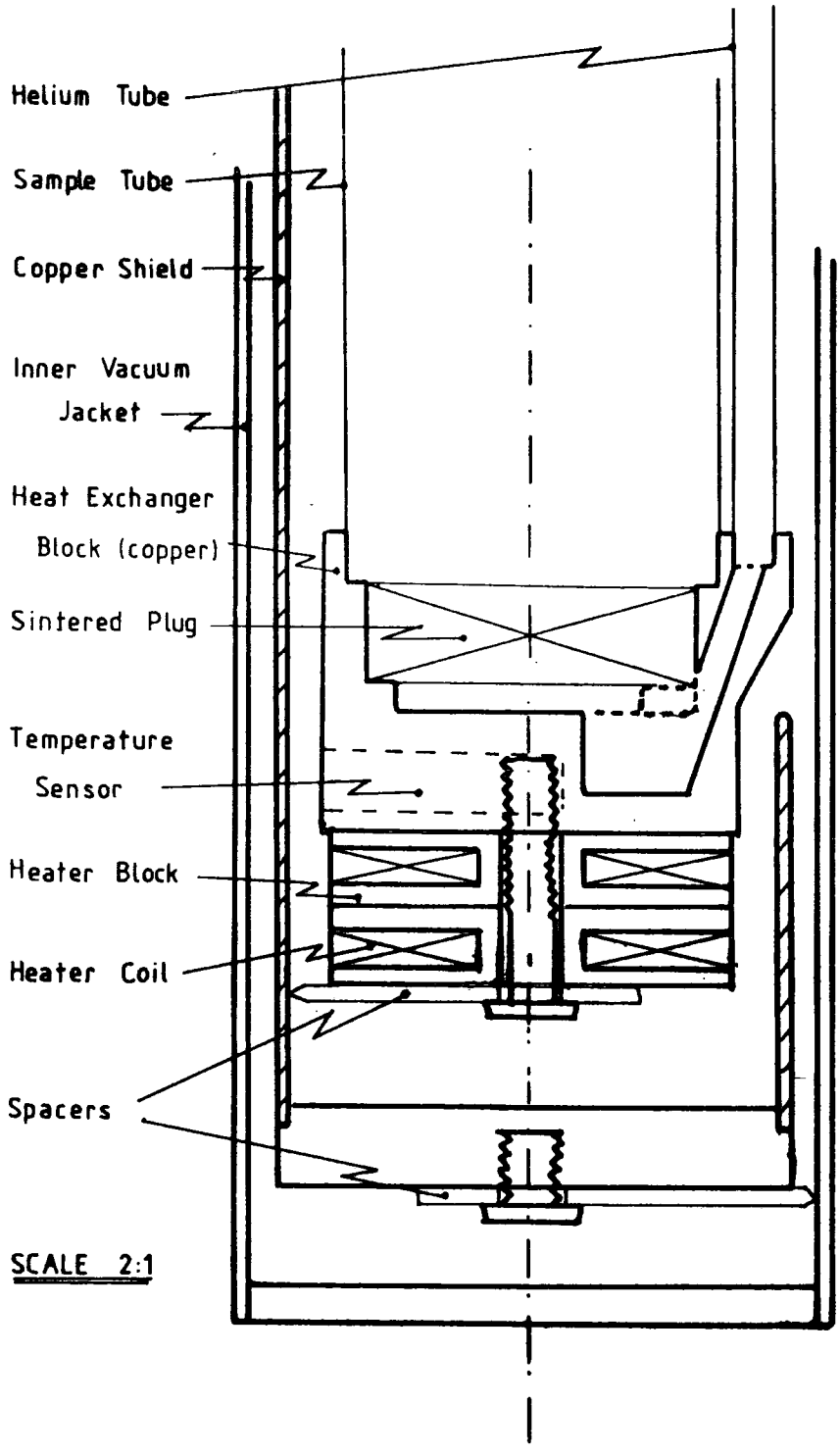


Fig A1.1 Original heater block design.



FigA1.2 Present Heater Block

The new design removes this trap and leaves the lowest point within the copper block. This can be warmed directly by the electric heater on the block and a small flow of helium from the reservoir will drive the gas off up the sample tube. There have not been any problems with plugging from this cause since the modification.

ii) The heater block is bolted onto the bottom of the copper block. There is also a spare heater in the event of one burning out during a run. This heater is more easily replaced (although the cryostatic containers still need to be dimantled!). It is constructed with motor vehicle exhaust putty as insulator for the non-inductively wound constantan coil. This convenient form of fire clay is easily painted on during winding and forms a good high temperature insulator to hold the windings. It is however dificult to remove and is best used in places where the whole heater can be replaced. With this temperatures around 400K can be obtained, limited by the increased boil off of helium due to heat leaks across the inner vacuum space and by the construction of the torque insert.

A trip was designed for the heater circuit which plugs into the temperature controller. The trip works by comparing the voltage and current across the heater coil and tripping when the resistance rises above a preset value. Unfortunately this trip does not work with the present constantan heater, as the resistivity of constantan does not alter sufficiently before it burns out. A second trip was therefore also installed, which works on a copper constantan thermocouple sandwiched between the heaters and trips a relay if the temperature rises above a preset value. This trip is a standard temperature controller available from RS.

A1.2 Other modifications.

1) Three carbon film resistors in parallel were added at the bottom and top of the magnet. These are just used to check on the temperature of the magnet during cool down or after a short period of idleness. Three resistors are used on a triple fail safe theory, as cycling between room temperature and 4.2K can cause resistors to fail.

2) Leads were connected up to the terminals of the magnet, through three large resistors in parallel. When the magnet current is altered (ramping) it takes a considerable period of time for the magnet to reach equilibrium due to its very large inductance.

This causes two problems. The current is used as a measure of the field and uncertainly in the actual current in the magnet is therefore an error in the field. Secondly the magnet can only be placed in persistant mode when the current has stabilized and a large error in current can cause a quench either going into persistant mode or coming out of it.

A meter in the power supply measures the voltage at the power supply terminals, but this also includes the voltage drop down the leads carrying the energising current to the magnet. The leads to the magnet terminals allow direct measurement of the voltage across the magnet, and when this is zero then the current in the magnet is the same as that in the power supply.

These are also useful to indicate whether the superconducting switch is normal or superconducting when the magnet is taken out of persistant mode as a slight voltage across the terminals is always present when the switch is normal.

3) A Hall Probe was installed in the space below the variable temperature insert. This allows for a measure of the field during ramping. This was installed for use in conjunstion with the VSM, to allow readings to be taken during ramping.

Appendix 2: Computer progams used

This appendix contains listing of the more important programs used during the work for this thesis. The programs for the micro are written in BBC Basic(*B*). Those for the mainframe, an AMDAHL 5020 with an MTS operating system are written in PASCAL(*P*)(for use with the PASCALJB compiler) or FORTRAN(*F*)(For use with the FORTRANVS compiler).

These programs are all stored on a tape registered as MJH1 at Durham University Computing Centre under the names given in the titles. *FS can be used to retrieve them.

General Programs

1) **LAUE.BBC**(*B*) - this is a program written to calculate the position of spots on a back scatter x-ray photograph from the crystal parameters and the alignment. Although it does no form factor and therefore intensity calculations it was found to be useful in aligning the samples.

2) **CRYSTAL.BBC**(*B*) - this is a program to draw on the screen the local enviroment of an atom within a tetragonal crystal system. It could be easily adapted for another system. The lattice constants are in line 350 and the number of atoms - 1 in line 390. The atom positions are given in cell coordinates from line 3230.

3) **MUTUAL.BBC**(*B*) - this calculates the mutual induction of the coil system in free space as a function of angle for a given coil size and separation. It is a simple finite element calculation and takes a long time for the micro to do. It was used during development of the instrument.

Torque magnetometer programs (BBC).

1) **TORQUE.BBC**(*B*) - this is the control program for the instrument, which controls the current in the coil, and calculates the torque as it goes. The data is stored on disk and the screen lists the data as it is collected.

2) **FIT.BBC**(*B*) - this calculates a fit to a reference run without a sample to give an angular correlation file used by TORQUE.BBC and TORCALC.BBC. It takes as input the data file produced by TORQUE.BBC.

3) **TORCALC.BBC**(*B*) - this recalculates the torque from the raw voltage and current data stored by TORQUE.BBC and a correlation from FIT.BBC. It is useful if a better reference is done after some data is collected.

4) **ROTHYS.BBC**(*B*) - this calculates the rotational hysteresis in a torque curve, either by straight line fits or by a sixth order polynomial fit to sets of points along the curve. In tests the results only differed by a few percent.

Data Utility Programs (BBC).

1) **SKETCH.BBC**(*B*) - this sketches the data on the screen.

2) **HP.BBC**(*B*) - this plots the data on a Hewlett Packard plotter.

KERMIT is used to transfer the data from the BBC to the University AMDAHL mainframe machine for curve fitting. The description of this program and copies are available from the Lancaster University Computing Centre.

Data processing programs (MTS).

1) **CONV.PAS**(*P*) is used to convert the transfered BBC data files to MTS text files that could be read easily on the MTS system. This means that a BBC data file can be directly transmitted to the mainframe.

2) **FIT.PAS**(*P*) is a program written to perform a least squares fit on the experimental data. The program includes the facility to test the routine with simulated experimental data with random Gausian noise. The program calls various subroutines from the NAG suite of subroutines. (main program is in file TFIT.PAS on the tape, other files %included are also required).

3) **FIT.FOR**(*F*) This is the interface for the NAG least squares routines. It sets up a datablock then calls the routines. There is also a subroutine (LSFUN1) which is required by the least squares subroutine and contains the function to be fitted.

4) **GRAPH.PLOT**(*P*) This is a flexible plotting routine to plot the data and or the fits. It uses the GHOST80 suite of plotting subroutines. These are written in FORTRAN77 and are all declared as external fortran routines at the beginning of the program, even though they are not all used by the program. A number of the procedures used are also used in FIT.PAS and so are not included in this listing. (main program is in GRAPH.PAS on the tape, other files %included are also required).

5) **GHOST.FOR**(*F*) This is just a series of fortran routines which were written to enable PASCAL procedures to call GHOST80 subroutines with strings as parameters. This means that all the GHOST80 routines can be accessed from PASCAL.

Listing of LAUE.BBC at 22:28:07 on MAY 12, 1987 for CCid=PHP7

```
 10 REM Laue  back  reflection  photograph  simulations.
 20 REM
 30 REM  For Cubic, Tetragonal and hexagonal systems.
 40 REM
 50 REM                    M.J.Hawton  Durham Univ. 1985/6.
 60 REM
 70 MODE 0
 80 Skip=FALSE
 90 REM
100 PROCAskinput
110 REPEAT
120   IF NOT Skip PROCAskplane
130   ON ERROR GOTO 260
140   IF NOT Skip PROCPlotscreen
150   ON ERROR GOTO 2300
160   Skip=FALSE
170   REPEAT
180     PRINT TAB(40,0)" press P:new plane, C:hard copy, Q:quit"
190     REPEAT:A$=GET$
200       UNTIL A$="q" OR A$="Q" OR A$="p" OR A$="P" OR A$="C" OR A$="c"
210     IF A$="c" OR A$="C" THEN PROCHardcopy
220     UNTIL A$<>"c" AND A$<>"C"
230   UNTIL A$="q" OR A$="Q"
240 END
250 :
260 Skip=TRUE:GOTO 110
270 :
280 DEFPROCPlotscreen
290 PROCPrintscreen
300 FOR Xd%=-Max% TO Max%
310   FOR Yd%=-Max% TO Max%
320     FOR Zd%=0TO Max%
330       BCC_cond= (Xd%+Yd%+Zd%)/2=(Xd%+Yd%+Zd%)DIV2
340       FCC_cond= (Xd%+Yd%)/2=(Xd%+Yd%)DIV2 AND (Xd%+Zd%)/2=(Xd%+Zd%)DIV2
350       FCC_cond=FCC_cond AND  (Yd%+Zd%)/2=(Yd%+Zd%)DIV2
360       Hex_cond= (-Xd%+Yd%+Zd%)/3=(-Xd%+Yd%+Zd%)DIV3
370       IF Group$="Simple cubic" THEN PROCPlot(Xd%,Yd%,Zd%)
380       IF Group$="BCC" AND BCC_cond THEN PROCPlot(Xd%,Yd%,Zd%)
390       IF Group$="FCC" AND FCC_cond THEN PROCPlot(Xd%,Yd%,Zd%)
400       IF Group$="Hexagonal" AND Hex_cond THEN PROCHexplot(Xd%,Yd%,Zd%)
410         IF Group$="Tetragonal" THEN PROCTetplot(Xd%,Yd%,Zd%)
420     NEXT:NEXT:NEXT
430 ENDPROC
440 :
450 :
460 DEFPROCPrint
470 *FX7,7
480 *FX5,2
490 *FX8,7
500 *FX6,0
510 CHAIN "P.NEWDUMP"
520 :
530 :
540 DEFPROCAskinput
550 CLS:PRINT"Laue Patterns":PRINT"-------------"''
560 INPUT'"Distance to screen is "'"(in cm.)";D
570 MaxPhi=ATN(10/D)
580 Xscale=0.0144:Yscale=0.0138
```

```
 590 Xcentre=610:Ycentre=505
 600 Xwidth=430:Ywidth=450
 610 PRINT''"Which crystal group"
 620 PRINT"(S=Simple cubic,F=FCC,B=BCC,H=Hexagonal,T=Tetragonal)"
 630 Endproc=FALSE:REPEAT:A$=GET$
 640   IF A$="F" OR A$="f" THEN Group$="FCC":Endproc=TRUE
 650   IF A$="B" OR A$="b" THEN Group$="BCC":Endproc=TRUE
 660   IF A$="S" OR A$="s" THEN Group$="Simple cubic":Endproc=TRUE
 670   IF A$="H" OR A$="h" THEN Group$="Hexagonal":Endproc=TRUE
 680   IF A$="T" OR A$="t" THEN Group$="Tetragonal":Endproc=TRUE
 690   IF NOT Endproc THEN VDU7
 700   UNTIL Endproc
 710 IF Group$="Hexagonal" OR Group$="Tetragonal":INPUT''"  C=";C;"  A=";A
 720 ENDPROC
 730 :
 740 :
 750 DEFPROCAskplane
 760 INPUT''"Indicies of normal plane are"''"H";H:INPUT"K";K
 770 IF Group$="Hexagonal" THEN I=-(H+K): PRINT "I=";I
 780 INPUT"L";L
 790 INPUT''"What is the maximum index?"'"(suggest not > 7)";Max%
 800 PRINT''"Do you want to see indicies for each point(y/n)":A$=GET$
 810 Prompt=FALSE:IF A$="Y" OR A$="y" THEN Prompt=TRUE
 820 :
 830 REM Calculate indicies and rotate if X=Y=0 as algorithm relies on the
 840 REM           component of the perp. vector not parrallel to 0,0,1
 850 REM All indicies are made to 'pseudo-cubic' indicies
 860 X%=H:Y%=K:Z%=L
 870 Rotate=FALSE
 880 OK=FALSE:IF Group$="Hexagonal" THEN OK=TRUE:X=H:Z=(K-I)/SQR(3):Y=L*A/C
 890 IF OK:IF X=0 AND Y=0 THEN Rotate=TRUE:Spare=Z:Z=Y:Y=X:X=Spare
 900 OK=FALSE:IF Group$="Tetragonal" THEN X=H:Y=K:Z=L*A/C:OK=TRUE
 910 IF OK:IF X=0 AND Y=0 THEN Rotate=TRUE:Spare=Z:Z=Y:Y=X:X=Spare
 920 OK=FALSE
 930 IF Group$<>"Hexagonal" ANDGroup$<>"Tetragonal"ANDX%=0 ANDY%=0:OK=TRUE
 940 IF OK :Rotate=TRUE:Spare=Z%:Z%=Y%:Y%=X%:X%=Spare
 950 ENDPROC
 960 :
 970 :
 980 DEFPROCPrintscreen
 990 CLS
1000 MOVE Xcentre+1,Ycentre+1
1010 FOR I%=-1 TO 1 STEP 2:FOR J%=-1 TO 1 STEP 2
1020      DRAW I%+Xcentre,I%*J%+Ycentre
1030      NEXT:NEXT
1040 MOVE Xcentre+Xwidth,Ycentre+Ywidth
1050 FOR I%=-1 TO 1 STEP 2
1060   FOR J%=-1 TO 1 STEP 2
1070      DRAW I%*Xwidth +Xcentre,I%*J%*Ywidth +Ycentre
1080      NEXT:NEXT
1090 PRINTTAB(0,0);"Laue back reflection from ";H;" ";K;" ";
1100 IF Group$="Hexagonal" THEN PRINT;I;" ";L ELSE  PRINT;L
1110 PRINTTAB(0,31);Group$;" max index:";Max%;", ";D;"cm to screen";
1120 IF Group$="Hexagonal" OR Group$="Tetragonal":PRINT;", C=";C;", A=";A;
1130 ENDPROC
1140 :
1150 :
1160 DEFPROCPlot(Xd%,Yd%,Zd%)
```

```
1170 IF Rotate THEN Spare%=Zd%:Zd%=Yd%:Yd%=Xd%:Xd%=Spare%
1180 ModXd=FNIMod(Xd%,Yd%,Zd%)
1190 IF ModXd=0 THEN ENDPROC
1200 XdotXd=FNIDot(X%,Y%,Z%,Xd%,Yd%,Zd%)
1210 ModX=FNIMod(X%,Y%,Z%)
1220 IF (ModX*ModXd)>1E-20 THEN CosPhi=(XdotXd/(ModX*ModXd))
1230 IF (ModX*ModXd)<=1E-20 THEN CosPhi=SGN(X%*Xd%)
1240 IF CosPhi>1 THEN CosPhi=1
1250 IF CosPhi<-1 THEN CosPhi=-1
1260 Phi=ACS(CosPhi)*2
1270 Deltheta=0
1280 IF Phi>PI/2:Phi=Phi-PI:Deltheta=PI:IF Phi>PI/2:Phi=Phi-PI:Deltheta=0
1290 IF ABS(Phi)>MaxPhi OR ABS(Phi)<1E-5 THEN ENDPROC
1300 Alpha=XdotXd/(ModX*ModX)
1310 Xb=Xd%-Alpha*X%
1320 Yb=Yd%-Alpha*Y%
1330 Zb=Zd%-Alpha*Z%
1340 NdotXb=FNRDot(-Y%,X%,0,Xb,Yb,Zb)
1350 ModXb=FNRMod(Xb,Yb,Zb)
1360 ModN=FNIMod(-Y%,X%,0)
1370 IF (ModN*ModXb)>1E-20 THEN CosTheta=NdotXb/(ModN*ModXb)
1380 IF (ModN*ModXb)<=1E-20 THEN CosTheta=SGN(Yb*X%)
1390 IF CosTheta>1 THEN CosTheta=1
1400 IF CosTheta<-1 THEN CosTheta=-1
1410 IF Zb<>0 THEN Theta=SGN(Zb)*ACS(CosTheta)+Deltheta
1420 IF Zb=0 THEN Theta=ACS(CosTheta)+Deltheta
1430 PROCPlotpoint
1440 ENDPROC
1450 :
1460 :
1470 DEFPROCPlotpoint
1480 R=D*TAN(Phi)
1490 Xs=R*SIN(Theta)/Xscale
1500 Ys=R*COS(Theta)/Yscale
1510 IF  ABS(Xs)>Xwidth OR ABS (Ys)>Ywidth   THEN ENDPROC
1520 Xs=Xs+Xcentre
1530 Ys=Ys+Ycentre
1540 PLOT69,Xs,Ys
1550 PRINT TAB(40,0)"                                          ";
1560 PRINT TAB(40,0);Xd%;" ";Yd%;" ";
1570 IF Group$="Hexagonal" THEN PRINT;Id;" ";
1580 PRINT;" ";Zd%;
1590 IF NOT Prompt THEN ENDPROC
1600 PRINT;" -any key to continue"
1610 REPEAT:A$=INKEY$(1):PLOT70,Xs,Ys:Now=TIME:REPEAT:UNTIL TIME>Now+10
1620   PLOT69,Xs,Ys:Now=TIME:REPEAT:UNTIL TIME>Now+10:UNTIL A$<>""
1630 PLOT69,Xs,Ys
1640 PLOT69,Xs,Ys
1650 ENDPROC
1660 :
1670 :
1680 DEFFNRMod(X,Y,Z)=SQR(X*X+Y*Y+Z*Z)
1690 DEFFNIMod(X%,Y%,Z%)=SQR(X%*X%+Y%*Y%+Z%*Z%)
1700 DEFFNRDot(X,Y,Z,X1,Y1,Z1)=X*X1+Y*Y1+Z*Z1
1710 DEFFNIDot(X%,Y%,Z%,X1%,Y1%,Z1%)=X%*X1%+Y%*Y1%+Z%*Z1%
1720 :
1730 :
1740 DEFPROCHexplot(Xd%,Yd%,Zd%)
```

```
1750 Id=-(Xd%+Yd%)
1760 IF ABS(Id)>Max% THEN ENDPROC
1770 Zd=(Yd%-Id)*0.57735
1780 Yd=Zd%*A/C
1790 Xd=Xd%
1800 PROCRPlot(Xd,Yd,Zd)
1810 ENDPROC
1820 :
1830 :
1840 DEFPROCTetplot(Xd%,Yd%,Zd%)
1850 Xd=Xd%:Yd=Yd%:Zd=Zd%*A/C
1860 PROCRPlot(Xd,Yd,Zd)
1870 ENDPROC
1880 :
1890 :
1900 DEFPROCRPlot(Xd,Yd,Zd)
1910 IF Rotate THEN Spare=Zd:Zd=Yd:Yd=Xd:Xd=Spare
1920 ModXd=FNRMod(Xd,Yd,Zd)
1930 IF ModXd>Max%+1 OR ModXd=0 THEN ENDPROC
1940 XdotXd=FNRDot(X,Y,Z,Xd,Yd,Zd)
1950 ModX=FNRMod(X,Y,Z)
1960 IF (ModX*ModXd)>1E-20 THEN CosPhi=(XdotXd/(ModX*ModXd))
1970 IF (ModX*ModXd)<=1E-20 THEN CosPhi=SGN(X*Xd)
1980 IF CosPhi>1 THEN CosPhi=1
1990 IF CosPhi<-1 THEN CosPhi=-1
2000 Phi=ACS(CosPhi)*2
2010 Deltheta=0
2020 IF Phi>PI/2:Phi=Phi-PI:Deltheta=PI:IF Phi>PI/2:Phi=Phi-PI:Deltheta=0
2030 IF ABS(Phi)>MaxPhi OR ABS(Phi)<1E-5 THEN ENDPROC
2040 Alpha=XdotXd/(ModX*ModX)
2050 Xb=Xd-Alpha*X
2060 Yb=Yd-Alpha*Y
2070 Zb=Zd-Alpha*Z
2080 NdotXb=FNRDot(-Y,X,0,Xb,Yb,Zb)
2090 ModXb=FNRMod(Xb,Yb,Zb)
2100 ModN=FNRMod(-Y,X,0)
2110 IF (ModN*ModXb)>1E-20THEN CosTheta=NdotXb/(ModN*ModXb)
2120 IF (ModN*ModXb)<=1E-20 THEN CosTheta=SGN(Yb*X)
2130 IF CosTheta>1 THEN CosTheta=1
2140 IF CosTheta<-1 THEN CosTheta=-1
2150 IF Zb<>0 THEN Theta=SGN(Zb)*ACS(CosTheta)+Deltheta
2160 IF Zb=0 THEN Theta=ACS(CosTheta)+Deltheta
2170 PROCPlotpoint
2180 ENDPROC
2190 :
2200 :
2210 DEFPROCHardcopy
2220 Yscale=0.0170:Xscale=0.0101
2230 Xwidth=580:Ywidth=350
2240 INPUT ''"What is the title";Title$
2250 Prompt=FALSE:PROCPlotscreen
2260 PRINT TAB(40,0)"       M.J.Hawton 1986"
2270 PRINT TAB(10,3);Title$
2280 PROCPrint
2290 ENDPROC
2300  REPORT:PRINT;" at line ";ERL
2310 STOP
```

Listing of CRYSTAL.BBC at 22:14:26 on MAY 12, 1987 for CCid=PHP7

```
 10 REM Program to plot local enviroments
 20 REM of ions in a tetragonal crystal structure.
 30 REM
 40 REM M.J.Hawton
 50 REM
 60 MODE 1
 70 PROCSetup
 80 PROCIndicies
 90 PROCOptions
100 REPEAT
110   PROCScreen
120   PRINTTAB(28,0);"R - redraw":PRINTTAB(28,1);"P - printer"
130   A$=GET$
140   IF A$="P" OR A$="p" THEN PRINTTAB(28,0);"          ":PRINTTAB(28,1);"
150   IF A$="R" OR A$="r" THEN PROCDirection:Redraw=TRUE ELSE PROCOptions
160   UNTIL FALSE
170 :
180 DEFPROCSetup
190 VDU19,0,0,0,0,0,0:VDU19,1,1,0,0,0,0:VDU19,2,3,0,0,0,0:VDU19,3,7,0,0,0
200 BG=2:REM Background colour
210 AT=3:REM Atom colour
220 LI=0:REM Line colour
230 Printer=FALSE:Cs=64:REM Sections of circle
240 INPUT"Do you want to take a hard copy";A$
250 IF A$="y" OR A$="Y" THEN Printer=TRUE
260 Xs=1:Xmin=350:Xmax=1200:Ymin=130:Ymax=900
270 REM Stretch in X direction for hard copy.
280 IF Printer THEN Xs=1.108:Xmin=320:Xmax=1090:Ymin=130:Ymax=900
290 La=0.5:Ax=120:Ay=650:Af=1.3:Xd=-16:Yd=16:REM Axis positioning
300 Sx=30:Ls=2.0:Lsy=50:Sy=200:Has=10:REM Scale positioning
310 DIM C%(Cs-1),S%(Cs-1),S(Cs-1),C(Cs-1),L(2),X(2),Y(2),Z(2),Min(2),
          Max(2),Bou%(2)
320 Step=2*PI/Cs:FOR N=0 TO Cs-1:C(N)=COS(N*Step):S(N)=SIN(N*Step):NEXT
330 X(0)=1:X(1)=0.6:X(2)=0
340 Y(0)=0:Y(1)=0.3:Y(2)=1 :REM default directions
350 A=8.8:C=12.19:L(0)=A:L(1)=A:L(2)=C : REM lattice constants
360 Fact=1.3:REM Factor for nearest neighbour criteria
370 Fact2=1.1:REM Factor for drawing nearest neighbour lines
380 Nearest=-1
390 Number%=67
400 DIM Type%(Number%),Pos(Number%,2),Scr(Number%,2)
410 DIM Order%(Number%),Near%(Number%),Neigh%(Number%)
420 DIM Sep%(Number%),Bound%(Number%,2),Size(Number%)
430 ENDPROC
440
450 DEFPROCIndicies
460 FOR N=0 TO Number%
470   READ Type%(N),Pos(N,0),Pos(N,1),Pos(N,2)
480   IF Type%(N)=0 THEN Size(N)=0.5 ELSE Size(N)=0.1
490   IF Type%(N)=1 THEN Size(N)=0.2
500   NEXT
510 ENDPROC
520
530 DEFPROCOptions
535 Redraw=FALSE
540 PROCCls:PRINTTAB(0,3)
550 Cell=FALSE:Connect=FALSE:Connear=FALSE
560 INPUT'"Do you want the whole cell drawn";A$
```

```
570 IF A$="y" OR A$="Y" THEN Cell=TRUE
580 Atom%=-1:IF NOT Cell THEN INPUT''"Which atom do you want";Atom%
590 INPUT''"Do you want the neighbours connected"''"together";A$
600 IF A$="y" OR A$="Y" THEN Connect=TRUE
610 IF NOT Cell THEN INPUT''"Do you want this atom connected to its"'
      "neighbours";A$:IF A$="y" OR A$="Y" THEN Connear=TRUE
620 PROCDirection
630 ENDPROC
640
650 DEFPROCDirection
660 INPUT '"Viewed from which direction"''"(I,J,K)";Di,Dj,Dk
665 PRINT '"Thank you":VDU7
670 Dx=A*Di:Dy=A*Dj:Dz=C*Dk
680 IF Dx=0 AND Dy=0 AND Dz=0 THEN 740
690 IF Dx=0 AND Dy=0 AND Dz<>0 THEN X(0)=1:X(1)=0:X(2)=0:Y(0)=0:Y(1)=1
                              :Y(2)=0:Z(0)=0:Z(1)=0:Z(2)=1:GOTO 740
700 R=SQR(Dx*Dx+Dy*Dy+Dz*Dz):R3=Dx*Dx+Dy*Dy:R1=SQR(R3)
710 X(0)=-Dy/R1:X(1)=+Dx/R1:X(2)=0
720 IF Dz<>0 THEN R2=SQR(R3+R3*R3/(Dz*Dz)):Y(0)=-Dx/R2:Y(1)=-Dy/R2
             :Y(2)=R1*R1/(Dz*R2) ELSE Y(0)=0:Y(1)=0:Y(2)=1
730 Z(0)=Dx/R:Z(1)=Dy/R:Z(2)=Dz/R
740 ENDPROC
750
760 DEFPROCScreen
780 IF Cell THEN PROCDrawcell
790 IF Atom%>=0 AND Atom%<=Number% THEN PROCDrawatom(Atom%)
800 ENDPROC
810
820 DEFPROCNeigh
830 FOR Nu=0 TO Number%:PROCNear(Nu)
840   Neigh%(Nu)=0
850   FOR Nu2=0 TO Number%:IF Near%(Nu2) THEN Neigh%(Nu)=Neigh%(Nu)+1
860     NEXT:PRINT "Atom ";Nu;" has ";Neigh%(Nu);" neighbours.":NEXT
870 ENDPROC
880
890 DEFPROCPosn(I,J,K)
900 X=Xs*((X(0)*L(0)*I+X(1)*L(1)*J+X(2)*L(2)*K)*Scale+Xmin-Xzero)
910 Y=(Y(0)*L(0)*I+Y(1)*L(1)*J+Y(2)*L(2)*K)*Scale+Ymin-Yzero
920 ENDPROC
930
940 DEFPROCPosat(N)
950 X=Xs*(Scr(N,0)*Scale+Xmin-Xzero)
960  Y=Scr(N,1)*Scale+Ymin-Yzero
970 ENDPROC
980
990 DEFPROCScr(N%)
1000 Scr(N%,0)=(X(0)*L(0)*Pos(N%,0)+X(1)*L(1)*Pos(N%,1)+X(2)*L(2)*Pos(N%,2))
1010 Scr(N%,1)=(Y(0)*L(0)*Pos(N%,0)+Y(1)*L(1)*Pos(N%,1)+Y(2)*L(2)*Pos(N%,2))
1020 Scr(N%,2)=(Z(0)*L(0)*Pos(N%,0)+Z(1)*L(1)*Pos(N%,1)+Z(2)*L(2)*Pos(N%,2))
1030 ENDPROC
1040
1050 DEFPROCDraw(S,I,J,K)
1060 IF I=0 THEN PROCDraw(S,1,J,K)
1070 IF J=0 THEN PROCDraw(S,I,1,K)
1080 IF K=0 THEN PROCDraw(S,I,J,1)
1090 PROCPosn(I,J,K):PROCCircle(S,X,Y):ENDPROC
1100 ENDPROC
1110
```

```
1120 DEFPROCDrawat(N):PROCPosat(N)
1130 PROCCircle(Size(N),X,Y):ENDPROC
1140
1150 DEFPROCCircle(R,X,Y)
1160 LOCAL N%:GCOL 0,AT:R=R*Scale
1170 FOR N%=0TOCs-1:C%(N%)=X+Xs*R*C(N%):S%(N%)=Y+R*S(N%):NEXT
1180 FOR N%=0TOCs-4 STEP 2:MOVEX,Y:PLOT4,C%(N%),S%(N%)
          :PLOT85,C%(N%+2),S%(N%+2)
1190   NEXT:MOVEX,Y:PLOT4,C%(Cs-2),S%(Cs-2):PLOT 85,C%(0),S%(0)
1200 GCOL 0,LI:MOVE C%(0),S%(0)
1210 FOR N%=1TOCs-1:DRAW C%(N%),S%(N%):NEXT
1220 DRAW C%(0),S%(0):ENDPROC
1230
1240 DEFPROCConnect(N1,N2)
1250 GCOL 0,LI
1260 PROCPosn(Pos(N1,0),Pos(N1,1),Pos(N1,2)):MOVE X,Y
1270 PROCPosn(Pos(N2,0),Pos(N2,1),Pos(N2,2)):DRAW X,Y
1280 ENDPROC
1290
1300 DEFPROCConat(N2,N1)
1310 LOCAL X1,X2,Z,F:GCOL 0,LI
1320 PROCPosat(N1):MOVE X,Y:X1=X:Y1=Y:PROCPosat(N2)
1330 X=X-X1:Y=Y-Y1:Z=(Scr(N2,2)-Scr(N1,2))*Scale
1340 F=Size(N2)*Scale/SQR(X*X+Y*Y+Z*Z)
1350 PLOT1,X*(1-Xs*F),Y*(1-F)
1360 ENDPROC
1370
1380 DEFPROCDrawcell
1390 Xf=0:Yf=0:Xf2=0:Yf2=0
1400 FOR D%=0 TO 2:Xf=Xf+ABS(X(D%)*L(D%)):Yf=Yf+ABS(Y(D%)*L(D%))
1410   IF X(D%)<0 THEN Xf2=Xf2+X(D%)*L(D%)
1420   IF Y(D%)<0 THEN Yf2=Yf2+Y(D%)*L(D%)
1430   NEXT
1440 Scale=(Xmax-Xmin)/Xf:Sc=(Ymax-Ymin)/Yf
1450 IF Sc<Scale THEN Scale=Sc
1460 Xzero=Xf2*Scale:Yzero=Yf2*Scale
1470 PROCScr(0):N2%=0:Zmin=Scr(0,2)
1480 PROCCls:La=1.5:PROCAxis
1490 PRINTTAB(0,31);"Unit cell  viewed from (";Di;",";Di;",";Dk;").";
                                    TAB(0,0)
1500 FOR N1=1 TO Number%:PROCScr(N1)
1510   IF Scr(N1,2)<Zmin THEN Zmin=Scr(N1,2):N2%=N1
1520   Order%(N1)=-2:NEXT
1530 Order%(N2%)=-1:Fi%=N2%:N3%=N2%
1540 REPEAT:Zmin=999999999:FOR N1=0 TO Number%
1550     IF Order%(N1)<>-2 THEN 1570
1560     IF Scr(N1,2)<Zmin THEN Zmin=Scr(N1,2):N2%=N1
1570     NEXT:IF N2%<>-1 THEN Order%(N3%)=N2%:N3%=N2%:Order%(N2%)=-1
1580   UNTIL Zmin=999999999
1590 N=Fi%:REPEAT:PROCDraw(Size(N),Pos(N,0),Pos(N,1),Pos(N,2))
1600   IF Connect THEN PROCNear(N):PROCConnectnear(N)
1610   N=Order%(N)
1620   UNTIL N=-1
1630 ENDPROC
1640
1650 DEFPROCDrawatom(N%)
1660 LOCAL N1%,N2%,N3%,D%
1670 IF NOT Redraw THEN PROCNear(N%):PROCListnear(N%)
```

```
1680 PROCScr(N%)
1690 N2%=N%:Zmin=Scr(N%,2):Xmin2=Scr(N%,0):Ymin2=Scr(N%,1)
1700 Xmax2=Scr(N%,0):Ymax2=Scr(N%,1)
1710 Near%(N%)=TRUE
1720 FOR N1%=0 TO Number%
1730   IF NOT Near%(N1%) THEN 1850
1740   FOR D%=0 TO 2
1750     IF NOT Bound%(N1%,D%) THEN 1770
1760     IF Pos(N1%,D%)<0.5 THEN Pos(N1%,D%)=Pos(N1%,D%)+1
                            ELSE Pos(N1%,D%)=Pos(N1%,D%)-1
1770     NEXT
1780   PROCScr(N1%)
1790   IF Scr(N1%,0)<Xmin2 THEN Xmin2=Scr(N1%,0)
1800   IF Scr(N1%,0)>Xmax2 THEN Xmax2=Scr(N1%,0)
1810   IF Scr(N1%,1)<Ymin2 THEN Ymin2=Scr(N1%,1)
1820   IF Scr(N1%,1)>Ymax2 THEN Ymax2=Scr(N1%,1)
1830   IF Scr(N1%,2)<Zmin THEN Zmin=Scr(N1%,2):N2%=N1%
1840   Order%(N1%)=-2
1850   NEXT
1860 Scale=(Xmax-Xmin)/(Xmax2-Xmin2):Sc=(Ymax-Ymin)/(Ymax2-Ymin2)
1870 IF Sc<Scale THEN Scale=Sc
1880 Xzero=Xmin2*Scale:Yzero=Ymin2*Scale
1890 Order%(N2%)=-1:Fi%=N2%:N3%=N2%
1900 REPEAT
1910   Zmin=999999999:FOR N1%=0 TO Number%
1920     IF NOT Near%(N1%) OR Order%(N1%)<>-2 THEN 1940
1930     IF Scr(N1%,2)<Zmin THEN Zmin=Scr(N1%,2):N2%=N1%
1940     NEXT:IF N2%<>-1 THEN Order%(N3%)=N2%:N3%=N2%:Order%(N2%)=-1
1950   UNTIL Zmin=999999999
1960 PROCCls
1970 PRINTTAB(0,30);"The ";:PROCPT(N%)
1980 PRINT " atom at ";:PROCPP(N%,0):PRINT;",";:PROCPP(N%,1)
          :PRINT;",";:PROCPP(N%,2):PRINT;".";
1990 PRINT';Nearest;" neigh., viewed from (";Di;",";Dj;",";Dk;").";
2000 PRINT TAB(0,0):PROCAxis
2010 N1%=Fi%:REPEAT
2020   PROCDrawat(N1%)
2030   IF (NOT Connect AND NOT Connear) OR
          (Connect AND NOT Connear AND N1%=N%) THEN 2090
2040   IF Order%(N1%)=-1 THEN 2090
2050   N2%=N1%:REPEAT:N2%=Order%(N2%)
2060     IF (NOT Connear AND N2%=N%) OR
            (NOT Connect AND N2%<>N% AND N1%<>N%) THEN 2080
2070     IF FNSep(N1%,N2%)-(Size(N1%)+Size(N2%))*1000000<Ave*Fact2
            AND NOT Boundary PROCConat(N1%,N2%)
2080     UNTIL Order%(N2%)=-1
2090   N1%=Order%(N1%)
2100   UNTIL N1%=-1
2110 FOR N1%=0 TO Number%
2120   IF NOT Near%(N1%) THEN 2170
2130   FOR D%=0 TO 2
2140     IF Pos(N1%,D%)<0 THEN Pos(N1%,D%)=Pos(N1%,D%)+1
2150     IF Pos(N1%,D%)>=1 THEN Pos(N1%,D%)=Pos(N1%,D%)-1
2160     NEXT
2170   NEXT
2180 ENDPROC
2190
2200 DEFPROCPT(N%)
```

```
2210 IF Type%(N%)=0 PRINT "Nd";ELSE IF Type%(N%)=1 PRINT "Fe";
                          ELSE PRINT "B";
2220 ENDPROC
2230
2240 DEFPROCPP(N%,D%):Pos%=Pos(N%,D%)*1000+0.5:PRINT;Pos%/1000;:ENDPROC
2250
2270 DEFPROCAxis
2280 GCOL 0,L1
2290 PROCPosn(0,0,0):Xz=X:Yz=Y:MOVE Ax,Ay
2300 PROCPosn(La/L(0),0,0):DRAW Ax+X-Xz,Ay+Y-Yz
2310 MOVE Ax+Af*(X-Xz)+Xd,Ay+Af*(Y-Yz)+Yd:VDU5:PRINT "X":VDU4
2320 MOVE Ax,Ay:PROCPosn(0,La/L(1),0):DRAW Ax+X-Xz,Ay+Y-Yz
2330 MOVE Ax+Af*(X-Xz)+Xd,Ay+Af*(Y-Yz)+Yd:VDU5:PRINT "Y":VDU4
2340 MOVE Ax,Ay:PROCPosn(0,0,La/L(2)):DRAW Ax+X-Xz,Ay+Y-Yz
2350 MOVE Ax+Af*(X-Xz)+Xd,Ay+Af*(Y-Yz)+Yd:VDU5:PRINT "Z":VDU4
2360 MOVE Sx,Sy:PLOT1,Lsy,0:MOVE Sx,Sy
2370 PROCPosn(0,0,Ls/L(2)):DRAW Sx+X-Xz,Sy+Y-Yz:PLOT1,Lsy,0
2380 MOVE Sx+0.7*Lsy,Sy:DRAW Sx+0.7*Lsy,Sy+(Y-Yz)/2-20
2390 MOVE Sx+0.7*Lsy,Sy+(Y-Yz)/2+20:DRAW Sx+0.7*Lsy,Sy+(Y-Yz)
2400 MOVE Sx+0.7*Lsy-16,Sy+16:DRAW Sx+0.7*Lsy,Sy:DRAWSx+0.7*Lsy+16,Sy+16
2410 MOVE Sx+0.7*Lsy-16,Sy-16+Y-Yz:DRAW Sx+0.7*Lsy,Sy+Y-Yz
                  :DRAWSx+0.7*Lsy+16,Sy-16+Y-Yz
2420 MOVE Sx+20,Sy+(Y-Yz)/2+16:VDU5:PRINT"0.2nm":VDU4
2430 ENDPROC
2440
2450 DEFPROCCls
2460 CLS:COLOUR L1:COLOUR 128+BG:GCOL 0,BG:MOVE 0,0:MOVE0,1024
       :PLOT85,1280,1024:MOVE 0,0:MOVE1280,0:PLOT85,1280,1024
2470 GCOL 0,L1:MOVE 10,1018:VDU5:PRINT"Structure of Nd";:PLOT 0,0,-16
          :PRINT"2";:PLOT 0,0,16:PRINT"Fe";:PLOT 0,0,-16:PRINT"14";
          :PLOT 0,0,16:PRINT"B.";:VDU4,23,1,0;0;0;0;
2480 ENDPROC
2490
2500 DEFPROCConnectnear(N)
2510 LOCAL M:FOR M=0 TO Number%
2520   IF Near%(M) AND NOT (Bou%(0) OR Bou%(1) OR Bou%(2))
                    THEN PROCConnect(N,M)
2530   NEXT
2540 ENDPROC
2550
2560 DEFPROCOrder(N%)
2570 LOCAL N1%,N2%,N3%,N4%
2580 FOR N1%=0TONumber%:Near%(N1%)=0:Order%(N1%)=N1%+1
                          :Sep%(N1%)=FNSep(N%,N1%)
2590   FOR N2%=0 TO 2:Bound%(N1%,N2%)=Bou%(N2%):NEXT
2600   NEXT
2610 Fi%=0:IF N%=0 THEN Fi%=1
2620 N4%=-1
2630 REPEAT
2640   Max%=Sep%(Fi%):N1%=Fi%:N2%=-1:N3%=Order%(Fi%)
2650   REPEAT
2660     IF Sep%(N3%)>Max% THEN Max%=Sep%(N3%):N2%=N1%
2670     N1%=N3%:N3%=Order%(N3%)
2680     UNTIL N3%=-1
2690   IF N2%=-1 THEN N1%=Order%(Fi%):Order%(Fi%)=N4%:N4%=Fi%:Fi%=N1%
                 ELSE N1%=Order%(N2%):Order%(N2%)=Order%(N1%)
                          :Order%(N1%)=N4%:N4%=N1%
2700   UNTIL Order%(Fi%)=-1
```

```
2710 Order%(N%)=Fi%:Order%(Fi%)=N4%
2720 PRINT "At Two Time=";TIME
2730
2740 End=FALSE:N3%=1:N1%=Order%(N%):Ave=Sep%(N1%):Near%(N1%)=TRUE
2750 REPEAT
2760   IF Sep%(N1%)<Ave*Fact THEN Ave=(Ave*N3%+Sep%(N1%))/(N3%+1)
                    :Near%(N1%)=TRUE:N3%=N3%+1 ELSE End=TRUE
2770   N1%=Order%(N1%)
2780   UNTIL End
2790 ENDPROC
2800
2810 DEFPROCNear(N%)
2820 LOCAL N1%,N2%
2830 FOR N1%=0TONumber%:Order%(N1%)=-2:Near%(N1%)=0:Sep%(N1%)=FNSep(N%,N1%)
2840   FOR N2%=0 TO 2:Bound%(N1%,N2%)=Bou%(N2%):NEXT:NEXT
2850 Fi%=0:IF N%=0 THEN Fi%=1
2860 Min%=Sep%(Fi%):N1%=Fi%:REPEAT:N1%=N1%+1
2870   IF N1%=N% THEN 2890
2880   IF Min%>Sep%(N1%) THEN Min%=Sep%(N1%):Fi%=N1%
2890   UNTIL N1%=Number%
2900 Ave=Sep%(Fi%):Near%(Fi%)=TRUE:Order%(N%)=Fi%:N2%=1:N3%=Fi%:End%=FALSE
2910 REPEAT:Fi%=-1:REPEAT:Fi%=Fi%+1:UNTIL Fi%<>N% AND NOT Near%(Fi%)
2920   Min%=Sep%(Fi%):N1%=Fi%:REPEAT:N1%=N1%+1
2930     IFN1%<>N% AND Min%>Sep%(N1%) AND NOT Near%(N1%)
                THEN Min%=Sep%(N1%):Fi%=N1%
2940     UNTIL N1%=Number%
2950   IF Min%<Ave*Fact THEN Ave=(Ave*N2%+Min%)/(N2%+1):Order%(N3%)=Fi%
                :Near%(Fi%)=TRUE:N2%=N2%+1:N3%=Fi% ELSE End%=TRUE
2960   UNTIL End%
2970 Nearest=N2%:Order%(N3%)=-1
2980 ENDPROC
2990
3000 DEFPROCListnear(N%)
3010 PROCCls
3020 IF Printer THEN VDU2
3030 PRINTTAB(0,3) "Nearest neighbours of ";:PROCPT(N%)
                :PRINT" atom (";Atom%;")."
3040 N1%=0:PRINT '"                    I      J      K        Sepn."'
3050 REPEAT
3060   PRINT;N1%;")";TAB(5);:PROCPT(N%):PRINT;"(";N%;")";TAB(12);
3070   PROCPP(N%,0):PRINTTAB(18);:PROCPP(N%,1):PRINTTAB(24);:PROCPP(N%,2)
3080   PRINTTAB(30);((Sep%(N%)+50) DIV 100)/10000
3090   N%=Order%(N%):N1%=N1%+1:UNTIL N%=-1
3100 PRINT''
3110 IF Printer THEN VDU3
3120 PRINT"PRESS A KEY TO CONTINUE";:A$=GET$:VDU7:PRINT" - THANK YOU ";
3130 IF Printer THEN VDU2,12,3
3140 ENDPROC
3150
3160 DEFFNSep(N1%,N2%)
3170 LOCAL J,M1%,Sepn:Sepn=0:Boundary=FALSE
3180 FOR M1%=0 TO 2:Bou%(M1%)=FALSE:Sep=ABS(Pos(N1%,M1%)-Pos(N2%,M1%))
3190   IF Sep>0.5 THEN Bou%(M1%)=TRUE:Boundary=TRUE:Sep=1-Sep
3200   J=L(M1%)*Sep:Sepn=Sepn+J*J:NEXT
3210 =1000000*SQR(Sepn)
3220
3230 REM Nd f sites
3240 DATA 0,0.26984,0.26984,0,          0,0.73016,0.73016,0
```

```
3250  DATA 0,0.76984,0.23016,0.5,       0,0.23016,0.76984,0.5
3260 REM Nd g sites
3270 DATA 0,0.85722,0.14278,0,          0,0.14278,0.85722,0
3280 DATA 0,0.35722,0.35722,0.5,        0,0.64278,0.64278,0.5
3290 REM Fe k1 sites
3300 DATA 1,0.22417,0.56755,0.12752,  1,0.56755,0.22417,0.12752
3310 DATA 1,0.77583,0.43245,0.12752,  1,0.43245,0.77583,0.12752
3320 DATA 1,0.27583,0.06755,0.37248,  1,0.06755,0.27583,0.37248
3330 DATA 1,0.72417,0.93245,0.37248,  1,0.93245,0.72417,0.37248
3340 DATA 1,0.27583,0.06755,0.62752,  1,0.06755,0.27583,0.62752
3350 DATA 1,0.72417,0.93245,0.62752,  1,0.93245,0.72417,0.62752
3360 DATA 1,0.22417,0.56755,0.87248,  1,0.56755,0.22417,0.87248
3370 DATA 1,0.77583,0.43245,0.87248,  1,0.43245,0.77583,0.87248
3380 REM Fe k2 sites
3390 DATA 1,0.03727,0.35963,0.17626,  1,0.35963,0.03727,0.17626
3400 DATA 1,0.96273,0.64037,0.17626,  1,0.64037,0.96273,0.17626
3410 DATA 1,0.46273,0.85963,0.32374,  1,0.85963,0.46273,0.32374
3420 DATA 1,0.53727,0.14037,0.32374,  1,0.14037,0.53727,0.32374
3430 DATA 1,0.46273,0.85963,0.67626,  1,0.85963,0.46273,0.67626
3440  DATA 1,0.53727,0.14037,0.67626,  1,0.14037,0.53727,0.67626
3450 DATA 1,0.03727,0.35963,0.82374,  1,0.35963,0.03727,0.82374
3460 DATA 1,0.96273,0.64037,0.82374,  1,0.64037,0.96273,0.82374
3470 REM Fe j1 site
3480 DATA 1,0.09790,0.09790,0.20449, 1,0.90210,0.90210,0.20449
3490 DATA 1,0.59790,0.40210,0.29551, 1,0.40210,0.59790,0.29551
3500 DATA 1,0.59790,0.40210,0.70449, 1,0.40210,0.59790,0.70449
3510 DATA 1,0.09790,0.09790,0.79551, 1,0.90210,0.90210,0.79551
3520 REM Fe j2 site
3530 DATA 1,0.31764,0.31764,0.24566, 1,0.68236,0.68236,0.24566
3540 DATA 1,0.81764,0.18236,0.25434, 1,0.18236,0.81764,0.25434
3550 DATA 1,0.81764,0.18236,0.74566, 1,0.18236,0.81764,0.74566
3560 DATA 1,0.31764,0.31764,0.75434, 1,0.68236,0.68236,0.75434
3570 REM Fe e site
3580 DATA 1,0.5,    0.5,    0.11478,  1,0.5,    0.5,    0.88522
3590 DATA 1,0,      0,      0.38522,  1,0,      0,      0.61478
3600 REM Fe c site
3610 DATA 1,0,      0.5,    0,        1,0.5,    0,      0
3620 DATA 1,0,      0.5,    0.5,      1,0.5,    0,      0.5
3630 REM B sites
3640  DATA 2,0.37736,0.62264,0,        2,0.62264,0.37736,0
3650  DATA 2,0.12264,0.12264,0.5,      2,0.87736,0.87736,0.5
```

Listing of MUTUAL.BBC at 22:29:08 on MAY 12, 1987 for CCid=PHP7

```
 10 REM   Calculation of the angular dependance
 20 REM   of the mutual inductance of a
 30 REM   circular and a square coil
 40 REM    - the square coil inclined at an
 50 REM   angle Phi from the normal to the
 60 REM   circular coil
 70 REM
 80 REM   M.J.Hawton   Durham  Sept 84
 90 REM
100 MODE 0
110 Open=FALSE:ON ERROR GOTO 310
120 PROCOpenfile
130 PROCDraw
140 PROCAsk
150 PROCInit
160 FOR Angle=0 TO 360 STEP Step
170   Phi=Angle*PI/180
180   PROCCalc(Phi)
190   Fact=1:Prefix$=""
200   IF ABS(M)<1 THEN Fact=1E-3:Prefix$="milli"
210   IF ABS(M)<1E-3 THEN Fact=1E-6:Prefix$="micro"
220   IF ABS(M)<1E-6 THEN Fact=1E-9:Prefix$="nano"
230   IF ABS(M)<1E-9 THEN Fact=1E-12:Prefix$="pico"
240   PRINT#File,Angle,M
250   M%=M/(Fact*1E-3):M=M%/1E3
260   PRINT"Angle=";Angle;"Mut. Ind.=";M;" ";Prefix$;" Henrys"
270   NEXT
280 CLOSE#File
290 END
300 :
310 IF Open CLOSE#File
320 REPORT:PRINT" at line ";ERL
330 END
340 :
350 DEFPROCInit
360 N%=N%*4
370 DIM cos(N%),sin(N%)
380 delta=2*PI/N%
390 FOR I%=1TON%
400   Theta=delta*(I%-0.5)
410   cos(I%)=COS(Theta):sin(I%)=SIN(Theta)
420   NEXT
430 ENDPROC
440 :
450 DEFPROCDraw
460 PROCTitle
470 MOVE950,10:DRAW 950,990
480 MOVE 800,950:DRAW 1150,950:DRAW 1030,650
490                DRAW 750,650:DRAW 870,950
500 MOVE 890,650:DRAW 890,630
510 MOVE 700,800:DRAW 1280,800
520 MOVE 1090,800:DRAW 1280,870
530 MOVE 700,200:DRAW 1200,200
540 MOVE 1150,200
550 FOR Phi=.1 TO 6.3 STEP .1
560   Y%=100*SIN(Phi)
570   X%=950+200*COS(Phi)+Y%/3
580   Y%=200+Y%
```

```
590    DRAW X%,Y%:NEXT
600 MOVE 950,200:DRAW 780,240
610 MOVE 665,95:DRAW 1235,305
620 MOVE 665,695:DRAW 1235,905
630 MOVE 1182,838
640 FOR PHI=0.5 TO 1.3 STEP 0.1
650    X%=1090+100*COS(PHI):Y%=800+100*SIN(PHI)
660    DRAW X%,Y%:NEXT
670 MOVE 1130,840:DRAW 1170,755
680 MOVE 820,930:DRAW840,950:DRAW840,930
690 MOVE 975,240:DRAW975,760
700 MOVE 960,220:DRAW 975,200:DRAW990,220
710 MOVE 960,780:DRAW 975,800:DRAW990,780
720 MOVE 962,220:DRAW 975,202:DRAW988,220
730 MOVE 962,780:DRAW 975,798:DRAW988,780
740 MOVE 780,820:DRAW780,800:DRAW800,820
750 MOVE 790,825:DRAW830,925
760 PRINTTAB(50,4);"a"
770 PRINTTAB(46,12);"<----b--->"
780 PRINTTAB(54,24);"r"
790 PRINTTAB(60,17);"h"
800 PRINTTAB(74,8);"Phi"
810 PRINTTAB(0,4)
820 ENDPROC
830 :
840 DEFPROCAsk
850 INPUT "width(a)(mm)=";a:a=a/1000
860 INPUT ''"width(b)(mm)=";b:b=b/1000
870 INPUT ''"Turns=";Turns1
880 INPUT ''"separation(h)(mm)=";h:h=h/1000
890 INPUT ''"radius(r)(mm)=";r:r=r/1000
900 INPUT ''"Turns=";Turns2
910 PRINT#File,"a="+STR$(a)+",b="+STR$(b)+",h="+STR$(h)+",r="+STR$(r)
920 PRINT#File,1,"Angle"," ","Pickup"
930 INPUT ''"Steps for the""integration (3-20)=";N%;''
940 PRINT#File,STR$(N%)+" int. steps"
950 INPUT '"Size of angular step in Phi (deg)";Step
960 ENDPROC
970 :
980 DEFPROCOpenfile
990 PROCTitle
1000 REPEAT
1010    INPUT"What file for storage""(* for catalogue)";File$
1020    IF File$="*" THEN *CAT 0
1030    IF File$="*" THEN *CAT 2
1040    UNTIL File$<>"*"
1050 File=OPENOUT(File$)
1060 Open=TRUE
1070 PRINT#File,"Calculation of Mutual Inductance"
1080 ENDPROC
1090 :
1100 DEFPROCTitle
1110 CLS:PRINT"Mutual Inductance"
1120 PRINT"-----------------"''
1130 ENDPROC
1140 :
1150 DEFPROCCalc(Phi)
1160 M=0
```

```
1170 CosPhi=COS(Phi):SinPhi=SIN(Phi)
1180 delA=8*a/N%:delB=8*b/N%
1190 FOR A=-a TO +a STEP 2*a
1200   FOR B=-b+delB/2 TO b  STEP delB
1210     FOR I%=1TON%
1220       PROCCalcR
1230       IF A=-a THEN M=M-r*delB*delta*sin(I%)/R
1240       IF A=a THEN M=M+r*delB*delta*sin(I%)/R
1250       NEXT:NEXT:NEXT
1260 B=b
1270 FOR A=-a+delA/2 TO a STEP delA
1280   FOR I%=1TON%
1290     PROCCalcR
1300     M=M+2*r*cos(I%)*CosPhi*delA*delta/R
1310     NEXT:NEXT
1320 M=M*1E-7*Turns1*Turns2
1330 ENDPROC
1340 DEFPROCCalcR
1350 R=SQR((h+A*SinPhi)^2+(B-r*cos(I%))^2+(A*CosPhi-r*sin(I%))^2)
1360 ENDPROC
1370 :
```

Listing of TORQUE.BBC at 22:50:26 on MAY 12, 1987 for CCid=PHP7

```
 10 REM   ATOM   torque measurements
 20 REM   ----
 30 REM
 40 REM  M.Hawton (Durham July 1985)
 50 REM
 60 MODE 7
 70 PROCSetup
 80 PROCProgram
 90 Finished=TRUE
100 PRINT
110 *FX2,0
120 *FX3,0
130  ON ERROR GOTO 200
140 IF Turn=0 THEN Turn=5
150 FOR Angle=Turn TO 0 STEP -10*SGN(Turn)
160   Ang=(Angle+720) MOD 360
170   IF Ang>5 PROCCurrent(Setcurrent,Ang)
180   NEXT
190 FOR Ang=15 TO 45 STEP 15:PROCCurrent(Setcurrent,Ang):NEXT
200  ON ERROR GOTO 240
210 *FX2,0
220 *FX3,0
230 PROCClose
240 IF NOT Finished REPORT:PRINT;" at line ";ERL
250 PROCCom(1,"Thank you for using this experiment")
     :PROCCom(2,"I'm now relaxing")
255 IF Finished FOR Sound= 1 TO 20:SOUND 1,-15,105,2:SOUND 1,-15,100,2
     :SOUND 1,-15,95,2:NEXT
260 IF Rereads%<>0 PROCCom(3,"Broodeal reread "+STR$(Rereads%)+" times")
270 VDU28,0,24,39,0
280 END
290 :
300 DEFPROCSetup
310 DIM Val(10),One(3),Rat(360)
320 VDU28,0,24,39,9
330 Wobbles=0:IEEEMini=6:IEEEWeston=9:?&FE62=3:BD=2:REM BD=No of brookdeals
340 Switch%=0:Rereads%=0:ReadBs=5:REM Brookdeal readings
350 Finished=FALSE:Title$="":ReadWs=3:REM Weston readings
360 ON ERROR GOTO 110
370 Fileopen=FALSE:Oldangle=0:Turn=0
380 PROCTitle:INPUT"What is the title of the run";Title$
390 PROCTitle
400 INPUT"What current setting (up to 4000)";Setcurrent
410 DIM Shunt(1):INPUT'"What is the shunt range"'"(H, M or L)";Sh$
420 IF Sh$="H" OR Sh$="h" THEN Shunt(1)=0.0740:Shunt(0)=0.0749
     :Sh$="High power"
430 IF Sh$="M" OR Sh$="m" THEN Shunt(1)=0.9886:Shunt(0)=1.0789
     :Sh$="Medium pow."
440 IF Sh$="L" OR Sh$="l" THEN Shunt(1)=10.076:Shunt(0)=10.031
     :Sh$="Low power"
450 IF Sh$<>"Low power" AND Sh$<>"Medium pow." AND Sh$<>"High power"
     THEN GOTO 410
460 PRINT '"Are you using a solenoid or split pair"'"(S/P)":A$=GET$:VDU7
470 IF A$="P" OR A$="p" THEN Pair=TRUE ELSE Pair=FALSE
480 IF Pair THEN PROCCom(7,"Split Pair") ELSEPROCCom(7,"Solenoid")
490 PROCIEEE
500 ENDPROC
510 :
```

```
520 DEFPROCInst
530 IF Pair THEN PROCRelay(-1):PROCRelay(-2):PROCRelay(8)
            ELSE PROCRelay(1):PROCRelay(2):PROCRelay(-8)
540 PROCCom(1,"Setting up Minicam")
550 FOR Ang=30 TO -0.001 STEP -15:PROCCurrent(Setcurrent,0):NEXT
560 PROCCom(1,"Please switch on Power Supply")
565 INPUT"How long do you want me to wait(mins)"'" ";Mins
570 S$="Waiting "+STR$(Mins)+" mins.":PROCCom(2,S$)
580 PROCCom(7,"Thank you, I'll do it all now"):PROCDelay(6000*Mins)
585 FOR Sound= 1 TO 20:SOUND 1,-15,105,2:SOUND 1,-15,100,2:NEXT
590 PROCCom(1,"Setting up Brookdeal")
600 FOR I=1 TO BD:PROCBd(I):PROCSendb("H 1;Y 0;D 0;M 0;O 0,0;F 1;T 6")
      :PROCSendb("A1 0;S 4;X 1;L 7;W 1"):NEXT
610 PROCCom(1,"Setting up Weston")
620 PROCSend("W","S0M0D0T1")
630 PROCCom(1,""):PROCSet
640 PROCRelay(-2-4*Pair):PROCSwitch
650 ENDPROC
660 :
670 DEFPROCYesNo
680 PRINT"(please answer Y or N)"
690 PROCCom(2,"Waiting for your reply")
700 REPEAT:A$=GET$:UNTIL A$="Y" OR A$="y" OR A$="n" OR A$="N"
710 IF A$="n" OR A$="N" THEN No=TRUE :Yes=FALSE ELSE Yes=TRUE : No=FALSE
720 PROCCom(2,"")
730 ENDPROC
740 :
750 DEFPROCDelay(D%)
760 PROCCom(3,"Waiting patiently")
770 D%=D%+TIME:REPEAT UNTIL TIME>D%
780 PROCCom(3,"")
790 ENDPROC
800 :
810 :
820 DEFPROCSwitch
830 PROCCom(2,"Switching Minicam Relays")
840 S$="DA1,42,"+STR$(Switch%)
850 PROCSend("M",S$)
860 PROCCom(2,"")
870 ENDPROC
880 :
890 DEFPROCBd(I):?&FE60=2*I-1:ENDPROC
900 :
910 DEFPROCRelay(N)
920 IF N>0 THEN Switch%=Switch% OR 2^(N-1)
          ELSE Switch%=NOT(NOT(Switch%) OR 2^(-1-N))
930 ENDPROC
940 :
950 DEFPROCCurrent(Mag,Ang)
960 IF ABS(FNMod(Oldangle-Ang+180)-180)>20
      THEN PROCCom(22,STR$(Ang)+" more than 20 degrees from")
           :PROCCom(23,STR$(Oldangle)+" -press a key to cont.")
           :SOUND 1,-15,60,100:A$=GET$
970 Turn=Turn+FNMod(Ang-Oldangle+180)-180
980 PROCCom(2,"Setting current")
990 Xc%=Mag*COS(RAD(Ang+180))
1000 Yc%=Mag*SIN(RAD(Ang+180))
1010 PROCRelay(-4*SGN(Yc%)):PROCRelay(-5*SGN(Xc%)):PROCSwitch
```

```
1020 PROCSend("M","DA1,16,"+STR$(ABS(Xc%)))
1030 PROCSend("M","DA1,17,"+STR$(ABS(Yc%)))
1040 Oldangle=Ang
1050 PROCCom(2,""):PROCCom(6,"Current angle set at "+STR$(Oldangle))
1060 ENDPROC
1070 :
1080 DEFPROCReadM(Ad%)
1090 PROCCom(2,"Reading Minicam a-d number "+STR$(Ad%))
1100 AD$="Ad1,"+STR$(Ad%)
1110 PROCSend("M",AD$)
1120 PROCCom(2,"")
1130 ENDPROC
1140 :
1150 DEFPROCTitle
1160 VDU28,0,24,39,0:CLS
1170 PRINTTAB(0,4);:VDU131,157,132,136
                   :PRINT"  ATOM (2."+STR$(BD)+") torque measurements"
1180 L%=LEN(Title$):PRINTTAB(0,5);:VDU131,157,132:PROCSp(17-L%/2)
                   :PRINT;Title$
1190 PROCCom(6,"")
1200 ENDPROC
1210 :
1220 DEFPROCSend(A$,S$)
1230 LOCAL Ad%,B$
1240 ?&FE60=0
1250 B$=LEFT$(A$,1)
1260 IF B$="M" OR B$="m" THEN Ad%=32+IEEEMini
1270 IF B$="W" OR B$="w" THEN Ad%=32+IEEEWeston
1280 IF B$="P" OR B$="p" THEN Ad%=36
1290 VDU2,1,27,21
1300 PRINT "WS";CHR$(Ad%);",";S$
1310 VDU6,3
1320 IF Ad%<>36 THEN PROCReply(A$)
1330 ENDPROC
1340 :
1350 DEFPROCReply(A$)
1360 LOCAL Ad%,B$,Name$
1370 ?&FE60=0
1380 B$=LEFT$(A$,1)
1390 IF B$="M" OR B$="m" THEN Ad%=64+IEEEMini:Name$="Minicam"
1400 IF B$="W" OR B$="w" Ad%=64+IEEEWeston:Name$="Weston"
1410 IF B$="T" Name$="Thurlby":*FX156,20,227
1420 PROCCom(3,"Waiting for reply from "+Name$)
1430 IF B$<>"T" VDU2,1,27,21:PRINT "RS";CHR$(Ad%);:VDU6,3 ELSE ?&FE60=2
1440 *FX2,1
1450 A=GET:Reply$="":REPEAT
1460   IF A<>10 AND A<>21 THEN Reply$=Reply$+CHR$(A)
1470   A=GET:UNTIL A=13
1480 ?&FE60=0:*FX15,0
1490 *FX2,2
1500 IF B$="T" THEN *FX156,8,227
1510 PROCCom(3,""):ENDPROC
1520 :
1530 DEFPROCSendb(S$)
1540 LOCAL A$,A,I
1550 *FX2,2
1560 VDU2:FOR I=1 TO LEN(S$):VDU 1,ASC(MID$(S$,I,1)):NEXT:VDU1,13,3
1570 PROCCom(3,"Waiting for reply from Brookdeal")
```

```
1580 A$="":Replys=0:*FX2,1
1590 REPEAT:A=GET
1600    IF A=13 OR A=44 THEN Replys=Replys+1:Val(Replys)=VAL(A$):A$=""
                        :ELSE A$=A$+CHR$(A)
1610   UNTIL A=42
1620 *FX2,2
1630 PROCCom(3,"")
1640 ENDPROC
1650 :
1660 DEFPROCIEEE
1670 ?&FE60=0
1680 *FX6,10
1690 PROCReset("03*")
1700 *FX3,7
1710 VDU27:PRINT"CN";
1720 *FX3,0
1730 ENDPROC
1740 :
1750 DEFPROCPrinter
1760 REM Set up clearway to Printer
1770 ?&FE60=0
1780 *FX6,0
1790 PROCReset("20-")
1800 ENDPROC
1810 :
1820 DEFPROCReset(S$)
1830 *FX7,7
1840 *FX8,7
1850 *FX156,8,227
1860 *FX5,2
1870 *FX2,2
1880 VDU2,21,1,2
1890 PRINT;STRING$(64,CHR$(13));"N!!!!!!!!!!!!";S$
1900 PRINT;"Y";
1910 VDU6,3
1920 *FX2,1
1930 REPEAT:A$=INKEY$(4):UNTIL A$=""
1940 *FX2,2
1950 ENDPROC
1960 :
1970 DEFPROCCom(N,C$)
1980 LOCAL X,Y,L%,X%
1990 VDU28,0,24,39,0
2000 L%=LEN(C$):X=POS:Y=VPOS:X%=19-L%/2
2010 PRINT TAB(0,N);" ";:PROCSp(X%)
2020 PRINTC$;:PROCSp(38-X%-L%)
2030 PRINTTAB(X,Y)"";
2040 VDU28,0,24,39,9
2050 ENDPROC
2060 :
2070 DEFPROCSp(N)
2080 LOCAL N1
2090 FOR N1=1 TO N:PRINT" ";:NEXT
2100 ENDPROC
2110 :
2120 DEFPROCSet:PROCCom(2,"Phasing Brookdeal"):FOR I=1 TO BD:PROCBd(I)
          :PROCSendb("A2 1"):PROCSettle:PROCSendb("A2 1"):PROCSettle
           :PROCSendb("A1 1"):NEXT:ENDPROC
```

```
2130 REMDEFPROCSet:FOR I=1 TO BD:PROCBd(I):PROCSendb("A1 1"):NEXT:ENDPROC
2140 :
2150 DEFPROCSettle
2160 LOCAL SI:FOR SI=0 TO 1
2170   PROCCom(2,"Waiting for Brookdeal to settle")
         :End=TIME+FNTime:REPEAT:UNTIL TIME>End
2180   REPEAT:PROCSendb("Z"):UNTIL (Val(1) MOD 64)DIV 32 =1
2190   NEXT
2200 ENDPROC
2210 :
2220 DEFFNTime:PROCSendb("T"):=3^(Val(1) MOD 2)*10^(2-INT((Val(1)+1)/2))
           *.5^Val(2)*1200:REM Calculates settle time(sec/100)
2230 :
2240 DEFFNSens:PROCSendb("X"):Fact=1/(10^Val(1)):PROCSendb("S")
         := Fact*5^INT(((Val(1)+2) MOD 3)/2)*2^INT(((Val(1)+1) MOD 3)/2)
             *10^(-INT((Val(1))/3)):REM Calculates sensitivity/Volts
2250 :
2260 DEFPROCReadb(N)
2270 LOCAL Va,BI:PROCCom(2,"Reading Brookdeal")
2280 Pause=FNTime*.8:Start=TIME+Pause/2
2290 REPEAT
2300   REPEAT:PROCSendb("Z"):UNTIL (Val(1) MOD 64)DIV 32 =1
2310   Next=TIME+Pause:Va=1:O1=(Val(1) MOD 32)DIV 16
2320   X=0:Y=0:Sens=FNSens
2330   REPEAT:UNTIL TIME>Start
2340   FOR BI=1 TO N
2350     PROCSendb("Q1"):X=Val(1)+X
2360     Y=0
2370     IF TIME>Next PROCSendb("Z"):Va=Va*((Val(1) MOD 64) DIV 32)
             :O1=O1+((Val(1) MOD 32) DIV 16):Next=TIME+Pause
2380     NEXT
2390   PROCSendb("Z"):Va=Va*((Val(1) MOD 64) DIV 32)
           :O1=O1+((Val(1) MOD 32) DIV 16)
2400   X=X*Sens/(2*N):Y=Y*Sens/(2*N)
2410   IF (Va<>1) OR (O1<>0) THEN Rereads%=Rereads%+1
2420   UNTIL (Va=1) AND (O1=0)
2430 PROCCom(2,"")
2440 ENDPROC
2450 :
2460 DEFPROCReadw(N)
2470 LOCAL Count,I:Volts=0
2480 PROCCom(2,"Reading Weston")
2490 Count=0:REPEAT
2500   FOR I=1 TO N
2510     PROCReply("W")
2520     Left=1:B$="":A$=MID$(Reply$,Left,1):REPEAT:B$=B$+A$
                 :Left=Left+1:A$=MID$(Reply$,Left,1):UNTIL A$=","
2530     Volts=Volts+VAL(B$):NEXT
2540   Volts=Volts/(N*1000)
2550   III%=Switch%/128
2560   Current=Volts/Shunt(III%)
2570   Count=Count+1:UNTIL Count>10 OR Volts<>0
2580 PROCCom(2,"")
2590 ENDPROC
2600 :
2610 DEFPROCOpen
2620 IF Fileopen THEN ENDPROC
2630 REPEAT
```

```
2640    INPUT "What file for data storage""(type * for catalogues)";File$
2650    PRINT:IF File$="*" THEN *CAT 0
2660    UNTIL File$<>"*"
2670 F%=OPENOUT(":0."+File$):Fileopen=TRUE
2680 REPEAT:INPUT "Which file has correlation data"'
                  "(type * for catalogue)";Cor$:PRINT
2690    IF Cor$="*" THEN *CAT 0
2700    UNTIL Cor$<>"*"
2710 INPUT "What sample";Sample$
2720 INPUT "What temperature (Kelvin) ";Temp$
2730 INPUT "What field (Tesla)";Field
2740 INPUT "What is the date";Date$
2750 INPUT "What is the TC resistance(ohms)";Res$
2755 INPUT "What is the needle valve flow ";Flow$
2760 INPUT "What is aux heater voltage(volts)";Heater$
2770 PRINT#F%,"ATOM (2."+STR$(BD)+") - torque measurements"
2780 PRINT#F%,Title$,6,"Rotation/Degrees","","Torque","AX pickup/mV",
           "BX pickup/mV","A current/A","B current/A","Thermocouple/mV"
2790 PRINT#F%,Date$,"Sample is "+Sample$,"Temperature is "+Temp$+"K"
2795 PRINT#F%,"Field is "+STR$(Field)+"Tesla","Correlation file "+Cor$,
                 "Data file "+File$
2800 PRINT#F%,Sh$+". current="+STR$(Setcurrent)
2810 PRINT#F%,"Temp. Cont.="+Res$+"ohms(He flow="+Flow$+")."
2820 IF VAL(Heater$)<>0 THEN PRINT#F%,"Aux heater at "+Heater$+"Volts."
2830 PRINT "Any comment ..."
2840 REPEAT:INPUT Com$:PRINT#F%,Com$:UNTIL Com$=""
2850 ENDPROC
2860 :
2870 DEFPROCClose
2880 IF Fileopen=FALSE THEN ENDPROC
2890 CLOSE#F%
2900 *DRIVE 0
2910 Fileopen=FALSE
2920 ENDPROC
2930 :
2940 DEFPROCReadC
2950 Cor%=OPENIN(Cor$)
2960 FOR I%=0 TO 359:INPUT#Cor%,I:Rat(I%)=I:NEXT
2970 FOR I%=0 TO 3:INPUT#Cor%,I:One(I%)=I:NEXT
2980 CLOSE#Cor%:ENDPROC
2990 :
3000 DEFPROCReadT
3010 REPEAT:PROCReply("T"):UNTIL LEN(Reply$)=10
3020 TC=VAL(MID$(Reply$,2,8))
3030 ENDPROC
3040 :
3050 :
3060 DEFPROCProgram
3070 @%=&20207
3080 PROCOpen
3090 PRINT "Do you want a details run?(Y/N)":A$=GET$:
     IF A$<>"N" AND A$<>"n" AND A$<>"Y" AND A$<>"y" THEN VDU7,7,7:GOTO 3090
3100 IF A$="N" OR A$="n" THEN Centre=0:Range=360
3110 IF A$="Y" OR A$="y" THEN INPUT"Centred around what angle(deg)";Centre
3120 IF A$="Y" OR A$="y" THEN INPUT"With what angular range(deg)";Range
3130 Centre=(((Centre+540)*1000) MOD 360000)/1000 - 180
3140 Start=Centre-Range/2-90:Endturn=Centre+Range/2+90
3150 INPUT"What angular step size(deg)";Step
```

```
3160 Stread=Centre-Range/2-Step*2.5:Endread=Centre+Range/2+Step*2.5
3170 PROCInst:PROCReadC:Control=FALSE
3180 PROCCom(8," Angle      Actual          Torque      ")
3190 PROCCom(1,"Taking readings")
3200 CLS:FOR Angle=0 TO Start-Step STEP SGN(Start-Step)*10
       :PROCCurrent(Setcurrent,Angle):NEXT
3210 FOR Angle=Start TO Stread STEP SGN(Step)*10
       :PROCCurrent(Setcurrent,Angle):NEXT
3220 For%=1
3230 FOR Angle=Stread TO Endread+Step/2 STEP Step:PROCReading(Angle):NEXT
3240 FOR Angle=Endread+10 TO Endturn+10 STEP SGN(Step)*10
       :PROCCurrent(Setcurrent,Angle):NEXT
3250 FOR Angle=Endturn TO Endread STEP -10*SGN(Step)
       :PROCCurrent(Setcurrent,Angle):NEXT
3260 For%=-1
3270 FOR Angle=Endread TO Stread-Step/2 STEP -Step:PROCReading(Angle):NEXT
3280 FOR Angle=Stread TO 0 STEP SGN(-Stread)*10
       :PROCCurrent(Setcurrent,Angle):NEXT
3290 ENDPROC
3300
3310 DEFPROCRead
3320 PROCReadT:PROCCom(7,"Thermocouple reading ="+STR$(TC))
3330 PROCRelay(8):IF BD=1 PROCRelay(-1):PROCSwitch:PROCBd(1):PROCSettle
3340 IF BD=2:PROCSwitch:PROCDelay(50)
3350 PROCReadw(ReadWs):AC=Current
3360 IF BD=2:PROCBd(2) ELSE PROCBd(1)
3370 PROCReadb(ReadBs):AX=X
3380 PROCRelay(-8):IF BD=1 PROCRelay(1):PROCSwitch:PROCBd(1)
       :PROCSettle:PROCReadw(ReadWs):BC=Current
3390 IF BD=2:PROCSwitch
3400 PROCBd(1)
3410 PROCReadb(ReadBs):BX=X :IF BD=2 PROCReadw(ReadWs):BC=Current
3420 IF ABS(AX)<1 AND ABS(BX)<1 THEN
       PROCCom(2,"Strange readings from Brookdeal")
       :PROCCom(6,"-press a key to continue"):SOUND1,-15,70,50:A=GET
3430 IF Pair XX=AX:AX=BX:BX=XX
3440 Phi=(45*SGN(BX)*(AX<0)-135*(BX<0)-45*(ABS(AX)>ABS(BX))
     *SGN(AX)*SGN(BX)+360) MOD 360:REM Calculates quadrant for Phi
3450 Ratio=(AX/BX):IF ABS(Ratio)>1 THEN Ratio=1/Ratio
3460 Sign=((Phi MOD 90)=45)*2 + 1:Slope=Sign*SGN(Ratio)
3470 Phi=FNMod(Phi-45*((Phi MOD 90)=45)+Sign*DEG(ASN(ABS(Ratio))/2))
     :REM Calculates rough value for Phi from resemblance to SIN curve
3480 IF Pair Phi=((Phi*100 +27000) MOD 36000)/100
3490 Cusp=FALSE
3500 FOR I=0 TO 3
3510   IF ABS(45+90*I-Phi)<45 THEN Side=Phi-One(I):Near=I
                :IF (45+90*I-Phi)*(One(I)-Phi)<0
                      THEN Phi=2*One(I)-Phi:Side=Phi-One(I)
3520   NEXT
3530 P%=INT(Phi):X2=P%:Y2=Rat(P%):PROCInc
3540 IF (X1-One(Near))*(X2-One(Near))<0 THEN PROCCusp:REM at cusp
3550 REPEAT
3560   IF (Y2-Ratio)*(Y1-Ratio)>0 AND (Y2-Ratio)*Slope<0 PROCInc
3570   IF (Y2-Ratio)*(Y1-Ratio)>0 AND (Y1-Ratio)*Slope>0 PROCDec
3580   IF (X1-One(Near))*(X2-One(Near))<0 THEN PROCCusp:REM at cusp
3590   UNTIL (Y1-Ratio)*(Y2-Ratio)<=0 OR Cusp
                                :REM now between consecutive points
3600  IF NOT(Cusp) THEN Phi=FNMod((Ratio-Y1)/(Y2-Y1)+X1)
```

```
                                        :REM Linearly interpolated
3610 ENDPROC
3620 :
3630 DEFPROCInc:X1=X2:Y1=Y2:IF X1>359.9 THEN X1=0
3640 P%=(P%+1) MOD 360:X2=P%:IF X2<0.01 THEN X2=360
3650 Y2=Rat(P%):ENDPROC
3660 DEFPROCDec:X2=X1:Y2=Y1::IF X2<0.01 THEN X2=360
3670 P%=(P%+359) MOD 360:X1=P%:Y1=Rat(P%):ENDPROC
3680 :
3690 DEFPROCReading(Angle)
3700 Angle=FNMod(Angle)
3710 IF Wobbles>0 Swing=Angle+5*Wobbles+5:FOR Sw%=1 TO Wobbles
       :PROCCurrent(Setcurrent,Angle):Swing=(ABS(Swing-Angle)-5)*-1^Sw%*-1
             *For%+Angle:PROCCurrent(Setcurrent,Swing):NEXT
3720 PROCCurrent(Setcurrent,Angle)
3730 PROCRead
3740 Torque=Field*(AC*0.0261*SIN(RAD(Phi))-BC*0.0253*COS(RAD(Phi)))
3750 IF AC<>0 THEN Angl=DEG(ATN(BC*0.0253/(AC*0.0261)))
              ELSE IF Angl>180 Angl=270 ELSE Angl=90
3760 IF AC<0 THEN Angl=Angl+180
3770 Angl=((1000*Angl+360000)MOD 360000)/1000
3780 PRINT;TAB(3);Angl;TAB(13);Phi;
3790 @%=&20810:PRINT;TAB(27);Torque;
3800 @%=&90A
3810 PRINT#F%,Phi,Torque,AX,BX,AC,BC,TC
3820 ENDPROC
3830 :
3840 DEFFNMod(N):=((N*1000+1080000) MOD 360000)/1000
3850 :
3860 DEFPROCCusp : REM tests if at cusp.
3870 IF (X1-One(Near))*Side>0 THEN X3=X1:Y3=Y1 ELSE X3=X2:Y3=Y2
                    :REM Choose side
3880 Cusp=TRUE
3890 Phi=FNMod((Ratio-Y3)*(One(Near)-X3)/(1-Y3)+X3):REM Linearly interpolated
3900 ENDPROC
```

Listing of FIT.BBC at 22:25:06 on MAY 12, 1987 for CCid=PHP7

```
 10 REM   Curve fitting to angular correlation measurement
 20 REM     Produces correlation file for P.TORQUE
 30 REM
 40 REM          - M.J.Hawton  Durham  1985
 50 MODE0
 60 Finished=FALSE
 70 PROCSetup
 80 PROCOpen
 90 PROCFit
100 PROCOut
110 Finished=TRUE
120 IF NOT Finished REPORT:PRINT " in line ";ERL
130 CLOSE#Filein%
140 CLOSE#Fileout%
150 PRINT "(";TIME/100;"sec) ";
160 END
170 :
180 DEFPROCSetup
190 Filein%=0:Fileout%=0:TIME=0
200 ON ERROR GOTO 120
210 DIM X(10),Y(10,2),A(10,2),M(7,7),YI(6,2),S(1,1,360),ch%(900),One(3)
220 FOR I=0 TO 10:X(I)=-100:Y(I,1)=1:Y(I,2)=1:NEXT
230 Ones=0:CLS :PRINT "Curve fitting program"''
240 ENDPROC
250 :
260 DEFPROCOpen
270 PRINT"From which file (* for catalogue)"
280 REPEAT:INPUT File$
290   IF File$="*" THEN *CAT 0
300   IF File$="*" THEN *CAT 2
310   UNTIL File$<>"*"
313 PRINT"From P.ANGLE or P.TORQUE (A or T)":A$=GET$:Angle=TRUE
                :IF A$="t" OR A$="T" THEN Angle=FALSE
320 Filein%=OPENIN(File$)
330 C%=0:REPEAT:C%=C%+1:ch%(C%)=BGET#Filein%:UNTIL ch%(C%)=&FF
340 PTR#Filein%=PTR#Filein%-1
350 :
360 PRINT "What file for the output"''"   F.File - Curve fit"
370 PRINT "    R.File - ratios"'"   C.File - correlation file"
380 PRINT "     (* for catalogue)"
390 REPEAT:INPUT "File=",File$
400   IF File$="*" THEN *CAT 0
410   IF File$="*" THEN *CAT 2
420   UNTIL File$<>"*"
430 Fileout%=OPENOUT("F."+File$)
440 ENDPROC
450 :
460 DEFPROCFit
470 PRINT '"(";TIME/100;"sec) Fitting  . . ."'
480 FOR A%=-180 TO 180
490   PROCfit(A%,+1,6)
500   NEXT
510 FOR B%=180 TO -180 STEP -1
520   PROCfit(B%,-1,6)
530   NEXT
540 ENDPROC
550 :
560 DEFPROCRange(P%,Q%,R%)
```

```
570 FOR S%=1 TO 6:REPEAT
580       IF ABS(X(S%)-P%)>180 THEN X(S%)=X(S%)+360*SGN(P%-X(S%))
590       UNTIL ABS(X(S%)-P%)<=180:NEXT
600 REPEAT
610   IF Q%*X(3) < Q%*P% OR  Q%*X(4) > Q%*P%  THEN PROCLoad
620   REPEAT
630     IF ABS(X(1)-P%)>180 THEN X(1)=X(1)+360*SGN(1-X(1))
640     UNTIL ABS(X(1)-P%)<=180
650   UNTIL Q%*X(3) > Q%*P% AND Q%*X(4) < Q%*P%
660 ENDPROC
670 :
680 DEFPROCfit(E%,F%,G2%)
690 IF F%*X(3)<F%*E% OR F%*X(4)>F%*E%
                        PROCRange(E%,F%,G2%):PROCSolv(G2%,1,2)
700 S((F%+1)/2,0,(E%+360) MOD 360)=FNY(E%,G2%,1)
710 S((F%+1)/2,1,(E%+360) MOD 360)=FNY(E%,G2%,2)
720 ENDPROC
730 :
740 DEFPROCLoad
750 FOR U%=10 TO 2 STEP -1
760   X(U%)=X(U%-1):Y(U%,1)=Y(U%-1,1):Y(U%,2)=Y(U%-1,2)
770   NEXT
780 IF NOT EOF#File in% AND Angle
                  THEN INPUT#File in%,X(1),Y(1,1),Y(1,2),AA,BB
785 IF EOF#File in% OR Angle
      ELSE INPUT#File in%,AA,BB,Y(1,1),Y(1,2),AC,BC,TT
      :IF AC<>0 THEN X(1)=DEG(ATN(BC*0.0253/(AC*0.0261)))
          ELSE IF X(1)>180 THEN X(1)=270 ELSE X(1)=90
786 IF EOF#File in% OR Angle ELSE IFAC<0 THEN X(1)=X(1)+180
787 IF EOF#File in% OR Angle ELSE X(1)=((1000*X(1)+360000)MOD 360000)/1000
790 PRINT;" ";X(1);
800 ENDPROC
810 :
820 DEFPROCSolv(N2%,Z1%,Z2%)
830 FOR Z%=Z1% TO Z2%:FOR G%=1 TO 6:YI(G%,Z%)=Y(G%,Z%):NEXT:NEXT
840 FOR I%=1 TO N2%:FOR J%=1 TO N2%:M(I%,J%)=X(I%)^(J%-1):NEXT:NEXT
850 FOR I%=N2%-1 TO 1 STEP -1:FOR J%=N2% TO I%+1 STEP -1
860     MI=M(I%,J%)/M(J%,J%):FOR K%=1 TO N2%
                      :M(I%,K%)=M(J%,K%)*MI-M(I%,K%):NEXT
870     FOR Z%=Z1% TO Z2%:YI(I%,Z%)=YI(J%,Z%)*MI-YI(I%,Z%):NEXT:NEXT:NEXT
880 FOR Z%=Z1% TO Z2%:A(1,Z%)=YI(1,Z%)/M(1,1):NEXT
890 FOR J%=2 TO N2%:FOR I%=1 TO J%-1
900     FOR Z%=Z1% TO Z2%:YI(J%,Z%)=YI(J%,Z%)-M(J%,I%)*A(I%,Z%):NEXT
910     NEXT:FOR Z%=Z1% TO Z2%:A(J%,Z%)=YI(J%,Z%)/M(J%,J%):NEXT:NEXT
920 ENDPROC
930 :
940 DEFPROCOut
950 PRINT ''"(";TIME/100;"sec) Saving curve fits in F.";File$
960 FOR D%=1 TO C%-1:BPUT#File out%,ch%(D%):NEXT
970 FOR A%=0 TO 359
980   PRINT#File out%,A%,S(0,0,A%),S(1,0,A%),S(0,1,A%),S(1,1,A%):NEXT
990 CLOSE#File out%
1000
1010 PRINT "(";TIME/100;"sec) Calculating ratios"
1020 FOR A%=0 TO 359:S(0,0,A%)=(S(0,0,A%)+S(1,0,A%))/(S(0,1,A%)+S(1,1,A%))
1030   IF ABS(S(0,0,A%))<1 THEN S(0,1,A%)=S(0,0,A%)
                           ELSE S(0,1,A%)=1/S(0,0,A%)
1040   NEXT
```

Listing of FIT.BBC at 22:25:06 on MAY 12, 1987 for CCid=PHP7

```
1050
1060 PRINT "(";TIME/100;"sec) Finding cusps in curve"
1070 FOR A%=1 TO 358:Find=FALSE:IF ABS(S(0,1,A%))>ABS(S(0,1,A%-1)) Find=TRUE
1080   IF Find AND ABS(S(0,1,A%))>ABS(S(0,1,A%+1)) Find=TRUE ELSE Find=FALSE
1090   IF Find AND ABS(S(0,1,A%))>0.85 THEN  PROCFind1(A%)
1100   NEXT
1110
1120 PRINT "(";TIME/100;"sec) Saving ratios in R.";File$
1130 Fileout%=OPENOUT("A."+File$)
1140 FOR D%=1 TO C%-1:BPUT#Fileout%,ch%(D%):NEXT
1150 FOR A%=0 TO 359:PRINT#Fileout%,A%,S(0,1,A%),0,0,0:NEXT
1160 CLOSE#Fileout%
1170
1180 PRINT "(";TIME/100;"sec) Saving correlation file C.";File$
1190 Fileout%=OPENOUT("C."+File$)
1200 FOR A%=0 TO 359:PRINT#Fileout%,S(0,1,A%):NEXT
1210 FOR A%=0 TO 3:PRINT#Fileout%,One(A%):NEXT
1220 ENDPROC
1230 :
1240 DEFPROCFind1(I%)
1250 FOR W%=1 TO 5:X%=I%+W%-3:X(W%)=X%:Y(W%,1)=S(0,0,X%):NEXT
1260 One(Ones)=FNZero(I%):PRINT One(Ones):Ones=Ones+1:ENDPROC
1270 :
1280 DEFFNZero(I%)
1290 PROCSolv(5,1,1):X=X(3):Last=FNY(X,5,1):Del=0.5
1300 REPEAT:X=X+Del:New=FNY(X,5,1)
1310   IF Last<>New Del=Del*(New-SGN(New))/(Last-New):Last=New
1320   UNTIL ABS(Del)<0.0001
1330 =X
1340 :
1350 DEFFNY(X,K%,Z%)
1360 LOCAL Yi,L
1370 Yi=0:FOR Y%=1 TO K%:Yi=Yi+A(Y%,Z%)*X^(Y%-1):NEXT:=Yi
>*SPOOL
```

Listing of TORCALC.BBC at 22:35:21 on MAY 12, 1987 for CCid=PHP7

```
 10 REM  Recalculate torques
 20 REM
 30 REM   M.J.Hawton     January 1986
 40 REM
 50 MODE 0
 60 CLS:PRINT"Recalculate Torques"
 70 PRINT"-------------------"
 80 PRINT''
 90 REM Torque formula
100 Torque$="Field*(AC*0.02537*SIN(RAD(Phi))-BC*0.02664*COS(RAD(Phi)))"
110 Open=FALSE
120 ON ERROR PROCError
130 REPEAT
140   REPEAT
150     End=FALSE
160     PRINT "From which file (or *N for catalogue)?"
170     INPUT File$
180     IF File$="*0" OR File$="*" THEN  *CAT 0
190     IF File$="*1" THEN *CAT 1
200     IF File$="*2" THEN *CAT 2
210     IF File$="*3" THEN *CAT 3
220     UNTIL File$<>"*" AND File$<>"*1" AND File$<>"*2"
                         AND File$<>"*3" AND File$<>"*0"
230   File=OPENIN(File$)
240   PRINT "Into which file"
250   INPUT Fileout$
260   Fileout=OPENOUT(Fileout$)
270   Open=TRUE
280   FieldFound=FALSE
290   REPEAT
300     char=BGET#File
310     PTR#File=PTR#File-1
320     IF char=&40 THEN PROCInteger
330     IF char=&FF THEN PROCData
340     IF char=&00 THEN PROCString
350     IF char<>&40 AND char<>&FF AND char<>&00 AND NOT EOF#File
                                    THEN PROCHex
360     IF EOF#File THEN PRINT ''"End of file"'' :End = TRUE
370     UNTIL End=TRUE
380   PROCExit
390   :
400   DEFPROCInteger
410   INPUT#File,In%
420   PRINT;"Integer";TAB(20);In%
430   PRINT#Fileout,In%
440   ENDPROC
450   :
460   DEFPROCString
470   INPUT#File,In$
480   PRINT In$
490   IF RIGHT$(In$,5)="Tesla" AND LEFT$(In$,9)="Field is "
          THEN Field=VAL(MID$(In$,10,LEN(In$)-14)):FieldFound=TRUE
500   PRINT#Fileout,In$
510   ENDPROC
520   :
530   DEFPROCHex
540   In=BGET#File
550   PRINT;"Hex found ";~In;"  (Decimal:";In;")"
```

Listing of TORCALC.BBC at 22:35:21 on MAY 12, 1987 for CCid=PHP7

```
560   ENDPROC
570   :
580   DEFPROCData
590   IF NOT(FieldFound) THEN INPUT "What is the field ";Field
600   Endoffile=FALSE
610   PRINT '''"Calculating Torque using:"''"      Torque=";Torque$''","."
615   PRINT#Fileout,"Torque recalculated using",Torque$
620   Count=-1
630   Count2=1
640   REPEAT
650     Phi=FNIn
660     Torque=FNIn
670     AX=FNIn
680     BX=FNIn
690     AC=FNIn
700     BC=FNIn
710     IF NOT(Endoffile) THEN PROCOut
720     Count=Count+1
730     PRINT ;~Count;
740     IF Count>14 THEN Count=-1:Count2=Count2+1:PRINTTAB(Count2);
750     UNTIL Endoffile
760   ENDPROC
770   :
780   DEFPROCOut
790   Torque=EVAL(Torque$)
800   PRINT#Fileout,Phi,Torque,AX,BX,AC,BC
810   ENDPROC
820   :
830   DEFFNIn
840   LOCAL X
850   IF NOT EOF#File THEN INPUT#File,X ELSE Endoffile=TRUE:X=0
860   =X
870   :
880   DEFPROCError
890   REPORT
900   PRINT " at line ";ERL
910   IF ERR=199 THEN PRINT "Hit any key to continue":A$=GET$
                    :PTR#File=PTR#File+256:GOTO 270
920   IF Open=TRUE THEN CLOSE#File:CLOSE#Fileout
930   PROCExit
940   :
950   DEFPROCExit
960   IF Open=TRUE THEN CLOSE#File:CLOSE#Fileout
970   PRINT"press Y to continue"
980   PRINT"      N to return to menu"
990   PRINT"      Q to quit"
1000  Awn$=GET$
1010  IF Awn$="Y" OR Awn$="y" THEN GOTO 60
1020  IF Awn$="N" OR Awn$="n" THEN CHAIN "$.MENU"
1030  VDU26:CLS
1040   END
```

Listing of ROTHYS.BBC at 22:32:46 on MAY 12, 1987 for CCid=PHP7

```
 10 REM   ATOM - rotation Hysterysis calc.
 20 REM   ----
 30 REM  M.Hawton (Durham July 1985)
 40 REM
 50 REM      Using Straight line fits
 60 REM
 70 VDU3,15
 80 MODE 7
 90 PROCSetup
100 PROCProgram
110 Finished=TRUE
120 IF NOT Finished THEN PRINT"Last error was ";
                         :REPORT:PRINT;" at line ";ERL
130 PROCClose
140 VDU28,0,24,39,0
150 END
160 :
170 DEFPROCSetup
180 VDU28,0,24,39,9
190 Fileopen=FALSE:Title$="":Title2$=""
200 Finished=FALSE:Straight=FALSE:Out=FALSE:First=TRUE
210 Poly=FALSE:Print=FALSE
220 ON ERROR GOTO 120
230 PROCTitle
240 DIM X(200),Y(200),M(6,6),A(6),YI(6)
250 ENDPROC
260 :
270 DEFPROCTitle
280 VDU28,0,24,39,0
290 CLS
300 PRINTTAB(0,4);:VDU131,157,132,136:PRINT"     Rotational Hysteresis Calc"
310 PRINTTAB(0,5);:VDU131,157,132:PROCSpace(17-LEN(Title$)/2):PRINTTitle$
320 PRINTTAB(0,6);:VDU131,157,132:PROCSpace(17-LEN(Title2$)/2):PRINTTitle2$
330 VDU28,0,24,39,9
340 PRINTTAB(0,0);
350 ENDPROC
360 :
370 DEFPROCSpacebar
380 LOCAL A$
390 *FX21,0
400 PRINT''"Press the space bar to continue"
410 REPEAT:A$=GET$:UNTIL A$=" "
420 PRINT "Thank you"''
430 ENDPROC
440 :
450 DEFPROCPrinter
460 *FX6,0
470 PROCReset("20-")
480 ENDPROC
490 :
500 DEFPROCReset(S$)
510 *FX7,7
520 *FX8,7
530 *FX5,2
540 VDU2,21,1,2
550 PRINT;STRING$(64,CHR$(13));"N!!!!!!!!!!!!";S$
560 PRINT;"Y";
570 VDU6,3
```

```
580 *FX2,1
590 REPEAT:A$=INKEY$(2):UNTIL A$=""
600 *FX2,0
610 ENDPROC
620 :
630 :
640 DEFPROCSpace(N)
650 LOCAL N1
660 N1=1:REPEAT:PRINT" ":N1=N1+1:UNTIL N1>N
670 ENDPROC
680 :
690 DEFPROCOpen
700 IF Fileopen THEN ENDPROC
710 REPEAT
720   INPUT "Which file is data stored in""(type * for catalogue)"";File$
730   IF File$="*" THEN *CAT 0
740   IF File$="*" THEN *CAT 2
750   UNTIL File$<>"*"
760 File=OPENIN(File$)
770 Fileopen=TRUE
780 IF Print THEN VDU2
790 INPUT#File,Title$,Title2$
800 PROCTitle
810 PRINT "From data file: "+File$
820 REPEAT:PRINT'"Do you want a straight line fit (Y/N)";
830   A$=GET$:PRINT;A$
840   IF A$="Y" OR A$="y" THEN Straight=TRUE
850   PRINT"Do you want a polynomial fit (Y/N)";
860   A$=GET$:PRINT;A$
870   IF A$="Y" OR A$="y" THEN Poly=TRUE
880   IF NOT Poly AND NOT Straight THEN VDU7
890   UNTIL Poly OR Straight
900 IF Poly PRINT "Do you want a plot file(Y/N)":A$=GET$
                          :PRINT;A$:IF A$="Y" OR A$="y" THEN Out=TRUE
910 IF Out THEN PROCOpenout
920 IF Out THEN PRINT#Fileout,Title$,Title2$
930 INPUT#File,NofYs,Xaxis$,Yaxis$
940 NofYs=NofYs-1
950 IF Out THEN PRINT#Fileout,1,Xaxis$,Yaxis$
960 DIM Y$(NofYs),Comment$(20)
970 FOR I=0 TO NofYs
980   INPUT#File,Y$(I):NEXT
990 IF Out THEN PRINT#Fileout,Y$(0)
1000 NofCs=0:Comment$(0)=""
1010 REPEAT:PROCTest
1020   IF String THEN NofCs=NofCs+1:INPUT#File,Comment$(NofCs)
1030   UNTIL NOT String
1040 FORI=0 TO NofCs
1050   PRINT Comment$(I)
1060   IF Out THEN PRINT#Fileout,Comment$(I)
1070   NEXT
1080 IF Out THEN PRINT#Fileout,"Rotational Hysteresis Fit"
1090 ENDPROC
1100 :
1110 DEFPROCTest
1120 REM Tests next bit of data
1130 Integer=FALSE:Real=FALSE:String=FALSE
1140 char=BGET#File
```

```
1150 PTR#File=PTR#File-1
1160 IF char=&40 THEN Integer=TRUE
1170 IF char=&FF THEN Real=TRUE
1180 IF char=&00 THEN String=TRUE
1190 ENDPROC
1200 :
1210 DEFFNLoad
1220 IF NOT EOF#File AND (Integer OR Real) :INPUT#File,Var:=Var
1230 =0
1240 :
1250 DEFPROCClose
1260 IF Fileopen THEN CLOSE#File
1270 IF Out THEN CLOSE#Fileout
1280 *DRIVE 0
1290 ENDPROC
1300 :
1310 DEFPROCProgram
1320 PRINT"Do you want a hard copy (Y/N)";
1330 A$=GET$:PRINTA$
1340 IF A$="Y" OR A$="y" THEN Print=TRUE
1350 IF Print THEN PROCPrinter
1360 PROCOpen
1370 I%=0
1380 PRINT "Reading in data . . . "
1390 REPEAT
1400    I%=I%+1
1410    X(I%)=FNLoad
1420    Y(I%)=FNLoad:FOR J%=1 TO NofYs:Rubbish=FNLoad:NEXT
1430    UNTIL EOF#File
1440 Nofpts=I%
1450 IF Straight PROCCalcstraight :PROCPrint
1460 IF Poly PROCCalcpoly :PROCPrint
1470 ENDPROC
1480 :
1490 DEFPROCCalcstraight
1500 PRINT "Calculating straight line fit . . "
1510 I%=1
1520 REPEAT: I%=I%+1:UNTIL X(I%) > 180
1530 Area=FNArea(180,Y(I%-1)+(180-X(I%-1))*(Y(I%)-Y(I%-1))/
                        (X(I%)-X(I%-1)),X(I%),Y(I%))
1540 I%=I%+1
1550 REPEAT
1560    Area=Area+FNArea(X(I%-1),Y(I%-1),X(I%),Y(I%))
1570    I%=I%+1
1580    UNTIL X(I%) < 180
1590 REPEAT
1600    Area=Area+FNArea(X(I%-1),Y(I%-1),X(I%),Y(I%))
1610    I%=I%+1
1620    UNTIL X(I%) >180
1630 Area=Area+FNArea(X(I%-1),Y(I%-1),180,Y(I%-1)+(180-X(I%-1))*
                                (Y(I%)-Y(I%-1))/(X(I%)-X(I%-1)))
1640 REPEAT: I%=I%+1:UNTIL X(I%) < 180
1650 Area1=Area:Area=0
1660 Area=FNArea(X(I%),Y(I%),180,Y(I%)+(180-X(I%))*
                        (Y(I%-1)-Y(I%))/(X(I%-1)-X(I%)))
1670 I%=I%+1
1680 REPEAT
1690    Area=Area+FNArea(X(I%),Y(I%),X(I%-1),Y(I%-1))
```

```
1700    I%=I%+1
1710    UNTIL X(I%) > 180
1720  REPEAT
1730    Area=Area+FNArea(X(I%),Y(I%),X(I%-1),Y(I%-1))
1740    I%=I%+1
1750    UNTIL X(I%) < 180
1760  Area=Area+FNArea(180,Y(I%)+(180-X(I%))*(Y(I%-1)-Y(I%))/
                                        (X(I%-1)-X(I%)),X(I%-1),Y(I%-1))
1770  Area2=Area
1780  ENDPROC
1790  :
1800  DEFPROCPrint
1810  PRINT '"Area Forwards=";Area1;" Nmdeg"
1820  PRINT "Area Backwards=";Area2;" Nmdeg"
1830  PRINT "Difference=";Area1-Area2;" Nmdeg"
1840  PRINT "Average hysteresis=";(Area1-Area2)/360;" Nm"
1850  PRINT
1860  ENDPROC
1870  :
1880  DEFFNArea(X1,Y1,X2,Y2)
1890  Diff=X1-X2
1900  IF ABS(Diff)>180 THEN Diff=Diff-360*SGN(Diff)
1910  Ar=(Y1+Y2)*(Diff)/2
1920  REM PRINT "From ";X1;" to ";X2;", area=";Area;"+";Ar;"=";Area+Ar
1930  =Ar
1940  :
1950  DEFPROCOpenout
1960   PRINT"Which file for curve fit"'" (* for catalogue)"
1970  REPEAT:INPUT File$
1980    IF File$="*" THEN *CAT 0
1990    IF File$="*" THEN *CAT 2
2000    UNTIL File$<>"*"
2010  Fileout=OPENOUT(File$)
2020  ENDPROC
2030  :
2040  :
2050  :
2060  DEFPROCCalcpoly
2070  PRINT "Calculating Polynomial fit"
2080  Area=0
2090  Start=-180
2100  Finish=180
2110  FOR I=-180 TO 180
2120    PROCfit(I,+1,6)
2130    NEXT
2140  Area1=Area
2150  Start=180:Finish=-180
2160  Area=0
2170  FOR I=180 TO -180 STEP -1
2180    PROCfit(I,-1,6)
2190    NEXT
2200  Area2=-Area
2210  ENDPROC
2220  :
2230  :
2240  DEFPROCRange(I,J,K)
2250  REPEAT
2260    IF J*X(3) > J*I OR  J*X(4) < J*I  THEN PROCNext
```

```
2270    REPEAT
2280      IF ABS(X(6)-I)>180 THEN X(6)=X(6)+360*SGN(I-X(6))
2290      UNTIL ABS(X(6)-I)<=180
2300    UNTIL J*X(3) < J*I AND J*X(4) > J*I
2310  ENDPROC
2320  :
2330  :
2340  DEFPROCFit(I,J,K)
2350  FOR II=1 TO 6:REPEAT
2360      IF ABS(X(II)-I)>180 THEN X(II)=X(II)+360*SGN(I-X(II))
2370      UNTIL ABS(X(II)-I)<=180
2380    NEXT
2390  IF First OR J*X(3)>J*I OR J*X(4)<J*I THEN PROCRange(I,J,K):PROCSolv(6)
2400  IF J*X(3)>J*I OR J*X(4)<J*I THEN PROCRange(I,J,K):PROCSolv(6)
2410  IF ((((I-Start)*J*100)+36000) MOD 36000)<18000 THEN PROCAddarea(J,K)
2420  Yi=0:FOR L=1 TO K:Yi=Yi+A(L)*I^(L-1):NEXT
2430  REM PRINT I;TAB(18);Yi
2440  IF Out PRINT#Fileout,(I+360)MOD360,Yi
2450  First=FALSE
2460  ENDPROC
2470  :
2480  :
2490  DEFPROCNext
2500  FOR III=0 TO Nofpts - 1
2510    X(III)=X(III+1)
2520    Y(III)=Y(III+1)
2530    NEXT
2540  X(Nofpts)=X(0)
2550  Y(Nofpts)=Y(0)
2560  ENDPROC
2570  :
2580  DEFPROCSolv(N)
2590  LOCALM,I,J,K
2600  FOR GI=1 TO 6:YI(GI)=Y(GI):NEXT
2610  FOR I=1 TO N:FOR J=1 TO N
2620      M(I,J)=X(I)^(J-1)
2630      NEXT:A(I)=0:NEXT
2640  FOR I=2 TO N
2650    FOR J=1 TO I-1
2660      MI=M(I,J)
2670      FOR K=1 TO N
2680        M(I,K)=M(I,K)-M(J,K)*MI/M(J,J)
2690        NEXT
2700      YI(I)=YI(I)-YI(J)*MI/M(J,J)
2710      NEXT:NEXT
2720  FOR J=N TO 1 STEP -1
2730    FOR I=1 TO N
2740      IF I<>J THEN YI(J)=YI(J)-M(J,I)*A(I)
2750      NEXT
2760    A(J)=YI(J)/M(J,J)
2770    NEXT
2780  ENDPROC
2790  :
2800  :
2810  DEFPROCAddarea(J,K)
2820  Begin=Start:Last=Finish
2830  IF (X(4)*J) < (Last*J)  THEN Last=X(4)
2840  Start=Last
```

```
2850 Yb=0:Y1=0
2860 FOR L=1 TO K
2870    Yb=Yb+A(L)*Begin^(L)/L
2880    Y1=Y1+A(L)*Last^(L)/L
2890    NEXT
2900 REM PRINT "From ";Begin;" to ";Last;", area=";Area;"+";Y1-Yb;"=";
2910 Area=Area+Y1-Yb
2920 REM PRINT Area
2930 ENDPROC
```

Listing of SKETCH.BBC at 00:18:33 on MAY 13, 1987 for CCid=PHP7

```
 10 REM   Sketchin Program
 20 REM
 30 REM  M.Hawton (Durham Feb 1985)
 40 REM
 50 REM  -  plots results on screen
 60 REM
 70 VDU3,15
 80 MODE 7
 90 PROCSetup
 95 PROCOptions
 98 MODE 0
100 PROCProgram
110 Finished=TRUE
120 IF Pixy PROCPen(0)
130 IF Pixy PROCMove(0,0)
140 *FX2,0
150 *FX3,0
160 PROCClose
170 IF NOT Finished THEN PRINT"Last error was ";
          :REPORT:PRINT;" at line ";ERL
190 VDU28,0,2,79,0
210 END
220 :
230 DEFPROCSetup
240 VDU28,0,24,39,9
250 Fileopen=FALSE:Title$="":Title2$=""
260 Finished=FALSE:Pixy=FALSE
270 ON ERROR GOTO 120
280 PROCTitle
290 *FX7,7
300 *FX8,7
310 *FX5,2
320 *FX2,2
340 *FX2,2
350 DIM Max(1),Min(1),Scale(1),Unit(1),Num(1),Diff(1),Fact(1)
360 ENDPROC
370 :
380 DEFPROCPen(I%):ENDPROC
390 :
400 DEFPROCMove(I%,J%):MOVE I%/2,J%/2+28:ENDPROC
410 :
420 DEFPROCDraw(I%,J%):DRAW I%/2,J%/2+28:ENDPROC
430 :
440 DEFPROCCharactersize(I%):ENDPROC
450 :
460 DEFPROCXaxis(Length%,Number%):LOCAL K%
      :FOR K%=1 TO Number%:PLOT 1,Length%/2,0
463   IF ((K% MOD 2) = 1) AND ((Length% MOD 2)=1) THEN PLOT 1,1,0
465   PLOT 0,0,5:PLOT 1,0,-10:PLOT 0,0,5:NEXT:ENDPROC
470 :
480 DEFPROCYaxis(Length%,Number%):LOCAL K%
      :FOR K%=1 TO Number%:PLOT 1,0,Length%/2
483   IF ((K% MOD 2) = 1) AND ((Length% MOD 2)=1) THEN PLOT 1,0,1
485   PLOT 0,5,0:PLOT 1,-10,0:PLOT 0,5,0:NEXT:ENDPROC
490 :
500 DEFPROCCurve(N1%,N2,Xscale,Yscale)
680 ENDPROC
690 :
```

```
 700 DEFPROCPrint(X,Y,S$)
 702 IF S$="" OR NOT Label THEN ENDPROC
 704 PROCMove(X,Y):VDU5:PRINT S$:VDU4:ENDPROC
 710 :
 720 DEFPROCRotate(N%):ENDPROC
 730 :
 740 DEFPROCMark(N%)
 742 IF N%=8 THEN PLOT 0,5,0:PLOT 1,-10,0:PLOT 0,5,5
       :PLOT 1,0,-10:PLOT 0,0,5
 744 IF N%=3 THEN PLOT 0,5,5:PLOT 1,-10,0:PLOT 1,0,-10
       :PLOT 1,10,0:PLOT 1,0,10:PLOT 0,-5,-5
 746 IF N%=1 THEN PLOT 0,1,1:PLOT 1,-2,0:PLOT 1,0,-2
       :PLOT 1,2,0:PLOT 1,0,2:PLOT 0,-1,-1
 748 ENDPROC
 750 :
 760 DEFPROCSpeed(N%):ENDPROC
 770 :
 780 DEFPROCTitle
 790 VDU28,0,24,39,0
 800 CLS
 810 PRINTTAB(0,4);:VDU131,157,132,136:PRINT"     Data plotting routine"
 820 PRINTTAB(0,5);:VDU131,157,132:PROCSpace(17-LEN(Title$)/2):PRINTTitle$
 830 PRINTTAB(0,6);:VDU131,157,132:PROCSpace(17-LEN(Title2$)/2):PRINTTitle2$
 840 PROCCom(7,"")
 845 PRINTTAB(0,0);
 850 ENDPROC
 860 :
 870 DEFPROCSpacebar
 880 LOCAL A$
 890 *FX21,0
 900 PRINT''"Press the space bar to continue"
 910 PROCCom(2,"Waiting for your reply")
 920 REPEAT:A$=GET$:UNTIL A$=" "
 930 PRINT "Thank you"''
 940 PROCCom(2,"")
 950 ENDPROC
 960 :
1240 :
1250 DEFPROCCom(N,C$)
1260 L%=LEN(C$):X=POS:Y=VPOS:X%=19-L%/2
1270 VDU28,0,24,39,0
1280 PRINT TAB(0,N);" ";:PROCSpace(X%)
1290 PRINTC$;:PROCSpace(38-X%-L%)
1300 VDU28,0,24,39,9
1310 PRINTTAB(X,Y)"";
1320 ENDPROC
1330 :
1340 DEFPROCSpace(N)
1350 LOCAL N1
1360 N1=1:REPEAT:PRINT" ";:N1=N1+1:UNTIL N1>N
1370 ENDPROC
1380 :
1390 DEFPROCOpen
1400 IF Fileopen THEN ENDPROC
1410 REPEAT
1420   INPUT "Which file is data stored in"'
                    "(type * for catalogue)"'';File$
1460   IF File$="*" THEN *CAT 0
```

```
1470    IF File$="*" THEN *CAT 2
1480    UNTIL File$<>"*"
1490  File=OPENIN(File$)
1500  Fileopen=TRUE
1510  INPUT#File,Title$,Title2$
1520  PROCTitle
1530  PROCCom(7,"From data file: "+File$)
1540  INPUT#File,NofYs,Xaxis$,Yaxis$
1550  J%=1400/NofYs:NofYs=NofYs-1
1560  DIM Comment$(15),Y$(NofYs),X(J%),Y(NofYs,J%)
1570  FOR I=0 TO NofYs
1580    INPUT#File,Y$(I):NEXT
1590  NofCs=0:Comment$(0)=""
1600  REPEAT:PROCTest
1610    IF String THEN NofCs=NofCs+1:INPUT#File,Comment$(NofCs)
1620    UNTIL NOT String
1630  FORI=0 TO NofCs
1640    PRINT Comment$(I):NEXT
1650  ENDPROC
1660  :
1670  DEFPROCTest
1680  REM Tests next bit of data
1690  Integer=FALSE:Real=FALSE:String=FALSE
1700  char=BGET#File
1710  PTR#File=PTR#File-1
1720  IF char=&40 THEN Integer=TRUE
1730  IF char=&FF THEN Real=TRUE
1740  IF char=&00 THEN String=TRUE
1750  ENDPROC
1760  :
1770  DEFFNLoad
1780  IF NOT EOF#File AND (Integer OR Real) :INPUT#File,Var:=Var
1790  =0
1800  :
1810  DEFPROCClose
1820  IF Fileopen=FALSE THEN ENDPROC
1830  CLOSE#File
1840  *DRIVE 0
1850  ENDPROC
1860  :
1870  DEFPROCOptions
1880  PROCOpen:PROCSpacebar
1890  DIM Mark(NofYs),Line(NofYs),Not(NofYs)
1910  FOR I=0 TO NofYs
1920    CLS:PROCCom(1,"Line Option"):PRINT''"For Y(";I+1;") :";Y$(I);''
          "Do you want:"''"  M - a mark only (dot,cross etc.)"'
          "  J - line joining points only"'"  C - marks and joining line"'
          "  N - nothing"
1940    Mark(I)=-1:Line(I)=0:Not(I)=FALSE
1950    REPEAT:A$=GET$
1960      IF A$="C" OR A$="c" OR A$="J" OR A$="j" THEN Line(I)=2
1980      IF A$="C" OR A$="c" OR A$="M" OR A$="m" THEN PRINT''"Choose from"'
            " 1 - dot"'" 3 - square"'" 8 - cross":INPUT Mark(I)
1990      IF A$="N" OR A$="n" Not(I)=TRUE
2010      IF (Line(I) < 1) AND (Mark(I) < 1) AND (NOT(Not(I))) VDU 7
2020      UNTIL (Line(I)>0) OR (Mark(I)>0) OR Not(I)
2030    NEXT
2040  DIM Lines%(23):FOR I=0 TO 23:Lines%(I)=0:NEXT
```

```
2050 W%=1:T%=0:Tall=FALSE:Whole=TRUE:Wide=TRUE:Zero=FALSE:Quick=TRUE
     :Axis=TRUE:Label=TRUE:Xscale=FALSE:Yscale=FALSE:Pen=FALSE
2060 CLS:PROCCom(1,"Format Options"):PRINT"   T - Tall plot"'
     "   W - Wide plot"'"   P - Whole page plot"'"   R - room for comments"
2070   PRINT "   Z - axis along zeros"'"   B - axis on bottom and left"'
     "   N - no axis or label drawn"'"   A - axis only (no labels)"
2080 PRINT "   M - min and max used for both axis"'
     "   X - X axis scale fixed"'"   Y - Y axis scale fixed"
2090 PRINT "   O - one pen plot"'"   C - coloured plot"'
     "   Q - quick plot"'"   G - good plot";
2100 PROCCom(2,"* marks option chosen"):PROCCom(3,"-press RET to continue")
2110 REPEAT
2120   IF Tall Lines%(8)=1:Lines%(9)=0
2130   IF Wide Lines%(9)=1:Lines%(8)=0
2140   IF NOT Whole Lines%(11)=1:Lines%(10)=0
2150   IF Whole Lines%(10)=1:Lines%(11)=0
2160   IF Zero Lines%(12)=1:Lines%(13)=0 ELSE Lines%(13)=1:Lines%(12)=0
2165   IF Label Lines%(15)=0 ELSE Lines%(15)=1
2170   IF NOT Axis Lines%(14)=1:Lines%(12)=0:Lines%(13)=0::Lines%(15)=0
       ELSE Lines%(14)=0
2180   Lines%(16)=1:Lines%(18)=0:Lines%(17)=0
2190   IF Yscale Lines%(18)=1:Lines%(16)=0
2200   IF Xscale Lines%(17)=1:Lines%(16)=0
2205   IF Pen Lines%(19)=1:Lines%(20)=0 ELSE Lines%(19)=0:Lines%(20)=1
2210   IF Quick Lines%(21)=1:Lines%(22)=0 ELSE Lines%(21)=0:Lines%(22)=1
2220   FOR L=0 TO 15:IF Lines%(L+8)=1 THEN PRINTTAB(0,L);"*";
         ELSE PRINTTAB(0,L);" ";
2230     NEXT
2240   A$=GET$
2250   IF A$="W" OR A$="w" THEN Tall=FALSE:Wide=TRUE:W%=1:T%=0
2260   REM IF A$="T" OR A$="t" Tall=TRUE:Wide=FALSE:W%=0:T%=1
2270   IF A$="P" OR A$="p" THEN Whole=TRUE
2280   REM IF A$="R" OR A$="r" THEN Whole=FALSE
2290   IF A$="Q" OR A$="q" THEN Quick=TRUE
2300   IF A$="G" OR A$="g" THEN Quick=FALSE
2305   IF A$="O" OR A$="o" THEN Pen=TRUE
2307   IF A$="C" OR A$="c" THEN Pen=FALSE
2310   IF A$="M" OR A$="m" THEN Yscale=FALSE:Xscale=FALSE
2320   IF A$="X" OR A$="x" THEN Xscale=TRUE
2330   IF A$="Y" OR A$="y" THEN Yscale=TRUE
2340   IF A$="Z" OR A$="z" THEN Zero=TRUE:Axis=TRUE:Label=TRUE
2350   IF A$="B" OR A$="b" THEN Zero=FALSE:Axis=TRUE:Label=TRUE
2360   IF A$="N" OR A$="n" THEN Axis=FALSE:Label=FALSE
2365   IF A$="A" OR A$="a" THEN Axis=TRUE:Label=FALSE
2370   UNTIL ASC(A$)=13
2380 ENDPROC
2383 :
2390 DEFPROCProgram
2420 IF Quick THEN PROCSpeed(9) ELSE PROCSpeed(2)
2430 PROCCharactersize(3)
2440 PROCPen(3)
2450 PROCRotate(3*T%)
2460 PROCPrint(2400*T%,1765+50*T%,Title$)
2470 IF Whole PROCPrint(2350*T%,1700+100*T%,Title2$)
2480 FOR I=0 TO NofCs
2490   IF NOT Whole AND Axis PROCPrint((400-50*I)*T%+1800,900+(100-50*I)*W%
                 ,Comment$(I))
2500   NEXT
```

```
2510 PROCPen(2):PROCCharactersize(5)
2520 IF NOT Whole PROCPrint(2260*T%,1650+50*T%,Title2$)
2530 PROCCharactersize(3)
2540 FOR I=0 TO 1:Max(I)=-9E30:Min(I)=9E30:NEXT
2550 I%=0
2560 REPEAT
2570   I%=I%+1
2580   X(I%)=FNLoad
2590   IF X(I%)<Min(0) Min(0)=X(I%)
2600   IF X(I%)>Max(0) Max(0)=X(I%)
2610   FOR J%=0 TO NofYs:Y(J%,I%)=FNLoad
2620     IF (NOT Not(J%)) AND Y(J%,I%)<Min(1) Min(1)=Y(J%,I%)
2630     IF (NOT Not(J%)) AND Y(J%,I%)>Max(1) Max(1)=Y(J%,I%)
2640     NEXT
2650   UNTIL EOF#File
2660 IF Xscale THEN CLS:PRINT'"Minimum X is ";Min(0)'"Maximum X is ";Max(0)
       :INPUT "Scale to go from";Min(0)'"                    to";Max(0)
2670 IF Yscale THEN CLS:PRINT'"Minimum Y is ";Min(1)'"Maximum Y is ";Max(1)
       :INPUT "Scale to go from";Min(1)'"                    to";Max(1)
2680 NofPs=I%
2690 FOR I=0 TO 1
2700   Diff(I)=ABS(Max(I)-Min(I))
2710   IF Diff(I)=0 THEN PRINT "Silly data - can't plot it":ENDPROC
2720   Fact(I)=1
2730   REPEAT
2740     IF Diff(I)>15 THEN Diff(I)=Diff(I)/10:Fact(I)=Fact(I)*10
2750     UNTIL Diff(I)<15
2760   REPEAT
2770     IF Diff(I)<1.5 THEN Diff(I)=Diff(I)*10:Fact(I)=Fact(I)/10
2780     UNTIL Diff(I)>1.5
2790   Unit(I)=0.1
2800   IF Diff(I)>2 THEN Unit(I)=0.2
2810   IF Diff(I)>5 THEN Unit(I)=0.5
2820   Unit(I)=Unit(I)*Fact(I)
2830   Min(I)=(INT(Min(I)/Unit(I)))*Unit(I)
2840   Max(I)=(INT(Max(I)/Unit(I)+1))*Unit(I)
2850   Diff(I)=Max(I)-Min(I)
2860   Num(I)=INT(Diff(I)/Unit(I)+0.1)
2870   IF I=0 OR NOT Whole Fact(I)=INT(1500/Num(I))
                  ELSE Fact(I)=INT(2100/Num(I))
2880   IF I=0 AND Whole AND Wide Fact(I)=INT(2300/Num(I))
2890   IF I=1 AND Wide AND Whole Fact(I)=INT(1400/Num(I))
2900   IF I=1 AND Wide AND NOT Whole Fact(I)=INT(1350/Num(I))
2910   Scale(I)=Fact(I)*Num(I)/Diff(I)
2920   NEXT
2930 PROCPen(1)
2940 YZero=150
2950 IF Zero THEN YZero=150-Min(1)*Scale(1)
2960 IF Tall AND Axis PROCMove(YZero,1650):PROCYaxis(-Fact(0),Num(0))
           ELSE IF Axis PROCMove(150,YZero):PROCXaxis(Fact(0),Num(0))
2970 FOR U%=0 TO Num(0)
2980   String=Unit(0)*INT(0.1+(Min(0)+U%*Diff(0)/Num(0))/Unit(0))
2990   IF (Min(0)/Unit(0)+U%+500.1) MOD 5 = 0 THEN PROCXlabel
3000   NEXT
3010 IF Whole PROCPrint(YZero*T%-130+1630*W%,YZero*W%-140+840*T%,Xaxis$)
3020 IF NOT Whole PROCPrint(YZero*T%-130+1230*W%,YZero*W%-140+840*T%,Xaxis$)
3030 PROCRotate(W%)
3040 XZero=150+1500*T%
```

Listing of SKETCH.BBC at 00:18:33 on MAY 13, 1987 for CCid=PHP7

```
3050 IF Zero THEN XZero=(1650+Min(0)*Scale(0))*T%+(150-Min(0)*Scale(0))*W%
3060 IF Axis AND Tall PROCMove(150,XZero):PROCXaxis(Fact(1),Num(1))
       ELSE IF Axis PROCMove(XZero,150):PROCYaxis(Fact(1),Num(1))
3070 FOR U%=0 TO Num(1)
3080   String=Unit(1)*INT(0.1+(Min(1)+U%*Diff(1)/Num(1))/Unit(1))
3090   IF  (Min(1)/Unit(1)+U%+500.1) MOD 5 = 0 THEN PROCYlabel
3100   NEXT
3110 IF Whole PROCPrint(1400*T%+(XZero-110)*W%,900*W%+(XZero+100)*T%,Yaxis$)
3120 IF NOT Whole PROCPrint(1000*T%+(XZero-110)*W%,900*W%+(XZero+100)*T%,
                                                        Yaxis$)
3130 PROCRotate(3*T%):Pens=1
3140 FOR I=0 TO NofYs
3150   IF NOT Not(I) Pens=1+(Pens MOD 3):PROCPen(Pens)
3160   IF NOT Not(I) AND Whole AND Tall PROCPrint(2400-I*50,800,Y$(I))
3170   IF NOT Not(I) AND NOT Whole AND Tall PROCPrint(1800+NofYs*50-I*50,
                                                        1700,Y$(I))
3180   IF Wide AND NOT Not(I) PROCPrint(1500,1750-65*I,Y$(I))
3190   PROCCalc(I)
3200   IF Mark(I)>0 PROCMarks(I)
3220   IF Line(I)=2 PROCJoin(I)
3230   IF Line(I)=1 PROCLine(I)
3240   NEXT
3250 ENDPROC
3260 :
3270 DEFPROCCalc(I)
3280 FOR J%=1TONofPs
3283   IF Y(I,J%)>Max(1) THEN Y(I,J%)=Max(1)
3284   IF Y(I,J%)<Min(1) THEN Y(I,J%)=Min(1)
3286   IF X(J%)>Max(0) AND I=0 THEN X(J%)=Max(0)
3287   IF X(J%)<Min(0) AND I=0 THEN X(J%)=Min(0)
3290   IF Tall XI=X(J%):X(J%)=(((Y(I,J%)-Min(1)))*Scale(1)+150)
3300   IF Tall AND  I=0 THEN Y(I,J%)=1650-((XI-Min(0))*Scale(0))
3310   IF Tall AND I<>0 THEN Y(I,J%)=Y(I-1,J%)
3320   IF Wide THEN Y(I,J%)=150+((Y(I,J%)-Min(1))*Scale(1))
3330   IF Wide AND  I=0 THEN X(J%)=150+((X(J%)-Min(0))*Scale(0))
3340   NEXT
3350 ENDPROC
3360 :
3370 DEFPROCMarks(I)
3380 FOR J%=1TO I%
3390   PROCMove(X(J%),Y(I,J%)):PROCMark(Mark(I))
3400   NEXT
3410 ENDPROC
3420 :
3430 DEFPROCLine(I)
3440 PROCCurve(1,I%,1,1)
3450 ENDPROC
3455 :
3510 :
3520 DEFPROCJoin(I)
3530 PROCMove(X(1),Y(I,1))
3540 FOR J%=2TO I%
3550   IF ABS((X(J%)-X(J%-1)))<1200 AND ABS((Y(I,J%)-Y(I,J%-1)))<1200
         THEN PROCDraw(X(J%),Y(I,J%)) ELSE PROCMove(X(J%),Y(I,J%))
3560   NEXT
3570 PROCMove(0,0)
3580 ENDPROC
3590 :
```

Listing of SKETCH.BBC at 00:18:33 on MAY 13, 1987 for CCid=PHP7

```
3600 DEFPROCXlabel
3610 IF Tall PROCMove(YZero,1650-U%*Fact(0)):PROCDraw(YZero-40,1650-U%*
       Fact(0)):PROCPrint(YZero-78,1690-U%*Fact(0),FNRound(STR$(String),2))
3620 IF Wide PROCMove(150+U%*Fact(0),YZero):PROCDraw(150+U%*Fact(0),
       YZero-40):PROCPrint(110+U%*Fact(0),YZero-90,FNRound(STR$(String),2))
3630 ENDPROC
3640 :
3650 DEFPROCYlabel
3660 IF Tall THEN PROCMove(150+U%*Fact(1),XZero)
             :PROCDraw(150+U%*Fact(1),XZero+37)
             :PROCPrint(120+U%*Fact(1),XZero+48,FNRound(STR$(String),2))
3670 IF Wide THEN PROCMove(XZero,150+U%*Fact(1))
             :PROCDraw(XZero-37,150+U%*Fact(1))
             :PROCPrint(XZero-50,120+U%*Fact(1),FNRound(STR$(String),2))
3680 ENDPROC
4000 :
4010 DEFFNRound(S$,N)
4020 LOCAL A$,ch$,Count
4030 FOR I=1 TO LEN(S$)
4040       ch$=MID$(S$,I,1)
4050      IF ch$="." THEN Count=1
4060      IF ch$="E" THEN Count=0:A$=FNTrail(A$)
4070      IF Count>0 THEN Count=Count+1
4080      IF Count<N+3 THEN A$=A$+ch$
4090      NEXT
4100  =FNTrail(A$)
4110 :
4120 DEFFNTrail(A$)
4130 LOCAL Carry,B$,I,L
4140 Carry=FALSE:Point=FALSE:B$=A$
4150 IF RIGHT$(A$,1)="9" THEN Carry=TRUE
4160 REPEAT
4170      IF RIGHT$(A$,1)="9" AND Carry THEN A$=LEFT$(A$,LEN(A$)-1)
4180      IF RIGHT$(A$,1)="0" AND NOT Carry THEN A$=LEFT$(A$,LEN(A$)-1)
4190      IF RIGHT$(A$,1)="." THEN A$=LEFT$(A$,LEN(A$)-1):Point=TRUE
4200      UNTIL Point OR ((RIGHT$(A$,1)<>"0" OR Carry) AND
                RIGHT$(A$,1)<>"." AND (RIGHT$(A$,1)<>"9" OR NOT Carry))
4210 IF Point=FALSE A$=B$:Carry=FALSE
4220 L=LEN(A$)
4230 FOR I=LEN(A$) TO 1 STEP -1
4240   IF MID$(A$,I,1)="9" AND Carry
       THEN A$=LEFT$(A$,I-1)+"0"+RIGHT$(A$,L-I)
       ELSE IF Carry
          THEN A$=LEFT$(A$,I-1)+CHR$(ASC(MID$(A$,I,1))+1)+RIGHT$(A$,L-I)
                        :Carry=FALSE
4250     NEXT
4260 IF Carry THEN A$="1"+A$
4270 :=A$
```

Listing of HP.BBC at 00:18:20 on MAY 13, 1987 for CCid=PHP7

```
 10 REM   HP    plotting   program
 20 REM   --
 30 REM  M.Hawton (Durham Dec 1986)
 40 REM
 50 *KEY 0 *COPY 1 0
 52 *KEY 1 *COPY 0 3A
 54 *KEY 3 *KEY 2 V.
 56 *KEY 9 RUN|M
 70 VDU3,15:MODE 7
 80  Split180=FALSE
 90 PROCSetup:PROCProgram
110 Finished=TRUE
120 IF Pixy PROCPen(0):PROCPen(0)
130 IF Pixy PROCMove(0,0):PROCPen(0)
140 *FX2,0
150 *FX3,0
160 PROCClose
170 IF NOT Finished THEN REPORT:PRINT;" at line ";ERL
180 PROCCom(1,"Thank you for using this plotter")
190 PROCCom(2,"I now await your next command")
200 VDU28,0,24,39,0
210 END
220 :
230 DEFPROCSetup
240 VDU28,0,24,39,9
250 Fileopen=FALSE:Title$="":Title2$=""
260 Finished=FALSE:Pixy=FALSE
270 ON ERROR GOTO 120
280 PROCTitle
290 *FX7,7
300 *FX8,7
310 *FX5,2
320 *FX2,2
330 PROCHP
340 *FX2,2
350 DIM Max(1),Min(1),Scale(1),Unit(1),Num(1),Diff(1),Fact(1)
360 ENDPROC
370 :
380 DEFPROCPen(I%):IF Pen THEN I%=SGN(I%)
385 PROCSend("SP"+STR$(I%)):ENDPROC
390 :
400 DEFPROCMove(I,J):PROCSend("PA,PU,"+STR$(I*4)+","+STR$(J*4)):ENDPROC
410 :
420 DEFPROCDraw(I,J):PROCSend("PA,PD,"+STR$(I*4)+","+STR$(J*4)):ENDPROC
430 :
440 DEFPROCCharactersize(I%):PROCSend("SI"+STR$(0.04+I%*0.04)+","
                +STR$(0.07+I%*0.07)):ENDPROC
450 :
460 DEFPROCXaxis(Length,Number%)
462 IF Tall PROCSend("TL1.0,0") ELSE PROCSend("TL-1.0,0")
464 FOR TICK=1 TO Number%:PROCSend("PR,PD,"+STR$(Length*4)+",0")
              :PROCSend("XT"):NEXT
465 PROCSend("PR,PU"):ENDPROC
470 :
480 DEFPROCYaxis(Length,Number%)
482 PROCSend("TL-1.0,0")
485 FOR TICK=1 TO Number%:PROCSend("PR,PD,0,"+STR$(Length*4))
              :PROCSend("YT"):NEXT
```

```
487 PROCSend("PR,PU"):ENDPROC
490 :
500 DEFPROCCurve(N1%,N2,Xscale,Yscale)
510 LOCAL I%
520 REM Draws curve through X(N1%),Y(N1%)...to X(N2),Y(N2)
530 REM IF N2<0 THEN curve closed
540 N2%=ABS(N2):IF (N2%-N1%)<3 THEN ENDPROC
550 VDU2
560 PRINT "PA,PU";
570 I%=N1%-1
580 REPEAT
590   I%=I%+1
600   IF ABS((X(I%)-X(I%-1))*Xscale)>600 THEN OK=FALSE
610   IF ABS((Y(I,I%)-Y(I,I%-1))*Yscale)>600 THEN OK=FALSE
620    IF I%=N1% THEN OK=TRUE
630   IF OK THEN PRINT;",";4*Xscale*X(I%);",";4*Yscale*Y(I,I%);
635   IF I%=N1% THEN PRINT ",PD";
640   UNTIL (I%>=N2%) OR NOT OK
650 PRINT",PU"
660 VDU3
670 IF NOT OK THEN PROCCurve(I%,N2%,Xscale,Yscale)
680 ENDPROC
690 :
700 DEFPROCPrint(X,Y,S$)
702 IF S$="" OR NOT Label THEN ENDPROC
704 PROCMove(X,Y):PROCSend("LB"+S$+CHR$(1)+CHR$(3)):ENDPROC
710 :
720 DEFPROCRotate(N%):LOCAL sign
722 sign=1:IF N%>1 sign=-1
723 PROCSend("DI"+STR$(sign*((1+N%)MOD2))+","+STR$(sign*(N%MOD2)))
725 ENDPROC
730 :
740 DEFPROCMark(N%)
741 IF N%=1 THEN PROCSend("PR,PD,1,1,PU,-1,-1")
742 IF N%=2 THEN PROCSend("PR,PU,30,0,PD,-30,30,-30,-30,30,
                          -30,30,30,PU,-30,0")
743 IF N%=3 THEN PROCSend("PR,PU,30,30,PD,-60,0,0,-60,60,0,0,60,pu,-30,-30")
744 IF N%=4 THEN PROCSend("PR,PU,0,30,pd,-25,-45,50,0,-25,45,pu,0,-30")
745 IF N%=7 THEN PROCSend("PR,PD,25,25,-50,-50,25,25,25,-25,-50,50,25,-25")
746 IF N%=14 OR N=15 THEN PROCMark(7):PROCMark(8)
748 IF N%=8 THEN PROCSend("PR,PD,30,0,-60,0,30,0,0,30,0,-60,0,30,PU")
749 ENDPROC
750 :
760 DEFPROCSpeed(N%):PROCSend("VS"+STR$(N%*4)):ENDPROC
770 :
780 DEFPROCTitle
790 VDU28,0,24,39,0
800 CLS
810 PRINTTAB(0,4);:VDU131,157,132,136:PRINT"      HP7470 plotting  routine"
820 PRINTTAB(0,5);:VDU131,157,132:PROCSpace(17-LEN(Title$)/2):PRINTTitle$
830 PRINTTAB(0,6);:VDU131,157,132:PROCSpace(17-LEN(Title2$)/2):PRINTTitle2$
840 PROCCom(7,"")
845 PRINTTAB(0,0);
850 ENDPROC
860 :
870 DEFPROCSpacebar
880 LOCAL A$
890 *FX21,0
```

Listing of HP.BBC at 00:18:20 on MAY 13, 1987 for CCid=PHP7

```
 900 PRINT'"Press the space bar to continue"
 910 PROCCom(2,"Waiting for your reply")
 920 REPEAT:A$=GET$:UNTIL A$=" "
 930 PRINT "Thank you"''
 940 PROCCom(2,"")
 950 ENDPROC
 960 :
 970 DEFPROCSend(S$)
 980 VDU2
 990 PRINTS$
1000 VDU3
1010 ENDPROC
1020 :
1030 DEFPROCHP
1050 *FX6,10
1052 PRINT '"Are you using Clearway? (Y/N)"'
1053 A$=GET$
1060 IF A$="Y" OR A$="y" THEN PROCReset("31-") ELSE *FX8,7
1070 ENDPROC
1080 :
1090 DEFPROCPrinter
1110 *FX6,0
1120 PROCReset("20-")
1130 ENDPROC
1140 :
1150 DEFPROCReset(S$)
1160 VDU2,21,1,2
1170 PRINT;STRING$(64,CHR$(13));"N!!!!!!!!!!!!";S$
1180 PRINT;"Y";
1190 VDU6,3
1200 *FX2,1
1210 REPEAT:A$=INKEY$(2):UNTIL A$=""
1220 *FX2,0
1230 ENDPROC
1240 :
1250 DEFPROCCom(N,C$)
1260 L%=LEN(C$):X=POS:Y=VPOS:X%=19-L%/2
1270 VDU28,0,24,39,0
1280 PRINT TAB(0,N);" ";:PROCSpace(X%)
1290 PRINTC$;:PROCSpace(38-X%-L%)
1300 VDU28,0,24,39,9
1310 PRINTTAB(X,Y)"";
1320 ENDPROC
1330 :
1340 DEFPROCSpace(N)
1350 LOCAL N1
1360 N1=1:REPEAT:PRINT" ";:N1=N1+1:UNTIL N1>N
1370 ENDPROC
1380 :
1390 DEFPROCOpen
1400 IF Fileopen THEN ENDPROC
1410 REPEAT
1420   INPUT "Which file is data stored in"'"(type * for catalogue)"'';File$
1460   IF File$="*" THEN *CAT 0
1470   IF File$="*" THEN *CAT 2
1480  UNTIL File$<>"*"
1490 File=OPENIN(File$)
1500 Fileopen=TRUE
```

Listing of HP.BBC at 00:18:20 on MAY 13, 1987 for CCid=PHP7

```
1510 INPUT#File,Title$,Title2$
1520 PROCTitle
1530 PROCCom(7,"From data file: "+File$)
1540 INPUT#File,NofYs,Xaxis$,Yaxis$
1550 J%=1600/NofYs:NofYs=NofYs-1
1560 DIM Comment$(20),Y$(NofYs),X(J%),Y(NofYs,J%)
1570 FOR I=0 TO NofYs
1580   INPUT#File,Y$(I):NEXT
1590 NofCs=0:Comment$(0)=""
1600 REPEAT:PROCTest
1610   IF String THEN NofCs=NofCs+1:INPUT#File,Comment$(NofCs)
1620   UNTIL NOT String
1630 FORI=0 TO NofCs
1640   PRINT Comment$(I):NEXT
1650 ENDPROC
1660 :
1670 DEFPROCTest
1680 REM Tests next bit of data
1690 Integer=FALSE:Real=FALSE:String=FALSE
1700 char=BGET#File
1710 PTR#File=PTR#File-1
1720 IF char=&40 THEN Integer=TRUE
1730 IF char=&FF THEN Real=TRUE
1740 IF char=&00 THEN String=TRUE
1750 ENDPROC
1760 :
1770 DEFFNLoad
1780 IF NOT EOF#File AND (Integer OR Real) :INPUT#File,Var:=Var
1790 =0
1800 :
1810 DEFPROCClose
1820 IF Fileopen=FALSE THEN ENDPROC
1830 CLOSE#File
1850 ENDPROC
1852
1854 DEFFNCh:LOCAL Mark
1856 PRINT''"Choose from"'" 1 - dot"'" 2 - diamond"'" 3 - square"
1857 PRINT" 4 - triangle"'" 7 - X"'" 8 - cross"'"14 - asterix"'"15 - star"
1858 INPUT Mark:=Mark
1860 :
1870 DEFPROCProgram
1880 PROCOpen:PROCSpacebar
1890 DIM Mark(NofYs),Line(NofYs),Not(NofYs)
1910 FOR I=0 TO NofYs
1920   CLS:PROCCom(1,"Line Option"):PRINT''"For Y(";I+1;") :";Y$(I)
1925   PRINT''"Do you want:"''"  M - a mark only"
1927   PRINT"  J - line joining points only"'"  C - marks and joining line"
1935   PRINT"  N - nothing"
1940   Mark(I)=-1:Line(I)=0:Not(I)=FALSE
1950   REPEAT:A$=GET$
1960     IF A$="C" OR A$="c" OR A$="J" OR A$="j" THEN Line(I)=2
1970     IF A$="B" OR A$="b" OR A$="L" OR A$="l" THEN Line(I)=1
1980     IF A$="B"OR A$="b"OR A$="C"OR A$="c"OR A$="M"OR A$="m"Mark(I)=FNCh
1990     IF A$="N" OR A$="n" Not(I)=TRUE
2010     IF (Line(I) < 1) AND (Mark(I) < 1) AND (NOT(Not(I))) VDU 7
2020     UNTIL (Line(I)>0) OR (Mark(I)>0) OR Not(I)
2030   NEXT
2040 DIM Lines%(23):FOR I=0 TO 23:Lines%(I)=0:NEXT
```

Listing of HP.BBC at 00:18:20 on MAY 13, 1987 for CCid=PHP7

```
2050 T%=1:W%=0:Tall=TRUE:Whole=FALSE:Wide=FALSE:Zero=FALSE:Quick=TRUE
2055 Axis=TRUE:Label=TRUE:Xscale=FALSE:Yscale=FALSE:Pen=FALSE
2060 CLS:PROCCom(1,"Format Options"):PRINT"     T - Tall plot"
2065 PRINT"     W - Wide plot""     P - Whole page plot"
2067 PRINT"     R - room for comments""     Z - axis along zeros"
2070 PRINT"     B - axis on bottom and left""     N - no axis or label drawn"
2075 PRINT"     A - axis only (no labels)"
2080 PRINT"     M - min and max used for both axis"
2085 PRINT"     X - X axis scale fixed""     Y - Y axis scale fixed"
2090 PRINT"     O - one pen plot""     C - coloured plot""     Q - quick plot"
2095 PRINT"     G - good plot";
2100 PROCCom(2,"* marks option chosen"):PROCCom(3,"-press RET to continue")
2110 REPEAT
2120   IF Tall Lines%(8)=1:Lines%(9)=0
2130   IF Wide Lines%(9)=1:Lines%(8)=0
2140   IF NOT Whole Lines%(11)=1:Lines%(10)=0
2150   IF Whole Lines%(10)=1:Lines%(11)=0
2160   IF Zero Lines%(12)=1:Lines%(13)=0 ELSE Lines%(13)=1:Lines%(12)=0
2165   IF Label Lines%(15)=0 ELSE Lines%(15)=1
2170   IF NOT Axis Lines%(14)=1:Lines%(12)=0:Lines%(13)=0::Lines%(15)=0
2175   IF Axis Lines%(14)=0
2180   Lines%(16)=0:Lines%(18)=0:Lines%(17)=0
2190   IF Yscale Lines%(18)=1:Lines%(16)=0
2200   IF Xscale Lines%(17)=1:Lines%(16)=0
2205   IF Pen Lines%(19)=1:Lines%(20)=0 ELSE Lines%(19)=0:Lines%(20)=1
2210   IF Quick Lines%(21)=1:Lines%(22)=0 ELSE Lines%(21)=0:Lines%(22)=1
2220   FOR L=0 TO 15
2230     IF Lines%(L+8)=1 PRINTTAB(0,L);"*"; ELSEPRINTTAB(0,L);" ";
2240     NEXT:A$=GET$
2250   IF A$="W" OR A$="w" THEN Tall=FALSE:Wide=TRUE:W%=1:T%=0
2260   IF A$="T" OR A$="t" Tall=TRUE:Wide=FALSE:W%=0:T%=1
2270   IF A$="P" OR A$="p" THEN Whole=TRUE
2280   IF A$="R" OR A$="r" THEN Whole=FALSE
2290   IF A$="Q" OR A$="q" THEN Quick=TRUE
2300   IF A$="G" OR A$="g" THEN Quick=FALSE
2305   IF A$="O" OR A$="o" THEN Pen=TRUE
2307   IF A$="C" OR A$="c" THEN Pen=FALSE
2310   IF A$="M" OR A$="m" THEN Yscale=FALSE:Xscale=FALSE
2320   IF A$="X" OR A$="x" THEN Xscale=TRUE
2330   IF A$="Y" OR A$="y" THEN Yscale=TRUE
2340   IF A$="Z" OR A$="z" THEN Zero=TRUE:Axis=TRUE:Label=TRUE
2350   IF A$="B" OR A$="b" THEN Zero=FALSE:Axis=TRUE:Label=TRUE
2360   IF A$="N" OR A$="n" THEN Axis=FALSE:Label=FALSE
2365   IF A$="A" OR A$="a" THEN Axis=TRUE:Label=FALSE
2370   UNTIL ASC(A$)=13
2380 PROCCom(1,""):PROCCom(2,""):PROCCom(3,"")
2390 CLS:PRINT'"Please check HP plotter is ready":PROCSpacebar
2400 Pixy=TRUE
2410 PROCCom(1,"Plotting on HP 7470 plotter")
2420 IF Quick THEN PROCSpeed(9) ELSE PROCSpeed(2)
2430 PROCCharactersize(3)
2440 PROCPen(3)
2450 PROCRotate(3*T%)
2460 PROCPrint(2400*T%,1750+50*T%,Title$)
2470 IF Whole PROCPrint(2350*T%,1700+100*T%,Title2$)
2480 FOR I=0 TO NofCs
2490   Pr=FALSE:IF NOT Whole AND Axis Pr=TRUE
2495   IF Pr PROCPrint((400-50*I)*T%+1800,900+(100-50*I)*W%,Comment$(I))
```

Listing of HP.BBC at 00:18:20 on MAY 13, 1987 for CCid=PHP7

```
2500    NEXT
2510 PROCPen(2):PROCCharactersize(5)
2520 IF NOT Whole PROCPrint(2260*T%,1650+50*T%,Title2$)
2530 PROCCharactersize(3)
2540 FOR I=0 TO 1:Max(I)=-9E30:Min(I)=9E30:NEXT
2550 I%=0
2560 REPEAT
2570    I%=I%+1
2580    X(I%)=FNLoad
2585    IF Split180 X(I%)=(((X(I%)+180)*1000) MOD 360000)/1000 -180
2590    IF X(I%)<Min(0) Min(0)=X(I%)
2600    IF X(I%)>Max(0) Max(0)=X(I%)
2610    FOR J%=0 TO NofYs:Y(J%,I%)=FNLoad
2620      IF (NOT Not(J%)) AND Y(J%,I%)<Min(1) Min(1)=Y(J%,I%)
2630      IF (NOT Not(J%)) AND Y(J%,I%)>Max(1) Max(1)=Y(J%,I%)
2640      NEXT
2650    UNTIL EOF#File
2660 IF Xscale CLS:PRINT'"Minimum X is ";Min(0)'"Maximum X is ";Max(0)
2665 IF Xscale INPUT "Scale to go from";Min(0)'"                to";Max(0)
2670 IF Yscale CLS:PRINT'"Minimum Y is ";Min(1)'"Maximum Y is ";Max(1)
2675 IF Yscale INPUT "Scale to go from";Min(1)'"                to";Max(1)
2680 NofPs=I%
2690 FOR I=0 TO 1
2700    Diff(I)=ABS(Max(I)-Min(I))
2710    IF Diff(I)=0 THEN PRINT "Silly data - can't plot it":ENDPROC
2720    Fact(I)=1
2730    REPEAT
2740      IF Diff(I)>15 THEN Diff(I)=Diff(I)/10:Fact(I)=Fact(I)*10
2750      UNTIL Diff(I)<15
2760    REPEAT
2770      IF Diff(I)<1.5 THEN Diff(I)=Diff(I)*10:Fact(I)=Fact(I)/10
2780      UNTIL Diff(I)>1.5
2790    Unit(I)=0.1
2800    IF Diff(I)>2 THEN Unit(I)=0.2
2810    IF Diff(I)>5 THEN Unit(I)=0.5
2820    Unit(I)=Unit(I)*Fact(I)
2830    Min(I)=(INT(Min(I)/Unit(I))*Unit(I)
2840    Max(I)=(INT(Max(I)/Unit(I)+1))*Unit(I)
2850    Diff(I)=Max(I)-Min(I)
2860    Num(I)=INT(Diff(I)/Unit(I)+0.1)
2870    IF I=0ORNOTWhole Fact(I)=INT(1500/Num(I)) ELSEFact(I)=INT(2100/Num(I))
2880    IF I=0 AND Whole AND Wide Fact(I)=INT(2300/Num(I))
2890    IF I=1 AND Whole AND Wide Fact(I)=INT(1400/Num(I))
2900    IF I=1 AND Whole AND NOT Wide Fact(I)=INT(1350/Num(I))
2910    Scale(I)=Fact(I)/Num(I)
2920    NEXT
2930 PROCPen(1)
2940 YZero=150
2950 IF Zero AND Min(1)<0 THEN YZero=150-Min(1)*Scale(1)
2955 IF Tall AND Axis PROCMove(YZero,1650):PROCYaxis(-Fact(0),Num(0))
2960 IF Wide AND Axis PROCMove(150,YZero):PROCXaxis(Fact(0),Num(0))
2970 FOR U%=0 TO Num(0)
2980    String=Unit(0)*INT(0.1+Min(0)/Unit(0)+U%*Diff(0)/Num(0))/Unit(0)
2990    IF (Min(0)/Unit(0)+U%+500.1) MOD 5 = 0 THEN PROCXlabel
3000    NEXT
3010 IF Whole PROCPrint(YZero*T%-130-1030*W%,YZero*W%-140+840*T%,Xaxis$)
3020 IF NOT Whole PROCPrint(YZero*T%-130-1230*W%,YZero*W%-140+840*T%,Xaxis$)
3030 PROCRotate(W%)
```

```
3040 XZero=150+1500*T%
3050 IF Zero AND Min(0)<0 THEN XZero=(1650+Min(0)*Scale(0))*T%+(150-Min(0)*
                        Scale(0))*W%
3060 IF Tall AND Axis PROCMove(150,XZero):PROCXaxis(Fact(1),Num(1))
3065 IF Wide AND Axis PROCMove(XZero,150):PROCYaxis(Fact(1),Num(1))
3070 FOR U%=0 TO Num(1)
3080   String=Unit(1)*INT(0.1+(Min(1)+U%*Diff(1)/Num(1))/Unit(1))
3090   IF (Min(1)/Unit(1)+U%+500.1) MOD 5 = 0 THEN PROCYlabel
3100   NEXT
3110 IF Whole PROCPrint(1400*T%+(XZero-110)*W%,900*W%+(XZero+100)*T%,Yaxis$)
3120 IFNOTWhole PROCPrint(1000*T%+(XZero-110)*W%,900*W%+(XZero+100)*T%,
                          Yaxis$)
3130 PROCRotate(3*T%):Pens=1
3140 FOR I=0 TO NofYs
3150   IF NOT Not(I) Pens=1+(Pens MOD 3):PROCPen(Pens)
3160   IF NOT Not(I) AND Whole AND Tall PROCPrint(2400-I*50,800,Y$(I))
3170   IFNOTNot(I) ANDNOTWhole ANDTall
                     PROCPrint(1800+NofYs*50-I*50,1700,Y$(I))
3180   IF Wide AND NOT Not(I) PROCPrint(1500,1750-50*I,Y$(I))
3190   PROCCalc(I)
3200   IF Mark(I)>0 PROCMarks(I)
3220   IF Line(I)=2 PROCJoin(I)
3230   IF Line(I)=1 PROCLine(I)
3240   NEXT
3250 ENDPROC
3260 :
3270 DEFPROCCalc(I)
3280 FOR J%=1TONofPs
3283   IF Y(I,J%)>Max(I) THEN Y(I,J%)=Max(I)
3284   IF Y(I,J%)<Min(I) THEN Y(I,J%)=Min(I)
3286   IF X(J%)>Max(0) AND I=0 THEN X(J%)=Max(0)
3287   IF X(J%)<Min(0) AND I=0 THEN X(J%)=Min(0)
3290   IF Tall XI=X(J%):X(J%)=(((Y(I,J%)-Min(I)))*Scale(1)+150)
3300   IF Tall AND  I=0 THEN Y(I,J%)=1650-((XI-Min(0))*Scale(0))
3310   IF Tall AND I<>0 THEN Y(I,J%)=Y(I-1,J%)
3320   IF Wide THEN Y(I,J%)=150+((Y(I,J%)-Min(I))*Scale(1))
3330   IF Wide AND  I=0 THEN X(J%)=150+((X(J%)-Min(0))*Scale(0))
3340   NEXT
3350 ENDPROC
3360 :
3370 DEFPROCMarks(I)
3380 FOR J%=1TO I%
3390   PROCMove(X(J%),Y(I,J%)):PROCMark(Mark(I))
3400   NEXT
3410 ENDPROC
3420 :
3430 DEFPROCLine(I)
3440 PROCCurve(1,I%,1,1)
3450 ENDPROC
3455 :
3510 :
3520 DEFPROCJoin(I)
3530 PROCMove(X(1),Y(I,1))
3540 FOR J%=2TO I%
3545   Pr=FALSE
3550   IF ABS((X(J%)-X(J%-1)))<400 AND ABS((Y(I,J%)-Y(I,J%-1)))<400 Pr=TRUE
3556   IF Pr PROCDraw(X(J%),Y(I,J%)) ELSE PROCMove(X(J%),Y(I,J%))
3560   NEXT
```

```
3580 ENDPROC
3590 :
3600 DEFPROCXlabel:IF NOT Axis THEN ENDPROC
3610 IFTall PROCMove(YZero,1650-U%*Fact(0))
        :PROCDraw(YZero-40,1650-U%*Fact(0))
3615 IF Tall PROCPrint(YZero-78,1690-U%*Fact(0),FNRound(STR$(String),2))
3620 IF Wide PROCMove(150+U%*Fact(0),YZero)
        :PROCDraw(150+U%*Fact(0),YZero-40)
3625 IF Wide PROCPrint(110+U%*Fact(0),YZero-90,FNRound(STR$(String),2))
3630 ENDPROC
3640 :
3650 DEFPROCYlabel:IF NOT Axis THEN ENDPROC
3663 IFTall PROCMove(150+U%*Fact(1),XZero):PROCDraw(150+U%*Fact(1),XZero+37)
3665 IF Tall PROCPrint(120+U%*Fact(1),XZero+50,FNRound(STR$(String),2))
3670 IF Wide PROCPrint(XZero-50,120+U%*Fact(1),FNRound(STR$(String),2))
3680 ENDPROC
4000  :
4010 DEFFNRound(S$,N)
4020 LOCAL A$,ch$,Count
4030 FOR I=1 TO LEN(S$):ch$=MID$(S$,I,1)
4050   IF ch$="." THEN Count=1
4060   IF ch$="E" THEN Count=0:A$=FNTrail(A$)
4070   IF Count>0 THEN Count=Count+1
4080   IF Count<N+3 THEN A$=A$+ch$
4090   NEXT
4100 =FNTrail(A$)
4110 :
4120 DEFFNTrail(A$)
4130 LOCAL Carry,B$,I,L
4140 Carry=FALSE:Point=FALSE:B$=A$
4150 IF RIGHT$(A$,1)="9" THEN Carry=TRUE
4160 REPEAT
4170   IF RIGHT$(A$,1)="9" AND Carry THEN A$=LEFT$(A$,LEN(A$)-1)
4180   IF RIGHT$(A$,1)="0" AND NOT Carry THEN A$=LEFT$(A$,LEN(A$)-1)
4190   IF RIGHT$(A$,1)="." THEN A$=LEFT$(A$,LEN(A$)-1):Point=TRUE
4195   Pr=(RIGHT$(A$,1)<>"0" OR Carry) AND (RIGHT$(A$,1)<>"9" OR NOT Carry)
4200   UNTIL Point OR ( Pr AND RIGHT$(A$,1)<>"." )
4210 IF Point=FALSE A$=B$:Carry=FALSE
4220 L=LEN(A$):FOR I=LEN(A$) TO 1 STEP -1
4235   Pr=FALSE :Pr=MID$(A$,I,1)="9" AND Carry
4240   IF Pr A$=LEFT$(A$,I-1)+"0"+RIGHT$(A$,L-I)
4243   F1=FALSE:IF Pr OR Carry F1=TRUE
4245   IF F1 A$=LEFT$(A$,I-1)+CHR$(ASC(MID$(A$,I,1))+1)+RIGHT$(A$,L-I)
4247   IF F1 Carry=FALSE
4250   NEXT
4260 IF Carry THEN A$="1"+A$
4270 :=A$
```

Listing of CONV.PAS at 00:55:56 on MAY 13, 1987 for CCid=PHP7

```
 program Convert;
  { read BBC data from a file and convert it to a text file

    M.J.Hawton    - Durham 1986
                  }

type str_80 = packed array [1 .. 80] of char;
     flag   = ( f_real, f_integer, f_string, f_char, f_end );

 var   count : integer;
       byte  : char;
     filein, fileout  : text   ;
     in_int : integer;
     in_str : str_80;
     in_rea : real;
     in_cha : char;
     in_fla : flag;
     number_per_line : integer;
     points : integer;
     bytecount : integer;
     title : array[ 1 .. 2 ] of str_80;
     title_check : boolean;

 function ebcd ( ascii : char ) : char;
        var  count : integer;
             code  : integer;
             bcd   : char   ;
    begin
        code := ord(ascii);
        if code < 32  then ebcd:= '?' else
        if code = 32  then ebcd:= ' ' else
        if code = 33  then ebcd:= '!' else
        if code = 34  then ebcd:= '"' else
        if code = 35  then ebcd:= '#' else
        if code = 36  then ebcd:= '$' else
        if code = 37  then ebcd:= '%' else
        if code = 38  then ebcd:= '&' else
        if code = 39  then ebcd:= ' ' else
        if code = 40  then ebcd:= '(' else
        if code = 41  then ebcd:= ')' else
        if code = 42  then ebcd:= '*' else
        if code = 43  then ebcd:= '+' else
        if code = 44  then ebcd:= ',' else
        if code = 45  then ebcd:= '-' else
        if code = 46  then ebcd:= '.' else
        if code = 47  then ebcd:= '/' else
        if code <58   then begin
           bcd := '0' ;
           count := 48 ;
           repeat
              if code > count then
                    begin
                    count := succ(count);
                    bcd := succ(bcd)
                    end;
           until code=count;
           ebcd := bcd    end          else
        if code = 58  then ebcd:= ':' else
```

```
        if code = 59  then ebcd:= ';' else
        if code = 60  then ebcd:= '<' else
        if code = 61  then ebcd:= '=' else
        if code = 62  then ebcd:= '>' else
        if code = 63  then ebcd:= '?' else
        if code = 64  then ebcd:= '@' else
        if code <74   then begin
           bcd := 'A' ;
           count := 65 ;
           repeat
              if code > count then
                    begin
                    count := succ(count);
                    bcd := succ(bcd)
                    end;
           until code=count;
           ebcd := bcd        end       else
        if code <83   then begin
           bcd := 'J' ;
           count := 74 ;
           repeat
              if code > count then
                    begin
                    count := succ(count);
                    bcd := succ(bcd)
                    end;
           until code=count;
           ebcd := bcd        end       else
        if code <91   then begin
           bcd := 'S' ;
           count := 83 ;
           repeat
              if code > count then
                    begin
                    count := succ(count);
                    bcd := succ(bcd)
                    end;
           until code=count;
           ebcd := bcd       end        else
        if code = 91  then ebcd:= '[' else
        if code = 92  then ebcd:= '\' else
        if code = 93  then ebcd:= ']' else
        if code = 94  then ebcd:= '^' else
        if code = 95  then ebcd:= '_' else
        if code = 96  then ebcd:= '#' else
        if code <106  then begin
           bcd := 'a' ;
           count := 97 ;
           repeat
              if code > count then
                    begin
                    count := succ(count);
                    bcd := succ(bcd)
                    end;
           until code=count;
           ebcd := bcd       end        else
        if code <115  then begin
           bcd := 'j' ;
```

```
            count := 106;
            repeat
               if code > count then
                     begin
                     count := succ(count);
                     bcd := succ(bcd)
                     end;
            until code=count;
            ebcd := bcd      end         else
         if code <123  then begin
            bcd := 's' ;
            count := 115;
            repeat
               if code > count then
                     begin
                     count := succ(count);
                     bcd := succ(bcd)
                     end;
            until code=count;
            ebcd := bcd      end         else
         if code = 123 then ebcd:= '{' else
         if code = 124 then ebcd:= '|' else
         if code = 125 then ebcd:= '}' else
         if code = 126 then ebcd:= '~' else
         if code > 127 then ebcd:= '?'
      end;


procedure readin ( var in_file : text;
                   var int  :  integer;
                   var rea  :  real;
                   var str  :  str_80;
                   var cha  :  char;
                   var fla  :  flag  );
   var   byte : char;
         count, total : integer;
         B1,B2,B3,B4,B5 : integer;
         sign, expon, mant  : integer;

   procedure readbyte;
    { this procedure strips out bytes apparantly added to the file during
         file transfer   }
      begin
      if not eof ( in_file ) then
           read ( in_file, byte );
      if not eof ( in_file ) then
         begin
         bytecount := bytecount + 1;
         if bytecount > 256 then
             begin
             if ord(byte) = 64 then
                    read ( in_file, byte )
                else
                    write ( output, ord(byte) );
             bytecount := 1
             end;
         end;
```

```
      end;

   begin
     for count:= 1 to 80 do str[count]:=' ' ;
     fla := f_char;
     if not eof(in_file) then readbyte;
     if (ord(byte)=64) then fla:=f_integer;
     if (ord(byte)=255) then fla:=f_real;
     if (ord(byte)=00) then fla:=f_string;

     if fla=f_string then begin
        if not eof(in_file) then begin
           readbyte;
           total := ord (byte);
           for count := 1 to total do begin
              if not eof(in_file) then readbyte;
              str[total - count + 1] := ebcd(    byte );
              end
           end
        end;

     if fla=f_integer then begin
        readbyte ; b1:=ord(byte);
        readbyte ; b2:=ord(byte);
        readbyte ; b3:=ord(byte);
        readbyte ; b4:=ord(byte);
        sign := 1 - 2 * (b1 div 128);
        b1 := b1 - 128* (b1 div 128);
        if sign > 0 then
            int := (((b1*256+b2)*256+b3)*256+B4)
            else
             int := 0-1*((((127-b1)*256+255-b2)*256+255-b3)*256+256-b4)
        end;

     if fla=f_real then begin
        readbyte ; b1 := ord(byte);
        readbyte ; b2 := ord(byte);
        readbyte ; b3 := ord(byte);
        readbyte ; b4 := ord(byte);
        readbyte ; b5 := ord(byte);
        sign := 1-2*  (b4 div 128);
        expon := b5-160;
        b4 := b4 - 128* (b4 div 128);
        rea := ((128+B4)*256+b3)*256 ;
        rea := sign*exp(ln(2)*expon)*(((rea+b2)*256+b1))
        end;

     if fla=f_char then cha := byte:

     end;


   begin

{  ************************     opening up files    ********************** }

      reset ( filein, 'unit=1' );
      bytecount := 0;
```

```
reset ( input , 'interactive' );
rewrite ( output , 'interactive' );
rewrite ( fileout, 'unit=2' );

title_check := true;
for count := 1 to 2 do begin
   readin ( filein, in_int, in_rea, in_str, in_cha, in_fla );
   if in_fla = f_string then title[count] := in_str
                else begin
                writeln ( output, ' *** ERROR IN TITLES *** ') ;
                title_check := false
                end
   end;

readin ( filein, in_int, in_rea, in_str, in_cha, in_fla );
  if in_fla = f_integer then begin
   writeln ( fileout, 'File with ', in_int , ' sets of information');
   number_per_line := 1 + in_int
   end
  else if in_fla = f_real    then begin
   in_int := round(in_rea);
   writeln ( fileout, 'File with ', in_int , ' sets of information');
   number_per_line := 1 + in_int
   end
  else writeln ( output, ' *** ERROR IN NUMBER  ** ');

if title_check then
   for count := 1 to 2 do writeln ( fileout, title[count] );

repeat
   readin ( filein, in_int, in_rea, in_str, in_cha, in_fla );
   if in_fla = f_integer then
         writeln ( output, ' *** ERROR IN COMMENTS *** ');
   if in_fla = f_string then writeln ( fileout, in_str  );
   if in_fla = f_char    then
         writeln ( output, ' *** ERROR IN COMMENTS *** ')
   until in_fla = f_real;

count := 0;
points:= 0;
repeat
   if in_fla=f_real then
         begin
         count := count+1;
         write   ( fileout, in_rea , ' ' )
         end
      else
         write ( output, ' *** ERROR IN DATA SECTION *** ');
   if not ( eof ( filein ) ) then
      readin ( filein, in_int, in_rea, in_str, in_cha, in_fla );
   if count = number_per_line then
      begin
      count := 0;
      writeln ( fileout);
      points := points + 1
      end;
  until eof(filein);
if count <>0 then writeln(fileout);
```

```
        writeln (output, ' file converted : ' ) ;
        writeln (output, number_per_line-1, ' graphs and ',points,' points.')

    end.
```

Listing of FIT.PAS at 00:50:23 on MAY 13, 1987 for CCid=PHP7

```
   (*   Malcolm Hawton    -    Durham  1986 *)
  program plotfit;

   (*    program either reads data from a file, or generates data randomly
         to test a generallised  fitting routine using the NAG
         subroutine  E04FDF  *)

const word_length = 25;

type    sort = ( a_word, an_integer, a_real, end_of_file );
       words  = string(word_length);
const        max_graph = 20;
             max_point = 300;
             line_length = 80;
             max_comment = 35;

type         line = packed array[ 1 .. line_length ] of char;
             graph = 1 .. max_graph;
             point = 1 .. max_point;
             comment = 1 .. max_comment;
             vector = array [ 1 .. max_point ] of real;
             two_lines = array [ 1 .. 2 ] of line;
             graph_lines = array [ graph ] of line;
             comment_lines = array [ comment ] of line;
             graph_vectors = array [ graph ] of vector ;
             graph_reals = array [ graph ] of real ;

const     max_coefficient_number = 10;

type      coefficient_number = 1 .. max_coefficient_number;
          coefficient = array [ coefficient_number ] of real;
          coefficient_lines = array [ coefficient_number ] of line;

var       current_graph, no_of_graphs : graph;
          filein, fileout, log, data : text;
          title : two_lines;
          y_axis, x_axis : line;
          number_string : string(30);
          y_label : graph_lines;
          comments : comment_lines;
          current_comment, no_of_comments : comment;
          fit_co, current_co_no, no_of_coeff, no_of_fixed, initial_fit,
                          no_to_fit : coefficient_number;
          initial_coeff, fit_coeff, error_coeff,
          simul_coeff, fixed_coeff : coefficient;
          name_coeff, name_fixed : coefficient_lines;
          line_read : line;
          out_files : integer;
          out_lines : array [ 1 .. 2, 1 .. 20 ] of integer;
          easy_direction, sum, temperature : real;
          current_point, no_of_points : point;
          x, weight : vector;
          y : graph_vectors;
          y_fit : vector;
          count, int, count2 : integer;
          lines_out, lines, fits : integer;
          iflag1, iflag2 : integer;
          torque_range, mid_torque, smallest_torque: real;
```

```
          ms_field, val_co, stand_dev  : real;
          sum_sq, x_max, x_min: real;
          y_max, y_min : graph_reals;
          word : packed array [ 1 .. 15 ] of char;
          flag : sort;
          answer, choice : char;
          weighted : boolean;
          blank_line : line;
          mass : real;

procedure test;
    { debugging procedure to allow break points to be set easily }
begin
end;

procedure  forfit ( var   m,n : integer;
                    var   x_in, x_corr, weight, y_in : vector;
                    var   initial_coeff : coefficient;
                    var   y_calc, y_corr, y_corr_fit : vector;
                    var   sum_sq : real;
                    var   fit_coeff, error_coeff : coefficient;
                    var   fixed_coeff : coefficient;
                    var   iflag1, iflag2 : integer );
          (* routine written in fortran to interface with E04FDF from
             NAG library version 11 *)
          (* returns y_calc as calculated curve,
               and fit_coeff as coefficients to fit curve if ifail=0 *)
          fortran;

function  g05ddf ( const mean, standard_deviation : real ) : real;
          (*  NAG routine - generates gaussian noise *)
         fortran;

procedure g05ccf ;
          (* NAG routine - randomises gaussian generator *)
          fortran;

procedure readin (
                    var   in_stream    : text;
                    var   read_word    : words ;
                    var   read_int     : integer;
                    var   read_real    : real;
                    var   read_flag    : sort);

   { reads data from a file and sees if it finds a word, number or an integer}

   { requires readin   types  }

   {  M.J.Hawton       Durham 1986 }

  type
    charset = set of char;

  var
    letters,
    numbers,                  aux,
    allowed,
    delimiters: charset;
```

```
    eoinput, decimal_point, exponent, int_f, real_f, end_of_line_done, found,
     digit, first:  boolean;
    read_char: char;
    count: 0 .. word_length;

  begin
    numbers := ['0', '1' .. '9','+','-'];
    aux := ['.','E' ];
    letters := ['a' .. 'z', 'A' .. 'Z', '1' .. '9', '0', '_', '.','+','-'];
    delimiters := ['=', ',', ' ', ';', ':'];
    read_word := '                                ';
    end_of_line_done := false;
    read_int := 0;
    read_real := 0.0;
    count := 0;
    read_flag := end_of_file;
    if (NOT (eof(in_stream)))
      then begin
        eoinput := false;
        found := false;
        digit:= false;
        int_f := true;
        real_f := true;
        decimal_point := false;
        exponent := false;
        first :=true;
        while ((NOT eof(in_stream)) AND (NOT eoinput)) do begin
          if (eoln(in_stream) AND NOT end_of_line_done)
            then begin
              read_char := ' ';
              end_of_line_done := true
              end
            else begin
              read(in_stream, read_char);
              end_of_line_done := false;
              end;
          if (read_char IN letters)
            then begin
              if ((found) and not exponent) then first := false;
              found := true;
              if (NOT (read_char IN numbers))
                then begin
                  int_f := false;
                  if (not (read_char in aux))
                    then
                      real_f := false
                    else begin
                      if (read_char = '.')
                        then begin
                          if (decimal_point OR exponent)
                            then
                              real_f := false
                            else
                              decimal_point := true;
                          end;
                      if (read_char = 'E')
                        then begin
                          if (exponent)
```

```
                              then
                                real_f := false
                              else
                                exponent := true;
                                first := true;
                            end;
                        end;
                    end
                  else
                    begin
                    if ((read_char = '+') OR (read_char = '-'))
                      then begin
                        if (not first)
                          then begin
                          real_f := false;
                          int_f := false;
                          end;
                      end
                      else
                        digit := true;
                    first := false;
                    end;
                if count < word_length
                  then begin
                    count := count + 1;
                    read_word[count] := read_char;
                    end;
                end;
            if ((read_char IN delimiters) AND found)
              then begin
                eoinput := true;
                read_flag := a_word;
                end;
            end;
          if ((real_f) AND found AND digit)
            then begin
              read_flag := a_real;
              readstr(read_word, read_real);
              end;
          if ((int_f) AND found and digit)
            then begin
              readstr(read_word, read_int);
              read_flag := an_integer;
              end;


        end
    end {readin};
procedure writestring ( var   text_file : text;
                        var   outline : line );

var   count, actual   : 1 .. line_length;

begin

   actual := line_length;
   while ((outline [actual] = ' ') and (actual > 1)) do actual:= actual-1;
   for count := 1 to actual do write ( text_file, outline[count] );
```

```
    end;

  procedure writelnstr  ( var   text_file : text;
                          var   outline : line );

  var   count, actual   : 1 .. line_length;

  begin

     actual := line_length;
     while ((outline [actual] = ' ') and (actual > 1)) do actual:= actual-1;
     for count := 1 to actual do write ( text_file, outline[count] );
     writeln

   end;
procedure read_file_in (
        var  filin : text;
        var  title : two_lines;
        var  x_axis, y_axis : line;
        var  y_label : graph_lines;
        var  no_of_comments : comment;
        var  no_of_graphs : graph;
        var  no_of_points : point;
        var  comments : comment_lines;
        var  x : vector;
        var  y : graph_vectors;
        var  x_min, x_max : real;
        var  y_min, y_max : graph_reals ) ;

   { reads data from file, which is down loaded from the BBC }

   { uses  readin  procedure
           file    definitions  }

   (*   Malcolm Hawton   -   Durham  1986 *)

  var
     current_graph     : graph;
     current_comment   : comment;
     com_read,blank_line : line;
     current_point     : 1 .. max_point+1;
     file_with         : packed array [ 1 .. 10 ] of char;
     flag              : sort;
     int, count        : integer;
     word              : words ;
     number, y_in      : real;

   begin

      for count:= 1 to line_length do
            blank_line[count]  := ' ';

      repeat
         readin(filin, word, int, number, flag)
         until ( flag=an_integer );
      readln(filin);
      no_of_graphs := int;
```

```
        readln ( filin, title[1] );
        readln ( filin, title[2] );
        readln ( filin, x_axis );
        readln ( filin, y_axis );
        for current_graph:= 1 to no_of_graphs do
            begin
            readln ( filin, y_label[current_graph] );
            end;
        com_read := 'start';
        current_comment := 1;
        while com_read <> blank_line do
            begin
            readln ( filin, com_read );
            comments[current_comment] := com_read ;
            current_comment := current_comment + 1;
            end;
        no_of_comments := current_comment - 1;
        current_point := 1;
        x_max := -99999;
        x_min := +99999;
        for current_graph := 1 to no_of_graphs do
            begin
            y_max [ current_graph ] := -999999;
            y_min [ current_graph ] := 999999;
            end;
        while not eof ( filin ) do
            begin
            readin(filin, word, int, number, flag);
            if ((flag=a_real) or (flag=an_integer)) then
               begin
               x[current_point]:=number;
               if x [current_point] > x_max then x_max:=x[current_point];
               if ( x [current_point] < x_min ) and
                  ( x [current_point] + 999 > 2 )
                                then x_min:=x[current_point];
                 for current_graph:= 1 to no_of_graphs do
                 begin
                 readin(filin, word, int, number, flag);
                 if ((flag=a_real) or (flag=an_integer)) then
                    begin
                    y_in:=number;
                    if ( abs ( y_in + 999 ) > 0.001 ) then
                       begin
                       if y_in > y_max[current_graph] then
                            y_max[current_graph]:= y_in;
                       if y_in < y_min[current_graph] then
                            y_min[current_graph]:= y_in
                      end;
                    y [ current_graph, current_point ] := y_in
                    end
                 end;
               readln ( filin );
               current_point := current_point + 1;
               end;
            end;
        no_of_points := current_point - 1;
        end;
    procedure write_file    (
```

Listing of FIT.PAS at 00:50:23 on MAY 13, 1987 for CCid=PHP7

```
        var   filout : text;
        const title : two_lines;
        const x_axis, y_axis : line;
        const y_label : graph_lines;
        const no_of_comments : comment;
        const no_of_graphs : graph;
        const no_of_points : point;
        const comments : comment_lines;
        const x : vector;
        const y : graph_vectors );


    { writes file in similar form to that from the BBC }

    {  requires file definitions }


    (*   Malcolm Hawton   -   Durham  1986 *)

   var
      current_graph     : graph;
      current_comment   : comment;
      blank80           : line;
      current_point     : point;
      count             : integer;
      fileout           : text;

    begin

       for count:= 1 to 80 do
             blank80[count]  := ' ';

       writeln(filout,'FILE WITH ',no_of_graphs:2,' GRAPHS' );
       writeln(filout,title[1] );
       writeln(filout,title[2] );
       writeln(filout,x_axis );
       writeln(filout,y_axis );
       for current_graph:= 1 to no_of_graphs do
           writeln(filout, y_label[current_graph] );
       for current_comment := 1 to no_of_comments do
           writeln(filout, comments[current_comment] );
       writeln(filout,blank80);
       for current_point := 1 to no_of_points do
           begin
           write   ( filout, x [current_point]:15   );
           for current_graph:= 1 to no_of_graphs do
              if ( y[current_graph, current_point] = -999 )
              then
              write  (filout, ' ', y [ current_graph, current_point ] )
              else
              write  (filout, ' ', y [ current_graph, current_point ]:15   );
           writeln(filout) ;
           end;
       end;
 procedure  fit ( var   m,n : integer;
                  var   x_in, x_corr, weight, y_in : vector;
                  var   initial_coeff : coefficient;
                  var   y_calc, y_corr, y_corr_fit : vector;
```

```
                var    sum_sq : real;
                var    fit_coeff, error_coeff : coefficient;
                var    fixed_coeff : coefficient;
                var    iflag1, iflag2 : integer );
        (* routine calculates fourier coefficients from K values,
           calls forfit and then calculates anisotropy constants back *)

    const  max_index = 5;

    type    index = 1 .. max_index;
        vector = array [ index ] of real;
        matrix = array [ index ] of vector;

    var     four1  , anis1,four2  , anis2, four3  , anis3   : vector;
        F_to_K, K_to_F : matrix;
        count : integer;
        col, row, max_const: index;
        current_coeff : coefficient_number;
        print, easy_dir : boolean;

    function mat_vec  ( mat : matrix; vec : vector;  size : index ):vector;

       var   row , col : index;
          vec1 : vector;

       function vec_prod ( vec1, vec2 : vector;  size : index ):real;

          var    row : index;
                 sum : real;

          begin
             sum := 0;
             for row := 1 to size
                   do sum := sum + vec1[row] * vec2[row];
             vec_prod := sum
          end;   { vec_prod }


       begin
          for row  := 1 to size
             do begin
             for col  := 1 to size
                do vec1[col ] := mat [col , row ];
             mat_vec [ row ] := vec_prod ( vec1, vec, size );
             end
       end;   { mat_vec  }

    begin {fit}


       if iflag1=1 then print:=false else print:=true;
       max_const := n - 4 ;
       for row := 1 to max_index
          do for col := 1 to max_index
             do begin
                F_to_K[col,row] := 0;
                K_to_F[col,row] := 0
                end;
```

```
K_to_F [ 1, 1 ] := 1;
K_to_F [ 2, 1 ] := 1;
K_to_F [ 2, 2 ] := 0 - 0.5;
K_to_F [ 3, 1 ] := 0.9375;
K_to_F [ 3, 2 ] := 0-0.75;
K_to_F [ 3, 3 ] := 0.1875;
K_to_F [ 4, 1 ] := 0.875 ;
K_to_F [ 4, 2 ] := 0-0.875;
K_to_F [ 4, 3 ] := 0.375 ;
K_to_F [ 4, 4 ] := 0-0.0625;
K_to_F [ 5, 1 ] := 0.8203125;
K_to_F [ 5, 2 ] := 0-0.9375;
K_to_F [ 5, 3 ] := 0.52734375;
K_to_F [ 5, 4 ] := 0-0.15625;
K_to_F [ 5, 5 ] := 0-0.01953125;
F_to_K [ 1, 1 ] := 1;
F_to_K [ 2, 1 ] := 2;
F_to_K [ 2, 2 ] := 0-2 ;
F_to_K [ 3, 1 ] := 3;
F_to_K [ 3, 2 ] := 0-8;
F_to_K [ 3, 3 ] := 16/3;
F_to_K [ 4, 1 ] := 4;
F_to_K [ 4, 2 ] := 0-20;
F_to_K [ 4, 3 ] := 32;
F_to_K [ 4, 4 ] := 0-16;
F_to_K [ 5, 1 ] := 5;
F_to_K [ 5, 2 ] := 0-40;
F_to_K [ 5, 3 ] := 112;
F_to_K [ 5, 4 ] := 0-128;
F_to_K [ 5, 5 ] := 51.2;
for row := 1 to max_const do
   begin
   current_coeff := row + 4;
   anis1[row] := initial_coeff[current_coeff];
   end;
four1:= mat_vec ( K_to_F , anis1, max_const );
for row := 1 to max_const do
   begin
   current_coeff := row + 4;
   initial_coeff[current_coeff] := four1[row];
   fit_coeff[current_coeff] := four1[row];
   error_coeff[current_coeff] := four1[row];
   end;

count := 0;
repeat
   easy_dir := true;
   forfit ( m , n ,  x_in, x_corr, weight, y_in,
     initial_coeff, y_calc, y_corr, y_corr_fit,
     sum_sq, fit_coeff, error_coeff, fixed_coeff,
     iflag1, iflag2        );
   count := 1 + count;
   for row := 1 to max_const do
      begin
      current_coeff := row + 4;
      four1[row] := initial_coeff[current_coeff];
      four2[row] := fit_coeff[current_coeff];
      four3[row] := error_coeff[current_coeff];
```

```
            if print then
              begin
              writeln ( 'Fourier coeff(',row:2,')=',four2[row],
                                                  '+-',four3[row]);
              writeln ( log, 'Fourier coeff(',row:2,')=',
                                  four2[row],'+-',four3[row]);
              end;
            end;
         if fit_coeff[5] < 0 then
            begin
            if print then
              writeln ( ' - not on easy direction ');
            easy_dir := false;
            fit_coeff[1] := fit_coeff[1] - 90;
            fit_coeff[4] := fit_coeff[4] - 90;
            fit_coeff[5] := fit_coeff[5] * (-1);
            fit_coeff[7] := fit_coeff[7] * (-1);
            fit_coeff[9] := fit_coeff[9] * (-1);
            initial_coeff := fit_coeff
            end;
         until (count>5) or easy_dir;

      anis1:= mat_vec ( F_to_K , four1, max_const );
      anis2:= mat_vec ( F_to_K , four2, max_const );
      anis3:= mat_vec ( F_to_K , four3, max_const );
      for row := 1 to max_const do
         begin
         current_coeff := row + 4;
         initial_coeff[current_coeff] := anis1[row];
         fit_coeff[current_coeff] := anis2[row];
         error_coeff[current_coeff] := anis3[row];
         end;

   end; {fit}


procedure generate (
         var  graph_coeff, fixed_coeff : coefficient;
         var  stand_deviation  : real ;
         var  title : two_lines;
         var  x_axis, y_axis : line;
         var  y_label : graph_lines;
         var  no_of_comments : comment;
         var  no_of_graphs : graph;
         var  no_of_points : point;
         var  comments : comment_lines;
         var  x : vector;
         var  y : graph_vectors;
         var  x_min, x_max : real;
         var  y_min, y_max : graph_reals );

   { generates a graph   }

   { uses  fit procedure written in fortran to interface
           with NAG routines     }

   (*   Malcolm Hawton   -   Durham  1986 *)
```

```
var
   current_co_no : coefficient_number;
   current_point : point;
   count         : integer;
   mean          : real;
   blank_line : line;
   read_word : words     ;
   read_int  : integer;
   read_real : real;
   read_flag : sort;

begin

   for count := 1 to line_length
      do blank_line[count] := ' ';

   title[1] := 'Simulated torque curve';
   title[2] := 'with gaussian noise';
   x_axis := 'Rotation/degrees';
   y_axis := 'Torque/NM';
   y_label[1] := 'simulated torque';
   no_of_graphs := 1;
   writeln ( 'How many points do you want generated');
   readln ( no_of_points );
   x_min := 0.0;
   x_max := 359.9;
   for current_point := 1 to no_of_points do
     begin
     y[1,current_point] := 0.0;
     weight[current_point] := 1.0;
     x[current_point] := x_min +
           (x_max - x_min) * (current_point - 1)/(no_of_points - 1)
     end;
   writeln ( log );
   writeln ( log, '**** generating simulated data ****');
   writeln ( 'fit to function defined LSFUN1 in FIT.FOR ');
   writeln ( 'Please give values for the coefficients ( or return )');
   for current_co_no := 1 to no_of_coeff do
     begin
     writestring ( output, name_coeff[current_co_no] );
     write ( '(',graph_coeff[current_co_no],')?' ); writeln;
     readin ( input, read_word, read_int, read_real  , read_flag );
     if ( read_flag=a_real ) or (read_flag=an_integer ) then
        graph_coeff[current_co_no] := read_real;
     comments[current_co_no]:=name_coeff[current_co_no];
     comments[current_co_no,18]:='=';
     writestr(number_string, graph_coeff[current_co_no]:10);
     for count := 1 to length(number_string) do
         comments[current_co_no,count+20] := number_string[count];
     writestring ( log, name_coeff[current_co_no] );
     write ( log, ' = ',graph_coeff[current_co_no]);
     writeln (log );
     end;
   no_of_comments := no_of_coeff + 1;
   for current_co_no := 2 to no_of_fixed do
               { note : fixed[1] is calc from fixed[2] }
     begin
```

```
      writestring ( output, name_fixed[current_co_no] );
      write ( '(',fixed_coeff[current_co_no],')?' ); writeln;
      readin ( input, read_word, read_int, read_real  , read_flag );
      if ( read_flag=a_real ) or (read_flag=an_integer ) then
         fixed_coeff[current_co_no] := read_real;
      comments[no_of_comments]:=blank_line;
      comments[no_of_comments]:=name_fixed[current_co_no ];
      comments[no_of_comments,18]:='=';
      writestr(number_string, fixed_coeff[current_co_no]:10);
      for count := 1 to length(number_string) do
          comments[no_of_comments,count+20] := number_string[count];
      no_of_comments := 1 + no_of_comments;
      writestring ( log, name_fixed[current_co_no] );
      write ( log, ' = ',fixed_coeff[current_co_no]);
      writeln (log );
      end;
   fixed_coeff[1]:=180/(3.14159*fixed_coeff[2]);
   writeln ( ' what is the standard deviation of the noise ');
   writeln ('(presently ',stand_deviation,')');
   readin ( input, read_word, read_int, read_real  , read_flag );
   if ( read_flag=a_real ) or (read_flag=an_integer ) then
        stand_deviation:= read_real;
   comments[no_of_comments]:=blank_line;
   comments[no_of_comments]:='stand. dev. noise =  ' ;
   writestr(number_string, stand_deviation);
   for count := 1 to length(number_string) do
       comments[no_of_comments,count+20] := number_string[count];
   no_of_comments := 1 + no_of_comments;
   comments[no_of_comments]:=blank_line;
   write ( log, 'noise standard deviation');
   write ( log, ' = ',stand_deviation );
   writeln (log);
   mean := 0;
   count := 0;
   repeat
      count := count + 1;
      iflag1 := 1;
      iflag2 := 0;
      fit (  no_of_points, no_of_coeff, x, y[9], y[10], weight,
             graph_coeff, y[1], y[2], y[3], sum_sq, fit_coeff,
             error_coeff , fixed_coeff, iflag1, iflag2 );
      y[10] := y[1];
      until (count >= 20 );

   y_min[1]:=9999999.0;
   y_max[1]:=-9999999.0;
   g05ccf;      (* randomise  random numbers *)
   for current_point := 1 to no_of_points do
       begin
       y[1,current_point] :=
              g05DDF ( mean, stand_deviation ) + y[1,current_point];
                          (*  add gaussian noise *)
       if (y[1,current_point]>y_max[1]) then y_max[1]:=y[1,current_point];
       if (y[1,current_point]<y_min[1]) then y_min[1]:=y[1,current_point]
       end;
   writeln ;
   writeln ( ' ******  simulated data generated **** ' );
   writeln ;
```

```
      end;

begin

    reset ( input  , 'interactive' );
    reset ( filein, 'unit=1');
    rewrite ( output , 'interactive' );
    rewrite ( fileout, 'unit=2' );
    rewrite ( log, 'unit=3 noempty' );
    rewrite ( data, 'unit=4 noempty' );

    for count := 1 to line_length do
        blank_line[count] := ' ';

    name_coeff[1] := 'phase shift (deg)       ';
    name_coeff[2] := 'torque offset           ';
    name_coeff[3] := 'side pull (sinx)        ';
    name_coeff[4] := 'phase side pull         ';
    name_coeff[5] := 'K1 / J/kg               ';
    name_coeff[6] := 'K2 / J/kg               ';
    name_coeff[7] := 'K3 / J/kg               ';
    name_coeff[8] := 'K4 / J/kg               ';
    name_coeff[9] := 'K5 / J/kg               ';
    no_of_coeff:= 9;
    for current_co_no := 1 to no_of_coeff do
       simul_coeff[current_co_no] := 0;
    name_fixed[1] := 'shear correction        ';
    name_fixed[2] := '(Ms*Field)              ';
    name_fixed[3] := 'torque offset           ';
    no_of_fixed:= 3;
    for current_co_no := 1 to no_of_fixed do
       fixed_coeff[current_co_no] := 0;
    stand_dev := 0;
    lines_out := 0;
    out_files := 0;
    fits := 0;
    weighted := false;

    writeln;
    writeln (   'Program to fit anisotropy constants to torque data ');
    writeln ;
    writeln (   ' This program fits a curve to a data file, mimimising the');
    writeln (   ' squared error. ');
    writeln (   ' The number of anisotropy constants are chosen, and whether');
    writeln (   ' the points nearer the origin are considered more important');
    writeln (   ' (weighting)' );
    writeln  :
    writeln ('Do you want to test the routine with simulated data(y/n)');
    readln ( answer );
    if  ((answer='y')or(answer='Y'))
       then begin
            generate    (  simul_coeff, fixed_coeff, stand_dev,
                           title, x_axis, y_axis, y_label,
                           no_of_comments, no_of_graphs, no_of_points,
                           comments, x, y, x_min, x_max, y_min, y_max );
            end

       else begin
```

```
            read_file_in (  filein, title, x_axis, y_axis, y_label,
                            no_of_comments, no_of_graphs, no_of_points,
                           comments, x, y, x_min, x_max, y_min, y_max );
            line_read :=  'curve read in : ' ;
            for count := 1 to 64 do
                 line_read[count+16] := y_label[1,count];
            writelnstr ( output, line_read )
            end;

    torque_range := y_max[1] - y_min[1];
    mid_torque   := (y_max[1] + y_min[1])/2;
    smallest_torque := abs(y[1,1]-mid_torque);
    easy_direction := x[1];
    for current_point := 1 to no_of_points do
      begin
      weight[current_point] := 1;
      if ( abs(y[1,current_point]-mid_torque) < smallest_torque ) then
         begin
         easy_direction := x[current_point];
         smallest_torque := abs(y[1,current_point]-mid_torque)
         end
      end;

    fit_coeff[1] :=-easy_direction;
    fit_coeff[2] := mid_torque;
    fixed_coeff[3] := mid_torque;
    fit_coeff[3] := 0.01 * torque_range;
    fit_coeff[4] := 0 ;
    fit_coeff[5] := 0.8  * torque_range;
    fit_coeff[6] :=-0.4  * torque_range;
    fit_coeff[7] := 0.1  * torque_range;
    fit_coeff[8] := 0.01 * torque_range;
    fit_coeff[9] := 0.01 * torque_range;

    writeln ( 'What is the value of Ms*Field ( joules )');
    readln ( Ms_field );
    writeln ( 'What is the temperature (Kelvin)');
    readln ( Temperature );
    fixed_coeff[2] := Ms_field;
    fixed_coeff[1] := 180/(3.14159*Ms_field); {shear_correction}
    no_of_fixed:= 2;
    no_of_coeff:= 6;

    writeln ( log );
    writelnstr ( log, y_label[1] );
    writeln ( log );
    writeln ( log, 'fixed coefficients  ' );
    for current_co_no := 1 to no_of_fixed do
       begin
       writestring ( log, name_fixed[current_co_no]);
       write ( log, ' =',fixed_coeff[current_co_no]);
       writeln ( log )
       end;
    writeln ( log );
    writeln ( log, 'initial coefficients  ' );

    for current_co_no := 1 to no_of_coeff do
       begin
```

```
      writestring ( output, name_coeff[current_co_no]);
      write ( output, ' =',fit_coeff[current_co_no]);
      writeln ( output );
      writestring ( log    , name_coeff[current_co_no]);
      write ( log    , ' =',fit_coeff[current_co_no]);
      writeln ( log     )
      end;

   repeat

     writeln ;
     writeln ( ' choose from:');
     writeln ;
     writeln ( ' 1 - 5 : no. of anisotropy const (presently ',
                    no_of_coeff - 4:2,').');
     writeln ( ' s - change starting coeff values.');
     if weighted then writeln ( ' w - stop weighting (presently on)')
                 else writeln ( ' w - switch weighting on (presently off)');
     writeln ( ' m - change Ms*Field value.');
     writeln ( ' f - fit (least squares)');
     writeln ( ' o - output to data file. ');
     writeln ( ' g - generate new simulated data. ');
     writeln ( ' e - end ');
     readln ( choice );
     if ( choice='1' ) then no_of_coeff := 5;
     if ( choice='2' ) then no_of_coeff := 6;
     if ( choice='3' ) then no_of_coeff := 7;
     if ( choice='4' ) then no_of_coeff := 8;
     if ( choice='5' ) then no_of_coeff := 9;

     if ((choice='s')or(choice='S')) then
        begin
        writeln ( ' give new value or -999 ');
        for current_co_no := 1 to no_of_coeff do
           begin
           writestring ( output, name_coeff[current_co_no] );
           writeln('=', fit_coeff[current_co_no]);
           readln ( val_co);
           if ( val_co<> -999 ) then fit_coeff[current_co_no] := val_co;
           end
        end;

     if ((choice='m')or(choice='M')) then
        begin
        writeln ( 'Ms*Field =',Ms_field,'Joules');
        writeln ( 'What is the new value of Ms*Field ( joules )');
        readln ( Ms_field );
        fixed_coeff[2] := Ms_field;
        fixed_coeff[1] := 180/(3.14159*Ms_field); {shear_correction}
        writeln ( log );
        writeln ( log, ' - new fixed coefficients  ' );
        for current_co_no := 1 to no_of_fixed do
           begin
           writestring ( log, name_fixed[current_co_no]);
           write ( log, ' =',fixed_coeff[current_co_no]);
           writeln ( log )
           end
        end;
```

```
if ((choice='w')or(choice='W')) then
   begin
      if weighted
      then begin
      weighted := false;
      writeln ( log, ' - weighting switched on    ' );
      for current_point := 1 to no_of_points do
         weight[current_point] := 1;
      end
   else begin
      weighted := true ;
      writeln ( log, ' - weighting switched off   ' );
      for current_point := 1 to no_of_points do
         weight[current_point] :=
         sin(3.14159*(0.01+0.98*(y[1,current_point]-y_min[1])
                     /torque_range));
      end
   end;


if ((choice='f')or(choice='F')) then
   begin
   initial_coeff := fit_coeff;
   iflag1 := 0;
   writeln ( log );
   writeln ( log, '- fitting to ',no_of_coeff:3,' coefficients' );
   writeln (  'least squares fit starting . . .' );
   fit ( no_of_points, no_of_coeff,x, y[5],y[1], weight, initial_coeff,
    y[2], y[3], y[4], sum_sq, fit_coeff, error_coeff,
    fixed_coeff, iflag1, iflag2);
   writeln ( log, 'indication flags are ',iflag1,',',iflag2 );
   writeln ( 'least squares fit finished . . . ');
   writeln ( 'indication flags are ',iflag1,',',iflag2 );
   for current_co_no := 1 to no_of_coeff do
     begin
     writestring ( log, name_coeff[current_co_no]);
     write ( log, ' =',fit_coeff[current_co_no]);
     write ( log, '+-',error_coeff[current_co_no]);
     writeln ( log );
     writestring ( output, name_coeff[current_co_no]);
     write ( output, ' =',fit_coeff[current_co_no]);
     write ( output, '+-',error_coeff[current_co_no]);
     writeln ( output )
     end;
   write ( data, temperature:5:0 );
   write ( data, ' ');
   for current_co_no := 1 to no_of_coeff do
      begin
      write ( data, fit_coeff[current_co_no]:15  );
      write ( data, ' ')
      end;
   for current_co_no :=      no_of_coeff to 10 do
      write ( data, ' 0.0' );
   writeln;
   writeln ( data );
   fits := fits + 1
   end;
```

```
     if ((choice='o')or(choice='O')) then
        begin
 {      for current_point := 1 to no_of_points do y[10, current_point] := 0;
        count2:= 0;
        repeat  } { loop to calculate the smooth curve
                    not using measured torques for shear correction }
{          count2:= count2+ 1;
           iflag1 := 1;
           iflag2 := 0;
           test;
           fit (  no_of_points, no_of_coeff, x, y[10], weight,   fit_coeff,
             y[2], y[9], y[4], sum_sq, initial_coeff, error_coeff ,
             fixed_coeff, iflag1, iflag2 );
           y[10] := y[2];
           until (count2>= 20 );
 }
        for current_co_no:= 1 to no_of_fixed do
           begin
           comments[no_of_comments]:=name_fixed[current_co_no];
           comments[no_of_comments,18]:='=';
           writestr ( number_string, fixed_coeff[current_co_no]:10 );
           for count := 1 to length(number_string) do
              comments[no_of_comments , count+20] := number_string[count];
           no_of_comments:=1+no_of_comments;
           end;

        for current_co_no:= 1 to no_of_coeff do
           begin
           comments[no_of_comments]:=name_coeff[current_co_no];
           comments[no_of_comments,18]:='=';
           writestr ( number_string, fit_coeff[current_co_no]:10 );
           for count := 1 to length(number_string) do
              comments[no_of_comments , count+20] := number_string[count];
           writestr ( number_string, error_coeff[current_co_no]:10 );
           for count := 1 to length(number_string) do
              comments[no_of_comments , count+31] := number_string[count];
           comments[no_of_comments,31]:='+';
           comments[no_of_comments,32]:='-';
           no_of_comments:=1+no_of_comments;
           end;


        if weighted then
           begin
           comments[no_of_comments] := ' weighted data points ';
           no_of_comments := 1 + no_of_comments;
           end;

        comments[no_of_comments] := blank_line;

        no_of_graphs := 5;
        y_label[2] := 'fit to ';
        for count := 1 to 73 do
           y_label[2,count+7] := y_label[1,count];

        y_label[3] := 'Corrected ';
        for count := 1 to 70 do
```

```
            y_label[3,count+10] := y_label[1,count];

         y_label[4] := 'Fit to cor. ';
         for count := 1 to 68 do
            y_label[4,count+12] := y_label[1,count];
         y_label[5] := 'Corrected angle/deg. ';

         write_file  (    fileout, title, x_axis, y_axis, y_label,
                             no_of_comments, no_of_graphs, no_of_points,
                             comments, x, y );

         lines :=  6 + no_of_graphs + no_of_comments + no_of_points;
         out_files := 1 + out_files;
         out_lines[1,out_files] := lines_out + 1;
         out_lines[2,out_files] := lines_out + lines;
         writeln ( lines:5, 'lines written, from line ',lines_out+1:5,' to ',
                              lines_out+lines:5,'.');
         writeln( log, '******  written to output file lines ',
                    lines_out+1:5,' to ', lines_out+lines:5,'.');
         lines_out := lines_out + lines;


         no_of_comments := no_of_comments - no_of_fixed - no_of_coeff    ;
         if weighted then no_of_comments := no_of_comments - 1
         end;

   if  ((choice='g')or(choice='G'))
      then begin
      no_of_coeff:= 9;
      generate     (  simul_coeff, fixed_coeff, stand_dev,
                             title, x_axis, y_axis, y_label,
                             no_of_comments, no_of_graphs, no_of_points,
                             comments, x, y, x_min, x_max, y_min, y_max );

      no_of_fixed:= 2;
      no_of_coeff:= 6;
      writeln ( log, '****** new data generated  ' );
      end

   until ((choice='e')or(choice='E'));
   writeln ( ' Thank you :  fits done',fits:3,', files output',out_files:3,
                        ', lines output',lines_out:5,'.');
   if out_files>0 then
      for count := 1 to out_files
         do writeln ( ' - file ',count:2,' lines ',out_lines[1,count]:5,
                                          ' to ',out_lines[2,count]:5 )


end.
```

```
Listing of FIT.FOR at 00:52:27 on MAY 13, 1987 for CCid=PHP7

        SUBROUTINE FORFIT ( M, N, X, XCORR, Y, WT, INCO, YCALC, YCORR,
     *          YCFIT, FSUMSQ, FITCO, ERRCO, FIXCO, IFLAG1, IFLAG2 )
C
       INTEGER  M, N, IFLAG1, IFLAG2, POINT, LIW, IW(10), LW, IFAIL,
     *                 CO, COUNT, CFLAG
C
       DOUBLE PRECISION NUMBER, X(1000), Y(1000), YCALC(1000),
     *          YCFIT(1000), WT(1000), INCO(10), FITCO(10), ERRCO(10),
     *          FSUMSQ, W(23220), YCORR(1000), CX(1000), CY(1000),
     *          CWT(1000), CJ(10), CINCO(10), FIXCO(10), CFIXCO(10),
     *          CXCORR(1000), XCORR(1000)
C
          COMMON CX, CY, CXCORR, CWT, CINCO, CFIXCO, CFLAG
C
C         COPY THE DATA INTO A COMMON BLOCK
C
          CFLAG=IFLAG2
          DO 50 CO = 1,10
          CINCO(CO) = INCO (CO)
          CFIXCO(CO) = FIXCO (CO)
50        CONTINUE
C
          DO 100  POINT = 1, M
            CX(POINT) = X (POINT)
            CY(POINT) = Y (POINT)
            CWT(POINT) = DSQRT(WT(POINT))
 100        CONTINUE
C
C       SET FIT COEFF TO INITIAL VALUES
C
          DO 200 COEF = 1 , 10
            FITCO(COEF) = INCO(COEF)
 200        CONTINUE
C
C         IF OPTION NOT TO MINIMISE THE SQUARE ERROR
C         THEN  SKIP THAT SECTION
C
          IF (IFLAG1 .EQ. 1) GO TO 1000
C
C ------------------------------------------------------------------------
C
C       SET UP VALUES REQUIRED BY LEAST SQUARES ROUTINE
C
            LW = 23220
            LIW = 10
            COUNT = 0
C
C      REPEAT CALL OF LEAST SQUARES ROUTINE UP TO TEN TIMES
C
 300        CONTINUE
            COUNT = COUNT + 1
            IFAIL = 1
            CALL E04FDF ( M, N, FITCO, FSUMSQ, IW, LIW, W, LW, IFAIL )
            IF ( ( IFAIL .EQ. 2 ) .AND. ( COUNT .LT. 10 ) ) GO TO 300
            IFLAG1 = IFAIL
C
C             SKIP THE CALCULATION OF VARIENCE IF THE CALL TO FIND A
```

```
C                MINIMUM WAS UNSUCCESFUL.
C
            IF ( IFAIL .EQ. 1 ) GO TO 2000
C
C-----------------------------------------------------------------------
C
            NS = 6*N + 2*M + M*N + 1 + (N*(N-1))/2
            IF  ( N .NE. 1 ) GO TO 350
               NS = NS + 1
 350        CONTINUE
            NV = NS + N
            IFAIL = 1
            CALL E04YCF ( 0, M, N, FSUMSQ, W(NS), W(NV), N, CJ, W, IFAIL )
C
            IFLAG2 = IFAIL
C
C                SKIP CALCULATION OF STANDARD ERROR IF ROUTINE FAILS
C
            IF ( ( IFAIL .EQ. 1 ) .OR. ( IFAIL .EQ. 2 ) ) GO TO 2000
C
             DO 400 COEFF = 1 , N
               ERRCO (COEFF) = DSQRT ( CJ (COEFF) )
 400           CONTINUE
C
C
C -----------------------------------------------------------------------
C
 1000       CONTINUE
C
C  ****** CALCULATING THE CURVE *******
C
C     ( N NEGATIVE => LSFUN1 CALC VALUE NOT ERROR )
            N = -N
            CALL LSFUN1 ( M, N, FITCO, YCALC )
            N = -N
C
C     ( M NEGATIVE => LSFUN1 CALC CORRECTED CURVE )
            M = -M
            CALL LSFUN1 ( M, N, FITCO, YCORR )
C
C     ( N & M NEGATIVE => LSFUN1 CALC FIT TO CORRECTED CURVE )
            N = -N
            CALL LSFUN1 ( M, N, FITCO, YCFIT )
            N = -N
            M = -M
C
            DO 500  POINT = 1, M
              XCORR(POINT) = CXCORR(POINT)
 500          CONTINUE
C
C
 2000       CONTINUE
            RETURN
          END
C
C
C*************************************************************************
C*************************************************************************
```

```
Listing of FIT.FOR at 00:52:27 on MAY 13, 1987 for CCid=PHP7

      SUBROUTINE  LSFUN1 (M,N, XC, FVECC )
C
C       SUBROUTINE CALCULATES USING ANISOTROPY CONST COEFFICIENTS
C
        INTEGER    M,N,POINT,COUNT,ABSN,ABSM,CFLAG
        DOUBLE PRECISION X, CX(1000), CY(1000), CXCORR(1000), COSX,
     *     CWT(1000), CFIXCO(10),CXC(10),XC(10), FVECC(1000), SINX
        COMMON CX,CY,CXCORR,CWT,CXC,CFIXCO,CFLAG
C
C
          ABSN=ABS(N)
          ABSM=ABS(M)
C
          DO 100 POINT = 1 , ABSM
C
C     DISPLACEMENT AND SIN(X) TERM
C
          FVECC(POINT) = XC(2) +
C         FVECC(POINT) = CFIXCO(3) +
     -              XC(3)*DSIN((CX(POINT)+XC(4))*3.14159/180)
C
C     SHEAR CORRECTION - DONE FOR ALL OPTIONS
C
          CXCORR(POINT) =  CX(POINT)-XC(1)
C         CXCORR(POINT) =  CX(POINT)+229.0
     -          +(CY(POINT)-FVECC(POINT))*CFIXCO(1)
          X = CXCORR(POINT) * 3.14159 / 180
C
C     DISPLACEMENT AND SIN(X) TERM, NOT CALC FOR FIT TO CORRECTED CURVE.
C
          IF ((M .GE. 0) .OR. (N .GE. 0)) GOTO 15
          FVECC(POINT) = 0
C
C     CALCULATED CURVE FIT NOT DONE FOR CORRECTED CURVE
C
15        CONTINUE
          IF ((M .LT. 0) .AND. (N .GE. 0)) GOTO 20
C
C     CALCULATION OF FOURIER CURVE FIT
C
           FVECC(POINT) = FVECC(POINT) - XC(5)*DSIN(2*X)
           IF ( ABSN .LT. 6 ) GO TO 20
            FVECC(POINT) = FVECC(POINT) - XC(6)*DSIN(4*X)
           IF ( ABSN .LT. 7 ) GO TO 20
            FVECC(POINT) = FVECC(POINT) - XC(7)*DSIN(6*X)
           IF ( ABSN .LT. 8 ) GO TO 20
            FVECC(POINT) = FVECC(POINT) - XC(8)*DSIN(8*X)
           IF ( ABSN .LT. 9 ) GO TO 20
            FVECC(POINT) = FVECC(POINT) - XC(9)*DSIN(10*X)
C
C       CALCULATION OF THE VALUE AND NOT THE ERROR
C
 20        CONTINUE
            IF  ( N .LT. 0 ) GO TO 50
            FVECC(POINT) = (CY(POINT) - FVECC(POINT))
C
C       FOR CORRECTED CURVE NO WEIGHTING IS REQUIRED
C
```

```
                IF  ( M .LT. 0 ) GO TO 50
                FVECC(POINT) = FVECC(POINT)*CWT(POINT)
C
C
 50             CONTINUE
 100            CONTINUE
C
              RETURN
            END
C                 (LSFUN1)
            END
```

Listing of GRAPH.PLOT at 01:04:01 on MAY 13, 1987 for CCid=PHP7

```
   (*   Malcolm Hawton   -   Durham  1986 *)
 program Graph;
   { read data from file and produce plot descriptor file for
     laser printer or calcomp plotter }

const word_length = 25;

type    sort = ( a_word, an_integer, a_real, end_of_file );
       words  = string(word_length);
const        max_graph = 20;
             max_point = 300;
             line_length = 80;
             max_comment = 35;

type         line = packed array[ 1 .. line_length ] of char;
             graph = 1 .. max_graph;
             point = 1 .. max_point;
             comment = 1 .. max_comment;
             vector = array [ 1 .. max_point ] of real;
             two_lines = array [ 1 .. 2 ] of line;
             graph_lines = array [ graph ] of line;
             comment_lines = array [ comment ] of line;
             graph_vectors = array [ graph ] of vector ;
             graph_reals = array [ graph ] of real ;

  type
     shortvector  = array [ 1 .. max_point ] of shortreal;
     graph_shortvectors = array [ graph ] of shortvector;
     type_of_graph = (  nothing, markers, joined, smooth, mark_joined,
                        mark_smooth );

  var
     together, comments_page          : boolean;
     current_graph,no_of_graphs       : graph;
     title                            : two_lines;
     current_comment,no_of_comments   : comment;
     comments                         : comment_lines;
     single_comment                   : line;
     current_point,no_of_points,
     start, last, first               : point;
     shortx_min,shortx_max            : shortreal;
     shorty_min_tog,shorty_max_tog    : shortreal;
     shorty_min,shorty_max            : array [ graph ] of shortreal;
     x_min, x_max                     : real;
     first_page        : boolean;
     y_min, y_max      : graph_reals;
     answer            : char;
     graph_type        : array[graph] of type_of_graph;
     x                 : vector;
     shortx            : shortvector;
     y                 : graph_vectors;
     shorty            : graph_shortvectors;
     x_axis,y_axis     : line;
     blank_line        : line;
     count,mark        : integer;
     line_count        : integer;
     y_label           : graph_lines;
     filein            : text;
```

```
%include readin.pro  { see fit.pas }
%include filein.pro  { see fit.pas }
%include wstr.pro    { see fit.pas }

  type shortarray = array[1 .. 100] of shortreal;

  procedure annotp ( const x_direct, y_direct : integer ); fortran;
                   { side of axis for annotation, 0=-ve, 1=+ve }
  procedure arc    ( const x_start, y_start, angle  : shortreal ); fortran;
                   { draw arc around positn, anti-clock }
  procedure arcell ( const x_st, y_st, ang, eccent : shortreal ); fortran;
                   { draw arc of an ellipse with eccentricity eccent }
  procedure axes; fortran;
                   { draw axes on the graph }
  procedure axessi ( const increment_x, increment_y : shortreal ); fortran;
                   { draw axes with fixed increments on the graph }
  procedure axexl ; fortran;
                   { draw logrithm x axis, linear y axis }
  procedure axexli ( const increment_y : shortreal ); fortran;
                   { draw logrithm x axis, linear y axis- fixed y incr. }
  procedure axexyl; fortran;
                   { draw logrithm x axis and y axis }
  procedure axeyl ; fortran;
                   { draw logrithm y axis, linear x axis }
  procedure axeyli ( const increment_x : shortreal ); fortran;
                   { draw logrithm y axis, linear x axis- fixed x incr. }
  procedure axnota ( const enable : integer ); fortran ;
                   { enables/disable axis/scale annotation }
  procedure axorig ( const x, y  : shortreal ); fortran;
                   { fixes the intersection point of the  axes  }
  procedure barcht ( const y_origin, bar_width : shortreal;
                     const x_posns, values : shortarray;
                     const start, stop : integer ); fortran;
                   { draws a histogram  -  see also histgm }
  procedure border; fortran;
                    { draws a border aroun vector window }
  procedure box ( const xmin, xmax, ymin, ymax : shortreal ); fortran;
                     { draws a box  .. }
  procedure cdefin ( const character_number : integer;
                     const specification_array : shortarray;
                     const spec   : integer  ); fortran;
                    { enables characters to be defined }
  procedure circle ( const radius : shortreal ); fortran;
                    { draws a circle around positn }
  procedure crlnfd; fortran;
                    { carriage return/ line feed }
  procedure cretrn; fortran;
                    { carriage return }
  procedure cspace ( const cxmin, cxmax, cymin, cymax : shortreal );fortran;
                    { defines character space for 'typewriter mode' }
  procedure ctrang ( const angle : shortreal ); fortran;
                    { orientation of characters within a string }
  procedure ctrfnt ( const character_set : integer ); fortran;
                    { font to be used }
  procedure ctrmag ( const size : integer ); fortran;
                    { character size in 0.001 * ND space units }
  procedure ctrobl ( const width : shortreal ); fortran;
```

```
                    { sets character width, default 1.0          }
  procedure ctrori ( const angle : shortreal ); fortran;
                    { orientation of string to be used  - default degrees }
  procedure ctrsiz ( const size : shortreal ); fortran;
                    { character size in vector space units }
  procedure curveo ( const xarray, yarray : shortvector;
                     const first, last : integer); fortran;
  procedure degree; fortran; { sets units to degrees }
{ procedure ellipse( const x_axis , y_axis  : shortreal ); fortran;
  }                { draw ellipse around current plot. pos }
  procedure frame; fortran;
                    { new sheet of paper }
  procedure gpstop ( const max_sheets : integer ); fortran;
                    { maximum number of graphs to be drawn }
  procedure grad; fortran; { sets  angular units to grads   }
  procedure gratic; fortran;
                    { draws graticule }
  procedure gratsi ( const interval_x, interval_y : shortreal ); fortran;
                    { draws graticule with fixed x and y intervals }
  procedure graxl ; fortran;
                    { draws graticule, x logrithm., y linear }
  procedure graxli ( const interval_y : shortreal ); fortran;
                    { draws graticule, x logrithm., y linear and fixed }
  procedure graxyl; fortran;
                     { draws graticule, x logrithm., y log. }
  procedure grayl ; fortran;
                    { draws graticule, y logrithm., y linear }
   procedure grayli ( const interval_x : shortreal ); fortran;
                    { draws graticule, y logrithm., y linear and fixed }
  procedure grend; fortran;
                     { last procedure to be called - closes pds }
  procedure histgm ( const x_origin, y_origin, bar_width : shortreal;
                     const values : shortarray;
                     const start, stop : integer ); fortran;
                   { draws a histogram, with bars up against each other }
  procedure hlinfd ( const number_of_half_lines : integer ); fortran;
                     { half line feeds }
  procedure hspace ( const number_of_half_spaces: integer ); fortran;
                     { typewriter spaces }
  procedure italic ( const enable : integer  ); fortran;
                     { enables ltalic characters }
  procedure join ( const x_pos, y_pos : shortreal ); fortran;
  procedure locate ( const x_pos, y_pos : shortreal ); fortran;
                     { defines a translation from positn to locate }
  procedure linefd (  const number_of_lines : integer ); fortran;
                     { line feeds }
  procedure map ( const xmin, xmax, ymin, ymax : shortreal ); fortran;
                {    - this area in vector space maps onto the area defined
                       in ND space by pspace }
  procedure mapfol; fortran;
                    { cancels mapping, and makes window like vector rectange }
  procedure mapxl ( const xmin, xmax, ymin, ymax : shortreal ); fortran;
                     {  - as map but x axis logarithm }
  procedure mapyl ( const xmin, xmax, ymin, ymax : shortreal ); fortran;
                     {  - as map but y axis logarithm }
  procedure mapxyl ( const xmin, xmax, ymin, ymax : shortreal ); fortran;
                     {  - as map but x axis  and y axis logarithm }
  procedure marker ( const mark : integer ); fortran;
```

```
   procedure mask ( const xmin, xmax, ymin, ymax : shortreal ); fortran;
                    { defines a mask - no plotting within it }
   procedure mskchr ( const enable : integer ); fortran;
                    { makes the mask affect characters }
{  procedure nscurv ( const xarray, yarray : shortvector;
                    const first, last : integer);   fortran;
 } procedure paper( const on_off : integer ); fortran;
                  { first procedure to be called - 1=on, 0=off }
{  procedure pcscen - see spcscn }
{  procedure pcsend - see spcsed }
   procedure piecht ( const x_origin, y_origin, radius : shortreal ;
                    const values: shortarray;
                    const number_of_sectors : integer ); fortran;
                   {draws a pie chart }
   procedure place  ( const x, y : integer   ); fortran;
                   { positions current character pointer (ccp) }
{  procedure plotcs - see spltcs }
   procedure plotnc ( const x, y : shortreal;
                    const character : integer ); fortran;
                   { write character at x,y (vector space) }
   procedure plotne ( const x, y : shortreal;
                    const number : shortreal;
                    const number_after_decimal_point : integer ); fortran;
                  { write a real number out d.pt. at x, y, e.g. 3.456E02 }
   procedure plotnf ( const x, y : shortreal;
                    const number : shortreal;
                    const number_after_decimal_point : integer ); fortran;
                    { write a real number out, d.pt. at x, y e.g. 345.6 }
   procedure plotni ( const x, y : shortreal;
                    const number : integer ): fortran;
                    { write an integer out }
   procedure positn ( const x, y : shortreal ); fortran;
   procedure pspace ( const pxmin, pxmax, pymin, pymax : shortreal );fortran;
                   {  defines the paper space in ND coord. - see map }
   procedure ptgraf ( const xarray, yarray : shortvector; const
                    first_point, last_point, plotchar : integer ); fortran;
                    { plot points and draws lines between them }
   procedure ptjoin ( const xarray, yarray : shortvector; const
                    first_point, last_point, plotchar : integer ); fortran;
                   { join the points, plotchar = -ve for closed curve }
   procedure ptplot ( const xarray, yarray : shortvector; const
                    first_point, last_point, plotchar : integer ); fortran;
                    { plot points, plotchar = 232 for +, etc. }
   procedure qadrnt; fortran; { sets units to quadrants }
   procedure radian; fortran; { sets units to radians }
   procedure rotate ( const angle : shortreal ); fortran;
                    { defines a rotation around positn fixed by positn }
   procedure scale  ( const x_scale, y_scale : shortreal ); fortran;
                    { defines a scale (enlargement) around current pl pt.}
   procedure scales; fortran;
                    { draws scales around the vector window }
   procedure scalsi ( const interval_x, interval_y : shortreal ); fortran;
                  { draws scales around the vector window, fixed intervals }
   procedure scarot ( const x_scale, y_scale, angle : shortreal ); fortran;
                    { defines a rotation aand a scaling }
   procedure scaxl ; fortran;
                    { draws scales, x, log and y  linear }
   procedure scaxli ( const interval_y : shortreal ); fortran;
```

```
                    { draws scales, x, log and y fixed, linear }
  procedure scaxyl; fortran;
                    { draws scales, x and y  log  }
  procedure scayl ; fortran;
                    { draws scales, y  log and x  linear }
  procedure scayli ( const interval_x : shortreal ); fortran;
                    { draws scales, y  log and x fixed, linear }
  procedure seccir ( const x_start, y_start, angle  : shortreal ); fortran;
                  { draw sector of a circle around positn, anti-clock }
  procedure secell ( const x_st, y_st, angle, eccen  : shortreal ); fortran;
                  { draw sector of an ellipse around positn, anti-clock }
{ procedure space  ( const number_of_spaces      : integer ); fortran;
       *****   space is a reserved word *****- use hspace }
                    { typewriter spaces }
  procedure spltcs ( const x,y : shortreal; const phrase : line;
                     const length : integer  ); fortran;
                       { calls  plotcs in Ghost-80 }
                       { types string from x,y }
  procedure spcscn ( const x,y : shortreal; const phrase : line;
                     const length : integer  ); fortran;
                       { calls  pcscen in Ghost-80 }
                       { types string centered on x, y }
  procedure spcsed (  const phrase : line;
                     const length : integer  ); fortran;
                      { calls  pcsend in Ghost-80 }
                      { types string right justified to x, y }
  procedure stcscn (  const phrase : line;
                     const length : integer  ); fortran;
                       { calls  tcscen in Ghost-80 }
                       { types string centered  on current char. pos }
  procedure stcsed (  const phrase : line;
                     const length : integer  ); fortran;
                      { calls  tcsend in Ghost-80 }
                      { types string right justified to current char. pos }
  procedure stypcs (  const phrase : line;
                     const length : integer  ); fortran;
                       { calls  typecs in Ghost-80 }
                       { types string from current char. pos }
  procedure tclipa ( const enable : integer ); fortran;
                   { means that windows and masks are transformed }
  procedure typenc ( const char : integer ); fortran;
                     { types  character at current char. pos. }
  procedure typene ( const number : shortreal;
                     const number_after_decimal_point : integer ); fortran;
                   { write a real number out d.pt. at c.c.p e.g. 3.456E02 }
  procedure typenf ( const number : shortreal;
                     const number_after_decimal_point : integer ); fortran;
                     { write a real number out, d.pt. at ccp  e.g. 345.6 }
  procedure typeni ( const number : integer ); fortran;
                     { write an integer out }
  procedure undlin ( const enable : integer  ); fortran;
                    { enables underlining       }
  procedure unmask ( const level : integer ); fortran;
                     { used to set mask level, and therefore free masks }
   procedure winchr ( const enable : integer ) ; fortran;
                   { makes window affect characters     }
  procedure window ( const wxmin, wxmax, wymin, wymax : shortreal);fortran;
                   { defines window in vector_sp. coord }
```

```
    procedure winfol: fortran;
                      { cancels window => = vect.space rectangle }
    procedure xaxis; fortran;
                     { draws x axis on the graph }
    procedure xaxisi ( const increment_x : shortreal ); fortran;
                     { draws x axis with fixed increments on the graph }
    procedure xaxisl: fortran;
                     { draws x axis for logrithm scale }
    procedure xscale: fortran;
                     { draws x scale round vector window }
    procedure xscali ( const increment_x : shortreal ); fortran;
                     { draws x scale with fixed increments on the graph }
    procedure xscall; fortran;
                     { draws x scale for logrithm scale }
    procedure xgrat; fortran;
                     { draws x graticule on graph }
    procedure xgrati ( const increment_x : shortreal ); fortran;
                     { draws x graticule with fixed interval }
    procedure xgratl; fortran;
                     { draws logrithmetic x graticule }
    procedure yaxis; fortran;
                     { draws y axis on the graph }
    procedure yaxisi ( const increment_y : shortreal ); fortran;
                     { draws y axis with fixed increments on the graph }
    procedure yaxisl; fortran;
                     { draws y axis for logrithm scale }
    procedure yscale; fortran;
                     { draws y scale round vector window }
    procedure yscali ( const increment_y : shortreal ); fortran;
                     { draws y scale with fixed increments on the graph }
    procedure yscall; fortran;
                     { draws y scale for logrithm scale }
    procedure ygrat; fortran;
                     { draws y graticule on graph }
    procedure ygrati ( const increment_y : shortreal ); fortran;
                     { draws y graticule with fixed interval }
    procedure ygratl; fortran;
                     { draws logrithmetic y graticule }

    procedure newpage;
{   ************************      NEWPAGE      ********************** }

        { draws border, comments and title for a graph }

       var
          present_comment   _: comment;
          counter       : integer;

       begin
         if ( not first_page ) then
                begin
                frame;
                unmask(0)
                end;
          first_page := false;
          pspace ( 0.05, 0.75, 0.0, 1.0);
          map ( 0.05, 0.75, 0.0, 1.0);
          window ( 0.05, 0.75, 0.0, 1.0);
```

```
          border;
          box ( 0.05, 0.75, 0.0, 1.0);
          box ( 0.45, 0.75, 0.7, 1.0);
          box ( 0.05, 0.45, 0.7, 0.943);
          cspace ( 0.05, 0.45, 0.7, 0.99);
          ctrmag (12);
          for counter := 1 to 2 do
             begin
             place ( 2, counter );
             stypcs ( title[counter], length(title[counter]) )
             end;
          cspace ( 0.05, 0.45, 0.65, 0.935);
          ctrmag (10);
          if (no_of_comments > 15) then ctrmag(9);
          if (no_of_comments > 17) then ctrmag(8);
          if (no_of_comments > 20) then ctrmag(7);
          for present_comment := 1 to no_of_comments do
             begin
             place ( 2, present_comment );
             single_comment := comments [present_comment];
             stypcs ( single_comment, length(single_comment) )
             end;
          ctrmag ( 12 );
          cspace ( 0.0 , 0.0 , 0.0, 0.0);
          spltcs ( 0.08, 0.68, y_axis, length(y_axis) );
          spltcs ( 0.5, 0.03, x_axis, length(x_axis) );
          mask ( 0.48, 0.75, 0.001,0.049 );
          mskchr (1);
          line_count := 1;
          mark    := 232;
          end;

    begin

{   ************************      READING DATA IN      ********************** }

       reset ( filein, 'unit=1' );
       reset ( input , 'interactive' );

       for count := 1 to line_length do
            blank_line[count] := ' ';

       read_file_in ( filein, title,  x_axis,  y_axis,  y_label,
                      no_of_comments,  no_of_graphs,  no_of_points,
                      comments,  x, y, x_min, x_max, y_min, y_max);

{   ***** funny business to plot different graphs *****

       for current_point := 1 to no_of_points do
         begin
         y[1,current_point] := y[1,current_point]/y[2,current_point];
         if (abs(y[1,current_point])>1) then
             y[1,current_point] :=   1 /y[1,current_point]
         end;
       y_max[1] := 1;
       y_min[1] := -1;
       no_of_graphs := 1;
  }
```

Listing of GRAPH.PLOT at 01:04:01 on MAY 13, 1987 for CCid=PHP7

```
{  ************************   CONVERSION TO SHORT  ********************** }

      { convert to short real to call ghost package  }
      shortx_max := x_max;
      shortx_min := x_min;
      for current_graph := 1 to no_of_graphs do
         begin
         shorty_max[current_graph] := y_max[current_graph];
         shorty_min[current_graph] := y_min[current_graph]
         end;

      for current_point := 1 to no_of_points do
         begin
         shortx[current_point] := x[current_point];
         for current_graph := 1 to no_of_graphs do
          shorty[current_graph,current_point]:=y[current_graph,current_point];
         end;


{  ************************   ASKING OPTIONS        ********************** }


      writeln ( ' GRAPH PLOTTING - using GHOST 80 ' );
      writeln ( ' ---------------- ' );
      writeln;
      writelnstr ( output, title[1] );
      writelnstr ( output, title[2] );
      writeln ( no_of_graphs, ' graphs read in ');
      writeln;

      writeln ( ' Do you want the graphs on one sheet or separate ? (o/s)');
      readln (  answer );
      if (( answer='o' ) or ( answer='O')) then together := true
                                          else together := false;
      writeln ( 'What do you want for each graph ');
      writeln ( '       N - nothing ');
      writeln ( '       M - mark    ');
      writeln ( '       J - joined points  ');
      writeln ( '       S - smooth line    ');
      writeln ( '       B - both marks and a smooth line ');
      writeln ( '       L - markers joined up with straight lines ');
      for current_graph := 1 to no_of_graphs do
          begin
          writelnstr ( output, y_label[current_graph] );
          readln (  answer );
          case answer of
            'N','n' : graph_type[current_graph] := nothing;
            'M','m' : graph_type[current_graph] := markers;
            'J','j' : graph_type[current_graph] := joined ;
            'S','s' : graph_type[current_graph] := smooth ;
            'B','b' : graph_type[current_graph] := mark_smooth ;
            'L','l' : graph_type[current_graph] := mark_joined ;
            otherwise   graph_type[current_graph] := nothing;
            end;
          end;

{  ************************     GRAPH   PLOTTING   ********************** }
```

```
paper(1);
first_page := true;
if (together) then
 begin
   newpage;
   shorty_min_tog := 9999999;
   shorty_max_tog := -9999999;
   for current_graph := 1 to no_of_graphs do
      if ( graph_type[current_graph] <> nothing ) then
         begin
         if ( shorty_min [current_graph] < shorty_min_tog )
            then  shorty_min_tog:= shorty_min[current_graph];
         if ( shorty_max [current_graph] > shorty_max_tog )
            then  shorty_max_tog:= shorty_max[current_graph]
         end;
   for current_graph := 1 to no_of_graphs do
         begin
         shorty_min[current_graph] := shorty_min_tog;
         shorty_max[current_graph] := shorty_max_tog
         end;
    pspace ( 0.05, 0.75, 0.0, 0.7);
    map (1.12*shortx_min-0.12*shortx_max,
    1.12*shortx_max-0.12*shortx_min,
    1.12*shorty_min_tog-0.12*shorty_max_tog,
    1.12*shorty_max_tog-0.12*shorty_min_tog);
    window (1.02 *shortx_min-0.02*shortx_max,
    1.02*shortx_max-0.02 *shortx_min,
    1.02*shorty_min_tog-0.02*shorty_max_tog,
    1.02*shorty_max_tog-0.02*shorty_min_tog);
    scales;
    border;
    if (shortx_min*shortx_max < 0.0 ) then
         begin
         axnota (0);
         yaxis;
         axnota (1)
         end;
    if (shorty_min_tog*shorty_max_tog < 0.0 ) then
         begin
         axnota (0);
         xaxis     ;
         axnota (1)
         end
    end;

for current_graph :=_1 to no_of_graphs do
  if ( graph_type[current_graph] <> nothing ) then
    begin
    {  ******  calc of limits  ******   }
    start := 1;
    while ( ( abs(y [ current_graph, start]+999)<0.001 )
       and (start< no_of_points) )  do start:=start+1;
    last:= start;
    while ( ( abs(y [ current_graph, last] + 999) > 0.001 )
       and (last < no_of_points) )  do last:=last+1;
    if   ( not together )  then  newpage;
    pspace ( 0.50, 0.73, 0.7, 0.98);
    cspace ( 0.47, 0.73, 0.7, 0.985);
```

```
            place ( 1, line_count );
            if ( ( graph_type[current_graph] = joined ) or
                   ( graph_type[current_graph] = smooth ) )
               then typenc ( 45 )
               else begin
                  typenc ( mark );
                  marker ( mark );
                  mark := mark + 1;
                  if ( mark = 233 ) then mark := 235
                  end;
            cspace ( 0.49, 0.745, 0.7, 0.985);
            place ( 1, line_count );
            stypcs ( y_label[current_graph], length(y_label[current_graph]));
            count := line_length;
            while ((y_label[current_graph,count]=' ') and (count>1)) do
                     count := count - 1 ;
            line_count := line_count + ((count+25) div 25);
            pspace ( 0.05, 0.75, 0.0, 0.7);
            map (1.12*shortx_min-0.12*shortx_max,
            1.12*shortx_max-0.12*shortx_min,
            1.12*shorty_min[current_graph]-0.12*shorty_max[current_graph],
            1.12*shorty_max[current_graph]-0.12*shorty_min[current_graph]);
            window (1.02 *shortx_min-0.02*shortx_max,
            1.02*shortx_max-0.02 *shortx_min,
            1.02*shorty_min[current_graph]-0.02*shorty_max[current_graph],
            1.02*shorty_max[current_graph]-0.02*shorty_min[current_graph]);
            if ( not together ) then
                begin
                scales;
                border;
                if ( shortx_max*shortx_min < 0.0 ) then
                     begin
                     axnota ( 0 );
                     yaxis ;
                     axnota ( 1 )
                     end;
                if ( shorty_max[current_graph]*shorty_min[current_graph]<0.0)
                     then begin
                     axnota ( 0 );
                     xaxis  ;
                     axnota ( 1 )
                     end
                end;
             if ( ( graph_type[current_graph] = markers ) or
                  ( graph_type[current_graph] = mark_joined ) or
                  ( graph_type[current_graph] = mark_smooth ) ) then
                  ptplot ( shortx, shorty[current_graph],start,last,0);
             if ( ( graph_type[current_graph] = joined ) or
                  ( graph_type[current_graph] = mark_joined ) )  then
                begin
                first:=start;
                current_point:=start;
                while current_point< last  do
                   begin
                   current_point:=first;
                   while current_point < last  do
                      begin
                      current_point := current_point + 1;
```

```
                    if ((abs ( shortx[current_point]
                            - shortx[current_point - 1] )
                            >((shortx_max - shortx_min )/2))
                            or(abs ( shorty[current_graph , current_point]
                          - shorty[current_graph , current_point - 1] )>
                            ((shorty_max [current_graph]
                                - shorty_min [current_graph])/2)))    then
                        begin
                        ptjoin ( shortx, shorty[current_graph], first,
                                            current_point - 1, 0);
                        first := current_point;
                        end;
                    end;
                end;
            ptjoin ( shortx, shorty[current_graph],first,last,0);
            end;
        if ( ( graph_type[current_graph] = smooth  ) or
             ( graph_type[current_graph] = mark_smooth ) ) then
            begin
            first:=start;
            current_point:=start;
            while current_point< last  do
               begin
               current_point:=first;
               while current_point < last  do
                  begin
                  current_point := current_point + 1;
                  if ((abs ( shortx[current_point]
                          - shortx[current_point - 1] )
                          >((shortx_max - shortx_min )/2))
                          or(abs ( shorty[current_graph , current_point]
                        - shorty[current_graph , current_point - 1] )>
                          ((shorty_max [current_graph]
                              - shorty_min [current_graph])/2)))    then
                      begin
                      curveo ( shortx, shorty[current_graph], first,
                                          current_point - 1);
                      first := current_point;
                      end;
                    end;
                end;
            curveo ( shortx, shorty[current_graph], first, last );
            end;
    end;
grend;
end.
```

```
      SUBROUTINE SPLTCS( X, Y, STRING, LENGTH)
      REAL  X, Y, XO, YO
      INTEGER*4 COUNT, LENGTH, TMP1, TMP2
      INTEGER*4 STRING(LENGTH)
      CHARACTER*80 LINE
C
      TMP1=0
      DO 10 COUNT=1, LENGTH
         IF (TMP1 .NE. ((COUNT-1)/4+1)) THEN
            TMP1=(COUNT-1)/4+1
            TMP2=STRING( TMP1 )
         ENDIF
      LINE ( COUNT:COUNT )=CHAR( ISHFT(IAND(TMP2,
     *                        -16777216), -24) )
      TMP2=ISHFT( TMP2, 8 )
10    CONTINUE
C
      CALL PLOTCS (X, Y, LINE(1:LENGTH))
      RETURN
      END
C
C
C
      SUBROUTINE SPCSCN( X, Y, STRING, LENGTH)
      REAL  X, Y, XO, YO
      INTEGER*4 COUNT, LENGTH, TMP1, TMP2
      INTEGER*4 STRING(LENGTH)
      CHARACTER*80 LINE
C
      TMP1=0
      DO 10 COUNT=1, LENGTH
         IF (TMP1 .NE. ((COUNT-1)/4+1)) THEN
            TMP1=(COUNT-1)/4+1
            TMP2=STRING( TMP1 )
         ENDIF
      LINE ( COUNT:COUNT )=CHAR( ISHFT(IAND(TMP2,
     *                        -16777216), -24) )
      TMP2=ISHFT( TMP2, 8 )
10    CONTINUE
C
      CALL PCSCEN (X, Y, LINE(1:LENGTH))
      RETURN
      END
C
      SUBROUTINE SPCSED( X, Y, STRING, LENGTH)
     _REAL  X, Y, XO, YO
      INTEGER*4 COUNT, LENGTH, TMP1, TMP2
      INTEGER*4 STRING(LENGTH)
      CHARACTER*80 LINE
C
      TMP1=0
      DO 10 COUNT=1, LENGTH
         IF (TMP1 .NE. ((COUNT-1)/4+1)) THEN
            TMP1=(COUNT-1)/4+1
            TMP2=STRING( TMP1 )
         ENDIF
      LINE ( COUNT:COUNT )=CHAR( ISHFT(IAND(TMP2,
     *                        -16777216), -24) )
```

```
            TMP2=ISHFT( TMP2, 8 )
10          CONTINUE
C
            CALL PCSEND (X, Y, LINE(1:LENGTH))
            RETURN
            END
C
C
C
C
            SUBROUTINE STYPCS( STRING, LENGTH)
            INTEGER*4 COUNT, LENGTH, TMP1, TMP2
            INTEGER*4 STRING(LENGTH)
            CHARACTER*80 LINE
C
            TMP1=0
            DO 10 COUNT=1, LENGTH
               IF (TMP1 .NE. ((COUNT-1)/4+1)) THEN
                   TMP1=(COUNT-1)/4+1
                   TMP2=STRING( TMP1 )
               ENDIF
            LINE ( COUNT:COUNT )=CHAR( ISHFT(IAND(TMP2,
     *                          -16777216), -24) )
            TMP2=ISHFT( TMP2, 8 )
10          CONTINUE
C
            CALL TYPECS ( LINE(1:LENGTH))
            RETURN
            END
C
C
C
            SUBROUTINE STCSCN( STRING, LENGTH)
            INTEGER*4 COUNT, LENGTH, TMP1, TMP2
            INTEGER*4 STRING(LENGTH)
            CHARACTER*80 LINE
C
            TMP1=0
            DO 10 COUNT=1, LENGTH
               IF (TMP1 .NE. ((COUNT-1)/4+1)) THEN
                   TMP1=(COUNT-1)/4+1
                   TMP2=STRING( TMP1 )
               ENDIF
            LINE ( COUNT:COUNT )=CHAR( ISHFT(IAND(TMP2,
     *                          -16777216), -24) )
            TMP2=ISHFT( TMP2, 8 )
10          CONTINUE
C
            CALL TCSCEN ( LINE(1:LENGTH))
            RETURN
            END
C
C
C
            SUBROUTINE STCSED( STRING, LENGTH)
            INTEGER*4 COUNT, LENGTH, TMP1, TMP2
            INTEGER*4 STRING(LENGTH)
            CHARACTER*80 LINE
```

```
C
            TMP1=0
            DO 10 COUNT=1, LENGTH
                IF (TMP1 .NE. ((COUNT-1)/4+1)) THEN
                    TMP1=(COUNT-1)/4+1
                    TMP2=STRING( TMP1 )
                ENDIF
            LINE ( COUNT:COUNT )=CHAR( ISHFT(IAND(TMP2,
           *                        -16777216), -24) )
            TMP2=ISHFT( TMP2, 8 )
10          CONTINUE
C
            CALL TCSEND ( LINE(1:LENGTH))
            RETURN
            END
```

Appendix 3: Switched Mode power supply.

In order to control the current to the cross coils easily a dedicated twin power supply was built. The supply was designed to work with the Minicam digital to analogue converters, which can be controlled from the micro. It can also be used a a simple power supply with manual controls. Only a basic voltage indication is given, as the supply was designed to be used in conjunction with computerized instrumentation. The circuit diagram is given in fig A3.1, although this is doubled up to provide a twin supply. A 5V auxiliary supply is also available, and this was normally used to power a calculator.

The specifications are as follows:

Maximum Output Voltage	30V
Maximum Output Current	1A (0.6A fuse)
Supply Voltage	250V (or 110V)
Programming voltage	0-400mV
Regulation	0.5%

Description of the circuit

The switched mode controller regulates the frequency of switching to provide the constant output voltage, the frequency, and therefore the level of this output voltage being determined by comparison of the output voltage with a reference voltage.

A transformer provides the basic AC supply at about 35V. This is then rectified in the normal way and is smoothed by C_1. R_1 draws a basic current from the supply.

C_2 and R_4 determine the timing of the switched mode supply.

C_3, C_4, C_6 and C_7 are decoupling capacitors to prevent 'ringing'.
R_2 prevents too large a surge current on switching the transistor T_1 which is switched by the switching IC, 78S40. R_3 provide the correct base current for this transistor.

The Inductor, L_1 and the diodes, D_1 - D_3, mean that the output current can be larger than the supply current, the extra being drawn up through these diodes.

S_1 selects between local control and remote control.

In remote control the supply is controlled by a control voltage which provides, through R_7 to R_9 a reference voltage for one side of the comparator. The other side is provided by the potential divider consisting of P_1, R_5 and R_6 on the output voltage. Adjusting this trimmer(P_1) alters the output voltage range for a given input range.

In local control the potential divider on the output voltage is made up of the potentiometers, P_2(coarse) and P_3(fine), which are mounted on the front of the supply, and R_{10}. In this local position the reference voltage is the 1.3V reference from the 78S40. C_5 provides the final smoothing for the supply.

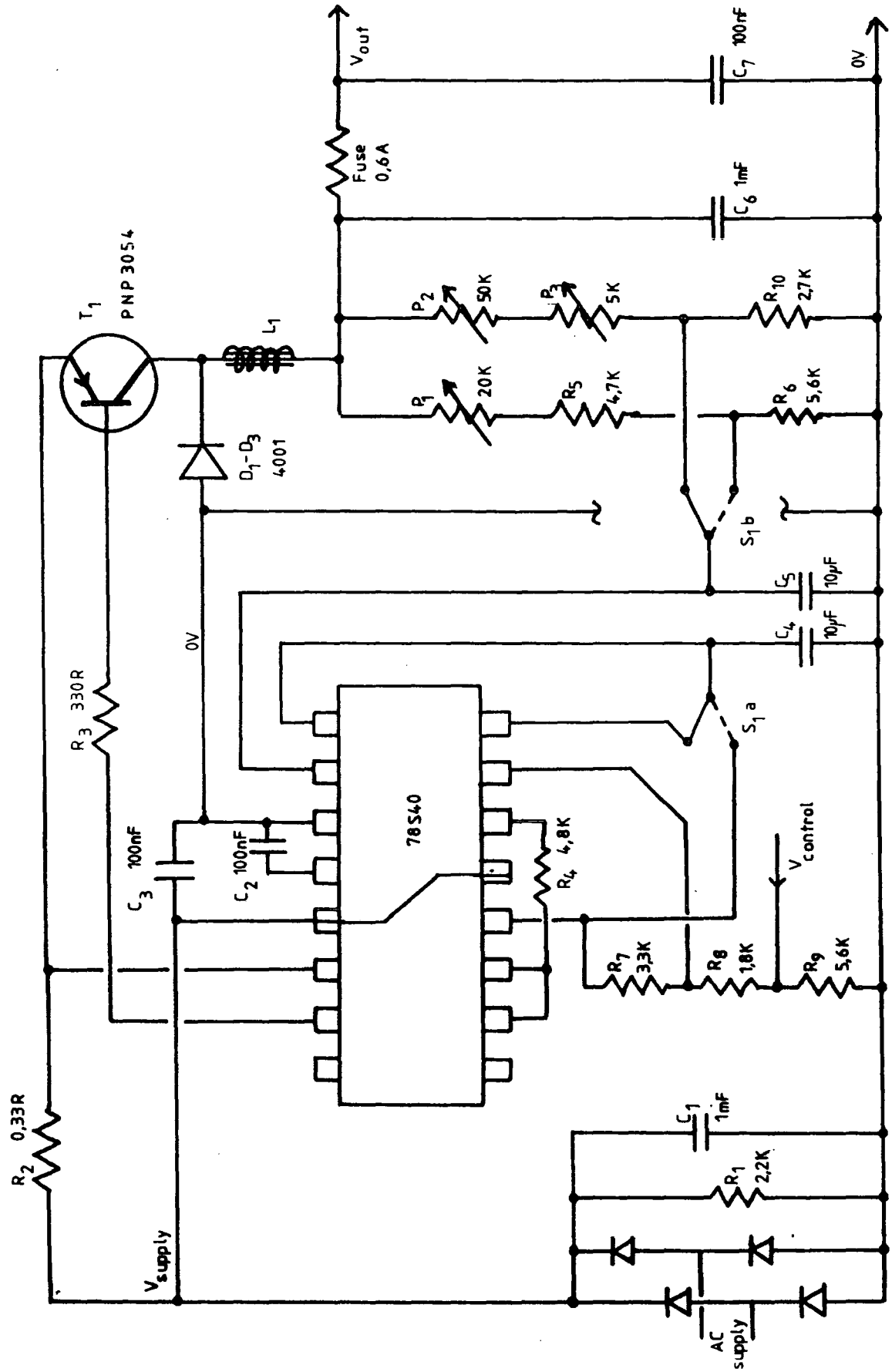


Fig A3.1 Switched mode power supply

Appendix 4: Relay board for Minicam interface.

In order to reverse the polarity on the power supply a relay board was designed and built for the Minicam interface. Minicam is a commercially built interface, available from Bede Scientific Equipment in Coxhoe, County Durham. It is of modular construction, with a basic 'crate' consisting of a power supply, giving +5V and +/- 15V to rails along the back. A Controller board plugs in one slot, with either an RS232 or IEEE488 interface or a direct memory map interface for a Commodore Pet. The Controller has an on board micro and controls data and address lines along the crate. Various boards are available to plug into the crate, but none were able to do the job of switching the power supply. In order to do this a relay board was designed, and this is now marketed as an option for the interface.

A circuit diagram is given in fig A4.1.

The board consists of address decoders, 4514 and 4515, and the address is selected with wire links to these decoders. An octal latch and Darlington driver chip, 5801A, switches the current for the eight relays. PB8402 relays are used and a 12 ohm resistor drops the 15V supply to drive these 12V relays. A reset button acts on the latch in the 5801A to unset all the relays. and a small capacitor ensures that the relays are not powered when the Minicab is switched on.

In order to accommodate the board the Minicam power supply was uprated, to provide a maximum current of 5A on the 5V rail and a total of 8 Amps on the 5V and +15V rails.

The relay board uses the same Minicam software as the digital to analogue converter, with the relays relating to the eight least significant binary digits. For example 6, which is binary 00000110, would switch relays 2 and 3 on and leave the rest off.

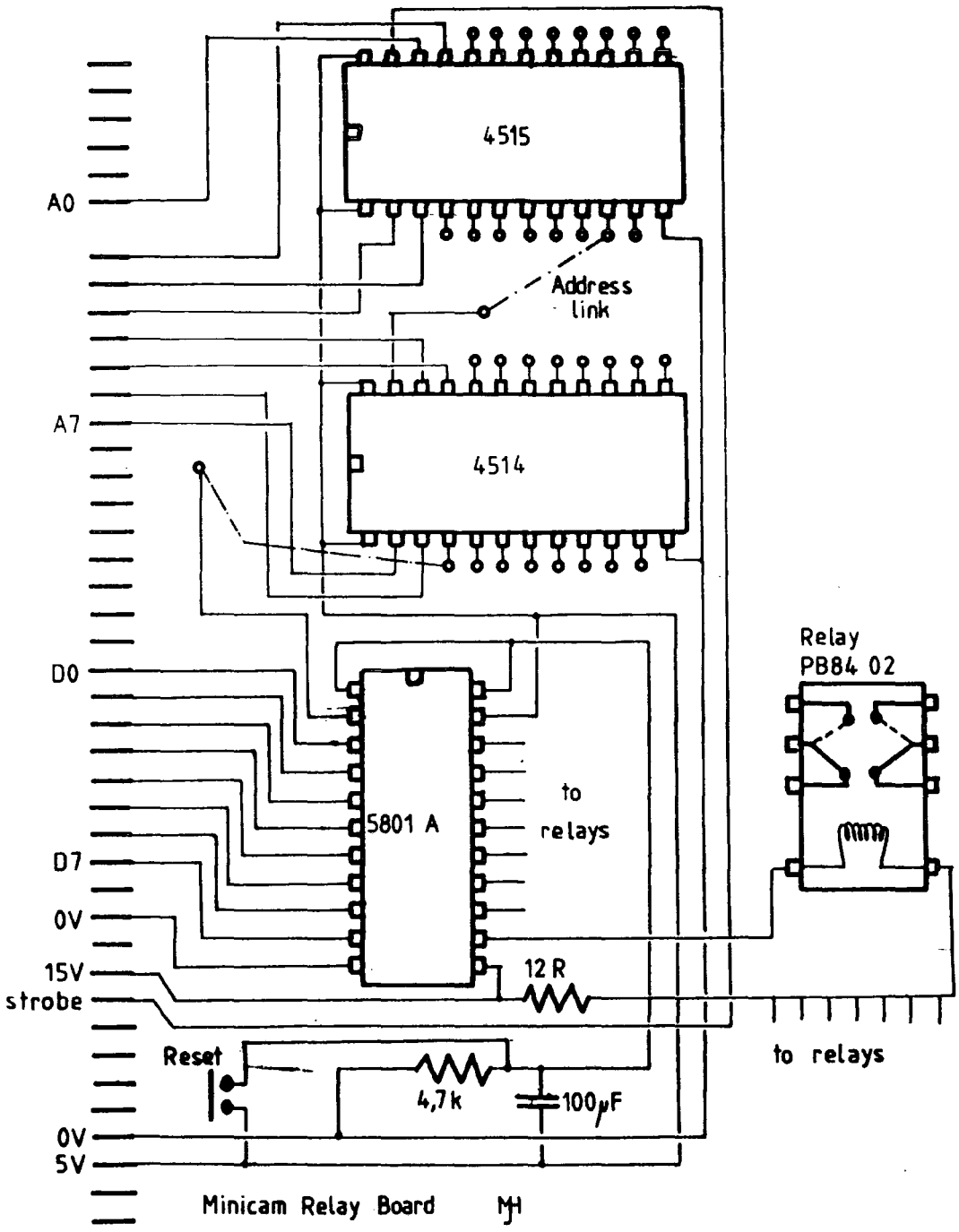


Fig A4.1 Minicam Relay Board

Appendix 5: RS232 Multiplexor for BBC micro

A5.1 Introduction

In order to use more than one instrument from a micro either a number of communication ports are required, or a sophisticated communications bus or ring system with every instrument on the bus or ring conforming to an agreed protocol.

The standard laboratory interfaces are the RS232 and the IEEE488.The former is a serial interface, and is defined primarily for communication between a terminal and a main frame computer. It has however been adopted by a large number of instrument and micro manufacturers as a laboratory standard. It is a one-to-one communications interface and to use one micro with more than one instrument requires either a number of RS232 ports on the micro, a switch of some sort or pulling connectors in an out.

The IEEE488 standard defines a bus upon which up to 32 instruments can be connected. The IEEE488 standard is very extensive and when fully implemented provides an extremely fast and efficient interface. There are two problems with this interface, the instruments and the controller. Very few systems implement the complete standard, and those that do are expensive. A system will therefore normally consist of a number of instruments with some of the protocol implemented and a controller with some, usually a different part, of the protocol implemented. Great care has to be taken to ensure that the protocol required for efficient communication is available.

In order to attach the instruments required for this experiment to a BBC micro it was decided to build a four way RS232 multiplexor which could be

controlled by the micro. This means that the BBC has effectively four RS232 ports, one port being selectable at a time.

A5.2 Design (see Fig A5.1)

The multiplexor is run from the BBC user port, which is connected internally to a 6522 Versatile Interface Adaptor (VIA). The user port provides up to 8 TTL compatible 0V or 5V lines which can be used either as logic input or output. Two of these lines are used as output from the micro to control the switch. Any system which provides a 5V logic signal can be used to switch the multiplexor. The user port also has a 5V power connection which is used to power the switch.

The user port lines go to a switch(SW1), which gives the choice of using the multiplexor as a manual switch or under remote control by the user port lines. The manual logic signals are obtained from a couple of two-way switches(SW2/3). The logic signals then go to buffers consisting of a 741 op-amp which provide both a buffered 5v logic signal and a 15V signal.
The 5V drives a 74139 2 bit decoder, which illuminates display LED's to indicate which line is connected.
The 15V drives a bank of AD7592 analogue switches. These have a very fast switching time (<1μs), and are therefore effectively instantaneous for the micro. Reed relays switch in 2-3ms and could therefore be caught out, if machine code routines were being used. The analogue switches switch the four lines from the BBC RS423 port, and sockets are provided which are compatible with the Clearway installed in the laboratory.

The power is taken from the user port, and a DC-DC converter supplies the ±15V needed for the 741s and the AD7592s. Loading resistors are needed on this converter to enable it to work smoothly.

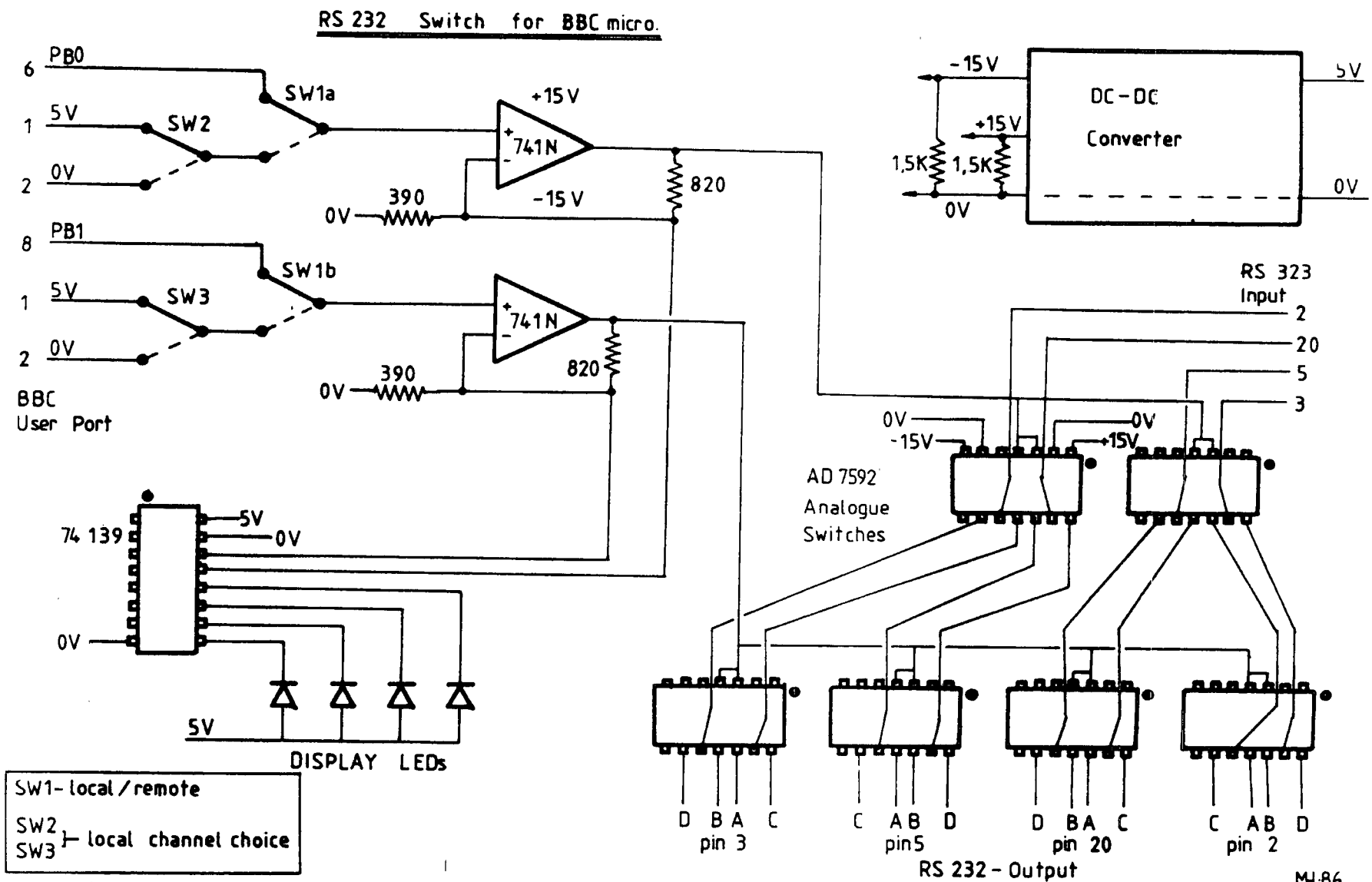


Fig A5.1 RS232 switch for BBC micro

Appendix 6: Relationship between Fourier and Anisotropy constants.

Introduction

When fitting coefficients using an iterative least squares method a formulation in terms of orthogonal coefficients is much preferred, both in allowing the method to converge and in preventing the trapping in 'false' local minima.

THe non-orthogonality of the standard anisotropy constants presents a problem, but this is easily resolved by fitting with Fourier coefficients and then calculating the anisotropy constants from these. The relationships between the first five Fourier coefficients of a torque curve and the anisotropy constants for a magnetically uniaxial system are given below. These can be easily computed for any other system by fourier analysing the expression for the torque, and inverting the derived matrix. If an inverse exists then the treatment is valid

The anisotropy energy of a uniaxial system can be expressed as:

$$E = K_1\sin^2(\theta) + K_2\sin^4(\theta) + K_3\sin^6(\theta) + K_4\sin^8(\theta) + K_5\sin^{10}(\theta) + \ldots \quad \text{(A6.1)}$$

Differentiating this gives a torque of:

$$T = K_1\sin(2*\theta) + 4*K_2\sin^3(\theta)\cos(\theta) + 6*K_3\sin^5(\theta)\cos(\theta) + 8*K_4\sin^7(\theta)\cos(\theta) + 10*K_5\sin^9(\theta)\cos(\theta) + \ldots \quad \text{(A6.2)}$$

This can be expressed in terms of Fourier coefficients:

$$T = F_2\sin(2*\theta) + F_4\sin(4*\theta) + F_6\sin(6*\theta) + F_8\sin(8*\theta) + F_{10}\sin(10*\theta) \quad \text{(A6.3)}$$

Equating A6.2 and A6.3 gives equations:

$$\begin{pmatrix} F_2 \\ F_4 \\ F_6 \\ F_8 \\ F_{10} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0.9375 & 0.875 & 0.8203125 \\ 0 & -0.5 & -0.75 & -0.875 & -0.9375 \\ 0 & 0 & 0.1875 & 0.375 & 0.52734375 \\ 0 & 0 & 0 & -0.0625 & -0.15625 \\ 0 & 0 & 0 & 0 & 0.01953125 \end{pmatrix} \begin{pmatrix} K1 \\ K2 \\ K3 \\ K4 \\ K5 \end{pmatrix} \quad \textbf{(A6.4)}$$

and inversely:

$$\begin{pmatrix} K1 \\ K2 \\ K3 \\ K4 \\ K5 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 0 & -2 & -8 & -20 & -40 \\ 0 & 0 & 16/3 & 32 & 112 \\ 0 & 0 & 0 & -16 & -128 \\ 0 & 0 & 0 & 0 & 51.2 \end{pmatrix} \begin{pmatrix} F2 \\ F3 \\ F4 \\ F6 \\ F8 \end{pmatrix} \quad \textbf{(A6.5)}$$

these formulation are used in procedure 'fit' in the PLOTFIT program (Appendix 2).

Appendix 7: Relationship between torque curves and Anisotropy constants.

Introduction

If torque measurements are made then they are most accurate at or around a low torque value, near the easy direction, as in this region the sample is most likely to be saturated. In a simple uniaxial system this means that the first anisotropy constant can be derived from the slope at the easy direction as it is the only non zero term. In the following we assume that the system can be modelled in terms of the first two anisotropy constants. The energy can then be expressed as:

$$E = K_1\sin^2(\theta) + K_2\sin^4(\theta) + \ldots \qquad \text{(A7.1)}$$

Differentiating this gives a torque of:

$$T = 2K_1\sin(\theta)\cos(\theta) + 4K_2\sin^3(\theta)\cos(\theta) + \ldots \qquad \text{(A7.2)}$$

This is only equal to zero if:

$$\sin(\theta) = 0 \qquad \text{(A7.3a)}$$

or $$\cos(\theta) = 0 \qquad \text{(A7.3b)}$$

or $$2K_1 + 4K_2\sin^2(\theta) \qquad \text{(A7.3c)}$$

Equation A7.3a is the position of the uniaxial axis, A7.3b is that of the basal plane and A7.3c is the position of the easy cone in cases where this occurs. Looking at the slope of the torque curve:

$$dT/d\theta = 2K_1\cos^2(\theta) - 2K_1\sin^2(\theta) + 12K_2\sin^2(\theta)\cos^2(\theta) + 4K_2\sin^4(\theta) \qquad \text{(A7.4)}$$

In the case of an easy axis (e.g. terbium and $Nd_2Fe_{14}B$ at high temperature) the slope at $\theta=0$ is given by:

$$dT/d\theta_{(\theta=0)} = 2K_1 \qquad \text{(A7.5)}$$

When an easy cone has developed then equation A7.5 is still valid, but as the uniaxial axis is now hard the measurement of the slope is less accurate, the slope at the easy direction is given by substituting A7.3c into A7.4:

$$dT/d\theta_{(\theta=\text{easy cone})} = -4K_1\cos^2(\theta) \qquad \text{(A7.6a)}$$

or

$$dT/d\theta_{(\theta=\text{easy cone})} = 8K_2\cos^2(\theta)\sin^2(\theta) \qquad \text{(A7.6b)}$$

Thus the first two anisotropy constants can be derived from measuring the slope at the easy direction and the separation of the easy directions.

Appendix 8 Shear Correction in Torque magnetometry.

When an angle is measured in any magnetic measurements then it is usually an angle between two simple obvious physical directions. In torque magnetometry the angle measured is between the field direction and an easily recognisable crystal axis. However the angle at which the moment lies within the sample is neither of these and this leads to the so-called 'shear' correction which has to be applied to the angle.

In figure A8.1 the situation is shown. The measured torque is a function of the angle that the moment subtends to the crystal axis, θ_1, and it is the form of this function that is of interest to measure. However the angle measured is $\theta_1 + \theta_2$. However the torque can also be seen to arise from the interaction of the moment with the external field, i.e.

$$\mathbf{T} = \mathbf{M} \times \mathbf{B} \qquad \text{(A8.1)}$$

or

$$T = M B \sin(\theta_2) \qquad \text{(A8.1b)}$$

therefore the measured angle has to be corrected by this θ_2 which can be calculated once the torque is measured if the field and the magnitude of the magnetic moment are known. As the correction is usually small, it is usually applied in the appoximate form:

$$\theta_2 = T / (M B) \qquad \text{(A8.2)}$$

and is therefore known as a shear correction as the correction to θ is proportional to T.

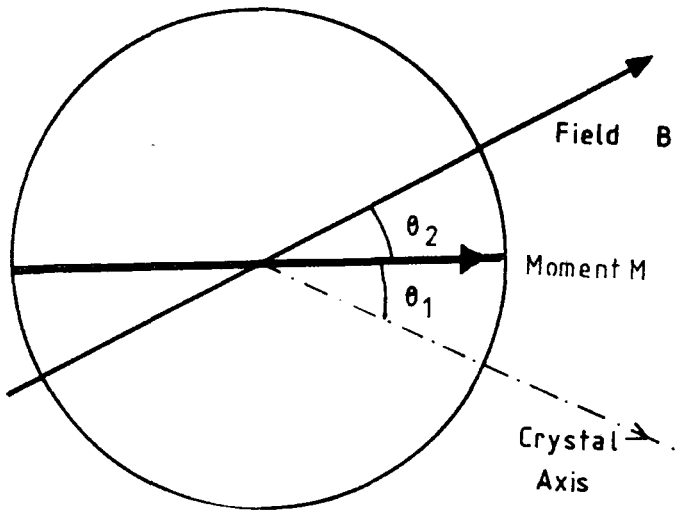


Fig A8.1 Moment in an applied field.