ERRATA and additional comment on 'Software Correction of Astronomical Telescope Pointing Errors' ...... May 1974.

PAGE	LINE	SHOULD READ
iv	6 from bottom	Appendix E
1.3	5 from bottom	Mortara (1969)
1.6	2 from bottom	88-inch Mauna Kea
1.12	1	Mortara (1969)
1.13	6 from bottom	88-inch Mauna Kea
1.14	17	while A.A.T. Project planners intend to ..
1.15	20	dis-assembly of,
2.1	10	and $\underline{x}_i = (x_{qi})$, ...
2.19	13 from bottom	R.M.S. error of 13.7 arcsecond ......
F.4	reference 2 from bottom	MORTARA, L.B.
F.6	reference 12	(SYDENHAM, P.H. should include the source) Proc. I.E.E., Vol. 115, No. 7, p.1056, July 1968.

Additional Comment (N.B. the relevant page is given in parentheses).

(1.2,1.3) The argument of the first 3 lines of page 1.3 is somewhat specious; radio source position accuracy could, perhaps, set pointing accuracy goals for optical instruments but not for steerable radio instruments.

(1.14) In addition to the treatments of software pointing correction by Meeks et al and Minnett et al, the author's attention has been brought to CLARK B.G. 'V.L.A. Telescope Pointing Analysis;' V.L.A. Computer Memorandum No. 104, N.R.A.O. June 28th 1973. This memorandum describes a proposed system of pointing corrections for interferometer antennas involving least-squares fitting of azimuth/elevation pointing data for the parameters of a linear model describing axis tilt, zero offsets, collimation error, secondary structure sag (sine of zenith angle) and receiver feed location. As in the author's own work the R.M.S. pointing error is minimized and proposal (similar to some of those in Chapter 5) given for automatic updating of pointing data.

(3.18)	'Composite fitting,' that is the use of an initial model fit followed by a surface fit, was not attempted with the pointing data from the Mt. Stromlo 74-inch since most of the improvement obtained from the model fitting was (in this case) due to parameters (like encoder offsets) which are also involved in a surface fit.
(5.3 point iii)	Rejection of data points which appear to be spurious with respect to a trend must depend upon the total number of points and the statistical nature of their distribution. The advice to reject all those displaced by more than twice the standard deviation is wrong and not consistent with the preceeding discussion of outlier rejection. It should also be mentioned that fitting with the L_1 norm may be more useful for locating outliers than least-squares fitting.
(5.5,5.15)	Consideration could be given to improved methods of storing the results of a fit which would facilitate the evaluation of such fits when using <u>small</u> computers. Further work in this area should include a reappraisal of the storage-requirements/numerical accuracy compromise mentioned by Cadwell and Williams (1961).
(5.5 footnote)	With regard to the number of bits precision required for various calculations, it should be noted that where angles (and simple trigonometric functions of them which are also periodic) alone are involved, the use of double length integer representation (32 bit two's-complement) on small 16-bit machines provides adequate accuracy and also high speed.
(A.6)	Slight inconsistencies exist in the accuracy to which the various correction formulas in Appendix A are given. The elliptic E-terms are usually less significant than the correction (in aberration) for the barycentre of the solar system, and also the additional term in the expression for z_3' in equation A.13, neither of which are given.

The author is indebted to Dr. W.N. Brouw, Dr. A.A. Hoag and Dr. M.R. Osborne for discussion concerning the above points.

G R Hovey

G.R. HOVEY.

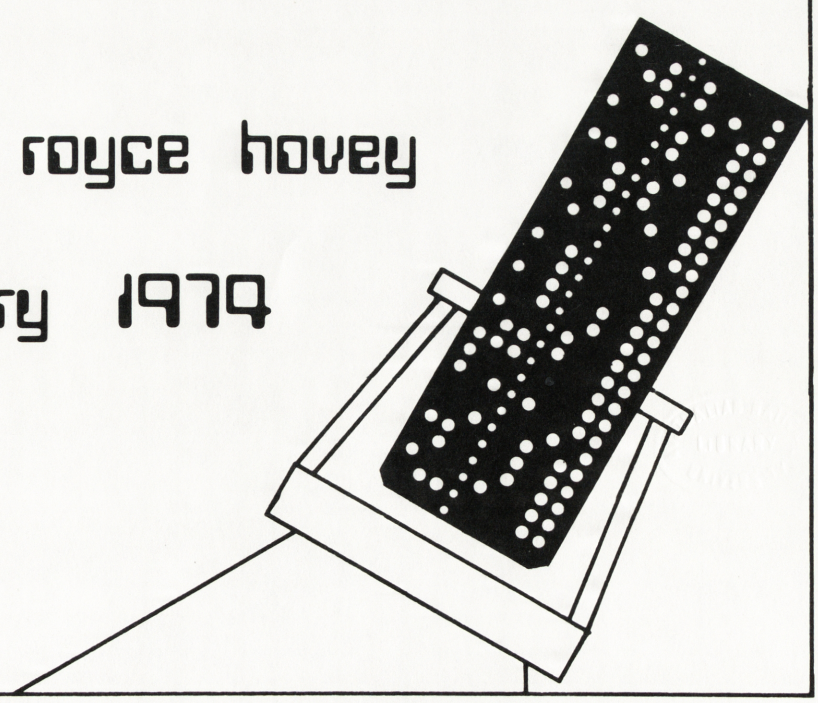
May, 1974.

SOFTWARE CORRECTION of ASTRONOMICAL TELESCOPE POINTING ERRORS

...submitted for the degree of
Doctor of Philosophy
of the
AUSTRALIAN NATIONAL UNIVERSITY
by
GARY ROYCE HOVEY
.......January 1974.........

gary royce hovey

january 1974



This thesis is entirely my own original work, with the exception of two computer programs and an experiment, the acknowledgement for which is given in accompanying footnotes. It has not previously been submitted for a degree at this, or any other university.

G R Hovey

ABSTRACT

This thesis is concerned with the pointing errors of astronomical telescopes, and examines means for their reduction which do not involve physical modifications to the instrument itself. The current trends in the engineering design of telescopes, which are relevant to pointing performance, are discussed in the introductory chapter, which contains a comprehensive review of the literature on the subject. The problem is, having sampled the pointing errors of an instrument at various points on the sky, to devise a numerical approximation to that pointing error data, which will enable the prediction of the error at a desired point of observation. Two distinct approaches are possible: model fitting when the causes of pointing error are known and quantifiable, and surface fitting which is more general. In Chapter 2, various algorithms for estimating the parameters in (nonlinear) models of the pointing error are investigated using data generated synthetically from a simple, but representative, error model. 'Descent' algorithms are shown to behave extremely poorly, whereas certain 'Gaussian-type' algorithms prove quite successful, even when the necessary model function derivatives are obtained by numerical evaluation, rather than analytically. Chapter 3 describes the generation of orthogonal polynomials in two dimensions, and their application to producing surfaces of optimum fit to the pointing data. The constraints on the manner in which pointing data can be acquired are severe, and their effect on the surface fitting procedure, and statistical properties of the fit, is described. Chapter 4 discusses the application of model and surface fitting to real data from a typical telescope of moderate size, and shows that the ultimate limit to the pointing improvement is set by the non-repeatable or hysteresial errors. The problems involved in devising error models are discussed, and an algorithm which permits efficient and simple experimentation with a given model is presented. The factors governing the choice of whether model or surface fitting should be employed, and estimates of how much data is required for satisfactory fitting are given, and the limitations of using model estimation techniques for locating and measuring the physical causes of error are delineated. The concluding chapter, Chapter 5, considers the practical implementation of computer pointing error correction. Two cases are considered, the generation of error fits to a previously collected data set, and an automatic correction package, which is unseen by the observer, and which progressively improves telescope pointing with the accumulation of fresh pointing data. The problems of implementing such a package are discussed, and hysteresis singled out as the most general and serious of them. Suggestions are made for future work in the areas of mechanical hysteresis, improvements to telescope collimation and instrument changeover procedures, and improved methods of approximating two-dimensional data. Appendices include an algorithm for mean to apparent place correction which is more suited to a real-time environment than the classical method, a discussion of ambiguity errors in gear driven digital shaft encoder pairs, and a description of a digital co-ordinate readout system designed by the author; the latter employs optical shaft encoders, a small computer, and a solar to sidereal frequency converter of the author's design, and is currently in service.

PREFACE

Telescope pointing errors, whilst by no means a major concern of astronomy, are highly detrimental to observing efficiency, and much of the effort in recent telescope design has been directed towards their reduction. The bulk of this effort has involved improved structures, drives or optic supports, and, although it has yielded beneficial results, telescopes still point erratically by tens of arcsecond. Here the philosophy of approach is to reduce pointing errors by computer correction, rather than by locating and remedying their causes. Its justification is threefold: (i) no single technique has so far proved universally effective; (ii) it is usually prohibitively expensive and time consuming to modify telescope components physically; and finally, (iii) computer or 'software' error correction is an ideal remedial measure for existing (particularly older) telescopes.

The idea is not new, being cited extensively throughout such references on telescope construction as the proceedings of the I.A.U. symposium number 27 (1965), and those of the E.S.O./C.E.R.N. conference (1971). However, very few of the references give details of the exact methods employed (or proposed) for error correction, nor the results of their application, if any. There is certainly a need for a comprehensive treatment of the problem, and this thesis aims to fulfil that need; it considers the general mathematical problem, the problems of pointing data acquisition, and the implementation of software correction methods on typical telescope and computer hardware. It is written for astronomers and telescope engineering personnel, and thus is apt to contain more lengthy descriptions of algorithms, calculations and definitions of terms etc., than would be the case were it aimed at the numerical analyst or computing theorist. It seeks to unify the astronomical, engineering, and computing aspects of the subject.

Although a factual summary of the contents appears in the abstract preceeding this preface, a brief mention of certain points of originality in the thesis is appropriate here. Although the model estimation algorithms used in Chapters 2 and 4 have been culled from the literature in the field, they have not, to the author's knowledge, been applied to model functions of such complexity before, nor to telescope pointing models as such. Since the success or otherwise of such algorithms is highly problem dependent, it is fortunate that some of the better algorithms have been shown here to be eminently satisfactory on the problem. (Certainly they behave sufficiently well as not to warrant effort to devise better model estimation algorithms). In Chapter 3 appears the first complete presentation of the computation of two-dimensional orthogonal polynomials suited to practical application; only a few treatments of surface fitting with a general data distribution occur in the literature, and these are far less detailed and lack a discussion of the statistical aspects of surface fitting. There appears to be some confusion amongst many telescope design and maintenance personnel as to the role of orthogonal polynomials in fitting. The author has often been referred to some specified standard polynomial sequence 'because it may prove more effective than others'. It is important to note that, once the dependent co-ordinate variables have been chosen, fits generated with different types of polynomial produce identical results, given unlimited arithmetic precision; however, orthogonal polynomials are optimal because they aviod the most serious causes of numerical error, and (once the polynomials are generated) involve less computational effort.

The scheme in Chapter 4, for compression and expansion of matrix equations, which permits the selective inclusion of model parameters in model estimation fits, although simple, has not previously been reported. It is more versatile than similar ideas in regression analysis, and more applicable to the type of data fitting problems encountered in the physical sciences. It is hoped that the author's original proposal for an automatic software correction package (in Chapter 5) proves a stimulus to workers involved in telescope design and operation. By and large, authors in the literature who consider the topic, can be divided into those who deem software pointing error correction prohibitively difficult to bother with, and those who propose to implement such a scheme, but fail to allow for the numerical and statistical problems involved. Here we establish the feasibility of software correction as well as delineating its problems and limitations. Two further original pieces of work appear in appendices. The computational method of mean to apparent place reduction in Appendix A is more suited to use in data processing, or telescope control tasks than the classical methods, which involve extraction of data from ephemerides etc. A very similar method has been published by Harris and Large (1967), but this was unknown to the author at the time Appendix A was written. Greater detail and explanation is given here, but the method does not differ substantially to that published. An improved method of converting a frequency based on the solar (or atomic) second to the equivalent sidereal frequency is given in Appendix D, and was published in Hovey (1973). The prototype converter based on this method is in service at Mt. Stromlo Observatory, A.N.U.

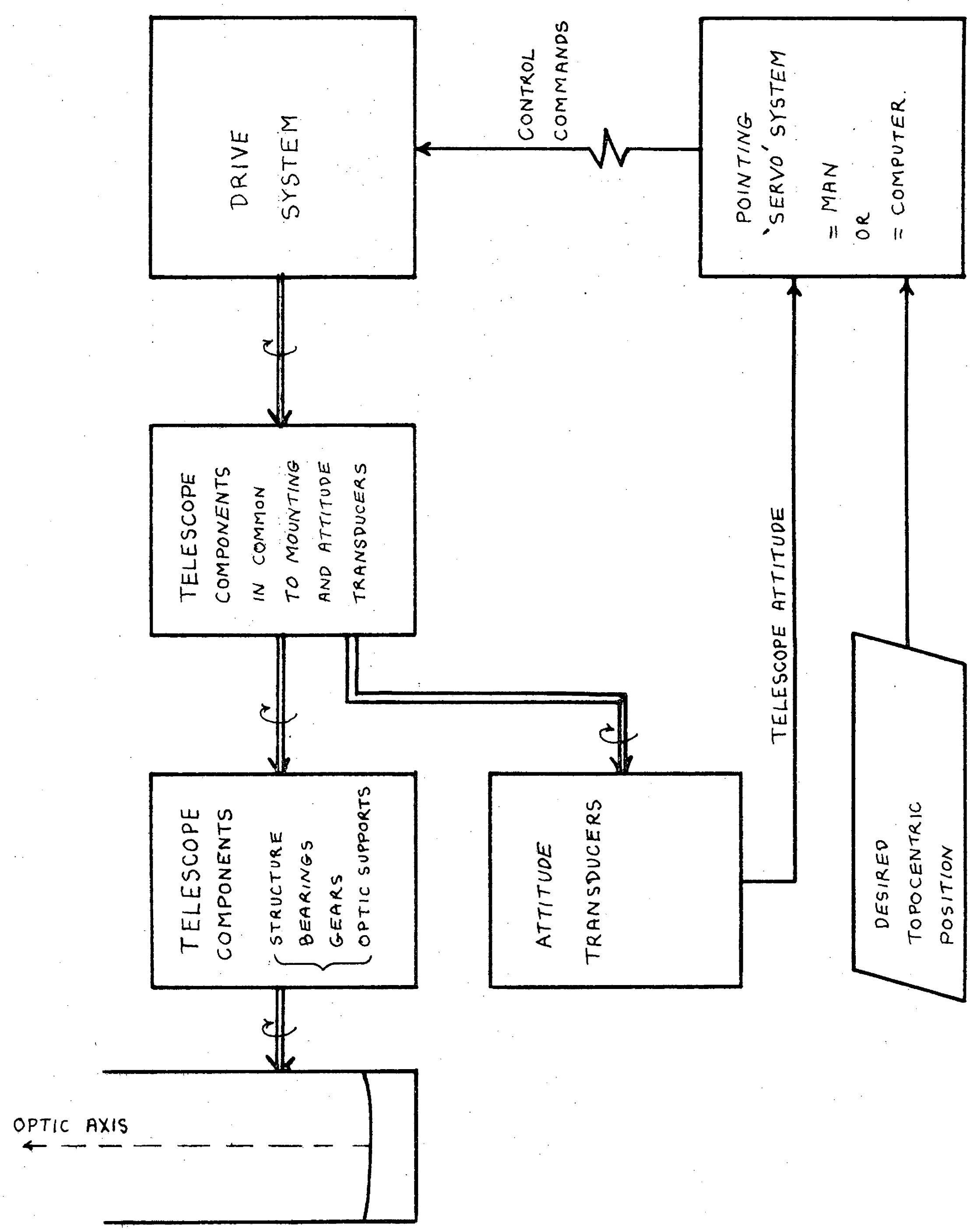
Finally, the author would like to thank personnel of Mt. Stromlo Observatory, Department of Engineering Physics, the A.N.U. Computer Centre and others for their suggestions and help in this interdisciplinary project.

CONTENTS

INDEX TO FIGURES

INDEX TO TABLES

FIG 1.1


DRIVE SYSTEM
CONTROL COMMANDS
POINTING 'SERVO' SYSTEM
= MAN
OR
= COMPUTER.
TELESCOPE COMPONENTS IN COMMON TO MOUNTING AND ATTITUDE TRANSDUCERS
TELESCOPE ATTITUDE
TELESCOPE COMPONENTS
STRUCTURE
BEARINGS
GEARS
OPTIC SUPPORTS
ATTITUDE TRANSDUCERS
DESIRED TOPOCENTRIC POSITION
OPTIC AXIS

CHAPTER ONE
INTRODUCTION

(1.1) TELESCOPE POINTING AND POINTING ERRORS

One of the most demanding positional control problems is the pointing of large astronomical telescopes, be they optical or steerable radio instruments. The initial acquisition of a celestial object (setting) involves orienting the telescope so that the co-ordinates read from the telescope axes match the known co-ordinates of the object. In practice, the position of the instrument's optical axis on the sky differs from the position given by the axis readout and the telescope is said to exhibit pointing errors. The magnitude and nature of these errors depends on the accuracy of the axis readout system and how directly it measures the position of the optic axis, i.e. how near the control loop illustrated in Figure 1.1 comes to enclosing the optic axis.

The necessary co-ordinates of an object are usually obtained from the known mean place of the object at a given epoch by applying corrections for proper motion, parallax, precession, nutation and aberration which results in the apparent place in the declination/right ascension co-ordinate system. This is transformed to a topocentric declination/hourangle frame and a correction for atmospheric refraction applied. Unlike the five astronomical corrections, refraction depends on wavelength, on environmental variables such as atmospheric pressure and temperature, and cannot be exactly predicted. A correction algorithm which is more suited to the on-line computation of such co-ordinates than are the classical methods is discussed in Appendix A, and even when approximations are used, the resultant topocentric positions are one or two orders of magnitude more exact than the pointing capabilities of existing telescopes.

To distinguish between the position of the telescope in its own frame, and the variously defined and derived astronomical co-ordinate systems, the term 'attitude' will be used for the former. After the telescope is set, it is usually required to 'track' (follow) the object; this may be done open-loop with manual adjustments to the tracking rate (guiding), or by a closed-loop system for example autoguiders or startrackers[@]. Tracking

---

@ Such exactitude is unfortunately necessary here since much of the relevant literature particularly in the space technology field features rather more loose usage of this terminology; see, for example Smith (1967) in which there is confusion of 'tracking' with 'pointing' and an error in the formula for resultant on-sky error.

methods and accuracies will be considered only in so far as they are related to the pointing accuracy of the instrument in question, and the author is concerned only with the case where there is a considerable complexity of structure, bearings, gears and mechanism between the closed control loop in Figure 1.1 and the optic axis. Nor are we concerned as to whether a human operator or a computer mechanism closes that loop. Automatic systems which close the loop around the optic axis itself can, of course, attain appreciably higher accuracies than blind pointing, depending on the mass of the element controlled[@]; see for example a survey article on star trackers by Seifert (1969).

It is difficult to assess the effect telescope pointing errors have had on astronomical research. Until recently in optical astronomy they were accepted with resignation. Although astronomical data and observations are relatively unaffected by pointing errors[@@], such errors cause a serious loss of observing efficiency. Astronomers become quite adept at visual recognition of star fields even when reversed by the telescope optics, and although such visual identification may never be obviated by improved pointing, the time required can be substantially reduced, since the star fields used can be commensurately smaller for increased confidence in the blind pointing accuracy of the instrument. With the increasing use of image intensification, video techniques and other fast electronic means of data acquisition, the time taken to set the instrument and locate the object is becoming a larger proportion of the total observation time.

Radio (and Infra-red) telescopes cannot directly form an image of the field and thus position determination is contingent on the pointing accuracy of the instrument. Certainly a more concerted effort to reduce pointing errors has been made by radio astronomers, for example, Struve et al (1960), Minnett et al (1967) and Meeks et al (1968)[@@@]. With the growing emphasis on correlating optical and radio sources and the increasing density of sources as instrument sensitivities improve, pointing accuracies will assume yet greater importance. The accuracy of interferometric

---

@ Often an autoguider or startracker servos a secondary optic surface or focal station rather than the whole instrument, allowing a much higher servo performance.

@@ Since, for example, position determination relies on the measurement of standard stars using instruments like transit telescopes and photographic zenith tubes (which are small and designed specifically for such work), and on offset measurements from such standards on photographic plates.

@@@ It is interesting to note that some of the better radio instrument pointing accuracies quoted are better than those of many optical telescopes.

position determination of radio sources is given by Fricke (1972) as 1 to 1.5 arcsecond in right ascension and 2 cosecδ arcsecond in declination (δ), and this sets a useful goal for the pointing accuracy of steerable instruments.

As well as degrading the efficiency of large conventionally operated telescopes, pointing inaccuracies constitute one of the major obstructions to the automation of astronomical telescopes. Maran (1967), Baker (1969) and Clarke (1970) cite automation as one means of increasing the efficiency of astronomical research, but it is only in the case of spaceborne telescopes operating with closed-loop control (startrackers etc.) that any degree of success has been achieved. Although there is currently little interest in completely automated ground-based telescopes, more and more telescope operations are becoming computer controlled, and a sufficient reduction of pointing error would permit telescope setting under program control.

Suitable pointing accuracy goals for proposed optical telescopes abound in the literature; a frequently occurring figure is ±5 arcsecond cited by Hoag (1965), the Anglo-Australian Telescope Project, Kitt Peak National Observatory, and others. To facilitate precise offsetting from an object already set on, e.g. when observing an object invisible to the astronomer, a differential pointing accuracy of ±0.1 arcsecond is suggested by Hoag (1965) and a similar figure is often given for the drive system tracking accuracy. A limit to the required pointing accuracy is set by the image diameter determined by the seeing and the optical aberrations of the instrument; Bowen (1967) gives 0.5 arcsecond for a typical image diameter caused by the optics, and 1 to 1.5 arcsecond for average good seeing. So although offset positioning could use almost any attainable accuracy, a practical figure for initial pointing accuracy of about an arcsecond is suggested here; the reduction of telescope errors to this level would be of substantial assistance to astronomical observation.

The past decade has seen an intense world-wide effort to construct versatile large optical telescopes, and review papers exist which show the dominant design trends, for example Baker (1969), and Gascoigne (1970). A summary (unfortunately incomplete) of some of the larger optical telescope projects at present under way was obtained from the ESO/CERN proceedings (1971), Solf (1971) and Matara (1969) and is given in Table 1.1. The figures for pointing accuracy are quite probably inconsistent (some are R.M.S., others peak etc.) but show that even with careful design, pointing errors are expected to amount to some tens of arcseconds. A similar summary gleaned from Findlay (1971) gives the pointing accuracy

Pointing accuracy of millimeter-wave radio telescopes; (A)=altazimuth, (P)=polar mounted.

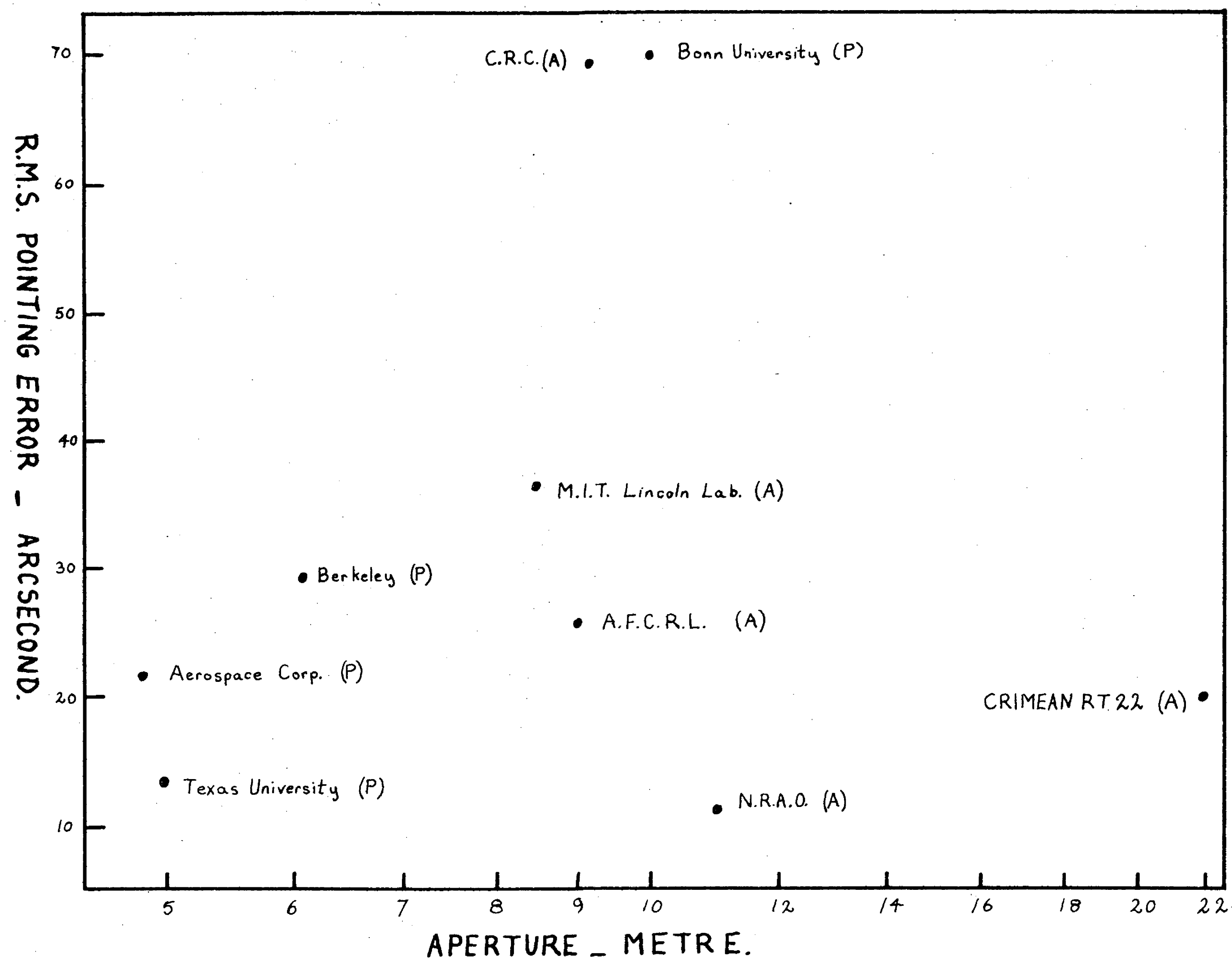


FIG 1.2

TABLE 1.1

INSTRUMENT	A.U.R.A. (1),(2) A.A.T. (3)	Carnegie Institute 'Du Pont'	Canadian Universities	French I.N.A.G.	Italian National Observatory	Max Planck Institute	Soviet	E.S.O.
SITE	(1) Kitt Peak (2) Cerro Tololo, Chile (3) Mt Siding Spring, Aust.	Las Campanas, Chile	Mt Kyobo, Canada	Pyrenees			Crimea	Cerro La Silla, Chile
APERTURE (metre)	3.8	2.54	3.98	3.6	3.5	2.2	6	3.66
MOUNTING TYPE	horseshoe modified yoke	fork	cantilevered fork	horseshoe modified yoke	fork	fork	Alt-azimuth	intermediate between a fork and a horseshoe modified yoke
POINTING ACCURACY (arcsecond)	10			20		10		60
COMMENTS	K.P.N.O. cite ±5 arcsecond within 30^{o} of zenith. A.A.T. hope for ±3 arcsecond after software correction.					pointing accuracy 10 arcsecond after refraction, flexure decollimation and circle errors are programmed out.	computer guided to 0.1 arcsecond, computer samples axes at 8 Hz and instrument mounts at 1 Hz.	flexion type tube. (a friction roller drive was originally planned).

of several millimeter-wave radio telescopes of various apertures and mounting types, and it can be seen from Figure 1.2 that the pointing accuracy of these structures is often only a factor of two worse than their optical counterparts.

(1.2) POINTING ERROR CAUSES

(1.2.1) Mountings

A telescope mounted in an earth-bound frame must be rotated about two different axes to observe any point on the available celestial hemisphere; one axis is fixed in the observatory frame (primary axis), and the attitude of the other (secondary) axis varies as the instrument rotates about this primary axis. Some space and airborne telescope pointing systems are of necessity more complicated (e.g. three-axis gimbals), see for example Fosth (1969) or Wischnia (1969), and these are not treated here. The primary and secondary axes are invariably orthogonal for reasons of independence and convenience, and it is the orientation of the primary axis, together with the position of the necessary bearings and loads, which catagorize astronomical telescope mounts. The following discussion assumes an optical telescope, but radio dishes have much in common. A more detailed and illustrated description of mounting types is given in a survey article by Bahner (1967).

An alt-azimuth mounting has its primary axis vertical, and thus is perfectly symmetrical with respect to gravitational loading; the secondary (altitude or elevation) axis keeps a constant attitude with respect to the vertical during rotation about the primary (azimuth) axis, and so the mount is expected to have the least gravitationally induced pointing error and the lowest mass for a given aperture (Mertz 1971). Owing to the restricted range, a simplified drive system, e.g. a hydraulic ram, is possible on the elevation axis. To track a celestial object, both axes and the focal station instrument mounts must be driven at varying rates, and because of rate limitations there is a dead-zone around the zenith, an area which is important to astronomy. Vaselevskis (1965) and Kuhne (1971) have considered the necessary practically attainable drive rates for the axes and Kuhne (1971) a complete plate holder rotation servo; they conclude that the advantages of such mounts are quite realizable for astronomical purposes. The largest optical telescope to be designed, the Soviet 6 metre instrument is alt-azimuth mounted and has occasioned a number of Russian studies, such as Mikhelson (1970). The control of such a mounting requires a computer, but for large instruments this constitutes a very small fraction of the total instrument cost; few radio instruments over

27 metre aperture are other than alt-azimuth mounted (Findlay 1971).

The equatorial or polar mounting has its primary axis directed close to the celestial pole, and thus a uniform drive rate about the primary axis alone is sufficient for tracking purposes, there is no serious dead-zone and, to first order, no field rotation except at the coudé focal station. Its simplicity and convenience has so far outweighed the fact that its gravitational asymmetry causes serious pointing errors and bearing problems. Many versions of the polar mount exist, the main ones being the English crossed-axis mounting, the fork, the German or asymmetric mount, the yoke (a fork with extended tynes joined and supported on an upper bearing) and the horseshoe modified yoke in which fork tynes are joined by a large horseshoe structure, near the declination bearings, which acts as a support bearing and rigidifies the tynes. Since there is access to the pole, good structural rigidity, and a large radius upon which to drive the polar axis, the last mentioned is becoming more commonly used particularly for large instruments.

The English mounting (Sisson 1965) requires a large counterweight on the end of the declination axis opposite the tube, and flexure of that axis can cause severe decollimation of the four-mirror coudé focus. It is becoming a less frequently used mounting for large instruments, the McDonald observatory 82-inch and 108-inch telescopes being two of the largest English mounted instruments. Most polar mounts, with the exception of the fork, can be tilted slightly to remove the effects of flexure of the polar axis and to some extent refraction, e.g. Arend (1951) and Bowen (1967). Flexure of the declination axis of a polar mount causes pointing (and tracking) errors. With a fork mounting however, it also causes field rotation which can be eliminated only at the cost of an increase in the declination pointing error; Vaselevskis (1962) shows that it is possible to design fork tynes which minimize both field rotation and pointing error in hourangle at the cost of the declination pointing error. Vaselevskis considers the rotation to be the worst effect but Kuhne (1957), by seeking to minimize the average pointing error over the sky, derives a latitude criterion for a choice between a fork and crossed-axis mounting. Unfortunately the criterion is somewhat arbitrary since it is critically dependent on the nature of the function averaged and the function used is not necessarily the most appropriate.

Various modifications to the fork mounting exist; the fork tynes may be bent for ergonomic reasons such as ease of access to instruments (e.g. 98-inch Maura Kea, Hawaii), or as in the case of the Isaac Newton 98-inch the tynes may be rigidly mounted on a large oil pad borne disk, Schematic diagram of Serrurier Truss telescope tube.

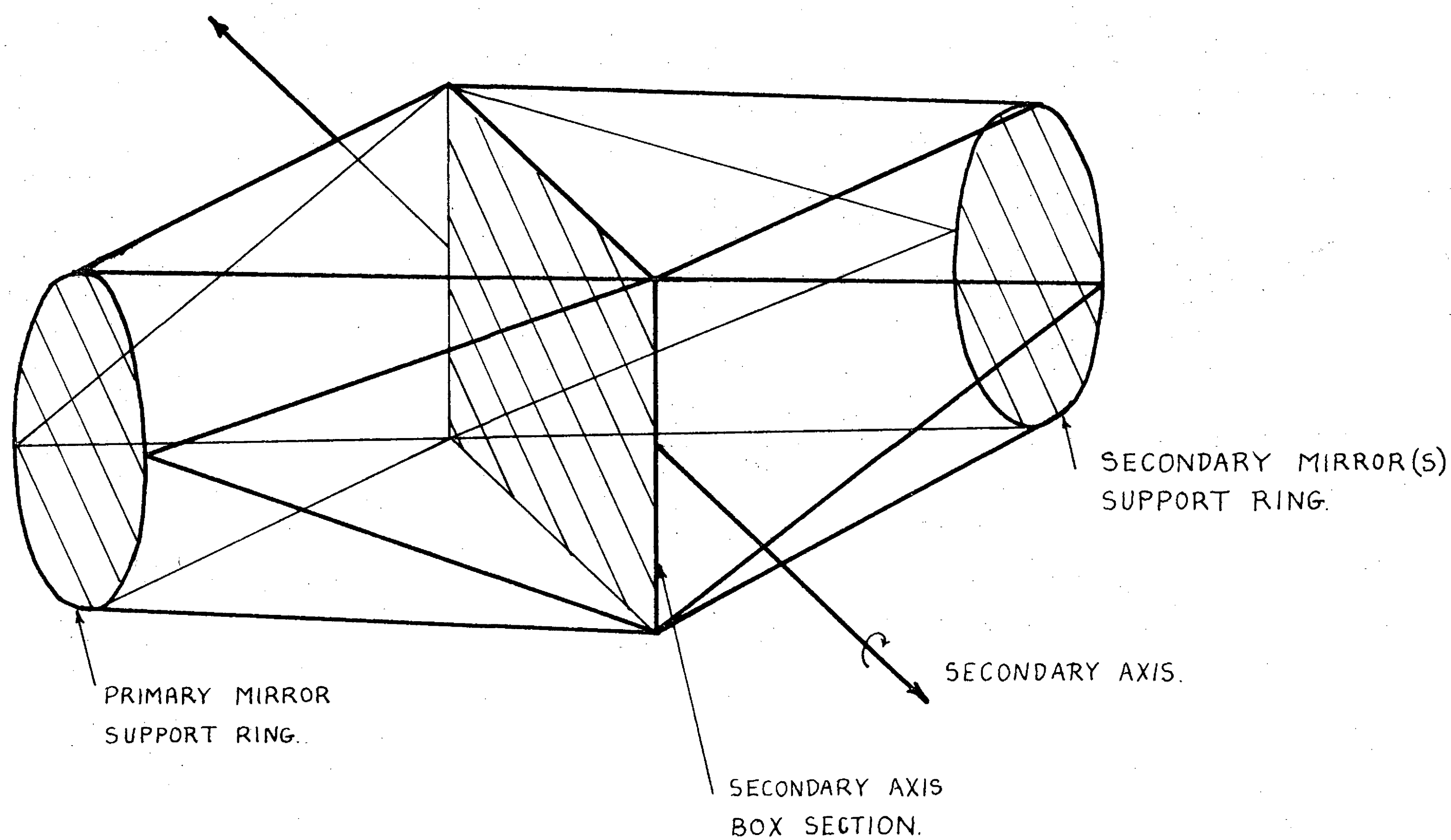


FIG 1.3

which serves as the polar axis (Sisson 1965). Completely asymmetric designs exist, e.g. Meinel (1971) (see also Meinel in Kuiper 1960), and can afford great convenience of operation; they are common up to 60 inch aperture but become excessively massive if used for large instruments. Other orientations of the primary axis are possible, for example the X-Y mounting which has it horizontal and (usually) directed north-south; it is used extensively for such applications as near-earth satellite tracking and has been suggested in the astronomical literature (where it is called the alt-alt). As Gascoigne (1970) notes, it has no advantages over other mounts for normal astronomical usage, except that a three-mirror coudé focal station is feasible.

The bearings associated with telescope mountings are the source of considerable trouble due to the effects of stiction or torque noise at the extremely low speeds involved. Conventional roller and ball bearings tend to slide and stick, and there is an increasing usage of oil pad type hydrostatic bearings, for example Pearson (1972). Estimates of hydrostatic bearing frictions are given by Barr (1969). The declination bearings in polar mounts give particular trouble for, with variation of hourangle, they must provide varying degrees of axial and radial thrust. Flexure bearings offer low stiction but are of no use when motion through large angles, or heavy loads are involved. Such bearings with breakaway torques of 0.001 lb.ft. and rotation angles of 4 degrees have been used in space vehicle applications. (Product Engineering November 1971)

(1.2.2) Telescope Structures

Strictly speaking, 'structure' includes the telescope mounting, but here we consider the telescope tube and other components which can be divorced from section (1.2.1). Gravitationally induced deformations in the telescope structure are possibly the largest single cause of pointing errors and one of the most difficult to measure and predict. One of the most important achievements to date which reduce this source of error is the Serrurier truss design of tube. Illustrated in Figure 1.3, it deforms as a parallelogram, permitting the optics to remain parallel and correctly spaced even though appreciably displaced from the centre section. For example (Rule 1965), the Hale 200-inch deflects approximately 1 cm for a translation between the end rings of only $\frac{1}{4}$ mm. The design is still effective for unequal tube half lengths since one can use unequal tube diameters, (Rule 1971), and the remaining pointing errors are usually due to the primary mirror cell and the secondary cage exerting a moment upon their respective end rings, as shown by Abdel-Gawad (1969) and Pope (1971).

The primary mirror movement and rotation is caused by mirror cell flexure (particularly when the four-fold symmetric truss tube is mated with a 120° spaced 3 point mirror support), and the secondary rotation is due to lack of rigidity of the secondary support drum. The two references last cited show that the three movements, that is rotations of the two mirrors and their relative translation, produce pointing errors and comatic aberration due to decollimation, and that in the case(s) considered (150-inch K.P.N.O. and A.A.T. telescopes) the pointing error is the more stringent criterion of design adequacy. If we were to assume a rigid tube mounted on torsionally elastic axes, telescope flexure would show a sine dependence on zenith angle.

Rule (1965) and Bertin (1971) draw attention to the desirability of keeping the structure mechanical resonance as high, and the mechanical Q as low as possible; most telescope structures have relatively undamped resonances in the region of 0.5 Hz, which is unfortunately similar to the periods involved in wind gusts and microseisms and imply an excessively long time for the structural motion to damp. For radio instruments the structure is associated not only with pointing but with the focussing and optical behaviour. Both active and passive forms of surface control are employed, for example Minnett et al (1967), Weidlinger (1967) and Findlay (1971), but one of the most promising techniques is the use of homologous deformations. It is possible, e.g. von Hoener (1967a), to design paraboloids which, with change of attitude, deform into other paraboloids of different focal length and axial direction. Pointing errors of typically 45 arcsecond can be thereby introduced, e.g. von Hoener (1967b); and thus effective methods of pointing error correction are highly desirable.

(1.2.3) Optics Support Systems

Pointing errors are also introduced by movements of the optical components themselves in their supports. Various support systems are treated at length in the literature, for example various authors in IAU Symposium No. 27, and in Crawford et al (1966). The actual deformations causing optical aberrations are also widely treated; the classical work here is by Couder and the articles by Schwesinger (1969) and Malvick (1972) give more recent suitable references. All support systems aim at applying uniform loading to the mirror for various attitudes, whilst allowing its positional constraint by a small number of adjustable supports (usually the kinematic requirement of three each axially and radially). Systems using passive air or mercury bags, active pneumatic systems and mechanical lever arrays are all used, but are not discussed here.

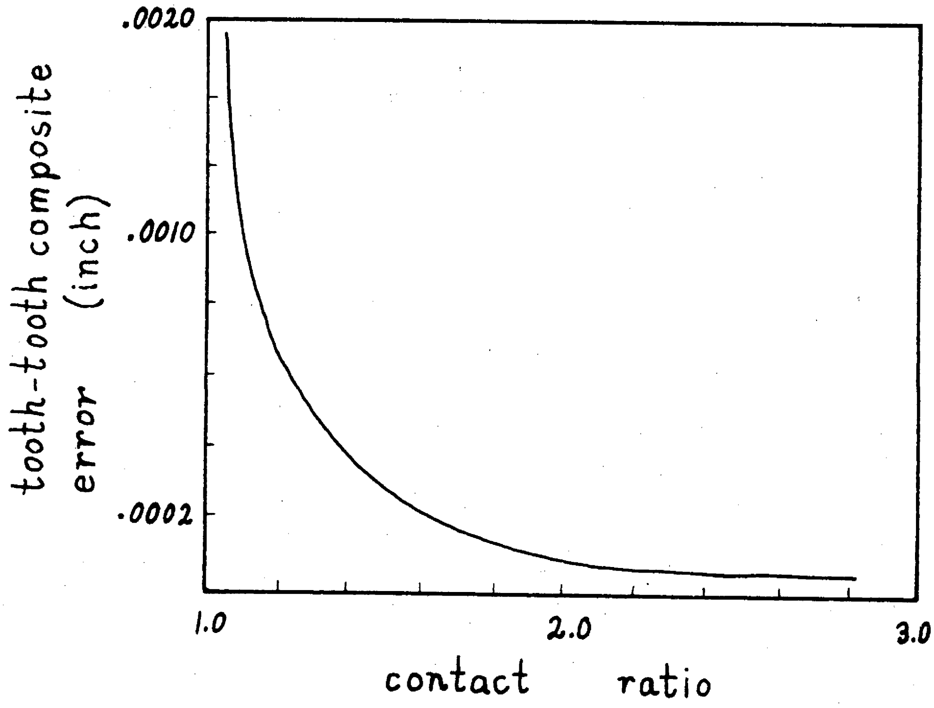
(1.2.4) Gearing and Drive Systems

No pointing errors would be introduced by the telescope drive system if the attitude encoding or readout was taken directly from the axes, and if there were no components in common to the drive and readout systems, but this is rarely the case in practice. Recently, tight, high-performance servo drives have become possible which can be applied very close to the axis, for example driving the final gearwheel. Trumbo (1965) describes a type of drive which takes digital rate information from the final pinion or wormshaft and uses it to control a torque motor. It is particularly suited to computer control of the tracking rate, and can be used to slew the instrument for setting purposes. Stepping motors can be used for very small instruments, e.g. Clarke (1971). Bertin (1971) notes that the highest resonant frequency of the mounting is achieved by applying the drive to large diameter components as close to the sections of high moments of inertia as possible, and that factors of improvement of the order of 1.5 can be obtained, for example, by driving the horseshoe of a horseshoe modified yoke mounting as opposed to driving the other end of the polar axis. The large diameter also implies a large number of teeth on the drive gear and lower tangential tooth loading which is an advantage, but the deformation of the gear is increased, and suitable geometry must be found to avoid the generation of pointing (and tracking) errors. Backlash in the final drive gears can be removed by a tangential preload, but this causes the drive system servo performance to be asymmetric, and for this reason such a scheme was rejected in favour of dual oppositely loaded pinions by the A.A.T. as described in Minnett (1971).

The final drive gears can be wormwheels, spur or helicoidal gears. Problems exist with worm gears in-as-much that they are not reversible, and either protective inertia must be added to the worm (which degrades the drive system response), or the worm must be mounted in a slide carriage (which degrades rigidity and alignment); also they have low efficiency e.g. 10 to 15% for a 1:720 ratio worm/wheel pair. Zero wear, which is governed by a critical ratio of maximum shear stress in the tooth contact zone to the shear yield point of the gear material, cannot be attained by the usual hardened steel worm and a bronze or meehanite[@] wormwheel; see Bertin (1971). However worm gears can be made more accurately than cylindrical gears because the wormwheel is generated by an envelope method essentially the same as using the worm as a hob.

@ Type of cast iron.

FIG 1.4


.0020
.0010
.0002
tooth-tooth composite
error (inch)
1.0
2.0
3.0
contact ratio

Bertin (1971) gives a figure of 5 micron for the tooth to tooth error, and 25 micron for the accumulated pitch error on a 3.6 metre diameter cylindrical gear; these figures correspond to angular figures of 0.5 and 2.5 arcsecond, and are approximately 2.5 times worse than those for a similar wormwheel, particularly if a duplex worm[@] is used. Baustian (1965) gives somewhat different figures but agrees that worms are more accurate. The tangential tooth to tooth composite error is the accumulation of pitch and profile errors and may be up to 2 arcsecond for high quality cylindrical gears (Bertin 1971). Combined with similar errors of the meshing pinion, a tangential jump of about 3.5 arcsecond is possible for a contact ratio (number of teeth in contact) of unity.

Martin (1967) found experimentally that the tooth to tooth composite error decreases quickly for increasing contact ratios, and for a given ratio is reasonably independent of pitch and pressure angle; his graph is reproduced in Figure 1.4. For a standard spur gear[@@] with a given number of teeth we can vary only the pressure angle to increase the contact ratio, and the limit is in the region of 2.9, with a pressure angle of 12.5°. The risk of tangential jumps disappears if the contact ratio approaches 6 or 8, and this is possible with helicoidal gears since we can vary the helix angle and the gear face width, e.g. tangential jumps can be reduced to 0.04 arcsecond on a 10 metre diameter helical gear of 175 mm width. Barr (1969) concludes, for cases involving roller bearings, that the extra accuracy of the worm is not realisable due to the manner in which the drive load is applied to the bearings, and quotes a helical gear with a contact ratio of 4.4. Groenveld (1969) gives the requirements of high positional accuracy and smooth slow speed running as being, (i) an involute helicoidal thread form, (ii) low (10°) pressure angle and (iii) the use of all-receeding tooth action.

A drive system employing a hardened steel friction roller is feasible, and is used on the French Chilean 60-inch. A similar drive was originally proposed for the 3.66-metre E.S.O. telescope at Cerro La Silla Chile, but extreme cleanliness is necessary for their successful operation and a helical gear system was eventually used. Barr (1969) dismisses steel belt type friction drives by showing that the circumferential stretch on an

---

@ The pressure angle and pitch varies along the length of a duplex worm, increasing the contact ratio.

@@ With equal addendum and module.

inch wide 150 inch long belt of 0.03 inch steel is of the order of 1/3 inch under slewing (fast motion) conditions, but his reasoning is related to the performance of the drive servos, not pointing or tracking errors.

Pointing errors produced by gear imperfections (assuming there are no tangential jumps etc.) are largely periodic with periods of a single revolution and low multiples of it caused primarily by eccentricity and pitch errors, and also errors at the tooth period and multiples of it caused by tooth profile errors. Tooth period errors are more usually a cause of tracking rather than pointing error. Kron (1960) gives one possible explanation of periodic errors in worm drives, viz. a worm which has been made with the correct hob but is of incorrect pitch diameter. The simpler forms of periodic error in worm drives can be removed by using a tilted thrust bearing on the worm shaft, or, as noted by Hardie et al (1962), an eccentrically mounted spur gear pair driving the worm shaft to produce a compensatory oscillation. Clearly both techniques are more difficult to implement than a form of programmed or software correction of pointing (and tracking) errors.

(1.3) TELESCOPE ATTITUDE READOUT SYSTEMS

It is important to distinguish pointing errors caused by the system for readout of the telescope axes from those which are due to the structure, mounting and optics supports, since the former is more easily modified or updated. Simple analogue systems, for example selsyns, often exhibit quite large errors, e.g. tens of arcseconds, particularly when large gear train ratios are required to produce the required angular resolution. If gear errors are not dominant, considerable improvement can be obtained by digitizing the selsyn transmitter outputs, e.g. as on the Parkes 210-foot radio dish. The ideal requirement of an absolute transducer with a resolution <u>and accuracy</u> of the order of an arcsecond is onerous, and has not been achieved on instruments of large aperture. Commercial absolute digital shaft encoders are available with a wide range of resolutions and codes, and are usually either brush contact type of resolutions of up to about 12 bits (5 arcminute), which combine simplicity of operation with economy, or optical types which are capable of higher resolutions up to 21 bits (0.7 arcsecond) and have longer rotation lives, but require more elaborate electronics. Both types employ either monostrophic codes, e.g. Gray-code, or use redundant tracks and lead-lag sensor selection to remove the inter-track ambiguity. It is often most convenient to feed the encoder output directly to a computer for reduction and formatting for display purposes, but many installations exist, or are proposed, which use special purpose hardware to handle the encoder data, for example Vokac

(1970). Matara (1969) discusses some of the errors which can occur in conventional systems.

Moiré fringe optical gratings can achieve very high resolutions and accuracies. N.E.L. Scotland have a master grating of 43,200 lines on a 17.5 inch diameter, and firms such as Baldwin-Rotax U.S.A. have developed similar masters; it is now fairly certain that the accuracy of these large gratings is of the order of an arcsecond. Moiré gratings have been used in incremental encoders, often with two read heads in phase-quadrature to give directional information. Russell (1966 and 1969) has developed a method of obtaining absolute readout from a series of concentric gratings of different but integrally related angular pitches, each of which can be electronically divided by integers as high as 60. The scheme depends on analogue segmentation of the serrusoidal waveforms from reading sensors in phase-quadrature, and has been applied to the elevation axis of a kinetheodolite at Edinburgh Observatory by Whitwell (1972). The use of multiple read heads can remove the effects of grating eccentricity on the accuracy of the readout, but a limit to the tolerable eccentricity is set by its effect on the amplitude of the serrusoidal signals from the averaged heads. Linear Moiré gratings ruled on steel tapes can be obtained in resolutions of a micron for a total range of 2 metres (Whitwell 1972), and could be used for angular readout by winding them on a carefully machined diameter.

The bearings in commercial shaft encoders can be precise with an adequately low runout, e.g. a runout of 30 micro-inch for 4 inch diameter angular contact ball bearings, and the constraint on the system accuracy is often the manner in which the encoder is driven from the axis. Serious wind-up errors occur in flexible disc or bellows couplings used to couple the encoder to the axis or to intermediate gearing. Bertin (1971) and Barr (1969) discuss the use of a friction roller system to drive the attitude encoders, and note that although more accurate than gear systems, extreme cleanliness is required, and there is positional creep with both predictable and random components. It is very doubtful that any coupled or separately driven encoder can realise an accuracy commensurate with the resolution required, and making the encoder integral with the mounting bearing structure, alone offers substantial improvement. Provided discontinuous effects like backlash are removed, encoder system errors are often quite smooth and repeatable, and thus in principle well suited for software elimination.

Many other types of angle transducers exist, and a survey article by Sydenham (1968) lists several; with two exceptions they have found little use in telescope attitude readout. Inductosyns are a magnetic device comprising two stator windings in quadrature, and a rotor with many poles; the sinusoidal output waveform is digitized using an analogue to digital converter (A.D.C.). Klock et al (1969) describe a 12 inch diameter, 2048 pole version incorporated into the U.S. Naval Observatory transit circle, which affords a resolution of 0.05 arcsecond, with a calibrated mean error of 0.5 arcsecond, when read with a 15 bit A.D.C. Struve et al (1960) describe similar units made by Farrand Optical Co. and installed on the N.R.A.O. 140-foot radio telescope, which is intended for source position measurement with an accuracy of 10 arcsecond.

A very similar device called a Raksyn, employing a toothed rotor and stator with capacitive sensing, is used on the 150-foot radio telescope at Algonquin Park Ontario, and is described by Ayre (1967). Like the Parkes 210-foot instrument, this instrument employs a master equatorial unit (M.E.U.) which is pointed appropriately and the main structure, which is altazimuth mounted, is slaved to it. Pointing accuracies of 30 arcsecond are quoted but are not limited by the Raksyns, which are used both on the M.E.U. (2 foot diameter) and on the telescope proper (9 foot diameter); 9 arcsecond is given as the accuracy of the M.E.U. Inductosyns and Raksyns appear to be less troublesome than optical encoders, and can be built to larger diameters, promising increased accuracy; they are, however, somewhat rare.

(1.4) SOFTWARE CORRECTION OF POINTING ERRORS

A decade ago the notion of using a computer to correct telescope errors would have been somewhat premature, but with the steadily increasing application of computers to telescopes for other reasons like data handling, and the decreased relative cost of small computer installations, it is now seen as an obvious step. The adoption of computer-control by astronomy has been slow by comparison to other fields; articles such as Clarke (1967) and Endeavour (1970) record the process. The computerization of various telescope functions along with increased use of small telescopes and photodetectors of higher quantum efficiency, is quoted by Maran (1967) and Disney (1973) as one of the possible cures for astronomical research bottlenecks. Complete automation of optical telescopes has been investigated, for example the 98-inch Mauna Kea telescope Hawaii, and a 24-inch instrument at M.I.T. described by McCord et al (1972). Remotely operated telescopes have been constructed; the twin 16-inch Edinburgh instrument is described by Reddish (1966), and a less successful 50-inch by Maran (1967), but the best examples have been the various orbiting space telescopes.

The incentive for computer correction of errors is firstly necessity, for no other technique is to date sufficiently successful. The second and possibly prime attraction of such a scheme is its possible application on existing, older telescopes, whose design and construction predates the various design improvements so far discussed. Its usefulness on more recent 'state-of-the-art' instruments is not impaired, since such improvements result in a higher degree of repeatability as well as smaller errors; hence the possibility of attaining the accuracy goals cited before. The Anglo-Australian Telescope Project hopes to reduce the A.A.T. 150-inch pointing errors from 10 arcsecond to 3 arcsecond by suitable software correction (design figures Pope 1971), while Solf (1971) intends programming out flexure, decollimation and circle errors as well as the usual refraction to achieve 10 arcsecond on the Max Planck 2.2-metre. The various methods proposed are only vaguely described and there are no general treatments of the problem in the literature; Lausten and Malm (1971) propose a table look-up procedure for the gear errors on the E.S.O. 3.6-metre instrument, while A.A.T. intend to obtain suitable values for parameters in error model functions (private communication). Smith (1967), considering radio telescopes, dismisses the problem for reasons of difficulty! Simpler cases can be solved quite neatly, for example the standard approach of Fourier analysis of meridian circle errors (single axis in declination) as in Dejaiffe (1970).

Two of the more elaborate treatments which appear in the literature involve parameter estimation studies of the Australian 210-foot radio telescope at Parkes and the M.I.T. Haystack antenna. Minnett et al (1967) use a six parameter model to represent the vertex co-ordinates, the direction of the optic axis and the focal length of the Parkes paraboloid as it distorts with motion in zenith angle, and fit 150 survey measurements of the paraboloid by a least-squares process to find a law for optimum focussing. Meeks et al (1968) used 172 settings on radio sources of known position corrected for refraction, and least-squares fitted for seven parameters describing azimuth axis tilt, azimuth offset, collimation error, elevation axis skew and gravitational flexure. The flexure term was simply the sum of a linear function of zenith angle, and a constant which can be taken to be the elevation offset, this giving better results than the expected sine function of zenith angle. The sumsquared error in each co-ordinate was separately minimised, giving two independent estimates of each parameter which were then averaged; applying the fitted model as pointing corrections, they obtained an improvement of a factor of 2 in the peak to peak error of the weighted azimuth co-ordinate A.cos(h) and a factor of 3 in elevation h.

This thesis contains a more general investigation of computer correction of telescope pointing errors, and although the final assessment of the various ideas discussed herein involves the use of pointing data from a specific telescope (see Chapter 4), guidelines are established which are of general utility. For the most part a polar mounted optical telescope is discussed, but nowhere is this assumption restrictive. Because of the size of typical pointing errors relative to the range of movement of the telescope, the best method of measuring pointing errors is by setting on suitable celestial objects, and comparing the expected topocentric co-ordinates@ declination δ_0 and hourangle H_0 with the corresponding values δ and H obtained from the telescope readout system. The pointing errors in the two co-ordinates are here defined in the sense

$$\left.\begin{aligned} \Delta\delta &= \delta_0 - \delta \\ \text{and} \quad \Delta H &= H_0 - H , \end{aligned}\right\} \quad \ldots 1.1$$

and are taken to be functions of δ and H, the telescope attitude.

Conceptually the simplest approach would be to isolate the various causes, e.g. gearwheel eccentricity, tube flexure and so on, and physically measure each of them separately. Despite the attractiveness of this method, it is very limited in practice; measuring a parameter requires dissembly of, or special modifications to the telescope, updating the measurement with change of telescope configuration is onerous, and difficulty of measurement increases the closer one tries to measure to the optic axis of the system (see Figure 1.1). Thus the basic question is that supposing we have sampled the errors over the area of the sky it is desired to use, how best to use this data to improve the pointing accuracy at any given point within this area. A limit to the improvement possible in any given case is set by the extent to which the errors are repeatable, i.e. for a given telescope configuration the extent to which they are single-valued functions of telescope attitude@@, and to which backlash and mechanical hysteresis are absent.

The approach taken clearly depends on how much is known about the cause of the errors. If no assumptions can be made as to the causal nature of the errors, we have the problem of two error surfaces $\Delta\delta$ and ΔH, in the variables δ and H, which must be interpolated or surface fitted

---

@ corrected for refraction and any effect not caused by the instrument itself.

@@ and perhaps other simple environmental variables such as temperature.

to effect the required improvement in pointing accuracy. This is considered in Chapter 3. Should we have insight into the error causes, then a mathematical model of the erratic telescope can be postulated, and the data fitted by finding the best values of the model-defining parameters according to some suitable criterion. Such models are rarely linear in the parameters and the problem of non-linear parameter estimation of telescope models is discussed in Chapter 2.

Fundamental to both surface and model fitting is the choice of a suitable criterion of fit. If we have n observations over the area of interest, and δ_c and H_c are the co-ordinates computed from δ and H on the basis of our surface or model function, we need to minimize a suitable norm@ of the 2n-dimensional error vector which has components of the form

$$(\delta_0 - \delta_c) \quad \text{or} \quad (H_0 - H_c) \; . \qquad \ldots \; 1.2$$

The two most common norms used in optimization methods are the least-square or L_2 norm and the minimax or L_∞ norm@@. The latter enables the maximum error to be minimized, and, although this would be highly desirable for practical reasons, L_∞ is a function peculiar to the data used and is not even a differentiable function of the parameters or of the error vector components, and thus is considerably more difficult to minimize. Although other more complicated norms exist, the leastsquares norm alone is both easily computed, and appropriate to our problem. In Chapter 2, it is shown that (with the choice of a suitable weighting function) the L_2 norm can represent a physically important angle on the sky. If the components of the error vector are normally distributed about a zero mean, then an estimate of the model parameters or surface coefficients obtained by minimizing the L_2 norm is identical with the statistical maximum-likelihood estimate, and thus the use of the L_2 norm is also dictated by regression theory. The computation of such estimates of the model parameters or surface fit coefficients is the key to software pointing error correction.

On the subject of model fitting, Box and Hunter (1965), and others note that it is necessary to distinguish between 'response surface optimization' in which we attempt to optimize a variable of interest, (in our case the root mean square error on the sky) and are not particularly

---

@ generalised definition of a vector's 'length'; the Euclidean norm corresponds to the physical length in the 3-dimensional case.

@@ known also as the Euclidean and Chebyshev norms respectively.

interested in the means by which it is done, and 'mechanism determination' in which we wish to ascertain the causes of the behaviour of that variable. 'Mechanism determination' requires better statistical design of the experiment, and more careful statistical testing of the results than the former does. The manner in which pointing error data can be acquired is quite restrictive; it is quite impossible to obtain data points at arbitrary positions on the sky, or obtain them on a uniformly spaced grid, and so designing an experiment by specifying the co-ordinates at which data is taken, e.g. Box and Coutie (1956), is out of the question. It is possible to obtain data on lines of constant δ_0, and whilst it would slightly simplify surface fitting procedures (see Chapter 3), it is prohibitively difficult and time consuming to be used in practice. The practical constraints of data gathering certainly favour the 'response surface optimization' approach but in Chapter 4, it is shown that, with suitable care, model fitting can locate some of the causes of pointing errors, though by no means all. However, it is, along with surface fitting, emminently satisfactory as a means of their reduction. In the final chapter (Chapter 5) the practicalities of automated pointing error reduction and the limitations imposed by hysteresial errors are discussed.

CHAPTER TWO

PARAMETER ESTIMATION IN TELESCOPE POINTING ERROR MODELS

(2.1) PRELIMINARY

It is generally supposed that a knowledge of the pointing error causes is an advantage in any scheme intended to reduce pointing errors, since a mathematical model can be proposed for the errors in terms of physical parameters of the instrument, for example gear eccentricities, misalignments, structural parameters etc. We represent the model by

$$y \simeq f(\underline{x},\underline{b}) \; , \qquad \ldots 2.1$$

where y is the experimentally observed dependent variable, $\underline{b} = (b_j)$, $j=1,\ldots k$ is the vector of k (unknown) parameters, and $\underline{x} = (x_q)$, $q=1,\ldots m$ is the vector of m independent variables. Suppose we have y_i, $i=1,\ldots n$ which are n observations of y, and $\underline{x}_i = x_{qi}$, $q=1,\ldots m$, $i=1,\ldots n$ the corresponding observations of the independent variables, then our problem is to find $\underline{B}$, an estimate of parameter vector $\underline{b}$, such that the sum of squares

$$\phi = \sum_{i=1}^{n} (y_i - f_i)^2 \qquad \ldots 2.2$$

is a minimum, where

$$f_i = f(\underline{x}_i, \underline{b}) \; , \qquad \ldots 2.3$$

and we are using the leastsquares criterion discussed in Chapter 1. It should be noted that 'sumsquare' ϕ in equation 2.2 is a function of $\underline{b}$ alone (for a given set of data y_i, $\underline{x}_i$, $i=1,\ldots n$) and can be written

$$\phi = \phi(\underline{b}) \; . \qquad \ldots 2.4$$

Thus the problem is essentially one of function optimization, but many algorithms have been designed specifically to minimize sums of squares, and it is the literature on leastsquares parameter estimation, much of it written by authors in the chemical engineering field, which is perhaps more relevant.

In our pointing error work the (two) components of vector $\underline{x}_i$ are the i^{th} observations of the telescope attitude δ and H taken from the axis readout system, y_i is either of the topocentric co-ordinates δ_0 or H_0 of the particular celestial object, and f_i is the corresponding co-ordinate δ_c or H_c computed from our model function. Since ϕ is scalar and one requires a unique 'best' estimate of the parameter vector one cannot simultaneously minimize the residuals

$$\left.\begin{aligned}\sum_{\text{data}} (\Delta\delta)^2 &= \sum_{\text{data}} (\delta_0 - \delta_c)^2 \,, \\ \text{and} \quad \sum_{\text{data}} (\Delta H)^2 &= \sum_{\text{data}} (H_0 - H_c)^2 \,.\end{aligned}\right\} \quad \ldots\ 2.5$$

Meeks et al (1968) minimize them separately and average the two $\underline{b}$ vectors obtained, but a far better scheme is that now described.

Where it is desired to weight certain of the n data points obtained from the experiment, equation 2.2 can be written

$$\phi = \sum_{i=1}^{n} (y_i - f_i)^2 \, w_i^{\,2} \,, \qquad \ldots\ 2.6$$

where w_i is the value of the weighting function for the i^{th} point. Here we extend this equation to include s such terms and define ϕ by

$$\phi = \sum_{l=1}^{s} \sum_{i=1}^{n} \left[y_i^{(l)} - f_i^{(l)}\right]^2 . \left[w_i^{(l)}\right]^2 . \qquad \ldots\ 2.7$$

If we use s=2 and make the following identifications:

$$\left.\begin{array}{lll}
\left.\begin{array}{l} x_{1i} = \delta \\ x_{2i} = H \end{array}\right\} & \text{telescope attitude reading for } i^{th} \text{ observation;} \\
\left.\begin{array}{l} y_i^{(1)} = \delta_0 \\ y_i^{(2)} = H_0 \end{array}\right\} & \text{topocentric position of object for } i^{th} \text{ observation;} \\
\left.\begin{array}{l} f_i^{(1)} = \delta_c \\ f_i^{(2)} = H_c \end{array}\right\} & \text{computed from model function;} \\
\left.\begin{array}{l} w_i^{(1)} = 1 \text{ for all } i \\ w_i^{(2)} = \cos\delta \end{array}\right\} & \text{weighting function;}
\end{array}\right\} \quad \ldots\ 2.8$$

then ϕ in equation 2.7 comprises the first order terms of $(\Delta R)^2$, the square of the resultant error on the celestial sphere, which is given (to second order) by

$$(\Delta R)^2 = (\Delta\delta)^2 + (\Delta H)^2 \cos^2\delta + (\Delta\delta)^2(\Delta H)^2/\,2 \,. \qquad \ldots\ 2.9$$

Thus by minimizing ϕ as defined in equation 2.7 we approximate very closely the minimization of a practically tangible variable, namely the Root Mean Square (R.M.S.) pointing error.

Fortunately our modified definition of ϕ is compatible with existing formulations of leastsquares problems[@]. Many of the more successful nonlinear parameter estimation algorithms are based on the traditional

---

@ In fact all we have done is to employ n.s data points with s different expressions for the weighting and model functions.

Gaussian approach of expanding the model function in a Taylor series about the current estimate of the parameter vector $\underline{b}_0$, retaining only terms to first order. This expansion is substituted in ϕ in equation 2.7 and the derivative of ϕ with respect to a general parameter b_j set to zero; this results in a system of matrix equations (called the 'normal equations')

$$A\,\underline{t} = \underline{g} \qquad \ldots\ 2.10$$

to be solved for vector $\underline{t}$ which is a correction vector, and allows us to iteratively improve our estimate of the parameter vector by

$$\underline{b} = \underline{b}_0 + \underline{t}\ . \qquad \ldots\ 2.11$$

The kxk matrix $A = (A_{jr})$, and the k-vector $\underline{g} = (g_j)$ are given by

$$A_{jr} = \sum_{l=1}^{s} \sum_{i=1}^{n} \left[w_i^{(l)}\right]^2 \cdot \frac{\partial f_i^{(l)}}{\partial b_j} \cdot \frac{\partial f_i^{(l)}}{\partial b_r}\ , \qquad \ldots\ 2.12$$

and $$g_j = \sum_{l=1}^{s} \sum_{i=1}^{n} \left[w_i^{(l)}\right]^2 \cdot \left[y_i^{(l)} - f_i^{(l)}\right] \cdot \frac{\partial f_i^{(l)}}{\partial b_j} \qquad \ldots\ 2.13$$

respectively and differ from those usually given only in the extra summation over the s parts of our sumsquare ϕ .

(2.2) ALGORITHMS FOR NONLINEAR PARAMETER ESTIMATION

A discussion of some of the algorithms which have been used in the literature for estimating the parameters of nonlinear models follows. A broad outline of this field is given, but, except for the algorithms actually implemented and tested later in this chapter, it is in no way detailed or complete. Most methods of any significance are discussed in a review article by Spang (1962), texts by Wilde (1964) and by Draper and Smith (1966)[@], and/or a review monograph by Kowalik and Osborne (1968).

Minimizing a function of one variable (or parameter), or minimizing a multivariate function along a line is comparatively straightforward. Direct search algorithms, which involve simple comparisons of a sequence of trial solutions in such a manner that the trials become closer together in either a golden or Fibonacci sequence, are very stable but slow, and more efficient behaviour is exhibited by algorithms which employ quadratic interpolation to find the line minimum. In an algorithm by Powell (in Powell 1964) a quadratic interpolation is fitted to three evaluations of the function, and the analytically calculated minimum of this interpolating quadratic is used to replace one of the original points. An alternative

---

@ See for example the bibliography on nonlinear methods in Draper and Smith (1966).

approach by Swann et al (1964)[@] uses the calculated quadratic minimum to improve a group of points bracketing the function minimum. All of the above algorithms are treated by Kowalik and Osborne (1968), and flowcharts of the Powell and 'golden section' direct search algorithms are given later. Direct search methods tend to have very slow convergence properties when used for higher dimensions particularly when approaching the minimum. Despite this, methods such as the Simplex method[@@], in which a geometric pattern of points is progressively translated and scaled so as to locate the minimum, and others which involve sequential searches in orthogonal directions, the most notable of which is due to Rosenbrock (1960), have found many applications.

(2.2.1) Descent Methods

The slow convergence of direct search methods is due to their use of function values alone, and to their simple utilization of these values. By contrast, descent methods employ the gradient vector of the function to be minimized in such a way that the progress of the algorithm is always directed 'downhill'. In general, the new estimate of the parameter vector is formed from the current estimate $\underline{b}_0$ by

$$\underline{b} = \underline{b}_0 + \alpha D^{-1}\underline{g} , \qquad \dots 2.14$$

where $\underline{g} = (g_j)$ is the negative gradient of ϕ and is given by

$$g_j = -\partial\phi/\partial b_j , \qquad \dots 2.15$$ [@@@]

α is the iterative step size and D is a positive definite weighting matrix. The most common such algorithm is the method of 'steepest descent' in which D is simply the unit matrix and the algorithm proceeds directly down the gradient orthogonal to contours of constant ϕ. Other weighting matrices are in use, for example Newton's method which uses

$$D_{jr} = \frac{\partial^2 \phi}{\partial b_j \partial b_r} , \qquad \dots 2.16$$

Such a scheme entails rather onerous computing requirements and in this chapter we restrict ourselves to methods which require the computation of (at most) only the first derivatives.

It is usual in nonlinear problems for the contours of constant ϕ to

---

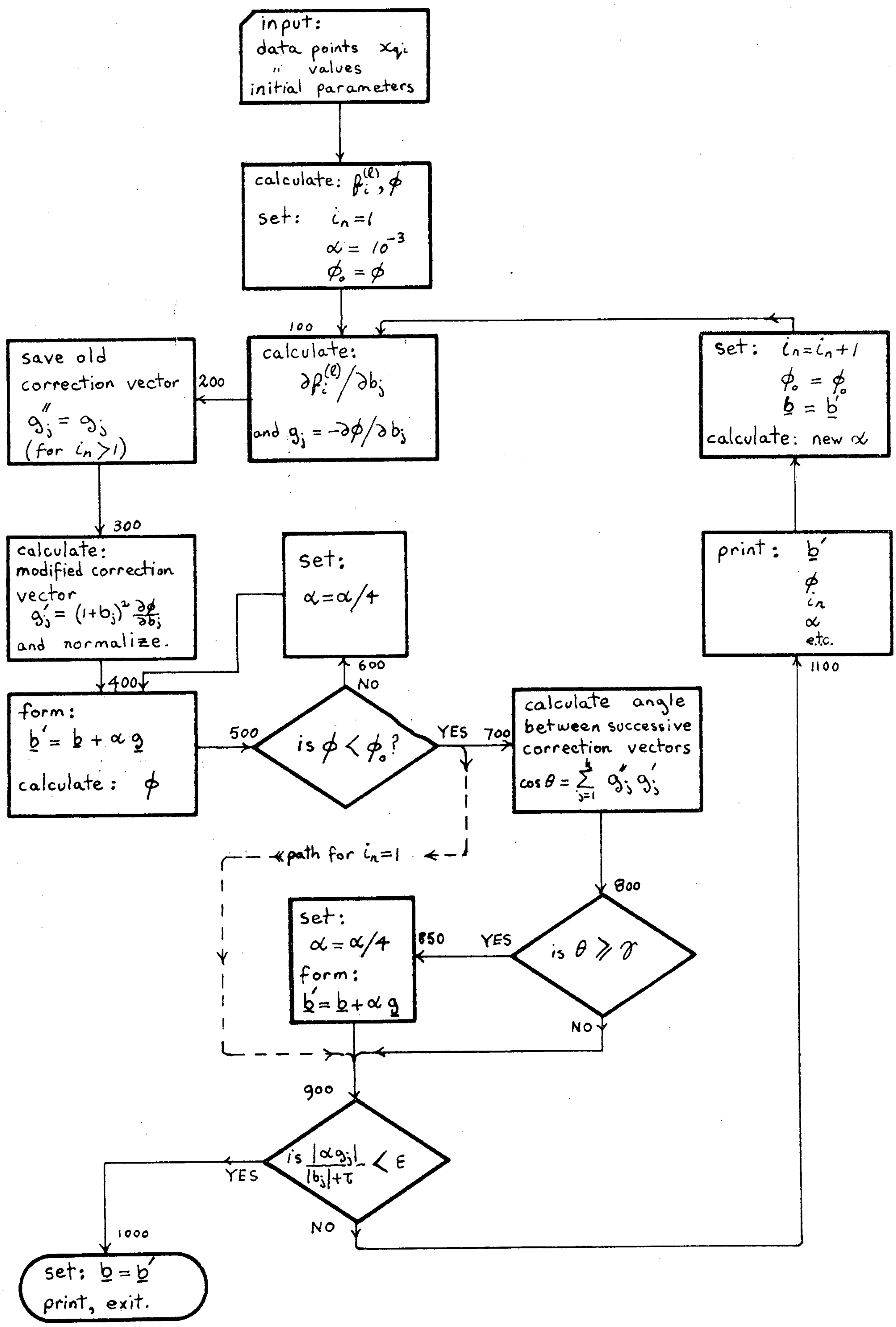
@ Called the Davies-Swann-Campey algorithm in the literature.

@@ In a k-dimensional problem a simplex is a set of k+1 points.

@@@ Vector $\underline{g}$ in equations 2.15 and 2.13 are identical apart from a multiplicative factor of 2.

FIG 2.1

FLOWCHART DETAILS FOR ROUTINE GRADNT.



legend over page.......

LEGEND TO FIGURE 2.1 GRADNT FLOWCHART

The small numbers against flowchart blocks correspond to the labels of FORTRAN statements in the computer code listing in Appendix D; other symbols are as follows:

x_{qi} = data points

$y_i^{(l)}$ = data values

$f_i^{(l)}$ = computed values of the model function

$\frac{\partial f_i^{(l)}}{\partial b_j}$ = computed values of the derivative of the model function

g_j = the gradient vector of equation 2.15

b_j = current estimate of the parameters

b'_j = temporary estimate of the parameters

g'_j = correction vector given by equation 2.18

g''_j = temporary value for normalized correction vector

where

$i = 1,\ldots n$

$j = 1,\ldots k$

$l = 1, 2$

$q = 1, 2$

i_n = sequential iteration number

α = step size, see equation 2.14

ϕ = residual sum of squares (current)

ϕ_o = residual sum of squares from previous iteration

θ = angle between successive correction vectors

γ = criterion for θ (here set to 80^o)

d_1, d_2 = constants for calculating new value for α see equation 2.19 (here d_1=0.5 d_2=1.0)

τ, ε = convergence criteria, $\tau = 10^{-15}$, $\varepsilon = 10^{-5}$.

be grossly elongated rather than near-circular, and frequently the gradient vector $\underline{g}$ points almost at right angles to the actual direction of the minimum. This seriously retards the progress of descent algorithms and the situation can often be improved by suitably scaling the parameters, or by a transformation of the parameter space. A transformation used with success by Marquardt (1959) is to replace parameter b_j by

$$b'_j = \arctan b_j \quad , \qquad \ldots\ 2.17$$

whereon the transformed gradient vector components g'_j are given by

$$g'_j = -(1+b_j^2).\partial\phi/\partial b_j \quad . \qquad \ldots\ 2.18$$

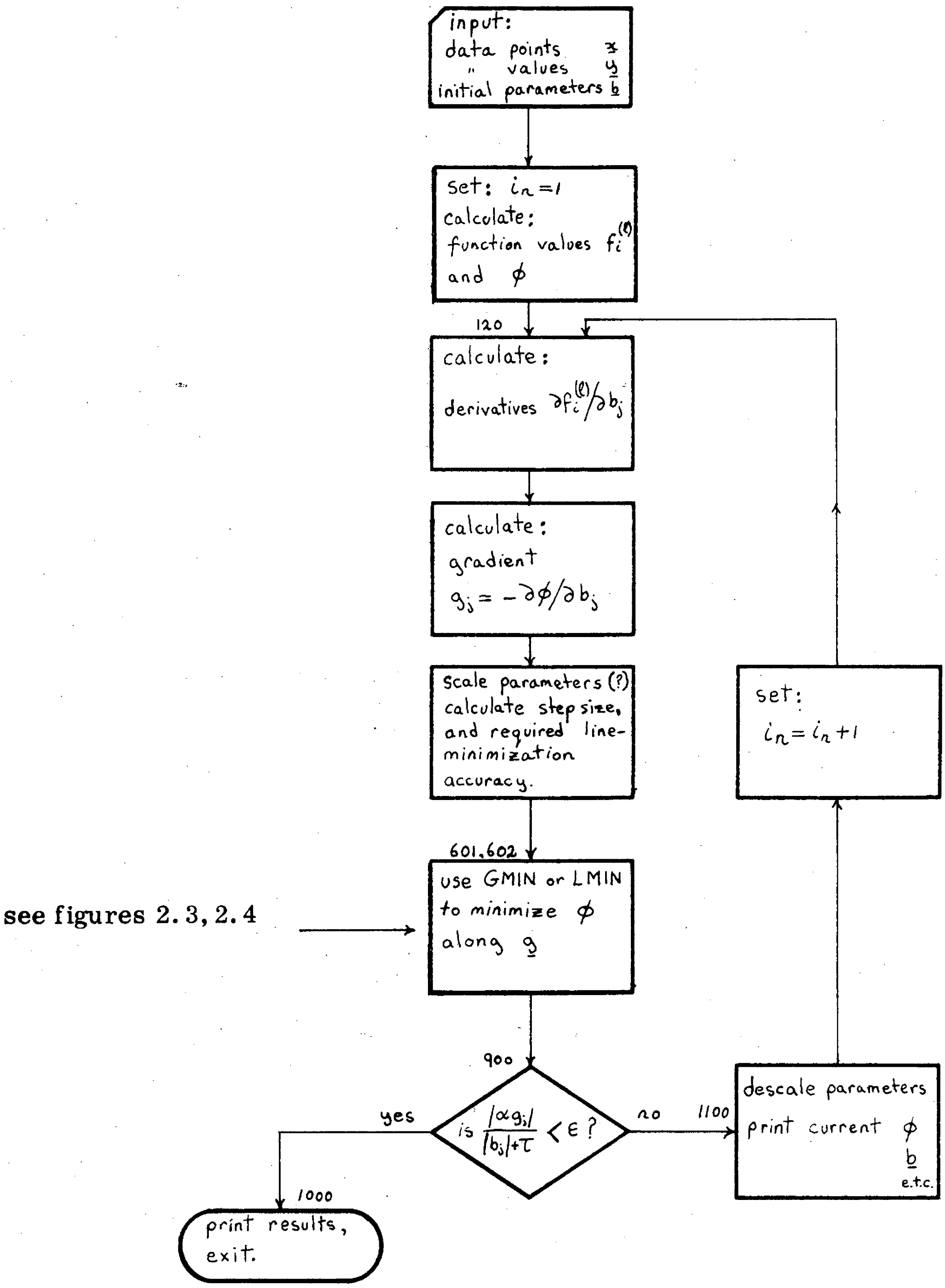
Some strategy must be employed in descent algorithms to determine a suitable value for the step size α. A commonly used one is that if the successive estimates lie approximately on a straight line then α is too small and should be increased for the next iteration, and if the estimates 'zig-zag' acutely α is too large. In an implementation of this by Marquardt (1959) the angle θ between successive correction vectors is calculated; if θ is greater than some criterion angle γ, α is divided by 4 and the new estimate of the parameter vector in equation 2.14 recalculated with this value of α. If θ is less than γ the routine advances to the next iteration and calculates a new value for α from

$$\alpha_{new} = \alpha(d_1 + d_2 \cos^3\theta) \quad , \qquad \ldots\ 2.19$$

where choice of d_1 and d_2 such that $0 < d_1 < 1$, and $(1-d_1) < d_2 \leqslant 1$ results in an increased value of α if θ is near zero, and a decreased value if θ becomes large. A similar scheme is used by Brown et al (1956). A steepest descent algorithm using the parameter transformation and the scheme for determining α given above was implemented by the author, and the detailed strategy of the routine (named GRADNT) is given in the flow-chart in Figure 2.1.

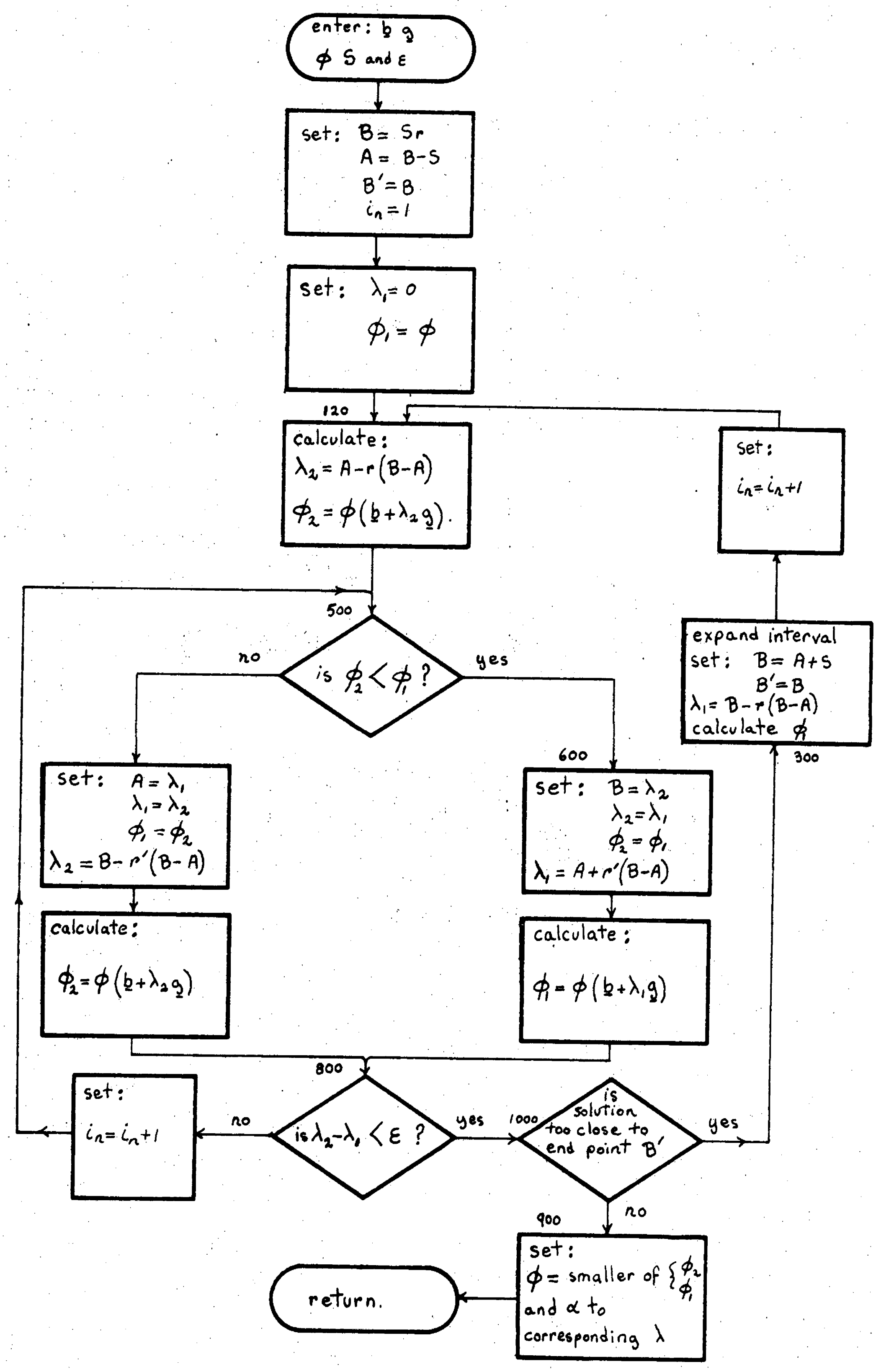
As discussed later in the presentation of numerical results, the strategy used above to regulate step size is still too coarse to ensure stable convergence, and a class of algorithms known as 'optimum gradient' algorithms attempt to line minimize the function along the chosen correction direction. Obviously the routine becomes inefficient if too great an effort is spent in line minimization before a new iteration and new correction direction are introduced, and various compromise strategies are found in the literature. Here, to test the basic strategy of using the steepest descent direction, a version of the steepest descent algorithm employing fairly complete line minimization within an iteration was implemented. Called GRAD2, it was used with both the golden-section and

FLOWCHART DETAILS FOR ROUTINE GRAD2.



legend as for figure 2.1 ....

FLOWCHART FOR GOLDEN-SECTION ROUTINE GMIN.



legend over page.......

LEGEND TO FIGURE 2.3 GMIN FLOWCHART

The small numbers against flowchart blocks correspond to the labels of FORTRAN statements in the computer code listing in Appendix D; other symbols are as follows:

$\phi = \phi(\underline{b})$ current residual sum of squares

$\underline{b} = (b_j)$ current parameter vector

$\underline{g} = (g_j)$ direction of required line minimization

S = step size within which the line minimum of ϕ is expected to lie

ε = line minimization convergence criterion, $\varepsilon = S/10^5$

r = the golden ratio (= 1.61803)

$r' = 1 - r$

$\left.\begin{matrix} A \\ B \end{matrix}\right\}$ = dynamic endpoints of an interval bracketing the minimum $(A < B)$

B' = original value of B

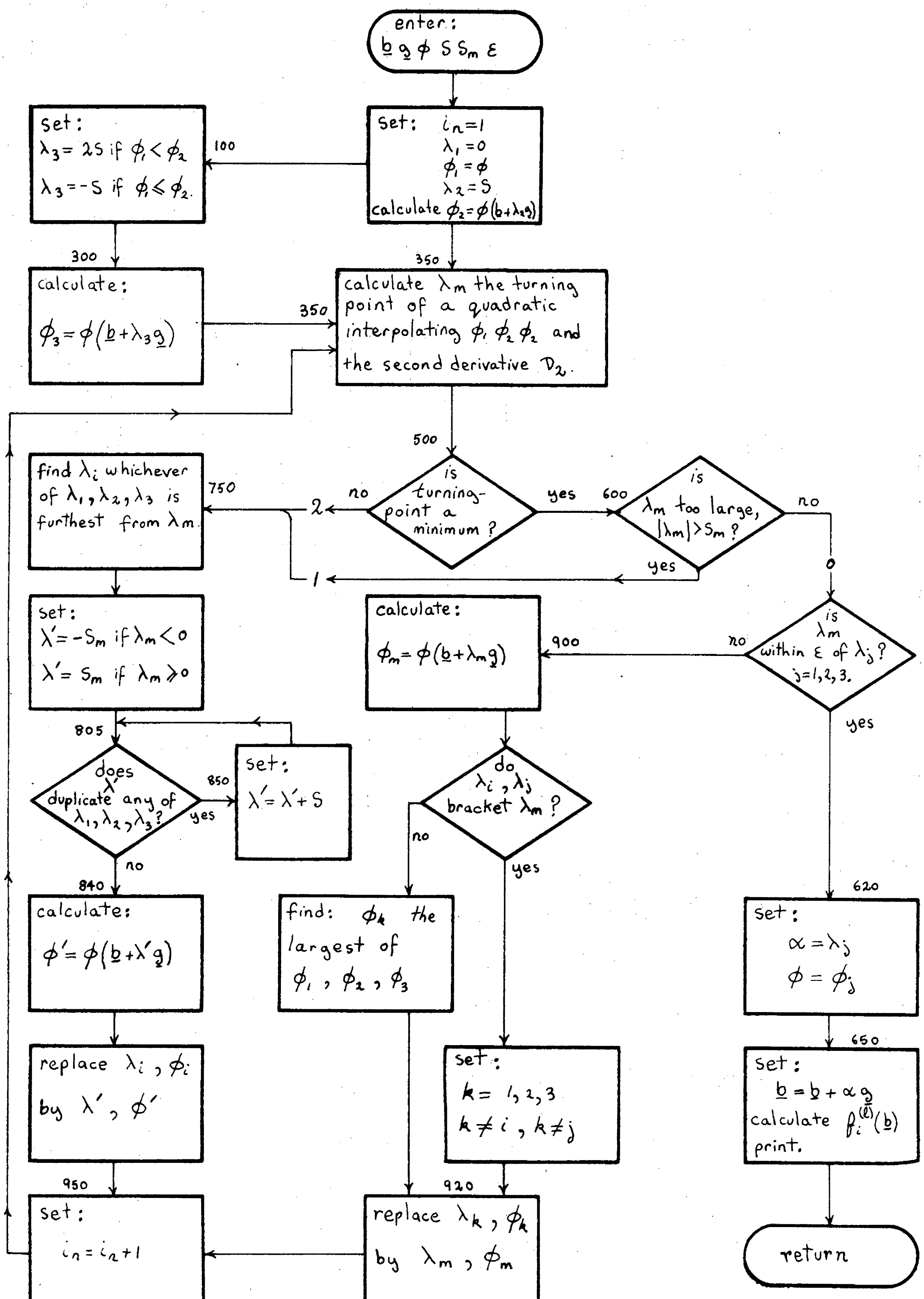
$\left.\begin{matrix} \lambda_1 \\ \lambda_2 \end{matrix}\right\}$ = trial steps along $\underline{g}$, see below $(\lambda_1 < \lambda_2)$

$\left.\begin{matrix} \phi_1 \\ \phi_2 \end{matrix}\right\}$ = evaluations of ϕ at the points $\left\{\begin{matrix} \underline{b} + \lambda_1 \underline{g} \\ \underline{b} + \lambda_2 \underline{g} \end{matrix}\right.$

α = final solution of step size along $\underline{g}$ which is within ε of line minimum of ϕ along $\underline{g}$

i_n = iteration number.

FLOWCHART FOR POWELL ROUTINE LMIN.



legend over page.......

LEGEND TO FIGURE 2.4 LMIN FLOWCHART

The small numbers against flowchart blocks correspond to the labels of FORTRAN statements in the computer code listing in Appendix D; other symbols are as follows:

$\underline{b} = (b_j)$ current parameter vector

$\underline{g} = (g_j)$ direction of required line minimization

$\phi = \phi(\underline{b})$ current residual sumsquare

S = typical working step size

S_m = maximum permissable step along $\underline{g}$

ε = line minimization convergence criterion, $\varepsilon = S/10^5$

$\left.\begin{matrix}\phi_1\\ \phi_2\\ \phi_3\end{matrix}\right\}$ = evaluations of ϕ at the points $\left\{\begin{matrix}\underline{b} + \lambda_1\underline{g}\\ \underline{b} + \lambda_2\underline{g}\\ \underline{b} + \lambda_3\underline{g}\end{matrix}\right.$

$\left.\begin{matrix}\lambda_1\\ \lambda_2\\ \lambda_3\end{matrix}\right\}$ = trial steps along $\underline{g}$ (see above)

λ_m = turning point of a quadratic interpolating ϕ_1, ϕ_2, ϕ_3 computed from

$$\lambda_m = \tfrac{1}{2}.\ \frac{(\lambda_2^2-\lambda_3^2)\phi_1 + (\lambda_3^2-\lambda_1^2)\phi_2 + (\lambda_1^2-\lambda_2^2)\phi_3}{(\lambda_2-\lambda_3)\phi_1 + (\lambda_3-\lambda_1)\phi_2 + (\lambda_1-\lambda_2)\phi_3}$$

D_2 = second derivative of quadratic interpolant computed from

$$D_2 = -2.\ \frac{(\lambda_2-\lambda_3)\phi_1 + (\lambda_3-\lambda_1)\phi_2 + (\lambda_1-\lambda_2)\phi_3}{(\lambda_1-\lambda_2)(\lambda_2-\lambda_3)(\lambda_3-\lambda_1)}$$

ϕ_m = sumsquare corresponding to λ_m

$\lambda_i, \lambda_j, \lambda_k$ = specified members of the set $(\lambda_1, \lambda_2, \lambda_3)$

λ = temporary value of step size

$\phi_i, \phi_j, \phi_k, \phi'$ = corresponding sumsquare values

i_n = iteration number.

Powell line minimization algorithms and is flowcharted in Figure 2.2. The flow details of the golden-section (program name GMIN) and Powell (named LMIN) routines are included for completeness and are seen in Figures 2.3 and 2.4 respectively. The routine LMIN is used also in Powell's (1965) algorithm for minimization of multivariate sums of squares, which is alluded to later.

Neither of the two steepest descent routines GRADNT or GRAD2 proved to be successful in numerical tests described later, and Kowalik and Osborne (1968) note that, even for k greater than 2, such algorithms often eventually approach the solution in a two-dimensional subspace, and can become trapped in a 'cage' whilst working along the level floor of a steep-sided valley. Various methods, for example those due to Davidon (1959), Swann et al (1964) and Powell (1964), have been devised to avoid this problem and employ conjugate directions, that is consecutive searches are conducted in directions in parameter space which are linearly independent; however implementations of them have not been investigated here.

(2.2.2) The Levenberg and Marquardt Algorithms

A number of the more successful algorithms to be found in the literature are based on the Gaussian approach mentioned earlier, and solve the matrix equation of equation 2.10. When the model function is linear in the parameters, equation 2.10 need only be solved once to yield the (unique) leastsquares parameter estimate. Matrix A is positive definite[@] but often ill-conditioned[@@] and certain factorizations of A like Choleski decomposition and the method of Golub (1965) have been found very useful in such cases. However, for nonlinear models the most important consideration is the strategy within an iteration, after equation 2.10 has been solved:- how best to deal with a solution for the correction vector $\underline{t}$ which is only an approximation.

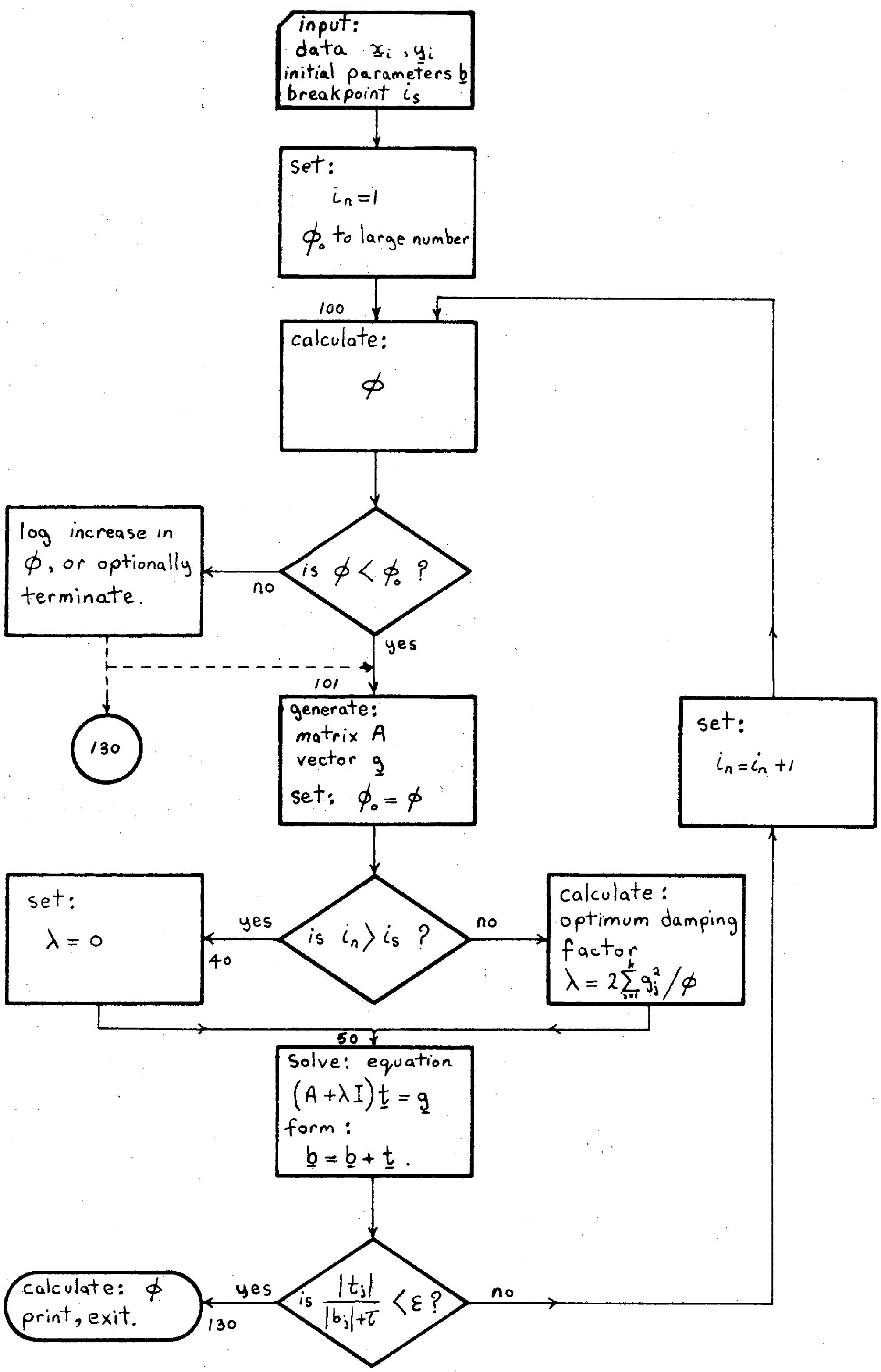
Simply solving equation 2.10 and correcting the current parameter vector by $\underline{t}$, as in equation 2.11, each iteration proves unstable for most problems, and so Levenberg (1944), with the idea of minimizing both ϕ and the length of the correction vector $\underline{t}$ simultaneously, solves a modified form of equation 2.10 viz.,

$$(A + \lambda I)\,\underline{t} = \underline{g} \qquad \ldots\ 2.20$$

@ All the eigenvalues of A are positive.

@@ Small perturbations in the elements of A produce extremely large errors in the numerically calculated inverse matrix A^{-1}.

FLOWCHART FOR LEVENBERG ROUTINE DLSQ.



legend over page.......

LEGEND TO FIGURE 2.5 DLSQ FLOWCHART

The small numbers against flowchart blocks correspond to the labels of FORTRAN statements in the computer code listing in Appendix D; other symbols are as follows:

i_n = sequential iteration number

i_s = iteration number after which damping is removed (break point iteration)

x_{qi} = the n data points

$y_i^{(l)}$ = the n data values

$\underline{b}$ = (b_j) the current estimate of the parameter vector

A = (A_{jr}) the kxk matrix in equation 2.12

$\underline{g}$ = (g_j) the k-vector in equation 2.13

where
$i = 1,\ldots n$
$l = 1, 2$
$j = 1,\ldots k$
$q = 1, 2$

ϕ = the current residual sum of squares (see equation 2.7)

ϕ_o = the value of ϕ saved from previous iteration

τ = 10^{-15} constant preventing division by zero

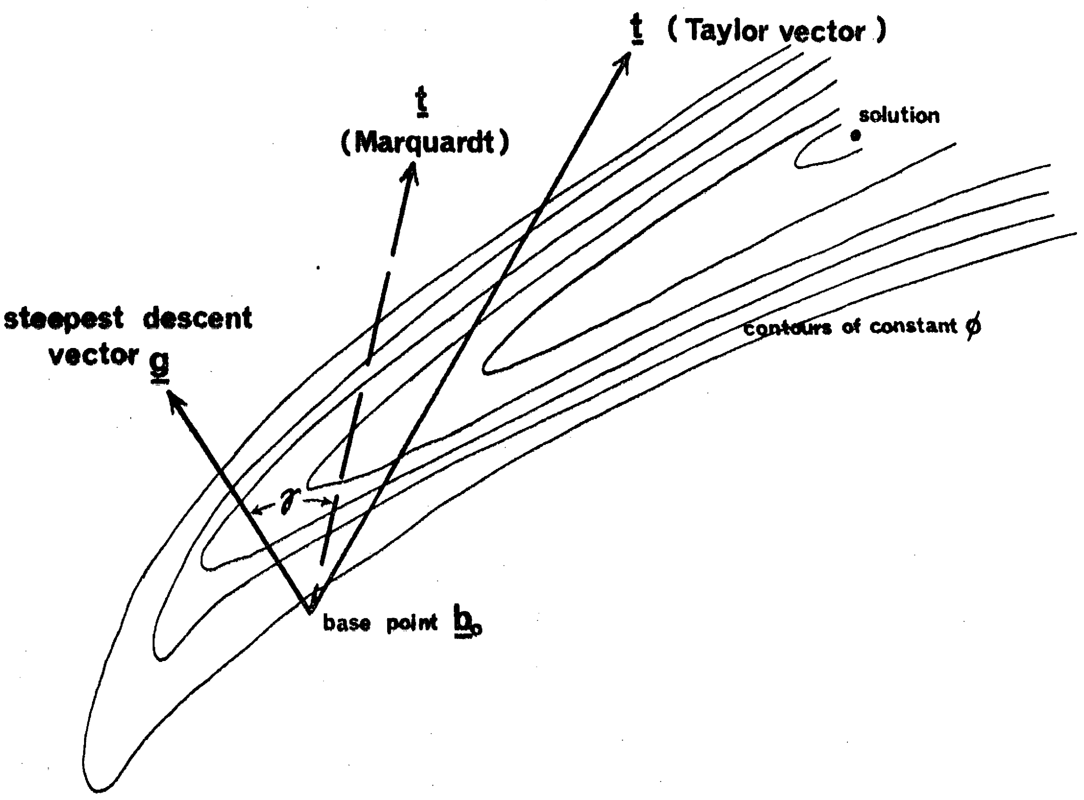
ε = 10^{-5} relative tolerance for convergence criterion

λ = damping factor added to diagonal elements of A (see equation 2.21)

$\underline{t}$ = (t_j) correction vector found from solution of equation 2.20

I = kxk unit matrix.

Typical situation encountered by Gaussian type algorithms.



In this equation, A and $\underline{g}$ are as given in equations 2.12 and 2.13 respectively, I is the unit matrix and λ is a 'damping factor' calculated from

$$\lambda = 2 \sum_{j=1}^{k} g_j^{\,2} / \phi \quad . \qquad \ldots\ 2.21$$

This optimized choice of damping factor λ has the effect of inhibiting the divergence of successive parameter estimates, which would be caused by nonlinearity or poor scaling in the model, but unfortunately markedly decreases the rate of convergence, and so the damping is switched off (λ is set to zero) after a certain number of iterations (called here the breakpoint iteration) in the implementation used here. The Levenberg algorithm was programmed here as routine DLSQ and is flowcharted in Figure 2.5.

A typical situation encountered by Gaussian-type algorithms is that depicted in Figure 2.6; the steepest descent vector $\underline{g}$ may often lie almost perpendicular to the Taylor direction. The algorithm by Marquardt (1963) uses a correction vector $\underline{t}$ which is an interpolation between the Taylor direction and $\underline{g}$ and in their common plane. Marquardt generates scaled matrix $A^* = (A^*_{jr})$ and vector $\underline{g}^* = (g^*_j)$ by

$$A^*_{jr} = A_{jr} / \sqrt{A_{jj} . A_{rr}} \quad , \qquad \ldots\ 2.22^{@}$$

and $$g^*_j = g_j / \sqrt{A_{jj}} \qquad \ldots\ 2.23$$

respectively and, like Levenberg, solves the equation

$$(A^* + \lambda I)\ \underline{t}^* = \underline{g}^* \qquad \ldots\ 2.24$$

The correction vector $\underline{t}$ is obtained from

$$t_j = t^*_j / \sqrt{A_{jj}} \ . \qquad \ldots\ 2.25$$

Marquardt shows that, as λ increases, the angle γ in Figure 2.6 decreases, and $\underline{t}$ rotates so as to approach the steepest descent direction $\underline{g}$. The basic strategy employed for the determination of λ is to increase λ within an iteration until a reduction in ϕ is obtained, but between iterations λ is decreased to ensure fast convergence when approaching the minimum. The Marquardt algorithm in its original form (and in the implementation here) involves re-inversion of the matrix $(A^* + \lambda I)$ whenever λ is changed; Jones (1970) shows that this may be obviated by using a matrix multiplication process involving the eigenvalues and eigenvectors of A^*. One of the advantages of adding λ to the diagonal elements of A^* is that the resulting matrix is always better conditioned than A^* itself, and cannot be singular for any value of λ which is larger

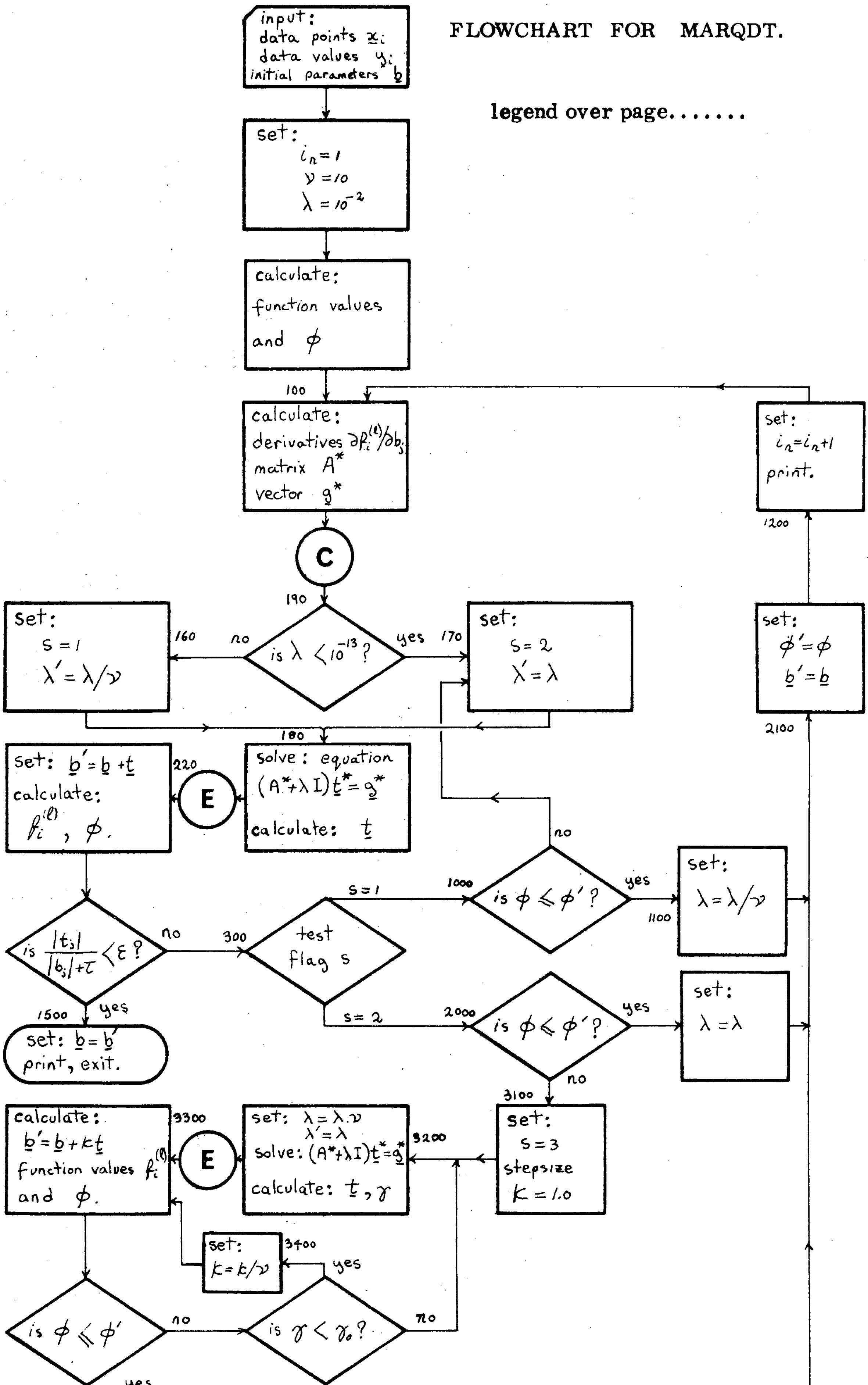
---

@ This produces ones on the leading diagonal of A^* which is effectively the matrix of correlation coefficients between the parameters.

FIG 2.7

FLOWCHART FOR MARQDT.

legend over page.......



LEGEND TO FIGURE 2.7 MARQDT FLOWCHART

The small numbers against flowchart blocks correspond to the labels of FORTRAN statements in the computer code listing in Appendix D; other symbols are as follows:

i_n = sequential iteration number

x_{qi} = data points

$y_i^{(l)}$ = data values

$\underline{b}$ = current estimate of parameter vector

$\underline{b}'$ = temporary parameter vector

A^*, $\underline{g}^*$, $\underline{t}^*$ These quantities result from the scaling of matrix A to give ones on the leading diagonal. They are related to A, $\underline{g}$ and $\underline{t}$ by equations 2.22 to 2.25

where
$i = 1, \ldots n$
$j = 1, \ldots k$
$l = 1, 2$
$q = 1, 2$
$r = 1, \ldots k$

λ = quantity added to diagonal in equation 2.24

λ' = temporary value of λ

ν = 10, constant for reducing λ by division

k = step size

s = flag indicating history of λ within an iteration

γ = angle between the vector $\underline{t}$ and direction of steepest descent

γ_0 = criterion angle (here set at $\pi/4$)

τ = 10^{-15} constant preventing division by zero

ε = 10^{-5} convergence criterion

C, E = points at which CMPRES, EXPAND routines are optionally included (see Chapter 4).

FIG 2.8

The search pattern of Jones' SPIRAL routine.

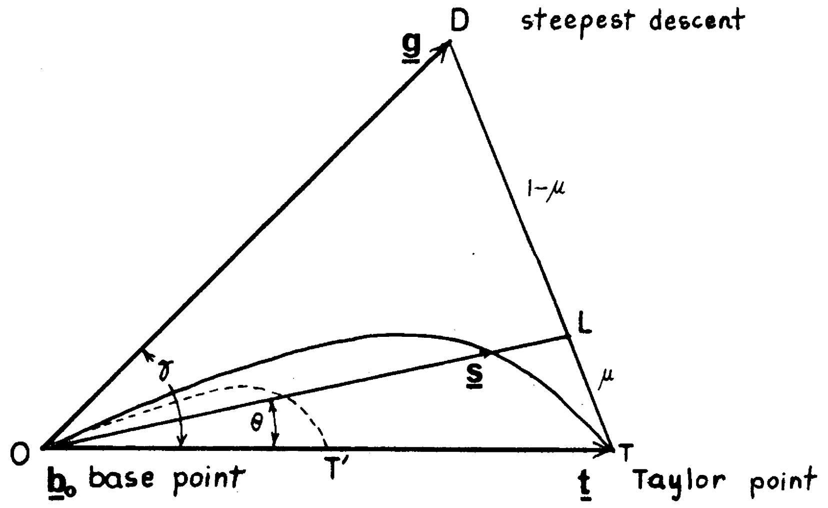
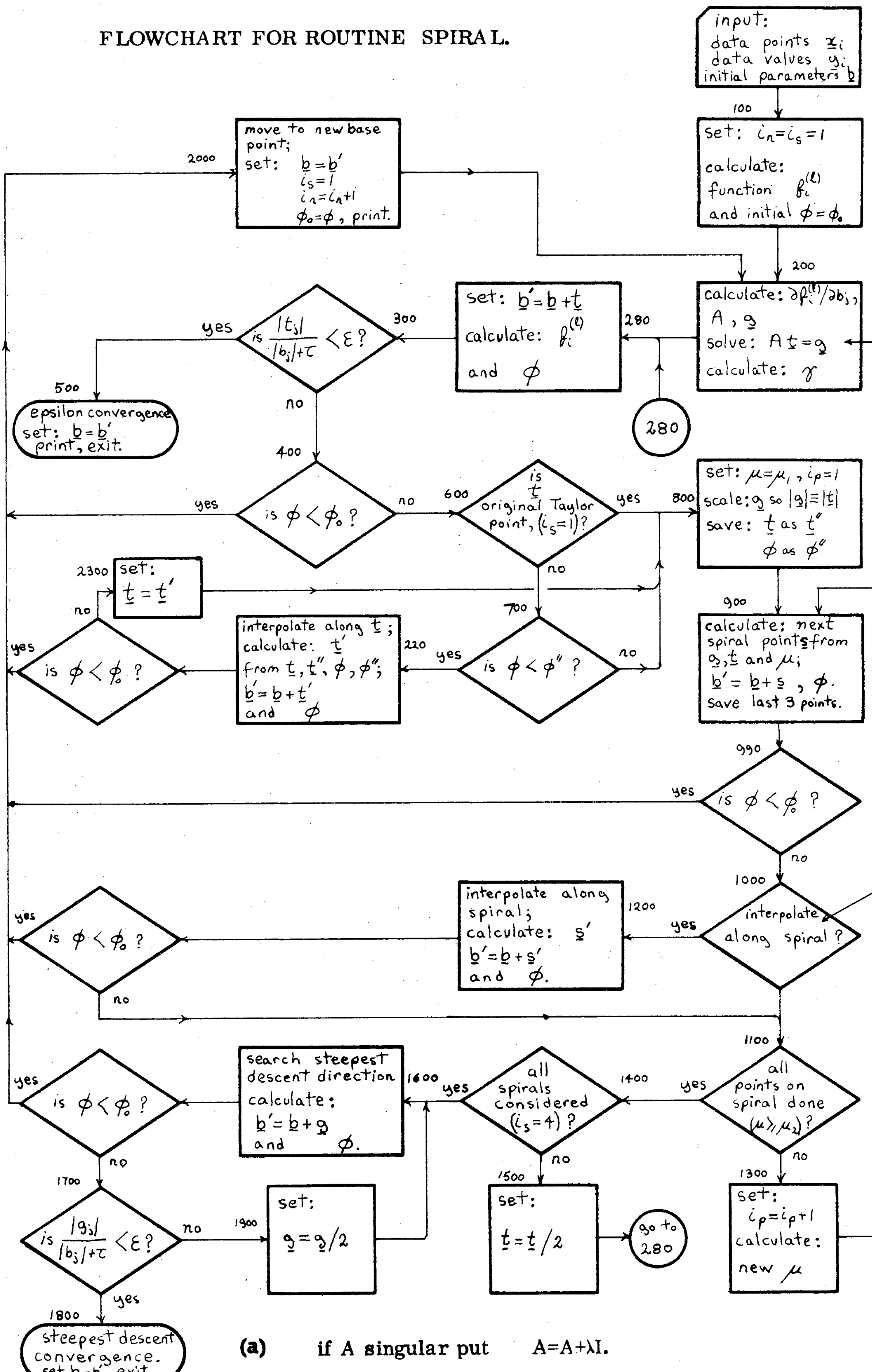


FIG 2.9

FLOWCHART FOR ROUTINE SPIRAL.



(a) if A singular put $A = A + \lambda I$.

(b) is $i_p > 2$ and last 3 points concave?

legend over page........

LEGEND TO FIGURE 2.9 SPIRAL FLOWCHART

The small numbers against flowchart blocks correspond to the labels of FORTRAN statements in the computer code listing in Appendix D; other symbols are as follows:

x_{qi} = data points

$y_i^{(1)}$ = data values

$f_i^{(1)}$ = computed function values

A = kxk matrix of equation 2.12

$\underline{g}$ = k-vector of equation 2.13, and steepest descent vector

$\underline{b}$ = current vector of parameter estimates

$\underline{b}'$ = temporary parameter vector

$\underline{s}$ = k-vector giving point on spiral

$\underline{s}'$ = point on spiral found by interpolating three other points which are downwardly concave in ϕ

$\underline{t}$ = Taylor point found from solving equation 2.10

$\underline{t}''$ = previous Taylor point saved

$\underline{t}'$ = Taylor point found from interpolating along Taylor direction

μ = scalar parameter generated by a recurrence relation to provide consecutive points along the spiral

μ_1 = initial value of μ

μ_2 = final value of μ

i_n = iteration number

i_s = number of spiral being searched

i_p = sequential number of point on spiral

γ = angle between Taylor direction $\underline{t}$ and steepest descent $\underline{g}$

ϕ = current sum of squares

ϕ_o = previous sum of squares

ϕ'' = value of ϕ for previous Taylor point

τ = 10^{-15} constant preventing division by zero for some $b_j=0$

ε = 10^{-5} convergence criterion than the numerical errors. Kowalik and Osborne (1968) are mildly critical of the algorithm noting that it is awkward to experiment with λ within an iteration, and that it is no disadvantage in Gaussian algorithms for $\underline{t}$ to lie well away from $\underline{g}$, since algorithms using the steepest descent direction rarely perform well. Despite this, the version of the Marquardt algorithm used here, called MARQDT and shown in Figure 2.7, behaves extremely well on the model used and in fact spends most of its time pursuing a path nearly perpendicular to $\underline{g}$.

(2.2.3) Jones' SPIRAL Algorithm

Another Gaussian type algorithm investigated here is the routine 'SPIRAL' originated by Jones (1970). Like the Marquardt algorithm it concerns itself with the area between, and in the plane of the steepest descent vector $\underline{g}$ and the Taylor direction $\underline{t}$, as shown in Figure 2.8. Searches for a reduced sum of squares ϕ are made along spirals connecting the base point O with the calculated Taylor point T in the isosceles triangle ODT. The points on the spiral are chosen so that they get closer together as they approach the steepest descent direction. The strategy of the SPIRAL algorithm, which is flowcharted in Figure 2.9 is as follows:

(i) The current Taylor point T is checked for a reduced sum of squares ϕ, and if found the next iteration is entered (the base point shifted to the Taylor point) otherwise,

(ii) points on the spiral curve TO starting from T are checked and if no reduced ϕ is found,

(iii) vector $\underline{t}$ is halved producing point T' and if this gives no reduction in ϕ the spiral OT' is searched as above;

(iv) if four spirals have been searched without finding a reduced ϕ the steepest descent direction is searched.

Jones also checks for downward concavity of ϕ for points on the spiral, and if this is encountered an interpolation is performed; he also compares the value of ϕ at the original and halved Taylor points for possible interpolation along the Taylor direction. Whereas in MARQDT consecutive search points within an iteration are generated by matrix inversion, the search points $\underline{s}$ in SPIRAL are generated by vector addition.

$$\underline{s} = D\underline{g} + T\underline{t} \quad , \qquad \ldots\ 2.26$$

where $\underline{s}$ is the search point on the spiral and D and T are scalar functions of a parameter μ , the index of the spiral point, which is generated by the recurrence relation

$$\mu_{next} = 2\mu/(1+\mu) \qquad \ldots\ 2.27$$

As soon as a satisfactory point $\underline{s}$ is found, the current parameter estimate $\underline{b}_0$ is updated by

$$\underline{b} = \underline{b}_0 + \underline{s} \qquad \ldots\ 2.28$$

(2.2.4) Algorithms which do not Require Derivatives

All the algorithms so far described require the analytic calculation of the model derivatives with respect to the parameters, which for complicated models can be extremely onerous. A few methods exist which require only function values. Fletcher in a review article (Fletcher 1965) compares three such algorithms by Swann et al (1964), Smith (1962) and Powell (1965). Possibly the best for our purpose would be the Powell algorithm which has found wide application. Powell initially uses the co-ordinate directions (in parameter space) as k linearly independent search directions, and, as in the Gaussian methods, solves the normal equations for the correction vector $\underline{d}$ but using numerically estimated derivatives. His 1964 line minimization algorithm (which is discussed earlier by the name LMIN and shown in Figure 2.4) is used to minimize ϕ along $\underline{d}$ and the direction $\underline{d}$ replaces one of the existing set of search directions. The necessary function evaluations along $\underline{d}$ are used to estimate derivatives in such a way that subsequent iterations do not require an excessive number of further evaluations.

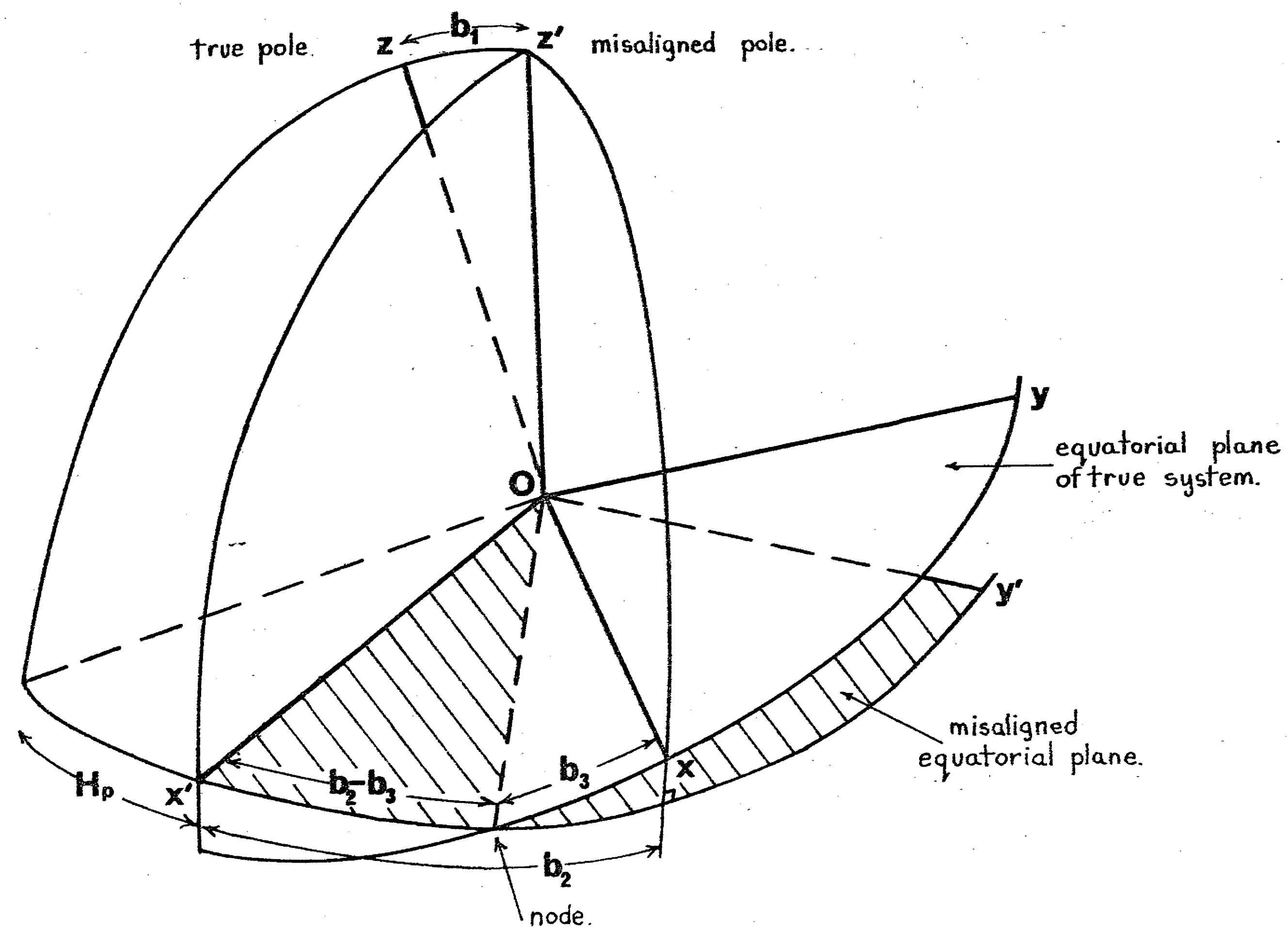
Unfortunately, an implementation of this algorithm was not developed in time for the numerical comparisons later in the chapter, and here we simply note the comparison of it with the Marquardt and SPIRAL algorithms in Jones (1970); in the majority of the standard test problems tried, it proved satisfactory but inferior to the two last mentioned methods. There is no reason to suspect that the model function used here would produce a comparison differing greatly to Jones', particularly since his comparison of SPIRAL and Marquardt does not disagree greatly with our own results. To assess the necessity of analytically calculated derivatives, a version of the Marquardt algorithm using numerically estimated derivatives was implemented, and is hereafter referred to as MARQT2. As in Jones (1970), a simple finite difference formula was used for the derivative estimations, with the step size e_j in each component b_j of $\underline{b}$ being given by

$$e_j = 5 \times 10^{-5}.|b_j| , \qquad \ldots\ 2.29$$

or $$e_j = 5 \times 10^{-5} \qquad \ldots\ 2.30$$

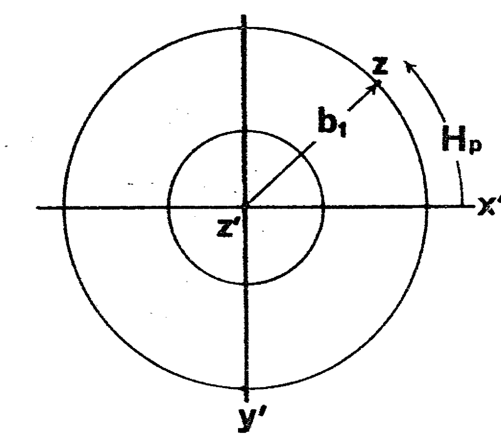
if equation 2.29 produced $e_j < 10^{-10}$.

Effect of Polar-misalignment and Hour-angle offset;



(A)

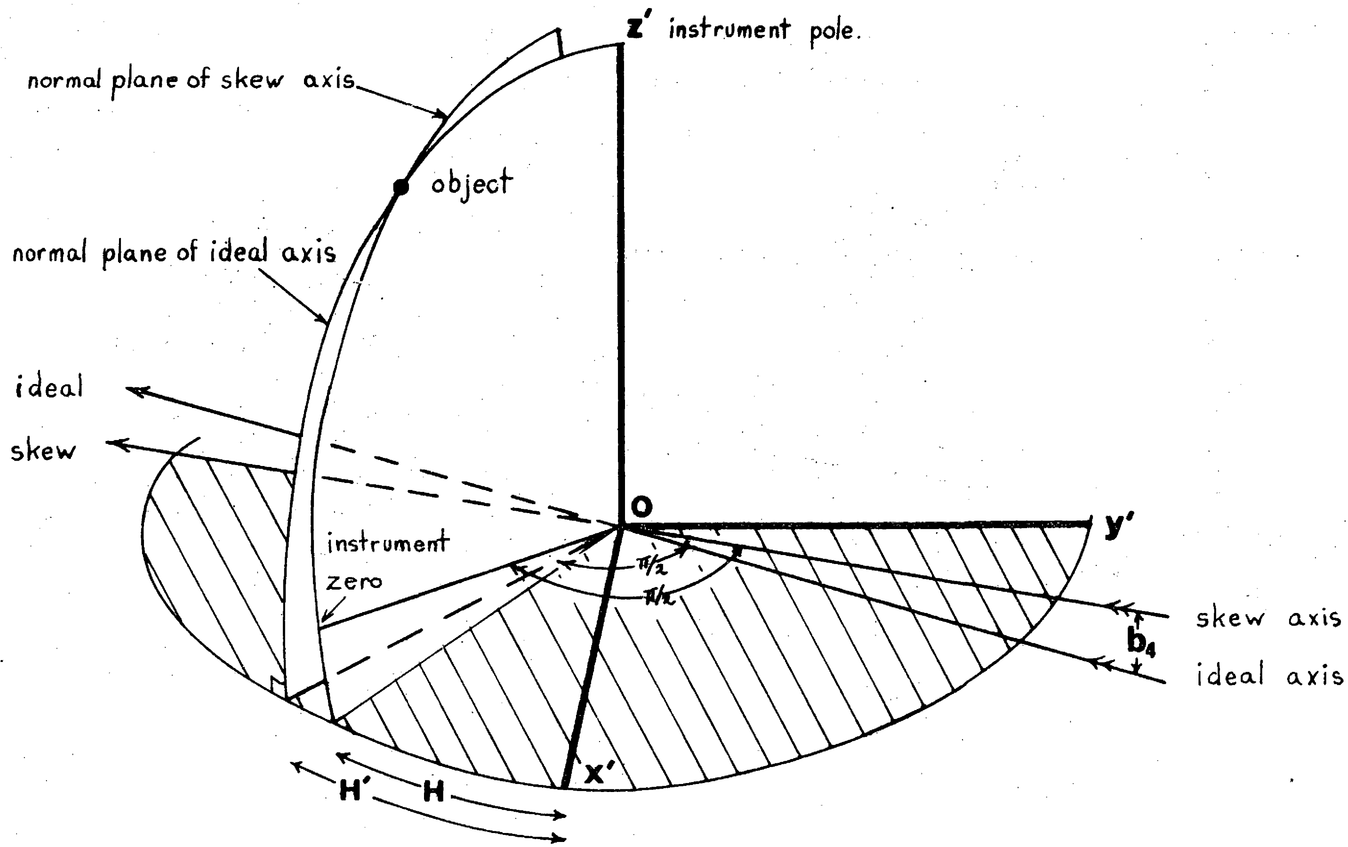
Looking along instrument pole Oz'.



(B)

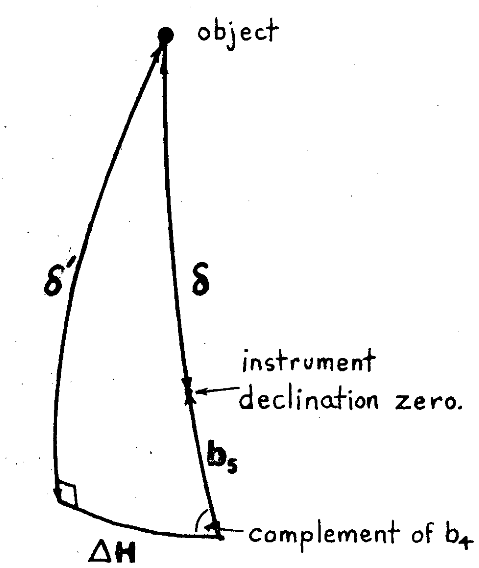
FIG 2.10

Effect of skewness and declination offset;



(A)

Section of part A.



(B)

FIG 2.11

(2.3) THE TELESCOPE POINTING ERROR MODEL

(2.3.1) Model Function

The model devised to test the methods of parameter estimation programmed is that of a perfectly rigid, equatorially mounted telescope with the instrument pole misaligned from the celestial pole, the two axes skew (not orthogonal) and fiduciary errors (zero offsets) in both axis transducers; it is described by five parameters b_1, b_2, b_3, b_4 and b_5 defined below. It is sufficiently simple to permit ease of experimentation with the appropriate computer programs yet useful in as much that it can be incorporated into more complicated models for actual pointing error investigations. It is representative of the type of functions to be expected in pointing error work.

The effect of polar misalignment and hourangle fiduciary error is shown in Figures 2.10a and 2.10b. Thus far the set of axes Ox'y'z' are still orthogonal, and the transformation between the correct equatorial system Oxyz and the misaligned system Ox'y'z' is given by the Eulerian angle transform using the first three of the parameters b_1, b_2 and b_3. With regard to quantities which are physically measureable on the telescope, b_1 is the polar misalignment, b_2 is the correct hourangle zero point with respect to the instrument zero point, and H_p is the hourangle of the true pole with respect to the instrument axes Ox'y'z' and is related to b_3 by

$$H_p = \pi/2 - b_3 \qquad \ldots\ 2.31$$

The remaining two parameters, b_4, the skewness or departure from orthogonality of the axes, and b_5, the declination offset are explained by Figures 2.11a and 2.11b. The skew declination axis is assumed to lie in the plane of the ideal (orthogonal) axis and the instrument pole z', with no loss of generality, since this can always be arranged by suitable choice of b_2, the hourangle offset.

To compute the declination and hourangle components of the model function δ_c and H_c respectively, the axis readout co-ordinates δ and H are first corrected for declination offset and skewness by

$$\left.\begin{aligned} \delta' &= \arcsin\left[\sin(\delta+b_5)\cos b_4\right], \\ \text{and}\quad H' &= H + \arctan\left[\tan(\delta+b_5)\sin b_4\right]. \end{aligned}\right\} \ldots\ 2.32$$

Polar misalignment and hourangle offset are then corrected by the Eulerian angle transformation equations:

$$x = \cos\delta'.\left[\cos(b_2-b_3).\cos(b_3+H') - \cos b_1.\sin(b_2-b_3).\sin(b_3+H')\right] + \sin\delta'.\sin b_1.\sin(b_2+b_3) \quad , \qquad \ldots 2.33a$$

$$y = -\cos\delta'.\left[\sin(b_2-b_3).\cos(b_3+H') + \cos b_1.\cos(b_2-b_3).\sin(b_3+H')\right] + \sin\delta'.\sin b_1.\cos(b_2+b_3) \quad , \qquad \ldots 2.33b$$

$$z = \cos\delta'.\sin b_1.\sin(b_3+H') + \sin\delta'.\cos b_1 \quad . \qquad \ldots 2.33c$$

Finally, the model function is given by the rectangular to polar transformation

$$\left.\begin{aligned} f^{(1)} &= \delta_c = \arcsin(z) \ , \\ \text{and} \quad f^{(2)} &= H_c = \arctan(-y/x) \ , \end{aligned}\right\} \ldots 2.34$$

where $-\pi/2 \leqslant \delta_c \leqslant \pi/2$ and $0 \leqslant H_c < 2\pi$.

(2.3.2) The Model Derivatives

The routines GRADNT, GRAD2, DLSQ, MARQDT and SPIRAL all require the derivatives of the model function with respect to the parameters. Jones (1970) and others regard the analytic calculation of derivatives as being well worthwhile, although they consider model functions somewhat less complicated than that used here. Techniques for simplifying and approximating telescope pointing error models by simply summing the component causes are discussed in Chapter 4, but here the exact model derivatives

$$\frac{\partial f_i^{(1)}}{\partial b_j} = \left\{\frac{\partial\delta_c}{\partial b_1}, \frac{\partial\delta_c}{\partial b_2}, \frac{\partial\delta_c}{\partial b_3}, \frac{\partial\delta_c}{\partial b_4}, \frac{\partial\delta_c}{\partial b_5}, \frac{\partial H_c}{\partial b_1}, \frac{\partial H_c}{\partial b_2}, \frac{\partial H_c}{\partial b_3}, \frac{\partial H_c}{\partial b_4}, \frac{\partial H_c}{\partial b_5}\right\} \ldots 2.35$$

are calculated analytically.

Equations 2.36 to 2.40 inclusive and equation 2.42 below (in which δ', H' and x, y, z are defined in equations 2.32 and 2.33 respectively) give these derivatives:

$$\partial f^{(1)}/\partial b_1 = \left[\cos\delta'.\sin(b_3+H').\cos b_1 - \sin\delta'.\sin b_1\right].(1-z^2)^{-\frac{1}{2}} \qquad \ldots 2.36$$

$$\partial f^{(1)}/\partial b_2 = 0 \qquad \ldots 2.37$$

$$\partial f^{(1)}/\partial b_3 = \cos\delta'.\sin b_1.\cos(b_3+H').(1-z^2)^{-\frac{1}{2}} \qquad \ldots 2.38$$

$$\partial f^{1}/\partial b_4 = \left[\cos\delta'.\cos b_1 - \sin\delta'.\sin b_1.\sin(b_3+H')\right].(1-z^2)^{-\frac{1}{2}}.\partial\delta'/\partial b_4 + \cos\delta'.\sin b_1.\cos(b_3+H').(1-z^2)^{-\frac{1}{2}}.\partial H'/\partial b_4 \qquad \ldots 2.39$$

$$\partial f^{(1)}/\partial b_5 = \left[\cos\delta'.\cos b_1 - \sin\delta'.\sin b_1.\sin(b_3+H')\right].(1-z^2)^{-\frac{1}{2}}.\partial\delta'/\partial b_5 + \cos\delta'.\sin b_1.\cos(b_3+H').(1-z^2)^{-\frac{1}{2}}.\partial H'/\partial b_5 \qquad \ldots 2.40$$

In equations 2.39 and 2.40 the derivatives of δ' and H' are given by

$$\partial\delta'/\partial b_4 = -\sin(\delta+b_5).\sin b_4.\left[1 - \sin^2(\delta+b_5).\cos^2 b_4\right]^{-\frac{1}{2}} \quad \ldots\ 2.41a$$

$$\partial\delta'/\partial b_5 = \cos(\delta+b_5).\cos b_4.\left[1 - \sin^2(\delta+b_5).\cos^2 b_4\right]^{-\frac{1}{2}} \quad \ldots\ 2.41b$$

$$\partial H'/\partial b_4 = \tan(\delta+b_5).\cos b_4\left[1 + \tan^2(\delta+b_5).\sin^2 b_4\right]^{-1} \quad \ldots\ 2.41c$$

and $$\partial H'/\partial b_5 = \sin b_4.\sec^2(\delta+b_5).\left[1 + \tan^2(\delta+b_5).\sin^2 b_4\right]^{-1} \quad \ldots\ 2.41d$$

The derivatives $\partial f^{(2)}/\partial b_j$ are given by equations of the form

$$\partial f^{(2)}/\partial b_j = (y\partial x/\partial b_j - x\partial y/\partial b_j)/(x^2 + y^2) \quad \ldots\ 2.42$$

where $j = 1,2,3,4,5$ and the derivatives $\partial x/\partial b_j$ and $\partial y/\partial b_j$ are in turn given by

$$\begin{aligned}\partial x/\partial b_1 = {} & \cos\delta'.\sin b_1.\sin(b_2-b_3)\ \sin(b_3+H') \\ & + \sin\delta'.\cos b_1.\sin(b_2-b_3)\end{aligned} \quad \ldots\ 2.43a$$

$$\begin{aligned}\partial y/\partial b_1 = {} & \cos\delta'.\sin b_1.\cos(b_2-b_3).\sin(b_3+H') \\ & + \sin\delta'.\cos b_1.\cos(b_2-b_3)\end{aligned} \quad \ldots\ 2.43b$$

$$\begin{aligned}\partial x/\partial b_2 = {} & -\cos\delta'.\left[\cos(b_3+H').\sin(b_2-b_3). + \cos b_1.\cos(b_2-b_3)\ \sin(b_3+H')\right] \\ & + \sin\delta'.\sin b_1.\cos(b_2-b_3)\end{aligned} \quad \ldots\ 2.43c$$

$$\begin{aligned}\partial y/\partial b_2 = {} & \cos\delta'.\left[\cos b_1.\sin(b_2-b_3).\sin(b_3+H') - \cos(b_2-b_3).\cos(b_3+H')\right] \\ & - \sin\delta'.\sin b_1.\sin(b_2-b_3)\end{aligned} \quad \ldots\ 2.43d$$

$$\begin{aligned}\partial x/\partial b_3 = {} & -\cos\delta'.\left[\cos(b_2-b_3).\sin(b_3+H') + \cos b_1.\sin(b_2-b_3).\cos(b_3+H')\right] \\ & - \partial x/\partial b_2\end{aligned} \quad \ldots\ 2.43e$$

$$\begin{aligned}\partial y/\partial b_3 = {} & \cos\delta'.\left[\sin(b_2-b_3).\sin(b_3+H') - \cos b_1.\cos(b_2-b_3).\cos(b_3+H')\right] \\ & - \partial y/\partial b_2\end{aligned} \quad \ldots\ 2.43f$$

$$\partial x/\partial b_4 = A.\partial\delta'/\partial b_4 + B.\partial H'/\partial b_4 \quad \ldots\ 2.43g$$

$$\partial y/\partial b_4 = C.\partial\delta'/\partial b_4 + D.\partial H'/\partial b_4 \quad \ldots\ 2.43h$$

$$\partial x/\partial b_5 = A.\partial\delta'/\partial b_5 + B.\partial H'/\partial b_5 \quad \ldots\ 2.43i$$

$$\partial y/\partial b_5 = C.\partial\delta'/\partial b_5 + D.\partial H'/\partial b_5 \quad \ldots\ 2.43j$$

The quantities A, B, C and D in equations 2.43g to 2.43j are given by

$$\begin{aligned}A = {} & \sin\delta'.\left[\cos b_1.\sin(b_2-b_3).\sin(b_3+H') - \cos(b_2-b_3).\cos(b_3+H')\right] \\ & + \cos\delta'.\sin b_1.\sin(b_2-b_3)\end{aligned} \quad \ldots\ 2.44a$$

$$B = -\cos\delta'.\left[\cos(b_2-b_3).\sin(b_3+H') + \cos b_1.\sin(b_2-b_3).\cos(b_3+H')\right] \quad \ldots\ 2.44b$$

$$C = \sin\delta' \cdot \left[\sin(b_2-b_3)\cdot\cos(b_3+H') + \cos b_1\cdot\cos(b_2-b_3)\cdot\sin(b_3+H')\right] + \cos\delta'\cdot\sin b_1\cdot\cos(b_2-b_3) \quad \ldots\ 2.44c$$

and
$$D = \cos\delta' \cdot \left[\sin(b_2-b_3)\cdot\sin(b_3+H') + \cos b_1\cdot\cos(b_2-b_3)\cdot\cos(b_3+H')\right] \quad \ldots\ 2.44d$$

where again the derivatives of δ' and H' are defined in equations 2.41a to 2.41d.

(2.4) THE PERFORMANCE AND COMPARISON OF THE ALGORITHMS

The six routines GRADNT, GRAD2, DLSQ, MARQDT, SPIRAL and MARQT2 described above were programmed in FORTRAN as subroutines for an I.B.M. 360/50 and later a UNIVAC 1108 computer; the code listings for these routines and also the necessary supporting subroutines are to be found in Appendix D. The parameter estimation program takes the form of a main section which merely reads a card containing the sequence in which the various subroutines above are to be executed. Two additional subroutines are required: DATGEN which, given n the number of data points and $\underline{b}$ a k-vector of telescope mounting parameters, generates pointing error data x_{li} (the independent variable) and $y_i^{(l)}$ (the 'experimentally measured' variable) where l = 1, 2 and i = 1,...n; and also DAPERT which takes the experimental variables $y_i^{(l)}$ above and perturbs them by adding to each a normally distributed pseudo-random@ number with zero mean and a specified standard deviation. These last two subroutines enable one to synthesize pointing error data such as would be taken from a telescope with appropriate parameter vector $\underline{b}$, and to superimpose on this a pseudo-random variation to permit assessment of an algorithm under real conditions, and also to test whether or not an algorithm produces <u>stable</u> solutions.

In the following computing runs several sets of model parameters were used and these are tabulated in Table 2.1. Labelled from A to F the models represent progressively more erratic telescope mountings. Model D (for example) represents a telescope mounting with a polar misalignment (b_1) of 30.9 arcsecond oriented at an hourangle ($\pi/2 - b_3$) of 245 degrees, a polar axis zero error (b_2) of -103 arcsecond, 41.2 arcsecond skewness of the axes (b_4), and a declination zero error (b_5) of 103 arcsecond. Such a mounting produces a R.M.S. pointing error of approximately 2 arcminute.

All of the algorithms require an initial estimate of the parameters

---

@ 'random' but for the fact that repeated computer runs would produce an identical set of such numbers; this is necessary for purposes of comparison.

TABLE 2.1

Test Model		A	B	C	D	E	F
	b_1	.00004	.0001	.00013	.00015	.0003	.01
Parameter	b_2	-.00006	.0002	.0002	-.0005	.0008	-.01
value in	b_3	.6	1.0	.14	-2.0005	1.0	-1.0
radian	b_4	-.00003	-.00015	-.0002	.0002	-.0006	.01
	b_5	.00005	-.0002	.00017	.0005	.0007	-.01
Sum of squares	ϕ	$.26^{-6}$	$.17^{-5}$	$.35^{-5}$	$.10^{-4}$	$.47^{-4}$	$.10^{-1}$
R.M.S. on-sky error		19 sec.	50 sec.	70 sec.	2 min.	4 min.	1 deg.

TABLE 2.2

Starting Point Number		SP1	SP2	SP3	SP4	SP5	SP6
	b_1	10^{-7}	.002	.01	0	-10^{-5}	-.00015
Parameter	b_2	10^{-7}	.002	.01	0	-10^{-5}	.0005
value in	b_3	10^{-7}	.002	.01	0	-10^{-5}	2.0005
radian	b_4	10^{-7}	.002	.01	0	-10^{-5}	-.0002
	b_5	10^{-7}	.002	.01	0	-10^{-5}	.0005
Sum of squares	ϕ	$.77^{-12}$	$.31^{-3}$	$.78^{-2}$	0	$.77^{-8}$	$.11^{-4}$
R.M.S. on-sky error		.03 sec.	11 min.	50 min.	0	3 sec.	2 min.

N.B. superscripts are decimal exponents e.g. $.26^{-6} = .26 \times 10^{-6}$

which is iteratively improved to obtain the eventual solution. The six initial estimates or starting points used in the computing runs are tabulated in Table 2.2 and referred to hereafter by SP1 etc. The residual sumsquare ϕ in Tables 2.1 and 2.2 has been calculated from equation 2.7 with x_{1i} substituted for $f_i^{(1)}$ (since $f_i^{(1)} = x_{1i}$ for a perfect mounting) and the R.M.S. on-sky error calculated from

$$\text{R.M.S.} = \sqrt{\phi/n}\ . \qquad \ldots\ 2.45$$

In all runs 30 data points were used and so the constants appearing in equations 2.7 to 2.13 above are n=30, m=2, k=5 and s=2.

Although problems were experienced in getting all the algorithms to run reliably, most of these were with certain constants or with minor points of strategy. The exceptions are the two steepest descent routines GRADNT and GRAD2 which, despite quite drastic modifications and redesigning, proved quite unsatisfactory as practical methods. The expression for the step size α in routine GRADNT given in Marquardt (1959) involves $\cos^4\theta$ and is in error since as the successive steps turn from being collinear to zig-zag, θ varies from 0 through $\pi/2$ to π, and $\cos\theta$ from +1 through 0 to -1. Thus an expression involving an odd integral exponent of $\cos\theta$ is necessary if α is to decrease as θ increases, and expressions with $\cos\theta$ and $\cos^5\theta$ were used here.

The choice of initial step size, the parameter transformation used, and the fundamental strategy of GRADNT were varied with little success. In all runs examined the routine exhibits the same tendency, namely a slow and steady reduction in ϕ along a straight path until a sudden sharp descent or a bend in the contours of ϕ is encountered, whereon the routine 'zig-zags' abruptly and in the attempt to find a further reduction in ϕ the step size is reduced ad nauseam until a floating point divide underflow occurs. Table 2.3 shows the progress of GRADNT starting at SP1 with data generated from model D. At iteration number 8 there is a sudden bend in the contours as shown by the decrease in $\cos\theta$; the routine then keeps dividing α by 4 but does not find a reduced ϕ before underflow occurs. If we limit the number of times α is divided by 4, in this case to 12, the column labelled 9 describes the result; the routine has run up the side of a valley and is proceeding in a straight line with such diminished steps that it cannot find the valley floor in a convenient number of iterations. No modification to the manner in which α is computed that was tried produced any substantial difference in the behaviour of GRADNT.

Like GRADNT, the routine GRAD2 runs into troubles early in the course of the solution. Which of the golden section (GMIN) and Powell (LMIN)

TABLE 2.3

The behaviour of GRADNT

Iteration number	1	2	3	4	5	6	7	8	9@
step size α	$.10^{-2}$	$.50^{-3}$	$.75^{-3}$	$.11^{-2}$	$.17^{-2}$	$.25^{-2}$	$.37^{-2}$	$.20^{-3}$	$.59^{-12}$
$\cos\theta$	–	.9996	.9999	.9995	.9981	.9982	.7784	.0656	.9945
ϕ	$.94^{-5}$	$.88^{-5}$	$.80^{-5}$	$.67^{-5}$	$.50^{-5}$	$.31^{-5}$	$.34^{-5}$	$.72^{-7}$	$.21^{-6}$
R.M.S.	115.	112.	107.	97.5	84.2	66.3	69.4	10.1	16.8

N.B. superscripts are decimal exponents e.g. $.75^{-3} = .75 \times 10^{-3}$

@ see text.

TABLE 2.4

The performance of GMIN and LMIN in the routine GRAD2

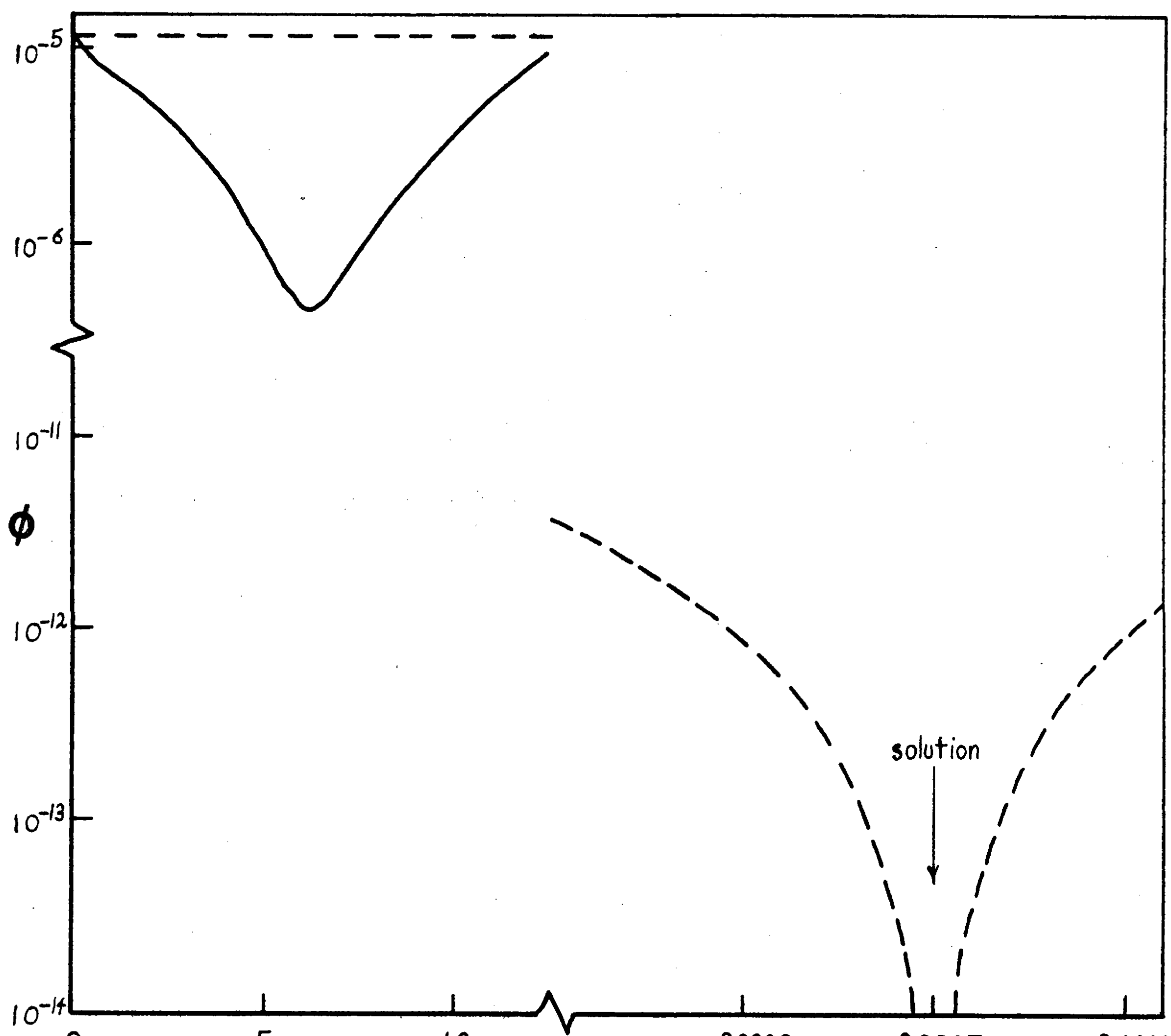
GRAD2 iteration number		1	2	3	4	
ϕ (radian)		$.46^{-6}$	$.74^{-7}$	$.57^{-7}$	$.56^{-7}$	
R.M.S. (arcsecond)		25.5	10.2	9.0	8.9	
Number of function evaluations	GMIN	23	45	67	89	
	LMIN	7	13	17	21	

N.B. superscripts are decimal exponents e.g. $.46^{-6} = .46 \times 10^{-6}$

FIG 2.12

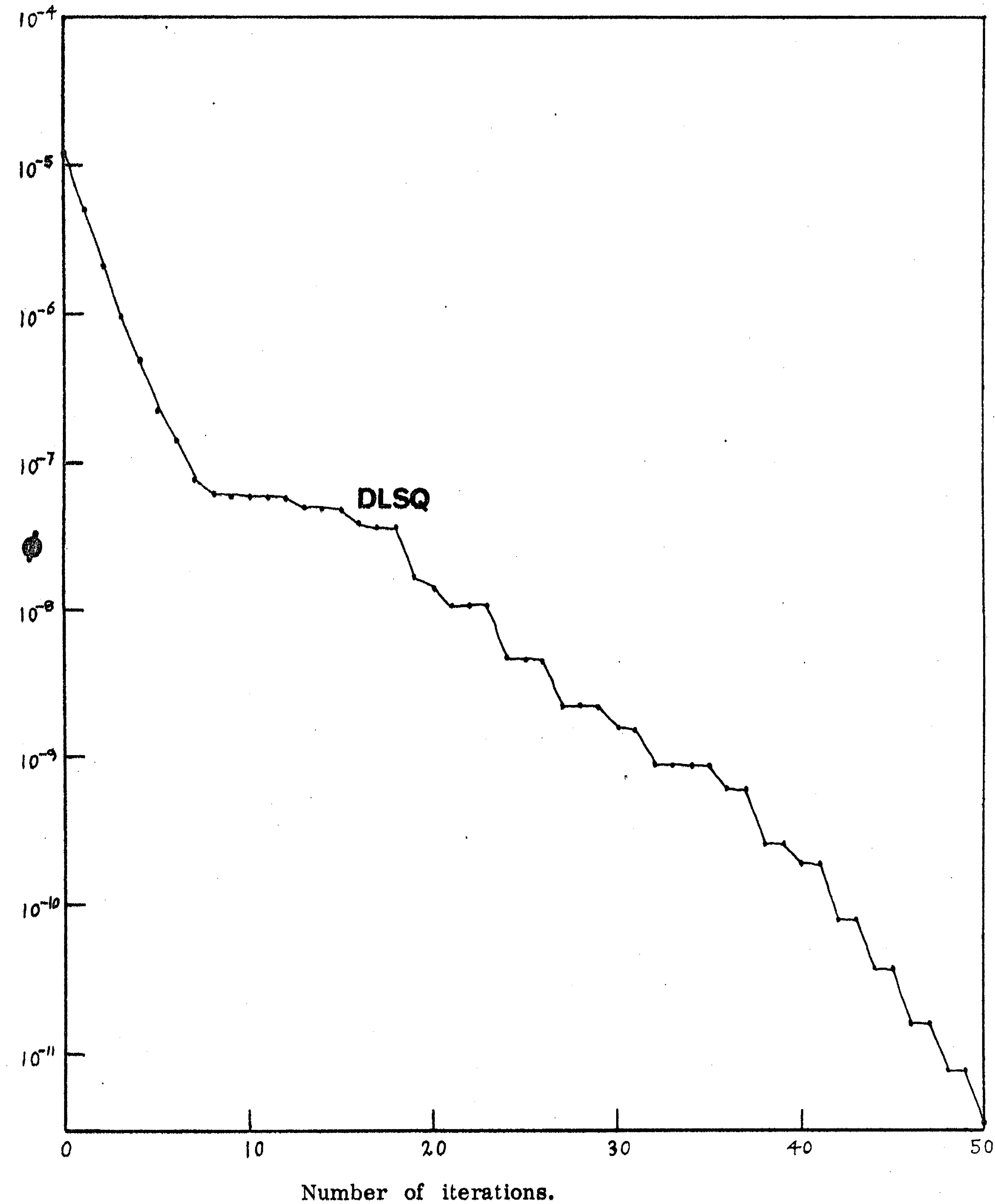
— — — — ϕ in the direction of the correct result for Model D.

———— ϕ in the direction of steepest descent.



Distance in parameter space from point SP1, radian x 10^{-4} .

DLSQ on Model D starting at SP4, $i_s = 50$.



line minimization routines is used, has very little effect on the progress of GRAD2, which, although exhibiting a much faster initial rate of convergence than GRADNT, stagnates at more or less the same value of ϕ, in this case $.56 \times 10^{-7}$ (equivalent to 8.9 arcsecond R.M.S.). There is however a marked difference in efficiency between GMIN and LMIN; the latter requires much fewer function evaluations to perform the line minimization and is much less critical of the step size and line minimization tolerance ε used. Table 2.4 shows the relative number of function evaluations required by GMIN and LMIN for the first four iterations of GRAD2 on model D and starting from SP1. It shows conclusively that the model function is such that it can be adequately represented by the quadratic approximation used in the Powell algorithm, and that Powell should be the algorithm used if line minimization is required. Nevertheless GRAD2 overall is no more satisfying than GRADNT, and because it uses the optimum gradient strategy it can be concluded that no algorithm employing the gradient or steepest descent direction will be found satisfactory for the problem treated here. It is not difficult to see why; the gradient direction for model D at starting point SP1 is oriented at an angle of 89.98 degrees to the actual direction of the solution, and such routines get trapped in an extremely narrow valley, whereas the solution can be reached by proceeding along a gently sloping and nearly straight path. Figure 2.12, in which the solid line gives the variation of ϕ along the steepest descent direction, and the broken one that for the solution direction, is illustrative of this.

The other algorithms DLSQ, MARQDT, SPIRAL and MARQT2, the estimated derivative version of MARQDT, perform sufficiently well for useful comparisons to be made. The convergence criterion in each is identical, namely that convergence is reached when

$$|t_j|/(|b_j| + \tau) \leqslant \varepsilon \quad \text{for all } j = 1,\ldots k \qquad \ldots\ 2.46$$

where the t_j are the components of the particular correction vector, $\varepsilon = 10^{-5}$ is the convergence tolerance and $\tau = 10^{-15}$ is to guard against underflow in case some $b_j = 0$, (see flowchart in Figure 2.9 for alternative ways that SPIRAL can converge). However to enable such comparisons to be made a suitable number must be decided for the iteration breakpoint i_s at which to 'switch off' the damping (set $\lambda = 0$) in the routine DLSQ, for otherwise the rate of convergence is rapidly decreased. A test was run using DLSQ with $i_s = 50$ on data generated from model D and starting from SP4, and the behaviour of ϕ with successive iterations is shown in Figure 2.13. After an initial steep decrease in ϕ for about 8 iterations the routine slows to a rate of convergence which is markedly inferior to that of the others (see Figure 2.14).

The routine was also run on the same data and starting point as above with various values of iteration breakpoint, and the effect of these is seen in Table 2.5. In all the runs the convergence was complete and the final parameter estimates were correct to better than 8 significant figures. Neither the total number of iterations required, nor the number required after the breakpoint, varies simply with the breakpoint i_s and this is attributed to the non-monotonic behaviour of DLSQ directly after the breakpoint. For the breakpoints tried below 26, ϕ at some stage increased rather than decreased, usually 2 or 3 iterations after the breakpoint. To discourage this potentially unstable behaviour we require a fairly large number for the breakpoint yet not so large that it prolongs convergence; on the basis of Figure 2.13 and Table 2.5 a breakpoint of 10 was chosen for all subsequent work with DLSQ.

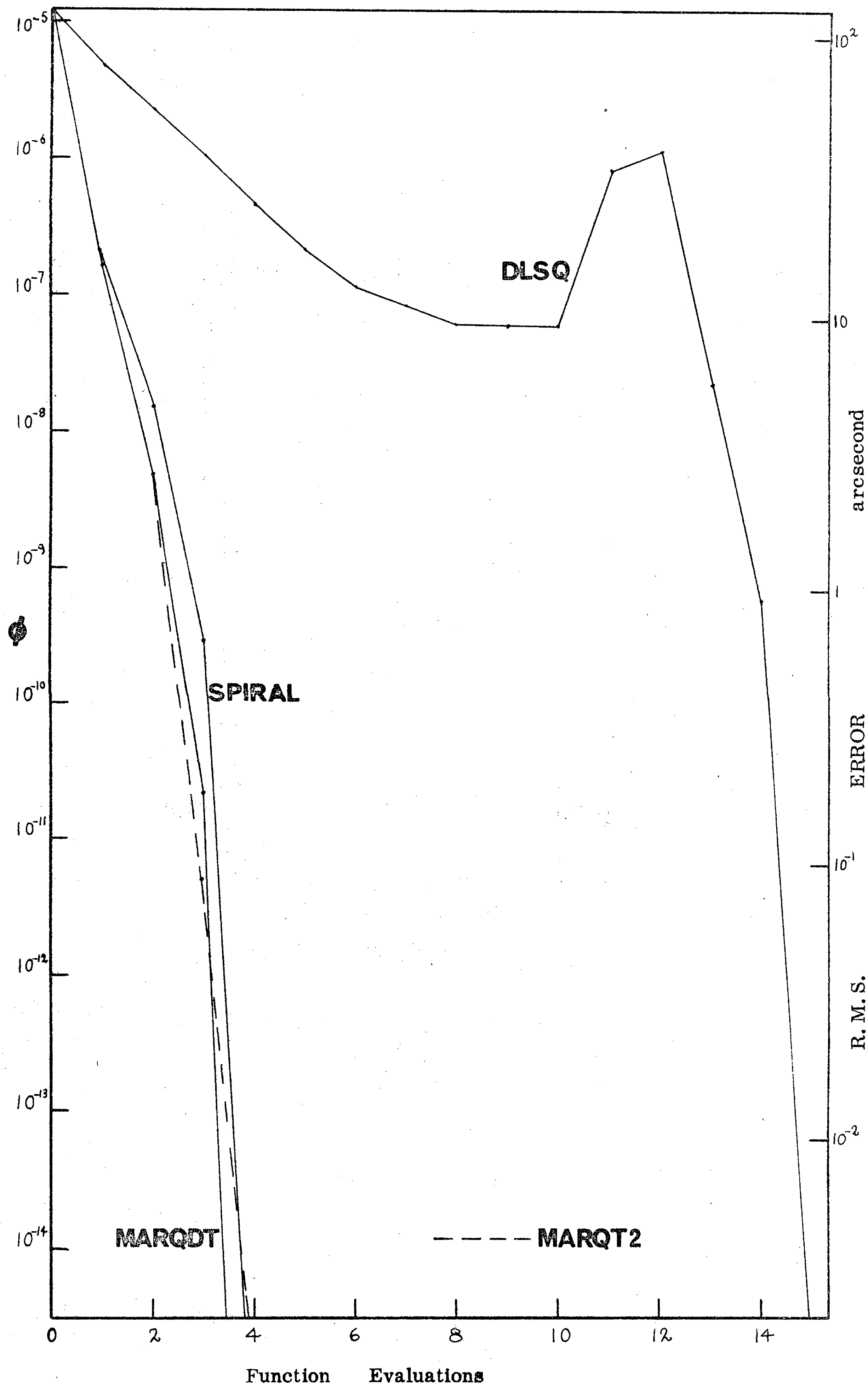
TABLE 2.5

BREAKPOINT i_s	3	5	8	11	14	19	26	50
iterations required for convergence	14	13	15	17	22	24	30	52
iterations required after breakpoint	11	8	7	6	8	5	4	2
iterations at which increased	5,6,8,11	7,8	10,11	13,14	16	21	none	none
final sumsquare ϕ @	2.5^{-27}	1.7^{-33}	2.7^{-32}	4.2^{-34}	1.7^{-34}	1.7^{-34}	2.0^{-34}	2.0^{-26}

@ superscripts are decimal exponents e.g. $2.5^{-27} = 2.5 \times 10^{-27}$.

Literature comparisons of parameter estimation algorithms applied to specific problems often compare computation time, number of iterations or number of function evaluations required to produce a convergence. The first is complicated by computational overheads and coding inefficiencies, and the second by the vague meaning of which loop in an iterative strategy one considers to be 'the iteration'. The number of function evaluations would appear to be the best criterion for comparing algorithms of quite different strategy and is used by Powell (1965), Jones (1970) and others. All the three routines, DLSQ, MARQDT and SPIRAL calculate the derivatives $\partial f_i^{(l)}/\partial b_j$ once per iteration, but vary as to the number of times the function itself $f_i^{(l)}$ is computed. DLSQ calculates it once; MARQDT usually calculates it once or twice (depending on whether λ needs decreasing or not), but if a reduced sum of squares ϕ is difficult to find, a function

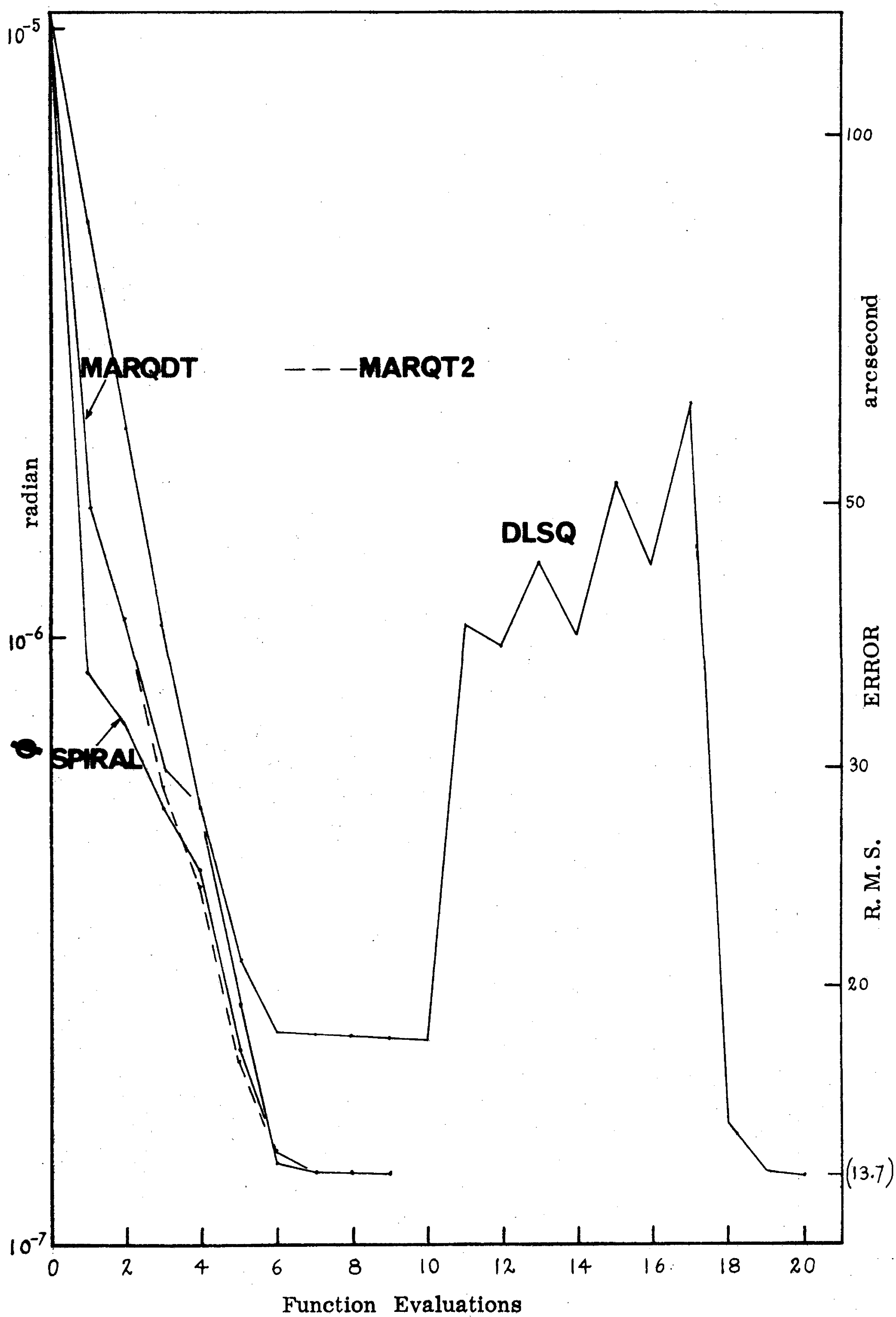
FIG 2.14



10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
Φ
10^{-10}
10^{-11}
10^{-12}
10^{-13}
10^{-14}
10^{2}
10
1
10^{-1}
10^{-2}
R. M. S. ERROR arcsecond
DLSQ
SPIRAL
MARQDT
MARQT2
0
2
4
6
8
10
12
14
Function Evaluations

FIG 2.15

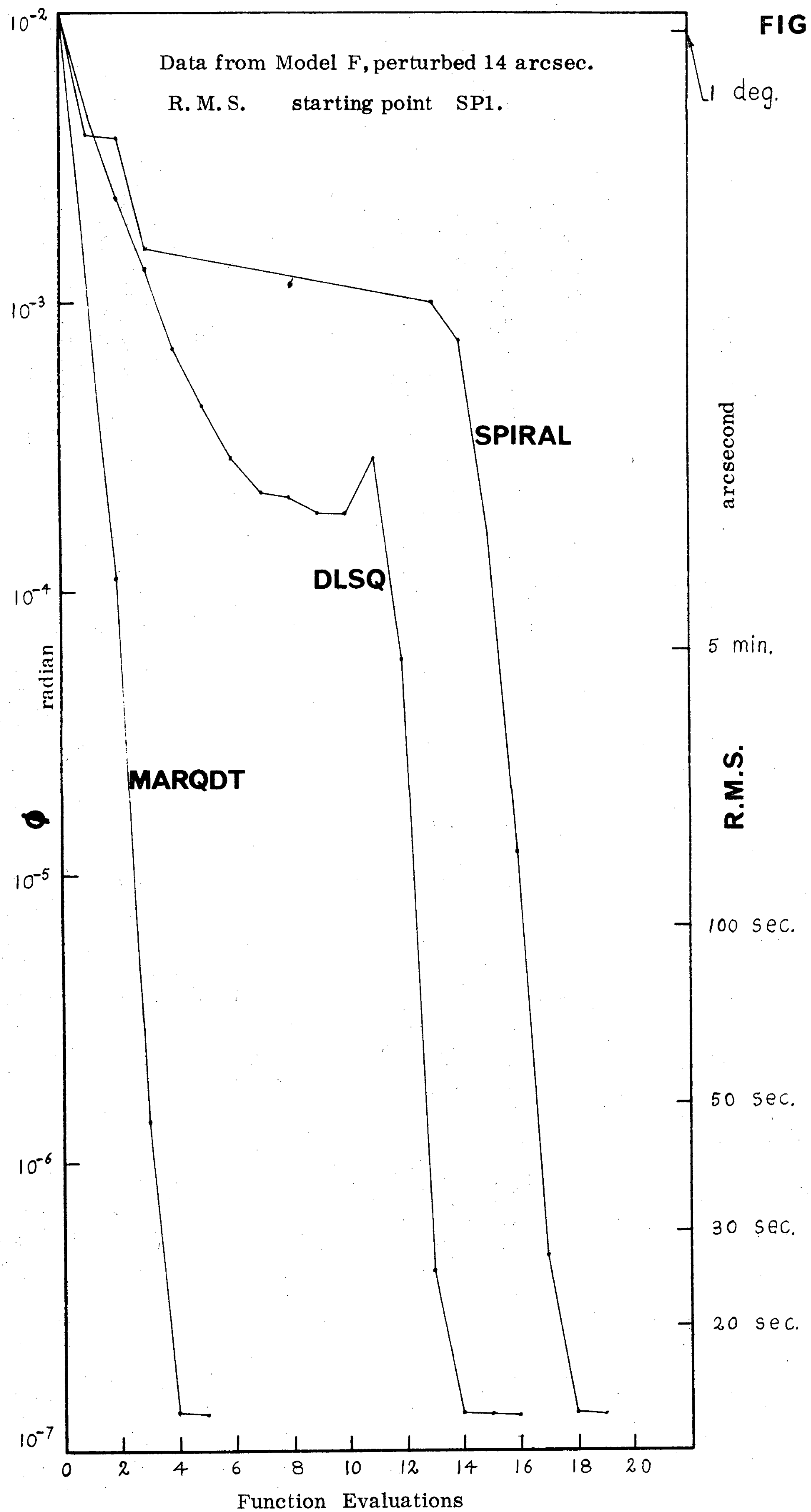
Tests with data perturbed by an additional error of 14 arcsecond r. m. s.



evaluation is required for every new trial value of λ generated (see Figure 2.7). SPIRAL evaluates $f_i^{(1)}$ a completely variable number of times depending on the number of points generated along a spiral and where on the spiral a reduced sum of squares ϕ was encountered.

Routines DLSQ, MARQDT, SPIRAL and MARQT2 were run on data generated from model D starting from each of the six starting points in Table 2.2, and in every case completely converged to the correct parameter values. Table 2.6 gives the number of iterations and function evaluations required for convergence, and only in the case of starting point SP4 did any of the routines experience trouble. MARQDT, SPIRAL and MARQT2 all experience divide errors due to division by zero when scaling the parameters, but if this is supressed, carry on and converge normally. MARQT2 after an initial few satisfactory iterations converges prematurely on an incorrect answer. Figure 2.14 shows the variation in ϕ with number of function evaluations for each of the routines starting from SP1. Paths in ϕ are not shown for the other starting points, but Figure 2.14 is certainly typical of the behaviour of the algorithms. DLSQ was often observed to oscillate before converging but the other two are restricted to a monotonic path by virtue of their internal check that ϕ is reduced after each iteration. A tendency existed for all routines (though to a lesser extent with DLSQ presumably because of its initial slow rate of convergence) to find an answer for parameter b_3 which included an additive constant $2m\pi$ where m is an integer, or to find both b_1 and b_3 negative; this is, of course, still a correct result.

To test the stability of the solutions, tests similar to those above were run but with the data of model D perturbed by an additional 14 arcsecond R.M.S. using the routine DAPERT which adds a normally distributed pseudo-random number (of known standard deviation) to each $y_i^{(1)}$. All four routines converged to a final R.M.S. error of 13.5 arcsecond and parameter estimates which were identical to a precision of 8 significant figures, in all cases. Table 2.7 shows the number of iterations and function evaluations required in each case; again for the case SP4 a divide error had to be suppressed before the figures for MARQDT and SPIRAL could be obtained and MARQT2 converged prematurely. The orientation parameter b_3 differed from the unperturbed estimate by 0.73 degrees and the other four parameters by an average of 3.4 arcsecond. Figure 2.15 shows the performance of the four routines starting from SP1 as a function of the number of function evaluations, and demonstrates the oscillatory behaviour of DLSQ once the damping is removed in contrast to the very uniform behaviour of the other three routines. The data of model D was also perturbed by various other amounts and Table 2.8 gives the number of iterations and



Data from Model F, perturbed 14 arcsec.
R. M. S. starting point SP1.
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
radian
ϕ
1 deg.
5 min.
100 sec.
50 sec.
30 sec.
20 sec.
arcsecond
R.M.S.
SPIRAL
DLSQ
MARQDT
0 2 4 6 8 10 12 14 16 18 20
Function Evaluations

FIG 2.16

function evaluations for the routines working on data with introduced R.M.S. perturbations of 2.8, 7, 14, 28 and 56 arcseconds. The routines were started at SP1 and for a given perturbation all converged to the same final value of R.M.S. error and a consistent parameter estimate.

TABLE 2.6

Unperturbed data from model D

Starting point number		ROUTINE DLSQ	MARQDT	SPIRAL	MARQT2
SP1	fn.	17	5	5	36 (6)
	iter.	17	5	5	6
SP2	fn.	18	14	17	59 (14)
	iter.	18	9	8	9
SP3	fn.	21	15	16	70 (15)
	iter.	21	11	7	11
SP4	fn.	17	10	34	premature
	iter.	17	7	7	convergence
SP5	fn.	17	7	5	42 (7)
	iter.	17	7	5	7
SP6	fn.	19	5	5	30 (5)
	iter.	19	5	5	5

UPPER figure is number of function evaluations required for convergence.
LOWER figure is number of iterations required for convergence.
Figure in parenthesis for MARQT2 gives the number of function evaluations if those used solely for estimating the derivatives are excluded.

Finally the four routines were tried on data generated from all six of the sets of model parameters given in Table 2.1. Starting point SP4 because of its symmetry and uniqueness is the obvious choice for such a comparison, but because of the numerical problems it causes all of the routines except DLSQ, it must be avoided and SP1 was used as a suitable alternative. Table 2.9 gives the performance of the routines on the various models, and Table 2.10 shows similar runs but with the data from each model perturbed by an additional 14 arcsecond. Model F (which admittedly represents a more erratic mounting than would normally be encountered in practice) caused considerable problems to the estimated derivative routine MARQT2, which became 'stuck' after only 2 or 3 iterations and then converged prematurely. Figure 2.16 shows the progress of DLSQ, MARQDT and SPIRAL on the perturbed data of model F starting from SP1. This is one

TABLE 2.7

Data perturbed by additional R.M.S. of 14 arcsecond

Starting point number		ROUTINE DLSQ	MARQDT	SPIRAL	MARQT2
SP1	fn.	21	9	18	54 (9)@
	iter.	21	9	9	9
SP2	fn.	18	14	17	64 (14)
	iter.	18	10	8	10
SP3	fn.	19	12	16	62 (12)
	iter.	19	10	7	10
SP4	fn.	19	10	34	premature
	iter.	19	7	7	convergence
SP5	fn.	20	24	36	87 (22)
	iter.	20	14	9	13
SP6	fn.	18	5	5	30 (5)
	iter.	18	5	5	5

TABLE 2.8

Data from model D various perturbations, starting point SP1

Introduced R.M.S. error arcsecond		ROUTINE DLSQ	MARQDT	SPIRAL	MARQT2	Final R.M.S. error arcsec
2.81	fn.	17	19	7	66 (16)@	2.74
	iter.	17	11	7	10	
17.04	fn.	17	16	7	54 (9)	6.86
	iter.	17	10	7	9	
14.07	fn.	21	9	18	54 (9)	13.71
	iter.	21	9	9	9	
28.14	fn.	16	8	34	48 (8)	27.42
	iter.	16	8	7	8	
56.28	fn.	15	19	8	56 (11)	54.85
	iter.	15	11	8	9	

@ Figure in parenthesis for MARQT2 gives the number of function evaluations if those used solely for estimating the derivatives are excluded.

TABLE 2.9

Solution to various models starting from SP1

MODEL		ROUTINE			
		DLSQ	MARQDT	SPIRAL	MARQT2
A	fn.	15	6	6	36 (6)[@]
	iter.	15	6	6	6
B	fn.	17	7	8	42 (7)
	iter.	17	7	8	7
C	fn.	14	7	6	42 (7)
	iter.	14	7	6	7
D	fn.	17	5	5	36 (6)
	iter.	17	5	5	6
E	fn.	17	18	66	122 (32)
	iter.	17	11	10	18
F	fn.	16	7	28	premature
	iter.	16	7	10	convergence

TABLE 2.10

Various models perturbed by additional 14 arcsec. R.M.S., SP1

MODEL		ROUTINE			
		DLSQ	MARQDT	SPIRAL	MARQT2
A	fn.	10	15	17	60 (15)[@]
	iter.	10	9	8	9
B	fn.	15	6	99	42 (7)
	iter.	15	6	11	7
C	fn.	12	5	18	30 (5)
	iter.	12	5	9	5
D	fn.	21	9	18	54 (9)
	iter.	21	9	9	9
E	fn.	17	6	7	42 (7)
	iter.	17	6	7	7
F	fn.	16	5	19	premature
	iter.	16	5	10	convergence

@ Figure in parenthesis for MARQT2 gives the number of function evaluations if those used solely for estimating the derivatives are excluded.

of a number of cases in which SPIRAL requires an abnormally large number of function evaluations, and the three routines show quite different overall rates of convergence.

(2.5) CONCLUDING DISCUSSION

The computing runs discussed above show that it is quite feasible to estimate the parameters in models of pointing errors typically exhibited by telescopes. The steepest descent routines proved quite unsatisfactory as practical methods and are incapable of coping with the topography of the ϕ surface. It was initially thought that the solution could be started with a descent method and one of the Gaussian type of algorithms used when nearing the solution, but even the initial progress of GRADNT and GRAD2 is unimpressive, and in any case the Gaussian routines experience little trouble in attaining the correct result wherever they are started. The only conclusion worthy of note which comes from the study of the two descent routines is the usefulness and efficiency of LMIN as a line minimization scheme.

The other four routines should prove quite satisfactory in practice. On the unperturbed data generated from the various sets of model parameters they all converge to a final sumsquare ϕ of between 10^{-25} and 10^{-33}, and for the perturbed data they converge to an identical parameter estimate and ϕ; this is to be expected since the process of leastsquares minimization ensures a unique solution, which for the perturbed case, lies well away from the region of cumulative machine error. The solutions are quite stable as is indicated by the fact that the fractional variation between the parameter estimates for the perturbed and unperturbed cases is approximately the same as that fraction of the total R.M.S. error in the data which is made up by the perturbation. It should be noted that the fits generated to the perturbed data are apparently better than would be expected from the error introduced into the data by some 2 to 3%. This is because some of the error in the resulting data has been fitted by the estimation process, and if standard deviations had been tabulated (with due regard to the appropriate degrees of freedom) instead of R.M.S. error, only very small differences would have been observed.

The Levenberg algorithm DLSQ suffers from a rate of convergence which is markedly inferior to the other (Gaussian) routines. Marquardt (1963) predicts this and likens the method to a steepest descent process; yet this is not entirely true since it is capable of steady, reliable progress towards the solution as is seen in Figure 2.13 where a residual error level of approximately an arcsecond is attained within 40 iterations. It is the only one of the algorithms which does not give trouble when certain awkward values of the parameters (e.g. zero) are encountered and could prove very useful in practice on problems which are less stable or more prone to numerical troubles. The device adopted here of suddenly switching off the damping often causes severe oscillations (see Figure 2.15) and is not recommended. Other schemes for progressively reducing λ could be devised but in view of the effectiveness of the Marquardt algorithm such effort is probably not warranted.

In about 60 percent of the computer runs described above the routine MARQDT proved the most efficient method; this was particularly noticeable when the starting point was distant from the eventual solution. When the progress of the routines on the ϕ surface is such that the correction vectors lie near the Taylor direction, MARQDT and SPIRAL proceed along quite similar paths, and in about 20 percent of the runs SPIRAL is in fact the superior routine. Occasionally, using SPIRAL, a large number of function evaluations are required within an iteration to search along the spiral paths for a reduced sumsquare ϕ , and in such cases the routine compares poorly with MARQDT, and even DLSQ. In Jones (1970), SPIRAL is shown to be substantially superior to MARQDT but it is not uncommon for comparisons of this nature to be both problem and data dependent.

MARQT2, the estimated derivative version of MARQDT, proved surprisingly effective; in 70 percent of the tests it follows the path of MARQDT very closely. On a few spurious occasions it is actually superior to MARQDT, and since most of these are for the perturbed data cases (which are the more typical of data to be encountered in practice) it is clear that estimated derivatives may suffice for many problems in practice where analytic differentiation of the model function is considered either excessively onerous, or an impedance to experimentation with the model. MARQT2 was, however, somewhat more susceptible to numerical problems and premature convergence, particularly when zero values of the parameters were encountered.

It is clear from the foregoing that the routine to be recommended for parameter estimation of pointing error models is Marquardt's 1963 algorithm MARQDT, preferably with the modification suggested by Jones (1970) included. The geometric nature of the process being modelled, and the practical limitations on both the domain and range of the model function ensure that such functions are well behaved even though large numbers of parameters may be involved in practice. The greatest difficulty in model estimation lies in formulating a model in which there are no redundant parameters. This is essential if we wish to use the parameter estimates as a basis for conclusions concerning the physical causes of error[@], and even when we do not, redundant parameters can cause unstable solutions and numerical trouble in the routines used. Chapter 4 demonstrates problems of this nature encountered in devising a model for a practical telescope.

@ see Chapter 1 for a brief discussion of 'mechanism determination v.s. response surface optimization'.

CHAPTER THREE
ERROR SURFACE FITTING AND INTERPOLATION

(3.1) PRELIMINARY

The main causes of telescope pointing errors are fairly evident: structural flexure, gear errors, encoder nonlinearities and mounting geometry errors. Yet the difficulty of deriving the model function in Chapter 2 for a given case, and the fact that there may exist numerous unexpected causes whose effects dominate the expected ones, suggest a consideration of the surface fitting process. That is, we attempt to approximate the error surfaces $\Delta\delta$ and ΔH (which are functions of the two variables δ, H) by some approximating functions $\psi^{(1)}$ and $\psi^{(2)}$ respectively. We need to decide the form of these approximating functions, and the criterion for a satisfactory approximation. For the reasons cited in Chapter 1, we use here the criterion of minimization of the leastsquares or L_2 norm; in fact we seek the minimum of ϕ defined in equation 2.7 since this minimizes the R.M.S. resultant error on the sky.

The approximating functions can be polynomials, or periodic functions (e.g. Fourier series), or rational functions etc; in all cases a number of coefficients or parameters must be determined according to the criterion above. There is a formal similarity to parameter estimation in this regard, but by making the approximating functions linear in the parameters (coefficients) we permit considerable simplification of the process by which these are determined, and by further restricting ourselves to polynomials we can avoid numerical instabilities in the computations. We explain this by briefly alluding to the one dimension case, namely curve fitting.

(3.2) CURVE FITTING

Let $x_1, \ldots x_n$ be n observations of the independent experimental variable, and let $y_1, \ldots y_n$ be the corresponding values of the dependent variable. We wish to find the coefficients c_j $j = 0, \ldots k$ in the approximating function ψ given by

$$\psi_i = \psi(x_i) = \sum_{j=0}^{k} c_j P_j(x_i), \qquad \ldots 3.1$$

where the P_j are polynomials in x[@], such that the expression

$$\phi = \sum_{i=1}^{n} \left[y_i - \psi_i\right]^2 w_i^2 = \sum_{i=1}^{n} \left[y_i - \sum_{j=0}^{k} c_j P_j(x_i)\right]^2 w_i^2 \qquad \ldots 3.2$$

@ For the c_j to be unique, the P_j must be linearly independent i.e. P_j must not be expressible as a linear combination of all the other P_q for $q \neq j$. In practice this may be arranged by having P_j of degree equal to j.

is minimized (where w_i is an arbitrary weighting function). Setting the derivatives of ϕ with respect to the coefficients c_j to zero, gives the matrix equation

$$A\,\underline{c} = \underline{b} \qquad \ldots\ 3.3$$

where $\underline{c} = (c_j)$ is the (k+1)-vector of coefficients, and matrix $A = (A_{rj})$, where for $r = 0,\ldots k$ and $j = 0,\ldots k$

$$A_{rj} = \sum_{i=1}^{n} P_r(x_i)\, P_j(x_i)\, w_i^2 \qquad \ldots\ 3.4$$

and

$$b_j = \sum_{i=1}^{n} y_i\, P_j(x_i)\, w_i^2 . \qquad \ldots\ 3.5$$

(3.3) ORTHOGONAL POLYNOMIALS

The matrix A in equation 3.3 is often ill-conditioned for arbitrary choice of the polynomials P_j (e.g. power series), and we utilize polynomials which are orthogonal, that is

$$\sum_{i=1}^{n} P_r(x_i)\, P_j(x_i)\, w_i^2 = 0 \quad \text{for all } r \neq j. \qquad \ldots\ 3.6$$

In this case A becomes a diagonal matrix and the coefficients c_j are given by

$$c_j = \left[\sum_{i=1}^{n} y_i\, P_j(x_i)\, w_i^2\right] \Big/ \left[\sum_{i=1}^{n} P_j^2(x_i)\, w_i^2\right]. \qquad \ldots\ 3.7$$

It is clear from equation 3.6 that the polynomials used depend on the range and distribution of the discrete data and on the weighting function used; thus we must generate orthogonal polynomials specifically for the data taken in the experiment.

A method of generating suitable polynomials is the Gram-Schmidt orthogonalization procedure as used by Cadwell and Williams (1961), and others. P_0 is taken identically as unity, and subsequent P_j given by

$$P_j = x^j + \text{a linear combination of } P_0,\ldots P_{j-1}. \qquad \ldots\ 3.8^{@}$$

The coefficients in the linear combination in equation 3.8 can be determined so that the resultant P_j satisfy equation 3.6. For an approximating function of degree k, some k(k+1)/2 such coefficients must be found, and a somewhat more convenient and widely used process is that described (though not originated) by Forsythe (1957). As above, Forsythe's method sets $P_0 = 1$, but the recurrence relation for P_j involves only the previous

---

@ This is the most usual form; in general x^j could be replaced by the j^{th} term of a sequence of basis functions which span the space formed from the product of k real lines R^k.

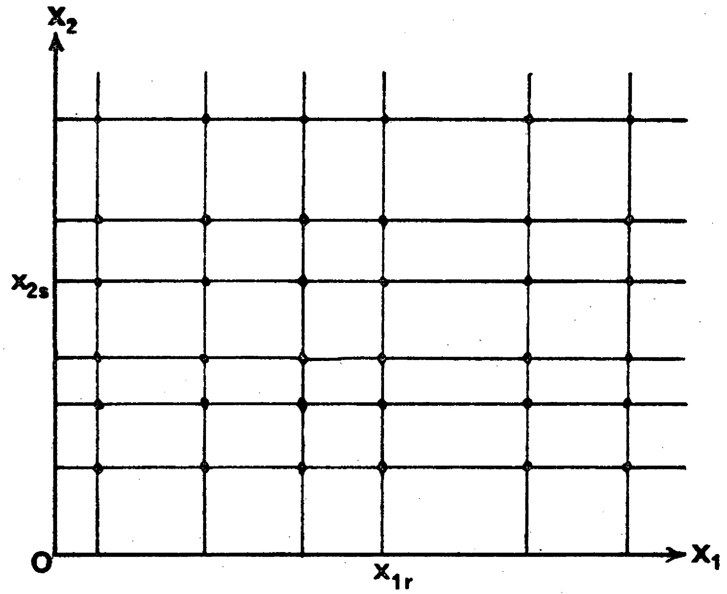


FIG 3.1

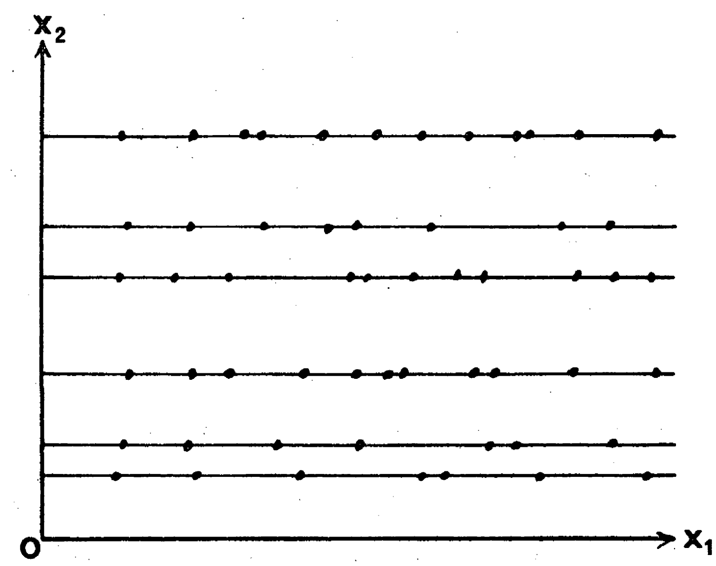


FIG 3.2

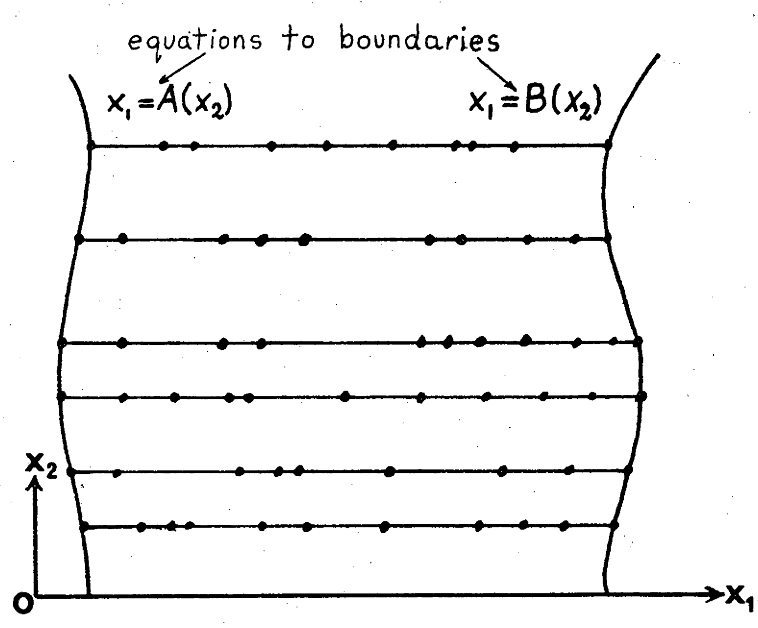


FIG 3.3

two terms,

$$P_j = x.P_{j-1} + \text{linear combination of } P_{j-1} \text{ and } P_{j-2} \qquad \ldots 3.9$$

Thus only 2k-1 coefficients in equation 3.9 need be computed to obtain the required orthogonal polynomials. Forsythe's method has had wide application in discrete data curve fitting problems e.g. Clenshaw (1960), Berztiss(1964), and Clenshaw and Hayes (1965), and is the method generalized to multiple dimensions in Weisfeld (1959).

(3.4) SURFACE FITTING WITH RESTRICTED DATA DISTRIBUTIONS

Surface fitting, that is fitting a scalar function of two independent variables $y_i = y(x_{1i}, x_{2i})$, can be regarded as an extension of curve fitting if certain restrictions are placed on the manner in which the data is distributed. If the data lies on the intersections of a rectangular grid (see Figure 3.1), then one can fit a series of orthogonal polynomials in x_1 to the data points along each line $x_2 = x_{2s}$, and then fit the coefficients obtained by a series of orthogonal polynomials in x_2. Hayes (in Hayes 1970) shows that this procedure is identical to a least-squares surface fit using product polynomials of the form $P_p(x_1).P_q(x_2)$.

If the data lies on lines parallel to the x_1 axis (say), but is distributed randomly along those lines (see Figure 3.2), we need to abandon our orthogonal polynomials in favour of polynomials which are the same for each line, or modify the Forsythe method by actually expressing each of the generated polynomials in terms of their Chebyshev expansion. The latter technique (Clenshaw 1960) permits surface fitting the data distribution of Figure 3.2 by repeated application of curve fitting, and indeed can be extended even further to cover the case where the boundaries of the data domain are not straight but are simple analytic functions as in Figure 3.3. Fitting the data of Figures 3.2 and 3.3 in this manner does not produce the true leastsquares surface fit as in the case of a complete grid, and the fit is not necessarily unique, but the results are deemed to be the same for most practical purposes (Clenshaw and Hayes 1965).

Unfortunately, the methods above are not applicable to our work with pointing errors which involve randomly distributed data. It could be argued that for the case of an equatorial mounting, a given celestial object could be used at different times and therefore different hourangles, and thus the data would lie on lines of constant declination. However this renders the collection of pointing error data difficult and time consuming, necessitates the interchanging of dependent and independent variables, and is out of the question if we envisage some form of automatic software package which will measure and process pointing errors whilst the telescope system is in normal astronomical use. So we are left with the problem of doing a leastsquares surface fit using some form of polynomials which are orthogonal over our randomly distributed two-dimensional data. Before describing this process we should distinguish the case of surface interpolation.

(3.5) SURFACE INTERPOLATION

Should the degree k of the approximating function ψ and the number of data points n be such that $k+1 \geqslant n$, we speak of surface interpolation, since, for example, if the number of data points is actually equal to the number of coefficients (or parameters) to be found, the residuals in equation 3.2 can all be forced to zero, and the surface made to pass exactly through each data point. If recourse must be made to complicated functions for ψ, there appears little point in interpolating as distinct from fitting, and in most of the literature on surface fitting, fairly simple functions, designed primarily with data smoothing in mind, are employed. Theilheimer and Starkweather (1961), and Birkhoff and de Boor (1965) work with cubic splines on the rectangular grid of Figure 3.1, while Birkhoff and Garabedian (1960) use cubic splines on data distributions ranging from rectangles to curvilinear triangles[@]. Ferguson (1964) and Coons (1967) express the surface in parametric form and deal with any data distribution which is topologically equivalent to a rectangular grid.

None of the above ideas suit our data distribution since, even for the last two mentioned, it is theoretically impossible to get an automatic routine to draw a 'twisted' grid through the data obtained. Thacher and Milne (1960) treat random multivariate data distributions and give a general determinant formula for the coefficients of the interpolating functions, but in many cases the set of interpolating functions chosen leads to ill-conditioning of the problem, and implicit in the scheme is repeated transformations of the interpolating functions until a satisfactory set is found. Since this process is far more difficult than calculation of two-dimensional orthogonal polynomials, and since it is difficult to arrange that the number of data and order of fit are equal, we restrict our interest to surface fitting using such polynomials; indeed, should one insist on interpolation rather than fitting one need only arrange for (k+1) and n to be equal in the routine to be described.

@ A cubic spline is a polynomial which runs exactly through a number of points (knots) on the function being interpolated; its first and second derivatives are continuous at the knots, and between adjacent knots it is of degree three.

(3.6) THE GENERATION OF TWO-DIMENSIONAL ORTHOGONAL POLYNOMIALS

Weisfeld (1959) generalizes the Forsythe orthogonalization procedure to several variables taking functions of the form $x_1^{s_1}.\, x_2^{s_2}.\, x_3^{s_3} \ldots$ and showing that orthogonal linear combinations of them can be constructed. Using Weisfeld's generalized notation for the specific case of two dimensions would be both unnecessary and unwieldy, and since the published usage of the method, e.g. Bain (1961) or (slightly altered) Cadwell and Williams (1961), does not include a complete description, the method of computing two-dimensional orthogonal polynomials for a given data set and weighting function is given here.

If we denote the n data points by $\underline{x}_i = (x_{1i}, x_{2i})$, $i = 1,\ldots n$, we require k+1 polynomials $P_j(\underline{x}_i)$ of maximum degree k which are orthogonal in the sense that

$$\sum_{i=1}^{n} P_j(\underline{x}_i)\, P_{j'}(\underline{x}_i)\, w_i^2 = 0 \quad \text{for all} \quad j \neq j', \qquad \ldots\ 3.10$$

where $j = 0,\ldots k$; $j' = 0,\ldots k$. We generate the P_j in groups from a set of basis functions comprising the monomials $x_1^s . x_2^t$. These monomials have an inherent two-dimensional ordering (by s and t), and if P_j introduces the term $x_1^s . x_2^t$ for the first time, and $P_{j'}$ the term $x_1^{s'} . x_2^{t'}$, then a one-dimensional ordering is induced in the P_j by defining

$$\left.\begin{array}{llll} j < j' & \text{if} & s+t < s'+t' & \\ & \text{or if} & s+t = s'+t' & \text{and} \quad t < t' . \end{array}\right\} \ldots\ 3.11$$

Thus the monomials are introduced into the P_j in the order

$$\begin{tikzcd}
1 \arrow[d] & & & \\
x_1 \arrow[r] & x_2 \arrow[dl] & & \\
x_1^2 \arrow[r] & x_1x_2 \arrow[r] & x_2^2 \arrow[dll] & \\
x_1^3 \arrow[r] & x_1^2x_2 \arrow[r] & x_1x_2^2 \arrow[r] & x_2^3
\end{tikzcd}$$

etc.

As in the one-dimensional case P_0 is set to one identically and the subsequent polynomials computed as follows:

$$
\begin{array}{lll}
0^{th} \text{ group} & P_0 = 1 & \\
1^{st} \text{ group} & P_1 = x_1\, P_0 - \alpha_{10}.P_0\,, & \\
 & P_2 = x_2\, P_0 - \alpha_{21}.P_1 - \alpha_{20}.P_0\,, & \qquad \ldots 3.12 \\
2^{nd} \text{ group} & P_3 = x_1\, P_1 - \alpha_{32}.P_2 - \alpha_{31}.P_1 - \alpha_{30}.P_0\,, & \\
 & P_4 = x_2\, P_1 - \alpha_{43}.P_3 - \alpha_{42}.P_2 - \alpha_{41}.P_1 - \alpha_{40}.P_0\,, & \\
 & P_5 = x_2\, P_2 - \alpha_{54}.P_4 - \alpha_{53}.P_3 - \alpha_{52}.P_2 - \alpha_{51}.P_1 - \alpha_{50}.P_0\,, & \\
 & \text{etc.} &
\end{array}
$$

Specifically, polynomial P_j in the g^{th} group is generated by a term which introduces $x_1^s . x_2^t$ and a linear combination of all the polynomials preceding P_j in that group and in the previous two groups. Denoting by $g(j)$ the group in which the polynomial P_j is found, we have

$$P_j = x_l\, P_q - \sum_r \alpha_{jr}\, P_r \qquad \ldots 3.13$$

where the summation is over all $r < j$ for which $g(j) - g(r) \leqslant 2$. The group number, polynomial number and indices of the introduced term are related by

$$g = s + t \qquad \ldots 3.14$$

and $$j = g(g+1)/2 + t \qquad \ldots 3.15$$

Table 3.1 shows the scheme for the first 21 polynomials and gives the appropriate values of l and q for equation 3.13.

An algorithm for finding l and q given only the j value is shown in Figure 3.4 and, this having been done, the α_{jr} in equation 3.13 are computed from

$$\alpha_{jr} = \frac{\sum_i x_l . P_q . P_r\, w_i^2}{\sum_i P_r^2\, w_i^2} = \frac{\sum_{i=1}^{n} x_{li}\, P_q(\underline{x}_i)\, P_r(\underline{x}_i)\, w_i^2}{\sum_{i=1}^{n} P_r^2(\underline{x}_i)\, w_i^2} \qquad \ldots 3.16$$

where the sums are over all the data points, and x_{li} is the value of the co-ordinate x_l at the i^{th} point. Equations 3.13 and 3.16 are analogous to the three term recurrence relation of Forsythe in the single variable case. This two-dimensional orthogonal polynomial generating scheme is used in the error surface fitting routine described below.

Algorithm for finding l and q given j.

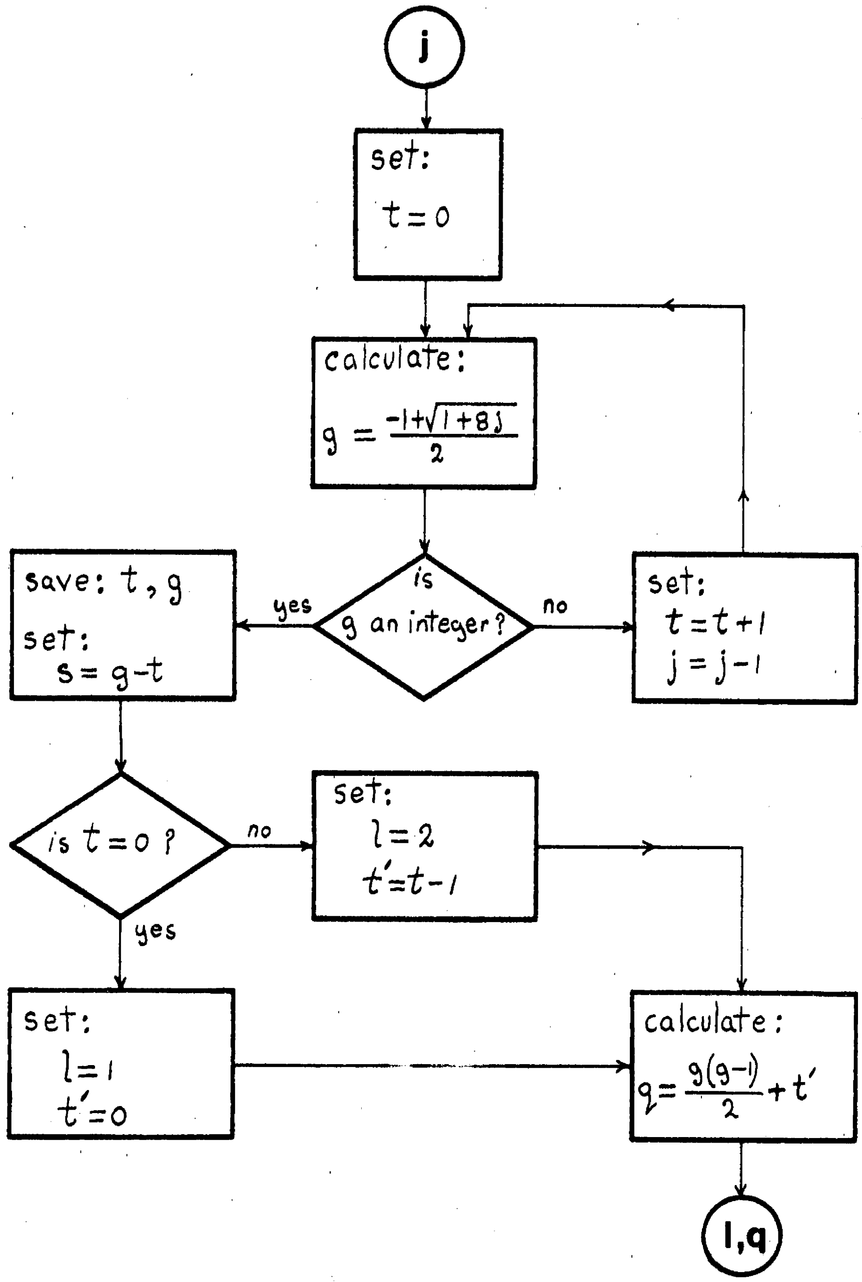


TABLE 3.1

Two-dimensional orthogonal polynomial computation scheme

resultant polynomial P	s	t	term introduced $x_1^s\ x_2^t$	l	q	group g = s+t
P_0	0	0	1	-	-	0
P_1	1	0	x_1	1	0	1
P_2	0	1	x_2	2	0	
P_3	2	0	x_1^2	1	1	2
P_4	1	1	$x_1\ x_2$	2	1	
P_5	0	2	x_2^2	2	2	
P_6	3	0	x_1^3	1	3	3
P_7	2	1	$x_1^2\ x_2$	2	3	
P_8	1	2	$x_1\ x_2^2$	2	4	
P_9	0	3	x_2^3	2	5	
P_{10}	4	0	x_1^4	1	6	4
P_{11}	3	1	$x_1^3\ x_2$	2	6	
P_{12}	2	2	$x_1^2\ x_2^2$	2	7	
P_{13}	1	3	$x_1\ x_2^3$	2	8	
P_{14}	0	4	x_2^4	2	9	
P_{15}	5	0	x_1^5	1	10	5
P_{16}	4	1	$x_1^4\ x_2$	2	10	
P_{17}	3	2	$x_1^3\ x_2^2$	2	11	
P_{18}	2	3	$x_1^2\ x_2^3$	2	12	
P_{19}	1	4	$x_1\ x_2^4$	2	13	
P_{20}	0	5	x_2^5	2	14	

(3.7) THE ERROR SURFACE FITTING ROUTINE

For fitting telescope error surfaces $\Delta\delta$, ΔH we make identifications similar to those labelled 2.8 in Chapter 2 as follows:

$$\left.\begin{array}{ll}
\left.\begin{array}{l} x_{1i} = \delta \\ x_{2i} = H \end{array}\right\} & \text{Telescope transducer readings for } i^{th} \text{ observation.} \\
\left.\begin{array}{l} y_i^{(1)} = \delta_o \\ y_i^{(2)} = H_o \end{array}\right\} & \text{Topocentric position of celestial object for } i^{th} \text{ observation.} \\
\left.\begin{array}{l} w_i^{(1)} \\ w_i^{(2)} \end{array}\right\} & \text{Weighting functions, see below.} \\
\psi_i^{(1)} = \text{fitting function for surface } \Delta\delta. & \\
\psi_i^{(2)} = \text{fitting function for surface } \Delta H. &
\end{array}\right\} \quad \ldots\ 3.17$$

Note that as in equation 3.1 the fitting functions are of the form

$$\psi_i^{(l)} = \sum_{j=0}^{k} c_j^{(l)} P_j^{(l)}(\underline{x}_i)\ , \qquad l = 1,\ 2; \qquad \ldots\ 3.18$$

and, if the weights for the two surfaces differ $(w_i^{(1)} \neq w_i^{(2)})$, not only will different sets of coefficients $c_j^{(1)}$, $c_j^{(2)}$ be obtained, but two distinct sets of (k+1) polynomials $P_j^{(1)}$, $P_j^{(2)}$ will be required. Fortunately this fits in well with our adoption of the resultant on-sky error for ϕ, our function to be minimized; we simply take the weights $w_i^{(1)} = 1$ and $w_i^{(2)} = \cos(x_{1i})$ in equation 3.19 below:

$$\phi = \sum_{l=1}^{2} \sum_{i=1}^{n} \left[y_i^{(l)} - \sum_{j=0}^{k} c_j^{(l)} P_j^{(l)} \right]^2 . \left[w_i^{(l)} \right]^2 . \qquad \ldots\ 3.19$$

Setting the derivatives of ϕ with respect to the 2(k+1) coefficients $c_j^{(l)}$ to zero produces expressions for the coefficients which are identical to those produced by minimizing the leastsquares fitting errors of the two surfaces $\Delta\delta$, ΔH separately,@ and are given by

$$c_j^{(l)} = \frac{\sum_{i=1}^{n} y_i^{(l)} P_j^{(l)} \left[w_i^{(l)} \right]^2}{\sum_{i=1}^{n} \left[P_j^{(l)} w_i^{(l)} \right]^2} \qquad \ldots\ 3.20$$

where $l = 1,\ 2$ and $j = 0, \ldots k$.

@ This is not true for the case of parameter estimation where the unique solution does not simultaneously minimize the errors in the two co-ordinates.

The routine computes fits for the various orders sequentially since, because the $P_j^{(l)}$ are orthogonal, the lower order coefficients are unaffected by later inclusion of higher order ones in the fitting function, and the j^{th} order sumsquare ϕ is given by

$$\phi_{(j)} = \phi_{(j-1)} - c_j^{(1)\,2} \cdot \sum_{i=1}^{n} (P_j^{(1)} w_i^{(1)})^2 - c_j^{(2)\,2} \cdot \sum_{i=1}^{n} (P_j^{(2)} w_i^{(2)})^2 . \quad \ldots\ 3.21$$

Within a given order of fit the polynomials $P_j^{(l)}$ are stored by the 2(j+1) n-vectors of their values at each data point. The storage requirements could be reduced considerably by storing the Forsythe coefficients α (equation 3.16) or, as suggested in Cadwell and Williams (1961), by storing the partial sums

$$\left.\begin{array}{l} \sum_{i=1}^{n} (x_{mi})^j \\ \sum_{i=1}^{n} y_i^{(l)} (x_{mi})^j \end{array}\right\} \quad j = 0,\ldots k; \quad m = 1,\ 2; \quad l = 1,\ 2. \qquad \ldots\ 3.22$$

Unfortunately both of these techniques can seriously increase the computing overheads and result in loss of accuracy, and so they are not used here; neither can the technique of Clenshaw (1960) be used since Chebyshev expansions only exist for functions of a single variable.

Loss of accuracy also results if the independent variable $\underline{x}$ is badly scaled, and the routine normalizes the components of $\underline{x}$ to the interval (-1, 1) by the relation

$$x'_{mi} = a_m x_{mi} + b_m \qquad \ldots\ 3.23$$

where m = 1, 2; $-1 \leqslant x'_{mi} \leqslant 1$, and a_m, b_m are suitable constants. If we also transform the $\underline{y}$ vectors and weights by

$$\left.\begin{array}{l} y_i^{(m)'} = a_m (y_i^{(m)} - x_{mi}) \\ w_i^{(m)'} = w_i^{(m)} / a_m \end{array}\right\} \ldots\ 3.24$$

where a_m is the same constant as in equation 3.23, we can leave the interpretation of ϕ (equation 3.19) unchanged, but now we are fitting the errors $\Delta\delta$, ΔH rather than the co-ordinates δ, H. Such a system offers numerical advantages especially when we use the resultant fit for the purpose of data interpolation.

The choice of transformation for the data $\underline{x}$, $\underline{y}$ and weight $\underline{w}$ given above is labelled 'method one' in the computer implementation which follows, and if our final criterion of a good fit is to be the practical one of an actual on-sky measurable angle, (as in equation 3.19), method one should be optimal. As a simple test of this, two other methods are considered. In 'method two' the same transformation is used for $\underline{x}$ and $\underline{y}$ but the weights $w_i^{(1)}$ and $w_i^{(2)}$ are all set to unity. In 'method three' the components of $\underline{x}$ are first normalized to the interval $(0, \pi)$ and then the cosine taken so that

$$x'_{mi} = \cos(a_m x_{mi} + b_m) \qquad \ldots 3.25$$

where $0 \leqslant (a_m x_{mi} + b_m) \leqslant \pi$, and the $\underline{y}$ vector given by

$$y_i^{(m)'} = y_i^{(m)} - x_{mi} \qquad \ldots 3.26$$

The reason for this strategy is that now y may be regarded as an even (or symmetric) function on the domain $(-\pi, \pi)$, and thus expressible as a half Fourier series

$$y = \sum_{j=0}^{\infty} a_j \cos(jx) = \sum_{j=0}^{\infty} a'_j (\cos x)^j \qquad \ldots 3.27$$

where x, y denote either of the $\underline{x}$ or $\underline{y}$ components respectively[@]. The fit is now similar to a Fourier analysis scheme in which various Fourier terms have been grouped into orthogonal terms. As for method two, unity weights are used.

A suitable method of determining at which order the fit is adequate is needed, and although we can, at each order, examine the root-mean-square (R.M.S.) on-sky error, we do not necessarily know the extent of the random error in the data which constitutes the practical limit. The variance-ratio F given by

$$F = (\phi_{(j-1)} - \phi_{(j)}) / \sigma_j^2 \qquad \ldots 3.28$$

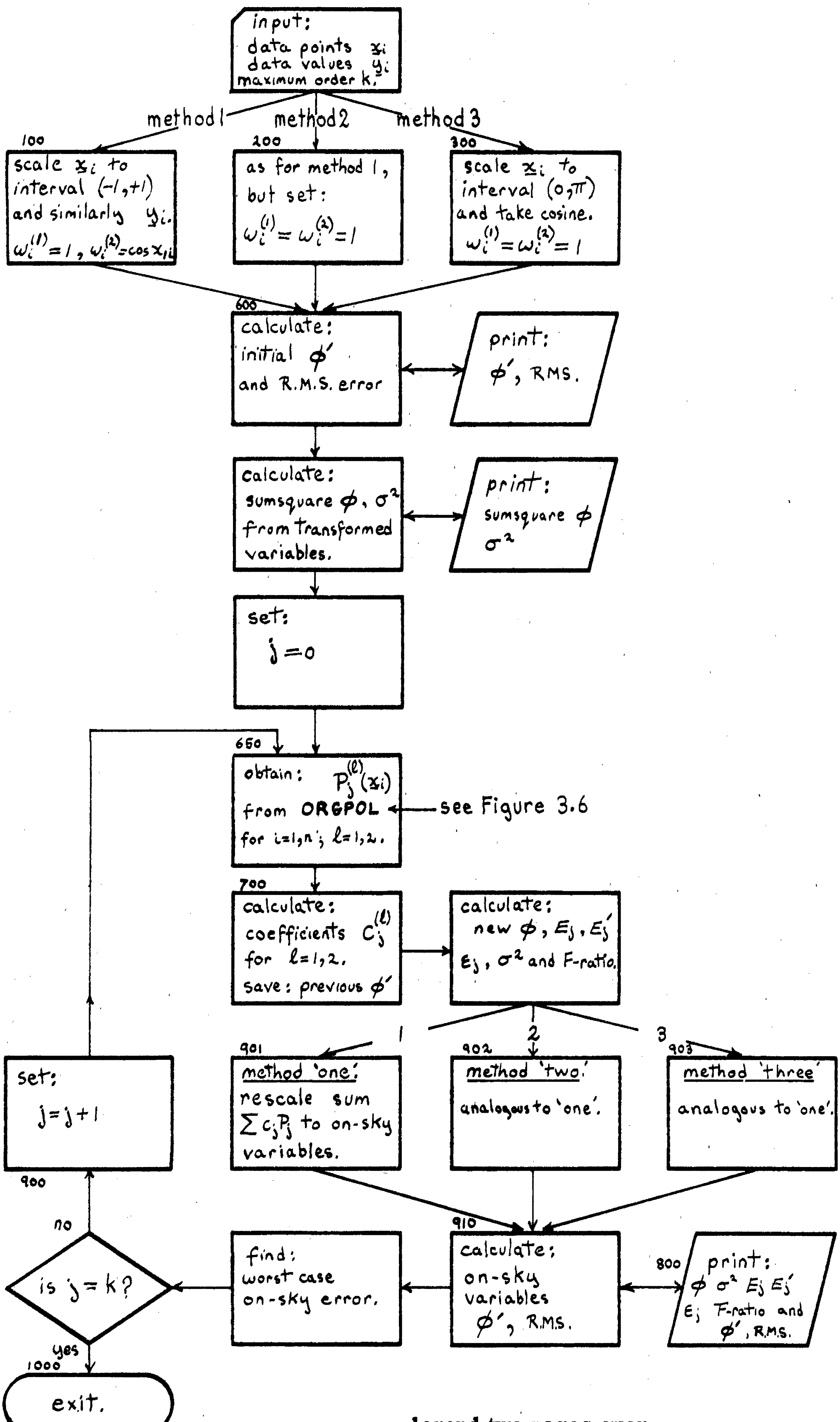
where $\sigma_j^2 = \phi_{(j)} / 2(n-j-1)$, $\qquad \ldots 3.29$

gives a measure of the improvement obtained by adding the j^{th} term, and is used by Hayes, Payne and others (see for example Hayes 1970). The routine calculates F after each order has been added and we discuss this in the next section (3.8).

A method for checking for numerical error propagation and loss of accuracy is included in the routine. Clenshaw and Hayes (1965) show that the expected fractional error in coefficient c_j is given by:

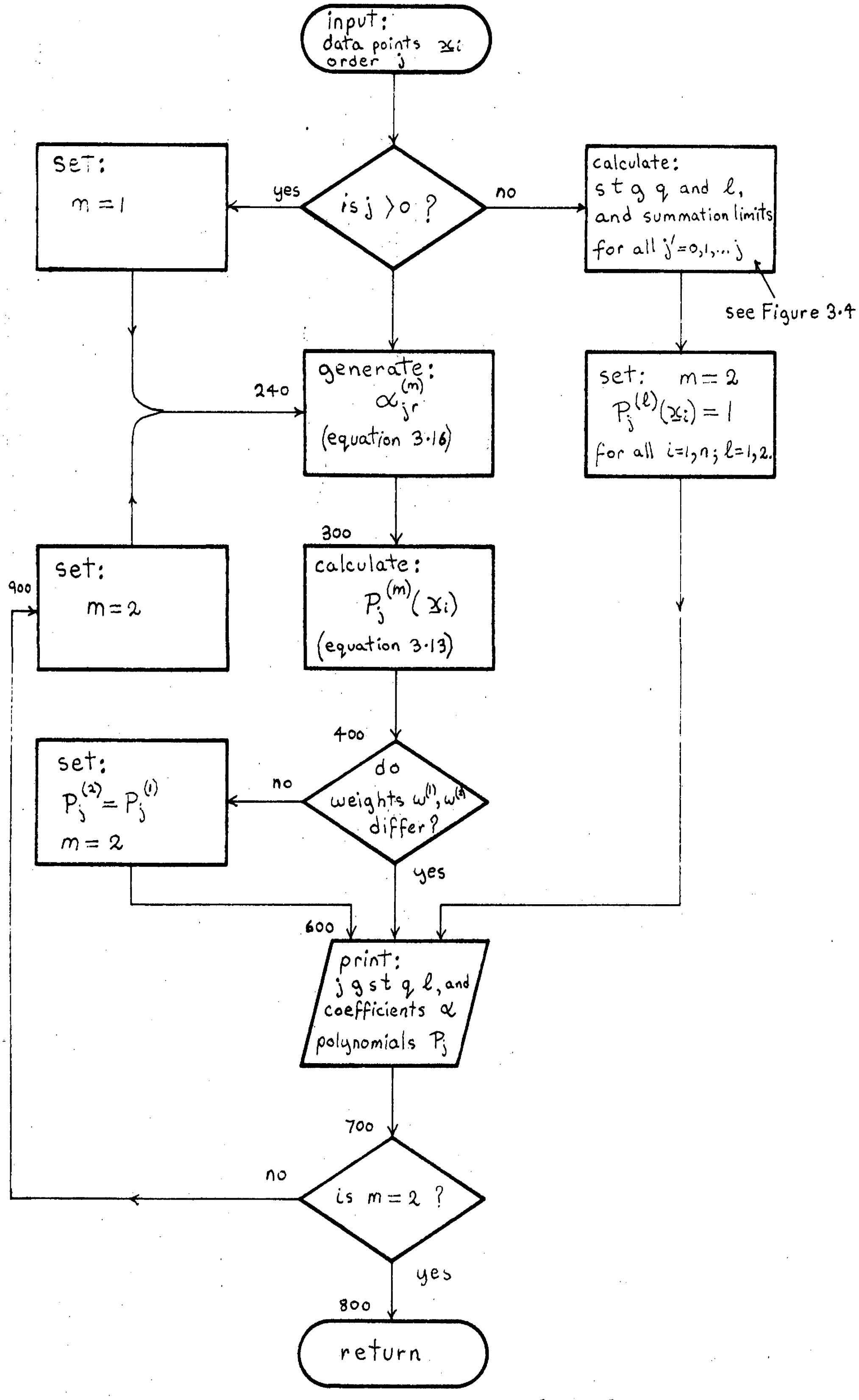
@ Note that it is not possible to use the sine in such a way because sin nx cannot be expressed as a power series in sinx and it would also impose the constraint $y(0) = y(\pi)$.

FLOWCHART FOR SURFACE FITTING ROUTINE ESFIT.



legend two pages over........

FLOWCHART FOR ORTHOGONAL POLYNOMIAL GENERATING ROUTINE ORGPOL.



legend over page.......

LEGEND TO FIGURES 3.5 and 3.6

The small numbers against the flowchart blocks correspond to the labels of FORTRAN statements in the computer code listings in Appendix D; other symbols are as follows:

x_{mi} = independent data vector

$y_i^{(l)}$ = dependent data vector

$w_i^{(l)}$ = weights

$\alpha_{jr}^{(l)}$ = Forsythe coefficients see equation 3.16

$P_j^{(l)}$ = $P_j^{(l)}(\underline{x}_i)$ = Orthogonal Polynomials

$c_j^{(l)}$ = Polynomial series coefficients.

(for the above: $i = 1,\ldots n$; $j = 0,\ldots k$; $m = 1, 2$; $l = 1, 2$; r see below)

r = summation index in equation 3.13, is always less than the order j.

n = number of data points.

k = maximum order of fit.

ϕ, σ^2 = sumsquare and variance respectively, obtained from the transformed variables.

R.M.S. = R.M.S. error on-sky.

ϕ' = sumsquare obtained from on-sky variables.

E_j, E'_j, ε_j = parameters related to numerical accuracy tests, see text.

F = F-ratio, see text.

j = current order of fit.

g = group containing polynomial P_j.

s, t : P_j introduces monomial term $x_1^s . x_2^t$.

q, l : special indices (peculiar to order j) required by equation 3.13.

$$\varepsilon_j = \underbrace{c_j}_{\text{theoretically exact}} - \underbrace{c_j}_{\text{actually computed}} \simeq (E_j/E'_j - 1)/2 \qquad \ldots\ 3.30$$

where E'_j is the reduction in ϕ when going from order $j-1$ to order j obtained in theory, that is by taking the last two terms of equation 3.21, and E_j is the value of the reduction in ϕ actually obtained by evaluating the fitting functions $\psi_i^{(1)}$, $\psi_i^{(2)}$ and then computing ϕ from equation 3.19. The parameter ε_j is also computed at each order of fit.

(3.8) THE COMPUTER ROUTINES AND NUMERICAL RESULTS

The error surface fitting routine was programmed in FORTRAN on a Univac 1108 using double precision arithmetic (60 bit floating point mantissa). A data generating routine (called DATGEN in computer listings) which synthesizes an erratic telescope mounting using the five parameter model of Chapter 2 is included, and is similar to the routine of the same name in Chapter 2 except that the $\underline{x}$ data are distributed randomly rather than in a grid. A flowchart of the actual fitting routine ESFIT is shown in Figure 3.5 and the orthogonal polynomial generating routine ORGPOL called by ESFIT is flowcharted in Figure 3.6. The only other major component of the fitting package is a routine FITEST which takes the fit obtained from ESFIT, as stored in the coefficients $c_j^{(1)}$ and $\alpha_{jr}^{(1)}$ and compares it with additional data generated from a different pseudo-random $\underline{x}$ distribution.

Four different sets of model parameters were used in the testing of ESFIT; these are given in Table 3.2 and referred to as parameter sets A, B, C and D.

TABLE 3.2

parameter set	model parameters (radian) b_1	b_2	b_3	b_4	b_5	generated on-sky R.M.S. error arcsecond
A	.001	-.001	-1.001	.001	.001	231.3
B	.0001	-.0005	-2.0005	.0002	.0002	104.2
C	-.0002	-.0002	-1.0002	.0001	.0001	71.9
D	.0001	-.0001	- .0001	.0001	.0001	29.2

Data was generated at 100 points (n=100) using each of the parameter sets above and fitted using the three methods of ESFIT with maximum order k set to 32. Table 3.3 gives the on-sky R.M.S. error and the value of σ^2 (calculated from equation 3.29 with the transformed variables) at certain

TABLE 3.3

model	order of fit k	method 1 σ^2	method 1 R.M.S.	method 2 σ^2	method 2 R.M.S.	method 3 σ^2	method 3 R.M.S.
generated from parameters A R.M.S.=231 arcsecond	6	$.51^{-8}$	20.	$.50^{-7}$	32.	$.94^{-7}$	37.
	10	$.22^{-8}$	13.	$.13^{-7}$	21.	$.54^{-7}$	34.
	15	$.44^{-9}$	5.6	$.24^{-8}$	7.5	$.27^{-7}$	21.
	21	$.13^{-9}$	2.9	$.44^{-9}$	3.9	$.18^{-7}$	19.
	28	$.25^{-10}$	1.2	$.67^{-10}$	1.5	$.75^{-8}$	13.
B 104"	6	$.15^{-9}$	3.4	$.16^{-8}$	5.7	$.45^{-8}$	7.9
	10	$.70^{-10}$	2.3	$.44^{-9}$	3.7	$.29^{-8}$	5.0
	15	$.15^{-10}$	1.0	$.79^{-10}$	1.4	$.15^{-8}$	5.0
	21	$.38^{-11}$	.51	$.18^{-10}$	.69	$.96^{-9}$	4.3
	28	$.78^{-12}$	.22	$.32^{-11}$	.29	$.40^{-9}$	3.0
C 72"	6	$.12^{-9}$	3.1	$.55^{-9}$	3.8	$.11^{-8}$	4.9
	10	$.30^{-10}$	1.5	$.12^{-9}$	2.0	$.58^{-9}$	4.0
	15	$.44^{-11}$	.56	$.22^{-10}$	.73	$.21^{-9}$	2.2
	21	$.98^{-12}$	2.6	$.54^{-11}$	.34	$.10^{-9}$	1.6
	28	$.22^{-12}$	.12	$.13^{-11}$	.17	$.50^{-10}$	.95
D 29"	6	$.28^{-10}$	1.5	$.12^{-9}$	1.9	$.16^{-9}$	2.1
	10	$.30^{-11}$	.48	$.19^{-10}$	.68	$.44^{-10}$	1.1
	15	$.97^{-12}$	2.6	$.46^{-11}$	.36	$.22^{-10}$	.73
	21	$.19^{-12}$	.11	$.89^{-12}$	.15	$.12^{-10}$	.47
	28	$.46^{-13}$	.05	$.17^{-12}$	.07	$.75^{-11}$	.41

N.B. Superscripts denote decimal exponent e.g. $.51^{-8} = .51 \times 10^{-8}$.

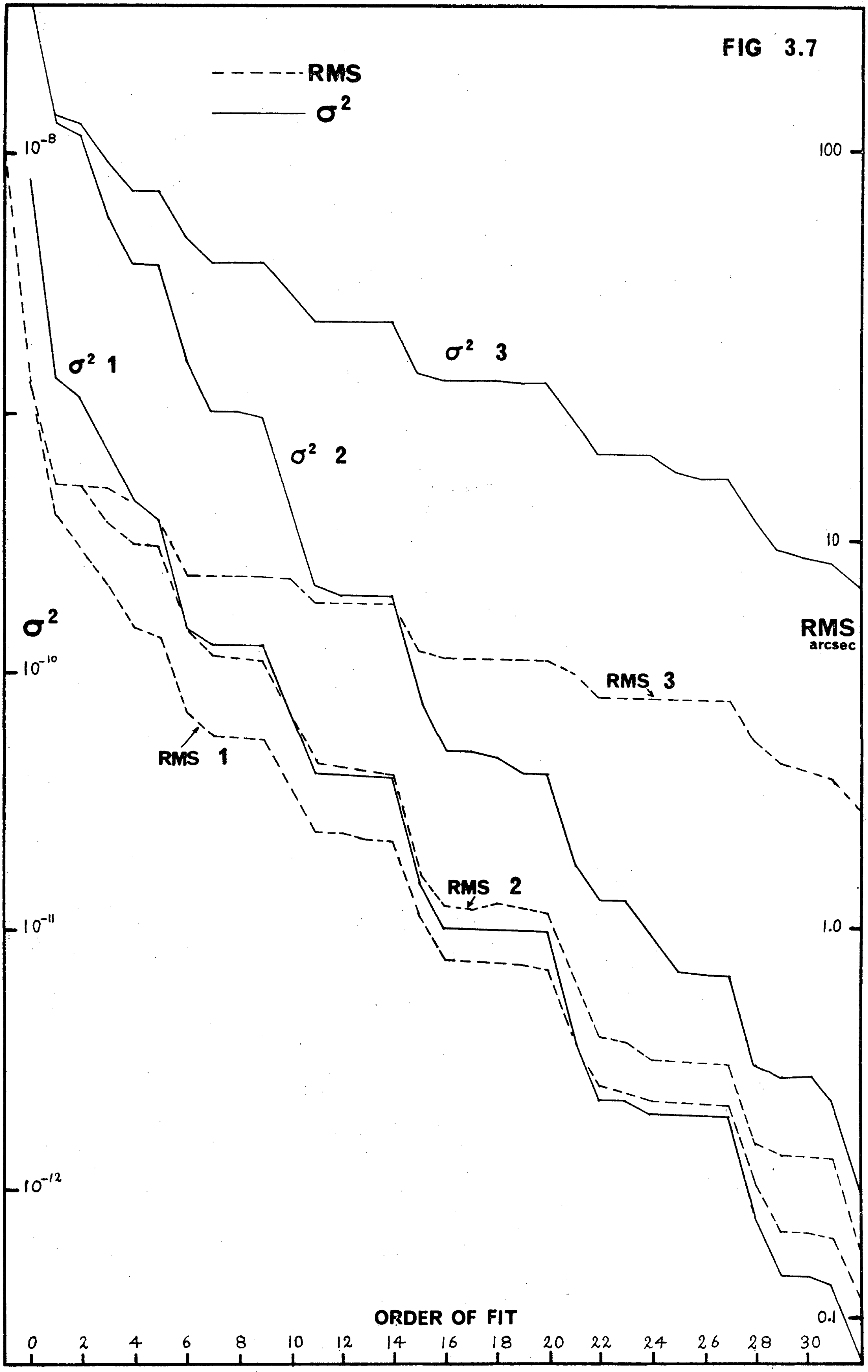


FIG 3.7
RMS
σ²
σ² 1
σ² 2
σ² 3
RMS 1
RMS 2
RMS 3
σ²
10⁻⁸
10⁻¹⁰
10⁻¹¹
10⁻¹²
100
10
1.0
0.1
RMS
arcsec
ORDER OF FIT
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30

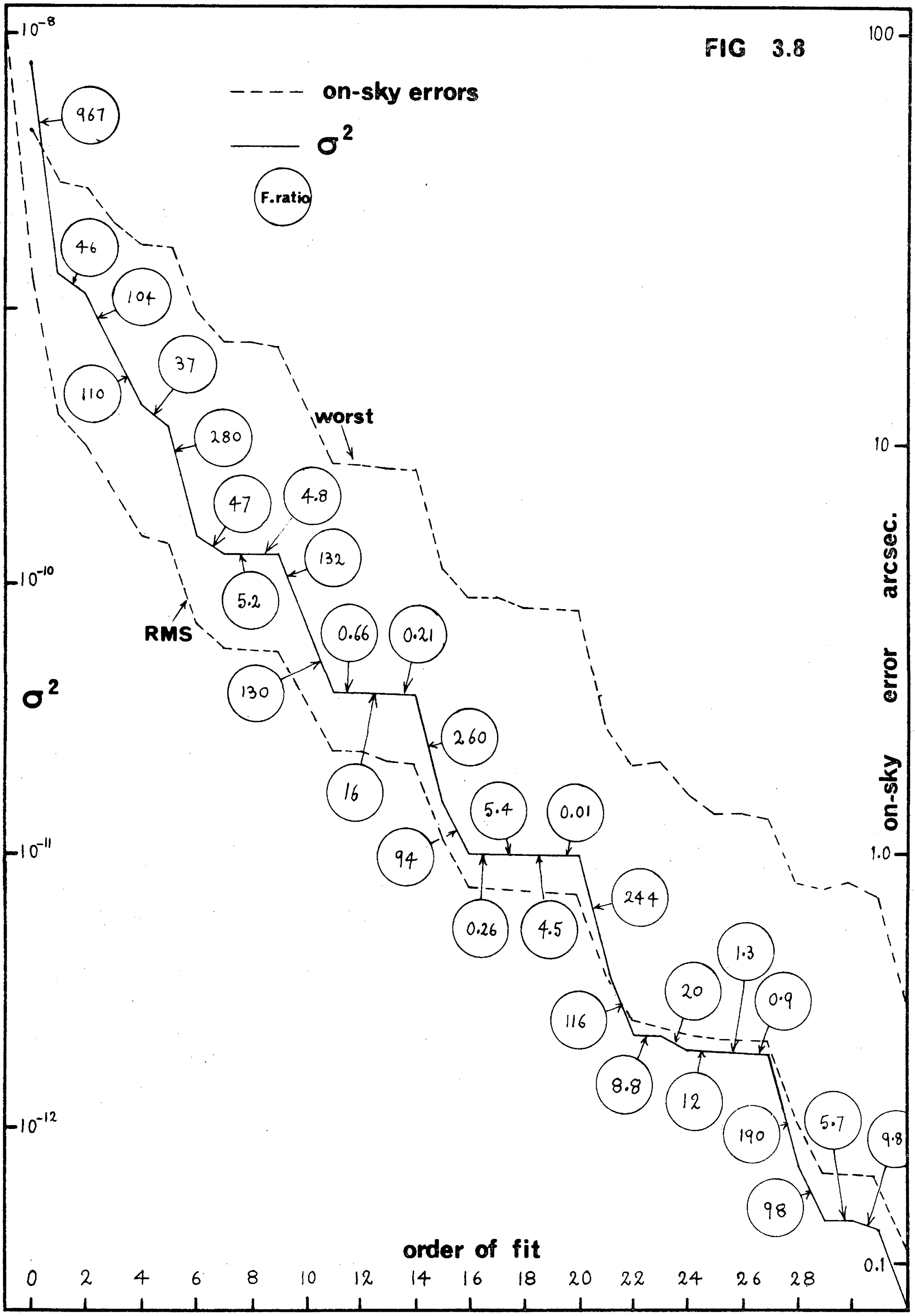
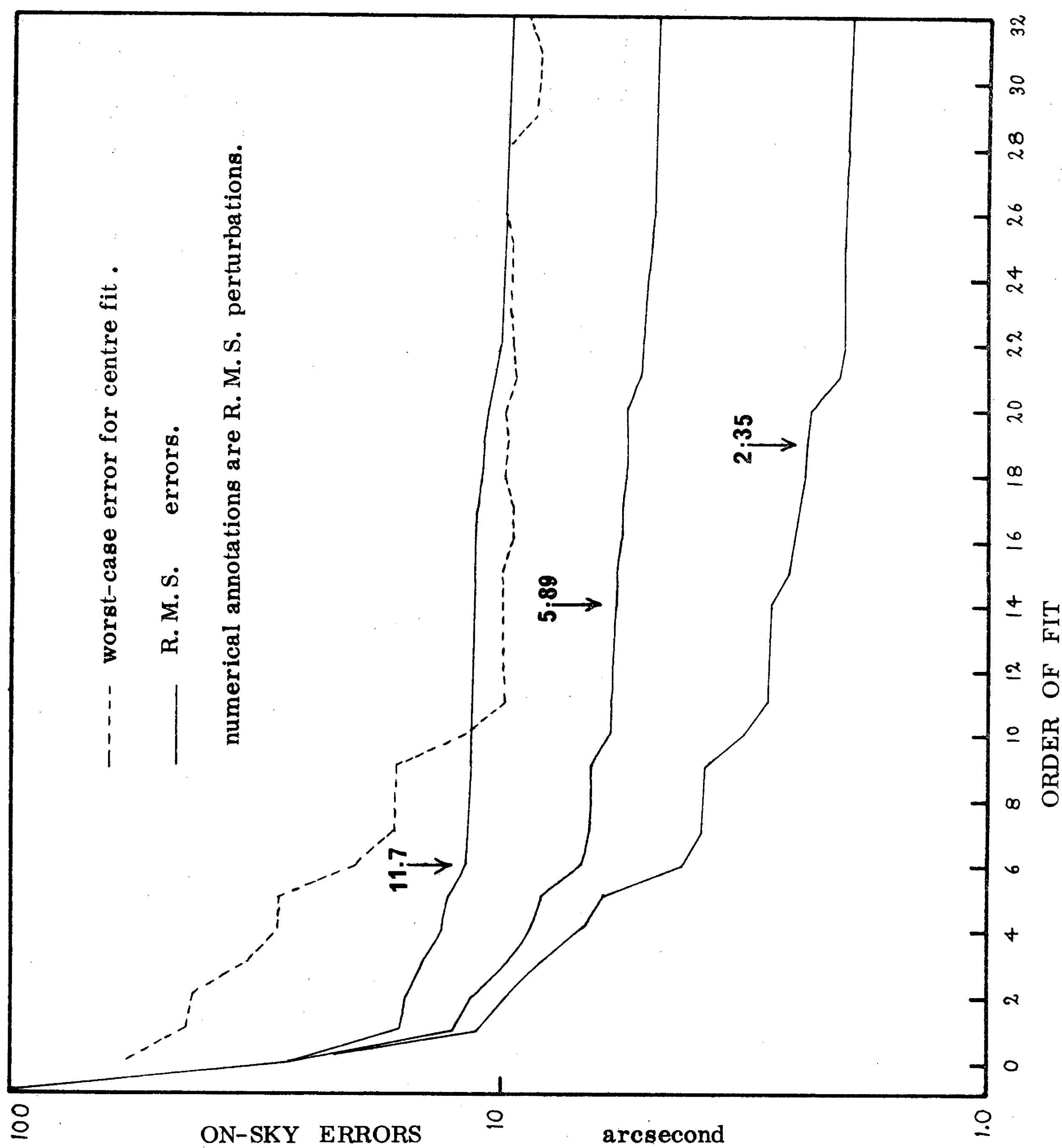


FIG 3.8
on-sky errors
σ²
F.ratio
worst
RMS
σ²
on-sky error arcsec.
order of fit
10⁻⁸
10⁻¹⁰
10⁻¹¹
10⁻¹²
100
10
1.0
0.1
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28
967
46
104
37
110
280
47
4.8
132
5.2
0.66
0.21
130
260
16
5.4
0.01
94
244
0.26
4.5
1.3
20
0.9
116
8.8
12
190
5.7
9.8
98

FIG 3.9



- - - worst-case error for centre fit .
—— R. M. S. errors.
numerical annotations are R. M. S. perturbations.
11.7
5.89
2.35
ORDER OF FIT
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
ON-SKY ERRORS arcsecond
100
10
1.0

orders of fit for each of the parameter sets and methods. In all twelve fits σ^2 decreases monotonically with the order of fit; the quantity ε_j (equation 3.30) is about 10^{-16} to 10^{-17} initially, and no larger than 10^{-12} around order 32, thus we are well away from conditions under which we would need to prematurely cease our fitting process because of numerical loss of accuracy.

The on-sky R.M.S. error behaves similarly to σ,^{2} excepting in methods two and three where there are occasional slight increases going from one order to the next. This is, of course, to be expected. However the decrease in σ^2 going from start to a given order, for methods two and three, is not impressive when compared to method one, and in Figure 3.7 we show the fits generated for parameters B. The R.M.S. error (broken lines) and σ^2 (solid) are plotted for each order and demonstrate firstly, the different performances of the methods, but also that, in each method, most of the decrease in σ^2 or R.M.S. occurs when certain specific terms e.g. x_1^5 at order 15, are added. Figure 3.7 and Table 3.3 confirm that there is little point in minimizing an objective function other than the expression for ϕ which represents the on-sky error, and in all the following computer runs method one alone was used.

The plot of σ^2 against polynomial order for ESFIT (method one) on data from parameters B is shown again in Figure 3.8, this time with the value of F calculated from equation 3.28 annotated. The critical value which F must exceed for the j^{th} order fit to be considered significant depends on (i) the number of degrees of freedom of the numerator of equation 3.28 which is 1; (ii) the number of degrees of freedom in the denominator which is 2(n-j-1), and (iii) the desired risk of falsely accepting the fit. For values of 2(n-j-1) between 10 and 400 this F value lies roughly between 5.0 and 3.9 respectively for a 5% risk, increasing to between 21 and 10 for a 0.1% risk. In Figure 3.8 we observe occasional runs of quite low F preceeding an order with a high F and which manifestly leads to a substantial improvement of fit. Thus, even more so than in curve fitting, it is virtually impossible to use the F ratio as a simple test for the order at which to discontinue the fitting process, at least for the case of exact data. The test is only slightly more useful in the case of data which includes random perturbations, for example in the runs shown in Figure 3.9 below. The two broken lines in Figure 3.8 are the on-sky worst case error (upper plot), and the R.M.S. error (lower plot). Both of these follow the step decreases in σ^2, the maximum error being always of the order of 3 to 5 times the R.M.S. error.

To produce Figure 3.9, data was generated from parameters B and perturbed by an additional 11.7, 5.89 and 2.35 arcsecond R.M.S. error (on-sky). The R.M.S. error of fit for each order is shown for these three cases, and the point at which the fit error is less than that introduced randomly into the data marked with an arrow. The routine manages to fit some of this random error since the number of data points (n=100), though adequate, is not large. The F-ratio was found to decrease suddenly at orders 7, 8 and 8 respectively for the fits, and although these points represent practical cut-off limits for the first two, the use of the test on fit three (random data R.M.S. = 2.35") would result in premature termination of the fit. The broken line in Figure 3.9 represents the worst case on-sky error for the centre fit (random data R.M.S. = 5.89") and is approximately 1.6 times the R.M.S. error at the point where a satisfactory fit has been achieved, whereafter it begins to oscillate.

Tables 3.4 and 3.5 are the results of using the routine FITEST to test fits generated by ESFIT to data from parameters B for various values of data point number n, and order k. In each case FITEST was applied using six different test data sets; these are labelled 1 to 6 while the original data from which the fit was generated is labelled 0. In Table 3.4 fits of order 10 were generated for numbers of data points equal to 200, 140, 100, 70, 35 and 20. The upper of the two entries is the R.M.S. on-sky error (in arcseconds) which results from evaluating the fit at the points of the new data set, whilst the lower figures in parentheses are the corresponding worst case on-sky errors. The last column contains the largest of these worst case errors for the data sets 1 to 6, and the penultimate column contains the largest value of the ratio of R.M.S. error of fit to the R.M.S. error of fit for the original data (data set 0).

There is no simple statistical test which one can apply to Table 3.4, but it is clear that for the case of k=10 decreasing the number of data points below about 100 rapidly increases the errors at points other than the original data points. Apart from a few spurious good fits the worst case error is approximately constant at 10 to 16 arcsecond for $n \geqslant 100$. Note that the extremely low R.M.S. error for the original fit in the case n=20 shows we are approaching the interpolation case where n=k+1 and the residual error goes to zero.

In Table 3.5, which is interpreted similarly to Table 3.4, n has been held constant at 100 and fits of orders k = 4, 6, 11, 15, 22, 28, 36 and 44 generated. Again it is clear that for n=100 there seems little point fitting to an order greater than about 11. However extremely good fits can be generated for orders of 20 and above and although the R.M.S. errors of fit for the additional data are poor compared to the original fit, both

TABLE 3.4

number of data points n	data set number of additionally generated data 0	1	2	3	4	5	6	R.M.S. ratio (see text)	largest worst case error
200	2.1	2.6	2.1	2.2	2.2	1.9	1.9	1.22	
	(14)	(16)	(12)	(15)	(15)	(15)	(12)		(15.6)
140	2.1	2.8	2.3	2.8	2.6	2.5	1.9	1.37	
	(13)	(16)	(6.7)	(16)	(15)	(16)	(12)		(16.1)
100	2.3	3.9	2.9	3.5	3.1	3.8	4.8	1.71	
	(12)	(9.2)	(5.8)	(16)	(15)	(16)	(14)		(16.2)
70	2.1	12	4.4	6.1	5.8	5.3	4.5	5.36	
	(8.9)	(21)	(11)	(19)	(23)	(17)	(14)		(22.6)
35	2.0	16.3	3.9	11	5.0	9.8	13	7.99	
	(4.9)	(25)	(9.1)	(18)	(8.2)	(25)	(25)		(25.2)
20	1.4	17	12	9.9	9.2	9.4	13	12.4	
	(3.0)	(38)	(22)	(21)	(23)	(26)	(29)		(37.7)

UPPER figures are R.M.S. errors.

LOWER figures are worst case errors.

TABLE 3.5

order of fit k	data set number of additionally generated data 0	1	2	3	4	5	6	R.M.S. ratio (see text)	largest worst case error
4	6.1	6.6	5.9	8.4	8.2	7.8	8.1	1.36	
	(29)	(22)	(24)	(32)	(31)	(32)	(30)		(32.0)
6	3.5	4.1	3.7	5.3	4.8	5.2	6.4	1.84	
	(20)	(9.8)	(13)	(23)	(22)	(23)	(21)		(23.4)
11	1.7	4.6	2.7	3.0	2.7	3.9	4.8	2.74	
	(8.9)	(13)	(6.6)	(11)	(10)	(12)	(11)		(12.8)
15	1.0	4.7	2.8	2.4	2.1	3.6	5.6	4.50	
	(4.7)	(13)	(6.4)	(8.1)	(6.5)	(8.4)	(6.6)		(13.1)
22	.38	4.6	2.6	2.2	1.8	3.6	4.8	12.5	
	(1.7)	(16)	(6.9)	(5.1)	(3.7)	(6.5)	(6.2)		(16.2)
28	.22	4.5	2.7	2.1	1.7	3.5	4.7	21.8	
	(.83)	(15)	(8.2)	(4.6)	(2.8)	(5.2)	(6.0)		(14.7)
36	.08	4.3	2.6	2.1	1.7	3.4	4.7	56.4	
	(.31)	(14)	(8.7)	(4.9)	(2.6)	(5.2)	(6.0)		(14.3)
44	.04	4.2	2.6	2.1	1.8	3.5	4.7	107.	
	(.16)	(13)	(8.6)	(5.5)	(3.7)	(6.2)	(6.3)		(13.4)

UPPER figures are R.M.S. errors.

LOWER figures are worst case errors.

the R.M.S. errors and the worst case errors steadily improve with the order of fit, and are a worthwhile improvement over the original R.M.S. errors in the raw data (104.2 arcsecond). This suggests that, in fact, it may be beneficial fitting to reasonably high orders (e.g. 20 to 30) even with insufficient data, provided one can tolerate an error at a given interpolation point which is many times the R.M.S. error of fit. Statistical techniques from regression theory permit calculation of confidence intervals for the value the fit predicts at a point of interpolation, but this has not been treated here.

(3.9) SURFACE FITTING: CONCLUDING DISCUSSION

The computations of the previous section show the fitting of error surfaces by orthogonal polynomials in the normalized co-ordinates to be a stable and well-behaved process. Various techniques which reduce rounding errors are referred to in Cadwell and Williams (1961), but even without these, the fitting routine was at no stage prejudiced by numerical errors. Clenshaw and Hayes (1965) experimenting with curve fitting up to very high orders (e.g. 90) found that E_j (see equation 3.30) eventually departs from E_j' and changes sign for high orders due to cumulative numerical errors, but there is no trace of this occurring in the fits above, primarily because of the arithmetic precision used, but also because even with order k set to 44 the highest power of x_1 or x_2 introduced is only 8.

An important consideration involved in the fitting process is that an adequate number of data points be used for the particular order fitted. Hayes (1970) by considering the extrema of the k^{th} degree Chebyshev polynomial gives an upper limit of π/M for k where M is the largest difference between the inverse cosines of adjacent data points. For evenly spread data this limit is very approximately $\sqrt{2n}$ where n is the number of data points. For the two-dimensional case there is no analogous argument, but simple mindedly we can take it also to be of order $\sqrt{2n}$ @. This is in agreement with the results shown in Tables 3.4 and 3.5 and suggests that only for impracticably large amounts of data would such surface fits acquire statistical validity; despite this, surface fitting undoubtedly constitutes a practically useful technique for telescope improvement.

The preceeding results suggest that where the telescope behaviour

---

@ Imagine the process as curve fitting with $\sqrt{n}$ points in each dimension separately, in which case the limit above is $\sqrt{2}.n^{\frac{1}{4}}$, but for surface fitting the order k is related to the index t of x_2^t (or x_1^t) approximately by $k \approx t^2/\sqrt{2}$, hence $k \approx \sqrt{2n}$.

follows the assumed mathematical model very closely, surface fitting produces significantly poorer results than we would expect a model estimation process to, and despite the simplicity of the particular model used here, this statement is most probably true in general. The next chapter (Chapter 4) shows that such mathematical models can often produce disappointing results when applied to practical data, and that in these cases the utility of surface fitting is enhanced. The reason for this is that in surface fitting, the orthogonal polynomials span the set of all polynomials of degree $\leqslant k$, and thus may completely represent all of the functions necessary for the description of the telescope errors. Moreover there is no reason to consider the use of any other type of polynomials, since fitting with them would achieve identical results but would be more prone to numerical error problems. Finally, the numerical results here also show that the definition of ϕ given in equation 3.19 is an appropriate objective function to minimize, and that the use of trigonometric functions of δ and H as our independent variables may not offer any advantage; the results for the case given here in fact show it to be inferior.

An extremely attractive technique which has found wide application and considerable success in curve fitting is that of piece-wise fitting. Many functions which are fitted only poorly in their entirety by high degree polynomials or other complicated fitting functions are quite adequately approximated by simple functions (e.g. cubic splines) if fitted piece-wise, especially if the boundaries between pieces can be optimally positioned. A suitable scheme for optimizing the configuration of pieces in two dimensions and ensuring that the boundary conditions are met unfortunately does not exist, and in the absence of such scheme the most useful technique is probably to model estimate for those error causes which are large, highly repeatable and well understood, and to use orthogonal polynomial surface fitting to further reduce residuals.

CHAPTER FOUR
THE APPLICATION OF SOFTWARE CORRECTION TECHNIQUES TO POINTING DATA FROM THE MT. STROMLO 74-INCH TELESCOPE

(4.1) PRELIMINARY

In the previous two chapters the stability and relative efficiency of various algorithms were investigated using data synthetically generated with a simple but representative model of telescope pointing errors. This chapter is a practical assessment of the application of the surface fitting and one of the parameter estimation algorithms to pointing error data obtained experimentally from the 74-inch equatorial reflector of Mount Stromlo Observatory, Department of Astronomy, Australian National University. The relevant descriptive details of this instrument appear in Appendix C, and it is sufficient here to note that it is not renowned for accurate pointing, being afflicted with a number of systematic and hysteresial errors, which can cause discrepancies of up to 3 minute of arc at large zenith and hourangles. As was noted in Chapter 1, our problem is more one of response surface optimization rather than mechanism determination,[@] yet in addition to a reduction in R.M.S. pointing error for the 74-inch, it is shown later that some conclusions regarding the nature and causes of error are indeed possible.

(4.2) POINTING DATA ACQUISITION

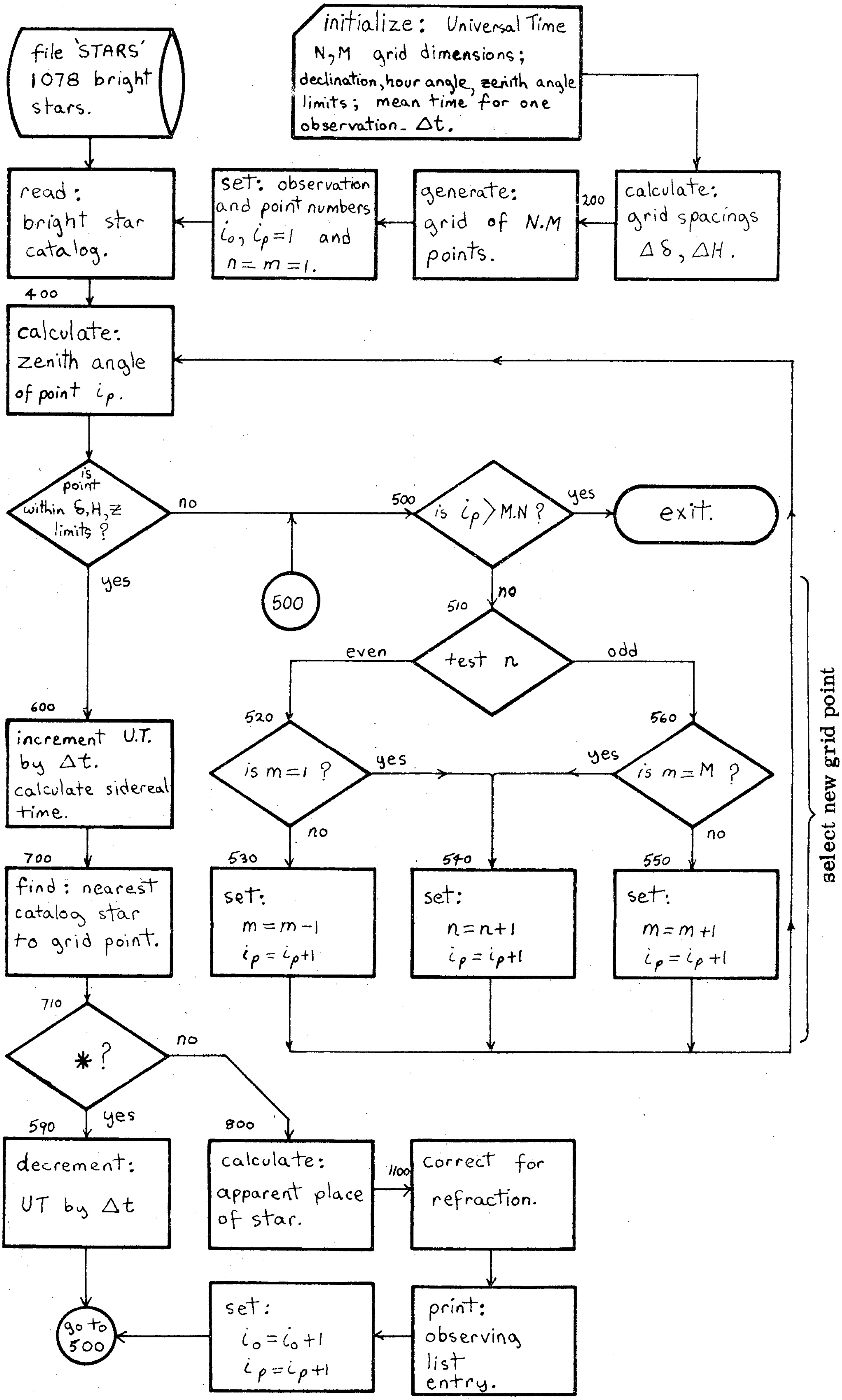
An attempt to obtain pointing data in late 1969 using a single operator and the existing selsyn position readout system was quite unsatisfactory. It resulted in an inadequate amount of data which contained large hysteresial errors inherent in the selsyn system, which were due to backlash in the selsyn transmitter gearing and stiction in the passive receivers. The author spent some time designing a digital readout system for the 74-inch using 15 bit optical shaft encoders geared 27:1 (47.25:1) to the polar (declination) axis, together with 8 bit brush contact encoders geared 1:1 with each axis to remove the readout ambiguity. The shaft encoder data reduction and display generation is performed by a Hewlett Packard H.P. 2100A minicomputer, which in addition to this function has become useful as a versatile on-line machine for astronomical data acquisition and instrument control. An improved way of generating sidereal time was devised (see Hovey 1973 or Appendix E) and the complete encoder and

---

@ In addition to Chapter 1, see for example Box and Hunter (1965) or Box and Coutie (1956).

FIG 4.1

FLOWCHART FOR CATALOG. OBS.



* has this star already been used for an adjacent grid point?

timing system is described in Appendix C[@]. In addition, a program enabling the rapid logging on teleprinter and paper-tape of the instantaneous position of the telescope axes and various diagnostic data from the encoder system was implemented[@@]. Thus, provided the co-ordinates of the next desired object are immediately available, data collection can proceed at a rate limited only by the time taken to set the telescope.

A routine was devised by the author which generates an observing list of bright stars distributed so that (i) the desired area of sky is evenly covered and, (ii) the sequence in which the stars are observed is such that the telescope need only be moved a minimum distance between observations. The total number of grid points used is set on the basis of an estimated rate of data collection (mean time between observations) and the grid spacing calculated from assumed co-ordinate limits for the telescope. Condition (i) above is achieved by generating a grid with uniform declination increments, but with hourangle increments inversely proportional to $\cos\delta$ (thus a rectangular grid in the variables δ and $H\cos\delta$ is generated); and (ii) by ordering the points in a rectangular 'zig-zag' manner. The actual list of stars is obtained by searching a computer file of bright star positions for an object whose apparent place corrected for refraction is nearest the generated grid point at the time of observation. This time is incremented by the estimated mean time for an observation whenever the file is searched for a star for the next grid point. The use of the same star for adjacent grid points is inhibited and the routine rejects points which lie outside the δ and H co-ordinate limits and below a zenith angle limit.

A flowchart for this routine, which is called CATALOG.OBS and which was coded in FORTRAN and run on a UNIVAC U1108, appears in Figure 4.1. It uses the author's UNIVAC system file (called STARS) containing the 1970.0 mean co-ordinates and magnitude data for the 1078 bright stars listed in the Astronomical Ephemeris, and employs algorithms for mean to apparent place computation and refraction correction which are described in Appendix A. A mean observation time of 3 minutes was assumed and a 19 by 19 point grid generated for observing lists for a pointing data run in early March 1973. The observations were made with the telescope for the most part east of the polar axis and used a Cassegrain offset guider head centered

@ Acknowledgement is due to Wayne Ruting (detailed hardware design), Ron Howe (software modifications and final implementation) and John Hart (shaft encoder mechanics), all of Mt. Stromlo Observatory.

@@ written and implemented by Ron Howe.

FIG 4.2

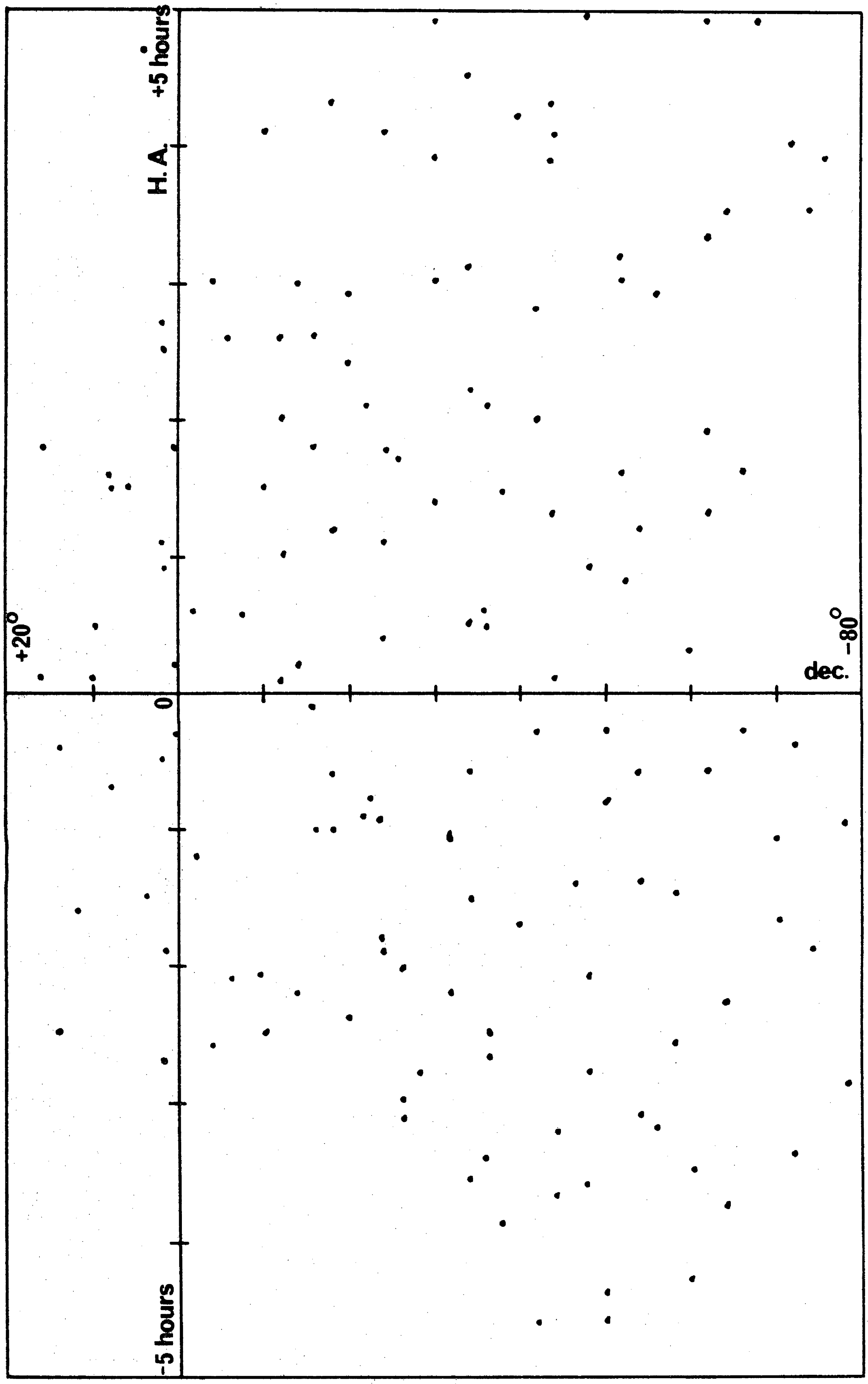
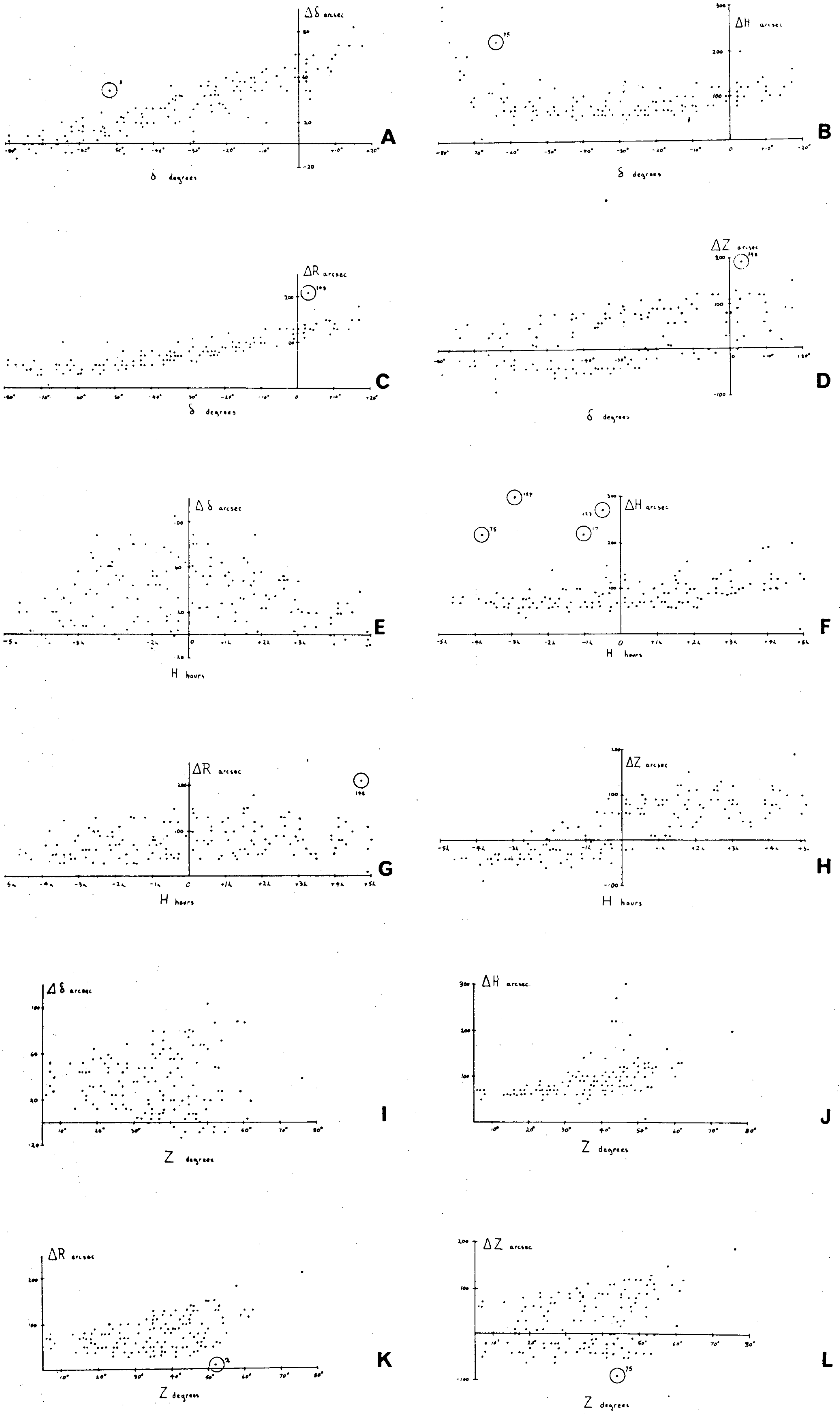


FIG 4.3



A
Δδ arcsec
δ degrees
B
ΔH arcsec
δ degrees
C
ΔR arcsec
δ degrees
D
ΔZ arcsec
δ degrees
E
Δδ arcsec
H hours
F
ΔH arcsec
H hours
G
ΔR arcsec
H hours
H
ΔZ arcsec
H hours
I
Δδ arcsec
Z degrees
J
ΔH arcsec
Z degrees
K
ΔR arcsec
Z degrees
L
ΔZ arcsec
Z degrees

on the rotational axis of the instrument mount. In addition to the observer, separate operators were used for the various tasks, setting in both co-ordinates, dome control, computer operation etc., and approximately 160 data points were obtained between $\delta = -80^{o}$ to $+20^{o}$, hourangle limits -4 hours to +6 hours, and above a zenith angle of 65^{o} @. The actual grid of stars used is shown in Figure 4.2. The total time required for the observations was $6\frac{1}{2}$ hours so that a mean observation time of $2\frac{1}{2}$ minutes would have been slightly more appropriate.

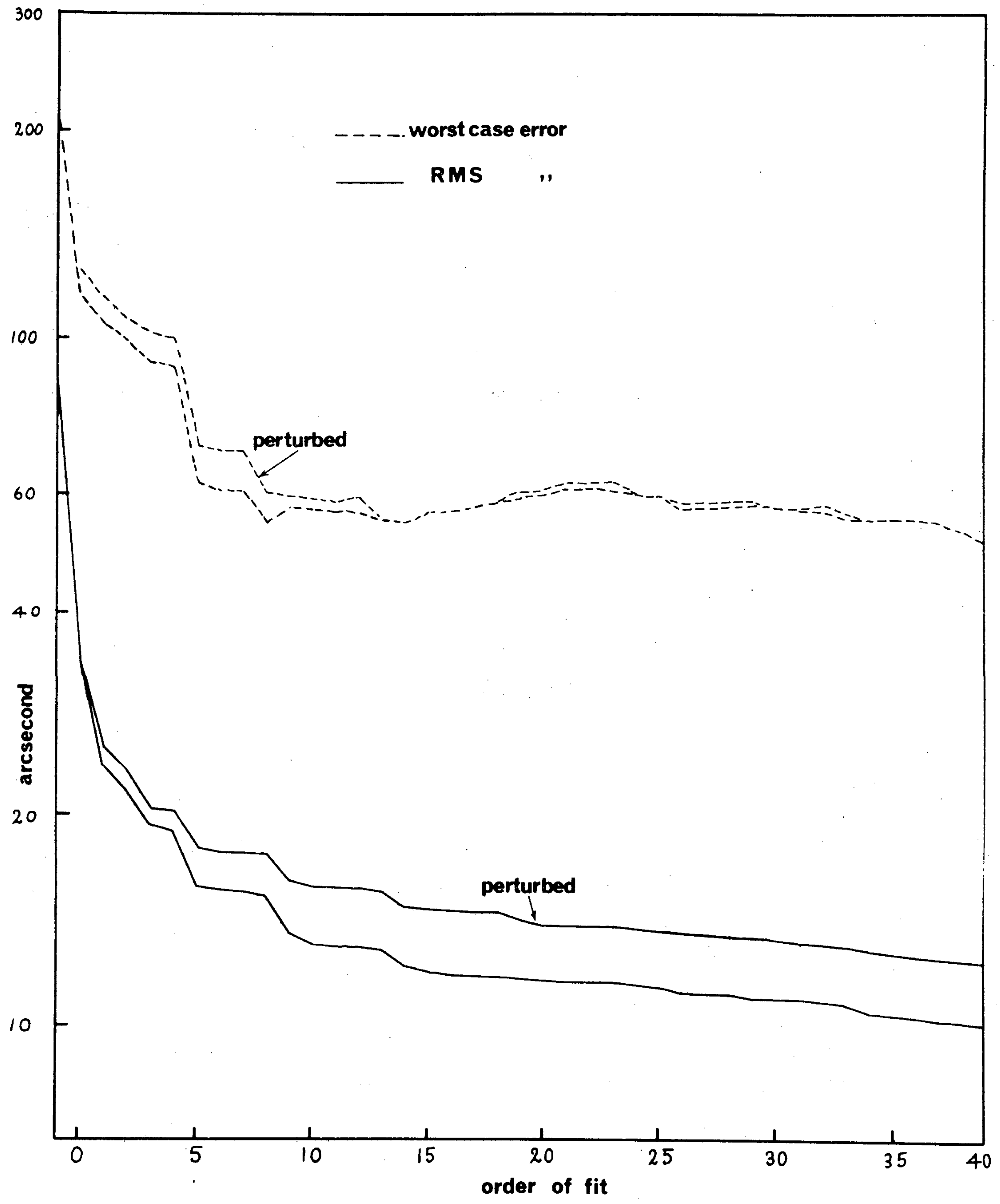
The pointing data was output from the H.P. 2100A on paper-tape and transferred to a U1108 system file for editing and preliminary processing. Some data had to be rejected because of obviously incorrect setting or premature logging of an observation, and an eventual data set of 148 points taken with the telescope east of the polar axis was obtained. This data set was processed by a routine which, using the algorithms for mean to apparent place and refraction correction of Appendix A, generates the $\underline{x}$ and $\underline{y}$ vectors (see equations 2.8, 3.17 and Chapters 2, 3 in general) which constitute the basic input to the surface fitting and parameter estimation routines. This preliminary processor called TA obtains the telescope position directly from the shaft encoder words in the raw data, and since the U1108 can be used with greater arithmetic precision and need not employ certain simplifications designed into the H.P. 2100A software, a useful assessment of the performance of the H.P. 2100A encoder and timing system software is possible. This assessment is mentioned again in Appendix B, and here we simply note that with the addition of some minor adjustments after the pointing tests the H.P. 2100A handling of the encoder and timing data is quite adequate.

Line-printer plots were generated of the pointing errors in declination ($\Delta\delta$), hourangle (ΔH), the resultant error on the sky (ΔR) and the co-ordinate errors resolved in zenith angle (ΔZ), against declination (δ), hourangle (H) and zenith angle (Z), and reductions of these plots are shown in Figures 4.3a to 4.3l. In general they show considerable scatter in the data, though much of this is due to the limitations of graphing a function of more than one argument in such a manner, and to the vagaries of the line-printer. No simple dependencies are evident but definite trends, for example the increase in $\Delta\delta$ and Δ R with more northern declinations (graphs 4.3a, 4.3c), the effect of hourangle on the zenith error

---

@ Note that the grid spacing is approximately 5 degrees and many grid points are not used since they either use the same star as an adjacent one, or exceed the zenith angle limit. The full observing list contained about 200 stars.

FIG 4.4



worst case error
RMS ,,
perturbed
perturbed
arcsecond
300
200
100
60
40
20
10
0
5
10
15
20
25
30
35
40
order of fit

(graph 4.3h), and the sudden nonlinear increase in Δ H, Δ R below 30° zenith angle (graphs 4.3j, 4.3k). We shall allude to these later when discussing the spurious data points and the possible causes of error. The considerable scatter in all the graphs with zenith angle abscissae (graphs 4.3i to 4.3l) show conclusively that there is no simple dependence of pointing error on zenith angle and that there would be little point in transforming our independent variables into an azimuth/zenith angle system.

(4.3) SURFACE FITTING THE 74-INCH TELESCOPE POINTING DATA

The surface fitting routine ESFIT described in Chapter 3 was applied to the 148 point data set with the maximum order of fit k set to 40. As a stability test the data was perturbed by a further 7.7 arcsecond R.M.S. (20 arcsecond peak to peak) error and then fitted. Figure 4.4 shows the R.M.S. error of fit (solid line) and the worst case error (broken line) for the original and perturbed data for sequential orders of fit. As in Chapter 3 most of the decrease in R.M.S. error occurs at certain orders e.g. 5, 9 and 14 where groups of polynomial terms become complete. The coefficients range from 0.27×10^{-2} to 0.7×10^{-6} with some of the larger values occuring for $k > 25$ when the solution attempts to fit the fast variations in the data. The R.M.S. error is 87.09 arcsecond in the original data, and decreases monotonically with the order of fit[@], but the worst case error increases beyond order 14 and even oscillates. The coefficients for the two runs, perturbed and original, differ by up to about 10% which is consistent with the ratio of the perturbation introduced R.M.S. to the original data set R.M.S. error. Numerical error accumulation as measured by the variable ε (see equation 3.30) was negligible.

It is necessary to test in some way the adequacy of the fit generated, that is to assess whether the residual sumsquare ϕ or R.M.S. error that remains after fitting, is primarily due to pure error in the data or inadequacy of the assumed model (in this case orthogonal polynomials). If the assumed model is adequate, the expressions

$$\left.\begin{aligned} \sigma_2^{\,2} &= \phi/2(n-k-1) \\ \text{and} \quad \nu_2 &= 2(n-k-1), \end{aligned}\right\} \quad \ldots \; 4.1$$

where n is the number of data points fitted to order k in both co-ordinates, give estimates of the variance of the experimental errors and its associated degree of freedom respectively. Ideally this should be compared to

---

@ This is inevitable since the R.M.S. error (or in fact ϕ) is the objective function being minimized by the routine.

an independently obtained estimate for the experimental error variance, but the amount of work involved and the disruption to telescope scheduling precludes the collection of substantial additional data. Instead the original data set of 148 points (called here XY1) is subdivided into 2, 3 and 4 data subsets, each containing approximately 1/2, 1/3 and 1/4 of the original 148 points respectively. Since the ordering of the original data set was of no significance this has been done by simply sorting alternate points. The data sets are listed in Table 4.1 with their number of points, R.M.S. and worst case errors.

TABLE 4.1

dataset	number of data points	R.M.S. error arcsecond	worst case error arcsec.
XY1	148	87.09	207.17
XY2	74	86.19	180.18
XY3	74	87.97	207.17
XY4	49	84.63	149.21
XY5	49	85.21	180.18
XY6	49	87.23	153.61
XY7	37	83.35	142.47
XY8	37	83.74	149.21
XY9	37	88.95	180.18
XY10	37	92.01	207.17

The nine data sets XY2, XY3...XY10 were fitted to maximum order 2, 5, 9 and 20 and the resultant fits tested against the data in XY1. Although this procedure is a more valid test than a succession of tests of, for example, XY4 against XY5 etc., none of the nine data subsets is entirely disjoint@ with the complete set XY1. Thus the variance and associated degree of freedom with which σ_2^2 in equation 4.1 must be compared, are given by

$$\left.\begin{aligned} \sigma_1^2 &= (\phi' - \phi) \,/\, 2(n' - n) \\ \text{and} \quad \nu_1 &= 2(n' - n) \end{aligned}\right\} \quad \ldots \; 4.2$$

where ϕ and n are as in equation 4.1, ϕ' is the residual sum of squares resulting from the test on a data set with n' (=148) points. This

---

@ disjoint in the sense of having no data points in common.

expression allows for the points which are common to the fitted and test data sets which in our case comprise all of the points in the fitted set. Note that in the case of complementary data sets XY2, XY3, testing the fit to XY2 on XY3 is equivalent to testing the same on XY1. If all data sets were quite disjoint, the more usual techniques of analysis of variance would be appropriate, and if repeat observations at the same point were practicable, a number of studies such as Anscombe and Tukey (1963) on the treatment of residuals, which involve two-way classifications of the data, could also be applied.

The results of these tests appear in Table 4.2 to Table 4.5, and the ratio σ_1^2/σ_2^2 calculated from expressions 4.1, 4.2 is compared with $F_{1-\alpha}(\nu_1,\nu_2)$ the upper α point of the Fisher distribution. If

$$\sigma_1^2/\sigma_2^2 \geqslant F_{1-\alpha}(\nu_1,\nu_2) \qquad \ldots\ 4.3$$

we reject the model or fit as being inadequate to describe the data in question, and we do so with significance level $100\alpha\%$, that is a $100\alpha\%$ probability that we have wrongly rejected it when in fact it was adequate. This test is common in regression theory and is used in Box and Coutie (1956), Beale (1960), and Draper and Smith (1966). The ratio σ_1^2/σ_2^2 is given in the tables for all of the tests, and the penultimate row contains the percentage level α at which the fit being tested would be rejected on the basis of relation 4.3 above; significances of 1% and 5% are in common usage. Together with the other quantities indicated by '@', α is averaged over the group of tests on data sets of similar number of points, but others like the R.M.S. and worst case errors are given for each test to show the variation involved. The final row gives the ratio of the degree of freedom of fit to the number of fitted parameters, and is discussed later.

Two other indications of the performance of the fits included in Tables 4.2 to 4.5 are R^2 the multiple regression coefficient, and the mean square ratio MSR. R^2 is expressed as a percentage and measures the proportion of the variation of the data about the mean which is explained by the fit. It is given by the ratio of SSR, the sum of squares due to the fit or regression, to SSM, the sum of squares of the original data corrected for mean, that is by

$$R^2 = \mathrm{SSR}/\mathrm{SSM} \qquad \ldots\ 4.4$$

where, for our case here, SSR and SSM are given by[@]:

@ In the case of surface fitting, $f_i^{(1)}$ is used to represent $\psi_i^{(1)}$ the evaluation of the polynomial series at the point $\underline{x}_i$.

TABLE 4.2

surface fit, k = 2.

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	148	74		49			37				
FIT	R^2% (group average)	79.5%	79.7%		81.7%			80.3%				@
FIT	MSR (group average)	225.	113.		83.8			58.3				@
FIT	RMS error (arcsec.)	21.8	21.5	21.9	20.8	17.7	21.3	19.3	19.0	23.0	23.8	
FIT	worst case (arcsec.)	98.6	72.3	95.4	48.3	47.0	70.3	49.5	48.3	73.4	89.4	
FIT	RMS (group average)		21.7		19.9			21.3				@
TEST	RMS error		21.8	21.8	22.2	21.9	22.1	22.3	22.3	21.9	22.1	
TEST	worst case		102.	97.3	103	99.0	100.	108.	105.	97.6	91.6	
COMPARISON	F-ratio σ_1^2/σ_2^2		1.03	.957	1.14	1.71	1.05	1.34	1.37	.795	.752	
COMPARISON	degrees of freedom ν_1		148		198			222				
COMPARISON	degrees of freedom ν_2		142		92			68				
COMPARISON	average probability α		51%		21%			49%				@
	ratio ν_2/p		23.7		15.4			10.3				

@ denotes quantity averaged over group of data subsets of similar n.

TABLE 4.3

surface fit, k = 5

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	148	74		49			37				
FIT	R^2% (group average)	89.2%	89.6%		90.3%			90.1%				@
FIT	MSR (group average)	214.	107.		72.5			51.8				@
FIT	RMS error (arcsec.)	15.8	14.9	16.1	14.7	13.7	15.3	13.7	14.9	15.2	16.4	
FIT	worst case (arcsec.)	61.1	49.2	55.0	44.1	37.8	49.2	42.6	35.6	45.0	48.1	
FIT	RMS (group average)		15.5		14.6			15.5				@
TEST	RMS error		16.0	16.3	17.2	16.7	16.2	18.8	17.5	16.4	16.2	
TEST	worst case		68.3	63.6	75.6	68.1	65.2	85.1	72.0	66.1	57.6	
COMPARISON	F-ratio σ_1^2/σ_2^2		1.20	.957	1.36	1.50	1.05	1.83	1.27	1.01	.817	
COMPARISON	degrees of freedom ν_1		148		198			222				
COMPARISON	degrees of freedom ν_2		136		86			62				
COMPARISON	average probability α		35%		16%			37%				@
	ratio ν_2/p		11.3		7.2			5.2				

@ denotes quantity averages over group of data subsets of similar n.

TABLE 4.4

surface fit, k = 9

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	148	74		49			37				
FIT	R^2% (group average)	[illegible]	92.0%		93.6%			93.7%				@
FIT	MSR (group average)	172.	88.2		59.8			42.5				@
FIT	RMS error (arcsec.)	13.4	12.8	12.7	11.5	11.6	12.4	11.7	11.3	12.7	12.4	
FIT	worst case (arcsec.)	57.6	51.0	43.9	43.9	33.0	47.6	42.1	30.8	44.1	31.5	
FIT	RMS (group average)		12.8		11.8			12.0				@
TEST	RMS error		14.0	14.2	15.4	15.1	14.4	16.9	17.1	14.6	14.3	
TEST	worst case		60.5	62.6	67.3	65.0	60.9	81.1	67.4	60.1	59.0	
COMPARISON	F-ratio σ_1^2/σ_2^2		1.20	1.30	1.76	1.63	1.20	1.79	1.99	1.03	1.05	
COMPARISON	degrees of freedom ν_1		148		198			222				
COMPARISON	ν_2		128		78			54				
COMPARISON	average probability α		10.2%		6.3%			2.2%				@
	ratio ν_2/p		6.4		3.9			2.7				

@ denotes quantity averaged over group of data subsets of similar n.

TABLE 4.5

surface fit, k = 20

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	148	74		49			37				
FIT	R^2% (group average)	94.3%	95.2%		96.2%			96.6%				@
FIT	MSR (group average)	102.	54.3		35.2			25.2				@
FIT	RMS error (arcsec.)	11.2	11.4	9.4	9.2	8.3	9.6	10.0	8.1	9.7	6.9	
FIT	worst case (arcsec.)	58.1	44.6	26.1	33.0	17.4	34.2	37.6	15.7	27.9	14.5	
FIT	RMS (group average)		10.4		9.0			8.7				@
TEST	RMS error		13.4	13.4	15.6	20.8	15.7	17.9	17.2	16.7	15.2	
TEST	worst case		64.0	69.3	69.4	138.	65.9	68.0	74.6	66.7	73.2	
COMPARISON	F-ratio σ_1^2/σ_2^2		1.24	2.23	2.18	5.13	1.98	1.71	2.49	1.56	2.65	
COMPARISON	degrees of freedom ν_1		148		198			222				
COMPARISON	ν_2		106		56			32				
COMPARISON	average probability α		6.0%		0.1%			2.6%				@
	ratio ν_2/p		2.5		1.3			.75				

@ denotes quantity averages over group of data subsets of similar n.

$$SSR = \sum_{i=1}^{n} \sum_{l=1}^{2} \left[f_i^{(l)} - x_{li} - (\overline{y-x}) \right]^2 . \left[w_i^{(l)} \right]^2 , \quad \ldots 4.5$$

and
$$SSM = \sum_{i=1}^{n} \sum_{l=1}^{2} \left[y_i^{(l)} - x_{li} - (\overline{y-x}) \right]^2 . \left[w_i^{(l)} \right]^2 , \quad \ldots 4.6$$

and the difference mean $(\overline{y-x})$ in the above equations by

$$(\overline{y-x}) = \sum_{i=1}^{n} \sum_{l=1}^{2} (y_i^{(l)} - x_{li}).w_i^{(l)} / 2n \ . \quad \ldots 4.7$$

The mean square ratio MSR is given by

$$MSR = \frac{SSR / (p-1)}{\sigma_2^2} \quad \ldots 4.8$$

where $p = 2(k+1)$ is the total number of parameters or coefficients fitted, and can be compared with the Fisher distribution. If

$$MSR \not> F_{1-\alpha}(p-1, \nu_2) \quad \ldots 4.9$$

the fit explains R^2 % of the variation in the data; a greater percentage of explained data would be expected by chance only in $100(1-\alpha)$% of such sets of data. Other criteria of adequacy of fit exist, for example, confidence limits can be placed on the values of the fitted coefficients or on the predicted function value at a desired point of interpolation. Unless we are already convinced of the fit's adequacy and are model fitting for parameters of physical interest, the former are of no great value to us. Prediction confidence intervals would have far greater utility@, but to be reliable, require an estimate of the variance σ_2^2 based on substantial additional data, and both measures are less valid and awkward to compute for the case of non-linear model fitting.

Inspection of Tables 4.2 to 4.5 shows that in general the data sets are all fitted to approximately the same level of R.M.S. error and R^2 for a given order k, and only for k=20 does the Fisher test show substantial inadequacy of fit, in part due to overfitting which has occurred because of an excessively small ratio ν_2/p. Occasionally, spuriously good fits are generated which are not consistent with the full data set XY1, for example the fit of order 9 to XY8 and most of the fits of order 20. For a given number of data points the mean square ratio MSR decreases with increasing order of fit, but in all cases is very large. Wetz (1964) found that for

---

@ Perhaps from the observer's viewpoint it may be even more useful to know the limits of the pointing error rather than the R.M.S. value.

the fit to be useful for predictive purposes MSR should exceed about four times $F_{1-\alpha}(p-1, \nu_2)$ instead of relation 4.9 above, and this is indeed the case for all surface fits shown. Unlike the lack of fit test (relation 4.3) the probability α associated with MSR in relation 4.9 is extremely low and is not given in the Tables. The residual errors of fit in declination $\Delta\delta$, and in hourangle $\Delta H \cos\delta$ were plotted against the two co-ordinates δ and H for the surface fits to the complete data set XY1. For k=9 and 20 these four graphs show a good normal distribution without trends or wedge-ness, but in the case k=5 very slight trends are evident, and in k=2 significant quadratic trends are visible in all the plots other than $\Delta\delta$ v.s. δ. All the cases, however, revealed a number of outlying data points and these are discussed later.

(4.4) MODEL FITTING THE 74-INCH TELESCOPE POINTING DATA

(4.4.1) An Extended Model for the 74-inch

The simple five parameter model used in Chapter 2 omits a number of suspected causes of the pointing errors of the Mt. Stromlo 74-inch. The full 148 data point set XY1 was fitted by this model using the Marquardt algorithm (program MARQDT in Chapter 2), and a system, described later in this chapter, which allows selected parameters to be frozen at their start values and not be involved in the iterative improvement. Table 4.6 shows these results and gives the R.M.S. and worst case errors of fit attained, and the parameter estimates with various combinations of parameters frozen. The initial R.M.S. error of 87.1 arcsecond of the data set is at best improved to approximately 33 arcsecond; far worse is the fact that nearly all of this improvement comes from fitting parameters b_3 and b_5 which are simply the zero errors (or fiduciary offsets) of the hourangle and declination encoders respectively! Thus an improved model is required which will adequately represent: (i) polar misalignment (which includes hourangle zero error), (ii) skewness of the axes, (iii) encoder zero and additional periodic errors, (iv) torsional elasticity of the axes and (v) flexure of the tube and secondary optic supports.

Errors $\Delta\delta$ in declination and ΔH in hourangle due to polar misalignment (i) are given by

$$\Delta\delta = \arcsin z - \delta . \qquad \ldots 4.10$$

$$\text{and} \quad \Delta H = \arctan(-y/x) - H . \qquad \ldots 4.11$$

where δ and H are our independent co-ordinate variables, and x, y and z are obtained from Euler expressions similar to equations 2.33. x, y and z involve the three parameters b_1 the polar misalignment, b_2 the hourangle zero error, and b_3 the orientation of the instrument pole as in Chapter 2. Skewness of the declination axis from the polar axis (ii) is accounted for by

$$\Delta\delta = \arcsin(\sin\delta.\cos b_4) - \delta \qquad \ldots 4.12$$

$$\Delta H = \arctan(\tan\delta.\sin b_4) , \qquad \ldots 4.13$$

where parameter b_4 is the departure from mutual perpendicularity of the axes.

TABLE 4.6

number of parameters operative	errors of fit (arcsec.)		@ parameter estimates (arcsec. except b_3 degrees)				
	R.M.S.	worst case	b_1	b_2	b_3	b_4	b_5
5	33.18	109.2	-8.4	94.3	-13.6	5.0	34.4
2	33.92	114.2		89.1			35.7
1	83.83	207.0	32.2				
1	49.2	121.3		89.1			
1	79.5	203.1					35.7
4	33.3	111.3	-9.9	92.6	.025		36.0

@ Blanks indicate parameter frozen at value of zero.

The encoder errors (iii) comprise the declination offset b_5, the eccentricity of the main encoder gear wheels (declination 1512 teeth, polar 864 teeth), eccentricity of the encoder pinions (32 teeth), and the first harmonic term of the composite tooth error. The latter includes the effect of accumulated pitch errors and the tooth profile error and in practice often appears as a varying but smooth trochoid or sinusoid. Noting that the hourangle zero error has already appeared as b_2, the errors are given by:

$$\Delta\delta = b_5 \qquad \text{zero error}$$

$$+ b_6 \cos\delta + b_7 \sin\delta \qquad \text{main gear eccentricity}$$

$$+ b_8 \cos(1512\delta) + b_9 \sin(1512\delta) \qquad \text{composite tooth error}$$

$$+ b_{10} \cos(\tfrac{1512}{32}\delta) + b_{11} \sin(\tfrac{1512}{32}\delta) \qquad \text{encoder pinion eccentricity} \qquad \ldots 4.14$$

and

$$\Delta H = b_{12} \cos H + b_{13} \sin H \qquad \text{main gear eccentricity}$$

$$+ b_{14} \cos(864H) + b_{15} \sin(864H) \qquad \text{composite tooth error}$$

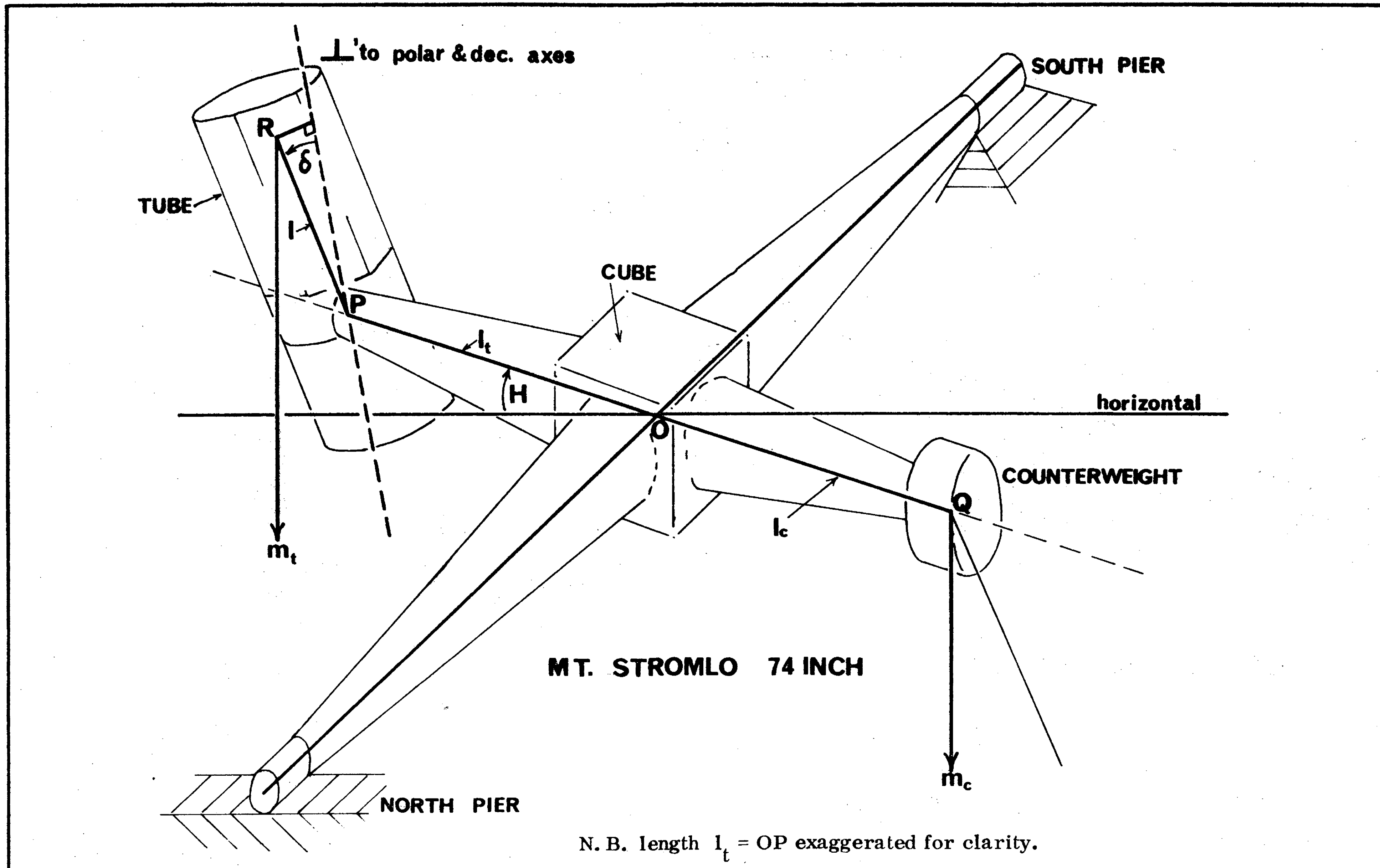

⊥'to polar & dec. axes
SOUTH PIER
R
δ
TUBE
l
CUBE
P
l_t
H
horizontal
O
COUNTERWEIGHT
Q
l_c
m_t
MT. STROMLO 74 INCH
NORTH PIER
m_c
N. B. length l_t = OP exaggerated for clarity.

FIG 4.5

$$+ b_{16} \cos(\tfrac{864}{32}H) + b_{17} \sin(\tfrac{864}{32}H) \quad \text{encoder pinion eccentricity} \qquad \ldots\ 4.15$$

Details of the encoder system mechanics appear in Appendix C, and a discussion of the level of error tolerated by the encoder processing electronics is given in Appendix B.

To completely and accurately analyse the structural flexure of the 74-inch telescope would be extremely difficult and as likely misleading; fortunately a satisfactory approximation can be devised. Since the polar axis is constantly loaded, polar axis flexure appears as polar misalignment above, and needs no further consideration. Assuming a general state of imbalance of the telescope, the torque about the polar axis depends on both declination and hourangle, but the effect is confined to hourangle and expressible on the basis of a (constant) torsional stiffness parameter for the polar axis, and a parameter which represents the telescope imbalance. The declination axis situation is somewhat more complex since there exists (a) a transverse force producing bending of the axis in a vertical plane, (b) a moment due to the imbalanced tube producing bending of the axis in the common plane of the tube and axis, and (c) a similar moment producing twisting of the declination axis, all three being dependent on δ and H as well as the imbalances. We take the view here that due to the extremely short and compact geometry of the declination axis effects (a) and (b) will be of secondary importance; indeed some (and at moderate latitudes most) of the declination axis flexure appears as if it were torsional movement in the polar axis.

Figure 4.5 shows a schematic diagram of the 74-inch English crossed axis mounting with the telescope east of the polar axis, and the forces on the structure idealized to m_t acting vertically through point R the centre of mass of the tube system, and m_c acting through Q the centre of mass of the counterweight and west end of the declination axis. For the effective distances $l = RP$ (which may be negative), $l_t = OP$ and $l_c = OQ$ the errors caused by lack of stiffness of the axes (iv) are given by

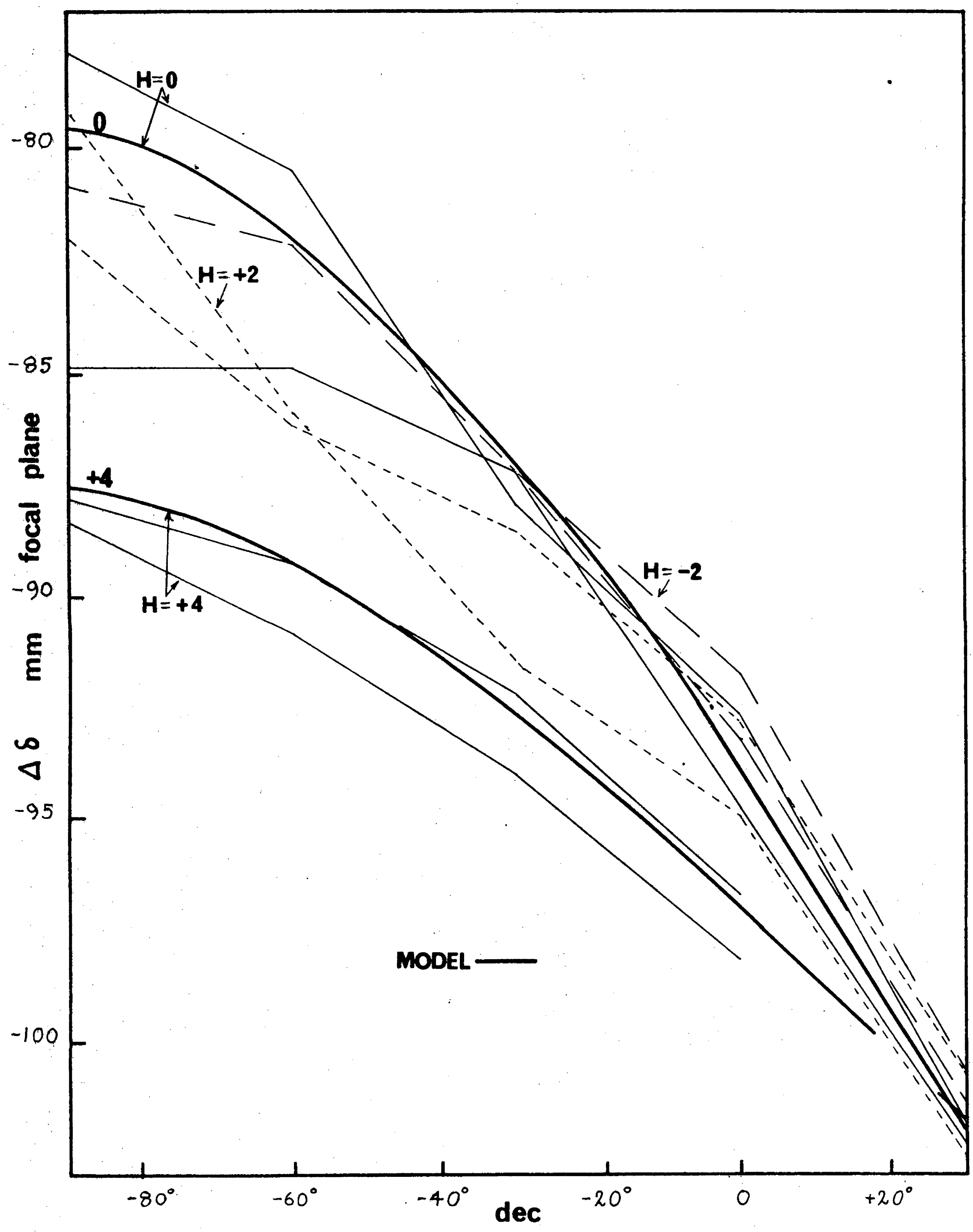
$$\Delta\delta = K_d . m_t . l(\cos\phi.\cos H.\sin\delta - \sin\phi.\cos\delta) \qquad \ldots\ 4.16$$

and $$\Delta H = K_p . \cos\phi\left[(m_c . l_c - m_t . l_t).\cos H + m_t . l.\cos\delta.\sin H\right] \qquad \ldots\ 4.17$$

where K_d and K_p are the declination and polar torsional stiffness constants and ϕ is the latitude (-35.32 degrees at Mt. Stromlo).

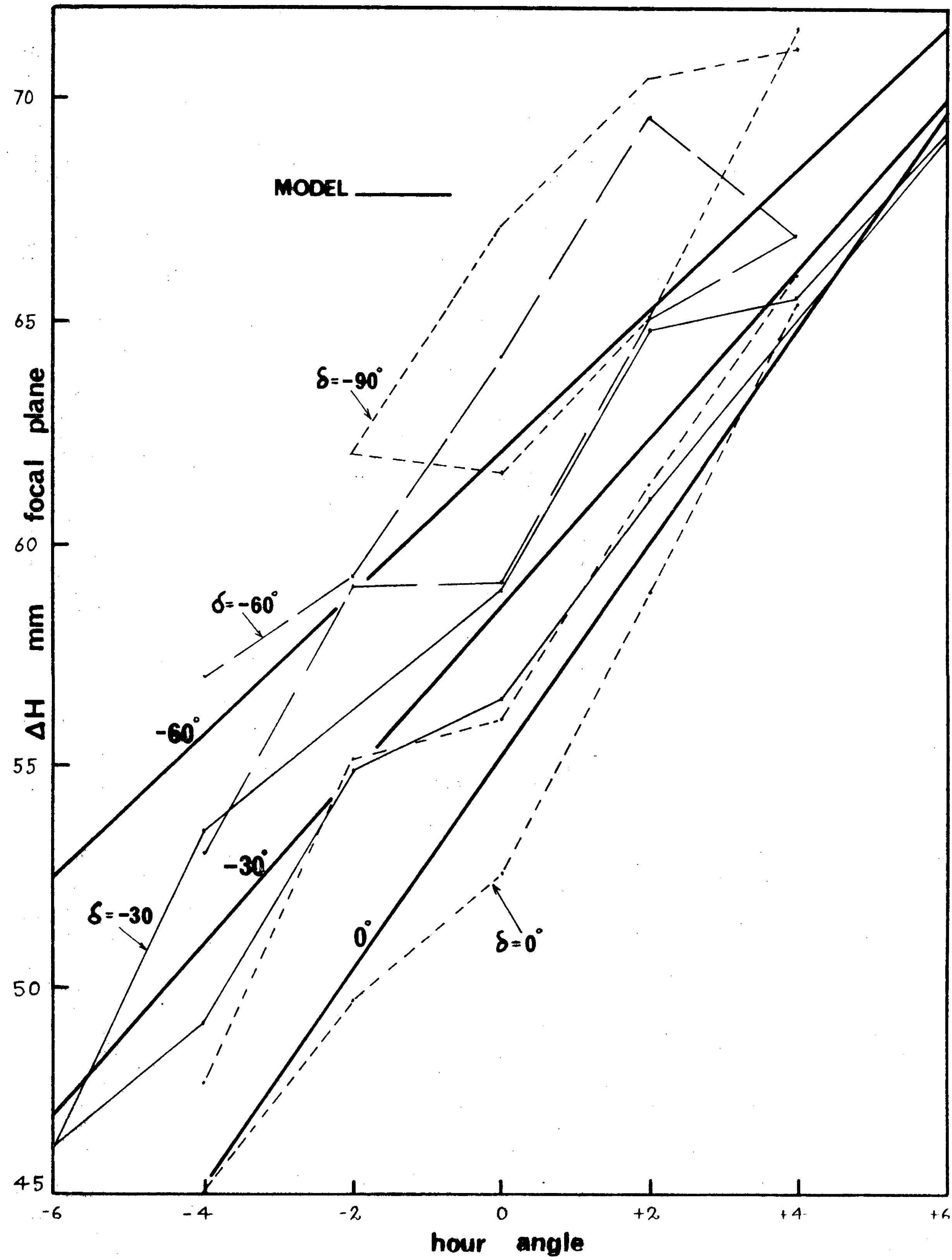
The remaining error cause considered is the non rigidity of the tube and optic supports (v). Again certain simplifying assumptions are necessary, and we assume that the telescope cell and primary point in the theor-

FIG 4.6



H=0
0
H=+2
H=-2
+4
H=+4
MODEL
-80
-85
-90
-95
-100
Δδ mm focal plane
-80°
-60°
-40°
-20°
0
+20°
dec

FIG 4.7



MODEL
δ=-90°
δ=-60°
-60°
-30°
δ=-30
0°
δ=0°
70
65
60
55
50
45
ΔH mm focal plane
-6
-4
-2
0
+2
+4
+6
hour angle

etical direction of the tube axis obtained by applying corrections (i) to (iv); this is justifiable because of the extremely stiff design of the cell. Unfortunately both the complicated truss type tube, and the 'spider' secondary support are statically indeterminate and excessively difficult to analyse, and we employ here a suitable empirical method for obtaining the corrections. An alignment laser was mounted rigidly in the cell and pointed so that the position of the reflected beam from the secondary can be measured in the Cassegrain focal plane using the offset guider optics. The position of the laser spot was noted for the succession of declinations -90, -60, -30, 0, +30 degrees and back again for constant hourangle settings of -6, -4, -2, 0, +2, +4 and +6 hours, with the telescope east of the polar axis, to enable the tube and secondary movement and the mechanical hysteresis to be assessed[@]. Although the laser optical configuration does not behave exactly as a star on the principle axis of the primary mirror, the empirical law obtained is very similar.

Figures 4.6 and 4.7 show the plots of $\Delta\delta$ v.s. δ and ΔH v.s. H respectively and demonstrate well the hysteresial nature of the 74-inch pointing errors. By manually loading various parts of the structure with the tube horizontal, the bulk of the hysteresis was traced to movement in the secondary mirror support drum and focussing system. There is no point in automatic fitting of such data but approximate equations may be fitted by eye to the graphs and are indicated by the solid thick curves. The equations used are

$$\Delta\delta = M_d \cos(a_1\delta + a_2)\cos(a_3 H) + C_d \qquad \text{... 4.18}$$

$$\Delta H = M_p\left[a_4 H + a_5 \sin(a_6\delta)H - a_7\delta\right] + C_p\ , \qquad \text{... 4.19}$$

where the constants $a_1 = 9/14$, $a_2 = 70$ degrees, $a_3 = .85$, $a_4 = 14.2/15\pi$, $a_5 = 4.8/15\pi$, $a_6 = 1.28$ and $a_7 = 10/15\pi$. The vertical scale units in Figures 4.6 and 4.7 are millimetres measured in the (Cassegrain) focal plane but the parameters M_d, M_p are fitted in the model estimation process and not constrained to the values used to produce the thick lines in the figures.

Owing to the smallness of pointing error corrections and to the orthogonality of the co-ordinate variables δ and H it is reasonable to simply superpose or add together the expressions for $\Delta\delta$ and ΔH given in equations 4.10 to 4.19 to form the model equation. This is common practice

---

@ Acknowledgement is due to Dr A.W. Rodgers and John Hart of Mt. Stromlo Observatory for the results of this experiment.

CORRELATION COEFFICIENT MATRIX A*

1.000	-.045	-.081	.062	.891	.871	-.681	.042	.042	-.017	-.017	-.035	-.282	-.049	-.049	.062	.062	-.233	-.238	.378	.860
-.045	1.000	-.387	-.639	.000	.000	.000	.000	.000	.000	.000	.978	.097	.026	.026	-.002	-.002	.114	.494	.000	.000
-.081	-.387	1.000	.607	-.187	-.193	.117	-.012	-.012	-.038	-.038	-.378	.052	.056	.056	.030	.030	.038	-.300	-.148	-.143
.062	-.639	.607	1.000	-.000	-.000	.001	.000	.000	.000	.000	-.579	-.015	.079	.079	.046	.066	-.022	-.572	.000	-.000
.891	.000	-.187	-.000	1.000	.966	-.803	.017	.017	-.042	-.042	.000	.000	.000	.000	.000	.000	.000	.000	.430	.942
.871	.000	-.193	-.000	.966	1.000	-.639	.049	.049	-.037	-.037	.000	.000	.000	.000	.000	.000	.000	.000	.628	.844
-.681	.000	.117	.001	-.803	-.639	1.000	.071	.071	.070	.070	.000	.000	.000	.000	.000	.000	.000	.000	.163	-.926
.042	.000	-.012	.000	.017	.049	.071	1.000	1.000	.045	.045	.000	.000	.000	.000	.000	.000	.000	.000	.104	-.007
.042	.000	-.012	.000	.017	.049	.071	1.000	1.000	.045	.045	.000	.000	.000	.000	.000	.000	.000	.000	.104	-.007
-.017	.000	-.038	.000	-.042	-.037	.070	.045	.045	1.000	1.000	.000	.000	.000	.000	.000	.000	.000	.000	.005	-.042
-.017	.000	-.038	.000	-.042	-.037	.070	.045	.045	1.000	1.000	.000	.000	.000	.000	.000	.000	.000	.000	.005	-.042
-.035	.978	-.378	-.579	.000	.000	.000	.000	.000	.000	.000	1.000	.057	.034	.034	.013	.013	.071	.420	.000	.000
-.282	.097	.052	-.015	.000	.000	.000	.000	.000	.000	.000	.057	1.000	.237	.237	-.102	-.102	.984	.808	.000	.000
-.049	.026	.056	.079	.000	.000	.000	.000	.000	.000	.000	.034	.237	1.000	1.000	-.082	-.082	.237	.145	.000	.000
-.049	.026	.056	.079	.000	.000	.000	.000	.000	.000	.000	.034	.237	1.000	1.000	-.082	-.082	.237	.145	.000	.000
.062	-.002	.030	.066	.000	.000	.000	.000	.000	.000	.000	.013	-.102	-.082	-.082	1.000	1.000	-.082	-.120	.000	.000
.062	-.002	.030	.066	.000	.000	.000	.000	.000	.000	.000	.013	-.102	-.082	-.082	1.000	1.000	-.082	-.120	.000	.000
-.233	.114	.038	-.022	.000	.000	.000	.000	.000	.000	.000	.071	.984	.237	.237	-.082	-.082	1.000	.815	.000	.000
-.238	.494	-.300	-.572	.000	.000	.000	.000	.000	.000	.000	.420	.808	.145	.145	-.120	-.120	.815	1.000	.000	.000
.378	.000	-.148	.000	.430	.628	.163	.104	.104	.005	.005	.000	.000	.000	.000	.000	.000	.000	.000	1.000	.117
.860	.000	-.143	-.000	.942	.844	-.926	-.007	-.007	-.042	-.042	.000	.000	.000	.000	.000	.000	.000	.000	.117	1.000

FIG 4.8

when the error caused by so doing is substantially less than the expected error of the fit, and has been done in the simple model used by Meeks, Ball and Hull (1968) in their calibration of the M.I.T. 'Haystack' antenna. Such a model is more prone to the accidental inclusion of redundant parameters than if the corrections were applied sequentially. For example the inclusion of both b_5 (equation 4.14) and C_d (equation 4.18) in our model function for $\Delta\delta$ would result in poor conditioning of the problem and retard convergence; thus neither C_d nor C_p appear in the model. Various convergence and stability problems with an initial model of this type were traced to the first term in equation 4.17 which, because $\cos\phi$ is constant, is redundantly linked with the term $b_{12} \cos H$ in equation 4.15.

It is necessary to distinguish cases like the above where complete redundancy exists, from others where a given term, for example $b_6 \cos\delta$, is redundantly linked with only part of another, in this case part of equation 4.16 which appears in the model as $b_{20} (\cos\phi \cos H \sin\delta - \sin\phi \cos\delta)$. Omission of one or other of the terms in b_6 or b_{20} may be deleterious to the model's ability to describe the data, and at worst results only in the trading of values between the coefficient b_6 and b_{20}. It is customary in model estimation to inspect the matrix A^* of equation 2.22, which because of the choice of scaling represents the matrix of coefficients of correlation between the parameters. Figure 4.8 shows this matrix for a fit by the model eventually arrived at (see Table 4.7) to the data set XY1. The matrix does not vary greatly from iteration to iteration[@] and the values shown occurred in the final iteration just before convergence. Parameter redundancy is indicated by large off-diagonal elements and whilst some redundancy still exists in the model it is in no way damaging; for example the periodic terms like $b_8 \cos(1512\cdot\delta)$ and $b_9 \sin(1512\cdot\delta)$ give rise to a near unity correlation coefficient but are both necessary if a sinusoidal error of unknown amplitude and phase is to be fitted.

The choice of scale can also be important; initial values of a_4, a_5 and a_7 in equation 4.19 which (although proportional to the values given) were extremely small, caused the parameter space[@@] of the problem to be excessively cramped in the corresponding dimension, resulting in slow convergence. It is not always possible to scale the model completely evenly

---

@ because the model is not too nonlinear.

@@ the k-dimension space in which the parameter vector $\underline{b} = (b_1, \ldots b_k)$ defines a point.

since essential 'orientation' parameters such as b_3 (as opposed to 'error magnitude' parameters like b_1, b_2, b_4 etc.) cannot be eliminated. However a test of a version of the model employing terms of the form $a.\sin(A\delta + b)$ in lieu of form $a'\cos(A\delta) + b'\sin(A\delta)$ in equations 4.14 and 4.15 convinced the author that it is best to eliminate all the nonessential ones. The above reasoning on design of the model led to the 21 parameter model given in full in Table 4.7 and whose derivations with respect to the parameters appear in Table 4.8.

To test the validity of adding the $\Delta\delta$ and ΔH to produce an appropriate model, an 'exact' version was produced by sequentially applying the five corrections in the order: encoder errors, torsional movement of the axes, skewness of the axes, polar misalignment and finally tube and secondary flexure. If we temporarily let f represent either $f^{(1)} = \delta_c$ or $f^{(2)} = H_c$ (the components of the model function), and f_0 the value from the telescope readout, then $f_1 = f_1(f_0)$ gives (i) the encoder corrections and the correction (u) is given by

$$f_u = f_u\,(f_{u-1}) \qquad \ldots\ 4.20$$

It would be onerous to attempt to derive full analytic expressions of the derivatives for the exact model, but it is easy to compute them iteratively by considering the total derivative of equation 4.20 with respect to parameter b_j :

$$\frac{df_u}{db_j} = \frac{\partial f_u}{\partial \delta_{u-1}} \cdot \frac{d\delta_{u-1}}{db_j} + \frac{\partial f_u}{\partial H_{u-1}} \cdot \frac{dH_{u-1}}{db_j} + \frac{\partial f_u}{\partial b_j} \qquad \ldots\ 4.21$$

At each stage df_u/db_j is computed and replaces the stored value df_{u-1}/db_j, the necessary 2k+4 partial derivatives being computed from analytic expressions.

The exact and approximate models were fitted to synthetic data of approximately 500 arcsecond R.M.S. error and their performance compared. Only in the first 2 to 4 iterations were differences of more than a few percent noticed in the sumsquare ϕ or parameter values, and a nearly identical path was taken to the solution, nearing which ϕ differed by approximately 1% and the parameter values by no more than 0.01%. When fitted to telescope data set XY1, the results were similar, with the converged value of ϕ different by less than 0.02% and the parameters by at most 0.5%. It is therefore safe to conclude that the difficulty in calculating the various extra analytic expressions needed for the derivatives of the exact model and the increased computing time overhead is not justified by the quite negligible improvement in accuracy; in the remainder

TABLE 4.7

74-inch model function

$$f^{(1)} = \delta_c = \arcsin z - \delta + \arcsin(\sin\delta\cos b_4) - \delta + b_5 + b_6\cdot\cos\delta + b_7\cdot\sin\delta + b_8\cdot\cos(1512\,\delta)$$

$$+ b_9 \sin(1512\,\delta) + b_{10} \cos(\tfrac{1512}{32}\delta) + b_{11} \sin(\tfrac{1512}{32}\delta) + b_{20}(\cos\phi\cos H\sin\delta - \sin\phi\cos\delta)$$

$$+ b_{21} \cos(a_1\delta + a_2)\cos(a_3 H).$$

$$f^{(2)} = H_c = \arctan(-y/x) - H + \arctan(\tan\delta\sin b_4) + b_{12}\cos H + b_{13}\sin H + b_{14}\cos(864H)$$

$$+ b_{15}\sin(864H) + b_{16}\cos(\tfrac{864}{32}H) + b_{17}\cos(\tfrac{864}{32}H) + b_{18}\cos\phi\cos\delta\sin H$$

$$+ b_{19}\left[a_4 H + a_5\sin(a_6\delta)H - a_7\delta\right].$$

TABLE 4.8

74-inch model function derivatives

<table>
<tr><td colspan="2">Define $E = (1-z^2)^{-\frac{1}{2}}$, then:
$\partial f^{(1)}/\partial b_1 = E\cos\delta \sin(b_2+H) \cos b_1 - E\sin\delta \sin b_1$
$\partial f^{(1)}/\partial b_2 = 0$, $\partial f^{(1)}/\partial b_3 = E\cos\delta . \sin b_1 \cdot \cos(b_2+H)$
Define $D^x_j = \partial x/\partial b_j$ and $D^y_j = \partial y/\partial b_j$, then:
$\partial f^{(2)}/\partial b_j = (yD^x_j - xD^y_j)/(x^2 + y^2)$ for j = 1, 2, 3; and:
D^x_1 is given by equation 2.43a, D^y_1 by equation 2.43b,
D^x_2 is given by equation 2.43c, D^y_2 by equation 2.43d,
D^x_3 is given by equation 2.43e, D^y_3 by equation 2.43f.</td></tr>
<tr><td colspan="2">$\partial f^{(1)}/\partial b_4 = -\left[1 - \sin^2\delta \cos^2 b_4\right]^{-\frac{1}{2}} . \sin\delta \sin b_4$
$\partial f^{(2)}/\partial b_4 = \tan\delta \cos b_4 / \left[1 + \tan^2\delta \sin^2 b_4\right]$</td></tr>
<tr><td>$\partial f^{(1)}/\partial b_5 = 1$
$\partial f^{(1)}/\partial b_6 = \cos\delta$, $\partial f^{(1)}/\partial b_7 = \sin\delta$,
$\partial f^{(1)}/\partial b_8 = \cos(1512\delta)$,
$\partial f^{(1)}/\partial b_9 = \sin(1512\delta)$,
$\partial f^{(1)}/\partial b_{10} = \cos(\frac{1512}{32}\delta)$,
$\partial f^{(1)}/\partial b_{11} = \sin(\frac{1512}{32}\delta)$,
$\partial f^{(1)}/\partial b_j = 0$ for j = 12, 13,...17.</td><td>$\partial f^{(2)}/\partial b_j = 0$ for j = 5, 6,...11.
$\partial f^{(2)}/\partial b_{12} = \cos H$, $\partial f^{(2)}/\partial b_{13} = \sin H$,
$\partial f^{(2)}/\partial b_{14} = \cos(864H)$,
$\partial f^{(2)}/\partial b_{15} = \sin(864H)$,
$\partial f^{(2)}/\partial b_{16} = \cos(\frac{864}{32}H)$,
$\partial f^{(2)}/\partial b_{17} = \sin(\frac{864}{32}H)$,</td></tr>
<tr><td>$\partial f^{(1)}/\partial b_{18} = \partial f^{(1)}/\partial b_{19} = 0$,</td><td>$\partial f^{(2)}/\partial b_{18} = \cos\phi \cos\delta \sin H$
$\partial f^{(2)}/\partial b_{19} = a_4 H + a_5 \sin(a_6\delta)H - a_7\delta$</td></tr>
<tr><td>$\partial f^{(1)}/\partial b_{20} = \cos\phi \cos H \sin\delta - \sin\phi\cos\delta$
$\partial f^{(1)}/\partial b_{21} = \cos(a_1\delta + a_2) \cdot \cos(a_3 H)$.</td><td>$\partial f^{(2)}/\partial b_{20} = \partial f^{(2)}/\partial b_{21} = 0$.</td></tr>
</table>

FIG 4.9

FLOWCHART FOR ROUTINE CMPRES.

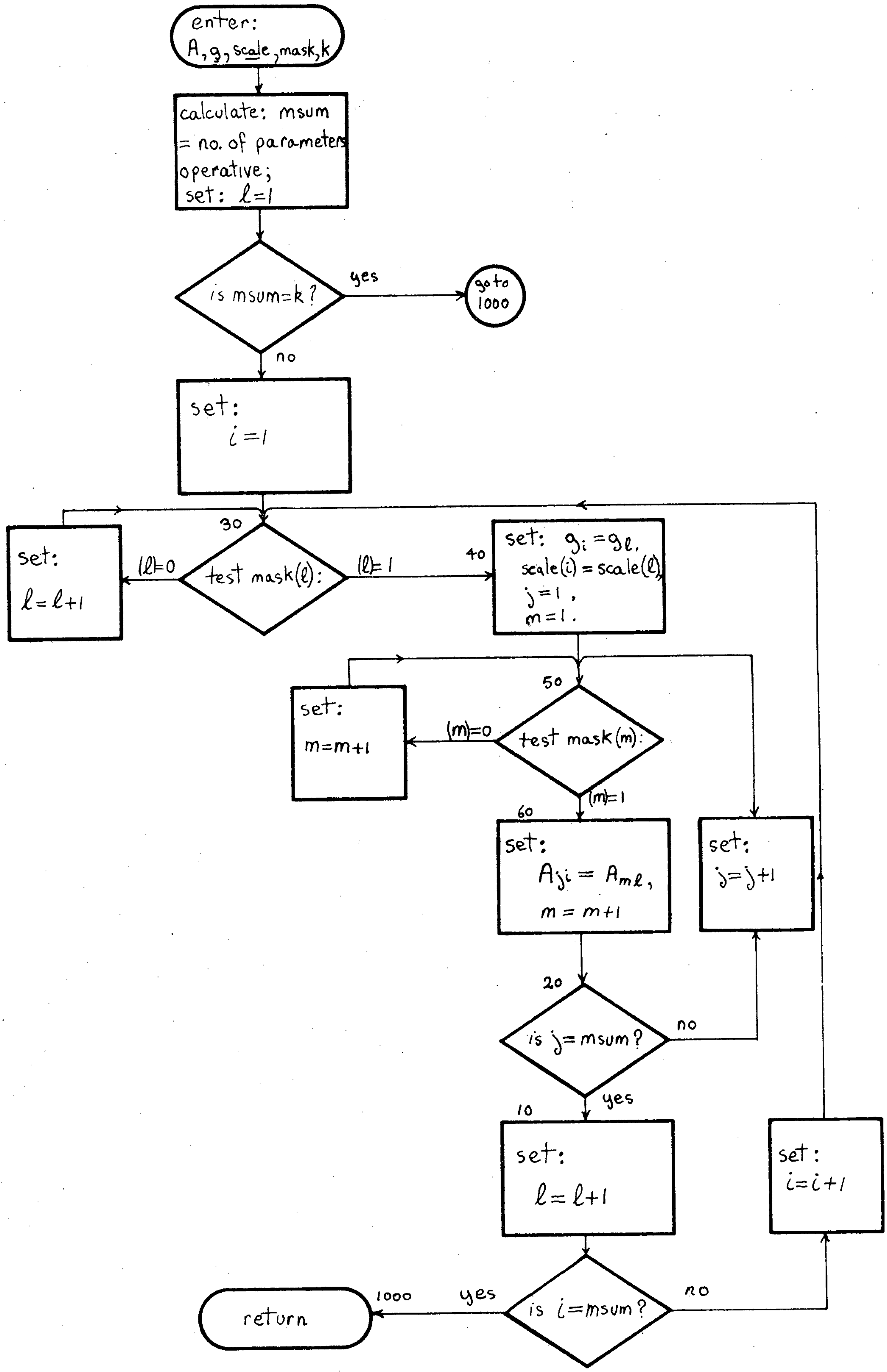
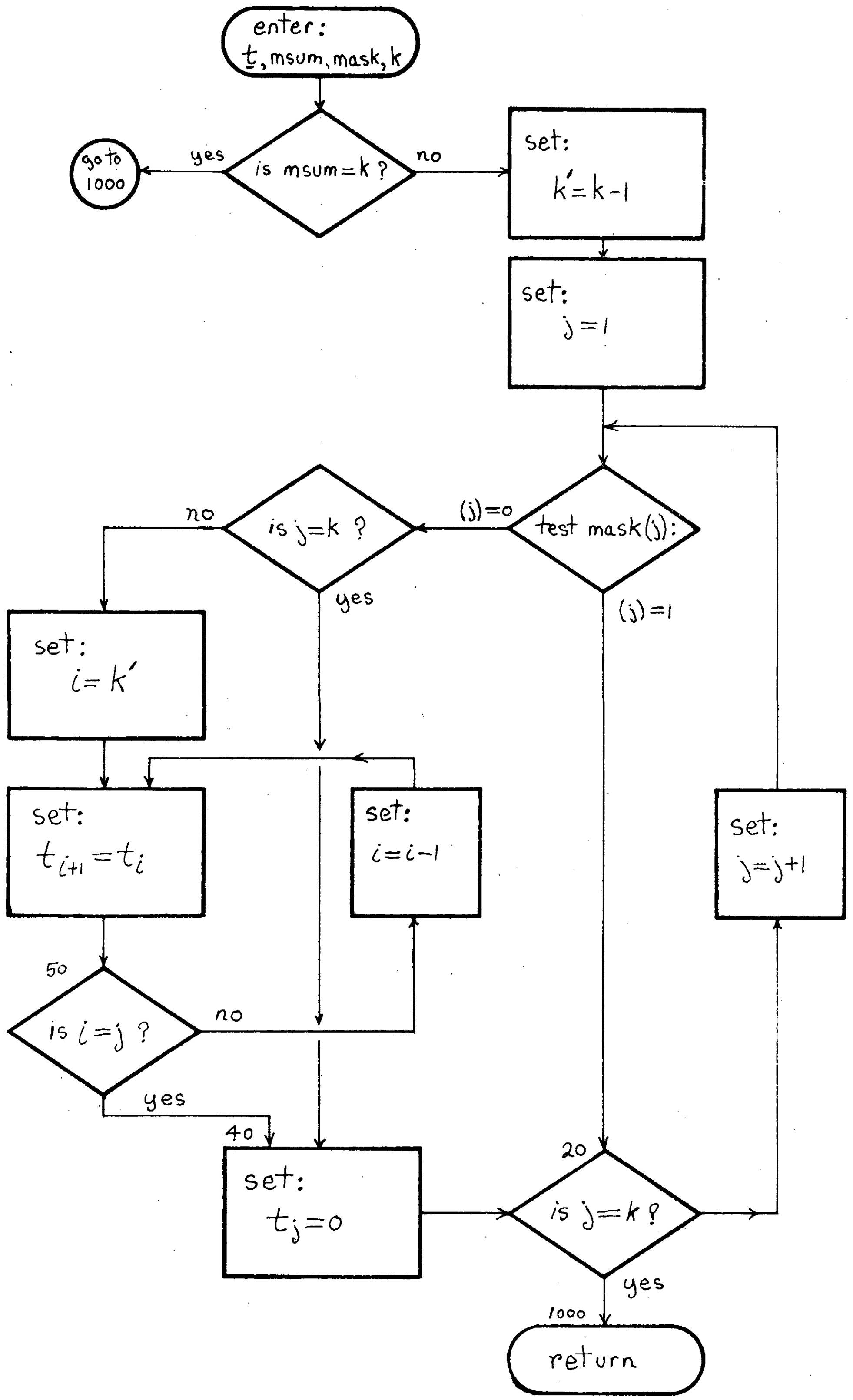


FIG 4.10

FLOWCHART FO ROUTINE EXPAND.



of this chapter the approximate model alone is used.

(4.4.2) Parameter 'Freezing'

In the applications below of the model to telescope data the Marquardt algorithm (MARQDT) has been used. It is efficient, stable and uncritical of the initial parameter values and in no case observed did it prove troublesome, except where the models used suffered extensively from poor scaling or redundant parameters. To assess which parameters of the model are important and which ones fail to contribute to the reduction of ϕ or R.M.S. error, a scheme is necessary which allows selected parameters to be 'frozen' at their start values and only the remaining parameters to be iteratively improved. Techniques somewhat akin to this are used in multiple regression work, see for example Chapter 6 of Draper and Smith (1966), but rely heavily on automatic selection of the parameters to be included in the model,[@] and suffer the disadvantage that parameters can only be included or excluded and not fixed at a value which may be desirable for physical reasons. They are also less useful for non linear models.

The method used here is to include with the initial parameter k-vector on input to the routine, another 'masking' k-vector which contains ones or zeros depending on whether a parameter is operative, or frozen, respectively. Immediately after matrix A^* and vector $\underline{g}^*$ are computed (see MARQDT flowchart Figure 2.7) they are 'compressed' by eliminating the rows and columns associated with the frozen parameters using a routine called CMPRES shown in the flowchart in Figure 4.9. Subsequently, whenever an updated parameter vector $\underline{b}$ is required, the correction vector $\underline{t}$, which is added to the previous $\underline{b}$, is 'expanded' using routine EXPAND (flowcharted in Figure 4.10); this restores the elements of $\underline{t}$ to their appropriate positions, filling the elements corresponding to the frozen parameters with zeros. Within the matrix equation solution section of the Marquardt algorithm the method merely operates with a reduced dimensionality and so the geometrical basis for its strategy is preserved.

(4.4.3) Fitting the Extended Model to Telescope Data

The full data set XY1 was fitted by the model using as the initial parameters $b_j = 10^{-8}$ radian for all $j \neq 3$, and $b_3 = 1.0$ radian, using a number of different masking vectors. The errors and sumsquare ϕ remaining after the fit, together with the estimates of the parameters produced by the various combinations of operative parameters are seen in Table 4.9.

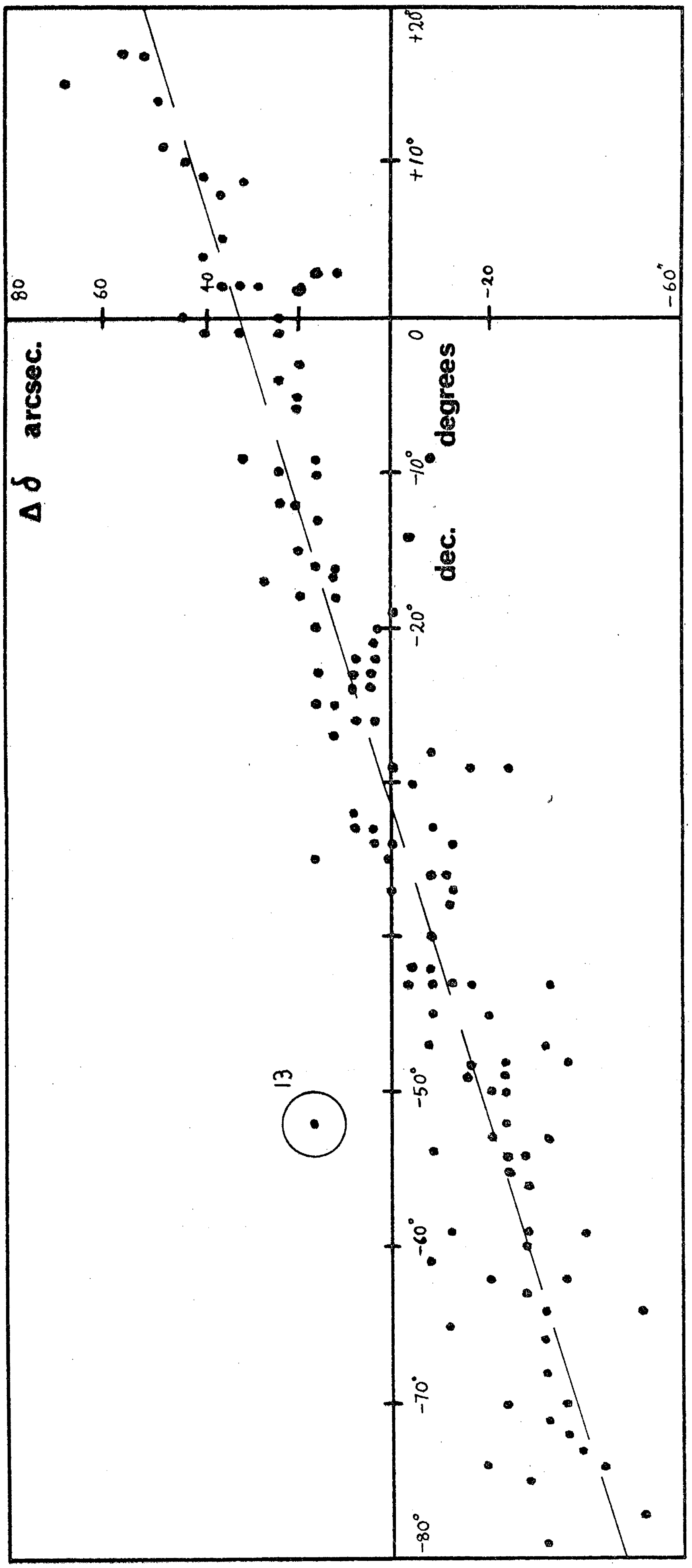
@ whereas physical insight may be more appropriate here.

TABLE 4.9

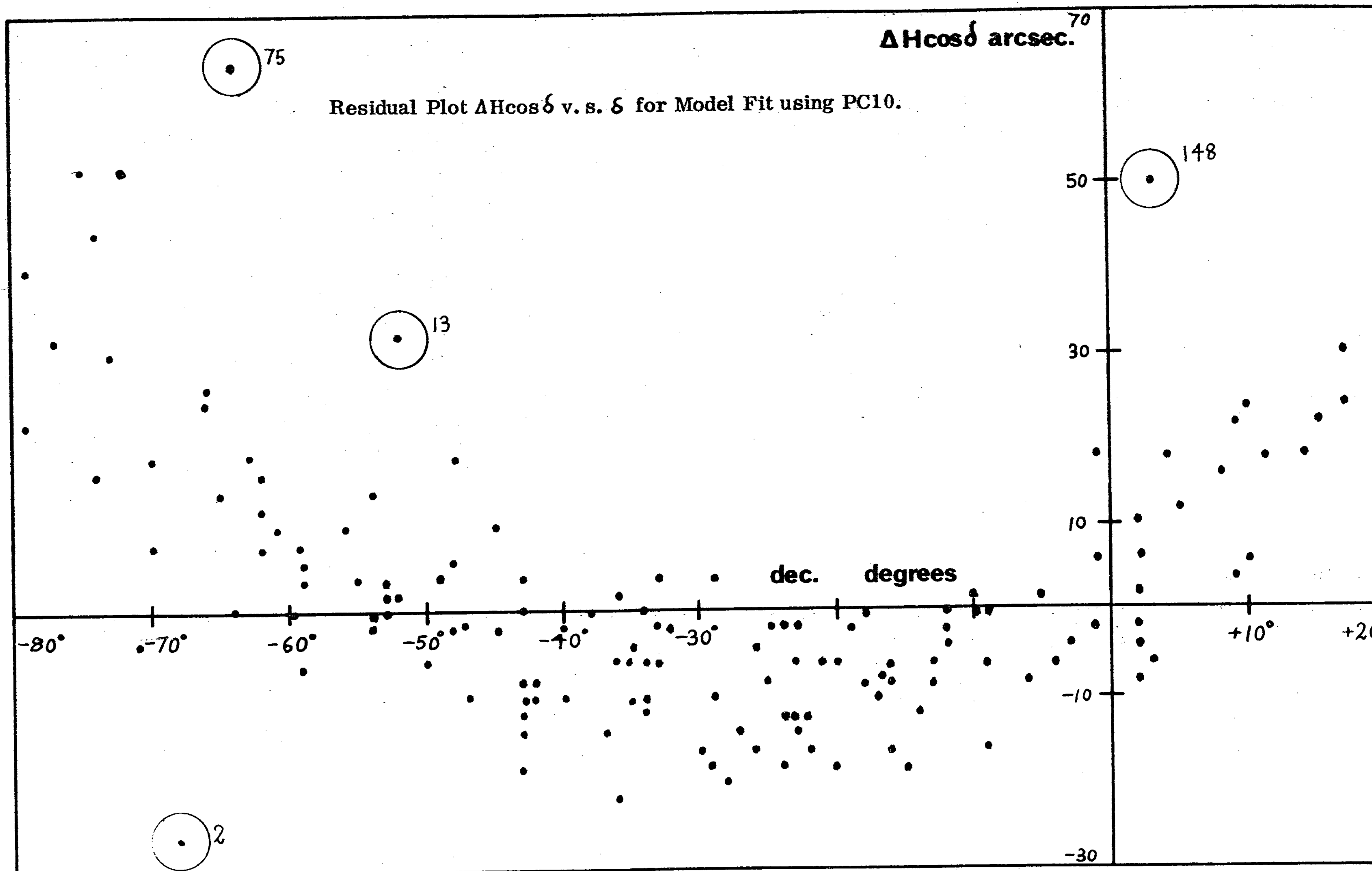
Model fits to XY1 data

parameter combination number	number of operative parameters	iterations	RESULTS OF FIT R.M.S.	worst	R^2%	ESTIMATED PARAMETER VALUE (units=arcseconds except b_3=degrees) Blanks indicate parameter is frozen at zero. 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
PC1	21	7	15.4	72.0	90.1	88	163	94	-50	9.8	18	-12	-.82	-.82	-1.2	-1.2	-70	371	.80	.80	-1.7	-1.7	-233	-486	-1.3	-122
PC2	8	9	20.4	74.7	82.0	9.8	95	130		2.3													76	-76	85	42
PC3	5	7	33.2	109.2	52.3	8.4	94	166	5.0	34																
PC4	7	5	23.8	111.2	75.3	13	94	154	2.1	62	9.8	61														
PC5	11	6	23.9	11.0	75.0	13	94	154	2.1	62	9.9	61	2.8	.28	.63	.63										
PC6	9	9	19.5	62.7	83.6	22	120	113	5.5	36	13	55					-34	36								
PC7	10	9	20.1	73.4	82.3	17	91	126	1.0	37	1.8	32	-.44										50		32	
PC8	17	7	19.7	63.6	83.9	22	120	113	5.6	36	13	55	-.55	-.55	-.32	-.32	-33	35	2.8	2.8	-.51	-.51				
PC9	6	7	25.0	76.2	72.9	31	84	123		70														113		-104
PC10	12	7	17.6	65.4	86.6	83	149	92		9.1	16	-8.5					-59	309					-284	-149	26	-108
PC11	7	12	19.7	74.9	83.9	9.5	95	130															76	-77	87	46
PC13	7	7	19.9	61.3	84.4	25	114	114			44	36					-29	39								
PC14	11	5	19.6	81.1	84.0	9.5	95	128							-1.3	-1.3					-1.3	-1.3	77	-81	87	46
PC15	11	22	18.4	65.7	85.1		143	162		22	-8.7	34					-53	10					83	-158	66	62
PC16	5	8	29.9	73.1	68.7	24	89	109		18													57			
PC17	5	9	31.5	86.6	54.2	21	85	126		22														95		
PC18	5	6	24.1	109.9	73.5	13	93	161		35															67	
PC19	5	7	28.0	109.5	69.1	15	94	152		81																-97

FIG 4.11



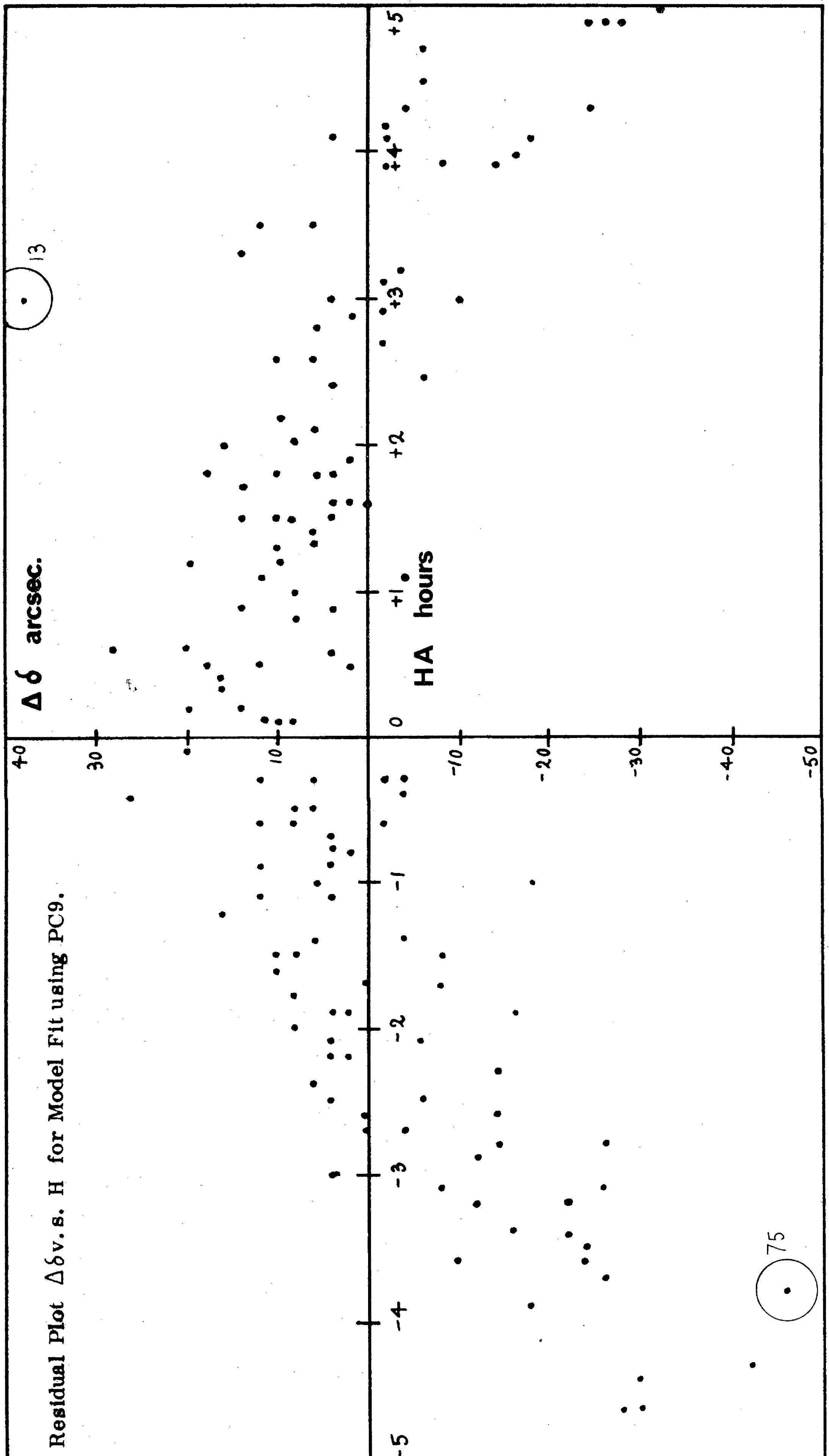
Residual Plot $\Delta\delta$ v.s. δ for Model Fit using PC3.



Residual Plot ΔHcosδ v.s. δ for Model Fit using PC10.

FIG 4.12

FIG 4.13



Residual Plot $\Delta\delta$ v.s. H for Model Fit using PC9.

Satisfactory convergence was obtained in each of the 19 cases, but for some a R.M.S. error of fit is obtained which is considerably poorer than that of the surface fits previously described. The estimated value for a given parameter varies considerably, even for fits with similar R.M.S. error, and this is partly due to the parameter redundancies discussed earlier. It does not detract from the usefulness of the fit for prediction purposes and the exact values of the parameters are possibly of less interest, except perhaps for the parameters governing certain gear errors which are always very small; this will be discussed later.

For the parameter combinations numbered PC 1, 2, 3, 6, 9, 10 and 13, fits were generated to data sets XY2, XY3 to XY10 and tested against the data set XY1 as was done for the surface fitting routine. The results are shown in Tables 4.10 to 4.16 which have the same format as Tables 4.2 to 4.5. The test variance σ_1^2 and its degree of freedom ν_1 has been calculated from equation 4.2, whilst the variance of fit σ_2^2 and ν_2 are given by

$$\left.\begin{aligned} \sigma_2^2 &= \phi/(2n-k') \\ \nu_2 &= 2n - k' \end{aligned}\right\} \quad \ldots\ 4.22$$

where k' is the number of operative parameters. In general the R.M.S. error of fit improves with increased numbers of parameters but the adequacy of the model as measured by the probability α deteriorates.

Line-printer plots of the residuals $\Delta\delta$ and $\Delta H \cos\delta$ against δ and H were produced, and, as in surface fits, the plots of $\Delta\delta$ v.s. δ and $\Delta H \cos\delta$ v.s. H are quite satisfactory with only very slight traces of trends or wedge-ness. The exception is the plot of $\Delta\delta$ v.s. δ for PC3 shown in Figure 4.11 which features a prominent linear trend in δ. $\Delta\delta$ goes through zero at approximately -35 degrees (which is the zenith if H=0), and since the operative parameters of PC3 are just those of the simple five parameter model used in Chapter 2, it is easily seen why the latter was inadequate, and why the surface fitting routine with its inclusion of linear terms in the co-ordinates rather than the predominantly trigonometrical terms of the model, fares better. With regard to the plots of $\Delta H \cos\delta$ v.s. δ and of $\Delta\delta$ v.s. H, only the fit with parameter combination PC1 (all parameters operative) is beyond reproach. A severe quadratic type of trend is present in the $\Delta H \cos\delta$ v.s. δ plot for all the other parameter combinations, and a similar (but inverted) trend is present in the plots of $\Delta\delta$ v.s. H for PC2, PC3 and PC9. These last mentioned defects are typified by those shown in Figures 4.12 and 4.13 respectively. In the three Figures 4.11, 4.12 and 4.13, outlying data

TABLE 4.10

parameter combination no. 1 (all parameters operative)

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	148	74		49			37				
FIT	R^2% (group average)	90.1%	91.2%		93.4%			93.3%				@
FIT	MSR (group average)	121.	62.8		48.1			32.8				@
FIT	RMS error (arcsec.)	15.4	14.4	14.7	13.5	11.7	13.2	12.9	11.5	12.9	15.7	
FIT	worst case (arcsec.)	72.0	45.3	54.1	46.7	24.0	41.3	43.6	20.9	38.8	43.1	
FIT	RMS (group average)		14.6		12.8			13.3				@
TEST	RMS error		16.2	16.4	17.4	17.1	16.8	17.6	17.0	17.7	18.8	
TEST	worst case		90.0	58.9	99.2	60.0	80.7	109.	66.5	74.4	64.2	
COMPARISON	F-ratio σ_1^2/σ_2^2		1.29	1.28	1.58	2.12	1.50	1.55	1.85	1.54	1.13	
COMPARISON	degrees of freedom ν_1		148		198			222				
COMPARISON	degrees of freedom ν_2		127		77			53				
COMPARISON	average probability α		7.5%		1.1%			8.1%				@
	ratio ν_2/p	13.1	6.05		3.67			2.52				

@ denotes quantity averaged over group of data subsets of similar n.

TABLE 4.11

parameter combination no. 2 (8 parameters operative)

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	148	74		49			37				
FIT	R^2% (group average)	82.0%	82.4%		83.6%			83.5%				@
FIT	MSR (group average)	188.	96.4		67.8			49.8				@
FIT	RMS error (arcsec.)	20.4	21.4	18.8	20.2	16.1	20.2	18.3	16.2	22.6	20.6	
FIT	worst case (arcsec.)	74.7	66.1	67.4	50.7	48.2	58.7	43.8	33.6	56.8	61.5	
FIT	RMS (group average)		20.1		18.8			19.4				@
TEST	RMS error		20.6	20.6	21.2	20.7	20.6	22.1	20.7	20.8	20.9	
TEST	worst case		83.0	67.4	84.4	78.6	75.6	95.8	75.2	71.1	61.5	
COMPARISON	F-ratio σ_1^2/σ_2^2		.814	1.33	1.06	1.80	.967	1.44	1.65	.708	.926	
COMPARISON	degrees of freedom ν_1		148		198			222				
COMPARISON	degrees of freedom ν_2		140		90			66				
COMPARISON	average probability α		47%		32%			42%				@
	ratio ν_2/p	36	17.5		11.3			8.24				

@ denotes quantity averages over group of data subsets of similar n.

TABLE 4.12

parameter combination no. 3 (5 parameters operative)

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	148	74		49			37				
FIT	R^2% (group average)	52.3%	52.6%		53.4%			53.2%				@
	MSR (group average)	79.7	39.8		26.7			19.8				@
	RMS error (arcsec.)	33.2	33.4	32.7	31.6	30.8	33.0	30.8	30.2	35.3	34.7	
	worst case (arcsec.)	109.	88.0	105.	60.5	80.8	76.5	60.1	51.8	83.1	102.	
	RMS (group average)		33.0		31.8			32.8				@
TEST	RMS error		33.3	33.3	33.5	33.4	33.3	33.7	33.4	33.4	33.5	
	worst case		113.	105.	115.	109.	110.	116.	109.	110.	102.	
COMPARISON	F-ratio σ_1^2/σ_2^2		.956	1.04	1.12	1.21	.976	1.17	1.22	.803	.845	
	degrees of freedom ν_1		148		198			222				
	ν_2		143		93			69				
	average probability α		50%		33%			52%				@
	ratio ν_2/p	58.2	28.6		18.6			13.8				

@ denotes quantity averaged over group of data subsets of similar n.

TABLE 4.13

parameter combination no. 6 (9 parameters operative)

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	148	74		49			37				
FIT	R^2% (group average)	83.6%	84.0%		84.7%			84.9%				@
	MSR (group average)	183.	94.3		62.7			47.3				@
	RMS error (arcsec.)	19.5	20.5	17.8	18.8	16.3	19.5	18.2	15.9	21.5	18.7	
	worst case (arcsec.)	62.7	62.5	51.4	46.5	48.1	59.2	44.9	36.8	54.9	43.3	
	RMS (group average)		19.1		18.2			18.6				@
TEST	RMS error		19.7	19.7	20.2	20.2	19.8	20.9	20.1	19.9	19.9	
	worst case		70.8	63.5	67.9	69.2	67.6	84.4	66.9	58.3	60.9	
COMPARISON	F-ratio σ_1^2/σ_2^2		.802	1.39	1.11	1.63	.945	1.26	1.58	.703	1.03	
	degrees of freedom ν_1		148		198			222				
	ν_2		139		89			65				
	average probability α		46%		31%			40%				@
	ratio ν_2/p		31.7		15.5			7.22				

@ denotes quantity averages over group of data subsets of similar n.

TABLE 4.14

parameter combination no. 9 (6 parameters operative)

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	148	74		49			37				
FIT	R^2% (group average)	72.9%	73.1%		73.6%			73.9%				@
FIT	MSR (group average)	156.	78.0		51.6			38.9				@
FIT	RMS error (arcsec.)	25.0	25.6	24.2	24.7	22.5	24.6	23.5	22.1	26.6	25.7	
FIT	worst case (arcsec.)	76.2	68.6	71.3	47.5	62.5	62.1	45.2	36.9	62.7	66.3	
FIT	RMS (group average)		24.9		23.7			24.4				@
TEST	RMS error		25.2	25.1	25.5	25.3	25.2	26.0	25.3	25.3	25.3	
TEST	worst case		82.6	71.3	80.8	81.2	79.6	90.0	77.3	75.9	66.3	
COMPARISON	F-ratio σ_1^2/σ_2^2		.901	1.11	1.03	1.31	1.01	1.20	1.31	.802	.880	
COMPARISON	degrees of freedom ν_1		148		198			222				
COMPARISON	degrees of freedom ν_2		142		92			68				
COMPARISON	average probability α		50%		33%			48%				@
	ratio ν_2/p	48.4	23.7		15.4			11.3				

@ denotes quantity averaged over group of data subsets of similar n.

TABLE 4.15

parameter combination no. 10 (12 parameters operating)

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	148	74		49			37				
FIT	R^2% (group average)	86.6%	87.0%		88.5%			88.4%				@
FIT	MSR (group average)	167.	88.0		64.4			45.4				@
FIT	RMS error (arcsec.)	17.6	19.0	15.3	17.3	12.7	17.3	16.8	13.4	18.5	16.1	
FIT	worst case (arcsec.)	65.4	66.9	56.7	43.5	25.3	58.4	44.2	30.9	57.1	51.8	
FIT	RMS (group average)		17.1		13.8			16.2				@
TEST	RMS error		17.9	17.9	18.9	18.9	18.2	20.1	18.6	18.5	18.1	
TEST	worst case		66.9	65.2	74.5	78.2	67.3	85.8	72.3	70.9	65.9	
COMPARISON	F-ratio σ_1^2/σ_2^2		.724	1.59	1.14	2.48	1.02	1.33	1.85	.846	1.12	
COMPARISON	degrees of freedom ν_1		148		198			222				
COMPARISON	degrees of freedom ν_2		136		86			62				
COMPARISON	average probability α		46%		24%			30%				@
	ratio ν_2/p	23.7	11.3		7.17			5.17				

@ denotes quantity averages over group of data subsets of similar n.

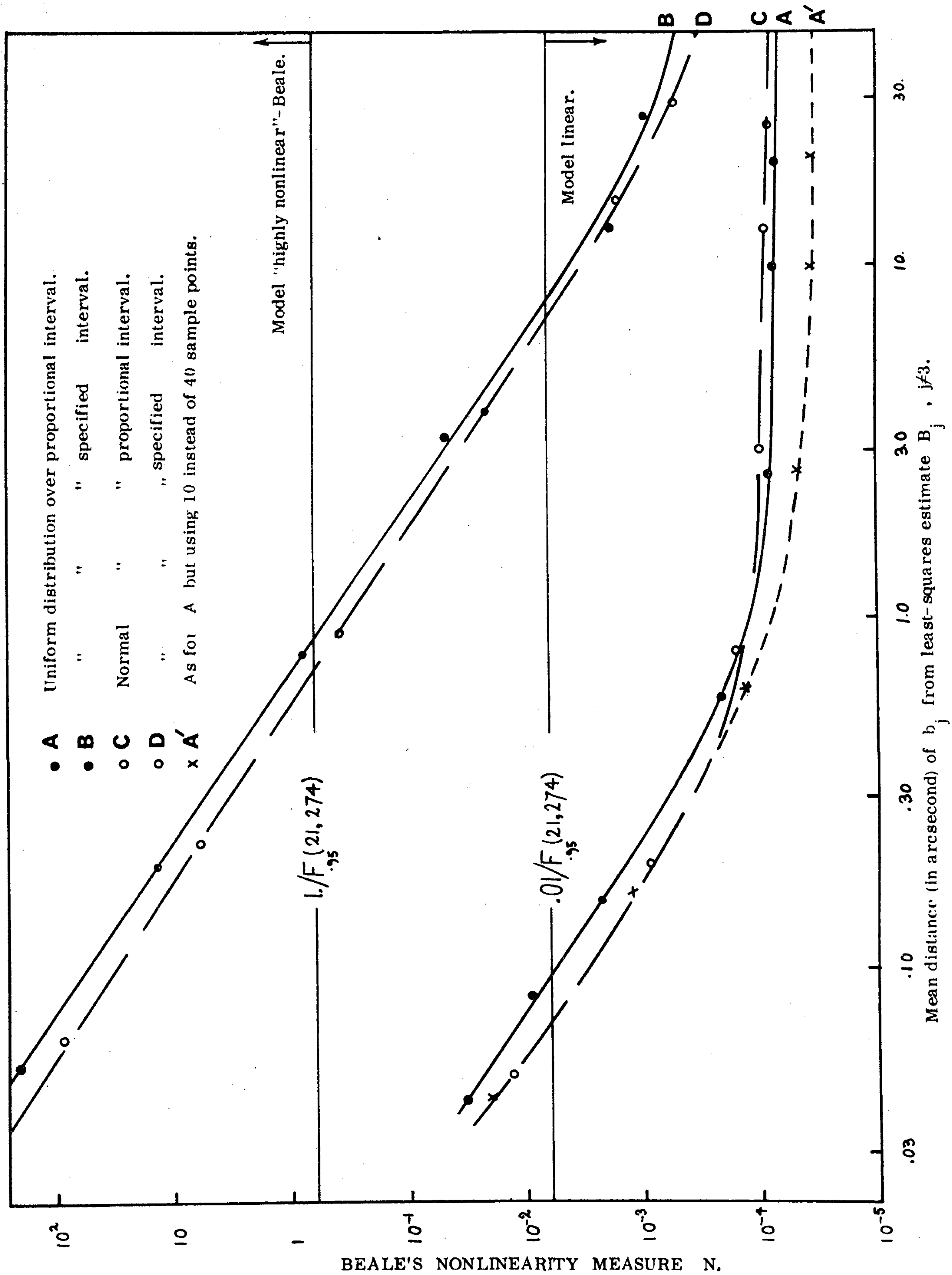
TABLE 4.16

parameter combination no. 13 (7 parameters operative)

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	148	74		49			37				
FIT	R^2% (group average)	84.4%	84.6%		85.4%			85.5%				@
FIT	MSR (group average)	237.	121.		81.3			61.8				@
FIT	RMS error (arcsec.)	19.9	21.0	18.4	19.6	16.7	19.9	18.5	16.7	22.0	19.2	
FIT	worst case (arcsec.)	61.3	58.5	57.6	45.3	53.3	54.6	44.1	31.2	50.2	46.5	
FIT	RMS (group average)		19.7		18.7			20.6				@
TEST	RMS error		20.0	20.0	20.4	20.4	20.2	21.2	20.3	20.3	20.3	
TEST	worst case		66.9	58.1	65.0	76.4	66.8	79.3	71.6	56.6	56.6	
COMPARISON	F-ratio σ_1^2/σ_2^2		.783	1.30	1.04	1.61	.964	1.28	1.49	.729	1.06	
COMPARISON	degrees of freedom ν_1		148		198			222				
COMPARISON	degrees of freedom ν_2		141		91			67				
COMPARISON	average probability α		50%		34%			37%				@
	ratio ν_2/p	41.2	20.2		13.0			9.58				

@ denotes quantity averaged over group of data subsets of similar n.

FIG 4.14



● A Uniform distribution over proportional interval.
● B " " " specified interval.
○ C Normal " proportional interval.
○ D " " , specified interval.
x A′ As for A but using 10 instead of 40 sample points.
Model "highly nonlinear"-Beale.
Model linear.
1./F.95 (21, 274)
.01/F.95 (21, 274)
B
D
C
A
A′
.03
.10
.30
1.0
3.0
10.
30.
Mean distance (in arcsecond) of b_j from least-squares estimate B_j, j≠3.
10^2
10
1
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
BEALE'S NONLINEARITY MEASURE N.

points have been circled and annotated with the observation number.

(4.4.4) The Appropriateness of Linear Statistics

Whilst inspection and comparison of sumsquares and variances is satisfactory, the use of the Fisher test to produce a measure of the model's adequacy is not theoretically valid for the case of a non-linear model; although the model used is intrinsically very nearly linear, an assessment of the appropriateness of applying linear statistics is pertinent. Beale (1960) has developed such a measure of non-linearity based on the departure of the solution locus@ from a plane which is tangent to it at the point on the locus corresponding to $\underline{B} = (B_j)$, the leastsquares estimate of $\underline{b}$. Using m different vectors $\underline{b}_q = (b_{jq})$ q=1,...m, in the neighbourhood of $\underline{B}$, Beale calculates a normalised measure of this departure N which in our case is given by

$$N = \frac{\sum_{q=1}^{m}\sum_{i=1}^{n}\sum_{l=1}^{2}\left[f_i^{(l)}(\underline{b}_q) - f_i^{(l)}(\underline{B}) - \sum_{j=1}^{k}(b_{jq}-B_j)\cdot\left.\frac{\partial f_i^{(l)}}{\partial b_j}\right|_{B_j}\right]^2\cdot\left[w_i^{(l)}\right]^2}{\sum_{q=1}^{m}\left\{\sum_{i=1}^{n}\sum_{l=1}^{2}\left[f_i^{(l)}(\underline{b}_q) - f_i^{(l)}(\underline{B})\right]^2\cdot\left[w_i^{(l)}\right]^2\right\}^2} \quad \ldots 4.23$$

The model is then regarded as adequately linear for linear statistics to be applicable if $N < .01/F_{1-\alpha}(k, \nu)$ and disastrously non-linear if $N > 1/F_{1-\alpha}(k, \nu)$, where ν is the degree of freedom of the variance of fit.

Beale's measure N in equation 4.23 was computed for a number of different distributions and total numbers of sample points using the exact version of the model. As predicted by Beale, the value of N obtained is largely independent of m the number of sample points and is more dependent upon their actual configuration. However Beale also states that N should not vary greatly with the distance of the points $f_i^{(l)}(\underline{b}_q)$ from $f_i^{(l)}(\underline{B})$, the point on the locus corresponding to the leastsquares estimate $\underline{B}$; this was not confirmed by any of the tests carried out. In Figure 4.14, N is plotted against the mean distance of the components of vectors $\underline{b}_q$ from the corresponding component of $\underline{B}$ for several different distributions of the sample points. The case designated A was formed by generating the b_{jq} as a <u>uniform</u> distribution over an interval which is proportional to the magnitude of B_j, case B is a <u>uniform</u> distribution over a specified interval

---

@ Here 'sample space' is the 2n-dimensional space containing the point $(y_1^{(1)}, y_2^{(1)}, \ldots y_n^{(1)}, y_1^{(2)}, \ldots y_n^{(2)})$, where $(y_i^{(1)}, y_i^{(2)}) = (\delta_{0i}, H_{0i})$ is the i^{th} observation point. The 'solution locus' is a k-dimensional surface in this space generated by the points $f_i^{(l)} = f^{(l)}(\underline{x}_i, \underline{b})$, regarding $\underline{b}$ as the variable, and is a hyper-plane if the model is exactly linear in the parameters.

which is the same for each component of the parameter vector[@], case C is a normal distribution of the b_{jq} over a proportional interval, and case D shows a normal distribution on the specified interval. 40 sample points were used and the intervals concerned were varied to obtain the abscissa of Figure 4.14. To show the typical variation of N with number of points used, a fifth case labelled A' has been included and is identical to A except that only 10 of the 40 points were used.

It is clear that N increases monotonically with decreasing size of the distribution of the sample points and is relatively unaffected by m or by the type of distribution. The suprisingly large difference between the cases with proportional and specified intervals is not easy to explain; numerical error propagation was originally considered but is now rejected on the basis of further tests and the quite smooth variation of N. Lines representing Beale's criteria are marked on Figure 4.14, and on the basis of the cases A and C we would not hesitate to deem the model sufficiently linear; however, the scheme's arbitrariness with regard to choice of the sample points necessitates our agreement with Jones (1970) that the scheme is not particularly useful. One further peculiarity is that whilst Beale states that the scheme is valid if the sample points are not too distant from $f_i^{(1)}(\underline{B})$, the results for the model here become less stable the smaller this distance becomes. A more extensive appraisal of Beale's non-linearity measure is given in Guttman and Meeter (1965) where a two parameter model which permits theoretical investigation is considered. Here we conclude that the limited and confusing information obtained from the test is not worth the computing effort involved.

(4.5) REJECTION OF OUTLIERS

In compiling the original 148 point data set XY1, certain observations with which trouble was experienced or whose accuracy was suspect were not used. An assessment of the fits so far described indicates that it might be advantageous to reject a number of others. Although work has been done on schemes which permit automatic rejection of spurious data points or 'outliers', their unfettered use on data is not a wise procedure, since it is rarely clear whether a peculiar data point is spurious or actually representative of an unnoticed trend. Grubbs (1950) gives rejection criteria based on the distance of the largest observation from the mean, and Anscombe (1960) discusses rejection rules involving the size of the largest residual compared to the standard deviation of the errors.

@ except for b_3, of course, which took the same interval in degrees as did the others in arcseconds.

Anscombe, by considering the 'insurance premium' (the increase in residual variance due to unnecessary use of the rejection rule), and the 'protection offered' (the chance that a spurious data point will escape detection) concludes that such rules are somewhat arbitrary when only a single sample of data is available, and highly ineffective since, for practical 'premiums', spurious points can easily escape detection. Irwin (1925) discusses a criterion based on the difference between the largest and next largest observations, and several other schemes are proposed in the literature; the commonly cited viewpoint is, however, that the only safe procedures are graphical inspection of the data and residuals, and, of course, distinguishing outliers purely on the basis of different or inconsistent conditions of observation and without regard to magnitude. Alternatives to rejection of a suspected outlier include data set truncation, where an equal percentage of high and low valued observations are rejected (irrespective of individual magnitudes), and 'Winsoration' (after C. Winsor) where the extreme observations are decreased in magnitude until they are equal to the next most extreme ones. Only rejection has been considered here.

The plots of errors v.s. co-ordinates shown in Figures 4.3a to 4.31 show occasional points lying well off the main stream or trend; these points are numbers 2, 75, 123, 124, 148 and to a lesser extent 13 and 17. Also, inspection of plots of the residuals resulting from surface fitting the data set XY1 reveals that points number 2, 13, 75 and 148 are not typical, and similarly, in the residual plots for the extended model fits with various parameter combinations we can single out points number 2, 13, 75, 148 and possibly 59, 74 and 100. Point 148 is also suspect since it was taken at an extremely large zenith angle (routine CATALOG.OBS checks the zenith angle of the generated grid point, not that of the associated star), at which gross misbehaviour of the telescope is expected and where the accuracy of the refraction correction cannot be guaranteed. The two consecutive points 123 and 124 appear to indicate a period of unusually poor observational accuracy and the points 2, 13, 75, 123, 124 and 148 are consistently the worst case residuals in most of the fits. It was therefore decided to delete these 6 points to form a data set of 142 points. As before, this modified data set is divided into data subsets: 2 of 71 points, 3 of 47 points and 4 of 35 points as detailed in Table 4.17.

Surface fits with k=2, 5, 9 and 20 were generated to the data subsets and the result tested against the full set XY1; these tests are tabulated in Table 4.18 to 4.21. Although the removal of the spurious points improves the R.M.S. error of the data set and subsets by only about 1%, the effect upon the fitting process is quite marked. Comparison of these

Tables with Tables 4.2 to 4.5 shows a 10 - 20% improvement in the R.M.S. error of fit, slightly less variation between subsets and a considerable improvement in worst case error. The adequacy of fit (indicated by α) has improved particularly for k=5 and 9, whilst the case k=20, although improved in this regard, still suffers from overfitting[@]. A small improvement of 2 to 4% in the proportion of variation explained (R^2) is evident, and there is also an increase in MSR. The latter, however, is sufficiently large in each case that the exact value is of no consequence. Residual plots of the surface fits to the modified data set are discussed below.

TABLE 4.17

data set	number of points	R.M.S. error arcsecond	worst case error
XY1	142	86.2	180.2
XY2	71	86.3	180.2
XY3	71	86.0	153.6
XY4	47	84.4	153.6
XY5	47	84.9	149.2
XY6	47	89.2	180.2
XY7	35	87.3	180.2
XY8	35	87.3	153.6
XY9	35	84.9	142.5
XY10	35	84.9	152.9

Similar fits and tests on modified data were run using the model estimation routine MARQDT with operative parameter combinations PC1, 2, 3, 6, 9, 10 and 13 as was done in Tables 4.10 to 4.16, and the results are given in Tables 4.22 to 4.28. In general the nature and extent of improvement is very similar to the surface fitting comparison except that the improvement in the R.M.S. error of fit is even larger (20 to 30 percent) and the adequacy of fit for the smaller data subsets is considerably better.

@ Overfitting results from using an excessively large number of parameters or coefficients for the number of data points and is indicated by very low ratios ν_2/p. In such cases the fitting functions are said to be fitted to the errors.

TABLE 4.18

surface fit, k = 2

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	142	71		47			35				
FIT	R^2% (group average)	84.1%	84.1%		84.3%			84.4%				@
FIT	MSR (group average)	240.	144.		95.3			92.1				@
FIT	RMS error (arcsec.)	18.2	18.0	18.3	16.7	17.8	19.6	18.3	17.9	17.0	18.6	
FIT	worst case (arcsec.)	48.7	48.5	44.6	37.1	48.8	44.2	47.6	45.4	38.6	38.0	
FIT	RMS (group average)		18.1		18.0			17.9				@
TEST	RMS error		18.2	18.4	18.2	18.6	18.4	18.6	18.5	18.5	18.6	
TEST	worst case		48.0	50.1	50.1	51.9	49.1	47.9	50.7	53.3	49.8	
COMPARISON	F-ratio σ_1^2/σ_2^2		.987	.982	1.21	1.06	.763	.964	.991	1.13	.922	
COMPARISON	degrees of freedom ν_1		142		190			214				
COMPARISON	degrees of freedom ν_2		136		88			64				
COMPARISON	average probability α		54%		49%			52%				@
	ratio ν_2/p	46.3	22.6		14.3			10.7				

@ denotes quantity averaged over group of data subsets of similar n.

TABLE 4.19

surface fit, k = 5

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	142	71		47			35				
FIT	R^2% (group average)	92.1%	92.2%		92.4%			92.6%				@
FIT	MSR (group average)	289.	140.		91.8			66.6				@
FIT	RMS error (arcsec.)	12.8	13.1	12.3	11.5	12.5	13.5	12.6	11.4	12.6	12.7	
FIT	worst case (arcsec.)	34.2	32.8	28.6	22.1	30.8	29.4	31.9	24.5	29.9	23.6	
FIT	RMS (group average)		12.7		12.5			12.3				@
TEST	RMS error		12.9	13.1	13.2	14.0	13.1	13.4	13.4	14.6	13.5	
TEST	worst case		32.4	36.2	37.1	36.1	37.1	37.5	31.4	37.8	40.1	
COMPARISON	F-ratio σ_1^2/σ_2^2		.855	1.15	1.30	1.21	.802	.974	1.25	1.20	.986	
COMPARISON	degrees of freedom ν_1		142		190			214				
COMPARISON	degrees of freedom ν_2		130		82			58				
COMPARISON	average probability α		51%		38%			37%				@
	ratio ν_2/p	22.7	10.8		6.7			4.8				

@ denotes quantity averages over group of data subsets of similar n.

TABLE 4.20

surface fit, k = 9

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	142	71		47			35				
FIT	R^2% (group average)	95.3%	95.5%		96.1%			96.2%				@
FIT	MSR (group average)	283.	142.		91.2			68.0				@
FIT	RMS error (arcsec.)	9.8	10.4	8.8	8.5	8.9	10.3	9.6	8.1	9.5	8.1	
FIT	worst case (arcsec.)	24.1	25.6	24.9	18.3	20.3	18.8	20.8	15.5	19.3	17.8	
FIT	RMS (group average)		14.6		9.2			8.8				@
TEST	RMS error		10.1	10.5	10.4	11.8	10.5	10.8	10.8	11.9	11.8	
TEST	worst case		25.3	26.4	27.6	34.8	24.6	31.3	27.1	43.2	32.9	
COMPARISON	F-ratio σ_1^2/σ_2^2		.776	1.62	1.35	1.67	.826	.970	1.43	1.24	1.76	
COMPARISON	degrees of freedom ν_1		142		190			214				
COMPARISON	ν_2		122		74			50				
COMPARISON	average probability α		46%		31%			21%				@
	ratio ν_2/p	13.2	6.1		3.6			2.5				

@ denotes quantity averaged over group of data subsets of similar n.

TABLE 4.21

surface fit, k = 20

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	142	71		47			35				
FIT	R^2% (group average)	96.8%	97.1%		97.8%			98.2%				@
FIT	MSR (group average)	179.	82.2		57.1			39.4				@
FIT	RMS error (arcsec.)	8.1	7.8	7.7	6.7	6.7	6.8	5.9	6.6	6.1	5.4	
FIT	worst case (arcsec.)	17.2	15.8	14.5	16.5	12.6	13.8	9.6	12.2	12.7	12.0	
FIT	RMS (group average)		7.8		6.7			6.0				@
TEST	RMS error		8.9	9.1	9.5	12.4	15.2	11.7	11.7	19.9	19.7	
TEST	worst case		21.9	20.4	26.5	55.8	89.2	37.6	31.0	99.7	140.	
COMPARISON	F-ratio σ_1^2/σ_2^2		1.15	1.26	1.42	2.60	3.91	1.96	1.56	5.57	7.01	
COMPARISON	degrees of freedom ν_1		142		190			214				
COMPARISON	ν_2		100		52			28				
COMPARISON	average probability α		17%		23%			25%				@
	ratio ν_2/p	5.9	2.4		1.2			0.7				

@ denotes quantity averages over group of data subsets of similar n.

TABLE 4.22

parameter combination no. 1 (all parameters operative)

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	142	71		47			35				
FIT	R^2% (group average)	92.7%	93.3%		93.8%			94.3%				@
FIT	MSR (group average)	164.	76.2		49.9			37.0				@
FIT	RMS error (arcsec.)	12.4	12.2	12.6	11.6	11.6	12.5	12.3	11.4	9.4	13.3	
FIT	worst case (arcsec.)	38.1	34.1	26.8	25.0	21.0	22.9	25.1	28.3	19.0	22.4	
FIT	RMS (group average)		12.4		11.9			11.6				@
TEST	RMS error		13.0	12.8	12.7	13.6	13.7	13.6	14.1	14.9	14.5	
TEST	worst case		34.1	40.5	38.7	53.9	52.9	32.8	37.6	55.0	44.0	
COMPARISON	F-ratio σ_1^2/σ_2^2		1.08	.913	1.01	1.23	.996	.904	1.19	2.09	.884	
COMPARISON	degrees of freedom ν_1		142		190			214				
COMPARISON	degrees of freedom ν_2		121		73			49				
COMPARISON	average probability α		51%		39%			42%				@
	ratio ν_2/p	12.5	5.8		3.5			2.3				

@ denotes quantity averaged over group of data subsets of similar n.

TABLE 4.23

parameter combination no. 2 (8 parameters operative)

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	142	71		47			35				
FIT	R^2% (group average)	85.4%	85.5%		85.9%			85.5%				@
FIT	MSR (group average)	230.	112.		75.1			53.8				@
FIT	RMS error (arcsec.)	17.4	17.0	17.8	16.9	16.3	18.1	18.6	16.5	14.9	19.1	
FIT	worst case (arcsec.)	55.1	55.2	41.8	39.5	34.0	46.6	53.0	37.1	38.7	43.8	
FIT	RMS (group average)		17.4		17.1			17.5				@
TEST	RMS error		17.4	17.4	17.5	17.8	18.0	17.6	17.4	17.8	17.6	
TEST	worst case		55.2	55.0	56.6	61.4	55.5	53.0	56.5	57.5	53.5	
COMPARISON	F-ratio σ_1^2/σ_2^2		1.05	.879	1.02	1.19	.907	.761	1.03	1.40	.706	
COMPARISON	degrees of freedom ν_1		142		190			214				
COMPARISON	degrees of freedom ν_2		134		86			62				
COMPARISON	average probability α		58%		45%			60%				@
	ratio ν_2/p	34.5	16.8		10.8			7.8				

@ denotes quantity averages over group of data subsets of similar n.

TABLE 4.24

parameter combination no. 3 (5 parameters operative)

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	142	71		47			35				
FIT	R^2% (group average)	54.8%	54.8%		55.1%			54.3%				@
FIT	MSR (group average)	84.5	41.6		27.4			19.4				@
FIT	RMS error (arcsec.)	30.6	30.2	31.0	29.4	29.7	32.3	30.5	29.8	29.9	32.4	
FIT	worst case (arcsec.)	84.4	83.7	68.7	56.1	55.8	78.2	82.6	54.0	66.1	69.8	
FIT	RMS (group average)		30.6		30.8			30.6				@
TEST	RMS error		30.6	30.6	30.7	30.8	31.0	30.7	30.6	30.8	30.6	
TEST	worst case		83.7	85.1	85.7	88.1	78.2	82.4	83.5	84.4	86.2	
COMPARISON	F-ratio σ_1^2/σ_2^2		1.01	.922	1.07	1.05	.836	.941	.999	1.01	.800	
COMPARISON	degrees of freedom ν_1		142		190			214				
COMPARISON	degrees of freedom ν_2		137		89			65				
COMPARISON	average probability α		58%		54%			63%				@
	ratio ν_2/p	55.8	27.4		17.8			13.0				

@ denotes quantity averaged over group of data subsets of similar n.

TABLE 4.25

parameter combination no. 6 (9 parameters operative)

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	142	71		47			35				
FIT	R^2% (group average)	86.6%	86.6%		87.2%			87.0%				@
FIT	MSR (group average)	222	108.		72.8			52.2				@
FIT	RMS error (arcsec.)	16.7	16.1	17.1	16.3	14.9	17.7	17.0	15.6	14.4	18.4	
FIT	worst case (arcsec.)	52.5	51.5	38.5	38.3	32.9	45.3	48.7	37.3	42.3	38.4	
FIT	RMS (group average)		16.6		16.3			16.3				@
TEST	RMS error		16.7	16.7	16.8	17.0	17.3	17.1	16.8	17.4	16.8	
TEST	worst case		51.5	53.3	55.3	55.1	52.5	52.1	53.8	53.8	52.8	
COMPARISON	F-ratio σ_1^2/σ_2^2		1.07	.845	.994	1.30	.850	.879	1.06	1.40	.678	
COMPARISON	degrees of freedom ν_1		142		190			214				
COMPARISON	degrees of freedom ν_2		133		85			61				
COMPARISON	average probability α		60%		48%			55%				@
	ratio ν_2/p	30.6	14.8		9.5			6.8				

@ denotes quantity averages over group of data subsets of similar n.

TABLE 4.26

parameter combination no. 9 (6 parameters operative)

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	142	71		47			35				
FIT	R^2% (group average)	74.9%	74.9%		75.2%			74.7%				@
FIT	MSR (group average)	166.	81.2		53.3			38.0				@
FIT	RMS error (arcsec.)	22.8	22.6	23.0	22.3	22.0	23.6	23.5	22.1	21.6	24.1	
FIT	worst case (arcsec.)	63.2	63.2	45.2	44.8	43.0	59.4	62.1	41.9	37.3	45.2	
FIT	RMS (group average)		22.8		22.6			22.8				@
TEST	RMS error		22.8	22.8	23.0	23.1	23.1	23.0	22.8	23.1	22.9	
TEST	worst case		63.2	63.2	66.1	64.1	59.4	62.1	62.6	64.4	63.9	
COMPARISON	F-ratio σ_1^2/σ_2^2		.995	.938	1.02	1.07	.876	.860	1.00	1.09	.796	
COMPARISON	degrees of freedom ν_1		142		190			214				
COMPARISON	ν_2		136		88			64				
COMPARISON	average probability α		58%		54%			63%				@
	ratio ν_2/p	46.3	22.7		14.7			10.7				

@ denotes quantity averaged over group of data subsets of similar n.

TABLE 4.27

parameter conbination no. 10 (12 parameters operative)

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	142	71		47			35				
FIT	R^2% (group average)	89.2%	89.4%		90.7%			90.1%				@
FIT	MSR (group average)	204.	100.		73.6			51.6				@
FIT	RMS error (arcsec.)	15.0	14.4	15.2	14.7	13.2	13.7	15.2	13.7	11.3	16.5	
FIT	worst case (arcsec.)	41.9	47.3	37.8	35.0	31.4	26.8	40.1	34.2	24.3	39.5	
FIT	RMS (group average)		14.8		13.9			14.2				@
TEST	RMS error		15.1	15.1	51.2	15.9	17.1	15.5	15.2	18.5	15.1	
TEST	worst case		47.3	47.3	51.8	61.3	77.9	47.3	49.5	88.8	44.3	
COMPARISON	F-ratio σ_1^2/σ_2^2		1.10	.872	.973	1.47	1.60	.873	1.08	2.70	.657	
COMPARISON	degrees of freedom ν_1		142		190			214				
COMPARISON	ν_2		130		82			58				
COMPARISON	average probability α		54%		30%			53%				@
	ratio ν_2/p	22.7	10.8		6.8			4.8				

@ denotes quantity averages over group of data subsets of similar n.

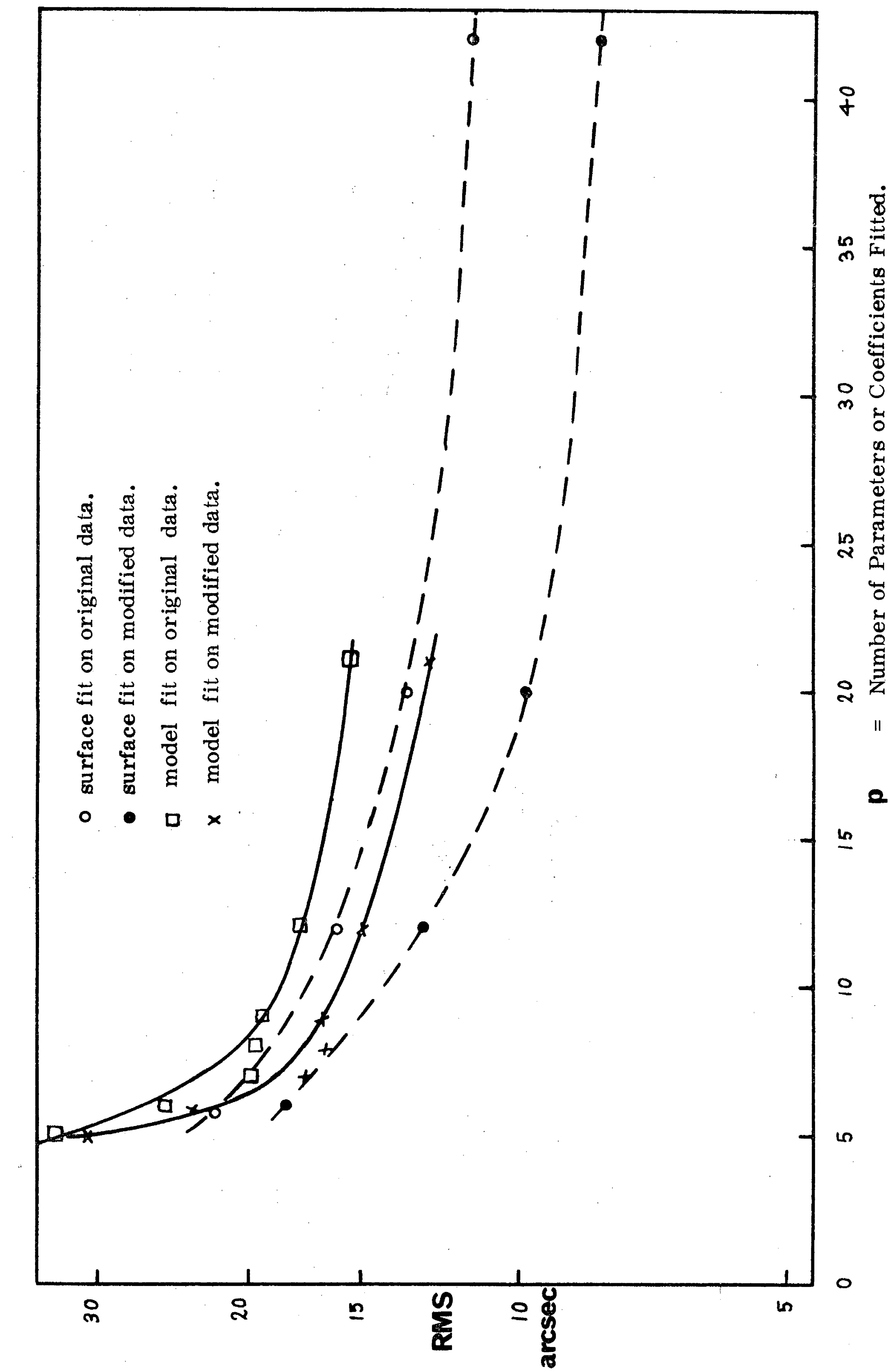
TABLE 4.28

parameter combination no. 13 (7 parameters operative)

	DATA SET	XY1	XY2	XY3	XY4	XY5	XY6	XY7	XY8	XY9	XY10	
	number of points n	142	71		47			35				
FIT	R^2% (group average)	87.4%	87.5%		87.8%			87.5%				@
FIT	MSR (group average)	277.	136.		90.7			66.1				@
FIT	RMS error (arcsec.)	17.4	16.7	17.9	17.1	16.0	18.1	18.5	16.3	14.6	19.4	
FIT	worst case (arcsec.)	56.7	55.3	42.2	39.6	37.2	49.2	53.4	38.2	38.7	42.0	
FIT	RMS (group average)		17.3		17.1			17.2				@
TEST	RMS error		17.4	17.4	17.5	17.6	17.9	17.5	17.5	17.6	17.4	
TEST	worst case		55.3	57.9	60.3	59.4	51.7	53.4	58.9	56.8	56.0	
COMPARISON	F-ratio σ_1^2/σ_2^2		1.11	.838	.994	1.22	.900	.783	1.09	1.45	.673	
COMPARISON	degrees of freedom ν_1		142		190			214				
COMPARISON	degrees of freedom ν_2		135		87			63				
COMPARISON	average probability α		56%		46%			57%				@
	ratio ν_2/p	39.6	19.3		12.5			9.0				

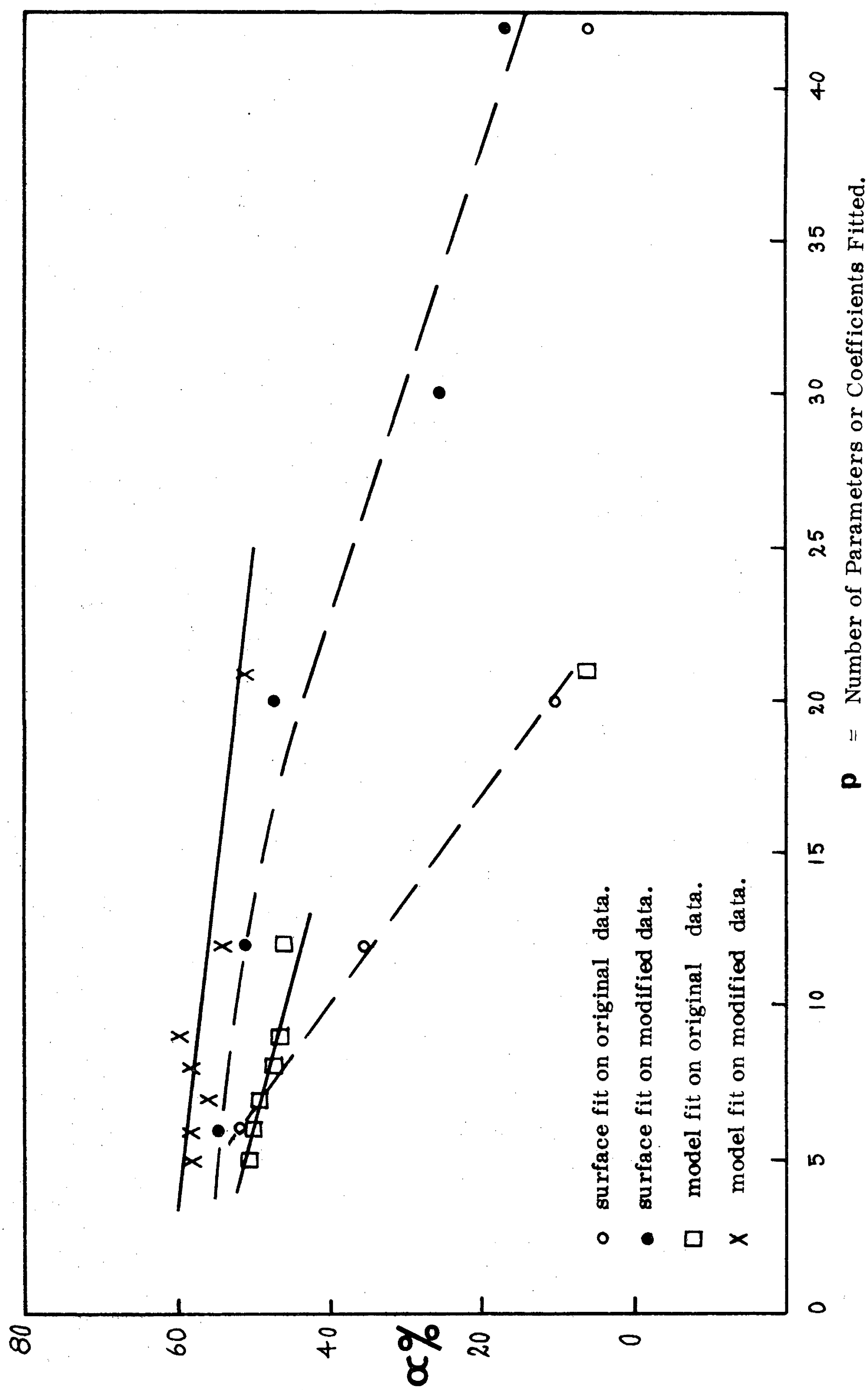
@ denotes quantity averaged over group of data subsets of similar n.

FIG 4.15



o surface fit on original data.
● surface fit on modified data.
□ model fit on original data.
x model fit on modified data.
RMS
arcsec
30
20
15
10
5
0
5
10
15
20
25
30
35
40
p = Number of Parameters or Coefficients Fitted.

FIG 4.16



α%
80
60
40
20
0
0
5
10
15
20
25
30
35
40
p = Number of Parameters or Coefficients Fitted.
o surface fit on original data.
● surface fit on modified data.
□ model fit on original data.
X model fit on modified data.

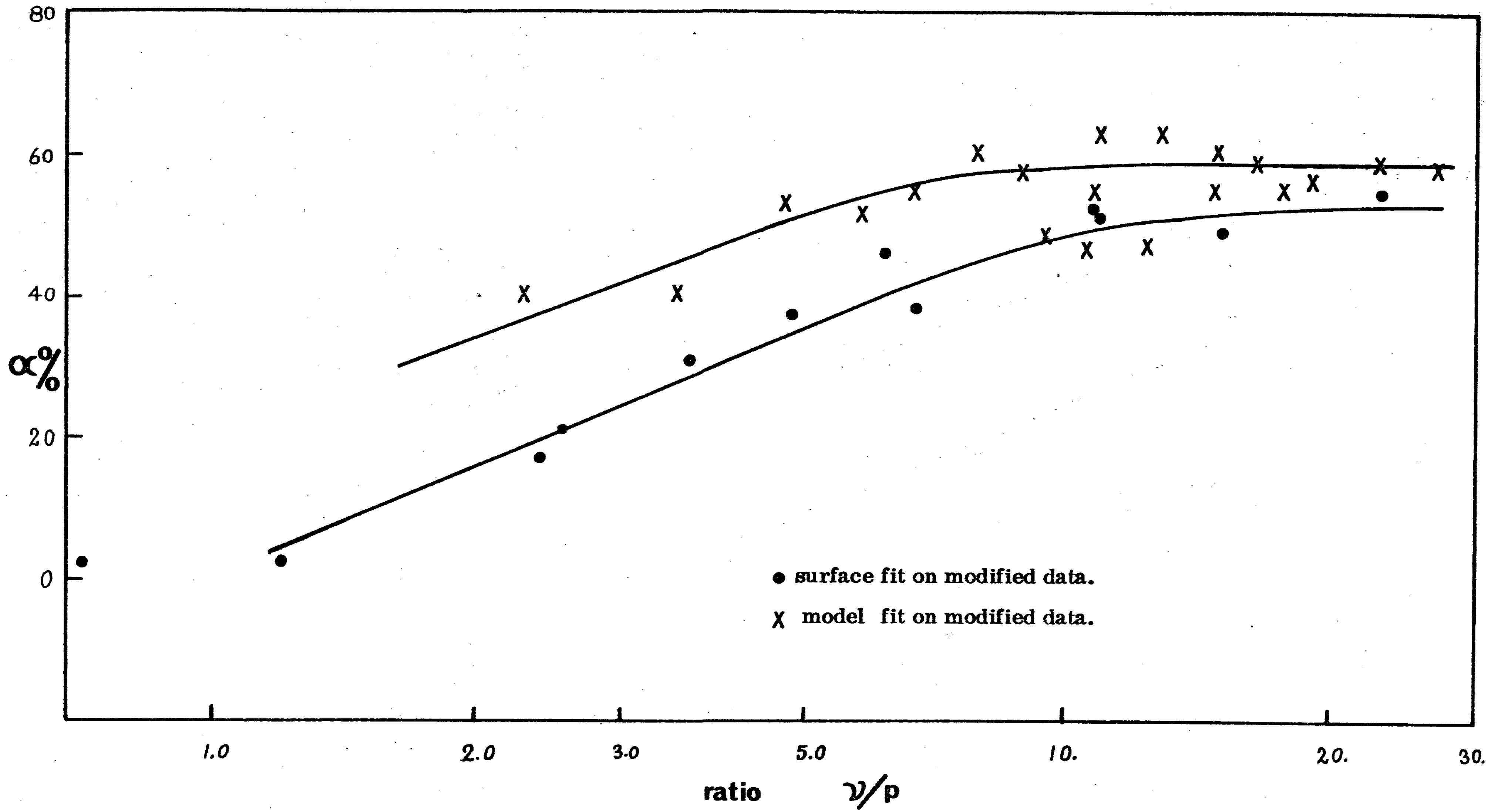

80
60
40
20
0
α%
1.0
2.0
3.0
5.0
10.
20.
30.
ratio ν/p
● surface fit on modified data.
X model fit on modified data.

FIG 4.17

The line-printer plots of the residuals of fit v.s. the co-ordinates also show the benefits of removing the spurious points. Those for the surface fits are improved with regards the scatter of the residuals, and the suggestion of a trend in $\Delta H \cos\delta$ v.s. H for the fit of order 5, and the trends in the plots of order 2 are rendered less significant. The model fit residual plots are also improved but only slightly so; the quadratic trends in $\Delta H \cos\delta$ v.s. δ and $\Delta\delta$ v.s. H, examples of which appear in Figures 4.12 and 4.13, are still very significant.

(4.6) DISCUSSION OF TELESCOPE DATA FITS

Despite the degree of randomness imposed on the results by the hysteresial nature of the telescope errors, interesting comparisons are possible. In Figure 4.15 the R.M.S. error of the surface and model fits to the complete data sets (both original and modified) is plotted against the effective order of fit p[@]. Quite distinct trends are visible and are delineated in the figure. They are different for the surface as opposed to the model fits in that surface fitting can produce superior fits for small values of p, but at p=8 to p=10 the fits have similar R.M.S. error. Rejection of outliers causes the trends for the modified data to be depressed by an (approximately) constant number of arcseconds from those of the original data set. In Figure 4.16 the significance percentage α averaged for the original and modified half data sets XY2 and XY3 is plotted against p. The trends in Figure 4.16 are much less definite than in the preceding figure, but show that worthwhile improvements in the significance level α can accrue from rejection of spurious data points, and that (at least on the basis of α) we can expect slightly more adequate fits from the model rather than the surface fitting process.

Overfitting occurs when the ratio of ν_2, the degree of freedom of fit, to p, the number of parameters or coefficients fitted, is too low. The literature is quite vague on the minimum necessary value for ν_2/p, but values of about 5 or 10 are taken to be desirable. Clearly, this is not always practicable when data acquisition is onerous and time consuming, and when quite complicated behaviour is being fitted. A plot of α averaged within each group of modified data subsets v.s. the ratio ν_2/p for the surface and the model fits is shown in Figure 4.17. There is a definite trend in the adequacy of the surface fits, and to a lesser extent in that of the

---

@ The effective order of fit p is the total number of parameters (p = k') or the total number of coefficients (p = 2k+2) fitted.

model fits (owing to the dependence which model fitting has on exactly which parameters are operative). It is not easy to discern the value of ν_2/p for which either type of fit becomes overfitted. However, even for the surface fits, Figure 4.17 indicates that reliable fitting results may be possible down to ratios as low as 2 provided reasonable attempts have been made to rid the data of spurious points.

Though complicated by the fact that our independent variable is two-dimensional, visual inspection of residual plots is the most useful single method of assessing the appropriateness of the model. For order k=5 and above, the surface fit residual plots show no sign of trends and it is clear that surface fitting allows fits with less systematic variation in the residuals. In general the model fits display a larger incidence of trends and uneveness in the plotted residuals, the exception being the fit with PC1 (all parameters operative) which compares well with the best of the surface fits above. In the fits to PC2, 3, 6, 9, 10 and 13 a trend of form $-\cos(\delta-\phi)$ is evident to varied extent in the plots of $\Delta H \cos\delta$ v.s. δ. The worst case of this particular trend is for the fit with PC10 and is seen in Figure 4.12. The simplest possible cause of such behaviour would be the lack of a term in $\delta-\phi$ in the model function component $f^{(2)}=H_c$; a similar term (the term in b_{20}) already appears in the component $f^{(1)}=\delta_c$. It is unlikely to be caused by parameter redundancy in the model since it appears in the case of the surface fit with k=2, but is undoubtedly affected by redundancy since the model fit with PC1 manages to remove it by employing the slight redundancy which exists between the terms in b_4 and b_{19}.

As well as the trend noted above, the model fit for PC3 has a residual trend involving a linear dependence of $\Delta\delta$ on δ, but since the fit, which uses only the five parameters of the simple model of Chapter 2, is very poor, there is very little to be learnt there-from. Two other trends noted: a cosH type trend in $\Delta\delta$ v.s. H in the fit with PC9 (see Figure 4.13), and very slight $\cos(\delta-\phi)$ tendency in $\Delta\delta$ v.s. δ in the PC13 fit, do appear to be caused by parameter redundancy. The parameter combination PC9 excludes the terms in b_6 and b_7, and b_{21} which is redundantly linked to them tends to act as a substitute, so introducing an excess of the term $\cos(a_3H)$. The minor trend in the PC13 fit is not observed in any other fit and is probably caused by the absence of b_5 which causes b_6 and b_7 to be adjusted. The interactions between parameters are best assessed by an examination of the coefficient correlation matrix in Figure 4.8.

(4.7) CONCLUDING DISCUSSION

The above study though specific, permits a number of important general conclusions on the use of purely computational correction methods for reducing pointing errors. They are listed and discussed below.

The construction of a suitable model for a given case is always a problem, but it is fortunately simplified by the fact that:

(a) simple additive models constructed by summing the separately calculated causes of error are perfectly adequate. This is primarily due to the smallness of the errors with respect to the co-ordinates and the orthogonality of the co-ordinates, and allows the simple calculation of analytic derivatives which can be advantageous in model fitting routines.

(b) Such models are usually intrinsically reasonably linear for the reasons just given, but in any case the use of a non-linearity measure such as Beale's is not recommended. The criterion is too arbitrary, the numerical results too dependent upon the exact siting of the sample points, and the computing time requirements are more onerous than any other program treated here (even those for the data fitting itself).

(c) It is not easy to devise a model with sufficiently independent parameters. Although in general, models may be more complicated and somewhat less linear than the one used, larger numbers of parameters imply an increasing amount of redundancy in the parameter set which is not always obvious from an inspection of the model function. A term appearing solely in $f^{(1)}$ cannot be redundant with one solely in $f^{(2)}$, but within $f^{(1)}$ or $f^{(2)}$ unexpected interdependencies occur which are detrimental to accuracy and adequacy of fit.

(d) Because of point (c) above, the fitting of models with large numbers of parameters is not as useful as would be expected, as a means of locating error causes or measuring physical parameters (unless tests show redundancy to be very slight or nonexistent). Table 4.9 shows the extent to which the fitted parameter value can vary owing to redundancy, however:

(e) because a model can contain terms which describe the telescope behaviour better than polynomials, the adequacy of fit and therefore the usefulness of the fit for prediction purposes is often better for model fitting than for surface fitting. The smallest possible number of parameters should be used, but, should the model fail to account for some trend in the telescope errors, a surface fit with equivalent p will be found superior. This is simply because the terms (polynomials) used in surface fitting span the set of all polynomial functions in the domain of δ, H which are of degree $\leqslant$ k. Visual inspection of residual v.s. co-ordinate plots is the most useful means of assessing adequacy of the model in this regard.

(f) For a given order p, surface fitting will always produce a fit of lower R.M.S. error, but there can be a range of p (= 8 to 10 in our case) where surface and model fitting are not significantly different.

(g) As is evident in Figure 4.3 surface fits should be generated with k = 2, 5, 9, 15 etc., (see Chapter 3, Table 3.1) so that all terms of degree k are employed; sudden decreases in R.M.S. error are often observed whenever a polynomial group is completed.

(h) Attempts to eliminate outliers or spurious data points are very worthwhile. In Figure 4.16 the adequacy of fit is shown to be more dependent on whether or not the spurious points are included than on whether surface or model fitting is used. Elimination of spurious points causes a decrease in the R.M.S. error which appears to be roughly independent of the order of fit and which of surface or model fitting is used.

(i) The adequacy of a surface fit is particularly contingent on the rejection of outliers since the polynomials are particularly suited to fitting a trend in the data which is unexpected, and which would not have been incorporated into a model function were model fitting employed.

(j) Finally, if there are very few spurious data points, a degree of freedom to order ratio ν_2/p of as low as 2 can be used when the aim is to satisfactorily correct telescope errors; for the determination of error causes ratios of at least 6 are required.

The contribution of the mechanical hysteresis error to the nett pointing error of the 74-inch telescope can be estimated, albeit rather roughly. It is true that at least a minute of arc hysteresial movement was found in the secondary support by manually pulling on it, but the level of hysteresis manifest in the pointing errors is (fortunately) far less, as is evidenced by the ability of the routines described to generate adequate fits to less than 20 arcsecond R.M.S. It is highly improbable that any more than a few arcsecond of this is due to the new encoder and timing system (E.T.S., see Appendix C) because of the basic design and the consistency indicated by various checks on it, and since the pointing observations were the subject of considerable care, it is concluded that most of the residual R.M.S. error is in fact hysteresial. A figure of about $\frac{1}{4}$ arcminute R.M.S.

is quite plausible and is in approximate agreement with the figure of 30 to 35 arcsecond peak to peak suggested by the laser data plots Figures 4.6, 4.7.

Despite the spirit of pessimism vested in point (d) above certain other conclusions relevant to the 74-inch are appropriate:

(k) Given the current amount of hysteresial movement in the secondary, composite gear tooth errors and the eccentricity of the encoder pinions are negligible. The terms involving b_8, b_9, b_{10}, b_{11}, b_{14}, b_{15}, b_{16} and b_{17}, because of their period are each quite independent, and in all the model fits generated the estimates of those parameters have very low values. It is unlikely that if the hysteresis were eliminated any significant change would be observed in those values and thus we can eliminate composite tooth errors and pinion errors from any discussion of error causes. It is unfortunately not possible to assess the eccentricity of the main instrument gears since the parameters b_6, b_7, b_{12} and b_{13} are each redundantly linked to several others.

By use of the parameter freezing technique described earlier it is often quite possible to eliminate those other parameters which are redundantly linked to the one of interest.

(l) Thus reasoning, it is possible using Table 4.9 to set an upper limit on the 74-inch axis skewness of about 6 arcsecond.

(m) The final conclusion specific to the 74-inch is that, on the basis of an inspection of the various error v.s. co-ordinate plots there is no obvious transformation of co-ordinates (e.g. into azimuth/zenith angle), or of errors (e.g. magnitude/position angle) which would facilitate improvements in the 74-inch data fitting.

It is clear that a definitive assessment of the use of model estimation in locating and measuring pointing error causes would require a telescope with a much lower level of hysteresial error, and since this type of error cannot be simply dealt with by the techniques discussed in the foregoing chapters, it should perhaps be given priority in any program of telescope pointing improvement. In the case of the Mt. Stromlo 74-inch, the author recommends that the secondary optics support and focussing system be redesigned as this is the major cause of that telescope's hysteresial pointing data.

The following and final chapter discusses the use of model and surface fitting in a practical environment, and emphasizes the fact that hysteresial error is indeed the most serious limit on their effectiveness.

CHAPTER FIVE
THE FUTURE OF SOFTWARE POINTING ERROR CORRECTION

(5.1) SOFTWARE CORRECTION IN PRACTICE

In the foregoing chapters, techniques for computing model and surface fits to telescope pointing error data have been described and evaluated. This has been done using a synthesized or previously collected data set and a large off-line computer, and was not subject to the various constraints and problems that beset the task of using such techniques in practice. The following discussion places these techniques in the context of their practical usage, and delineates the difficulties which are likely to be encountered.

It should firstly be noted that the process of calculating a suitable fit to a set of pointing error data is significantly more demanding in terms of computing time, storage requirements and numerical precision, than evaluating that fit at a given point on the sky for the purpose of error correction or reduction. Since it can always be arranged that the errors in the co-ordinates are evaluated, rather than the corrected co-ordinate values per se, a lower degree of numerical precision can be tolerated when only fit evaluation is required. This may often permit a reduction by a factor of 2 in the storage (from double precision to single precision), and of 2 to 4 in execution time required for evaluation. Quite apart from this, the programs used in previous chapters indicate that, for both types of fit, approximately 3 times the number of instructions and at least 10 times the data storage are required for fit calculation, compared to evaluation. Comparisons of execution time are more variable and range from a factor of 10 to 200. Thus, interpolation or evaluation of a fit is quite suited to implementation on a small computer on-line to the telescope concerned, but fit generation on such a machine may require a closer look at the storage and time overheads.

Two distinct modes of software error correction can be envisaged. In one, a continuously active program in the telescope computer invites the astronomer to log any definite identifications of an object during the course of normal astronomical observation; this accumulated data is fitted by an automatic fitting routine in the computer, when other higher priority tasks, such as telescope or instrument control and astronomical data acquisition, are dormant. This is discussed below in section (5.3). Alternatively, the generation of an error correction fit can be another maintenance-type task like instrument change-over or optics aluminizing, for which the telescope must be 'down'. In this case a grid of bright stars may be observed as in Chapter 4, and processed to yield a '<u>static data set</u>'. From the foregoing paragraph it is clearly simpler for this preliminary processing, and the generation of a fit to the data set, to be performed on a large computer off-line. However, such a machine is not always available, and, for reasons of over-all efficiency (as in any scheme that necessitates the transfer of data between machines), may not be desirable. Because of the increasing extent to which fast access, large capacity disk storage is being utilized in small computer installations, it is the author's contention that software correction schemes can be completely implemented on small telescope computers.

(5.2) THE 'STATIC DATA SET' APPROACH

Whatever size machine is used, the desirable procedure for the static data set approach can be specified as follows:

(i) Firstly, it is important that graphs of the pointing errors v.s. co-ordinates be produced and examined for trends and hysteresial effects. Other ordinates which should be graphed are the resultant error ΔR, the error resolved along the zenith circle ΔZ and also along any other direction of physical significance for the telescope concerned. Additional abscissae should include the zenith angle and any direction of special significance. Although it has not been done in Figure 4.3, the points should be identified with their co-ordinates (other than that which is being graphed as the abscissa) since surface trends in two dimensions are not easily noticed in simple ordinate-abscissa plots. The use of contour plots of an error against two co-ordinates, e.g. δ and H, would be very useful; there is an increasing usage of such plots in other fields which involve two-dimensional data, such as seismic data processing and in geomorphological studies (see for example Harbaugh and Merriam 1968, who describe their use in trend surface analysis of geomorphological data). Obvious trends visible in these graphs may alter or reinforce our ideas on the causes of error, and may suggest a suitable transformation of the co-ordinate variables, which will amplify the trends and permit more effective fitting. The level of hysteresial error can be roughly assessed by comparing the error at adjacent observations which are within say 6 degrees of each other; error jumps in excess of about 10% of the total error variation usually indicate hysteresis rather than a fast-varying repeatable error cause.

(ii) It should be noted at this point that, if surface fitting alone was contemplated and the graphs show simple but distinct trends, a model, no more complicated than necessary to describe the trends, should be used and the residuals to this model fit, surface fitted. In such cases this will always be more effective than a surface fit alone.

(iii) Outlying or spurious data points should be removed from the data set. Data points whose observational accuracy is suspect for <u>any</u> reason should be removed, whatever form the graphs take. If there are no simple trends visible in the graphs these are the <u>only</u> points which should be considered as spurious; if obvious trends exist, points which are inconsistent to it may also be removed, but only if they are placed at least twice the standard deviation from the centre of the trend. (See, for example, data points number 148 and 75 in Figure 4.3). During the fitting process, points which are <u>consistently</u> the worst case errors in fits of different order may also be advantageously discarded.

(iv) A surface fit in the co-ordinate variables δ and H should be generated. The R.M.S. on-sky error should be graphed against the order of fit, to determine the appropriate order to use for evaluation purposes, and it will often be best to use the various orders of fit k = 2, 5, 9 etc. at which the groups of orthogonal polynomials become complete. Lack of fit due to inadequacy of the fitting function@, can be tested for by dividing the data set in half and testing the fit to that half on the other half, as in Chapter 4. The F-ratio test should be employed as a criterion in this cross checking, but there is little point in dividing the data set into more than two (or at most three) parts, since overfitting can occur with small data subsets. The use of the F-test on <u>consecutive</u> orders of fit, to assess significance of fit with increasing order is extremely misleading; see for example Figure 3.8. A modification to this test which renders it somewhat more useful is given in section (5.3.3).

(v) Surface fits of various orders should be generated using the variables cos δ and cosH, where δ and H have been scaled as in Chapter 3, and also fits using the co-ordinate variables transformed in any way which is suggested by the graphs of step (i) above, or

---

@ i.e. inappropriate choice of independent variables and order of fit.

the physical nature of the telescope. The various tests mentioned in (iv) should be carried out.

(vi) If a suitable model for the error causes can be devised, it should be fitted using the Marquardt algorithm and with estimated derivatives. The effectiveness and accuracy of numerically estimated derivatives depends on the numerical precision used, and where unstable behaviour of the model estimation program is observed in a reduced precision environment, recourse should be had to analytically calculated derivatives. The tests of step (iv) above should be used, and model parameters of less importance may be frozen at desired values, to attempt to distinguish the role played by the various error causes.

(vii) The model equations and the correlation coefficient matrix A^* (see Figure 4.8 and equation 2.22) should be inspected for instances of parameter redundancy. Such redundancy may not prejudice the fit's utility for error evaluation purposes, but must temper any conclusions about the physical nature of the error causes obtained in step (vi).

(viii) Finally, the fit which is accepted for evaluation purposes should of course be the one with the lowest R.M.S. error, consistent with a satisfactory result for the lack of fit test on the divided data subsets.

(5.3) AN AUTOMATIC ERROR CORRECTION PACKAGE

The static data set mode of software pointing error correction described above requires a substantial level of human involvement, judgement and decision making, not to mention about a night of observing time, which is a scarce commodity given the current pressures on the observational scheduling of larger telescopes. Daylight observation of bright stars is quite possible[@], but, because of heating of the telescope structure and optics, and the restriction that the observations cannot be too near the sun, is not particularly practicable. Hence we return to our notion of an automatic package which is not seen by the telescope user, and which processes sequentially obtained pointing data, and corrects the telescope pointing on the basis of the best currently available fit. Surface fitting, because of the way in which <u>independent</u> fitting terms are added in a simple hierarchical manner, is better suited to automation,

---

@ In the author's experience, the daytime limiting magnitude in the case of the Mt. Stromlo 74-inch is approximately magnitude 4.

and is more tolerant of reduced precision than a model fitting process (largely because the latter involves matrix inversion). Even so, double precision arithmetic would be required on 16 bit word machines, and possibly even triple precision for astronomical co-ordinate corrections where required@. Matrix inversion would almost certainly require triple precision.

(5.3.1) Storage Requirements

Provided suitable program segmentation or overlaying is employed, and disk memory available, the storage requirements of software correction are certainly practicable, even for quite small machines. Table 5.1 shows estimates of the data storage necessary when generating surface and model fits, and when evaluating them at a point on the sky at which error correction is desired. The storage is given as a number of variables, and should be multiplied by the number of words occupied by a real variable in the implementation to be used. The estimates have been obtained from the routines mentioned in earlier chapters by neglecting unnecessary storage, and assuming, in the case of each dimensioned variable, the most compact form of storage. Approximate instruction code requirements are given in Table 5.2 in numbers of words. In estimating these, large amounts of 'housekeeping' and experimental code that appears in the listings in Appendix D had to be judiciously neglected, and so the estimates should be taken as a guide rather than exact figures for a particular implementation. The instruction code estimates are also exclusive of the various arithmetical and trigonometrical functions, which are required by all of the program segments.

There is little difference between the two types of fit in the case of fit evaluation, but somewhat more storage is required to generate a surface as opposed to a model fit, primarily because, during the fitting process, the orthogonal polynomials $P_j^{(1)}(\underline{x}_i)$ are stored by their values at each data point. This is not strictly necessary, see for example equations 3.22, and the reference to Cadwell and Williams 1961 in section (3.7), but more compact storage is gained only at the expense of decreased numerical accuracy, which is not tolerable in small word-length machines. Extensive use can be made of integer formats (multiple length where necessary) for storing pointing <u>data</u>, but fits must be stored by the Forsythe coefficients

---

@ A precision of 60 bits in the mantissa of a real variable was used in the computations discussed in earlier chapters; the level of error introduced into astronomical correction algorithms by the use of a 24 bit mantissa is briefly noted in Appendix A.

TABLE 5.1

Data storage requirements for fit generation and evaluation purposes;

> n = number of observations, p = order of fit;
> numerical estimates assume $n = 100$, $p \doteq 20$;
> storage is in terms of number of variables, and must be multiplied by 2 (3) if double (triple) precision arithmetic is used.

SURFACE FITTING	generation	evaluation	MODEL FITTING	generation	evaluation
data points $\underline{x}_i$	2n	2	data points $\underline{x}_i$	2n	2
data values $\underline{y}_i$	2n	2	data values $\underline{y}_i$	2n	
coefficients of fit $c_j^{(1)}$	p	p	function values $\underline{f}_i$	2n	2
			parameters $\underline{b}$	p	p
polynomial values $\underline{P}_j^{(1)}$	np	p	masking vector for parameter freezing	p	
			vector $\underline{g}$	p	
			solution vector $\underline{t}$	p	
Forsythe coefficients $\alpha_{jr}^{(1)}$	p(p-2)/4	p(p-2)/4	matrix (stored as upper triangle) A	p(p+1)/2	
			derivatives (estimated) $\partial f/\partial b_j$	2n + p	
			@ derivatives (analytic) $\partial f/\partial b_j$	2np	
TOTALS	$np+4n+p^2/4+p/2$	$3p/2+p^2/4+4$	TOTALS	$8n+11p/2+p^2/2$ [@ $6n+9p/2+p^2/2+2np$]	p + 4
numerical estimate	2610	134	numerical estimate	1110	24@@

@ analytic derivatives will only rarely be necessary in practice.

@@ this estimate is somewhat theoretical since incidental storage requirements would be at least this much again.

$\alpha^{(l)}_{jr}$ and the coefficients of fit $C^{(l)}_{j}$ and require a real number representation.

TABLE 5.2

Approximate instruction code requirements for fit generation and evaluation

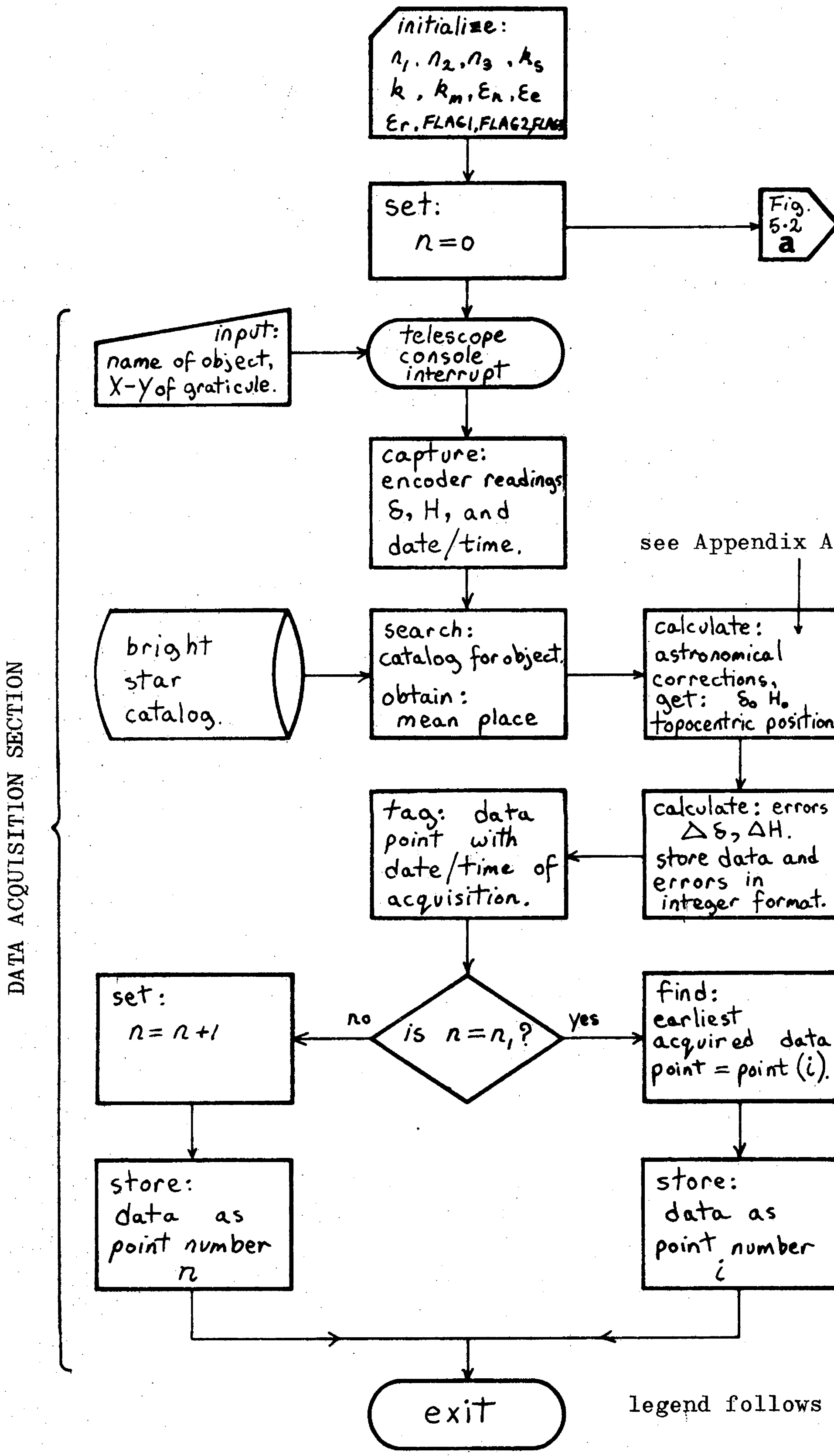
SURFACE FITTING	generation	evaluation	MODEL FITTING	generation	evaluation
polynomial group initialization	210	210	model function evaluation	630	630
orthogonal polynomial generation	570		matrix equation setting up	330	
orthogonal polynomial evaluation		290	matrix inversion	440	
surface fitting routine	740		model fitting routine	780	
TOTAL	1520	500	TOTAL	2180	630

(5.3.2) Automatic Selection of Best Fit

The fit to be used for error interpolation must be automatically selected by our error correction package, and this is not possible if a complicated model with a large number of parameters is employed. Thus surface fitting must be predominantly used in the package. The package could use to advantage an initial model fit which has only a few parameters, each of them continually operative, and only those parameters which specify a definite physical error cause known to be active, for example variation of instrument mass, simple flexure terms and optical misalignments. A preliminary pointing data run may be necessary to determine whether this is worthwhile and which parameters should be involved, but such a run need not be as extensive as for the static data set mode of operation.

Successively higher orders of surface fit should be calculated until either a nominated level of R.M.S. pointing accuracy is attained, or until a Fisher test on the ratio of the variances of fit calculated on the successive orders of fit $k = 2, 5, 9, 14$ etc. (at which polynomial groups are completed) indicates insignificance of fit, which ever is the sooner. The two most important parameters which the system should display to the user as an indication of the performance of the package in real time, are the R.M.S. residual error of the fit being used as the interpolant, and the multiple regression coefficient R^2 (see equations 4.4 to 4.7). The latter measures the fraction of variation in the data which is accounted

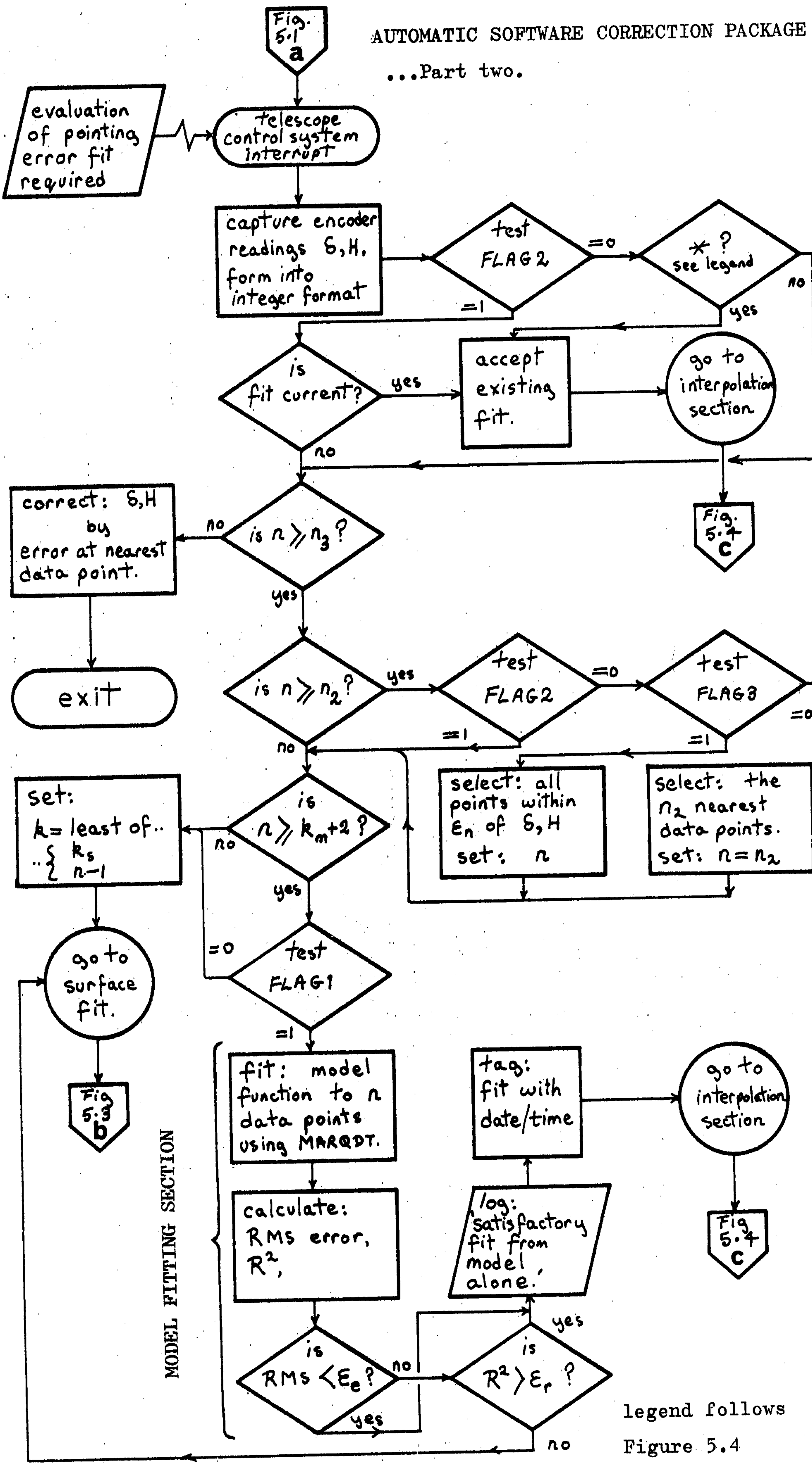
AUTOMATIC SOFTWARE CORRECTION PACKAGE ...Part one.



legend follows Figure 5.4

AUTOMATIC SOFTWARE CORRECTION PACKAGE

...Part two.

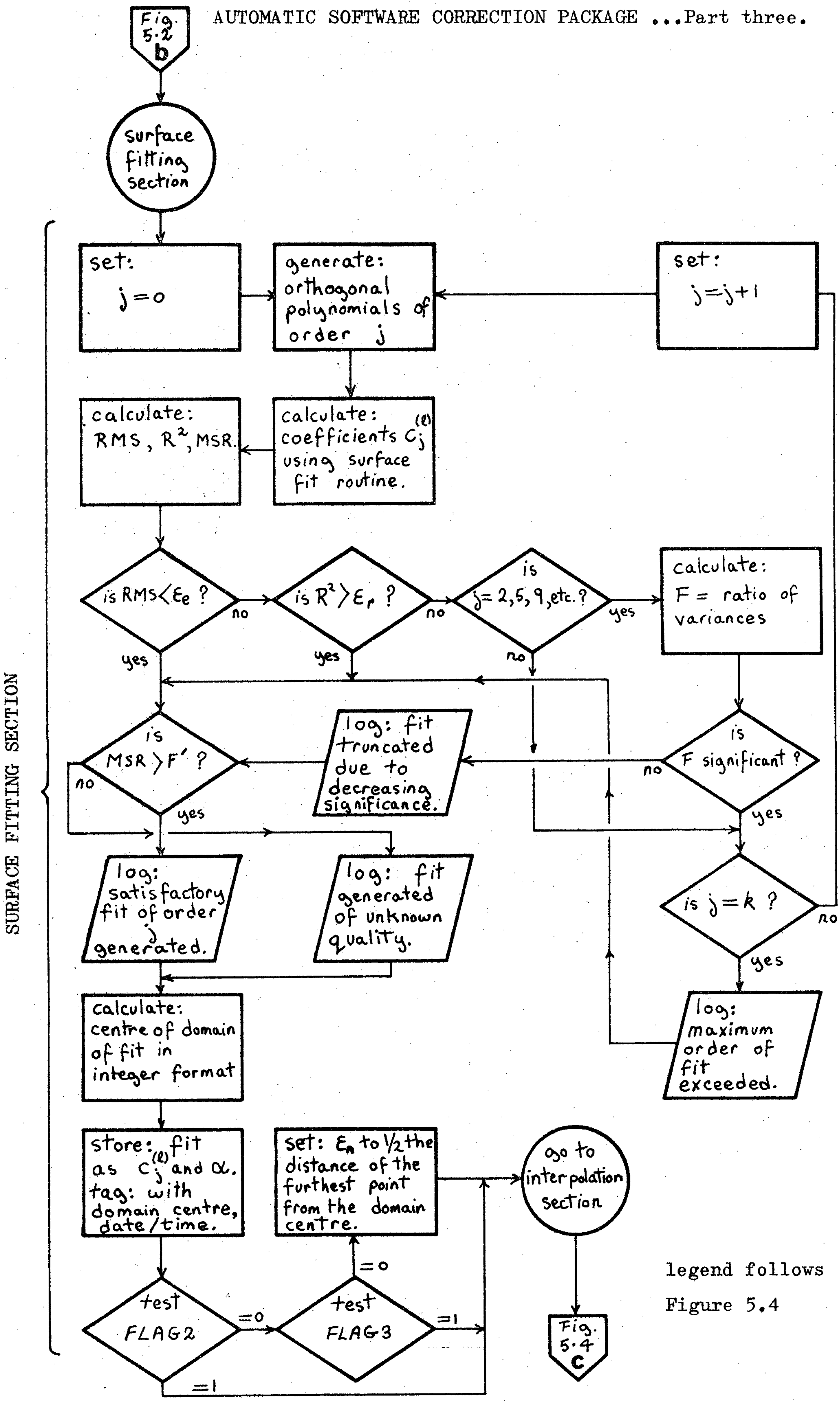


legend follows

Figure 5.4

FIG 5.3

AUTOMATIC SOFTWARE CORRECTION PACKAGE ...Part three.

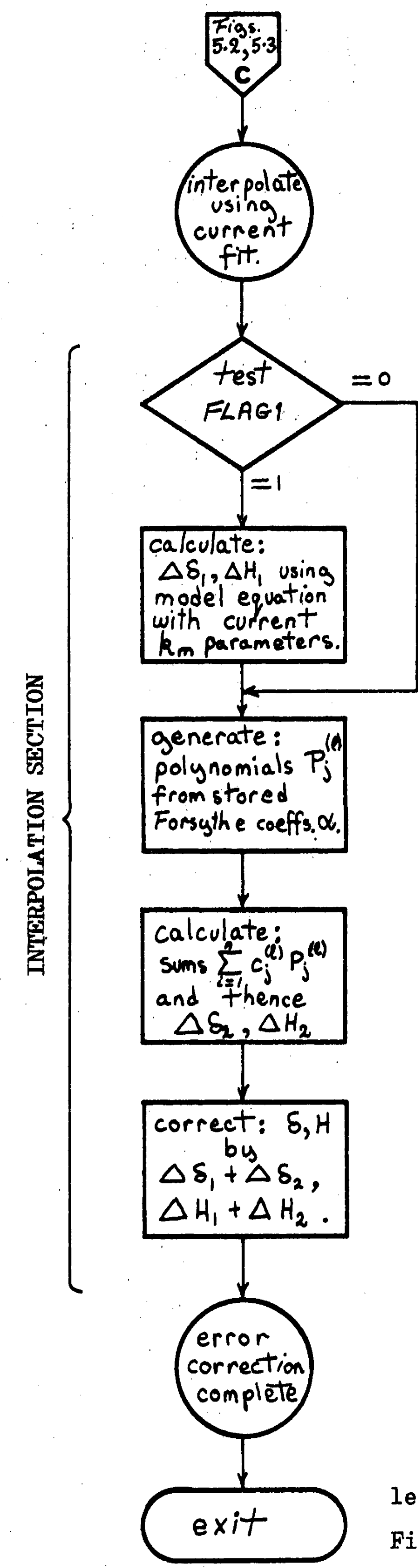


legend follows

Figure 5.4

FIG 5.4

AUTOMATIC SOFTWARE CORRECTION PACKAGE ...Part four.



legend follows this Figure.

LEGEND TO FIGURES 5.1, 5.2, 5.3, 5.4 and ASSOCIATED TEXT

n_1 = maximum permissable number of data points (set by storage allocated).

n_2 = maximum desirable number of data points.

n_3 = minimum desirable number of data points.

k_s = maximum permissable order of surface fit.

k = current limiting order of fit (calculated by algorithm).

k_m = number of (operative) model parameters to be used.

ε_n = radius defining a neighbourhood for restricted domain fitting.

ε_e = Root Mean Square tolerance on fit.

ε_r = percentage tolerance on multiple regression coefficient R^2.

FLAG1 = 1 if an initial model fit is desired;
= 0 if not.

FLAG2 = 1 if global fitting (<u>all</u> data points used in fit) desired;
= 0 if restricted domain fitting (a subset of data points in the immediate neighbourhood is fitted) desired.

FLAG3 = 1 if, in restricted domain fitting, the number of points n in the domain is to be determined by the neighbourhood radius ε_n;
= 0 if the domain is to include the n_2 nearest points.

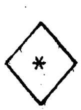
R.M.S. = Root Mean Square residual error of fit.

R^2 = multiple regression coefficient defined in equations 4.4 to 4.7.

MSR = mean square ratio of equation 4.8.

F = ratio of the variances of consecutive orders of fit at which polynomial groups are completed.

F' = $4F_{1-\alpha}(p-1, \nu_2)$, the tolerance on the mean square ratio (MSR) to ensure the fit is useful for prediction purposes; see equation 4.9 and associated text.



= (in Figure 5.2) Decision: is point δ,H within ε_n of centre of domain of current fit, <u>and</u> have no points been added since that fit?

for by the fit, and, along with MSR, the mean square ratio given in equation 4.8, may also be utilized in automatic procedures to determine the minimum order of fit necessary. Even if MSR is not used as a criterion, it should be calculated, and used as an indication of the probable effectiveness of the fit for prediction purposes.

(5.3.3) A Suggested Strategy for an Automatic Correction Package

Figures 5.1, 5.2, 5.3 and 5.4 show the details of a proposed automatic correction package which embodies the ideas so far discussed. It is assumed that whether or not an initial model should be used has been ascertained, and that if necessary, the model function has been incorporated into the system. On initialization, the parameters n_1, n_2, n_3, k_s, k_m, ε_n, ε_e and ε_r, which are defined in the legend following Figure 5.4, are set, and the flag FLAG1 set to 1 or 0 according to whether a model fit is desired or not. Other options available include a choice of whether global fitting (use of all data points for each fit) is to be used, or whether fits should be generated to subsets of data points in the immediate locality of the point at which error evaluation is required. The latter, called 'restricted domain fitting'[@] here, is controlled by FLAG2. When using restricted domain fitting a further flag FLAG3 indicates whether the domain to be used is to be decided on the basis of a certain set number of data points n_2, or by an angular radius ε_n, which defines a 'neighbourhood' or region inside which the data points of the restricted domain must lie.

It is assumed that some form of integer representation for the pointing data is used (in the interests of fast sorting of points and other 'housekeeping' tasks), and that each data point is tagged with an index number (necessary for array calculations), and also an integer code representing the date and time of acquisition. Allowing a precision of about a minute (of time) for the latter and about ½ arcsecond for the co-ordinates δ, H and the errors $\Delta\delta$, ΔH, it can be seen that 5 to 6 words of storage are required for each data point. This is unavoidable since, even for global fitting where a single unique fit is stored, all the data must be retained so that the fit can be improved with the accumulation of more data. In the case of restricted domain fitting a number of fits for different areas could be stored, but since $4k_s^2 + 9k_s + k_m + 5$ quantities need to be stored for each fit, and since such a procedure may result in

---

@ In a sense this is piece-wise fitting, but without the important constraint that the fits to the various pieces match along their common boundaries in some way.

the use of an inferior fit to evaluate the error at a general point, this is not recommended. Each fit generated is tagged with the date and time of its generation.

The data accumulation section of the algorithm is shown in Figure 5.1, and accumulates pointing data until n_1, the available storage allocation, is exceeded[@], whereon it substitutes the point immediately acquired for the earliest point. When the telescope control system requires the pointing error to be evaluated at a given point in the sky, the algorithm checks whether the existing stored fit is current and appropriate (Figure 5.2); in the case of global fitting, this is contingent on whether or not data points have been added since the fit was generated, but for restricted domain fitting, there is the additional test that the point of evaluation is within the neighbourhood ε_n of the centre of the existing fit domain. ε_n, depending on FLAG3, is either set at initialization time, or is calculated on the basis of the distribution of points in the existing fit domain (see Figure 5.3). If the existing fit cannot be used, a new fit is generated and, if FLAG1 = 1 and there is adequate data, includes an initial model fit, as well as a surface fit whose order is chosen using techniques discussed in section (5.3.2) above. If the total number of data points n exceeds n_2, restricted domain fitting can be employed, and the data set to be fitted is chosen as the nearest n_2 data points (FLAG3=0), or as all points within radius ε_n (FLAG3=1).

(5.4) THE PROBLEMS INVOLVED IN SOFTWARE ERROR CORRECTION

The use of on-line disk storage with small telescope computers makes the implementation of an automatic error correction algorithm, such as described in section (5.3) above, quite practicable. When tracking an object, or when setting on different objects in the same region of sky, the frequency with which new fits must be generated will be quite low, particularly if global fitting is used; thus much of the algorithm can be rolled in from the disk in a number of segments as required, and the whole process can be relegated to a fairly low priority. Similarly, if telescope time can be set aside for occasional pointing error data runs, for example in poorer conditions or during twilight, software correction employing a static data set approach, of section (5.2), can certainly be implemented in the form of a concatenation of disk-resident programs. However, there exist a number of problems which are common to whichever of the two approaches is used.

@ Alternatively an age limit could be imposed on the data points.

(5.4.1) Change of Telescope Configuration

Pointing data acquired using a particular telescope configuration, that is, a particular instrument, focal station, balance weight setting etc., will not in general be pertinent to the behaviour of the telescope differently configured. Since instrument changeover and other alterations occur quite frequently[@], there is the problem that error fits become obsolete rapidly, and there is insufficient time spent in the one configuration to permit the accumulation of a satisfactory number of data points. The only remedy applicable here, is to alter the telescope hardware or operating procedures so that change of configuration produces less variation in pointing error.

The two most important configuration variables are the optical adjustments and changes to the state of balance of the telescope. Particularly for telescopes for which optical changeovers involve the manual replacement of a mirror or other components, rather than mechanically switched systems, for example flip mirrors, the position of the optic axis changes with change of configuration and is not even constant for a given configuration. A consistent scheme for collimating the telescope is required, and the increasing use of small, low power (typically several milliwatt) alignment lasers is a step in the correct direction. Focal stations should incorporate a fiduciary or reference point in the focal plane, to which the optic axis is consistently adjusted whatever optics are employed. Balance changes with different instrument weights can be similarly obviated, by the (automatic?) adjustment of the balance weights for a specific state of balance of the telescope in a given attitude. Often a (known) imbalance is to be preferred, since many telescopes exhibit more constant and repeatable pointing errors with suitable preloads and biassing forces.

Provided the pointing behaviour can be made consistent for a given configuration, the change in behaviour for different configurations may be allowed for by additional parameters, e.g. optic misalignments, balance weight and flip mirror positions etc., which are included in the model estimation process, and, either initialized differently for different configurations, or fitted using a <u>limited</u> amount of fresh pointing data, obtained after a changeover. Thus an existing global fit could be slightly

---

@ e.g. almost weekly for optical telescopes, whose usage scheduling is based on the lunar month for reasons of sky background brilliance. Radio telescopes are far less affected by instrument changes.

modified to accomodate changeovers. Alternatively, data taken for different configurations can be distinguished, separately stored, and the appropriate data set selected; this is, of course, only practicable if a relatively small number of different instruments, focal stations etc., are used. Often different telescope attitudes may have to be considered different 'configurations' in this context; an example of this is the discontinuity in excess of an arcminute in declination, observed when the Mt. Stromlo 74-inch is used on the opposite side of the pier[@]. A further point which should be noted is that the extent to which pointing errors are hysteresial may also be configuration dependent.

(5.4.2) Alteration of Telescope System Hardware

Another factor which may jeopardize the usefulness of a set of pointing error data, is the unavoidable removal of, and maintenance on equipment and components which, either influence the direction of the optic axis (e.g. optic supports, focussing units), or comprise part of the attitude read-out chain (e.g. encoders, encoder drive gears). Ideally the component removed for repair or modification should be replaced exactly as found, and, if this is at all possible, it should be attempted. Where a component does not contribute substantially to the hysteresial errors, and is responsible for certain independent parameters of an error model, the techniques of 'parameter freezing', described in Chapter 4, may be useful in refitting just those parameters to a limited amount of new data. This will only be possible if (i) the component contributes a noticeable fraction of the total pointing error and in the manner described by the parameters concerned, and (ii) those parameters are not badly redundant. Unfortunately there is little which can be done if the component is not related to any error cause incorporated into the model; the surface fitting section of the correction algorithm requires a completely new set of pointing data, to generate a fit comparable with the previous one, once any such error contributing component is disturbed.

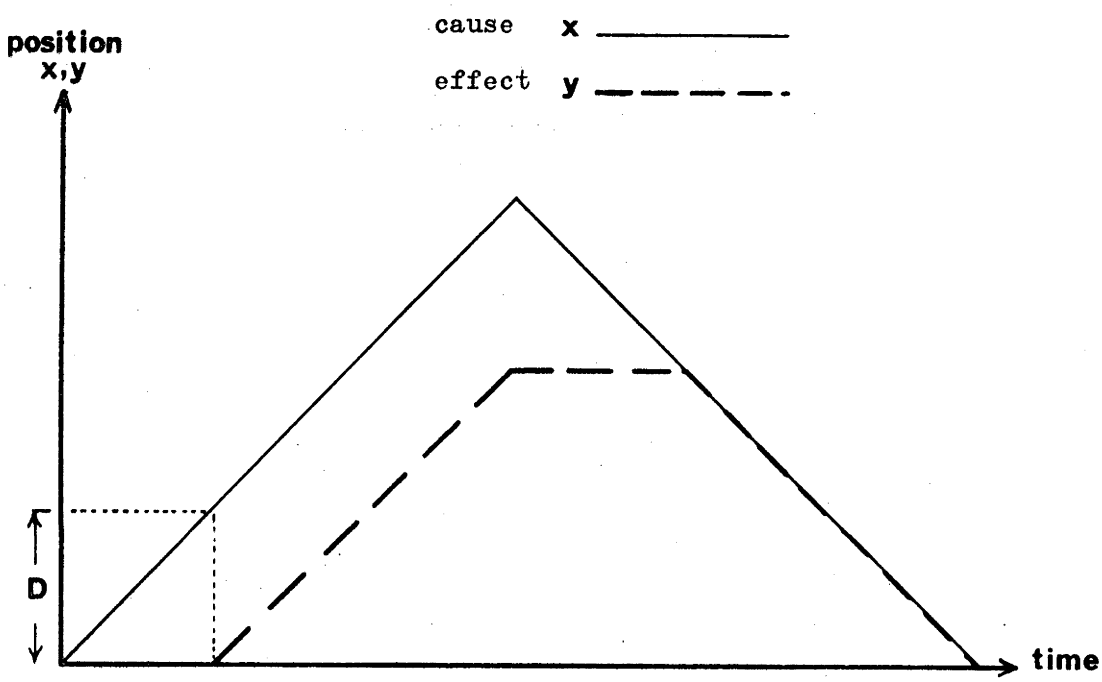
(5.4.3) Hysteresial Errors

Although the various problems mentioned above will assume greater or lesser importance for different telescopes, the problem which is of universal concern, and which most endangers the viability of software error correction, is the presence of hysteresis or nonrepeatable errors. There

---

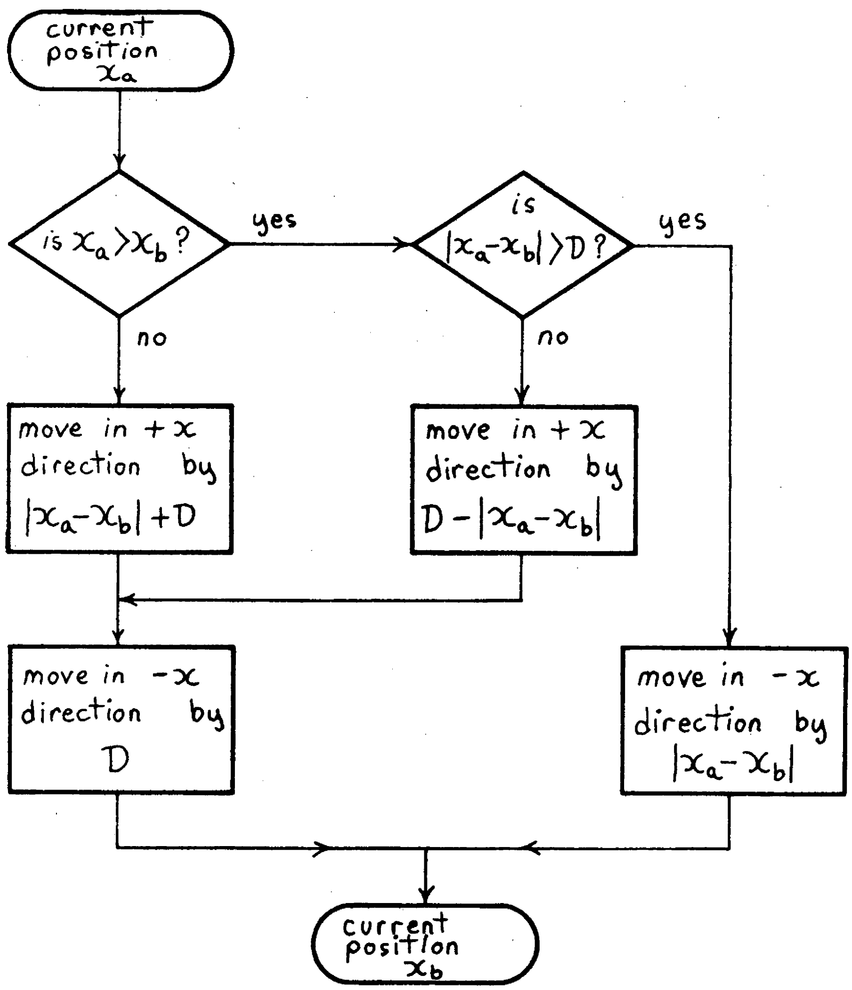
@ This probably is due to the use of a diametrically split instrument gear in the cube (see Appendix C). The gear was split in half to allow it to fit through the cube access hatch during installation.

FIG 5.5

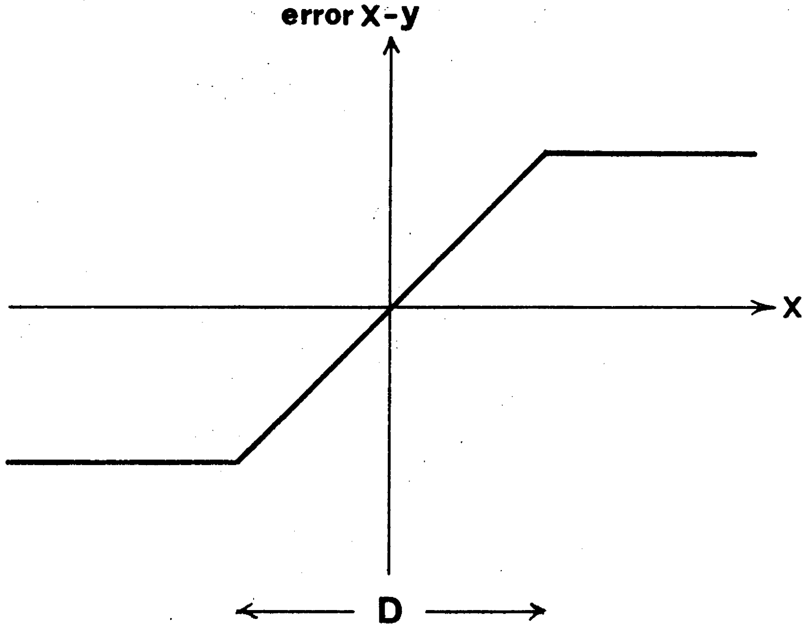


ONE-DIMENSIONAL HYSTERESIS

TRIVIAL ALGORITHM FOR OVERCOMING
ONE-DIMENSIONAL DEAD-ZONE HYSTERESIS

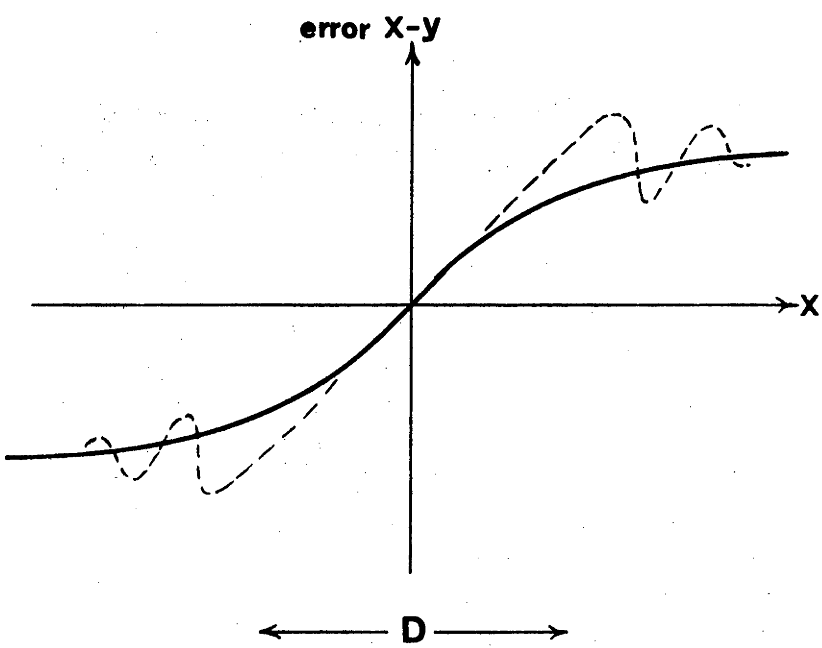


N.B. D is an upper bound to the size of the dead-zone.



SIMPLE DEAD-ZONE HYSTERESIS

(A)

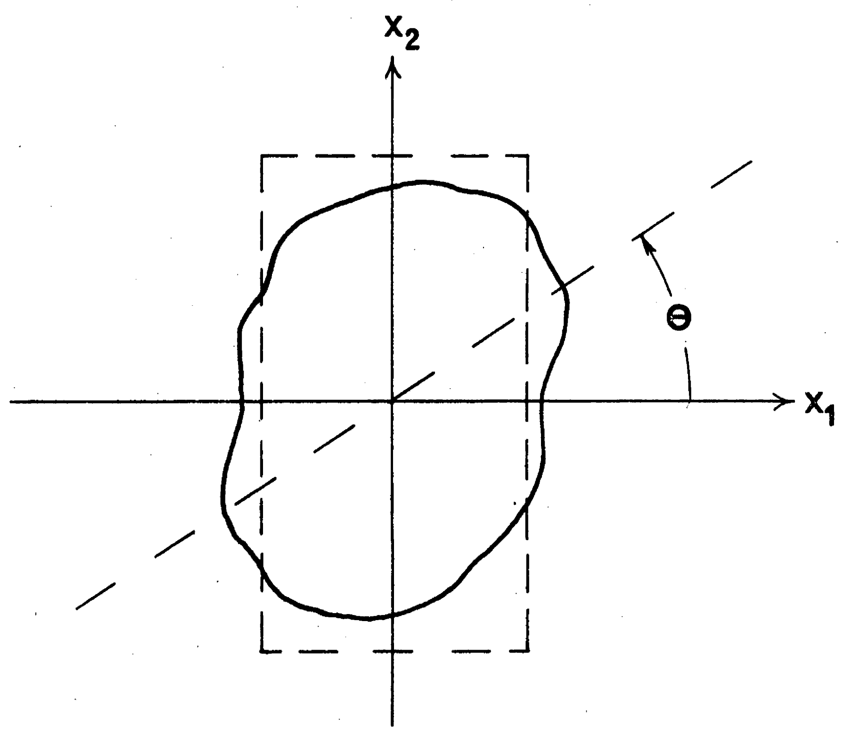


PRACTICAL HYSTERESIAL BEHAVIOUR DUE TO ELASTIC EFFECTS (solid line), or STICTION (broken line).

(B)

FIG 5.8

TYPICAL DEAD-ZONE AREA IN TWO DIMENSIONS



is no way to estimate errors which are a discontinuous (and unknown) function of the co-ordinate variables, unless the statistics of their distribution is known, in which case use could be made of confidence intervals which specify an interval inside which the error has a known probability of lying. In general nothing can be known of the hysteresial error statistics, and it is doubtful whether pointing error confidence intervals are of much use to the observer, if the hysteresis is large. A more rewarding scheme would be to prescribe a process to obtain, or a route to, a given point on the sky, in such a manner that the resulting pointing actually becomes repeatable.

The problem is trivial in the case of one-dimensional positioning. The position v.s. time graph of Figure 5.5 portrays the nature of simple 'backlash' or 'dead-zone' type of hysteresis, for a one-dimensional case involving a dead-zone of size D. To overcome such hysteresis it is sufficient merely to ensure the final positioning is done in the -x direction only. A (trivial) algorithm for moving from position x_a to position x_b is given in Figure 5.6 for the sake of completeness. In practice, hysteresial behaviour is more complicated than simple backlash, and in Figure 5.7 the hysteresial error v.s. position graphs for backlash (Figure 5.7 A) is contrasted with the behaviour of systems which involve elastic effects, such as structural 'take-up' or gear teeth deflection (solid line Figure 5.7 B), or stiction (broken line same figure). A suitable upper bound on the hysteresial error D can conveniently be defined similarly to the waveform rise time in electronics: the position change D required to produce 90%, or some other specified amount[@], of the total observed hysteresis; this enables any oscillatory or asymptotic effects to be included, without using an excessively large value for D. Whatever value for D is used, even if it is a (known) function of position D(x), pointing can be made repeatable by the same algorithm as before (see Figure 5.6).

In two dimensions the problem is far from simple unless the two motions are entirely independent; in this case the algorithm above could be applied separately in the two co-ordinates to produce repeatable pointing, which can then be the subject of the model and surface fitting techniques so far presented. If simple backlash is present in the two independent co-ordinates, the area of cause/effect discontinuity is rectangular as shown by the broken line in Figure 5.8. For a more general case of simple

---

@ Another convenient choice here would be the total hysteresis observed minus the required pointing error tolerance.

backlash, the dead area may be as shown by the solid line in Figure 5.8, and $D = D(x_1, x_2, \theta)$ is now a function of both co-ordinates, and also of position angle θ, and by definition is periodic in θ with period π. Complete knowledge of $D = D(x_1, x_2, \theta)$ is unnecessary, since the largest diameter of the area in Figure 5.8, or the largest projections of it onto the co-ordinate axes, can be used in practice, provided the hysteresial process is no more complicated than a simple backlash effect.

One of the restrictions on the practical use of such notions as above to counter hysteresis, is that the hysteresial process is rarely simple backlash. The error versus co-ordinate curve is more often as shown in Figure 5.7 B for both co-ordinates, and the two motions interact to the extent that the hysteresis in one is affected by the positional history of the other. D for practical telescopes may often be an extremely large angle on the sky, and thus an impracticably long slew may be required about an axis to remove a worthwhile fraction of the error. If we add to these problems the fact that, the hysteresial behaviour of a mechanism which is itself composed of a number of hysteresial components, cannot be treated simply as if it exhibits a positional cause/effect delay, it is clear that hysteresial errors impose a very tangible limit to the efficacy of software correction techniques.

(5.5) CONCLUSION

(5.5.1) Summary of Conclusions from Previous Chapters

The results of previous chapters make a clear case for the feasibility of software pointing error correction, as well as highlighting its attendant difficulties; here we summarize briefly the conclusions to be made from these chapters. The utility of model estimation is assured whenever the experimenter has a firm conviction, or evidence, of the exact causes of pointing error. It will no doubt be found most useful for telescopes of recent design, where elaborate analysis of the pointing performance has already been done. Even where many of the error causes have been physically measured independently, the model estimation routines, along with the scheme of 'parameter freezing' of Chapter 4, will be found most useful for overall testing and improvement. The data perturbation experiments of Chapter 2 show such processes to be quite stable, and although some descent type algorithms are shown to behave extremely badly, routines of the Gaussian type (Levenberg, Marquardt, SPIRAL) are all satisfactory. The algorithm which is to be preferred for reasons of both performance and its uncritical nature, is the Marquardt algorithm, and it is suggested that, on new problems or models, it could be started from a point very close to the origin of the parameter space[@], and with an initial value of λ calculated as per Levenberg (1944). The use of numerically estimated derivatives should suffice for most practical problems, and the performance of model fits will not depend on the fitting routine used, as much as on the properties of the postulated model.

In Chapter 3, two-dimensional orthogonal polynomial fitting of the pointing error surfaces is examined, and shown to be stable and possibly less critical numerically than model estimation. Although statistical assessments of the adequacy of fit indicate that very large numbers of data points are desirable, satisfactory fits can be generated up to polynomial order $\sqrt{2n}$, where n is the number of data points. The decision as to whether to model or surface fit is often a critical one, since some models may not be approximated well by polynomial fits. Surface fitting is by far the method most adaptable to the requirements of an automatic software correction package, due to the hierarchical generation of higher orders of fit, and the applicability of conventional statistical techniques for selection of a satisfactory fit.

Chapter 4 delineates the difficulties imposed by practical pointing data, and the vagaries of telescope error causes. The constraints on the manner in which data can be collected are far more serious than for other similar problems,[@@] and virtually rule out the use of model estimation for 'mechanism determination': the selection and physical measurement of an error cause. In theory, mechanism determination is also precluded by the inappropriateness of conventional statistical tests for assessing non-linear models, fortunately however, because of the magnitude of the errors compared to the co-ordinate variables, most telescope models will not be too non-linear. Although simple additive models are shown to be quite adequate, model construction is difficult, because of the ease with which redundant parameters are incorporated, and monitoring the elements of the correlation coefficient matrix A^* is helpful here. Because of redundancy, model fits with very large numbers of parameters may be less effective than expected. Visual inspection of the plots of residual error against co-ordinates, is the most useful single indication of model adequacy, outlying data points, and the prevailing level of hysteresial error ; provision for this must be incorporated into any model fitting routine

---

@ e.g. $b_j \simeq 10^{-5}$ or 10^{-7} for $j = 1,\ldots k$.

@@ Consider for example, the simple two-dimensional data grids of ship hull fitting, and geomorphological data.

intended for practical application. Despite reservations as to statistical validity, both surface and model fits will usually be found useful for error correction purposes, even when ν/p, the ratio of the degree of freedom of fit to the number of fitted quantities, is as low as 2.

(5.5.2) Suggestions for Further Work

The section (5.4.3) above, as well as defining the limitations of our software correction scheme, also provides a direction for future endeavour. It is evident that, in any projected program of telescope hardware improvement, highest priority must be given to eliminating the causes of hysteresial error; even if these causes are not the largest contributors to the total error, they are the ones least capable of being improved by software methods alone. With special regard to the Mt. Stromlo 74-inch, this priority should be allocated initially to improvement or redesign of the secondary mirror support. On the basis of the results of Chapter 4, the application of software error correction to that instrument would result in a R.M.S. pointing accuracy of about 20 arcsecond, and, with suitable improvement in the secondary support hysteresis, a figure in the region of 10 arcsecond R.M.S. may indeed be feasible. Other improvements to the 74-inch should include a rationalized and calibrated balancing system, and an extension of the current usage of alignment lasers to maintain an absolute relationship between the optic axis and the encoder readout, which would survive configuration changeovers. The author would also like to see the practical implementation of the type of automatic correction algorithm discussed above in section (5.3). There is little doubt that, if incorporated into existing telescope computer systems, the benefits of increased pointing accuracy would be well worth the storage and processor time overheads required. It is suggested that the basic ideas embodied in the algorithm discussed in (5.3.3) be extended, and evaluated on a suitable general purpose, large telescope.

In a more theoretical vein, the phenomenon of hysteresis in mechanical systems is worthy of attention in its own right. The behaviour of a hysteresial mechanism may be best understood from a purely statistical viewpoint, and although this does not easily fit in with the approach to repeatable errors adopted here, it may result in analyses or formulations of the problem, which give as their tangible result, a confidence interval or other type of probabalistic specification for the error of such mechanisms. Most advanced texts on control system theory touch on the analysis of systems employing components with dead-zones or backlash in their (discontinuous) transfer functions, but the prime interest is in the (worst case) stability of such systems, and not in the statistics of their behaviour. State-space techniques are far more applicable here than the more classical approaches to control theory, and within this framework, there is clearly room for a statistical, unified treatment of discontinuous mechanical processes such as hysteresis.

For many telescopes, the pointing behaviour will be found to be largely repeatable, even if erratic, and the greater part of this thesis will apply directly. Yet a number of possible improvements can be suggested. In the static data set case in section (5.2), the operator can assess visually graphs of the errors and residuals of fit v.s. co-ordinates, and can evaluate many different (and often inconsistent) factors to arrive at a decision; in such cases automatic outlier rejection is ill advised. Contrarily, with a completely automatic package, there is a definite need for data rejection procedures to guard against extreme cases. Comparing the currently existing fit, with a fit which includes in addition the data point in question, is one possible strategy. Points which would decrease significantly the adequacy of fit[@] could be rejected advantageously, and statistics of their behaviour logged for possible later investigation. Because of the increasing extent to which computer control and analysis systems have become self contained, and lack interaction with the human operator, it is no longer tenable to shun completely the notion of automatic data screening, and research in this area could be beneficial to many such systems.

Model fitting, because of the conscious selection and formulation of the mathematical model by the experimenter, is not a process conducive to automation. On the other hand, in a surface fit employing orthogonal polynomials in a given variable, the order of fit alone can be altered to facilitate the search for a satisfactory fit. The case for orthogonal polynomials has already been made in section (3.3), and so a method of altering or transforming the co-ordinate variables used (and therefore the basis functions which constitute our orthogonal polynomial functions), is required. Automatic transformation of variables has been used by Thacher and Milne (1960) in surface interpolation studies, but only in a limited fashion. A variable transformation scheme operating within a standard surface fitting algorithm is actually two nested optimization loops, and poses special problems of selection and uniqueness of the solution. Nevertheless, it constitutes an avenue for improvement which would be applicable

---

@ as measured by tests involving subdivisions of the data set as in Chapter 4.

to a gamut of data approximation techniques.

The final line of endeavour suggested here, is the construction of a two-dimensional analog of a class of methods in one-dimension that employ piecewise fitting, and which automatically optimize the positions of the boundaries between pieces. One such routine involves piecewise fitting with cubic splines[@], and dynamically varies the position of the knots to achieve an optimum fit. It has proven extremely effective in a number of curve fitting applications in which conventional methods behave poorly. Mention should also be made of an automatic curve fitting package developed by Payne, in Hayes (1970). A piecewise surface fitting routine with optimal piece boundary positioning, is a difficult, but worthwhile objective.

In general the more accurate and precise a mechanism is required to be the more complex the design necessary to achieve that precision, and as complexity increases so too does the difficulty of tracing the exact cause of a slight defect in the mechanism's accuracy through its multifarious parts and functions. There is clearly a level in the scale of accuracy, where the application of software error correction becomes advantageous. In the case of astronomical telescope pointing errors, that level has been reached, and the author believes that software correction of pointing errors can be effective, and, in contrast to Smith (1967), is difficult, but not 'too difficult'.

@ see footnote page 3.4.

APPENDIX A.

Algorithms for the Computation of Astronomical Corrections

The topocentric position of a celestial object required for telescope pointing purposes is obtained from its catalogued or otherwise known mean place at a given epoch by applying the following corrections:

(i) PROPER MOTION which is the intrinsic motion of the object with respect to the celestial co-ordinate system and rarely exceeds 1 or 2 arcsecond per year,

(ii) ANNUAL PARALLAX due to the earth's orbital motion which is periodic and even for the closest object does not exceed an amplitude of 0.7 arcsecond,

(iii) Lunisolar and planetary PRECESSION of the plane of the earth's orbit which is a secular drift of approximately 50 arcsecond per year,

(iv) NUTATION which is periodic of several arcsecond amplitude and has the same cause as (iii) above, and finally

(v) STELLAR ABERRATION, a periodic shift of 20 arcsecond dependent on the instantaneous direction of the earth's motion with respect to that of the light ray from the object.
This astronomical position is then transformed into topocentric co-ordinates by accounting for the earth's rotation and then

(vi) ATMOSPHERIC REFRACTION, which is a function of the zenith angle of the object and the environmental variables atmospheric pressure and temperature, is corrected.

Conventionally the reduction of the catalogued mean place (α_0,δ_0) to the apparent true place[@] at date (α_3'',δ_3'') proceeds by calculating the mean place at the nearest beginning of a Besselian year (α_2,δ_2) as an intermediate step, and expresses the apparent true place at date as an expansion in terms of that mean place; see Table A.1. The reduction is applied to a fixed star in such a manner that only two types of function occur: functions of co-ordinates (α_2,δ_2); and functions of the date in the form of the standard Day Numbers A, B, C, D, E (Besselian) or f, g, G, h, H, i (Independent), and the second order Day Numbers J and J', all of which are tabulated in references and ephemerides. Certain refinements incorp-

---

@ Note that an astronomical position may be TRUE or MEAN according to which equinox and equator is used as the reference frame, and APPARENT or ASTROMETRIC according to whether the aberration correction has been included or not. Although such terms as 'apparent true' are necessary here to distinguish the various cases, they are not in general astronomical usage.

orated in the method permit accurate reductions of mean to apparent place by a minimum amount of computation, and if necessary by manual methods; see for example Woolard and Clemence (1966), Porter and Sadler (1953) and the Explanatory Supplement to the Astronomical Ephemeris (1961).

Telescope pointing and various other astronomical tasks need an algorithm enabling a digital computer to perform the reduction without the onerous requirement that the machine have access to the tabulated Day Numbers. The machine could calculate the Day Numbers as required, but it can be shown that this is wasteful of time and less accurate than a method of direct calculation which is now described. This direct algorithm is very similar to that of Harris and Large (1967) to the extent that, although independently discovered, it post-dates the above and cannot be published. A comparison of the direct and conventional methods is clearly shown in Table A.1. In the direct algorithm the mean place at date is found directly, by applying proper motion and precession corrections. Aberration at date is then applied with the necessary parameters referred to the mean equinox and equator of date, and finally nutation at date is applied using the 13 most significant terms of the nutation series (all terms with coefficients greater than 0.01 arcsecond). The eventual result is the apparent true place at date and the only input data necessary are the mean co-ordinates and proper motions at epoch, and the Julian Dates of epoch and of date.

A DIRECT MACHINE ALGORITHM FOR MEAN TO APPARENT PLACE REDUCTION

Starting with the Julian Dates of epoch t_1 and of date t_3, the mean co-ordinates at epoch (α_0, δ_0) and the yearly proper motions μ_α, μ_δ calculate

$$\tau = (t_3 - t_1)/365.24220 \qquad \text{... A.1}$$

and so obtain the co-ordinates (α_1, δ_1) by

$$\left.\begin{aligned} \alpha_1 &= \alpha_0 + \mu_\alpha . \tau \\ \text{and} \quad \delta_1 &= \delta_0 + \mu_\delta . \tau \end{aligned}\right\} \text{... A.2}$$

Transform (α_1, δ_1) to rectangular co-ordinates (x_1, y_1, z_1) by

$$\left.\begin{aligned} x_1 &= \cos\alpha_1 \cos\delta_1 \\ y_1 &= \sin\alpha_1 \cos\delta_1 \\ z_1 &= \sin\delta_1 \ . \end{aligned}\right\} \text{... A.3}$$

Calculate time intervals T_0 and T in tropical centuries by

$$\left.\begin{aligned} T_0 &= (t_1 - 2415020.313)/36524.220 \\ T &= (t_3 - t_1)/36524.220 \end{aligned}\right\} \text{... A.4}$$

TABLE A.1

Conventional method	Method suitable for direct machine calculation
$\alpha_0\delta_0$ Mean place at epoch t_1 referred to mean equinox + equator of t_1.	$\alpha_0\delta_0$ mean place at epoch t_1 ....
$\downarrow$ proper motion using μ_α, μ_δ for t_1 applied from t_1 to t_2 (BYR).	$\downarrow$ proper motion to date using μ_α, μ_δ for t_1 applied from t_1 to date t_3.
$\alpha_1\delta_1$ mean co-ordinates at epoch t_2 (BYR) referred to equinox + equator of t_1.	$\alpha_1\delta_1$ mean co-ordinates at date t_3 referred to equinox + equator of t_1.
$\downarrow$ precession from t_1 to epoch t_2 (BYR).	$\downarrow$ precession from epoch t_1 to date t_3 directly.
$\alpha_2\delta_2$ mean place at epoch t_2 (BYR) referred to mean equinox + equator of t_2.	
$\downarrow$ aberration at date t_3, using parameters referred to mean equinox + equator of t_2 (BYR), and proper motion from t_2 to date t_3.	
$\alpha_2'\delta_2'$ apparent co-ordinates at date t_3 referred to mean equinox + equator of t_2 (BYR).	$\alpha_3\delta_3$ mean place at date t_3 referred to mean equinox + equator of t_3.
$\downarrow$ precession from epoch t_2 (BYR) to date t_3.	$\downarrow$ aberration at date t_3 using parameters referred to mean equinox of date.
$\alpha_3'\delta_3'$ apparent mean place at date t_3 referred to mean equinox + equator of t_3.	$\alpha_3'\delta_3'$ apparent mean place at date t_3 ....
$\downarrow$ nutation at date t_3.	$\downarrow$ nutation at date t_3.
$\alpha_3''\delta_3''$ apparent true place at date t_3 referred to true equinox + equator of t_3.	$\alpha_3''\delta_3''$ apparent true place at date t_3 ....

LEGEND FOR TABLE A.1 (and text).

SYMBOLS

α = right ascension

δ = declination

x,y,z = equatorial rectangular co-ordinates.

SUBSCRIPTS indicate the equinox and equator to which the co-ordinate is referred.

0,1 for equinox and equator of initial epoch t_1

2 for equinox and equator of intermediate epoch t_2, (nearest beginning of Besselian year)

3 for equinox and equator of t_3, the date of observation.

PRIMES

Astrometric co-ordinates are not primed,
apparent mean co-ordinates have single primes,
apparent true co-ordinates have double primes.

UNITS

All angles are in radian measure,
all times t_1, t_2, t_3 etc. are in Julian Date form,
sidereal times θ, θ_g, θ_{gm} etc. are in radian.

epoch t_1 = epoch of catalogue,

t_2 = beginning of Besselian year closest to date (abbreviated BYR),

t_3 = date of observation.

and thence the precessional parameters ζ_0, z and θ by

$$\left.\begin{aligned} \zeta_0 &= (1.11713192^{-2} + 6.768^{-6}\,T_0)T + 1.464^{-6}\,T^2 + 8.7^{-8}\,T^3, \\ z &= \zeta_0 + 3.835^{-6}\,T^2, \\ \text{and} \quad \theta &= (9.7189726^{-3} - 4.146^{-6}\,T_0)T - 2.065^{-6}\,T^2 - 2.04^{-7}\,T^3. \end{aligned}\right\} \quad \ldots \text{ A.5}^{@}$$

Calculate the precessional rotation matrix elements:

$$\left.\begin{aligned} S_{11} &= \cos\zeta_0.\cos\theta.\cos z - \sin\zeta_0.\sin z, \\ S_{12} &= -\sin\zeta_0.\cos\theta.\cos z - \cos\zeta_0.\sin z, \\ S_{13} &= -\sin\theta\,\cos z, \\ S_{21} &= \cos\zeta_0.\cos\theta.\sin z + \sin\zeta_0.\cos z, \\ S_{22} &= -\sin\zeta_0.\cos\theta\,\sin z + \cos\zeta_0.\cos z, \\ S_{23} &= -\sin\theta.\sin z, \\ S_{31} &= \cos\zeta_0.\sin\theta, \\ S_{32} &= -\sin\zeta_0.\sin\theta, \\ \text{and} \quad S_{33} &= \cos\theta. \end{aligned}\right\} \quad \ldots \text{ A.6}$$

Calculate the mean (rectangular) co-ordinates at date (x_3,y_3,z_3) by

$$(x_3,y_3,z_3) = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad \ldots \text{ A.7}$$

Unlike precession (or nutation), stellar aberration does not merely involve a simple rotation of the co-ordinate frame. We set the aberration constant k to

$$k = 9.93674^{-5} \quad (20.496 \text{ arcsecond}), \quad \ldots \text{ A.8}$$

and calculate time interval T in Julian centuries by

$$T = (t_3 - 2415020.0)/36525.0\ . \quad \ldots \text{ A.9}$$

Calculate the geometric mean longitude of the sun L, the mean longitude of the solar perigee π, the solar mean anomaly M, the mean obliquity of the ecliptic ε_0 and the eccentricity of the solar orbit e by

$$\begin{aligned} L &= 4.881627938 + 628.3319510\,T + 5.2796^{-6}\,T^2, \\ \pi &= 4.908229463 + 3.000526416^{-2}\,T + 7.9025^{-6}\,T^2 + 5.82^{-8}\,T^3, \end{aligned}$$

@ Note superscripts appended to figures are decimal exponents, e.g. $6.768^{-6} = 6.768 \times 10^{-6}$.

$$\left.\begin{aligned} M &= 6.256583784 + 628.3019457\ T - 2.6180^{-6}\ T^2 - 5.82^{-8}\ T^3, \\ \varepsilon_0 &= 4.093197474^{-1} - 2.2711097^{-4}\ T - 2.86^{-8}\ T^2 + 8.8^{-9}\ T^3, \\ \text{and } e &= 1.675104^{-2} - 4.180^{-5}\ T - 1.26^{-7}\ T^2. \end{aligned}\right\} \ldots \text{A.10}$$

Calculate the geometric elliptic longitude of the sun $\odot$ using the 'equation of centre' approximated by

$$(\odot - L) = (2e - e^3/4)\cdot\sin M + (5/4)e^2\cdot\sin 2M + (13/12)e^3\cdot\sin 3M \ \ldots \text{A.11}$$

Calculate the aberration Day Numbers C and D by

$$\left.\begin{aligned} C &= -k(\cos\odot.\cos\varepsilon_0 + e\cos\pi.\cos\varepsilon_0)\ , \\ \text{and } D &= -k(\sin\odot + e\sin\pi)\ . \end{aligned}\right\} \ldots \text{A.12}$$

The terms in e in equations A.12 (the elliptic or 'E-terms') are approximately constant for fixed objects such as stars, and are included in the mean place of the object in the catalogue; for the majority of applications they should thus be ommitted from the expression for C and D. Moving objects such as planets require the E-terms to be included, and Scott (1964) discusses their inclusion for very precise stellar reductions. Aberration caused by the diurnal rotation of the earth amounts to about 0.3 arcsecond in amplitude but is not considered here. The rectangular mean co-ordinates at date (x_3', y_3', z_3') are obtained as in Woolard and Clemence (1966) and Scott and Hughes (1964), from the equations

$$\left.\begin{aligned} x_3' &= x_3 - D\ , \\ y_3' &= y_3 + C\ , \\ z_3' &= z_3 + C\tan\varepsilon_0\ . \end{aligned}\right\} \ldots \text{A.13}$$

To obtain the nutation correction use T defined in equation A.9 to calculate the nutation arguments

$$\left.\begin{aligned} l &= 5.168000340 + 8328.691103\ T + 1.60425^{-4}\ T^2 + 2.51^{-7}\ T^3, \\ l' &= 6.256583574 + 628.3019457\ T - 2.618^{-6}\ T^2 - 5.8^{-8}\ T^3, \\ F &= 0.1963650568 + 8433.466291\ T - 5.6045^{-5}\ T^2 - 5.8^{-9}\ T^3, \\ D &= 6.121523940 + 7771.377193\ T - 2.5065^{-5}\ T^2 + 3.3^{-8}\ T^3, \\ \text{and } \Omega &= 4.523601514 - 33.75714624\ T + 3.6264^{-5}\ T^2 + 3.9^{-8}\ T^3. \end{aligned}\right\} \ldots \text{A.14}$$

Calculate the nutation in longitude $\Delta\psi$ and in obliquity $\Delta\varepsilon$ from the standard series (only terms with coefficients greater than 0.01 arcsecond are included here)

$$\begin{aligned} \Delta\psi = & -(8.35465^{-5} + 8.421^{-8}\ T)\ \sin\Omega + 1.0123^{-6}\ \sin 2\Omega \\ & - 6.1712^{-6}\ \sin(2F-2D+2\Omega) + 6.1135^{-7}\ \sin(l') \\ & - 2.41^{-7}\ \sin(l'+2F-2D+2\Omega) + 1.04^{-7}\ \sin(2F-2D+2\Omega-l') \end{aligned}$$

$$+\ 6.01^{-8}\sin(2F-2D+\Omega). \qquad \ldots\ A.15$$

$$\Delta\varepsilon = (4.4651^{-5} + 4.4^{-9}T)\cos\Omega - 4.383^{-7}\cos 2\Omega$$
$$+\ 2.677^{-6}\cos(2F-2D+2\Omega) + 1.047^{-7}\cos(l'+2F-2D+2\Omega) \qquad \ldots\ A.16$$

The short period nutation terms (terms with periods of less than 35 days) are given by the further series

$$\Delta\psi = \Delta\psi - 9.876^{-7}\sin(2F+2\Omega) + 3.272^{-7}\sin l$$
$$-\ 1.658^{-7}\sin(2F+\Omega) - 1.265^{-7}\sin(l+2F+2\Omega)$$
$$-\ 7.22^{-8}\sin(l-2D) + 5.53^{-8}\sin(2F+2\Omega-l) \qquad \ldots\ A.17$$

in longitude, and

$$\Delta\varepsilon = \Delta\varepsilon + 4.286^{-7}\cos(2F+2\Omega) + 8.87^{-8}\cos(2F+\Omega)$$
$$+\ 5.48^{-8}\cos(2F+2\Omega+l) \qquad \ldots\ A.18$$

in obliquity. Using A.10, A.16 and A.18 calculate the true obliquity ε from

$$\varepsilon = \varepsilon_0 + \Delta\varepsilon, \qquad \ldots\ A.19$$

and the nutation rotation matrix elements:

$$\left.\begin{aligned}
S_{11} &= \cos\Delta\psi, \\
S_{12} &= -\sin\Delta\psi.\cos\varepsilon_0, \\
S_{13} &= -\sin\Delta\psi.\sin\varepsilon_0, \\
S_{21} &= \sin\Delta\psi.\cos\varepsilon, \\
S_{22} &= \cos\Delta\psi.\cos\varepsilon_0.\cos\varepsilon + \sin\varepsilon_0.\sin\varepsilon, \\
S_{23} &= \cos\Delta\psi.\sin\varepsilon_0.\cos\varepsilon - \cos\varepsilon_0.\sin\varepsilon, \\
S_{31} &= \sin\Delta\psi.\sin\varepsilon, \\
S_{32} &= \cos\Delta\psi.\cos\varepsilon_0.\sin\varepsilon - \sin\varepsilon_0.\cos\varepsilon, \\
\text{and } S_{33} &= \cos\Delta\psi.\sin\varepsilon_0.\sin\varepsilon + \cos\varepsilon_0.\cos\varepsilon.
\end{aligned}\right\} \qquad \ldots\ A.20$$

The (rectangular) apparent true co-ordinates of date (x_3'', y_3'', z_3'') are given by the matrix equation

$$(x_3'', y_3'', z_3'') = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix} \cdot \begin{bmatrix} x_3' \\ y_3' \\ z_3' \end{bmatrix} \qquad \ldots\ A.21$$

Finally transform (x_3'', y_3'', z_3'') to equatorial polar co-ordinates (α_3'', δ_3'') by using the equations

$$\sin(\alpha_3'') = y_3''/(x_3''^2 + y_3''^2)^{\frac{1}{2}},$$

A.8

and $\cos(\alpha_3'') = x_3''/(x_3''^2 + y_3''^2)^{\frac{1}{2}}$, ... A.22a

where α_3'' lies in the interval 0 to 2π, and

$$\tan(\delta_3'') = z_3''/(x_3''^2 + y_3''^2)^{\frac{1}{2}} , \qquad \text{... A.22b}$$

where δ_3'' lies in the interval $-\pi/2$ to $+\pi/2$.

TOPOCENTRIC CALCULATIONS RELEVANT TO TELESCOPE POINTING

Except for parallax which, if required, is most easily included with the aberration calculation (see Explanatory Supplement to the Astronomical Ephemeris 1961), equations A.1 to A.22 give all the necessary astronomical corrections; the transformation to topocentric co-ordinates requires the calculation of the local mean sidereal time θ as follows.
Truncate the Julian Date t_3 to an integral value and add 0.5 to obtain t_{gm} a Julian Date corresponding to a Greenwich midnight, and calculate time interval T in Julian centuries by

$$T = (t_{gm} - 2415020.0)/36525.0 \qquad \text{... A.23}$$

Using

$$\partial\theta/\partial t = 6.30038749 \quad \text{(radian/day)} \qquad \text{... A.24}$$

as the sidereal rate, calculate the Greenwich sidereal time at t_{gm} and at t_3 by

$$\theta_{gm} = 1.739935893 + 628.3319510\ T + 6.7558^{-6}\ T^2 \qquad \text{... A.25}$$

and $\theta_g = \theta_{gm} + (t_3 - t_{gm}).\partial\theta/\partial t$, ... A.26

respectively. Finally, from the local east longitude λ_e calculate θ, the local mean sidereal time by

$$\theta = \theta_g + \lambda_e . \qquad \text{... A.27}$$

Should apparent sidereal time be required it is given by

$$\theta_{apparent} = \theta + \Delta\psi\cos\varepsilon , \qquad \text{... A.28}$$

where θ is obtained from A.27, and $\Delta\psi\cos\varepsilon$ is the 'equation of the equinoxes' and can be obtained from A.15, A.17 and A.19. Since observatories are equipped with sources of mean sidereal time (see for example Appendix C), it would be most appropriate to use as telescope pointing co-ordinates the apparent true declination but the apparent mean right ascension, in conjunction with the mean sidereal time. A proposal of this kind was put forward by Atkinson and Sadler (1951) but unfortunately was never adopted, and so here the topocentric co-ordinates declination δ, and hour angle H are obtained by:

$$\left.\begin{aligned} \delta &= \delta_3'' , \\ \text{and} \quad H &= \theta_{\text{apparent}} - \alpha_3'' . \end{aligned}\right\} \quad \ldots \text{A.29}$$

The refraction correction used is quite standard and is taken from Woolard and Clemence (1966). Given the barometric pressure B in millimetres of mercury and the temperature t in degrees Celsius the refraction index μ_0 is calculated from

$$\mu_0 = 1 + 1.0534^{-4}.B/(273+t) , \quad \ldots \text{A.30}$$

and atmospheric height parameter H_0 is taken as

$$H_0 = 1.2541^{-3} . \quad \ldots \text{A.31}$$

If ϕ is the local geodetic latitude, the zenith distance Z of an object is obtained from the equation

$$\cos Z = \cos\phi.\cos\delta.\cos H + \sin\phi \sin\delta , \quad \ldots \text{A.32}$$

and the refraction coefficient R calculated by

$$R = (\mu_0-1).(1-H_0).\tan Z - (\mu_0-1).\left[H_0-\tfrac{1}{2}(\mu_0-1)\right].\tan^3 Z. \quad \ldots \text{A.33}$$

Resolved into topocentric declination and hour angle the refraction correction becomes

$$\left.\begin{aligned} \delta &= \delta + R(\sin\phi.\cos\delta - \cos\phi.\sin\delta.\cos H)/ \sin Z, \\ \text{and} \quad H &= H - R \cos\phi \sin H/(\cos\delta \sin Z). \end{aligned}\right\} \quad \ldots \text{A.34}$$

δ and H so calculated are our telescope pointing angles.

CONCLUSION

The direct mean to apparent place reduction algorithm presented above has an accuracy of better than 0.1 arcsecond (even close to the celestial poles); an absolute worst case error of 0.077 arcsecond can result from the omission of the nutation terms of amplitude less than 0.01 arcsecond. A FORTRAN IV implementation of the algorithm with 56 bit precision on an I.B.M. 360/50 computer was tested against a number of FK4 stars in A.P.F.S. The worst difference noted was 0.06 arcsecond, but it should be noted that reduction of the machine precision to 24 bits produced differences as large as 0.4 arcsecond due to numerical error propagation. Harris and Large (1967) quote a similar accuracy for their method but they include all terms in the nutation series and the accuracy in their case appears to be limited by a less precise form of the aberration correction than the one given here. A polar co-ordinate version of the direct algorithm was also tested and although about 20% slower the discrepancies between it and the rectangular one given were quite small (typically 2×10^{-4} arcsecond). A breakdown of the timing for the I.B.M. implementation of the rectangular version is given in Table A.2; the Univac U 1108 version used in the body of the thesis and listed in Appendix D is considerably faster.

TABLE A.2

Polar to Rectangular Co-ordinate Transformation:	4.2 ms
Precession Correction:	7.6 ms
Aberration Correction:	11.2 ms
Nutation Correction:	16.8 ms
Rectangular to Polar Co-ordinate Transformation:	2.8 ms
TOTAL	42.6 ms

APPENDIX B
Anti-ambiguity Requirements of Linked Shaft Encoders

Despite widely varying pointing accuracies, the precision with which telescopes are pointed is usually an arcsecond in declination and a decisecond of time (equal to 1.5 arcsecond about the axis) in hourangle@. Taking the more critical case of declination, a resolution of one part in 1,296,000 is required and, although single turn shaft encoders with resolutions of 2^{20} (1 part in 1,048,576), and 2^{21} (1 part in 2,097,152) are commercially available, they are rarely employed in practice. Even incremental encoders of such high resolution are extremely expensive@@, and it is not possible to drive them in a manner which realizes an accuracy commensurate with the resolution, since wind-up errors in flexible-disc or bellows couplings are typically tens of arcsecond. It is unlikely that a readout precision (and accuracy) of an arcsecond can be obtained directly from the axis without designing the axis encoder as an integral part of the bearing and structure; hence it is accepted practice to employ encoders of lesser resolution (e.g. 2^{15}), geared to the axis with gears of as high a quality available.

No problem exists with an incremental system, but if (in the interests of consistent accuracy and avoidance of sudden loss of position with power failures, noise etc.) an absolute encoder is used, a means of counting complete turns, or removing the ambiguity from the readout is necessary. For a completely absolute system this is most easily done with a second encoder, which may be an inexpensive, coarse brush type, geared 1:1 with the axis. However, the available encoder resolutions and gear ratios are rarely such that the output is exactly the number of turns of the fine encoder, nor even related to its resolution, and the angular jitter between the two encoders further complicates the computation of the revolution number. Analogous problems would occur within a shaft encoder when bits change out of sequence, but are avoided by employing monostrophic codes@@@, or by using duplicated tracks in phase quadrature and various means of lead/lag brush or read-head selection. These techniques could in principle

---

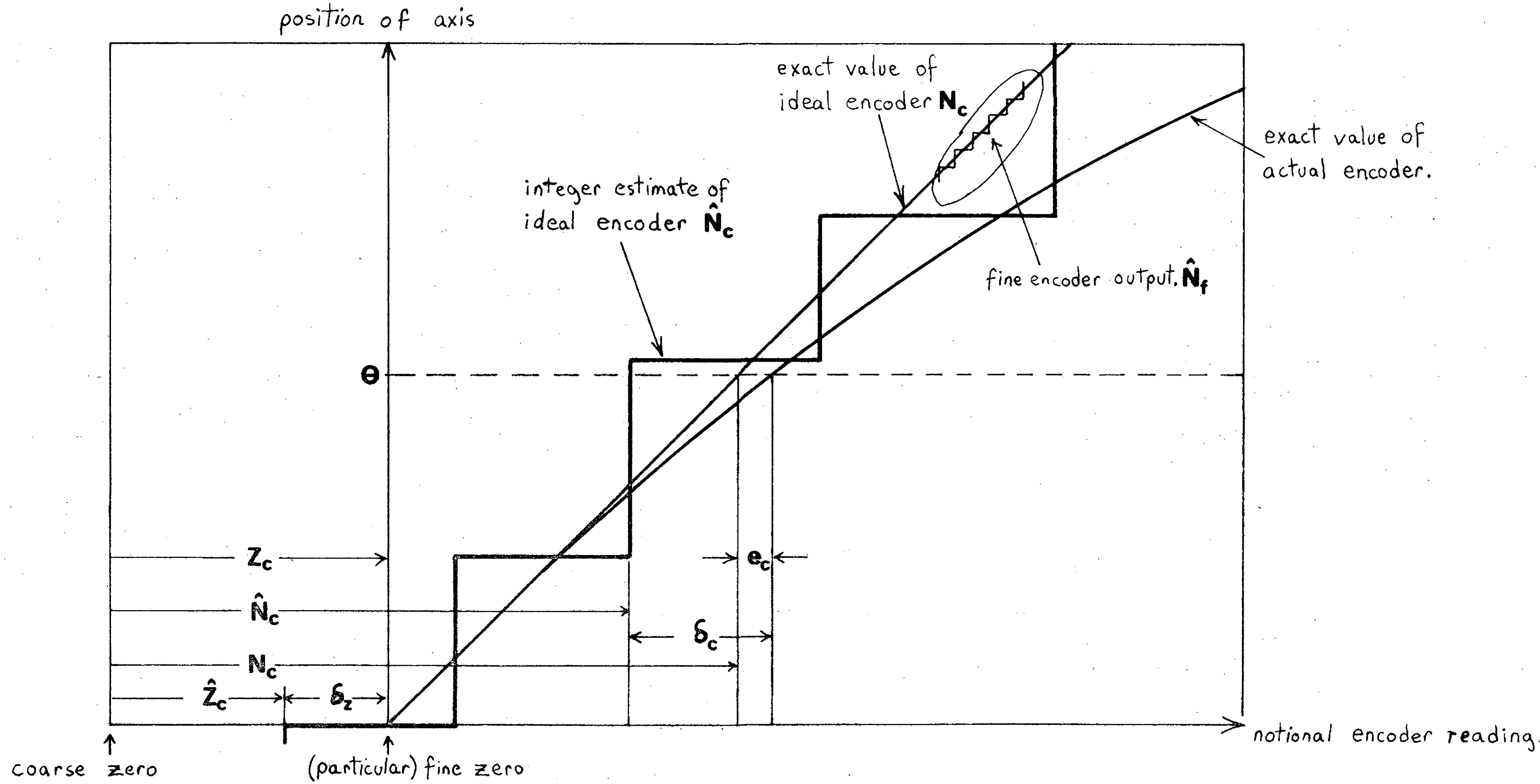
@ 10 arcsecond and 1 second of time respectively are common for very small instruments, but future instruments will doubtlessly require a precision even greater than the above.

@@ for example U.S.$19,000 for a 2^{20} incremental encoder compared to U.S.$2,000 each for the 2^{15} absolute units used in the Mt. Stromlo Observatory 74-inch encoder system (see Appendix C).

@@@ codes in which only one bit of the word changes in going from i to i+1.

FIG B.1

DEFINITION OF QUANTITIES ASSOCIATED WITH THE COARSE ENCODER.



be applied to physically separate shaft encoders, but, since special encoders would be required, are rarely used. Here we consider the anti-ambiguity requirements of two gear-linked absolute encoders, and the nature and extent of errors in such systems.

Consider coarse and fine encoders of resolution R_c and R_f respectively, geared together with a speed ratio $T_f:T_c$. The coarse encoder is assumed to be 1:1 with the axis concerned, and, although T_c and T_f can be thought of as the number of teeth on the mating gears attached to the coarse and fine encoder shafts respectively, T_c and T_f are assumed to be relatively prime, and, of course, $T_c/T_f > 1$. Let the theoretically exact reading of the fine encoder be the real quantity N_f, then if n is the number of complete fine encoder revolutions from a specified fine encoder zero, a general position θ of the axis, measured in fine bits from that zero, is given by

$$\theta = n R_f + N_f . \qquad \text{... B.1}$$

If we consider the coarse encoder to be comprised of a perfectly linear unit giving exact reading N_c, and differing from that in practice by the total error e_c, where e_c includes all the gear, code disk and drive eccentricity errors, and if Z_c is the (exact) reading of this ideal encoder at the above fine encoder zero, then θ is also given by

$$\theta = (N_c - Z_c)\, T_c\, R_f \,/\, R_c\, T_f . \qquad \text{... B.2}$$

The numbers physically read from the encoders are integer estimates of N_c+e_c and N_f, denoted here by $\widehat{N}_c$ and $\widehat{N}_f$ respectively, and differ from N_c+e_c and N_f by the truncation errors δ_c, δ_f defined by

$$\left.\begin{aligned} \delta_c &= N_c - \widehat{N}_c + e_c , \\ \text{and} \quad \delta_f &= N_f - \widehat{N}_f , \end{aligned}\right\} \qquad \text{... B.3}$$

Likewise, if we set the axis to the fine zero referred to above and read the coarse encoder, we obtain integer $\widehat{Z}_c$ which is related to Z_c by

$$Z_c = \widehat{Z}_c + \delta_z , \qquad \text{... B.4}$$

where the truncation error δ_z is constant (once we have defined the zero point), and may be taken to include the value of e_c at the zero point. The various quantities are defined in Figure B.1, and, from the equations above, it can be seen that the integer n is given exactly by

$$\begin{aligned} n &= \frac{(N_c - Z_c)\, T_c}{R_c\, T_f} - \frac{N_f}{R_f} \\ &= \frac{(\widehat{N}_c - \widehat{Z}_c - e_c + \delta_c - \delta_z)\, T_c}{R_c\, T_f} - \frac{(\widehat{N}_f + \delta_f)}{R_f} . \end{aligned} \qquad \text{... B.5}$$

The computer or arithmetic hardware interfaced to the pair of encoders must contain a routine for calculating $\hat{n}$, an estimate of n; thence the estimate $\hat{\theta}$ of the position of the axis (in fine bits) is calculated by

$$\hat{\theta} = \hat{n} R_f + \hat{N}_f . \qquad \ldots \text{B.6}$$

$\hat{\theta}$ is the required axis position subject only to the usual digitization error of ±1 fine bit, and, of course, the accuracy of the fine encoder, and may be scaled to convenient units and a convenient axis zero point as desired[@]. Here we are concerned with the possible error in calculating n given by

$$\Delta n = n - \hat{n} , \qquad \ldots \text{B.7}$$

and with assessing just how much latitude is allowed by various methods of calculating $\hat{n}$.

<u>Method A.</u>

The simplest method of calculating $\hat{n}$, is to perform an integer division of the (integer) quantity $(\hat{N}_c - \hat{Z}_c).T_c$ by $R_c\ T_f$, that is, by calculating

$$\hat{n} = \text{INT}\left\{\frac{(\hat{N}_c - \hat{Z}_c).T_c}{R_c\ T_f}\right\} \qquad \ldots \text{B.8}$$

where the function INT denotes truncation to an integer. Since the remainder of this division must be integral and lie in the interval $\left[0,\ R_c T_f - 1\right]$, we obtain the inequality

$$0 \leqslant (\hat{N}_c - \hat{Z}_c).T_c - R_c.T_f.\hat{n} \leqslant R_c\ T_f - 1 \quad , \qquad \ldots \text{B.9}$$

which, by equations B.3 and B.5, gives

$$0 \leqslant R_c\ T_f \Delta n + (e_c - \delta_c + \delta_z).T_c + N_f.R_c.T_f / R_f \leqslant R_c.T_f - 1 \qquad \ldots \text{B.10}$$

It should be noted that the difference between the quantities N_c and $\hat{N}_c$ is a linear function of N_f. That is,

$$\delta_c - e_c = \delta_a + N_f.R_c.T_f / R_f\ T_c , \qquad \ldots \text{B.11}$$

where, for a particular configuration of encoders and zero setting, δ_a is a constant, and the argument is clarified by reference to Figure B.1. Using equation B.11, the inequality in equation B.10 becomes

$$0 \leqslant R_c\ T_f \Delta n + (\delta_z - \delta_a).T_c \leqslant R_c\ T_f - 1 \quad . \qquad \ldots \text{B.12}$$

If we desire no error in the calculated antiambiguity number $\hat{n}$, we can set

---

@ It is often simpler to perform this scaling on the raw quantities $\hat{N}_f$, $\hat{N}_c$ etc. before computing the antiambiguity number $\hat{n}$.

$\Delta n = 0$, and obtain the approximate inequality

$$0 \leqslant \delta_z - \delta_a < X , \qquad \text{... B.13}$$

where $X = R_c . T_f \,/\, T_c$ is the number of coarse bits in a complete revolution of the fine encoder. Finally, using equation B.11, and with due regard to the intervals in which the various quantities lie, we have (approximately)

$$-X < e_c < X - 4 \qquad \text{... B.14}$$

as the bound on the total permissable error e_c of the coarse encoder, and a permissable range for e_c of 2X-4 coarse bits.

In the case of the Mt. Stromlo 74-inch encoder installation (see Appendix C), the declination axis is the more critical with regard to this error, and since the relevant quantities are $R_c = 256$, $T_f = 4$ (32), and $T_c = 189$ (1512)@, X in equations B.15, B.16 has the value 5.42, and the coarse encoder error has a permissable range of 6.8 coarse bits. The argument has assumed that the function INT in equation B.8 involves truncation, and that nothing at all is known about the quantity δ_z other than that it lies in the range -1 to +1. If, in equation B.8, a numerical procedure which <u>rounds</u> rather than truncates is used, the error e_c has an identical range but is now distributed symmetrically, that is

$$2 - X < e_c < X - 2 \,. \qquad \text{... B.15}$$

If δ_z is known and the coarse zero point $\hat{Z}_c$ adjusted accordingly, the range can, in certain cases, be extended by as much as 2 coarse bits, but this is rarely practicable.

<u>Method B.</u>

An alternative, (and more involved) means of calculating the anti-ambiguity number $\hat{n}$, is to perform the integer division of $(\hat{N}_c - \hat{Z}_c) T_c R_f - \hat{N}_f R_c T_f$ by $R_c . T_f R_f$ to obtain

$$\hat{n} \;=\; \mathrm{INT} \left\{ \frac{(\hat{N}_c - \hat{Z}_c) . T_c R_f \;-\; \hat{N}_f R_c T_f}{R_c T_f R_f} \right\} . \qquad \text{... B.16}$$

Similarly to Method A, the constraints on the integer remainder of this division imply that

$$0 \leqslant R_c T_f R_f \Delta n + (e_c - \delta_c + \delta_z) . T_c R_f + \delta_f R_c T_f \leqslant R_c T_f R_f - 1 \qquad \text{... B.17}$$

@ Figures in parenthesis are the actual numbers of teeth on the respective gears.

Setting $\Delta n = 0$ and by neglecting δ_f the error in the fine encoder (which is very small compared to the coarse errors), we obtain the approximate relation

$$0 \leqslant e_c - \delta_c + \delta_z < X \ . \qquad \ldots \text{B.18}$$

Finally, the bounds on the error e_c are given by

$$2 < e_c < X-1 \ , \qquad \ldots \text{B.19}$$

and the permissable range of error is X-3. For the case of the declination axis cited above, this implies a range for e_c of 2.4 coarse bits. Again the range is symmetrical about zero if rounding rather than truncation is used in equation B.16.

Thus the simpler of the two methods is, in fact, the more tolerant of angular jitter errors between the gear linked encoders. The reason for this is that, in Method A, the ommission of a term in $\hat{N}_f$, in equation B.8, actually simulates the effect of the truncation process involved in the relation between quantities $\hat{N}_c$ and N_c. The method originally incorporated into the Mt. Stromlo Encoder and Timing System (E.T.S.) software@ was, in a sense, intermediate between the two, since it employed Method A but inspected the two most significant bits of $\hat{N}_f$, and, if $\hat{N}_f < R_f/4$, it incremented $\hat{n}$. The U1108 telescope pointing data processor TA.MAIN used the 'brute-force' technique of calculating $\hat{n}$ as per Method A, trying the integers from $\hat{n}-2$ to $\hat{n}+2$, and accepting the one which gave the minimum residual R_n, where

$$R_n = \left| nR_f + \hat{N}_f - (\hat{N}_c - \hat{Z}_c)\, T_c\, R_f \,/\, R_c\, T_f \right| \ . \qquad \ldots \text{B.20}$$

The value of the residual R_n was recorded by TA.MAIN, permitting an interesting assessment of the 74-inch E.T.S. reliability. The maximum angular 'jitter' between the coarse and fine encoders noted for the 148 observations made in March 1973, amounted to ±1.34 coarse bits in hourangle, and ±1.08 in declination. Since one bit must be allowed for the coarse encoder quantization error, it is clear that the design of the 74-inch E.T.S. has been reasonably conservative.

@ by Ron Howe; Mt. Stromlo Observatory.

APPENDIX C.

74-inch Telescope Timing and Attitude Readout System

The Mt. Stromlo Observatory 74-inch Grubb Parsons telescope is a general purpose English crossed-axis mounted instrument and was installed in 1951. Its focal stations comprise a four-mirror Coudé of focal ratio f/31, a Cassegrain of f/18 and a (now rarely used) f/5 Newtonian. Radial roller bearings on both piers and a ball thrust bearing on the north pier support the polar axis which is driven by a large spur gear for slewing purposes and a clutch-connected wormwheel of 720 teeth for tracking and fine motion. The declination axis is supported by a pair of axially preloaded taper roller bearings in the cube section where the axes intersect, and has a fast motion drive similar to the polar axis. Fine motion in declination is by means of an 'A-frame' which can be clamped to a drive ring on the cube, and its apex driven with respect to the tube by a screw driven by a DC servo motor. A velocity servo is fed into one input of a differential driving the main polar worm, and a DC servo motor into the other. The two DC servo motors mentioned were initially used for manual guiding adjustments but are now part of an automatic guiding servo which can accept input signals from a Coudé autoguider head as well as manual guide commands.

The original attitude readout system for the 74-inch comprises three selsyn transmitters geared up with different ratios onto large instrument spur gears on each axis. In declination the transmitters feed passive selsyn receivers on the telescope control panel, and, to accomodate the two cases of using the telescope east or west of the polar axis, the declination receiver dials are equipped with rotatable masks which effect a change in the labelling of the finer two of the three dials. The polar axis readout is similar except that the transmitter signals feed the receivers via following transformers which perform the analogue subtraction of sidereal time. Sidereal time is displayed on a similar group of three dials. The readout error in this system can exceed an arcminute or so and is caused firstly by the use of passive receivers (which are inherently prone to stiction), and also by excessive backlash in the transmitter gearing; the two coarser transmitters are geared down again from the pinion driving the fine one, and so all three dials of each group are affected. The recently installed digital readout system about to be described is now used for all routine setting and positioning of the telescope, the old selsyn system proving useful as a back-up in case of trouble and for maintenance purposes.

AXIS ENCODER INSTALLATION DETAILS

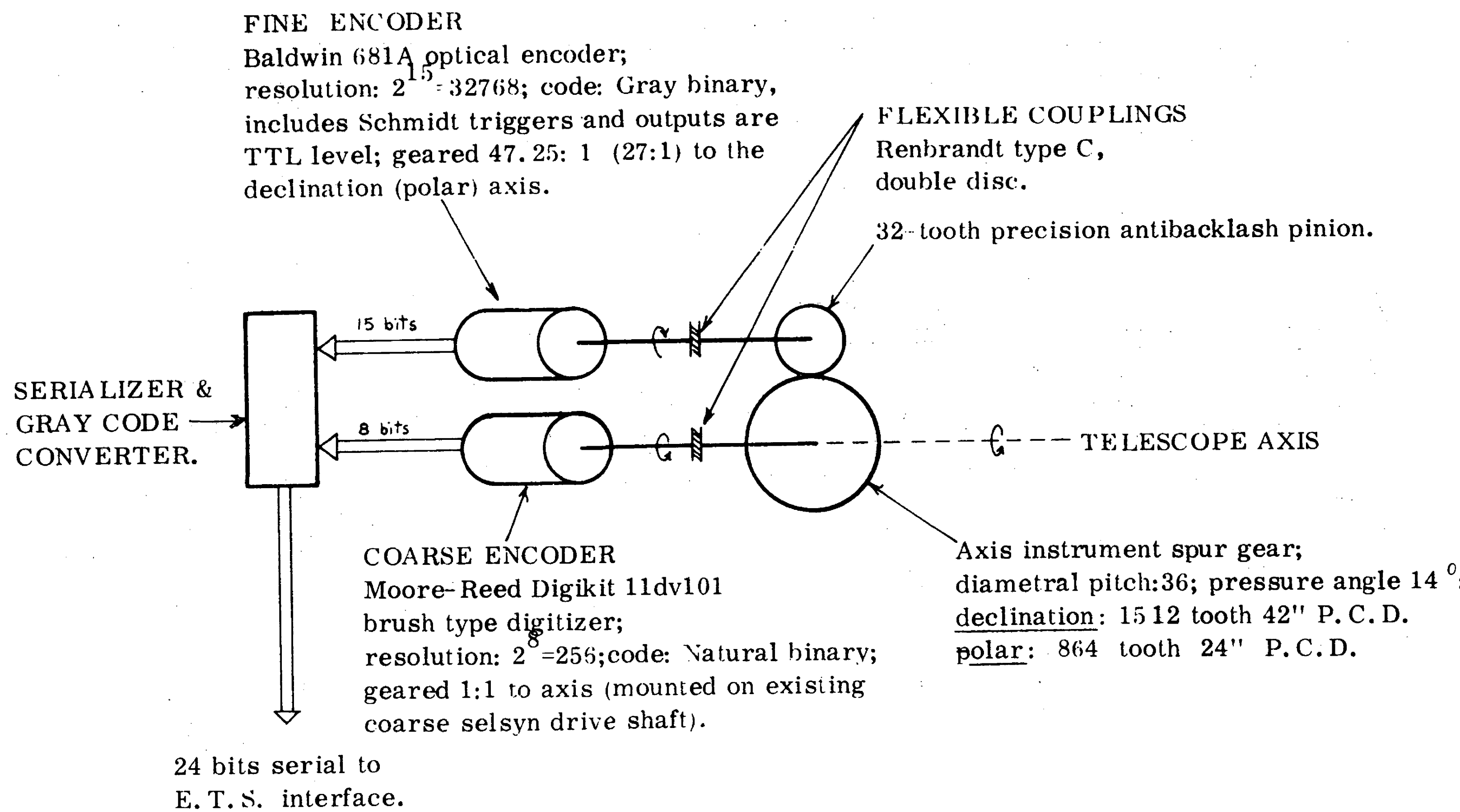
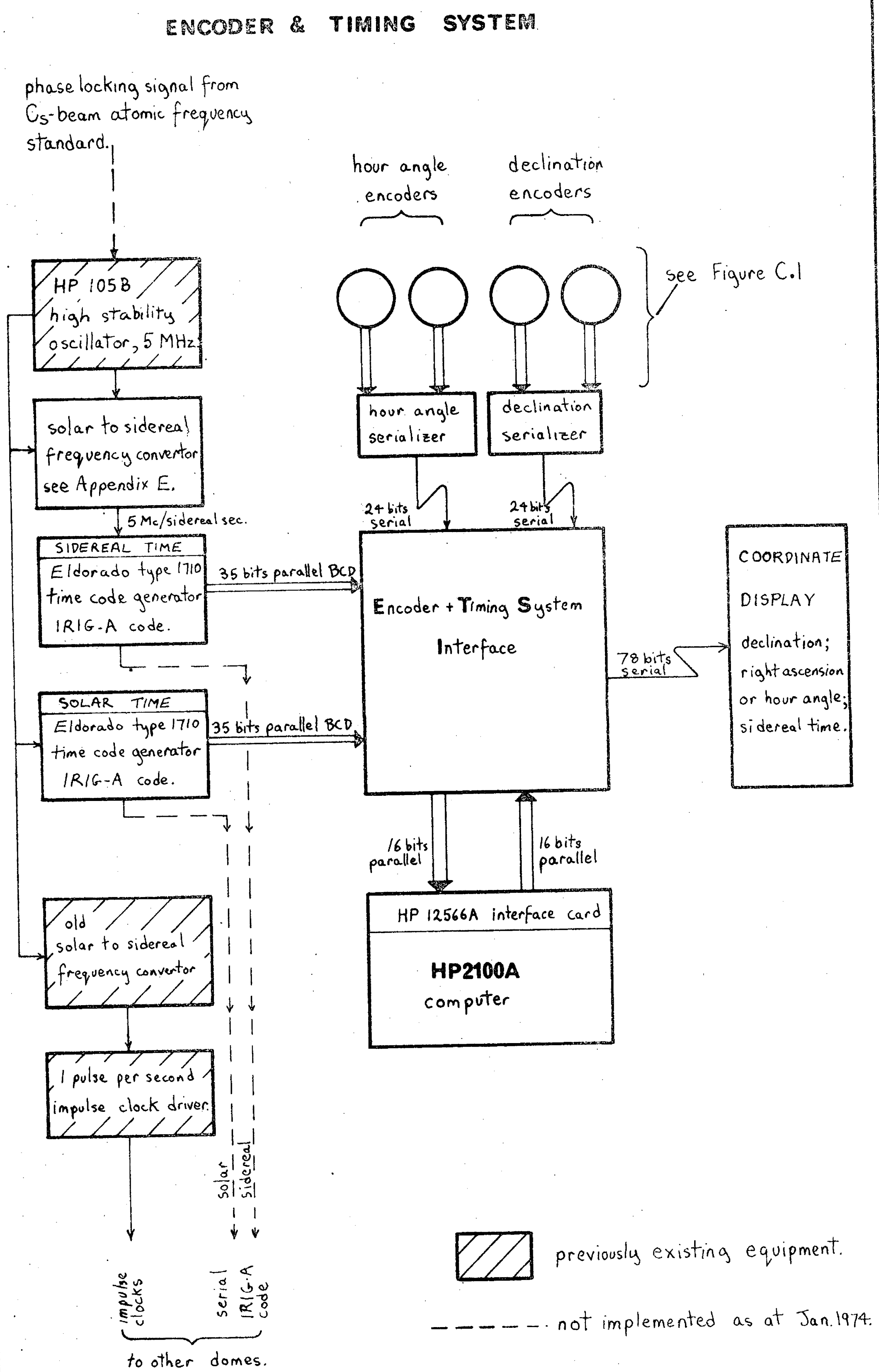


FIG C.1

FIG C.2


ENCODER & TIMING SYSTEM
phase locking signal from Cs-beam atomic frequency standard.
HP 105B high stability oscillator, 5 MHz
solar to sidereal frequency convertor see Appendix E.
5 Mc/sidereal sec.
SIDEREAL TIME
Eldorado type 1710 time code generator IRIG-A code.
35 bits parallel BCD
SOLAR TIME
Eldorado type 1710 time code generator IRIG-A code.
35 bits parallel BCD
old solar to sidereal frequency convertor
1 pulse per second impulse clock driver.
solar
sidereal
impulse clocks
serial IRIG-A code
to other domes.
hour angle encoders
declination encoders
see Figure C.1
hour angle serializer
declination serializer
24 bits serial
24 bits serial
Encoder + Timing System Interface
78 bits serial
COORDINATE DISPLAY
declination; right ascension or hour angle; sidereal time.
16 bits parallel
16 bits parallel
HP 12566A interface card
HP2100A computer
previously existing equipment.
not implemented as at Jan. 1974.

To improve the observing efficiency of the 74-inch and to permit realistic pointing error tests the author proposed a system of digital readout employing coarse/fine digital shaft encoder pairs on each axis, and a Hewlett Packard 2100A minicomputer which could also be used as the telescope data acquisition and instrument control machine. The author was responsible for the overall system design and for the encoder data handling software, but Wayne Ruting (detailed electronic design), John Hart (shaft encoder mechanics) and Ron Howe (software modifications and system adjustments) of Mt. Stromlo Observatory were responsible for the construction and implementation of it.

The heart of the 74-inch encoder and timing system (E.T.S.) comprises two 15 bit Baldwin optical encoders geared to the instrument spur gears on each axis by precision antibacklash pinions and double disc flexible couplings; see Figure C.1. The flexible couplings used have a torsional rigidity of 10 arcsecond per inch-ounce of applied torque and, with typical encoder starting torque of roughly an inch-ounce, undoubtedly comprise the weak link as far as accuracy is concerned. The ambiguity involved in the geared up fine encoders is removed by coarse 8 bit brush encoders driven at axis speed from the existing coarse selsyn shafts. Anti-ambiguity requirements of such paired encoder systems are discussed in Appendix B. This configuration of encoders gives a bit resolution of 0.837 arcsecond in declination and 1.47 arcsecond (equivalent to 0.098 second of time) in hourangle. Due to the slow rate of rotation of shaft encoders used on telescope axes, the bearing and brush limited lifetime for the system is in excess of 30 years, but the practical mean time between failures (M.T.B.F.) will probably be dictated by that of the optical encoder lamps which are rated at 20,000 hours (27 months), if not the encoder servicing electronics.

A block diagram of the encoder and timing system (E.T.S.) is shown in Figure C.2. The E.T.S. computer interface accepts data from the encoders and the sidereal and solar time code generators, multiplexing it into the H.P. 2100A via a standard H.P. 12566A T.T.L. dual input/output interface card; it also accepts the computer-generated display data from the card and serially transmits it to the 'Nixie-tube' type co-ordinate display. Upon command from the interface, the 23 encoder data bits together with a fail flag are captured and serialized for transmission to the interface by a module (shown also in Figure C.1) which also converts the Gray code from the fine encoders to Natural Binary. The timing data comes from two Eldorado 1710 time code generators which produce time to a decisecond in B.C.D. and also the standard serial I.R.I.G.-A code for serial transmission to other telescope systems. The sidereal time code generator is fed from a standard 5MHz frequency source, which is a Hewlett Packard 105B oscillator, via a phase-locked solar to sidereal frequency converter of the author's design (described in Appendix E). The second time code generator is normally fed directly from the standard oscillator giving solar time but is useful as a back-up should the sidereal one give trouble. The 78 bits (B.C.D.) of co-ordinate display data are transmitted serially once a second to the display which is buffered to prevent flicker, and equipped with suitable dimming circuitry for use in the darkened dome. Degrees, arcminutes and arcseconds are displayed in declination, and hours, minutes, seconds and deciseconds of time are displayed in right ascension and sidereal time; the polar co-ordinate displayed is selectable:- right ascension or hourangle.

The E.T.S. software was written in H.P. relocatable assembler and comprises three separate programs which perform (a) data capture, (b) data reduction computations and (c) display generation. The data capture routine is actuated every decisecond upon receipt of an interrupt from the sidereal time code generator and captures the 8 words of encoder and timing data loading it into an 8-word data block in core. Although it does not use Direct Memory Access (because of the initialization overheads), it employs a number of ruses which enable the capture to require only about 25 μs. The data capture routine is a mere 14 words in length and is permanently core resident. The reduction program, which is scheduled only when the raw 8-word data is to be used, is much larger (about 700 words), and performs the following tasks:

(i) it checks system fail flags for encoder lamp open circuit, power failure and other contingencies;

(ii) it converts days, hours, minutes, seconds and deciseconds of time from B.C.D. to separate integer variables and stores them;

(iii) it separates declination and hourangle fine and coarse encoder words into four separately stored integers;

(iv) it computes sidereal time in scaled double length integer format (D.L.I.);

(v) it computes declination δ in D.L.I. format;[@]

(vi) it computes hour angle H in D.L.I. format;[@]

(vii) it computes zenith angle and then refraction, and corrects δ and H for refraction.[@@]

@ See also Appendix B for a discussion of encoder anti-ambiguity programming.

@@ Use here is made of a high speed, relaxed accuracy trigonometric function generator similar to that given in Aus and Korn (1969).

(viii) If the right ascension display is selected it computes right ascension α.

(ix) It optionally computes azimuth angle e.g. for dome control.

The original version of the reduction program employed the standard Hewlett Packard floating point arithmetic subroutines and took about 12 ms to execute, but the final implementation of it[@] uses double length integer arithmetic throughout and is much faster with an execution time of approximately 2 ms. It can, if required, be FORTRAN called from another program, an instrument control routine for example, and is called by the co-ordinate display generator which converts the various co-ordinates to B.C.D. format ready for serial transmission to the display hardware. The display generator is usually scheduled with a low priority interrupt every second. The E.T.S. software is in current use as a foreground program in the H.P. real time executive R.T.E. system used on the 74-inch computer.

@ written by Ron Howe.

APPENDIX D
Computer Program Source Code Listings

This appendix contains the FORTRAN source listings for nearly all of the experimental programs used in the thesis. They are programmed for a Univac 1108, but in the majority of cases require only trivial modifications to run on an I.B.M. 360 series machine. The programs first listed are the main programs for observation list generation (CATALOG.OBS), parameter estimation (PEST.MAIN), surface fitting (SURFIT.MAIN), and telescope pointing data processing (TA.MAIN), respectively, and require an extensive set of subroutines which are then listed alphabetically. They are the programs used to obtain the results in the body of the thesis, and, like any developmental programs which undergo modifications and improvements as the need occasions, do not necessarily represent definitive, or even efficient implementations of the procedures or computations. Where a flowchart for an algorithm has been given, the numbers against blocks correspond to label numbers in the listings. Subroutines which are not listed here are RANDU and GAUSS, the Univac uniform and normal pseudo-random distribution generators respectively, READ, a free-field data input routine written by the author and, of course, the usual FORTRAN library of trigonometric and arithmetic functions. Virtually all of the computations use double precision real arithmetic, i.e. a mantissa of 60 bits length, to avoid the propagation of numerical errors.

CATALOG.OBS

```
C       ELT OBS  PROG CREATES OBSERVING LISTS,  MOD=12
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION RO(3),RI(3),RAS(1078),DEK(1078),HGRID(25,25),
       1DGRID(25,25),KSTAR(25,25)
C
C       FIRST OPTION<3 LIST ALL STARS.IF=2 INCLUDE FILECHECK.
C                >3 FORM LIST NEAR GRID.IF=4 ALSO LIST ALL.
C           IF=7 LIST DMIN ETC.
C       SECOND OPTOIN =1 SUPPRESS PRINTING
C
C       CONSTANTS
C
        PI=3.141592653589793D0
        PIO=PI/180.D0
C       ........................................
        WRITE(3,20)
 20     FORMAT(' ENTER OPTIONS (3I1)')
        READ(1,22) NNA,NNB,NNC
 22     FORMAT(3I1)
        WRITE(3,26)
 26     FORMAT(' ENTER DAY AND HOUR OF 1973 (I3,I2)')
        READ(1,24) KDAY,KH
 24     FORMAT(I3,I2)
        DAY=KDAY
        WRITE(3,25)KDAY,KH,NNA,NNB,NNC
 25     FORMAT(/' OBSERVING LISTS FOR DAY',I4,' START HOUR',I3,
       1' OPTIONS= ',3I2)
C       ...............................................
C       DEFAULT PARAMETERS.
        ECC=0.D0
        KSHORT=0
        MEAN=0
        ELONG=9.D0+56.D0/60.D0+1.35D0/3600.D0
        ELONG=ELONG*PI/12.D0
        PHI=35.D0+19.D0/60.D0+17.5D0/3600.D0
        PHI=-PHI*PIO
        TZONE=10.D0/24.D0
        BAR=760.D0
        TEMP=12.
C       TJ1=1970.0
        CALL RESLYR(1970.D0,TJ1)
C       ...............................................
C       TSTEP IN MIN.
        TSTEP=3.
        NPT=19
        MPT=19
        TJ2=2441682.5D0
C       TSTART IN HOURS A.E.S.T.
        TSTART=KH
C
        TJ2=TJ2+TSTART/24.D0-TZONE+DAY
        HSTART=5.D0
        IPTFIN=NPT*MPT
C
        DLIM1=-80.
        DLIM2=20.
        HLIM1=6.
        HLIM2=20.
        ZLIM=65.
C
        EPSD=((DLIM2-DLIM1)/MPT)*PIO
        EPSH=((HLIM1+24.D0-HLIM2)/NPT)*PI/12.D0
C
C
C
        DO 23 K=1,25
        DO 23 KK=1,25
 23     KSTAR(K,KK)=0
C       ...............................................
C
C
C
        IF(NNA.LT.3) GO TO 100
        IF(NNA.GT.3) GO TO 200
C
C
C       .......................................................
C       FORM GRID
C
 200    DO 210 N=1,NPT
        DO 210 M=1,MPT
        DGRID(N,M)=DLIM1*PIO+(M-1)*EPSD
        HGRID(N,M)=HSTART*PI/12.D0-(N-1)*EPSH/DCOS(DGRID(N,M))
        CALL SHIFT(HGRID(N,M))
        CALL SHIFT2(DGRID(N,M))
 210    CONTINUE
C
C       SET OBSERVATION NO. + 1ST PT ON GRID
C
        IOBS=1
        IPT=1
        N=1
        M=1
C
C       READ STAR CATALOG.
C
        DO 300 J=1,1078
        READ(11)ISTAR,DEK(J),RAS(J),ISIGN,ID,IM,IS,JH,JM,S,AMAG,BMAG,MVAR
        IF(NNA.NE.4) GO TO 300
        DEC=DEK(J)
        RA=RAS(J)
        WRITE(3,10) ISTAR,DEC,RA,ISIGN,ID,IM,IS,JH,JM,S,AMAG,BMAG,MVAR
 300    CONTINUE
        GO TO 400
C
C       SELCT NEW PT ON GRID
C
 500    IF(IPT.GE.IPTFIN) GO TO 6000
        IPT=IPT+1
C       IS N EVEN
 510    AN=N
        AN=AN/2.D0
        NA=AN
        AN=(AN-NA)
        IF(AN.LT.1.D-6) GO TO 520
C       ODD.
 560    IF(M.EQ.MPT) GO TO 540
 550    M=M+1
        GO TO 570
C       EVEN.
 520    IF(M.EQ.1) GO TO 540
 530    M=M-1
        GO TO 570
C       INCREMENT N
 540    N=N+1
 570     GO TO 400
C
 590    TJ2=TJ2-TSTEP/1440.D0
        GO TO 500
C
C       CHECK PT OK FOR TELESCOPE.
C
 400    DEC=DGRID(N,M)
        HA=HGRID(N,M)
C       CHECK DEC
        IF(DEC.LT.DLIM1*PIO.OR.DEC.GT.DLIM2*PIO) GO TO 500
C       CHECK HA
        IF(HA.GT.HLIM1*PI/12.D0.AND.HA.LT.HLIM2*PI/12.D0) GO TO 500
C       CHECK ZENITH ANGLE
        CALL ZANGLE(HA,DEC,ZA,PHI)
        IF(ZA.GT.ZLIM*PIO) GO TO 500
C
C       INCREMENT TIME BY TSTEP + FIND LOCAL ST.
C
 600    TJ2=TJ2+TSTEP/1440.D0
        CALL STIME(TJ2,ELONG,ST,MEAN)
C
C       FIND NEAREST STAR TO PT. ON GRID.
C
 700    DMIN=1.D0
        RA=ST-HA
        CALL SHIFT(RA)
        DO 710 J=1,1078
        DIFF=(DEC-DEK(J))**2+(RA-RAS(J))**2*DCOS(DEC)**2
        DIFF=DSQRT(DIFF)
        IF(DIFF.GE.DMIN) GO TO 710
        DMIN=DIFF
        MINEXT=MINJ
        MINJ=J
C       NEAREST STAR IS MINJ NEXT IS MINEXT.
 710    CONTINUE
C
C       REDUNDANT STAR CHECK.
        KSTAR(N,M)=MINJ
        K1=KSTAR(N,M-1)
        K2=KSTAR(N,M+1)
        K3=KSTAR(N-1,M)
        K4=KSTAR(N+1,M)
        IF(K1.EQ.MINJ.OR.K2.EQ.MINJ.OR.K3.EQ.MINJ.OR.K4.EQ.MINJ) GOTO590
C
         IF(NNA.EQ.7)WRITE(3,3000)MINJ,N,M,DEC,HA,RA,ZA,DMIN
 3000   FORMAT(/20X,' MINV,N,M,D,H,R,Z,DIF=',I4,2I3,5D12.5)
C
C       APPLY ASTRON. CORRECTIONS
C
 800    CALL PTRX(RAS(MINJ),DEK(MINJ),RI)
        CALL PRECES(RI,TJ1,TJ2,RO)
        CALL ABERR(RO,TJ2,ECC,RI)
        CALL NUTATE(RI,TJ2,RO,KSHORT)
         CALL RTPX(RA,DEC,RO)
C
C       CORRECT FO R  REFRACTION.
C
 1100    HA=ST-RA
        CALL REFRAK(HA,DEC,BAR,TEMP,HAO,DECO,PHI)
        DEC=DECO
        RA=ST-HAO
        CALL SHIFT(RA)
        CALL ZANGLE(HA,DEC,ZA,PHI)
C
C       PRINT ...........................
C
 200    CONTINUE
        CALL COORDS(ST,DEC,IH,IM,TSEC,LSIGN,LD,LM,LS)
        CALL COORDS(RA,DEC,KH,KM,SEC,LSIGN,LD,LM,LS)
        IF(NNB.EQ.1)GO TO 1220
       _WRITE(3,1210)MINJ,KH,KM,SEC,LSIGN,LD,LM,LS,IH,IM,TSEC,IOBS,N,M,
       1MINEXT
 1220    CONTINUE
C
 1210   FORMAT(/' STAR ',I4,' APPARENT PL. RA=',2I3,F5.1,'    DEC=',1R1,
       CI2,2I3,' SID TIME SHOULD BE ',2I3,F5.1,'IOBS,N,M,NEXTSTAR=',
       C3I4,I5)
C
        IF(NNA.EQ.7)WRITE(3,1230)MINJ,RA,DEC,HA,ZA
 1230   FORMAT(' MINJ,RA,DEC,HA,ZA=',I5,4F11.6)
C
C       INCREMENT OBSERVATION NO.
C
        IOBS=IOBS+1
        GO TO 500
C
C       EXIT...............................................
 6000   WRITE(3,6010)IOBS
 6010   FORMAT(/// ' ALL POINTS ON GRID FINISHED (THANK CHRIST) IOBS=',I6)
        STOP
C
C       ..........................................................
C
C
C
C
 100    DO 1000 I=1,1078
C       READ IN STAR.CATALOG.
        READ(11)ISTAR,DEC,RA,ISIGN,ID,IM,IS,JH,JM,S,AMAG,BMAG,MVAR
C       WRITE CHECK
        IF(I.NE.ISTAR) STOP
        IF(NNA.NE.2) GO TO 11
        WRITE(3,10)ISTAR,DEC,RA,ISIGN,ID,IM,IS,JH,JM,S,AMAG,BMAG,MVAR
 10     FORMAT(' FILECHECK ',I4,2D18.10,2X,1R1,3I3,2X,2I3,3(2X,F4.1),I2)
 11     CONTINUE
C
C       DO ASTRON CORRECTIONS
        CALL PTRX(RA,DEC,RI)
        CALL PRECES(RI,TJ1,TJ2,RO)
        CALL ABERR(RO,TJ2,ECC,RI)
        CALL NUTATE(RI,TJ2,RO,KSHORT)
        CALL RTPX(RAA,DECA,RO)
C
C       CONVERT COORDS.
C
        CALL COORDS(RAA,DECA,KH,KM,SEC,LSIGN,LD,LM,LS)
       _WRITE(3,30)ISTAR,TJ2,KH,KM,SEC,LSIGN,LD,LM,LS,JH,JM,S,
       1ISIGN,ID,IM,IS,AMAG
 30    _FORMAT(/' STAR ',I4,' UNREFRAK APP PLCE AT',F9.1,' RA=',2I3,F5.1,
       1' DEC=',1R1,I2,2I3,' (1970.0 MP=',2I3,F5.1,2X,1R1,I2,2I3,F4.1,')')
C
 1000    CONTINUE
        STOP
        END
```

PEST.MAIN

```
C       ELT=MAIN  G R HOVEY TELESCOPE DATA PROCESSING VERSION MOD= 10.
C       NONLINEAR PARAMETER ESTIMATION PROGRAM      ***UNIVAC 1108 VERSION*
C       FIRST CARD IS CONTROL CARD WITH  SEQUENCE NOS.   E.G......
C       11.6537     11 GIVES PROGRAM EXECUTED. PARMS 6,5,3,7 (UP TO 4)
C       ARE PASSED TO THAT PROG. AS OPTIONS
C
C       SECOND CARD CONTAINS DIMENSION CONSTANTS  N,M,K,NS.
C
C
        IMPLICIT REAL*8(A-H,O-Z)
        PARAMETER NNN=160,KKK=25
        DIMENSION     OPTION(20),NOPT(4),DIM(4)
        DIMENSION PARM(KKK),MASK(KKK),GTEMP(KKK),SCALE(KKK),G(KKK),T(KKK)
        DIMENSION X(NNN,2),Y(NNN,2),FN(NNN,2),W(NNN,2),DF(NNN,KKK,2)
        DIMENSION     A(KKK,KKK),ATEMP(KKK,KKK)
        DEFINE DARCOS(XDUM)=DACOS(XDUM)
         DEFINE DARSIN(XDUM)=DASIN(XDUM)
C
C
C       READ CONTROL CARD..
        WRITE(3,1010)
 1010   FORMAT(' INPUT(ANY FMT)CONTROL NO.S EG 2.01 3.0...')
        CALL READ(NVAR,OPTION,NERR)
        IF(NERR.EQ.1) GO TO 999
        ISTOP=NVAR
C
 2000   FORMAT('  CONTROL#= ',10F9.4/12X,10F9.4)
C
        WRITE(3,2000)OPTION
C
C       READ DIMENSION CONSTANTS..
        WRITE(3,1020)
 1020  _FORMAT(' INPUT(ANY FMT) N=NO OF PTS,M(=2),K=NO OF PARMS,',
       1'NS(=2).')
         CALL READ(NVAR,DIM,NERR)
        IF(NERR.EQ.1.OR.NVAR.NE.4) GO TO 999
        N=DIM(1)
        K=DIM(3)
        NS=DIM(4)
C
C       SORT OUT PROG NO. AND OPTIONS...
C
        DO 200 I=1,ISTOP
        NPROG=OPTION(I)
        PROG=OPTION(I)-NPROG+0.000001
        JPROG=0
        DO 300 KK=1,4
        PROG=(PROG-JPROG+0.000001)*10.D0
        M=DIM(2)
        JPROG=PROG
 300    NOPT(KK)=JPROG
C
        WRITE(3,50)N,M,K,NS,NPROG,NOPT
 50    _FORMAT(/' *****HOVEY NONLIN. PARMESTIMATION,NMKS= ',4I4,' *******S
       1TART'/' *****PROGRAM SELECTED=',I5,'  OPTIONS ',4I3/)
C
C       SELECT PROGRAM
        GO TO (1,2,3,4,5,6,7,8,9,10,11,12),NPROG
C
C       RETURN
 1000    CONTINUE
        WRITE(3,60) NPROG
 60     FORMAT(/' *****END NONLIN. PARMEST. PROG NO.',I4/)
C
 200    CONTINUE
        STOP
C
C       ILLEGAL CARD
 999    WRITE(3,998)NVAR
 998    FORMAT(' ****ILLEGAL CARD FORMAT',I6,'VARIABLES READ')
        STOP
C
C
C       PROG 1 = DATGEN
 1      CALL DATGEN(N,M,K,NS,NOPT    ,PARM,X,Y)
        GOTO 1000
C
C       PROG 2 = LEVENBERG DAMPED LEAST SQUARES ANALYSIS.
 2     _CALL DLSQ(N,M,K,NS,NOPT    ,X,Y,PARM,FN,DF,W,A,ATEMP,G,GTEMP,T,SCALE
       1E,MASK)
         GO TO 1000
C
C       PROG 3 = MARQUARDT ALGORITHM.
 3     _CALL MARQRT(N,M,K,NS,NOPT,X,Y,PARM,FN,DF,W,A,ATEMP,G,GTEMP,T,SCALE,
       1,MASK)
         GO TO 1008
C
 4     _CALL SPIRAL(N,M,K,NS,NOPT,X,Y,PARM,FN,DF,W,A,ATEMP,G,GTEMP,T,SCALE,
       1,MASK)
         GO TO 1000
C
 5       CALL DAPERT(N,M,NS,X,Y,W,NOPT)
         GO TO 1000
C
 6       CALL GRADNT(N,M,K,NS,NOPT,X,Y,PARM,FN,DF,W,G,GTEMP,T,SCALE,MASK)
         GO TO 1000
C
C
C
C
C        PROG SEVEN=TELESCOPE DATA INPUT ROUTINE
 7       CALL RDATIN(N,M,K,NS,NOPT,X,Y)
         GO TO 1000
C
C        8=PROG MOTEST.
 8       CALL MOTEST(N,M,K,NS,NOPT,X,Y,FN,W,PARM,MASK)
         GO TO 1000
C
C
C
C        PROG 9 = BEALE NONLIN TEST.
 9       CALL BEALE(N,M,K,NS,NOPT,X,Y,PARM,FN,DF,W,G,SCALE)
         GO TO 1000
C
C        PROG 10 =MARQUARDT WITH ESTIMATED DERIVATIVES.
 10    _CALL MARQT2(N,M,K,NS,NOPT,X,Y,PARM,FN,DF,W,A,ATEMP,G,GTEMP,T,SCALE,
       1,MASK)
         GO TO 1000
C
```

```
C       PROG 11 = GRAD2
11      CALL GRAD2(N,M,K,NS,NOPT,X,Y,PARM,FN,DF,W,G,GTEMP,T,SCALE,MASK)
        GO TO 1000
C
C
12     -CALL POWELL(N,M,K,NS,NOPT,X,Y,PARM,FN,DF,W,A,ATEMP,G,GTEMP,T,
       1SCALE,MASK)
        GO TO 1000
C       END
```

SURFIT.MAIN

```
C       ELT= MAIN, ERROR  SURFACE FITTING ROUTINE. MOD # 10.
C       G.R. MOVEY  ENG. PHYSICS A.N.U.
C       ERROR SURFACE FITTING PROGRAM       *******UNIVAC 1108 *********
C       FIRST CARD IS CONTROL CARD WITH  SEQUENCE NOS.    E:G.......
C       11.6537     11 GIVES PROGRAM EXECUTED, PARMS 6.5,3,7 (UP TO 4)
C       ARE PASSED TO THAT PROG. AS OPTIONS
C
C       SECOND CARD CONTAINS DIMENSION CONSTANTS  N,K.
C
C
C       PROG#4 NEEDS XY DATA IN F.LE 13 AS PER TA. ROUTINE.
C
C
        IMPLICIT REAL*8(A-H,O-Z)
        PARAMETER NN=160,KK=45
        DIMENSION    OPTION(25),NOPT(4),DIM(4)
        DIMENSION X(NN,2),Y(NN,2),C(KK,2),P(NN,KK,2)
        DIMENSION W(NN,2),JL(KK),JQ(KK),JG(KK)
        DIMENSION ALPHA(KK,KK,2),XX(NN,2),YY(NN,2)
        DIMENSION JS(KK),JT(KK),JRA(KK),JRB(KK)
C
C
C       READ CONTROL CARD..
        WRITE(3,1010)
1010    FORMAT(' INPUT (ANY FMT) CONTROL NOS. EG. 2.1201 1.0 ETC.')
        CALL READ(NVAR,OPTION,NERR)
        IF(NERR.EQ.1) GO TO 999
        ISTOP=NVAR
C
C       READ DIMENSION CONSTANTS..
        WRITE(3,1020)
1020    FORMAT(' INPUT (ANY FMT) N=NO. OF PTS. & K= MAX ORDER.')
         CALL READ(NVAR,DIM,NERR)
        IF(NERR.EQ.1.OR.NVAR.NE.2) GO TO 999
        N=DIM(1)
        K=DIM(2)
        KO=K+1
C
C       SORT OUT PROG NO. AND OPTIONS...
C
        DO 200 I=1,ISTOP
        NPROG=OPTION(I)
        PROG=OPTION(I)-NPROG+0.000001
        JPROG=0
        DO 300 KKK=1,4
        PROG=(PROG-JPROG+0.000001)*10.00
        JPROG=PROG
300     NOPT(KKK)=JPROG
C
        WRITE(3,50)N,K,NPROG,NOPT
50      FORMAT(/' ******MOVEY SURFACE FITTING PROGRAM..N,K= '
       1,2I5/' PROGRAM SELECTED= ',I5,' OPTIONS ',4I3/)
C
C       SELECT PROGRAM
        GO TO (1,2,3,4,5,6,7,8,9),NPROG
C
C       RETURN
1000     CONTINUE
        WRITE(3,60) NPROG
60      FORMAT(/' *******END SURFACE FITTING PROGRAM ',I4)
C
200     CONTINUE
        STOP
C
C       ILLEGAL CARD
999     WRITE(3,998)NVAR
998     FORMAT(' ####ILLEGAL CARD FORMAT',I6,'VARIABLES READ')
        STOP
C
C
C       PROG 1 = DATGEN
1       CALL DATGEN(N,K,KO,NOPT ,X,XX,Y,2100.00,6502197.00,9275.00)
         GO TO 1000
C
C       PROGRAM TWO = ERR SURFACE FITTING ROUTINE
2      -CALL ESFIT(N,K,KO,NOPT,X,XX,Y,YY,W,P,ALPHA,
       1C,JS,JT,JG,JQ,JL,JRA,JRB   )
         GO TO 1000
C        PROGRAM THREE.= FIT TEST ROUTINE.
3      -CALL FITEST(N,K,KO,NOPT,X,XX,Y,YY,W,P,ALPHA,C,
       1JS,JT,JG,JQ,JL,JRA,JRA)
         GO TO 1000
C
C        PROG =4 TELESCOPE DATA INPUT ROUTINE.
4        CALL TDATIN(N,K,KO,NOPT,X,Y)
         GO TO 1000
5        CONTINUE
6        CONTINUE
7        CONTINUE
8        CONTINUE
9        CONTINUE
C
C
         STOP
         END
```

TA.MAIN

```
C       ELT=A TELESCOPE DATA PROCESSING ROUTINE MOD=27.
C
C:::::OPTION ONE =0 MINUM PRINTING,=1 SOME, =2 ALL,=3 DEBUG PRINTING
C
C:::::OPTION TWO =0 READS IN RAW DATA  AND GENERATES
C        DATA X,Y IN FILE 13.,=1 AS ABOVE IGNORING REFRACTION
C        =2 READ IN AND PLOT LASER DATA.
C
C:::::OPTION THREE=1 INCLUDES A CHECK ON ENCODER SYSTEM.
C
C:::::OPTION  FOUR =1 PLOTS TELESCOPE ERRORS,=2 PRINTS+PLOTS,=3 PRINTS O
C
C:::::OPTION FIVE NOT=0 SID TIME TAKEN FROM ETS RATHER THAN U1108.
C
C:::::OPTION SIX NOT=0 STAR COORDS TAKEN FROM ETS RATHER THAN U1108.
C
C
C
C        REQUIRES STAR CATALOG IN FILE 11.
C        REQUIRES FILE 13 TO WRITE X-Y DATA IN.
C
         IMPLICIT REAL*8(A-H,O-Z)
         REAL DD,DH,DRFS,DZ,D,H,ZA
         PARAMETER NN=200
         DIMENSION DEC(NN),HA(NN),DECD(NN),HAD(NN),TIME(NN),ZA(NN),SAVE(NN)
         DIMENSION NSTAR(NN),X(NN,2),Y(NN,2),DD(NN),DH(NN),DRES(NN),DZ(NN)
         DIMENSION R1(3),R0(3),R(6)
         DIMENSION H(NN),D(NN)
         DIMENSION KE(24,NN),KR(9)
         EQUIVALENCE (DECD,X),(HAD,X(1+NN)),(DEC,Y),(HA,Y(1+NN))
         INTEGER FLD
         DEFINE Q(JJ,NJ)=BLURP(JJ,NJ)
C
         PI=3.14159265358979300
         TP=2.00*PI
         PBT=PI/2.00
         PI0=PI/180.00
         ASEC=PI0/3600.00
         HR=PI/12.00
         TSEC=HR/3600.00
         TMIN=HR/60.00
         AMIN=PI0/60.00
C
         TSX=2.00**16
         ITSX=2**16
         TSF=2.00**15
         DSC=189.00*TSX
         HSC=3375.00*TSX
C
         ECC=0.00
C        NOTE..KSHORT=0 INCLUDES SHORT PERIOD NUT TERMS,MEAN=0 GIVES MEAN S'
         KSHORT=0
         MEAN=9
         ELONG=9.00*HR+56.00*TMIN+1.3500*TSEC
         TZONE=10.00/24.00
         TEMP=15.6
         BAR=696.800
         PHI=-35.00*PI0-19.00*AMIN-17.5*ASEC
C
C        READ OPTIONS
C
         WRITE(3,10)
10      -FORMAT(/' ENTER OPTIONS,NO. OF OBSERVATIONS & DAY OF'
       1'1973 (6I1/I3/I3)')
         READ(1,20)NA,NB,NC,ND,NE,NF,NOBS,NDAY
20       FORMAT(6I1/I3/I3)
         WRITE(3,30) NOBS,NA,NB,NC,ND,NE,NF,NDAY
30      -FORMAT(' TELESCOPE DATA ANALIS FOR',I4,' POINTS,OPTNS=',6I2,
       1' DATE=',I4)
C
C        TJ1=1973.0,TJ2 STARTS AT TJ=0/1/73.
         CALL AFSLYR(1970.00,TJ1)
         TJ=2441682.5D0
C        INITIALIZE TJ FOR DAY (REMEMBER DAYLIGHT SAVING END 4/3/73.)
         DAY=NDAY
         TJ=TJ-TZONE+DAY
C
C
         IF(NB.EQ.2) GO TO 1100
C
C        READ IN RAW TELESCOPE DATA.
C
100      CONTINUE
         WRITE(3,101) NOBS
101      FORMAT(' ENTER ',I4,' OBSERVATIONS OF 4 LINE DATA...')
         WRITE(3,112)
112      FORMAT('1',40('*'),'RAW DATA INPUT',40('*'))
         DO 120 I=1,NOBS
         READ(1,102) KN,LABEL,KA,KB,KC
102      FORMAT(I6,6X,1R3,1X,I5,7X,I5,7X,I5)
         LABTST=3R0BS
         IF(LABEL.EQ.LABTST) GO TO 103
         WRITE(3,104) KN
104      FORMAT(//' OBSVN. SEQUENCE ERROR AT LINE NO.  ',I6,//)
103      CONTINUE
         NSTAR(I)=KB
         IF(NA.GE.3) WRITE(3,105) KN,KA,KB,KC,I
105      FORMAT(/' L N. ',I6,' INITL OBS NO.,STAR,TRAK,NEW OBS NO.=',4I9)
C
         DO 110 J=1,3
         READ(1,107) KR
107      FORMAT(I6,6X,8(O6,2X))
         DO 110 L=1,8
         LL=8*(J-1)+L
         KE(LL,I)=KR(L+1)
110      CONTINUE
         IF(NA.GE.3) WRITE(3,106) (KE(LP,I),LP=1,24)
106      FORMAT(8(2X,O6))
120      CONTINUE
C
C
C
C        PROCESSRAW DATA GET SOLAR TIME  AND CHECK ENCODER PERFORMANCE IF NC
C
300      CONTINUE
         WRITE(3,312)
312      FORMAT('1',40('*'),'DATA REDUCTION& ENCODER CHECK',40('*'))
         DATA MASK/076000000000/
         DATA MFD,MCD,MFH,MCH/-269903,140,180218,-249/
         GD=1812.00/32.00
         GH=864.00/32.00
         DDDMAX=0.00
         DHHMAX=0.00
         DSTMAX=0.00
         DJITMX=0.00
         HJITMX=0.00
C
C        *****************************************************************
         DO 450 J=1,NOBS
C        GET SOLAR TIME IN RADIAN
         S=IBD(0,4,KE(6,J))*0.100+IBD(0,7,KE(8,J))
         JM=IBD(7,7,KE(8,J))
         JH=IBD(0,6,KE(7,J))
         KDAY=IBD(6,10,KE(7,J))
         TIME(J)=JH*HR+JM*TMIN+S*TSEC
         IF(NA.GT.2.OR.NC.NE.0) WRITE(3,302) J,NSTAR(J),KDAY,JH,JM,S
302     -FORMAT(/1X,20('*'),' OBSVN. ',I5,' STAR NO. ',I6,' AT DAY ',I5,
       1' UT=HMS=',2I5,F6.1,' *******')
C
C        GET DISPLAY OUTPUT.
         KK=KE(22,J)
         JH=IBD(0,6,KK)
         ISIG=IBD(7,1,KK)
         IS=IBD(8,7,KK)
         KK=KE(21,J)
         IM=IBD(0,7,KK)
         ID=IBD( 7,8,KK)
         IF(ISIG.EQ.0) ID=-ID
         KK=KE(23,J)
         S=IBD(1,7,KK)
         JM=IBD(8,7,KK)
         KK=KE(24,J)
         S=S+IBD(11,4,KK)*0.100
         SS=IBD(4,4,KE(6,J))*0.100
         KK=KE(5,J)
         SS=SS+IBD(0,7,KK)
         JMS=IBD(7,7,KK)
         JHS=IBD(0,6,KE(4,J))
         IF(NC.NE.0)WRITE(3,304)JHS,JMS,SS,JH,JM,S,ID,IM,IS
304     -FORMAT(' ETS DISPLAY WAS SID TM=',2I5,F6.1,'  RA/HA=',2I5,F6.1,
       1' DEC=',I6,2I5)
C
C        REDUCE ETS DERIVED COORDS.
         IF(NC.NE.0)WRITE(3,306)
306     -FORMAT(' ETS-COMPUTED COORDS UPPER=RAW HA,ST,DEC/LOWER=CORR',
       1'ECTED HA,RA,DEC .BRACKETS=RADIAN.')
         KK=KE(9,J)+KE(10,J)*ITSX
         ETSST=KK*TP/HSC
C        CHECK TIME OPTION IF.NE.0 SAVE ETSST IN SAVE(J).
         IF(NE.NE.0) SAVE(J)=ETSST
         KK=KE(11,J)+KE(12,J)*ITSX
         ECHA=KK*TP/HSC
         KK=KE(13,J)+KF(14,J)*ITSX
         ECRA=KK*TP/HSC
         KK=KE(19,J)+KE(20,J)*ITSX
         EHA=KK*TP/HSC
C
         ISIG=FLD(20,1,KE(16,J))
         KK=FLD(21,15,KE(16,J))*ITSX+KE(15,J)
         IF(ISIG.EQ.1)KK=(KK+MASK)
         ECDEC=KK*TP/DSC
         ISIG=FLD(20,1,KE(18,J))
         KK=FLD(21,15,KE(18,J))*ITSX+KE(17,J)
         IF(ISIG.EQ.1) KK=(KK+MASK)
         EDEC=KK*TP/DSC
C
         CALL COORDS(ETSST,EDEC,MH,MM,SS,LSIGN,LD,LM,LS)
         CALL COORDS(EHA,EDEC,KH,KM,S,LSIGN,LD,LM,LS)
        -IF(NC.NE.0)WRITE(3,310)EHA,KH,KM,S,ETSST,MH,MM,SS,EDEC,LSIGN,LD,
       1LM,LS
C308      FORMAT(' (',D16.8,') HMS= ',2I4,F6.1,40X,'(',D16.8,') DMS= ',
C       1R1,I3,2I4)
         CALL COORDS(ECHA,ECDEC,KH,KM,S,LSIGN,LD,LM,LS)
         CALL COORDS(ECRA,ECDEC,MH,MM,SS,LSIGN,LD,LM,LS)
        -IF(NC.NE.0)WRITE(3,310)ECHA,KH,KM,S,ECRA,MH,MM,SS,ECDEC,LSIGN,
       1LD,LM,LS
310     -FORMAT(2(' (',D16.8,') HMS= ',2I4,F6.1),' (',D16.8,') DMS= ',
       11R1,I3,2I4)
C
C
350      CONTINUE
C
C
C
C        COMPUTE COORDS HAD,DECD FROM RAW ENCODER WORDS.
C
400      CONTINUE
C
         NFD=FLD(21,15,KE(1,J))
         NFH=FLD(21,15,KE(3,J))
         NCD=FLD(20,8,KE(2,J))
         NCH=FLD(28,8,KE(2,J))
         IF(NA.GT.2)WRITE(3,402)J,NFD,NCD,NFH,NCH
402     -FORMAT(' OB NO.',I4,' RAW FINE,COARSE ENCODER WORDS DEC=',
       12(2X,I6),' HA= ',2(2X,I6))
C         NFD=NFD
         NFH=-NFH
         NCD=-NCD+MCD
         NCH=NCH+MCH
C        GET ANTIAMBIGUITY FACTORS.
         AD=NCD*GD*2.00**7
         MD=AD/TSF
         AH=NCH*GH*2.00**7
         MH=AH/TSF
         DMIN=1.D+20
         HMIN=1.D+20
         DO 401 L=-2,1,1
         NND=MD+L
         NH=MH+L
         RD=(NND*TSF+NFD)
         RH=(NH*TSF+NFH)
         RDD=DABS(RD-AD)
         RHH=DABS(RH-AH)
         IF(RDD.GT.DMIN) GO TO 403
         DMIN=RDD
         DMINB=(RD-AD)/(GD*2.00**7)
         NAMD=NND
403      IF(RHH.GT.HMIN) GO TO 405

         HMIN=RHH
         HMINB=(RH-AH)/(GH*2.00**7)
         NAMH=NH
405      CONTINUE
401      CONTINUE
C
         IF(NC.NE.0)WRITE(3,404)HMINB,NAMH,DMINB,NAMD
404     -FORMAT(' DIFFTL ENCODER JITTER(COARSE BITS) & ANTAMBIG NO.= HA ',
       1F10.2,I6,'  DEC= ',F10.2,I6)
C
         DECD(J)=(NAMD*TSF+NFD+MFD)*TP/(GD*TSF)
         HAD(J)=(NAMH*TSF+NFH+MFH)*TP/(GH*TSF)
         CALL SHIFT2(DECD(J))
         CALL SHIFT(HAD(J))
         DECDJ=DECD(J)
         HADJ=HAD(J)
         CALL COORDS(HAD(J),DECD(J),KH,KM,S,LSIGN,LD,LM,LS)
         IF(NA.GT.1)WRITE(3,406)J,KH,KM,S,LSIGN,LD,LM,LS
406     -FORMAT(' OB NO.',I4,' COORDS COMPUTED BY U1108 FOR ENCODER WDS',
       1'HA(HMS)= ',2I4,F6.1,' DEC (DMS)= ',1R1,I3,2I4)
C        CHECK COORD OPTION IF NONZERO USE ETS VALUES.
         IF(NF.EQ.0)GO TO 430
         DECD(J)=EDEC
         HAD(J)=EHA
430      CONTINUE
C
C
```

```
      IF(NC.EQ.0)GO TO 420
C     COMPUTE U1108-ETS DIFFS
      DDD=(DECDJ-EDEC)/ASEC
      DHH=(HADJ-EHA)/ASEC
      TJ2=TJ+TIME(J)/TP
C     LOOK OUT FOR MIDNIGHTS
      IF(TIME(J).LT.4.D0)TJ2=TJ2+1.D0
      CALL STIME(TJ2,ELONG,ST,EQUIN,MEAN)
      DST=(ST-ETSST)/TSEC
      IF(DABS(DDD).LE.DDDMAX) GO TO 411
      DDDMAX=DABS(DDD)
      IDMAX=J
411   IF(DABS(DHH).LE.DHHMAX) GO TO 412
      DHHMAX=DABS(DHH)
      IHMAX=J
412   IF(DABS(DST).LE.DSTMAX) GO TO 413
      DSTMAX=DABS(DST)
      ISMAX=J
413   CONTINUE
      IF(DABS(HMINB).LE.HJITMX) GO TO 414
      HJITMX=DABS(HMINB)
      IHJIT=J
414   IF(DABS(DMINB).LE.DJITMX) GO TO 415
      DJITMX=DABS(DMINB)
      IDJIT=J
415   CONTINUE
      WRITE(3,410)DDD,DHH,DST
410  _FORMAT(' U1108-ETS DIFFERENTIALS DEC,HA(ARCSEC),ST(SECTM)=',
     13F10.2)
420   CONTINUE
C
450   CONTINUE
      *********************************************************
C
C
C
C
C     READ IN STAR CATALOG FILE 11. +DO ASTRON CORRECTIONS
C
500   CONTINUE
     _IF(NC.NE.0)WRITE(3,502)DDDMAX,IDMAX,DHHMAX,IHMAX,DSTMAX,
     1ISMAX,DJITMX,IDJIT,HJITMX,IHJIT
502  _FORMAT('1 MAXIMUM U1108-ETS DIFFS IN DEC,HA,ST &MAX D,H ',
     1'ENCODERJITTS/OBSNO IN BKTS',/5(F10.2,' ATOBSNO(',I4,')'))
      WRITE(3,501)
501  _FORMAT('1 CATALOG PLACE OF STARS USED IN ORDER OF LOCN.',
     1' IN FILE ..STARS..')
      DO 520 I=1,1078
      READ(11)ISTAR,DEK,RA,IP,ID,IM,IS,JH,JM,S,AMAG,BMAG,MVAR
      DO 520 J=1,NOBS
      IF(NSTAR(J).NE.ISTAR)GO TO 520
      DEC(J)=DEK
C     NOTE HA(J) CONTAINS RA
      HA(J)=RA
      IF(NA.GE.2)WRITE(3,510)J,NSTAR(J),DEC(J),RA,IP,ID,IM,IS,JH,JM,S
510  _FORMAT(' OBS ',I3,' STAR ',I4,' CATLG DEC,RA=',2D20.10,3X,1R1,I2,
     12(1X,I2),I6,I3,F5.1)
520   CONTINUE
C
      WRITE(3,542)
542   FORMAT('1 FINAL APPARENT PLACES IN ORDER OF OBSERVATION.')
C
C
      DO 530 J=1,NOBS
C     GET J.D. FOR OBSVN. FROM UT THEN CORRECT MEAN PLACE.
      TJ2=TJ+TIME(J)/TP
C      CORRECT FOR GOING THRU MIDNIGHT
      IF(TIME(J).LT.4.D0) TJ2=TJ2+1.D0
C
      CALL PTRX(HA(J),DEC(J),RI)
      CALL PRECES(RI,TJ1,TJ2,RO)
      CALL ABERR(RO,TJ2,ECC,RI)
      CALL NUTATE(RI,TJ2,RO,KSHORT)
      CALL RTPX(RA,DEK,RO)
C
      CALL STIME(TJ2,ELONG,ST,EQUIN,MEAN,
C     CHECK TIME OPTION, ST MAY BE STORED IN SAVE(J) OF ALL PLACES...
      IF(NE.NE.0) ST=SAVE(J)+EQUIN
      HAL=ST-RA
      TIME(J)=ST
C     TOPOCENTRIC COORDS...HAL,DEK
      DEC(J)=DEK
      HA(J)=HAL
      CALL SHIFT(RA)
      CALL SHIFT2(DEK)
      CALL COORDS(RA,DEK,KH,KM,S,LSIGN,LD,LM,LS)
      CALL COORDS(ST,DEK,MH,MM,SS,LSIGN,LD,LM,LS)
     _IF(NA.GE.2)WRITE(3,544)J,NSTAR(J),DEK,RA,LSIGN,LD,LM,
     1LS,KH,KM,S,MH,MM,SS
544   FORMAT(' OBS ',I3,' STAR ',I4,' UNREFRAK  DEC,RA=',2D18.10,3X,
     11R1,I2,2(1X,I2),I6,I3,F5.1,' ST=',I6,I3,F5.1)
      IF(NB.EQ.1) GO TO 540
C
C     REFRACTION CORRECTION
C
560   CALL REFRAK(HAL,DEK,BAR,TEMP,HAO,DECO,PHI)
      DEC(J)=DECO
      HA(J)=HAO
      RA=ST-HAO
540   CALL SHIFT2(DEC(J))
      CALL SHIFT(HA(J))
      CALL SHIFT(RA)
      CALL COORDS(RA,DECO,MH,MM,SS,LSIGN,LD,LM,LS)
      CALL COORDS(HA(J),DEC(J),KH,KM,S,LSIGN,LD,LM,LS)
     .IF(NA.GE.2)WRITE(3,541)J,NSTAR(J),DEC(J),HA(J),LSIGN,LD,LM,
     1LS,KH,KM,S,MH,MM,SS
541  _FORMAT(' OBS ',I3,' STAR ',I4,'      FINAL DEC,HA=',2D18.10,3X,
     11R1,I2,2(1X,I2),I6,I3,F5.1,' RA=',I6,I3,F5.1)
530   CONTINUE
C
C     WRITE X,Y DATA IN FILE 13.
C
600   WRITE(13)NOBS
      DO 610 J=1,NOBS
      WRITE(13)X(J,1),X(J,2),Y(J,1),Y(J,2)
      IF(NA.GT.1)WRITE(3,620)J,X(J,1),X(J,2),Y(J,1),Y(J,2)
620   FORMAT(' OBS ',I4,' X,Y= ',4D20.10)
610   CONTINUE
      WRITE(3,630) NOBS
630   FORMAT(' FILE 13 WRITTEN WITH ',I5,' DATA POINTS')
C
C     CALCULATE PHY + MAX ERROR.
C
700   PHY=0.D0
      ERRMAX=0.D0
      DO 730 J=1,NOBS
      DD(J)=DEC(J)-DECD(J)
      DH(J)=HA(J)-HAD(J)
      A=DD(J)**2+DH(J)**2*DCOS(DECD(J))**2
      DRES(J)=DSQRT(A)/ASEC
      PHY=PHY+A
      IF(DRES(J).LE.ERRMAX) GO TO 730
      ERRMAX=DRES(J)
      IWORST=J
730   CONTINUE
      RMS=DSQRT(PHY/NOBS)/ASEC
      WRITE(3,720) PHY,RMS,ERRMAX,IWORST
720  _FORMAT(' PHY FOR OBS= ',D20.10,' RMS+ERRMAX(ASEC)= ',2F11.3,
     1' (AT OBS NO. ',I3,')')
      IF(ND.EQ.0) GO TO 1500
C
C     RESOLVE ERRORS ALONG ZENITH LINE +PRINT ERRORS IN ARC SEC.
C
800   CONTINUE
      DO 810 J=1,NOBS
      CALL ZANGLE(HA(J),DEC(J),ZAY,PHI)
      CALL ZANGLE(HAD(J),DECD(J),ZAX,PHI)
      ZA(J)= ZAX/PIO
      DZ(J)=(ZAY-ZAX)/ASEC
      DD(J)=DD(J)/ASEC
      DH(J)=DH(J)/ASEC
      D(J)=DECD(J)/PIO
      HAG=HAD(J)
      CALL SHIFT2(HAG)
      H(J)=HAG/HR
     _ IF(ND.GT.1)WRITE(3,820)J,DD(J),DH(J),DRES(J),DZ(J),D(J),H(J),
     1ZA(J)
820  _FORMAT(' OBS ',I3,'DD= ',F8.2,' DH= ',F8.2,' DRES= ',F8.2,' DZ= '
     1,F8.2,' (ARCSEC) D,H,Z= ',3F12.8)
810   CONTINUE
      IF(ND.LT.3) GO TO 1000
      GO TO 1500
C
C     PLOT 4 ERRORS (ARCSEC)  AGAINST 3 COORDS (RAD)
C
1000  CONTINUE
      CALL PLOT(NOBS,D,DD)
      CALL PLOT(NOBS,D,DH)
      CALL PLOT(NOBS,D,DRES)
      CALL PLOT(NOBS,D,DZ)
      CALL PLOT(NOBS,H,DD)
      CALL PLOT(NOBS,H,DH)
      CALL PLOT(NOBS,H,DRES)
      CALL PLOT(NOBS,H,DZ)
      CALL PLOT(NOBS,ZA,DD)
      CALL PLOT(NOBS,ZA,DH)
      CALL PLOT(NOBS,ZA,DRES)
      CALL PLOT(NOBS,ZA,DZ)
C
C
1500  WRITE(3,1510)
1510  FORMAT(/' THAT IS THE END FOLKS')
      STOP
C
C*********************************************************
C
C     LASER DATA INPUT
C
1100  CONTINUE
      NOBS=70
      WRITE(3,1120) NOBS
1120  FORMAT(' ENTER ',I4,' LASER DATA PIONTS.')
      SCALE=1.D0/33800.D0
C
      DO 1150 KKK=1,2
      DO 1150 KK=1,7
      DO 1150 K=2,1,-1
      M=5
      CALL READ(M,R,IER)
      DO 1150 J=1,5
      I=J+(KK-1)*5+(KKK-1)*7*5
      X(I,1)=-90.D0+30.D0*(J-1)
      Y(I,K)=R(J)
      X(I,2)=-6.D0+2.D0*(KK-1)
      IF(NA.GT.2.AND.K.EQ.1)WRITE(3,1170)I,X(I,1),X(I,2),Y(I,1),Y(I,2)
1170  FORMAT(' PT.LASER ',I4,'X,Y READIN= ',4F9.3)
1150  CONTINUE
      J=0
      DO 1140 K=1,70
      IF(Y(K,1).LT.1.D-10) GO TO 1160
      X(K,1)=X(K,1)*PIO
      X(K,2)=X(K,2)*HR
      Y(K,1)=-Y(K,1)*SCALE +X(K,1)
      Y(K,2)=Y(K,2)*SCALE+X(K,2)
      J=J+1
      NSTAR(K)=J
      GO TO 1140
1160  NSTAR(K)=100
1140  CONTINUE
      NOBS=J
      DO 1180 K=1,70
      J=NSTAR(K)
      X(J,1)=X(K,1)
      X(J,2)=X(K,2)
      Y(J,1)=Y(K,1)
      Y(J,2)=Y(K,2)
1180  CONTINUE
      DO 1190 K=1,NOBS
      IF(NA.GE.3)WRITE(3,1200)K,X(K,1),X(K,2),Y(K,1),Y(K,2)
1200  FORMAT(' SORTED PT. NO. ',I4,' X,Y= ',4D20.10)
1190  CONTINUE
C
C     NOW HAVE STANDARD X,Y VECTORS
C
      GO TO 700
C
C
      END
```

ABERR

```
C     ELT=I   ABERR MOD=1
      SUBROUTINE ABERR(RG,TJ,ECC,RA)
C     STELLAR ABERRATION CORRECTION.
      IMPLICIT REAL*8(A-H,O-Z)
C     CALCULATES STELLAR ABERRATION .
C     IF ECC ZERO OMITS ELLIPTIC TERMS.
      REAL*8 M,RG(3),RA(3)
      PI=3.14159265358979D0
      PIO=PI/180.D0
      ABCON=20.496D0*PIO/3600.D0
      T=(TJ-2415020.D0)/36525.D0
      SUN=279.696678D0+36000.768925*T+0.3025D-3*T*T
      SPI=281.220833D0+1.719175D0*T+0.45278D-3*T*T+0.33333D-5*T*T*T
      M=129596579.10*T-0.54*T*T
      M=358.475845+M/3600.D0
      EPS=23.452294D0-0.0130125D0*T
      E  =0.01675104D0-4.18D-5*T
      SUN=SUN*PIO
      M=M*PIO
      SPI=SPI*PIO
      EPS=EPS*PIO
C     CALC. TRUE LONG. OF SUN FROM EQ'N. OF CENTRE.............
      CORR=(2.D0*E-E**3/4.D0)*DSIN(M)+1.25D0*E*E*DSIN(2.D0*M)+13.D0*E**3*
     C*DSIN(3.D0*M)/12.D0
      SUN=SUN+CORR
C     ***********************************************************
      IF(ECC.EQ.0.D0)GO TO 1
      ECC=E
1     CONTINUE
      SS=DSIN(SUN)
      CS=DCOS(SUN)
      SP=DSIN(SPI)
      CP=DCOS(SPI)
      CE=DCOS(EPS)
C     ***********************************************************
      C=-ABCON*(CS*CE+ECC*CP*CE)
      D=-ABCON*(SS+ECC*SP)
C
      RA(1)=RG(1)-D
      RA(2)=RG(2)+C
      RA(3)=RG(3)+C*DTAN(EPS)
2     RETURN
      END
```

ANABEL

```
C     ELT=F   ANABEL  MOD NO.  3
      SUBROUTINE ANABEL(K,KD,JS,JT,JG,JQ,JL,JRA,JRB)
C     GIVEN J COMPUTES S,T,G,Q,L
C     AND R-INDEX LIMITS FOR ALL J=0....K.
C
      DIMENSION JS(KD),JT(KD),JG(KD) ,JQ(KD),JL(KD),JRA(KD),JRB(KD)
C
      JS(KD)=0
      JT(KD)=0
      JG(KD)=0
      JRA(KD)=-1
      JRB(KD)=-1
C
      DO 100 J=1,K
      JTEMP=J
      JT(J)=0
C     CALC. G AND TEST IF INTEGRAL.
20    AG=SQRT(1.+8.*JTEMP)
      AG=(AG-1.)/2.+1.E-5
      KG=AG
      BG=KG
      BG=AG-BG
      IF(BG.LT.1.E-4) GO TO 30
      JT(J)=JT(J)+1
      JTEMP=JTEMP-1
      GO TO 20
C
30    JG(J)=KG
      JS(J)=KG-JT(J)
      IF(JT(J).EQ.0) GO TO 40
      JL(J)=2
      JTD=JT(J)-1
      GO TO 60
C
40    JL(J)=1
      JTD=0
C
60    KQ=KG*(KG-1)
      JQ(J)=KQ/2+JTD
C     CALC. INDEX LIMITS..
      JB=J-1
      IF(J.GT.5) GO TO 70
      JA=0
      GO TO 90
C
70    JA=J
80    JA=JA-1
      JC=JA
      IF(JC.EQ.0) JC=KD
      KG=JG(J)-JG(JC)
      IF(KG.LE.2) GO TO 80
      JA=JA+1
C
90    JRA(J)=JA
      JRB(J)=JB
100   CONTINUE
      RETURN
      END
```

BEALE

```
C     ELT=Q   BEALE NONLINEARITY TEST(NEW VERSION)  MOD 16.
      SUBROUTINE BEALE(N,M,K,NS,NOPT,X,Y,BD,FM,DF,FD,BM,SCALE)
C     PROG # 9
C
C     NOPT(1) GT 0 ENABLS PRINT
C
C      NOPT(2)GIVES METHOD,=0 UNIFORM DISTRIBUTION OVER INTERVAL
C      PROPORTIONAL TO PARM,=1   "        " OVER SPECIFIED    "
C      =2   NORMAL DISTRIBUTION WITH S.DEV PROPORTIONAL ETC
C       =3   "          "          "    "    SPECIFIED.
C
C     NOPT(3,4) GIVES NO. OF POINTS TO BE USED.
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(N,M),Y(N,NS),FM(N,NS),DF(N,K,NS),FD(N,NS),BD(K)
      DIMENSION BM(K),SCALE(K),BDIF(2),FDIF(2),BDINC(2),BDAV(30),NOPT(4)
      REAL STRAND,RAND(30),TRAND(100),RANDY
C
      PI=3.14159265358979D0
      PIO=PI/180.D0
      SEC=PIO/3600.D0
C
```

```
      MMM=NOPT(4)+NOPT(3)*10
      NPR=NOPT(1)
      METH=NOPT(2)
      STRAND=654779.
      CALL RANDU(STRAND,1)
      TRAND(1)=STRAND*2.**34
      CALL RANDU(TRAND,MMM)
C
C
C
C
C     MDIS=NO. OF DIFFERENTLY SCALED DISTRIBUTIONS
C     START=STARTING PARM
C     FACTOR =ITERATIVE  MULTIPLIER
C
      MDIS=10
      START=1.D-3
      FACTOR=2.D0
C
C
C
C
C
      CALL FCTN(N,M,K,NS,X,BD,FD,IFF)
      IF(NPR.GE.4)WRITE(3,904)(I,FD(I,1),FD(I,2),I=1,N)
904   FORMAT(' FD= ',I4,2D20.10)
      CALL DERV(N,M,K,NS,X,BD,DF,IFF)
      IF(NPR.GE.4)WRITE(3,903)(((DF(I,J,L),J=1,K),L=1,2),I=1,N)
903   FORMAT(1X,11D9.2,1X/1X,11D9.2/1X,11D9.2/1X,11D9.2//)
      PHY=0.D0
      DO 110 I=1,N
      PHY=PHY+(Y(I,1)-FD(I,1))**2
      PHY=PHY+(Y(I,2)-FD(I,2))**2*DCOS(X(I,1))**2
110   CONTINUE
      NU=2*N-K
      SIGSQ=PHY/NU
      WRITE(3,120)METH,MMM,SIGSQ,NU,PHY
120  _FORMAT(/' BEALE(',I2,')NONLIN TEST ON',I4,' PNTS OF SAMPLE SPACE',
     1/' SIGSQ(NU)= ',D16.6,I4,' PHY=',D16.6)
C
C
      DO 1000 IT=1,MDIS
C
C
C
C
C
      IF(METH.EQ.1.OR.METH.EQ.3)GO TO 550
C     METH=0,2  PROPORTIONAL...................................
      SKALE=START*FACTOR**IT
      DO 510  J=1,K
510   SCALE(J)=SKALE*DABS(BD(J))
      GO TO 100
C
C     METH=1,3      INTERVAL ETC SPECIFIED   ..........................
550   SKALE=START*100.D0*FACTOR**IT
      DO 530 J=1,K
530   SCALE(J)=SKALE*SEC
      SCALE(3)=SKALE*PIO
      GO TO 100
C
C
100   CONTINUE
      R=0.D0
      S=0.D0
      DO 1050 J=1,K
1050  BDAV(J)=0.D0
      MM=1
C
C
200   CONTINUE
      IF(METH.GE.2) GO TO 230
C
C
C     UNIFORM DISTRIBUTION    ......................................
      RAND(1)=TRAND(MM)*2.**34
      CALL RANDU(RAND,K)
      IF(NPR.GE.2) WRITE(3,130)IT,MM,STRAND,RAND
130   FORMAT(/I4,'TH TEST',I4,'TH PNT.  STRAND,RAND=',(5F14.5))
C
      DO 210 J=1,K
      DR=(RAND(J)-0.5)
      BM(J)=BD(J)+DR*SCALE(J)
210   CONTINUE
      GO TO225
C
C
C                                   ..................................
C
C     NORMAL DISTRIBUTION
C
230   IRAND=TRAND(MM)*2.**34
      DO 240 J=1,K
      CALL GAUSS(IRAND,0.37,0.0,RANDY)
      BM(J)=BD(J)+RANDY*SCALE(J)
240   CONTINUE
C
C
C                            ...................................
C
C
C
C     GENERATE FUNCTION VALUES
225   IF(NPR.GE.2)WRITE(3,220)IT,BM
220   FORMAT(' PARM VECTOR',I4,'=',(5D18.10))
C
300   CALL FCTN(N,M,K,NS,X,BM,FM,IFF)
      IF(NPR.GE.4)WRITE(3,905)(II,FM(II,1),FM(II,2),II=1,N)
905   FORMAT(' FM=  ',I4,2D20.10)
C
      RINC=0.D0
      SINC=0.D0
C
C
C
C     GENERATE RINC,SINC
C
      DO 320 J=1,K
320   BDAV(J)=BDAV(J)+DABS(BM(J)-BD(J))
C
      DO 360 I=1,N
      BDIF(1)=0.D0
      BDIF(2)=0.D0
      DO 310 J=1,K
      DO 310 L=1,NS
      BDINC(L)=(BM(J)-BD(J))*DF(I,J,L)
      IF(NPR.GE.4)WRITE(3,906)BDINC
906   FORMAT('  BDINC= ',2D20.10)
      BDIF(L)=BDIF(L)+BDINC(L)
310   CONTINUE
C
      WT2=DCOS(X(I,1))**2
      FDIF(1)=FM(I,1)-FD(I,1)
      FDIF(2)=FM(I,2)-FD(I,2)
      RINC=RINC+(FDIF(1)-BDIF(1))**2+(FDIF(2)-BDIF(2))**2*WT2
      SINC=SINC+FDIF(1)**2+FDIF(2)**2*WT2
      IF(NPR.GE.3)WRITE(3,901)BDIF,FDIF,RINC,SINC
901   FORMAT(' BDIF1,2,FDIF1,2,RINC,SINC= ',6D15.7)
360   CONTINUE
C
C
C     ACCUMULATE NUMERATOR,DENOMINATOR AND RETURN FOR NEXT MM POINT
C
      S=S+SINC**2
      R=R+RINC
      IF(MM.GE.MMM)GO TO 400
      MM=MM+1
      GO TO 200
C
C     CALCULATE BEALE FOR THIS DISTRIUTION
C
400   QNORM=R/S
      EN=K
      EN=EN*SIGSQ*QNORM
      SDIS=DSQRT(S/MMM)
      SDIS=DSQRT(SDIS/(2.D0*N))/SEC
      WRITE(3,410)IT,SKALE,QNORM,R,S,SDIS,EN,SIGSQ,NU
410  _FORMAT(//' TEST',I4,' SKALE=',D13.5,' QNORM=',D16.6,' R,S,SDIS=',
     13D13.6/' **NTHETA= ',D16.6,' BASED ON SIGSQ(NU)=',D16.6,I4)
C
C     CALC + WRITE AVERAGE B-DIFFS
C
      BDAVG=0.D0
      DO 1060 J=1,K
       IF(J.EQ.3)GO TO 1060
      BDAVG=BDAVG+BDAV(J)
1060  CONTINUE
      BDAVG=BDAVG/(MMM*(K-1)*SEC)
      BDAVG3=BDAV(3)/(MMM*PIO)
C
      WRITE(3,1070) BDAVG,BDAVG3
1070 _FORMAT(' PARM DIFF AVG.(SEC)= ',D18.7,' PARM(3)DIFF DEGREES= ',
     1D18.7)
C
C     RETURN FOR NEXT DISTRIBUTION SKALE
C
1000  CONTINUE
      RETURN
      END
```

BESLYR

```
C     ELT=F BESLYR MOD=2
      SUBROUTINE BESLYR(YR,TJ)
C     CALCULATES J.D. OF BEGINNING OF BESSELLIAN YEAR ,
      DOUBLE PRECISION YR,TJ,TAU
      TAU=YR-1900.D0
      TJ=2415020.31352D0+TAU*(365.242198790D0-0.856D-8*TAU)
      RETURN
      END
```

CMPRES

```
C     ELT=N   CMPRES    MOD= 4.
      SUBROUTINE CMPRES(K,MASK,A,G,SCALE,MSUM)
C     CALCULATES MSUM=SUM OF MASKING FUNCTION & COMPRESSES MATRIX A  &
C     VECTORS G & SCALE    FROM K DIM. TO MSUM DIM. (IF MSUM<K).
C     N.B.  MASK(J)=0 FREEZES PARM(J).
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION MASK(K),A(K,K),G(K),SCALE(K)
      L=1
      MSUM=0
      DO 100 J=1,K
      IF(MASK(J).NE.0) MSUM=MSUM+1
100   CONTINUE
C
      IF(MSUM.EQ.K) GO TO 1000
      DO 10 I=1,MSUM
30    IF(MASK(L).NE.0) GO TO 40
      L=L+1
      GO TO 30
C
40    G(I)=G(L)
      SCALE(I)=SCALE(L)
      M=1
      DO 20 J=1,MSUM
50    IF(MASK(M).NE.0) GO TO 60
      M=M+1
      GO TO 50
C
C     SET A(J,I)=A(M,L)
60    INDEX=J+(I-1)*MSUM
      A(INDEX)=A(M,L)
C
20    M=M+1
10    L=L+1
C
1000  RETURN
      END
```

COMPOL

```
C     ELT=I    COMPOL  MOD NO.= 1
      SUBROUTINE COMPOL(N,K,KD,IW,X,P,ALPHA,JS,JT,JG,JQ,JL,JRA,JRB)
C     EVALUATES ORTHOGONAL POLYNOMIALS GIVEN MAXIMUM ORDER K
C     AND FORSYTHE COEFFICIENTS ALPHA.
C
      IMPLICIT  REAL*8(A-H,O-Z)
      DIMENSION X(N,2),P(N,KD,2),ALPHA(KD,KD,2)
      DIMENSION JS(KD),JT(KD),JG(KD),JQ(KD),JL(KD),JRA(KD),JRB(KD)
C
      J=0
      CALL ANABEL(K,KD,JS,JT,JG,JQ,JL,JRA,JRB)
      DO 10 I=1,N
      DO 10 M=1,2
10    P(I,KD,M)=1.D0
      GO TO 500
C
C     CALC. P(J)
100   KQ=JQ(J)
      IF(KQ.EQ.0) KQ=KD
      L=JL(J)
      JA=JRA(J)
      JB=JRB(J)
C
      DO 180 I=1,N
C     FORM SUM
      A=0.D0
      DO 110 JR=JA,JB
      JC=JR
      IF(JR.EQ.0) JC=KD
110   A=A+ALPHA(J,JC,M)*P(I,JC,M)
C
180   P(I,J,M)=X(I,L)*P(I,KQ,M)-A
C
C     CHECK IF W(I,1)=W(I,2).
200   IF(IW.EQ.0) GO TO 300
C     CHECK IF BOTH M DONE.
400   IF(M.EQ.2) GO TO 500
      M=M+1
      GO TO 100
C
C
300   DO 310 I=1,N
310   P(I,J,2)=P(I,J,1)
C
C     ARE ALL ORDERS FINISHED.
500   IF(J.EQ.K) GO TO 600
      M=1
      J=J+1
      GO TO 100
C
600   RETURN
      END
```

COORDS

```
C     ELT=Y    COORDS    MOD=1
      SUBROUTINE COORDS(RA,DEC,KH,KM,S,LSIGN,LD,LM,LS)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C     CONVERTS RADIAN RA,DEC TO SEPARATE HOURS(DEG.) MIN,SEC
C     TO PRECISION OF 0.1 SEC TIME        1 SEC ARC
C     LSIGN=1R- OR1R+ DEPENDING ON SIGN OF DEC
C
      PI=3.14159265358979300
      PIO=PI/180.D0
C
C
C     CONVERT COORDS.
      RAA=RA*12.D0/PI
      KH=RAA
      RAA=(RAA-KH)*60.D0
      KM=RAA
      S=(RAA-KM)*60.D0
      IF(S.LT.59.9D0) GO TO 50
      S=0.D0
      KM=KM+1
      IF(KM.NE.60) GO TO 50
      KM=0
      KH=KH+1
      IF(KH.NE.24) GO TO 50
      KH=0
50    CONTINUE
C
C
      LSIGN=1R+
      DECA=DEC/PIO
      IF(DECA.GE.0.D0)GO TO 60
      LSIGN=1R-
      DECA=-DECA
60    LD=DECA
      DECA=(DECA-LD)*60.D0
      LM=DECA
      DECA=(DECA-LM)*60.D0
      LS=(DECA+0.5D0)
      IF(DECA.LT.59.5D0) GO TO 70
      LS=0
      LM=LM+1
      IF(LM.NE.60) GO TO 70
      LM=0
      LD=LD+1
70    CONTINUE
      RETURN
      END
```

DATGEN (PEST)

```
C     ELT=B    DATGEN TEST DATA GENERATOR MOD=3.
      SUBROUTINE DATGEN(N,M,K,NS,NOPT    ,PARM,X,Y)
C     PROGRAM   1
C     TELESCOPE POINTING ERROR MODEL DATA GENERATOR
C
C     ----------------------------------------------------------OPTIONS
C      NOPTION(1)=0.........................
C     K CARDS CONTAIN PARM VECTOR
C
C     NOPT(2) GE 1 ENABLES PRINT..
C     --------------------------------------------------------------------
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION NOPT    (4)
      DIMENSION X(N,M),Y(N,NS),PARM(K)
      PI=3.14159265358979 3
      PIO=PI/180.D0
      NPR=NOPT(2)
C
C
      IF (NOPT    (1).EQ.0) GO TO 90
      RETURN
90    WRITE(3,91)K
91    FORMAT(' ENTER THE ',I4,'MODEL PARMS(1 PER RECORD)')
      DO 10  J=1,K
```

```
      CALL READ(MM,OP,MER)
      PARM(J)=OP
10    CONTINUE
C
C
C     GENERATE X-VECTOR IN INTERVAL DEC1..DEC2,    HA1..HA2.
20    CONTINUE
      DEC1=-80.*PIO
      DEC2=20.*PIO
      HA1=-1.
      HA2=1.
C     SELECT GRID SQT(N) SQUARE
      AN=N
      LN=DSQRT(AN)
      AN=LN
      DDEC=(DEC2-DEC1)/AN
      DHA=(HA2-HA1)/AN
      DO 30 I=1,LN
      AI=I
      DO 30 J=1,LN
      AJ=J
      II=I*LN+J-LN
      X(II,1)=DEC1+AJ*DDEC
      X(II,2)=HA1+AI*DHA
      IF(NPR.EQ.1) WRITE(3,601) AI,AJ,X(II,1),X(II,2)
601   FORMAT(' 601 DATGEN ',4D15.8)
30    CONTINUE
C     COMPLETE DATA POINTS
      LSTOP=N-LN**2
      IF(LSTOP.LT.1) GO TO 100
      DO 40 J=1,LSTOP
      AJ=J
      LL=J+LN**2
      X(LL ,1)=(DEC2-DEC1)/2.D0
      X(LL,2)=(HA2-HA1)/2.D0-AJ*DHA/10.D0
      IF(NPR.EQ.1)WRITE(3,630) AJ,X(LL,1),X(LL,2)
630   FORMAT('  630 DATGEN ',3D15.8)
40     CONTINUE
C
C
100   CONTINUE
C     CALL MODEL FUNCTION
      CALL FCTN(N,M,K,NS,X,PARM,Y,1)
C     CALC. PHYE
      PHYE=0.D0
      DO 120 I=1,N
      ARG=X(I,1)
      ARG=(Y(I,2)-X(I,2))**2*DCOS(ARG)**2
      PHYE=PHYE+(Y(I,1)-X(I,1))**2+ARG
120   CONTINUE
      RMS=DSQRT(PHYE/N)*3600.D0/PIO
C
C     PRINT
C
       WRITE(3,60)PARM
60    FORMAT('  POINTINGERROR DATA + DATA VALUES GEN. BY DATGEN'/'  PARM
     1S=',5(D15.8))
      WRITE(3,121) PHYE,RMS
121   FORMAT(' SUMSQUARE RES ERROR = ',D15.8,' RMS ARCSEC=',F10.4)
      DO 70 I=1,N
70    IF(NPR.NE.0)WRITE(3,61)I,X(I,1),X(I,2),Y(I,1),Y(I,2)
61    FORMAT(' X,Y,(',I3,')=',4D15.8)
      RETURN
      END
```

DATGEN (SURFIT)

```
C     ELT=R  DATGEN MOD # 6
      SUBROUTINE DATGEN(N,K,KD,NOPT,X,XX,Y,RA,RB,RC)
C     PROGRAM 1
C     TELESCOPE POINTING ERROR DATA SIMULATOR (RANDOM VERSION)
C
C     .........................................,OPTIONS....
C     FIRST CARD IS 5 MODEL PARMS FOR MODEL AS IN PE.
C     SECOND CONTAINS P-P PERTURBATION AMPLITUDE IN ARCSEC.
C
C     NOPT(1) GE 0 ENABLES PRINTING. IF =9 X DISTRIBUTION IS PLOTTED
C     NOPT(2)=0 GENERATES 5 PARM MODEL,=1 GENERATES POLYNOMIAL TEST
C             =2 READS IN PERT AMPLITUDE & PERTURBS EXISTING DATA.
C     ...............................................................
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION NOPT(4),X(N,2),XX(N,2),Y(N,2)
      DIMENSION W(2)
      LOGICAL NS
      REAL KXA(200),KXB(200)
      PI=3.14159265358979300
      PIO=PI/180.D0
      SEC=3600.D0/PIO
      NPR=NOPT(1)
      KXA(1)=RA
      KXB(1)=RB
C
C
C     PERT ONLY IF NOPT(2)=2
      IF(NOPT(2).EQ.2) GO TO 300
C
C
C     GENERATE X VECTOR IN RANGE DEC1..DEC2 , HA1..HA2
      DEC1=-80.*PIO
      DEC2=20.*PIO
      HA1=-1.
      HA2=1.
C
      CALL RANDU(KXA,N)
      CALL RANDU(KXB,N)
      DO 100 I=1,N
      X(I,1)=DEC1+KXA(I)*(DEC2-DEC1)
      X(I,2)=HA1+KXB(I)*(HA2-HA1)
100   CONTINUE
C
C
      IF(NOPT(2).GE.1)GO TO 200
C     CALL MODEL FCTN.
      CALL FCTN(N,NOPT,X,Y)
      GO TO 270
C
C     CALL POLYNOMIAL TEST GENERATOR.
200   CONTINUE
      CALL PLTEST(N,NOPT,X,XX,Y)
C
C
270   IF(NPR.LT.1)GO TO 295
      DO 290 I=1,N
290   WRITE(3,280) X(I,1),X(I,2),Y(I,1),Y(I,2),KXA(I),KXB(I)
280   FORMAT(' X,Y,RANDS= ',4D15.8,2E15.8)
C     CALCULATE PHI
295   PHI=0.D0
      DO 120 I=1,N
      ARG=X(I,1)
      ARG=(Y(I,2)-X(I,2))**2*DCOS(ARG)**2
      PHI=PHI+(Y(I,1)-X(I,1))**2+ARG
120   CONTINUE
      RMS=DSQRT(PHI/N)*SEC
      WRITE(3,60)PHI,RMS
60    FORMAT(' TEST DATA GENERATED WITH RES SUM SQRS,RMS(SEC)=',D15.8
     1,3X,D15.8/' ........INPUT PERT AMP IN ARCSEC.')
C
C
C     DATA PERTURBATION FOLLOWS.
300    CONTINUE
       WRITE(3,301)
301    FORMAT(' ENTER PERT AMPLITUDE IN ARCSEC (ANY FMT).')
C
       CALL READ(NVAR,PERT,NERR)
       IF(NERR.EQ.1.OR.NVAR.NE.1)GO TO 999
       IF(PERT.LT.1.D-10)GO TO 800
C
       APERT=PERT*PI/(180.D0*3600.D0)
       KXA(1)=RC
       NN=N*2
       CALL RANDU(KXA,NN)
       PHI=0.D0
       DO 600 I=1,N
       W(1)=1.D0
       W(2)=DCOS(X(I,1))
       DO 600 J=1,2
       IRAND=(I-1)*2+J
       YSHIFT=(KXA(IRAND)-0.5D0)*APERT
       Y(I,J)=Y(I,J)+YSHIFT
       PHI=PHI+YSHIFT**2*W(J)**2
       IF(NPR.NE.0) WRITE(3,601)I,J,Y(I,J),YSHIFT
601    FORMAT(' YIJ= ',2I4,5X,D15.8,'  (',D15.8,') ')
600    CONTINUE
       RMS=DSQRT(PHI/N)*SEC
C
       WRITE(3,602)PERT,PHI,RMS
C
C      CALC. FINAL PHI
       PHI=0.D0
        DO 620 I=1,N
       WW=DCOS(X(I,1))
       PHI=PHI+(Y(I,1)-X(I,1))**2
        PHI=PHI+(Y(I,2)-X(I,2))**2*WW**2
620    CONTINUE
       RMS=DSQRT(PHI/N)*SEC
       WRITE(3,630) PHI,RMS
630    FORMAT(' FINAL PHI,RMS OF DATA=',D15.8,D15.8)
C
       GO TO 810
602    FORMAT(' DATA PERTURBED ',D13.6,'ARCSEC.(P-P),PHI INTRO=',
      1D15.8,'RMS ERROR(SEC)=',D15.8)
C
C
800     WRITE(3,820)
820   FORMAT(' NO DATA PERTURBATION')
C
C
810   CONTINUE
C
C
C     OPTIONAL PLOT OF X DATA
      IF(NPR.NE.9) GO TO 880
      NS=.FALSE.
      DO 850 I=1,N
      XX(I,1)=X(I,1)
850   XX(I,2)=X(I,2)
      CALL PLOTLP(1,XX,N,2,1,NS,'*')
880   CONTINUE
      RETURN
C
C     READ FAIL
999   WRITE(3,998)NVAR
998   FORMAT(' ILLEGAL CARD INPUT,NO. OF VARIABLES READ=',I4)
      END
```

DAPERT

```
C     ELT=G , DAPERT DATA PERTURBATION ROUTINE  MOD# 4.
      SUBROUTINE DAPERT(N,M,NS,X,Y,W,NOPT)
C     PROGRAM 5
C     TAKES DATA Y(N,NS) PERTURBS IT WITH  STANDARD DEVIATION NOPT  ARC
C
      REAL*8 Y(N,NS),X(N,M),W(N,NS),PERT,YSHIFT,PHY,STD,RMS,SEC
      REAL YRAND,PERTR
      DIMENSION NOPT(4)
      SEC=3.14159265360D0/(180.D0*3600.D0)
C
      NPR=0
C
      PHY=0.D0
      CALL WT(N,M,NS,X,W)
C     OBTAIN PERT
      PERT=NOPT(1)*1000+NOPT(2)*100+NOPT(3)*10+NOPT(4)
      PERTR=PERT*SEC
      WRITE(3,20) PERTR,PERT
20    FORMAT(//' DATA PERTURBATION ROUTINE..Y(N,NS) PERTURBED WITH'
     1' STANDARD DEV.OF',E15.8,F10.4,' RAD,ASEC FOLLOWS')
      IRAND=791253
      DO 50 I=1,N
      DO 50 J=1,NS
      CALL GAUSS(IRAND,PERTR,0.0,YRAND)
      YSHIFT=YRAND/W(I,J)
      Y(I,J)=Y(I,J)+YSHIFT
      PHY=PHY+YSHIFT**2*W(I,J)**2
      IF(NPR.NE.0)WRITE(3,30) I,J,Y(I,J),YSHIFT
30    FORMAT(/2I4,5X,D15.8,'     (',D15.8,' )')
50    CONTINUE
      STD=DSQRT(PHY/(2.D0*N-1.D0))
      RMS=DSQRT(PHY/N)/SEC
C
      WRITE(3,60) PHY,STD,RMS
60    FORMAT(/'  SUMSQUARE OF PERTS GEN BY DAPERT=', D15.8,' STDEV.RMS='
     1D15.8,F10.4)
      RETURN
      END
```

DERV (5 PARM.)

```
C     ELT=DS , 5-PARM MODEL DERIVATIVES(COMPATIBLE WITH NEW)  MOD=2.
      SUBROUTINE DERV(N,M,K,NS,XX,PARM,DF,IDERV)
C     CALC. TELESCOPE POINTING ERROR FN DERIVATIVES WRT MODEL PARAMETERS
C      MISALIGNED,SKEW FOTL. TELESCOPE WITH ZERO OFFSETS.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION  XX(N,M),DF(N,K,NS),PARM(K)
C     CAUTION TEMPORARY DIMENSIONED VARIABLES..
      DIMENSION  DEC(160),HA(160),GG( 5),FF( 5)
      DEFINE DARCOS(XDUM)=DACOS(XDUM)
       DEFINE DARSIN(XDUM)=DASIN(XDUM)
C
C
      IDERV=IDERV+1
      NPR=0
      THETA=PARM(1)
      PSI=PARM(2)-PARM(3)
      PHY=PARM(3)
      ETA=PARM(4)
      DEE=PARM(5)
C
      ST=DSIN(THETA)
       CT=DCOS(THETA)
      SP=DSIN(PSI)
      CP=DCOS(PSI)
      SE=DSIN(ETA)
      CE=DCOS(ETA)
      DO 150 J=1,N
      DEC(J)=XX(J,1)
      HA(J)=XX(J,2)
      DEK=DARSIN(DSIN(DEC(J)+DEE)*CE)
      HAK=HA(J)+DATAN(DTAN(DEC(J)+DEE)*SE)
      CPH=DCOS(PHY+HAK)
      SPH=DSIN(PHY+HAK)
      X=DCOS(DEK)*(CP*CPH-CT*SP*SPH)+DSIN(DEK)*ST*SP
      Y=-DCOS(DEK)*(SP*CPH+CT*CP*SPH)+DSIN(DEK)*ST*CP
      Z=DCOS(DEK)*ST*SPH+CT*DSIN(DEK)
      CD=DCOS(DEK)
      SD=DSIN(DEK)
      FAC=1.D0/(X**2+Y**2)
      DEK=DARSIN(Z)
      HAK=DATAN(-Y/X)
C
      IF(NPR.EQ.1) WRITE(3,625) DEC(J),HA(J),DEK,HAK
625   FORMAT(' 625 DERV ',4D15.8)
      FAK=1.D0/DSQRT(1.D0-Z**2)
C
      GG(1)=FAK*(CD*SPH*CT-ST*SD)
      GG(2)=0.0
      GG(3)=FAK*CD*ST*CPH
C     (DEFINE SUB SET OF PARTIALS..)
      PAC=1.D0/DSQRT(1.D0-DSIN(DEC(J)+DEE)**2*CE**2)
      PAK=1.D0/(1.D0+DTAN(DEC(J)+DEE)**2*SE**2)
      GDD=-DSIN(DEC(J)+DEE)*SE*PAC
      FDD=PAK*DTAN(DEC(J)+DEE)*CE
      GED=PAC*DCOS(DEC(J)+DEE)*CE
      FED=PAK*SE/DCOS(DEC(J)+DEE)**2
C
      GD=(CT*CD-ST*SPH*SD)*GDD+CD*ST*CPH*FDD
      GG(4)=GD*FAK
      GE=(CT*CD-ST*SD*SPH )*GED+CD*ST*CPH*FED
      GG(5)=GE*FAK
C
C
      DX=CD*SP*SPH*ST+SD*SP*CT
      DY=CD*CP*SPH*ST+SD*CP*CT
      FF(1)=FAC*(Y*DX-X*DY)
      DX=-CD*(CPH*SP+CT*CP*SPH)+SD*ST*CP
      DY=-CD*(CP*CPH-CT*SP*SPH)-SD*ST*SP
      FF(2)=FAC*(Y*DX-X*DY)
      DX=-CD*(CP*SPH+CT*SP*CPH)
      DY= CD*(SP*SPH-CT*CP*CPH)
      FF(3)=FAC*(Y*DX-X*DY)-FF(2)
C
      XG=ST*SP*CD-SD*(CP*CPH-CT*SP*SPH)
      XF=CD*(CP*SPH+CT*SP*CPH)
      YG=SD*(SP*CPH+CT*CP*SPH)+CD*ST*CP
      YF=CD*(-SP*SPH+CT*CP*CPH)
      DX=XG*GDD-XF*FDD
      DY=YG*GDD-YF*FDD
      FF(4)=FAC*(Y*DX-X*DY)
      DX=XG*GED-XF*FED
      DY=YG*GED-YF*FED
      FF(5)=FAC*(Y*DX-X*DY)
      DO 150 KK=1,K
      DF(J,KK,1)=GG(KK)
      DF(J,KK,2)=FF(KK)
150   CONTINUE
      IF(NPR.NE.1) GO TO 627
      WRITE(3,626) ((DF(J,KK,1),KK=1,K),J=1,K)
      WRITE(3,626) ((DF(J,KK,2),KK=1,K),J=1,K)
626   FORMAT(' 626 DERV ',5D15.8)
627   RETURN
      END
```

DERV (74 INCH.)

```
C     ELT=D        EXTENDED 74" MODEL DERIVATIVES    MOD=E9
      SUBROUTINE DERV(N,M,K,NS,XX,B,DF,IDERV)
C     CALC. TELESCOPE POINTING ERROR FN DERIVATIVES WRT MODEL PARAMETERS.
C      MISALIGNED,SKEW FOTL. TELESCOPE WITH ZERO OFFSETS.
C     PERIODIC ENCODER GEAR ERRORS + STRUCTURAL FLEXURE.
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION  XX(N,M),DF(N,K,NS),B(K),GG(30),FF(30)
C
C
      IDERV=IDERV+1
      NPR=0
      PI=3.14159265358979300
      PIO=PI/180.D0
      PHI=-(35.D0+19.D0/60.D0+17.5D0/3600.D0)*PIO
```

```
      THETA=B(1)
      PHY=B(3)
      PSI=B(2)-B(3)
C
      ST=DSIN(THETA)
       CT=DCOS(THETA)
      SP=DSIN(PSI)
      CP=DCOS(PSI)
C
      ACONST=15.DD*PI
      A1=9.DO/14.DO
      A2=70.DO*PIO
      A3=0.85DO
      A4=14.2DO/ACONST
      A5=4.8DO/ACONST
      A6=1.28DO
      A7=10.DO/ACONST
C
      PDA=1512.DO
      PDB=PDA/32.DO
      PHA=864.DO
      PHB=PHA/32.DO
C
C
      DO 150 J=1,N
      DEC=XX(J,1)
      HA=XX(J,2)
      CPH=DCOS(PHY+HA)
      SPH=DSIN(PHY+HA)
      CD=DCOS(DEC)
      SD=DSIN(DEC)
      CH=DCOS(HA)
      CPDA=DCOS(PDA*DEC)
      SPDA=DCOS(PDA*DEC)
      CPDB=DCOS(PDB*DEC)
      SPDB=DCOS(PDB*DEC)
      CPHB=DCOS(PHB*HA)
      SPHB=DCOS(PHB*HA)
      CPHA=DCOS(PHA*HA)
      SPHA=DCOS(PHA*HA)
      SH=DSIN(HA)
      X= CD*(CP*CPH-CT*SP*SPH)+SD*ST*SP
      Y=- CD*(SP*CPH+CT*CP*SPH)+SD*ST*CP
      Z= CD*ST*SPH+CT*SD
      FAC=1.DO/(X**2+Y**2)
C
C
      FAK=1.DO/DSQRT(1.DO-Z**2)
C
      GG(1)=FAK*(CD*SPH*CT-ST*SD)
      GG(2)=0.0
      GG(3)=FAK*CD*ST*CPH
C
      DX=CD*SP*SPH*ST+SD*SP*CT
      DY=CD*CP*SPH*ST+SD*CP*CT
      FF(1)=FAC*(Y*DX-X*DY)
      DX=-CD*(CPH*SP+CT*CP*SPH)+SD*ST*CP
      DY=-CD*(CP*CPH-CT*SP*SPH)-SD*ST*SP
      FF(2)=FAC*(Y*DX-X*DY)
      DX=-CD*(CP*SPH+CT*SP*CPH)
      DY= CD*(SP*SPH-CT*CP*CPH)
      FF(3)=FAC*(Y*DX-X*DY)-FF(2)
C
C
C     TWO
      GG(4)=-SD*DSIN(B(4))/DSQRT(1.DO-SD**2*DCOS(B(4))**2)
      FF(4)=DTAN(DEC)*DCOS(B(4))/(1.DO+DTAN(DEC)**2*DSIN(B(4))**2)
C
C     THREE
      GG(5)=1.DO
      GG(6)=CD
      GG(7)=SD
      GG(8)=CPDA
      GG(9)=SPDA
      GG(10)=CPDB
      GG(11)=SPDB
      DO 160 KK=12,19
160   GG(KK)=0.DO
      DO 170 KK=5,11
170.  FF(KK)=0.DO
      FF(12)=CH
      FF(13)=SH
      FF(14)=CPHA
      FF(15)=SPHA
      FF(16)=CPHB
      FF(17)=SPHB
C
C     FOUR
      GG(20)=DCOS(PHI)*CH*SD-DSIN(PHI)*CD
      FF(18)=DCOS(PHI)*CD*SH
      FF(20)=0.DO
C
C     FIVE
      GG(21)=DCOS(A1*DEC*A2)*DCOS(A3*HA)
      FF(21)=0.DO
      FF(19)=(A4+A5*DSIN(A6*DEC))*HA-A7*DEC
C
C
      DO 150 KK=1,K
      DF(J,KK,1)=GG(KK)
      DF(J,KK,2)=FF(KK)
150   CONTINUE
      IF(NPR.NE.1) GO TO 627
      WRITE(3,626) ((DF(J,KK,1),KK=1,K),J=1,N)
      WRITE(3,626) ((DF(J,KK,2),KK=1,K),J=1,N)
626   FORMAT(' 626 DERV ',5D15.8)
627   RETURN
      END
```

DERV (74 INCH EXACT)

```
C     ELT=D  EXTENDED 74"(EXACT)MODEL DERIVATIVES    MOD=2F
      SUBROUTINE DERV(N,M,K,NS,XX,B,DF,IDERV)
C     CALC. TELESCOPE POINTING ERROR FN DERIVATIVES WRT MODEL PARAMETERS
C      MISALIGNED,SKEW EQTL. TELESCOPE WITH ZERO OFFSETS.
C     PERIODIC ENCODER GEAR ERRORS + STRUCTURAL FLEXURE.
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION  XX(N,M),DF(N,K,NS),B(K),DP(30),HP(30)
C
C
      IDERV=IDERV+1
      NPR=0
      PI=3.14159265358979300
      PIO=PI/180.DO
      PHI=-(35.DO+19.DO/60.DO+17.5DO/3600.DO)*PIO
      THETA=B(1)
      PHY=B(3)
      PSI=B(2)-B(3)
C
      ST=DSIN(THETA)
       CT=DCOS(THETA)
      SP=DSIN(PSI)
      CP=DCOS(PSI)
C
      ACONST=15.DO*PI
      A1=9.DO/14.DO
      A2=70.DO*PIO
      A3=0.85DO
      A4=14.2DO/ACONST
      A5=4.8DO/ACONST
      A6=1.28DO
      A7=10.DO/ACONST
C
      PDA=1512.DO
      PDB=PDA/32.DO
      PHA=864.DO
      PHB=PHA/32.DO
C
C
      DO 150 J=1,N
      DEC=XX(J,1)
      HA=XX(J,2)
C
C     ONE      ENCODER ERRORS
C
      CD=DCOS(DEC)
      SD=DSIN(DEC)
      CH=DCOS(HA)
      CPDA=DCOS(PDA*DEC)
      SPDA=DCOS(PDA*DEC)
      CPDB=DCOS(PDB*DEC)
      SPDB=DCOS(PDB*DEC)
      CPHB=DCOS(PHB*HA)
      SPHB=DCOS(PHB*HA)
      CPHA=DCOS(PHA*HA)
      SPHA=DCOS(PHA*HA)
      SH=DSIN(HA)
      DO 810 I=1,K
      DF(J,I,1)=0.DO
810   DF(J,I,2)=0.DO
      DF(J,5,1)=1.DO
      DF(J,6,1)=CD
      DF(J,7,1)=SD
      DF(J,8,1)=CPDA
      DF(J,9,1)=SPDA
      DF(J,10,1)=CPDB
      DF(J,11,1)=SPDB
      DF(J,12,2)=CH
      DF(J,13,2)=SH
      DF(J,14,2)=CPHA
      DF(J,15,2)=SPHA
      DF(J,16,2)=CPHB
      DF(J,17,2)=SPHB
C
      DEC=DEC+B(5)+B(6)*CD+B(7)*SD+B(8)*CPDA+B(9)*SPDA
     1+B(10)*CPDB+B(11)*SPDB
      HA=HA+B(12)*CH+B(13)*SH+B(14)*CPHA+B(15)*SPHA
     1+B(16)*CPHB+B(17)*SPHB
C
C     TWO      TORSIONAL MOV'T OF AXES
C
      CD=DCOS(DEC)
      SD=DSIN(DEC)
      CH=DCOS(HA)
      SH=DSIN(HA)
      TD=DTAN(DEC)
      SDP=DSIN(DEC-PHI)
      CPHI=DCOS(PHI)
      DD=1.DO+B(20)*CH*DCOS(DEC-PHI)
      DH=-B(20)*SH*SDP
      HD=-B(18)*SH*SD*CPHI
      HH=1.DO-B(22)*SH*CPHI
      DO 820 I=1,K
      DP(I)=0.DO
820   HP(I)=0.DO
      HP(22)=CH*CPHI
      HP(18)=CD*SH*CPHI
      DP(20)=CH*SDP
      DO 821 I=1,K
      DF(J,I,1)=DF(J,I,1)*DD+DF(J,I,2)*DH+DP(I)
821   DF(J,I,2)=DF(J,I,1)*HD+DF(J,I,2)*HH+HP(I)
C
      DEC=DEC+B(20)*CH*SDP
      HA=HA+(B(22)*CH+B(18)*CD*SH)*CPHI
C
C     THREE    SKEWNESS OF AXES
C
      CD=DCOS(DEC)
      SD=DSIN(DEC)
      TD=DTAN(DEC)
      CH=DCOS(HA)
      SH=DSIN(HA)
      C4=DCOS(B(4))
      S4=DSIN(B(4))
      TANN=1.DO+TD**2*S4**2
      ROOT=DSQRT(1.DO-SD**2*C4**2)
      DH=0.DO
      DD=C4*CD/ROOT
      HH=1.DO
      HD=S4/(CD**2*TANN)
      DO 831 I=1,K
      DP(I)=0.DO
831   HP(I)=0.DO
      DP(4)=-SD*S4/ROOT
      HP(4)=TD*C4/TANN
      DO 832 I=1,K
      DF(J,I,1)=DF(J,I,1)*DD+DF(J,I,2)*DH+DP(I)
832   DF(J,I,2)=DF(J,I,1)*HD+DF(J,I,2)*HH+HP(I)
      DEC=DASIN(SD*C4)
      HA=HA+DATAN(TD*S4)
C
C     FOUR   MISALIGNMENT FROM POLE
C
      CPH=DCOS(PHY+HA)
      SPH=DSIN(PHY+HA)
      CD=DCOS(DEC)
      SD=DSIN(DEC)
      CH=DCOS(HA)
      SH=DSIN(HA)
      X=CD*(CP*CPH-CT*SP*SPH)+SD*ST*SP
      Y=-CD*(SP*CPH+CT*CP*SPH)+SD*ST*CP
      Z=CD*ST*SPH+CT*SD
C
      DO 840 I=1,K
      DP(I)=0.DO
840   HP(I)=0.DO
C
      FAC=1.DO/(X**2+Y**2)
      FAK=1.DO/DSQRT(1.DO-Z**2)
C
      DP(1)=FAK*(CD*SPH*CT-ST*SD)
      DP(2)=0.DO
      DP(3)=FAK*CD*ST*CPH
C
      DX=CD*SP*SPH*ST+SD*SP*CT
      DY=CD*CP*SPH*ST+SD*CP*CT
      HP(1)=FAC*(Y*DX-X*DY)
      DX=-CD*(CPH*SP+CT*CP*SPH)+SD*ST*CP
      DY=-CD*(CP*CPH-CT*SP*SPH)-SD*ST*SP
      HP(2)=FAC*(Y*DX-X*DY)
      DX=-CD*(CP*SPH+CT*SP*CPH)
      DY= CD*(SP*SPH-CT*CP*CPH)
      HP(3)=FAC*(Y*DX-X*DY)-HP(2)
C
      DD=FAK*(CT*CD-ST*SPH*SD)
      DH=FAK*CD*ST*CPH
      DX=CD*(-CP*SPH-CT*SP*CPH)
      DY=-CD*(-SP*SPH+CT*CP*CPH)
      HH=FAC*(Y*DX-X*DY)
      DX=-SD*(CP*CPH-CT*SP*SPH)+CD*ST*SP
      DY=SD*(SP*CPH+CT*CP*SPH)+CD*ST*CP
      HD=FAC*(Y*DX-X*DY)
      DO 841 I=1,K
      DF(J,I,1)=DF(J,I,1)*DD+DF(J,I,2)*DH+DP(I)
841   DF(J,I,2)=DF(J,I,1)*HD+DF(J,I,2)*HH+HP(I)
C
      HA=DATAN(-Y/X)
      DEC=DASIN(Z)
C
C     FIVE   FLEXURE OF TUBE AND OPTIC SUPPORTS
C
      DD=1.DO-A1*B(21)*DCOS(A3*HA)*DSIN(A1*DEC+A2)
      DH=-A3*B(21)*DCOS(A1*DEC+A2)*DSIN(A3*HA)
      HH=1.DO+B(19)*(A4+A5*DSIN(A6*DEC))
      HD=B(19)*(A5*A6*HA*DCOS(A6*DEC)-A7)
      DO 850 I=1,K
      DP(I)=0.DO
850   HP(I)=0.DO
      DP(21)=DCOS(A1*DEC+A2)*DCOS(A3*HA)
      HP(19)=(A4+A5*DSIN(A6*DEC))*HA-A7*DEC
      DO 851 I=1,K
      DF(J,I,1)=DF(J,I,1)*DD+DF(J,I,2)*DH+DP(I)
851   DF(J,I,2)=DF(J,I,1)*HD+DF(J,I,2)*HH+HP(I)
C
C     ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
150   CONTINUE
      IF(NPR.NE.1) GO TO 627
      WRITE(3,626) ((DF(J,KK,1),KK=1,K),J=1,N)
      WRITE(3,626) ((DF(J,KK,2),KK=1,K),J=1,N)
626   FORMAT(' 626 DERV ',5D15.8)
627   RETURN
      END
```

DGELG

```
C
C        PURPOSE
C           TO SOLVE A GENERAL SYSTEM OF SIMULTANEOUS LINEAR EQUATIONS.
C
C        USAGE
C           CALL DGELG(R,A,M,N,EPS,IER)
C
C        DESCRIPTION OF PARAMETERS
C           R      - DOUBLE PRECISION M BY N RIGHT HAND SIDE MATRIX
C                    (DESTROYED). ON RETURN R CONTAINS THE SOLUTIONS
C                    OF THE EQUATIONS.
C           A      - DOUBLE PRECISION M BY M COEFFICIENT MATRIX
C                    (DESTROYED).
C           M      - THE NUMBER OF EQUATIONS IN THE SYSTEM.
C           N      - THE NUMBER OF RIGHT HAND SIDE VECTORS.
C           EPS    - SINGLE PRECISION INPUT CONSTANT WHICH IS USED AS
C                    RELATIVE TOLERANCE FOR TEST ON LOSS OF
C                    SIGNIFICANCE.
C           IER    - RESULTING ERROR PARAMETER CODED AS FOLLOWS
C                    IER=0  - NO ERROR,
C                    IER=-1 - NO RESULT BECAUSE OF M LESS THAN 1 OR
C                             PIVOT ELEMENT AT ANY ELIMINATION STEP
C                             EQUAL TO 0,
C                    IER=K  - WARNING DUE TO POSSIBLE LOSS OF SIGNIFI-
C                             CANCE INDICATED AT ELIMINATION STEP K+1,
C                             WHERE PIVOT ELEMENT WAS LESS THAN OR
C                             EQUAL TO THE INTERNAL TOLERANCE EPS TIMES
C                             ABSOLUTELY GREATEST ELEMENT OF MATRIX A.
C
C        REMARKS
C           INPUT MATRICES R AND A ARE ASSUMED TO BE STORED COLUMNWISE
C           IN M*N RESP. M*M SUCCESSIVE STORAGE LOCATIONS. ON RETURN
C           SOLUTION MATRIX R IS STORED COLUMNWISE TOO.
C           THE PROCEDURE GIVES RESULTS IF THE NUMBER OF EQUATIONS M IS
C           GREATER THAN 0 AND PIVOT ELEMENTS AT ALL ELIMINATION STEPS
C           ARE DIFFERENT FROM 0. HOWEVER WARNING IER=K - IF GIVEN -
C           INDICATES POSSIBLE LOSS OF SIGNIFICANCE. IN CASE OF A WELL
C           SCALED MATRIX A AND APPROPRIATE TOLERANCE EPS, IER=K MAY BE
C           INTERPRETED THAT MATRIX A HAS THE RANK K. NO WARNING IS
C           GIVEN IN CASE M=1.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           SOLUTION IS DONE BY MEANS OF GAUSS-ELIMINATION WITH
C           COMPLETE PIVOTING.
C
C     ..................................................................
C
      SUBROUTINE DGELG(R,A,M,N,EPS,IER)
C
C
      DIMENSION A(1),R(1)
      DOUBLE PRECISION R,A,PIV,TB,TOL,PIVI
      IF(M)23,23,1
C
C        SEARCH FOR GREATEST ELEMENT IN MATRIX A
```

```
    1 IER=0
      PIV=0.D0
      MM=M*M
      NM=N*M
      DO 3 L=1,MM
      TB=DABS(A(L))
      IF(TB-PIV)3,3,2
    2 PIV=TB
      I=L
    3 CONTINUE
      TOL=EPS*PIV
C     A(I) IS PIVOT ELEMENT. PIV CONTAINS THE ABSOLUTE VALUE OF A(I).
C
C
C     START ELIMINATION LOOP
      LST=1
      DO 17 K=1,M
C
C     TEST ON SINGULARITY
      IF(PIV)23,23,4
    4 IF(IER)7,5,7
    5 IF(PIV-TOL)6,6,7
    6 IER=K-1
    7 PIVI=1.D0/A(I)
      J=(I-1)/M
      I=I-J*M-K
      J=J+1-K
C     I+K IS ROW-INDEX, J+K COLUMN-INDEX OF PIVOT ELEMENT
C
C     PIVOT ROW REDUCTION AND ROW INTERCHANGE IN RIGHT HAND SIDE R
      DO 8 L=K,NM,M
      LL=L+I
      TB=PIVI*R(LL)
      R(LL)=R(L)
    8 R(L)=TB
C
C     IS ELIMINATION TERMINATED
      IF(K-M)9,18,18
C
C     COLUMN INTERCHANGE IN MATRIX A
    9 LEND=LST+M-K
      IF(J)12,12,10
   10 II=J*M
      DO 11 L=LST,LEND
      TB=A(L)
      LL=L+II
      A(L)=A(LL)
   11 A(LL)=TB
C
C     ROW INTERCHANGE AND PIVOT ROW REDUCTION IN MATRIX A
   12 DO 13 L=LST,MM,M
      LL=L+I
      TB=PIVI*A(LL)
      A(LL)=A(L)
   13 A(L)=TB
C
C     SAVE COLUMN INTERCHANGE INFORMATION
      A(LST)=J
C
C     ELEMENT REDUCTION AND NEXT PIVOT SEARCH
      PIV=0.D0
      LST=LST+1
      J=0
      DO 16 II=LST,LEND
      PIVI=-A(II)
      IST=II+M
      J=J+1
      DO 15 L=IST,MM,M
      LL=L-J
      A(L)=A(L)+PIVI*A(LL)
      TB=DABS(A(L))
      IF(TB-PIV)15,15,14
   14 PIV=TB
      I=L
   15 CONTINUE
      DO 16 L=K,NM,M
      LL=L+J
   16 R(LL)=R(LL)+PIVI*R(L)
   17 LST=LST+M
C     END OF ELIMINATION LOOP
C
C
C     BACK SUBSTITUTION AND BACK INTERCHANGE
   18 IF(M-1)23,22,19
   19 IST=MM+M
      LST=M+1
      DO 21 I=2,M
      II=LST-I
      IST=IST-LST
      L=IST-M
      L=A(L)+.5D0
      DO 21 J=II,NM,M
      TB=R(J)
      LL=J
      DO 20 K=IST,MM,M
      LL=LL+1
   20 TB=TB-A(K)*R(LL)
      K=J+L
      R(J)=R(K)
   21 R(K)=TB
   22 RETURN
C
C
C     ERROR RETURN
   23 IER=-1
      RETURN
      END
```

DLSQ

```
C     ELT.PROG = DLSQ. LEVENBERG ALGORITHM   MODIFICATION NO. = 4A
      SUBROUTINE DLSQ(N,M,K,NS,NOPT    ,X,Y,PARM,FN,DF,W,A,ATEMP,G,GTEMP,
     1T,SCALE,MASK)
C     PROGRAM 2
C     LEVENBERG DAMPED LEAST SQUARES PARM ESTIMATION.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(N,M),Y(N,NS),FN(N,NS),DF(N,K,NS)
      DIMENSION   PARM(K),A(K,K),G(K),ATEMP(K,K),GTEMP(K),MASK(K)
      DIMENSION   W(N,NS),T(K)
      DIMENSION DATA(2),NOPT    (4),SCALE(K)
C
C     -------------------------------------------------------OPTIONS
C     NOPTION(1)=0..............................
C     CARD 1 = INITIAL PARM VALUE + CORRES. MASK,ETC.
C
C     NOPT(2) GE 1 ENABLES PRINT
C
C     -------------------------------------------------------------
C
      KOUNT1=0
      KOUNT2=0
      PHIO=1.D30
      ITER=1
      EPS=1.D-5
      TAU=1.D-15
      NPR=NOPT(2)
C
      ITERSW=10
C
      PI=3.14159265358979
      PIO=PI/180.D0
      SEC=PI/(180.D0*3600.D0)
C     ....................................................DATA INP
      IF (NOPT    (1).EQ.0) GO TO 11
      RETURN
C
11    CONTINUE
      WRITE(3,112)K
112   FORMAT(' ENTER',I4,' PARMS AND MASKS LIKE A GOOD BASTARD')
      DO 10 KK=1,K
      CALL READ(NN,DATA,NERR)
      PARM(KK)=DATA(1)
      MASK(KK)=DATA(2)
10    CONTINUE
C
C
C     ....................................................START
111   CONTINUE
      CALL WT(N,M,NS,X,W)
C     ENTRY $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
100   CONTINUE
C     CALC. PHI
      CALL FCTN(N,M,K,NS,X,PARM,FN,KOUNT1)
      CALL PHI(N,NS,Y,FN,W,PHIN,WORST,IWORST)
      RMS=DSQRT(PHIN/N)/SEC
C
      WRITE(3,9) ITERSW,ITER,ALAMDA,PHIN,RMS
      DO 7 J=1,K
7     WRITE(3,8) PARM(J),MASK(J)
9     FORMAT('   LEVENBERG DLSQ ANALYSIS , ITERSW=',I5,'BEGIN ITER NO.',I
     15,' LAM= ',D12.5,' PHI,RMS=',D15.8,F10.4/
     2'   CURRENT PARMS (+ MASKS) ARE....')
8     FORMAT(1X,D15.8,I4)
C
C
C     CHECK AGAINST PREVIOUS PHI
      IF(PHIN.GE.PHIO) GO TO 20
101   CONTINUE
      PHIO=PHIN
C
C
C
C     GENERATE MATRIX A.  AND  G.
      CALL DFRV(N,M,K,NS,X,PARM,DF,KOUNT2)
      SCALE(1)=9.0
      CALL MATEQN(N,M,K,NS,Y,FN,W,DF,A,G,SCALE)
C
C     IS ITER NO. GT. ITERSWITCH
      IF(ITER.GT.ITERSW) GO TO 40
C     CALC. OPTIMUM VALUE FOR ALAMDA
      GSUM=0.D0
      DO 30 J=1,K
30     GSUM=GSUM+G(J)**2
      ALAMDA=2.D0*GSUM/PHIO
      GO TO 50
C
C     TAYLOR CONVERGENT AREA
40     ALAMDA=0.D0
C
C     IF PARMS MASKED CONDENSE A*,G*
C      ABSENT.....
50    CONTINUE
C
C     SOLVE EQN FOR CORRECTION VECTOR
      DO 70 J=1,K
      GTEMP(J)=G(J)
      DO 70 I=1,K
      ATEMP(I,J)=A(I,J)
      IF(I.EQ.J) ATEMP(I,J)=ATEMP(I,J)+ ALAMDA
70    CONTINUE
      IF(NPR.LT.1) GO TO 632
      WRITE(3,632)((ATEMP(J,JJ),GTEMP(J),JJ=1,K),J=1,K)
632   FORMAT(  ' 632 DLSQ',6D17.10)
      WRITE(3,631) GSUM,ALAMDA
631   FORMAT(' 631 DLSQ',2D15.8)
637   CONTINUE
      CALL DGELG(GTEMP,ATEMP,K,1,1.E-15,IER)
      IF(IER.NE.0) GO TO 250
      DO 75 J=1,K
      T(J)= GTEMP(J)
      IF(NPR.LT.1) GO TO 75
      WRITE(3,633) T(J)
633   FORMAT(' 633 DLSQ',D15.8)
75    CONTINUE
C
60     CONTINUE
C     IF PARMS MASKED RCONSTRUCT ORIGINAL PARM VECTOR
C     ABSENT
C
C
C     FORM NEW PARM VECTOR
      DO 80 J=1,K
80    PARM(J)=PARM(J)+T(J)
C
C     TEST MAGNITUDE OF CORR. VECTOR
      ITEST=0
      DO 90 J=1,K
      TEST=DABS(T(J))/(DABS(PARM(J))+TAU)
      IF(TEST.GE.EPS) ITEST=1
90    CONTINUE
      IF(ITEST.EQ.0) GO TO 130
C
      ITER=ITER+1
C
      GO TO 100
C
C
130   CONTINUE
      CALL FCTN(N,M,K,NS,X,PARM,FN,KOUNT1)
      CALL PHI(N,NS,Y,FN,W,PHIN,WORST,IWORST)
      RMS=DSQRT(PHIN/N)/SEC
      WRITE(3,120) ITER,PHIN,RMS,KOUNT1,KOUNT2
120   FORMAT(//' ITER NO.',I6,' FINAL PHI,RMS=',D15.8,F10.4,' KOUNT1,2='
     1,2I6,/' CONVERGED PARM VECTOR FOLLOWS..')
C
      CALL SHIFT(PARM(3))
      DO 122 J=1,K
      DPARM=PARM(J)/PIO
      SPARM=PARM(J)/SEC
      WRITE(3,121)J,PARM(J),DPARM,SPARM
122   CONTINUE
121   FORMAT(' PARM NO. ',I4,D15.8,' RAD ',F12.5,' DEG ',F11.4,'SEC')
      RETURN
C
C
C     ABEND......................
20    CONTINUE
      WRITE(3,21) ITER
21    FORMAT(' ITER NO.',I5,'  PHI INCREASING....ABEND.')
      GO TO 101
C
250   WRITE(3,251) ITER,IER
251   FORMAT(////'  ITER=',I5,'   MATX EQN SOLN ROUTINE DGELG FAIL,IER='
     1,I4)
      RETURN
      END
```

ESFIT

```
C     ELT=G   ESFIT MOD NO.  21
      SUBROUTINE ESFIT(N,K,KD,NOPT,X,XX,Y,YY,W,P,ALPHA,C,
     1JS,JT,JG,JQ,JL,JRA,JRB)
C     PROGRAM TWO
C     SURFACE FITTING ROUTINE TAKES DATA X,Y SCALES OR TRANSFORMS TO XX,'
C     AND GENERATES WEIGHTS ACCORDING TO OPTION NOPT(2).
C     RETURNS COEFF. ALPHA AND INDEX CONSTS L,Q,G AND VALUE OF P AT EACH I
C
C
C     NOPT(1)=0 DISABLES PRINTING,=1 SOME ETC,=9 PRINTS AT END.
C
C     NOPT(2)=1,2,3 SELECTS METHOD.(USUALLY =1)
C
C
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(N,2),XX(N,2),Y(N,2),YY(N,2),W(N,2),P(N,KD,2)
      DIMENSION ALPHA(KD,KD,2),JL(KD),JQ(KD),JG(KD),C(KD,2),NOPT(4)
      DIMENSION JS(KD),JT(KD),JRA(KD),JRB(KD),WB(2)
      REAL FRAT,FISH
C
      PI=3.14159265358979 D0
      PIO=PI/180.D0
      SEC=3600.D0/PIO
      NPX=NOPT(1)
      IF(NOPT(1).EQ.9) NOPT(1)=0
      NPR=NOPT(1)
C
C.....DETERMINE METHOD...
      WRITE(3,20) NOPT,N,K
20    FORMAT(' START ERROR SURFACE FITTING ROUTINE OPTNS=',4I2,' N,K=',2
     1I6)
      JOPT=NOPT(2)
      GO TO (100,200,300,400),JOPT
400   RETURN
C
C............................................................
C.....TREAT PHI AS RMS ERROR ON SKY....
100   CONTINUE
C     FIND XMAX,XMIN..
      XMAXA=X(1,1)
      XMINA=XMAXA
      XMAXB=X(1,2)
      XMINB=XMAXB
      DO 110 I=2,N
      IF(X(I,1).GT.XMAXA) XMAXA=X(I,1)
      IF(X(I,1).LT.XMINA) XMINA=X(I,1)
      IF(X(I,2).GT.XMAXB) XMAXB=X(I,2)
      IF(X(I,2).LT.XMINB) XMINB=X(I,2)
110   CONTINUE
C     SCALE X VECTORS TO INTERVAL -1...+1AND Y SIMILARLY.
      SCALEA=2.D0/(XMAXA-XMINA)
      SCALEB=2.D0/(XMAXB-XMINB)
      CONSTA=1.D0-SCALEA*XMAXA
      CONSTB=1.D0-SCALEB*XMAXB
C
      DO 120 I=1,N
      XX(I,1)=CONSTA+SCALEA*X(I,1)
      XX(I,2)=CONSTB+SCALEB*X(I,2)
      YY(I,1)=SCALEA*(Y(I,1)-X(I,1))
      YY(I,2)=SCALEB*(Y(I,2)-X(I,2))
      W(I,1)=1.D0/SCALEA
      W(I,2)=DCOS(X(I,1))/SCALEB
      IF(NPR.GT.1) WRITE(3,121)I,XX(I,1),XX(I,2),YY(I,1),YY(I,2),W(I,2)
121   FORMAT(' XXYYCOS=',I4,5D15.8)
120   CONTINUE
      IF(NPR.GT.1)WRITE(3,122)SCALEA,SCALEB,CONSTA,CONSTB
122   FORMAT('  METHOD 1.SCALES,CONSTS=',4D15.8)
      IW=1
      GO TO 600
C..............................................................
C     FIT ERRORS YY=SCALE*(Y-X)   ........................
200   CONTINUE
C     FIND XMAX,XMIN..
      XMAXA=X(1,1)
      XMINA=XMAXA
      XMAXB=X(1,2)
      XMINB=XMAXB
      DO 210 I=2,N
      IF(X(I,1).GT.XMAXA) XMAXA=X(I,1)
      IF(X(I,1).LT.XMINA) XMINA=X(I,1)
      IF(X(I,2).GT.XMAXB) XMAXB=X(I,2)
      IF(X(I,2).LT.XMINB) XMINB=X(I,2)
210   CONTINUE
C     SCALE X VECTORS TO INTERVAL -1...+1AND Y SIMILARLY.
      SCALEA=2.D0/(XMAXA-XMINA)
      SCALEB=2.D0/(XMAXB-XMINB)
      CONSTA=1.D0-SCALEA*XMAXA
      CONSTB=1.D0-SCALEB*XMAXB
```

```
C
      DO 220 I=1,N
      XX(I,1)=CONSTA+SCALEA*X(I,1)
      XX(I,2)=CONSTB+SCALEB*X(I,2)
      YY(I,1)=SCALEA*(Y(I,1)-X(I,1))
      YY(I,2)=SCALEB*(Y(I,2)-X(I,2))
      W(I,1)=1.D0
      W(I,2)=1.D0
      IF(NPR.GT.1) WRITE(3,121)I,XX(I,1),XX(I,2),YY(I,1),YY(I,2)
220   CONTINUE
      IF(NPR.GT.1)WRITE(3,122) SCALEA,SCALEB,CONSTA,CONSTB
      IW=0
      GO TO 600
      ****************************************************************
C
C     FIT USING PERIODIC FNS XX=COS(SCALED X)
300   CONTINUE
C     FIND XMAX,XMIN..
      XMAXA=X(1,1)
      XMINA=XMAXA
      XMAXB=X(1,2)
      XMINB=XMAXB
      DO 310 I=2,N
      IF(X(I,1).GT.XMAXA) XMAXA=X(I,1)
      IF(X(I,1).LT.XMINA) XMINA=X(I,1)
      IF(X(I,2).GT.XMAXB) XMAXB=X(I,2)
      IF(X(I,2).LT.XMINB) XMINB=X(I,2)
310   CONTINUE
C     SCALE X VECTORS TO INTERVAL 0-PI
      SCALEA=PI/(XMAXA-XMINA)
      SCALEB=PI/(XMAXB-XMINB)
      CONSTA=-SCALEA*XMINA
      CONSTB=-SCALEB*XMINB
C
      DO 320 I=1,N
      XX(I,1)=CONSTA+SCALEA*X(I,1)
      XX(I,2)=CONSTB+SCALEB*X(I,2)
      XX(I,1)=DCOS(XX(I,1))
      XX(I,2)=DCOS(XX(I,2))
      YY(I,1)=Y(I,1)-X(I,1)
      YY(I,2)=Y(I,2)-X(I,2)
      W(I,1)=1.D0
      W(I,2)=1.D0
      IF(NPR.GT.1) WRITE(3,121)I,XX(I,1),XX(I,2),YY(I,1),YY(I,2)
320   CONTINUE
      IF(NPR.GT.1)WRITE(3,122) SCALEA,SCALEB,CONSTA,CONSTB
      IW=0
C****************************************************************
C
C
C
C     COMPUTE INITIAL RMS ERROR ON SKY
600   PHY=0.D0
      DO 610 I=1,N
      PHY=PHY+((Y(I,2)-X(I,2))*DCOS(X(I,1)))**2
610   PHY=PHY+(Y(I,1)-X(I,1))**2
      RMS=DSQRT(PHY/N)*SEC
      WRITE(3,620) PHY,RMS
620   FORMAT(' ESFIT INITIAL PHY(RAD) &RMS ERR. ON SKY(SEC)= ',2D15.8)
C
C     COMPUTE INITIAL TRANSFORMAED PHI & SIGSQ
      PHI=0.D0
      DO 630 I=1,N
      DO 630 M=1,2
630   PHI=PHI+((YY(I,M)-XX(I,M))*W(I,M))**2
      SIGSQ=PHI/(2.D0*(N-K-1))
      WRITE(3,640) PHI,SIGSQ
640   FORMAT(' INIT. VAL OF PHI,SIGSQ FOR XFORMED VARIABLES= ',2D15.8)
C
C
C*****CONSTRUCT ORTHOGONAL POLNS SEQUENTIALLY
      J=0
650   CONTINUE
      CALL ORGPOL(N,J,K,KD,NOPT,IW,XX,W,P,ALPHA,JS,JT,JG,JQ,JL,
     1JRA,JRB)
C
C
C*****CALCULATE COEFFS C(J)
700   JJ=J
      IF(J.EQ.0) JJ=KD
      DO 720 M=1,2
      A=0.D0
      B=0.D0
      DO 710 I=1,N
      A=A+W(I,M)**2*P(I,JJ,M)*YY(I,M)
710   B=B+(W(I,M)*P(I,JJ,M))**2
720   C(JJ,M)=A/B
C
C     SAVE PREVIOUS PHI AS PHID SIGSQ ASSIGSQD.
      PHID=PHI
      SIGSQD=SIGSQ
C
C
C*****COMPUTE NEW PHI PHY SIGSQ FRATIO PHITHEORETIC ETC.
750   PHI=0.D0
      DO 740 M=1,2
      DO 740 I=1,N
C     CALC. SERIES
      SUM=0.D0
      JC=-1
730   JC=JC+1
      JR=JC
      IF(JR.EQ.0)JR=KD
      SUM=SUM+C(JR,M)*P(I,JR,M)
      IF(JC.NE.J) GO TO 730
C
740   PHI =PHI+((YY(I,M)-SUM)*W(I,M))**2
      SIGSQ=PHI/(2.D0*(N-J-1))
C     CALC. PHITHEORETIC BASED ON PREVIOUS PHI AND LATEST C(J).
C     AND THE REDUCTION IN PHI..EJD(THEORY) & EJ (COMPUTED)
      EJD=0.D0
      DO 760 I=1,N
      DO 760 M=1,2
760   EJD=EJD+(C(JJ,M)*P(I,JJ,M)*W(I,M))**2
      PHIT=PHID-EJD
      EJ=PHID-PHI
C     CALC. EXPECTED ERROR IN C(J) =EPS
      EPS=0.5D0*(EJ/EJD-1.D0)
C     CALC. FRATIO
      FR=EJ/SIGSQ
C     TEST FORMATS............
991   FORMAT(' ONE')
992    FORMAT(' TWO')
993   FORMAT(' THREE')
994   FORMAT(' FOUR ')
995   FORMAT(' FIVE ')
C*****COMPUTE ON -SKY PHY,RMS ERROR.
      PHY=0.D0
      ERRMAX=0.D0
      DO 770 I=1,N
C
C     FORM SERIES.
      YA=0.D0
      YB=0.D0
      DO 765 JC=0,J
      JR=JC
      IF(JR.EQ.0) JR=KD
      YA=YA+C(JR,1)*P(I,JR,1)
765   YB=YB+C(JR,2)*P(I,JR,2)
766   FORMAT(' SUM (',2I5,') ',2D15.8)
C     RETRANSFORM YY TO Y
C
       GO TO(901,902,903),JOPT
901   CONTINUE
C
902   YA=YA/SCALEA+X(I,1)
      YB=YB/SCALEB+X(I,2)

      GO TO 910
C
903   YA=YA+X(I,1)
      YB=YB+X(I,2)
910   CONTINUE
C
C
C     STORE SUMS IN YY IF FINAL ORDER REACHED
      IF(J.NE.K) GO TO 780
      YY(I,1)=YA
      YY(I,2)=YB
780   CONTINUE
C
      NPS=0
      IF(NPR.GT.1) NPS=2
      IF(NPX.EQ.9.AND.J.EQ.K) NPS =2
      IF (NPS.GT.1)WRITE(3,766) I,M,YA,YB
C
      ERR=(YA-Y(I,1))**2
      ERR=ERR+((YB-Y(I,2))*DCOS(X(I,1)))**2
      IF(ERR.GT.ERRMAX) ERRMAX=ERR
770   PHY=PHY+ERR
      RMS=DSQRT(PHY/N)*SEC
      ERRMAX=DSQRT(ERRMAX)*SEC
C
C
C*****PRINT
800   CONTINUE
      WRITE(3,805) J,PHY,RMS,ERRMAX
805   FORMAT(1X,I4,'TH ORDER ON-SKY PHY=',D15.8,
     1'  RMS(ARCSEC)=',D15.8,'    ERRMAX=',D15.8)
      WRITE(3,810)J,PHI,PHIT,SIGSQ
810   FORMAT(I4,' TH ORDER SUMSQUARES(XFORMED),THEORVALUE,SIGSQ ',2D17.1
     10,D13.6)
      WRITE(3,820)EJ,EJD,FR
820   FORMAT(' COMPUTED,THEOR REDUCTION IN PHI ,F-RATIO=',2D15.8,D13.6)
      WRITE(3,830) J,C(JJ,1),C(JJ,2),EPS
830   FORMAT(' C(',I3,' 1,2 )=',2D15.8,' EXPECTEDFRACTIONAL ERROR=',
     1D20.8,' ***********************')
C
C
C     IS FINAL ORDER  K REACHED YET..
      IF(J.EQ.K) GO TO 1000
900   J=J+1
      GO TO 650
C
C*****EXIT
1000  CONTINUE
C
C     STORE N,NU2,PHI,SIGSQ IN W( ,1) FOR TESTS
C
       NU2=2*(N-K-1)
       W(1,1)=N
       W(2,1)=NU2
       W(3,1)=PHI
       W(4,1)=SIGSQ
C
C
C        FINAL PRINTING  ......................................
C
       WRITE(3,1010)J,RMS,PHI,SIGSQ,NU2
1010  FORMAT(//' FINAL RESULTS AFTER ',I3,' ORDER,RMS ERR='F10.4,
     1' PHI=',D13.6,' SIGSQ,NU2=',D13.6,I5)
C
C     GET Y AVERAGE
       RESAVG=0.D0
       DO 1200 I=1,N
1200   RESAVG=RESAVG+Y(I,1)-X(I,1) + (Y(I,2)-X(I,2))*DCOS(X(I,1))
       RESAVG=RESAVG/(2.D0*N)
C
       WRITE(3,1230)RESAVG
1230   FORMAT(/'  **** RESID AVGE= ',D13.6)
C
C
C      GET SUMSQURE DUE REGRESS,AND CORR FOR MEAN
       SSE=0.D0
       SSR=0.D0
       SSM=0.D0
       WB(1)=1.D0
       DO 1210 I=1,N
       WB(2)=DCOS(X(I,1))**2
       DO 1210 L=1,2
       SSE=SSE+(YY(I,L)-Y(I,L))**2*WB(L)
       SSR=SSR+(YY(I,L)-X(I,L)-RESAVG)**2*WB(L)
       SSM=SSM+(Y(I,L)-X(I,L)-RESAVG)**2*WB(L)
1210   CONTINUE
C
C      CALC MULT REGRESS COEFF=RSQ,MEAN SQR RATIO=SQRAT
       NUU1=2*K+1
       RSQ=SSR*100.D0/SSM
       SQRAT=SSR/(SIGSQ*NUU1)
       FRAT=SQRAT
       PROB=100.D0*(1.D0-FISH(FRAT,NUU1,NU2))
C
C
       WRITE(3,1220)SSE,SSR,SSM,SQRAT,NUU1,NU2,PROB,RSQ
1220  FORMAT(//' PHI,SSR,SSM=',3D14.6,' MEANSQRAT=',F13.5,2I5,
     1' PROB,MULTREGCOEFF=',F10.5,F8.4)
C
C
       IF(NPX.NE.9) GO TO 1100
       DO 1030 JJ=0,J
       JC=JJ
       IF(JJ.EQ.0) JC=KD
       WRITE(3,1020) JJ,C(JC,1),JJ,C(JC,2)
1020   FORMAT(' COEFF(',I3,' 1 )=',D15.8,'    COEFF(',I3,' 2 )=',D15.8)
       IF(JJ.EQ.0) GO TO 1030
       JA=JRA(JC)
       JB=JRB(JC)
       DO 1040 JK=JA,JB
       JD=JK
       IF(JK.EQ.0) JD =KD
1040   WRITE(3,1050)JJ,JK,ALPHA(JC,JD,1),ALPHA(JC,JD,2)
1050   FORMAT(' ALPHA(',I2,1X,I2,' M)=',2D15.8)
1030   CONTINUE
1100   CONTINUE
       RETURN
       END
```

EXPAND

```
C      ELT=0   EXPAND    MOD0 3.
       SUBROUTINE EXPAND(K,MASK,T,MSUM)
C      REQUIRES MSUM=SUM OF MASKING FN.(FROM CMPRES).IF MSUM<K EXPANDS
C      VECTOR T FROM MSUM DIM.  TO K DIM. FILLING IN WITH 0.D0'S
C
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION MASK(K),T(K)
C
C
       IF(MSUM.EQ.K) GO TO 1000
C
       KM=K-1
       DO 20 J=1,K
       IF(MASK(J).NE.0) GO TO 20
C
C
       IF(J.EQ.K) GO TO 40
       DO 50 I=KM,J,-1
       T(I+1)=T(I)
50     CONTINUE
40     T(J)=0.D0
C
20     CONTINUE
C
1000   RETURN
       END
```

FCTN (5 PARM)

```
C      ELT=F5  ,5-PARM MODEL FUNCTION(COMPATIBLE WITH NEW)MOD=5.
       SUBROUTINE FCTN(N,M,K,NS,XX,PARM,FN,IFCTN)
C      CALC. TELESCOPE POINTING ERROR MODEL FUNCTION FOR
C      MISALIGNED,SKEW EQTL MTG WITH ZERO OFFSETS.
C
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION   XX(N,M),FN(N,NS),PARM(K)
C       CAUTION TEMPORARY DIMENSIONED VARIABLES,...
       DIMENSION  DEC(160),HA(160)
       DEFINE DARCOS(XDUM)=DACOS(XDUM)
        DEFINE DARSIN(XDUM)=DASIN(XDUM)
C
       IFCTN=IFCTN+1
       NPR=0
       THETA=PARM(1)
       PSI=PARM(2)-PARM(3)
       PHY=PARM(3)
       ETA=PARM(4)
       DEE=PARM(5)
C
C
C
       ST=DSIN(THETA)
       CT=DCOS(THETA)
       SP=DSIN(PSI)
       CP=DCOS(PSI)
       SE=DSIN(ETA)
       CE=DCOS(ETA)
C
       IF(NPR.EQ.1) WRITE(3,609) ST,CT,SP,CP,SE,CE
609    FORMAT(/' 609 FCTN',6D15.8)
       DO 130 J=1,N
       DEC(J)=XX(J,1)
       HA(J)=XX(J,2)
       DEK=DARSIN(DSIN(DEC(J)+DEE)*CE)
       HAK=HA(J)+DATAN(DTAN(DEC(J)+DEE)*SE)
       CPH=DCOS(PHY+HAK)
       SPH=DSIN(PHY+HAK)
       X=DCOS(DEK)*(CP*CPH-CT*SP*SPH)+DSIN(DEK)*ST*SP
       Y=-DCOS(DEK)*(SP*CPH+CT*CP*SPH)+DSIN(DEK)*ST*CP
       Z=DCOS(DEK)*ST*SPH+CT*DSIN(DEK)
       IF(NPR.EQ.1) WRITE(3,610) DEK,HAK,CPH,SPH,X,Y,Z
610    FORMAT(/' 610FCTN',7D15.8)
       HAK=DATAN(-Y/X)
       DEK=DARSIN(Z)
       FN(J,1)=DEK
       FN(J,2)=HAK
       IF(NPR.LT.1) GO TO 130
       WRITE(3,611) XX(J,1),XX(J,2),FN(J,1),FN(J,2)
611    FORMAT(' 611 FCTN ',4D15.8)
130    CONTINUE
       RETURN
       END
```

FCTN (74 INCH)

```
C      ELT=F     EXTENDED 74=  MODEL FUNCTION   MODIFICATION NO. E9
       SUBROUTINE FCTN(N,M,K,NS,XX,B,FN,IFCTN)
C      CALC. TELESCOPE POINTING ERROR MODEL FUNCTION FOR
C      MISALIGNED,SKEW EQTL MTG WITH ZERO OFFSETS.
C      PERIODIC ENCODER GEAR ERRORS + STRUCTURAL FLEXURE.
C
C
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION   XX(N,M),FN(N,NS),B(K),DD(8),DH(8)
C
       IFCTN=IFCTN+1
       NPR=0
       PI=3.14159265358979300
       PIO=PI/180.D0
       PHI=-(35.D0+19.D0/60.D0+17.5D0/3600.D0)*PIO
       THETA=B(1)
       PHY=B(3)
       PSI=B(2)-B(3)
C
```

```
      ST=DSIN(THETA)
      CT=DCOS(THETA)
      SP=DSIN(PSI)
      CP=DCOS(PSI)
C
C
      ACONST=15.D0*PI
      A1=9.D0/14.D0
      A2=70.D0*PIO
      A3=0.85D0
      A4=14.2D0/ACONST
      A5=4.8D0/ACONST
      A6=1.28D0
      A7=10.D0/ACONST
C
      PDA=1512.D0
      PDB=PDA/32.D0
      PHA=864.D0
      PHB=PHA/32.D0
C
      DO 130 J=1,N
      DEC=XX(J,1)
      HA=XX(J,2)
      CPH=DCOS(PHY+HA)
      SPH=DSIN(PHY+HA)
      CD=DCOS(DEC)
      SD=DSIN(DEC)
      CH=DCOS(HA)
      SH=DSIN(HA)
      CPDA=DCOS(PDA*DEC)
      SPDA=DSIN(PDA*DEC)
      CPDB=DCOS(PDB*DEC)
      SPDB=DSIN(PDB*DEC)
      CPHB=DCOS(PHB*HA)
      SPHB=DSIN(PHB*HA)
      CPHA=DCOS(PHA*HA)
      SPHA=DSIN(PHA*HA)
      X=CD*(CP*CPH-CT*SP*SPH)+SD*ST*SP
      Y=-CD*(SP*CPH+CT*CP*SPH)+SD*ST*CP
     /Z=CD*ST*SPH+CT*SD
      IF(NPR.EQ.1) WRITE(3,610) DEC,HA,CPH,SPH,X,Y,Z
610   FORMAT(/' 610FCTN',7D15.8)
      HAK=DATAN(-Y/X)
      DEK=DASIN(Z)
C
      DD(1)=DEK-DEC
      DH(1)=HAK-HA
C
C
      DD(2)=DASIN(SD*DCOS(B(4)))-DEC
      DH(2)=DATAN(DTAN(DEC)*DSIN(B(4)))
C
C
     .DD(3)=B(5)+B(6)*CD+B(7)*SD+B(8)*CPDA+B(9)*SPDA
     1+B(10)*CPDB+B(11)*SPDB
     .DH(3)=B(12)*CH+B(13)*SH+B(14)*CPHA+B(15)*SPHA
     1+B(16)*CPHB+B(17)*SPHB
C
C
      DD(4)=B(20)*(DCOS(PHI)*CH*SD-DSIN(PHI)*CD)
      DH(4)=B(18)*DCOS(PHI)*CD*SH
C
C
      DD(5)=B(21)*DCOS(A1*DEC+A2)*DCOS(A3*HA)
      DH(5)=B(19)*((A4+A5*DSIN(A6*DEC))*HA-A7*DEC)
C
C
      FN(J,1)=DEC
      FN(J,2)=HA
      DO 140 JJ=1,5
      FN(J,1)=FN(J,1)+DD(JJ)
140   FN(J,2)=FN(J,2)+DH(JJ)
C
C
      IF(NPR.LT.1) GO TO 130
      WRITE(3,611) XX(J,1),XX(J,2),FN(J,1),FN(J,2)
611   FORMAT(' 611 FCTN ',4D15.8)
130   CONTINUE
      RETURN
      END
```

FCTN (74 INCH EXACT)

```
C     ELT=F EXTENDED 74=(EXACT)MODEL FUNCTION    MOD=2F
      SUBROUTINE FCTN(N,M,K,NS,XX,B,FN,IFCTN)
C     CALC. TELESCOPE POINTING ERROR MODEL FUNCTION FOR
C     MISALIGNED,SKEW EQTL MTG WITH ZERO OFFSETS.
C     PERIODIC ENCODER GEAR ERRORS + STRUCTURAL FLEXURE.
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION    XX(N,M),FN(N,NS),B(K)
C
      IFCTN=IFCTN+1
      NPR=0
      PI=3.14159265358979D0
      PIO=PI/180.D0
      PHI=-(35.D0+19.D0/60.D0+17.5D0/3600.D0)*PIO
      THETA=B(1)
      PHY=B(3)
      PSI=B(2)-B(3)
C
C
C
      ST=DSIN(THETA)
      CT=DCOS(THETA)
      SP=DSIN(PSI)
      CP=DCOS(PSI)
C
C
      ACONST=15.D0*PI
      A1=9.D0/14.D0
      A2=70.D0*PIO
      A3=0.85D0
      A4=14.2D0/ACONST
      A5=4.8D0/ACONST
      A6=1.28D0
      A7=10.D0/ACONST
C
      PDA=1512.D0
      PDB=PDA/32.D0
      PHA=864.D0
      PHB=PHA/32.D0
C
      DO 130 J=1,N
      DEC=XX(J,1)
      HA=XX(J,2)
C
C     ONE      ENCODER ERRORS
C
      CD=DCOS(DEC)
      SD=DSIN(DEC)
      CH=DCOS(HA)
      SH=DSIN(HA)
      CPDA=DCOS(PDA*DEC)
      SPDA=DSIN(PDA*DEC)
      CPDB=DCOS(PDB*DEC)
      SPDB=DSIN(PDB*DEC)
      CPHB=DCOS(PHB*HA)
      SPHB=DSIN(PHB*HA)
      CPHA=DCOS(PHA*HA)

      SPHA=DSIN(PHA*HA)
C
     .DEC=DEC+B(5)+B(6)*CD+B(7)*SD+B(8)*CPDA+B(9)*SPDA
     1+B(10)*CPDB+B(11)*SPDB
      HA=HA+B(12)*CH+B(13)*SH+B(14)*CPHA+B(15)*SPHA
     1+B(16)*CPHB+B(17)*SPHB
C
C     TWO      TORSIONAL MOV'T OF AXES
C
      CD=DCOS(DEC)
      CH=DCOS(HA)
      DEC=DEC+B(20)*CH*DSIN(DEC-PHI)
      HA=HA+(B(22)*CH+B(18)*CD*DSIN(HA))*DCOS(PHI)
C
C     THREE    SKEWNESS OF AXES
C
      TD=DTAN(DEC)
      DEC=DASIN(DSIN(DEC)*DCOS(B(4)))
      HA=HA+DATAN(TD*DSIN(B(4)))
C
C     FOUR   MISALIGNMENT FROM POLE
C
      CPH=DCOS(PHY+HA)
      SPH=DSIN(PHY+HA)
      CD=DCOS(DEC)
      SD=DSIN(DEC)
      CH=DCOS(HA)
      SH=DSIN(HA)
      X=CD*(CP*CPH-CT*SP*SPH)+SD*ST*SP
      Y=-CD*(SP*CPH+CT*CP*SPH)+SD*ST*CP
      Z=CD*ST*SPH+CT*SD
C
      HA=DATAN(-Y/X)
      DEC=DASIN(Z)
C
C     FIVE   FLEXURE OF TUBE AND OPTIC SUPPORTS
C
      DEK=DEC+B(21)*DCOS(A1*DEC+A2)*DCOS(A3*HA)
      HA=HA+B(19)*((A4+A5*DSIN(A6*DEC))*HA-A7*DEC)
C
C
      FN(J,1)=DEK
      FN(J,2)=HA
130   CONTINUE
      RETURN
      END
```

FITEST

```
C     ELT=H    FITEST   MOD NO. = 12
     _SUBROUTINE FITEST(N,K,KD,NOPT,X,XX,Y,YY,W,P,ALPHA,C,
     1JS,JT,JG,JQ,JL,JRA,JRB)
C
C       ...PROGRAM THREE...TESTS ACCURACY OF ERR. SURFACE FIT
C
C     NOPT(1)GT 0 ENABLES PRINT,=9 GIVES PLOT OF RESIDS,=8 BOTH.
C     NOPT(2)=1,2,3 GIVES METHOD.
C     NOPT(3)=0 GENERATE FRESH DATA,=1 USE EXISTING DATA,=2 READ IN DATA.
C     NOPT(4)=9 SUPPRESSES COMPARISON..ONLY SUM GENERATED.
C
      IMPLICIT  REAL*8(A-H,O-Z)
      DIMENSION NOPT(4),X(N,2),XX(N,2),Y(N,2),YY(N,2),W(N,2)
      DIMENSION ALPHA(KD,KD,2),C(KD,2),P(N,KD,2)
      DIMENSION JS(KD),JT(KD),JG(KD),JQ(KD),JL(KD),JRA(KD),JRB(KD)
      DIMENSION NDAT(4)
C     CAUTION TEMPORARILY DIMENSIONED VARIABLES.
      REAL DEC(160),HA(160),DD(160),DH(160),DR(160),FRATIO,FISH
      REAL*8 MEG(2)
      PI=3.14159265358979D0
      PIO=PI/180.D0
      SEC=3600.D0/PIO
      NPR=NOPT(1)
       IF(NOPT(1).EQ.9) NPR=0
      METH=NOPT(2)
      IDATA=NOPT(3)
      NDAT(1)=NPR
      NDAT(2)=0
      NDAT(3)=0
      NDAT(4)=0
C
C     GET NFIT,NU2,PHIFIT,SIG2 FROM W
C
      NFIT=W(1,1)
      NU2=W(2,1)
      PHIFIT=W(3,1)
      SIG2=W(4,1)
C
C
C
      IF(IDATA.EQ.0) GO TO 10
      GO TO (100,200),IDATA
C     ****************************************************************
10    WRITE(3,20)N,K,METH
20   _FORMAT(/' ******SURFACE FIT TEST ROUTINE N,K,METHOD=',3I4/
     1' GENERATES ADDITIONAL DATA USING DATGEN'/' ***PLSE INPUT 2 RANDOM
     2 PARMS.')
      CALL READ(NVAR,MEG,NERR)
      IF(NERR.EQ.1.OR.NVAR.NE.2) GO TO 999
      WRITE(3,30) MEG
30    FORMAT(' NEW DATA GEN. WITH RANDOM PARMS=',2F15.1)
      RC=9275.D0
      RA=MEG(1)
      RB=MEG(2)
      CALL DATGEN(N,K,KD,NDAT,X,XX,Y,RA,RB,RC)
C     NOW HAVE NEW X,Y DATA.
      GO TO 400
C     ****************************************************************
100   WRITE(3,110) N,K,METH
110  _FORMAT(/' *****ERR. SURFACE FIT TEST ROUTINE USING EXISTING DATA
     1 N,K,METHOD=',3I4)
      GO TO 400
C     ****************************************************************
C     READ IN NEW TELESCOPE DATA FROM FILE 14.
200   CONTINUE
C
      READ(14)NOBS
      WRITE(3,210)NOBS
210  _FORMAT(' N=NO. OF DATA PTS HAS BEEN RESET TO ',I5,
     1' FOR FOLLOWING FIT TESTS.')
      N=NOBS
      DO 220 J=1,NOBS
      READ(14)X(J,1),X(J,2),Y(J,1),Y(J,2)
      IF(NPR.GT.1)WRITE(3,230)J,X(J,1),X(J,2),Y(J,1),Y(J,2)
230   FORMAT(' XY FILE 14 ENTRY # ',I4,5X,4D20.10)
220   CONTINUE
C     ****************************************************************
C     DETERMINE METHOD.
400   GO TO (1000,2000,3000),METH
C     ****************************************************************
C....TREAT PHI AS RMS ERROR ON SKY....
1000   CONTINUE
C     FIND XMAX,XMIN..
      XMAXA=X(1,1)
      XMINA=XMAXA
      XMAXB=X(1,2)
      XMINB=XMAXB
      DO 1100 I=2,N
      IF(X(I,1).GT.XMAXA) XMAXA=X(I,1)
      IF(X(I,1).LT.XMINA) XMINA=X(I,1)
      IF(X(I,2).GT.XMAXB) XMAXB=X(I,2)
      IF(X(I,2).LT.XMINB) XMINB=X(I,2)
1100   CONTINUE
C     SCALE X VECTORS TO INTERVAL -1...1AND Y SIMILARLY.
      SCALEA=2.D0/(XMAXA-XMINA)
      SCALEB=2.D0/(XMAXB-XMINB)
      CONSTA=1.D0-SCALEA*XMAXA
      CONSTB=1.D0-SCALEB*XMAXB
C
      DO 1200 I=1,N
      XX(I,1)=CONSTA+SCALEA*X(I,1)
      XX(I,2)=CONSTB+SCALEB*X(I,2)
      YY(I,1)=SCALEA*(Y(I,1)-X(I,1))
      YY(I,2)=SCALEB*(Y(I,2)-X(I,2))
      W(I,1)=1.D0/SCALEA
      W(I,2)=DCOS(X(I,1))/SCALEB
      IF(NPR.GT.1) WRITE(3,121)I,XX(I,1),XX(I,2),YY(I,1),YY(I,2),W(I,2)
121   FORMAT(' XXYYCOS=',I4,5D15.8)
1200   CONTINUE
      IF(NPR.GT.1)WRITE(3,122)SCALEA,SCALEB,CONSTA,CONSTB
122   FORMAT('  METHOD 1.SCALES,CONSTS=',4D15.8)
      IW=1
      GO TO 500
C*********************************************************************
C     FIT ERRORS YY=SCALE*(Y-X)    *************************
2000   CONTINUE
C     FIND XMAX,XMIN..
      XMAXA=X(1,1)
      XMINA=XMAXA
      XMAXB=X(1,2)
      XMINB=XMAXB
      DO 2100 I=2,N
      IF(X(I,1).GT.XMAXA) XMAXA=X(I,1)
      IF(X(I,1).LT.XMINA) XMINA=X(I,1)
      IF(X(I,2).GT.XMAXB) XMAXB=X(I,2)
      IF(X(I,2).LT.XMINB) XMINB=X(I,2)
2100   CONTINUE
C     SCALE X VECTORS TO INTERVAL -1...1AND Y SIMILARLY.
      SCALEA=2.D0/(XMAXA-XMINA)
      SCALEB=2.D0/(XMAXB-XMINB)
      CONSTA=1.D0-SCALEA*XMAXA
      CONSTB=1.D0-SCALEB*XMAXB
C
      DO 2200 I=1,N
      XX(I,1)=CONSTA+SCALEA*X(I,1)
      XX(I,2)=CONSTB+SCALEB*X(I,2)
      YY(I,1)=SCALEA*(Y(I,1)-X(I,1))
      YY(I,2)=SCALEB*(Y(I,2)-X(I,2))
      W(I,1)=1.D0
      W(I,2)=1.D0
      IF(NPR.GT.1) WRITE(3,121)I,XX(I,1),XX(I,2),YY(I,1),YY(I,2)
2200   CONTINUE
      IF(NPR.GT.1)WRITE(3,122) SCALEA,SCALEB,CONSTA,CONSTB
      IW=0
      GO TO 500
C     ****************************************************************
C     FIT USING PERIODIC FNS XX=COS(SCALED X)
3000   CONTINUE
C     FIND XMAX,XMIN..
      XMAXA=X(1,1)
      XMINA=XMAXA
      XMAXB=X(1,2)
      XMINB=XMAXB
      DO 3100 I=2,N
      IF(X(I,1).GT.XMAXA) XMAXA=X(I,1)
      IF(X(I,1).LT.XMINA) XMINA=X(I,1)
      IF(X(I,2).GT.XMAXB) XMAXB=X(I,2)
      IF(X(I,2).LT.XMINB) XMINB=X(I,2)
3100   CONTINUE
C     SCALE X VECTORS TO INTERVAL 0-PI
      SCALEA=PI/(XMAXA-XMINA)
      SCALEB=PI/(XMAXB-XMINB)
      CONSTA=-SCALEA*XMINA
      CONSTB=-SCALEB*XMINB
C
      DO 3200 I=1,N
      XX(I,1)=CONSTA+SCALEA*X(I,1)
      XX(I,2)=CONSTB+SCALEB*X(I,2)
      XX(I,1)=DCOS(XX(I,1))
      XX(I,2)=DCOS(XX(I,2))
      YY(I,1)=Y(I,1)-X(I,1)
     /YY(I,2)=Y(I,2)-X(I,2)
      W(I,1)=1.D0
      W(I,2)=1.D0
      IF(NPR.GT.1) WRITE(3,121)I,XX(I,1),XX(I,2),YY(I,1),YY(I,2)
3200   CONTINUE

      IF(NPR.GT.1)WRITE(3,122) SCALEA,SCALEB,CONSTA,CONSTB
      IW=0
C     ****************************************************************
C     OPTIONALLY SUPRESS COMPARISON.
500   IF(NOPT(4).NE.9) GO TO 700
C     RUBBARA.
600   RETURN
```

```
C      ..............................................................
C
C      SUM SERIES ETC.
700    PHI=0.D0
       PHY=0.D0
       ERRMAX=0.D0
C      COMPUTE POLNS AT DATA PTS.
       CALL COMPOL(N,K,KD,IR,XX,P,ALPHA,JS,JT,JG,JQ,JL,JRA,JRB)
C
       DO 710 I=1,N
C      CALC SERIES.
       YYA=0.D0
       YYB=0.D0
       DO 720 J=0,K
       JJ=J
       IF(J.EQ.0) JJ=KD
       YYA=YYA+C(JJ,1)*P(I,JJ,1)
720    YYB=YYB+C(JJ,2)*P(I,JJ,2)
C      ..............................................................
C      TRANSFORM YY BACK TO Y FOR ON-SKY NURGLING.
       GO TO (810,820,830),METH
810    CONTINUE
820    CONTINUE
       YA=YYA/SCALEA+X(I,1)
       YB=YYB/SCALEB+X(I,2)
       GO TO 900
830    CONTINUE
       YA=YYA+X(I,1)
       YB=YYB+X(I,2)
C      ...........NOW HAVE Y AGAIN...................................
900    CONTINUE
C      PRINT SUMS.
       IF(NPR.GT.0)WRITE(3,910)I,YA,YB
910    FORMAT(' SERIES SUM AT ',I3,'TH PT.=',2D15.8)
C
       PHI=PHI+((YY(I,1)-YYA)*W(I,1))**2
       PHI=PHI+((YY(I,2)-YYB)*W(I,2))**2
C
C      STORE SUMS IN YY
       YY(I,1)=YA
       YY(I,2)=YB
C
C      FIND ERRMAX.
       ERR=(Y(I,1)-YA)**2+((Y(I,2)-YB)*DCOS(X(I,1)))**2
       PHY=PHY+ERR
       IF(ERR.GT.ERRMAX) ERRMAX=ERR
710    CONTINUE
C      ..............................................................
C
C
C      ASSESMENT OF FIT
C
       ERRMAX=DSQRT(ERRMAX)*SEC
       SIGSQ=PHI/(2.D0*(N-K-1))
       RMS=DSQRT(PHY/N)*SEC
C
       WRITE(3,930) PHI,SIGSQ
930    FORMAT(' PHI OF XFORMED VARIABLES=',D15.8,' SIGSQ=',D15.8)
        WRITE(3,940) PHY,RMS,ERRMAX
940   _FORMAT(' ON-SKY PHY=',D15.8,' RMS(ARCSEC)=',D15.8,
      1' MAXIMUM ON-SKY ERR.=',D15.8)
C
       NU1=2*(N-NFIT)
       SIG1=(PHY-PHIFIT)/NU1
       FRATIO=SIG1/SIG2
       PROB=(1.D0-FISH(FRATIO,NU1,NU2))*100.D0
       WRITE(3,950)PHY,PHIFIT,SIG1,SIG2,FRATIO,NU1,NU2,PROB
950   _FORMAT(/' PHI,PHIFIT,SIG1,2= ',4D13.5,' F,NU1,2,PROB= ',F7.4,
      12I4,F11.5/)
C
C
C
C      :::::::::       :                          ::
C      PLOT AND/OR LIST RESIDUALS
       IF(NOPT(1).LT.6)GO TO 4000
       DO 4010 I=1,N
       DEC(I)=YY(I,1)/PIO
       HADES=YY(I,2)
       CALL SHIFT2(HADES)
       HA(I)=HADES*12.D0/PI
       DD(I)=(Y(I,1)-YY(I,1))*SEC
       DH(I)=(Y(I,2)-YY(I,2))*DCOS(Y(I,1))*SEC
       DR(I)=SQRT(DD(I)**2+DH(I)**2)
       IF(NPR.GT.0)WRITE(3,4020)I,DEC(I),HA(I),DD(I),DH(I),DR(I)
4020  _FORMAT(' FITTED POINT NO.=',I4,'  DEC=',F10.4,'  HA=',F10.4,
      1' DEC ERR=',F10.4,' HA ERR=',F10.4,'  RESLNT= ',F10.4)
4010   CONTINUE
C
        CALL APLOT(N,DEC,' DEC ',DD,' DDEC',1)
        CALL APLOT(N,DEC,' DEC ',DH,'DH*COS',2)
        CALL APLOT(N,HA ,' HA  ',DD,'DDEC ',3)
        CALL APLOT(N,HA ,' HA  ',DH,'DH*COS',4)
        CALL APLOT(N,DEC,' DEC ',DR,'DRES  ',5)
        CALL APLOT(N,HA ,' HA  ',DR,'DRES  ',6)
C                    :                          :::::
4000   RETURN
C
C
C      READ FAIL..
999    WRITE(3,998) NVAR
998    FORMAT(//' READ ROUTINE FAILURE,NO. OF VARIABLES READ WAS ',I4)
       RETURN
       END
```

GMIN

```
C      ELT=GMIN GOLDEN SECTION LINE MINIMIZATION.      MOD 11
       SUBROUTINE GMIN(N,M,K,NS,X,Y,W,PHY,FN,B,T,BNEW,SCALE,ALPHA,ETOL,
      1STEP,STEPM,IT,NPR,K1)
C
C
C      ETOL=ACCURACY
C      ETC..................
C
       DIMENSION X(N,M),W(N,NS),FN(N,NS),Y(N,NS),B(K),T(K),BNEW(K)
       DIMENSION SCALE(K)
       IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
       IT=1
       TAU=(DSQRT(5.D0)-1.D0)/2.D0
       GOLD=1.D0-TAU
C
C
       BINT=STEP*TAU
       AINT=BINT-STEP
       BORIG=BINT
C
       IF(NPR.GT.0)WRITE(3,40)ETOL,STEP,STEPM,AINT,BINT,PHY
40    _FORMAT(' GOLDEN SECTN.LINEMIN*E,STEP,MAX=',3D10.3,' A,B,FY1=',
      13D13.6)
C
C
       X1=0.D0
       F1=PHY
       GO TO 120
C
C
300    CONTINUE
       BORIG=BINT
C      GET X1,F1
       X1=BINT-TAU*(BINT-AINT)
       DO 100 J=1,K
100    BNEW(J)=(B(J)+X1*T(J))*SCALE(J)
       CALL FCTN(N,M,K,NS,X,BNEW,FN,K1)
       CALL PHI(N,NS,Y,FN,W,F1,WZ,IWZ)
C
C      GET X2,F2
120    X2=AINT+TAU*(BINT-AINT)
       DO 110 J=1,K
110    BNEW(J)=(B(J)+X2*T(J))*SCALE(J)
       CALL FCTN(N,M,K,NS,X,BNEW,FN,K1)
       CALL PHI(N,NS,Y,FN,W,F2,WZ,IWZ)
C
C
C
500    IF(F2.GT.F1) GO TO 600
C
       LAM=2
       AINT=X1
       X1=X2
       X2=BINT-GOLD*(BINT-AINT)
       F1=F2
       DO 510 J=1,K
510    BNEW(J)=(B(J)+X2*T(J))*SCALE(K)
       CALL FCTN(N,M,K,NS,X,BNEW,FN,K1)
       CALL PHI(N,NS,Y,FN,W,F2,WZ,IWZ)
       GO TO 800
C
C
600    LAM=1
       BINT=X2
       X2=X1
       X1=AINT+GOLD*(BINT-AINT)
       F2=F1
       DO 610 J=1,K
610    BNEW(J)=(B(J)+X1*T(J))*SCALE(J)
       CALL FCTN(N,M,K,NS,X,BNEW,FN,K1)
       CALL PHI(N,NS,Y,FN,W,F1,WZ,IWZ)
C
C      TEST FOR LINE MIN CONVERGENCE
C
800    CONTINUE
       IF(NPR.GT.0)WRITE(3,810)IT,LAM,F1,F2,X1,X2,AINT,BINT
810   _FORMAT(' GMIN/IT,LAM=',2I4,' F1,2=',2D12.5,'  X1,2=',2D12.5,
      1'A,BINT=',2D12.5)
       IF(DABS(X2-X1).LT.ETOL) GO TO 1000
       IT =IT+1
       GO TO 500
C
C      CHECK X2 NOT WITHIN ETOL OF ORIGINAL  END OF INTERVAL BORIG.
1000   IF(LAM.EQ.1.OR.(BORIG-X2).GT.ETOL*2.D0) GO TO 900
       BINT=AINT+STEP
C
       IF(NPR.GT.0)WRITE(3,1100)AINT,BINT
1100   FORMAT(' INTERVAL EXPANDED TO LENGTH STEP..A,B= ',2D14.6)
       GO TO 300
C
C
C      EXIT  WITH LESSER  OF F1,F2
900    ALPHA=X1
       PHY=F1
       IF(F1.LT.F2) GO TO 920
       ALPHA=X2
       PHY=F2
920    DO 910 J=1,K
910    B(J)=B(J)+ALPHA*T(J)
C      (N.B. SCALED PARM VECTOR RETURNED)
       CALL FCTN(N,M,K,NS,X,B,FN,K1)
       K1=K1-1
       RETURN
       END
```

GRAD2

```
C      ELT,PROG=GRAD2          MODIFICATION NO.=148
       SUBROUTINE GRAD2(N,M,K,NS,NOPT,X,Y,PARM,FN,DF,W,G,GTEMP,T,SCALE,
      1MASK)
C      PROGRAM  11
C      MODIFIED STEEPEST DESCENT ALGORITHM FOR PARM. S ESTIMATION.
C      OPTIMUM GRADIENT VERSION
C
C
C
C      -------------------------------------------------------OPTIONS
C      IF NOPT(1)=0 READS PARM,MASK FROM  K  CARDS.
C
C      NOPT(2) GE 1 ENABLES PRINT**
C      ------------------------------------------------------------
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION   DATA(2),NOPT(4)
       DIMENSION   X(N,M),Y(N,NS),FN(N,NS),W(N,NS),DF(N,K,NS)
       DIMENSION PARM(K),T(K),G(K),GTEMP(K),MASK(K),SCALE(K)
       REAL*8 MEG
C
C      INITIALIZE
       PI=3.14159265360O
       PIO=PI/180.D0
       SEC=PIO/3600.D0
       ITER=1
        MEG=1.D0
       BEE=1.D-6
       DEL=5.D-5
       TAU=1.D-15
       EPS=1.D-5
       KOUNT1=0
       KOUNT2=0
       NPR=NOPT(2)
C      SET LINE MIN. OPTION. 1=POWELL,2=GOLDEN SECTION.
       MIN=NOPT(3)
C
C      ......................................................... DATA INPUT
C
       IF(NOPT(1).EQ.0) GO TO 6
6      CONTINUE
       WRITE(3,7)
7      FORMAT(' ****ENTER PARMS +MASKS')
       DO 10 J=1,K
       CALL READ(NVAR,DATA,NERR)
       IF(NVAR.NE.2.OR.NERR.EQ.1) GO TO 4010
       PARM(J)= DATA(1)
       MASK(J)= DATA(2)
10     CONTINUE
C
C
C      ........................................................... START.
C      CALC CURRENT PHI
       CALL FCTN(N,M,K,NS,X,PARM,FN,KOUNT1)
       CALL WT(N,M,NS,X,W)
       CALL PHI(N,NS,Y,FN,W,PHIO,WORST,IWORST)
       RMERR=DSQRT(PHIO/N)
       RMS=RMERR/SEC
C
C      PRINT
       WRITE(3,50) PHIO,RMS
50    _FORMAT(//' OPTIMISED STEEPST DESCNT ROUTINE.INITIAL PHI,RMS IS..'
      1,D15.8,F12.6/'  INITIAL PARMS(MASKS) ARE..')
       DO 60 J=1,K
60     WRITE(3,70) PARM(J),MASK(J)
70     FORMAT(2X,D15.8,'     (',I2,' )')
       PHY=PHIO
C
C      *$*$*$$$*$$$*$*$*$*$*$*$*$*$*$*$*$*$*$$*$*$*$*$$*$*$*$$$   RE
120    CALL DERV(N,M,K,NS,X,PARM,DF,KOUNT2)
       GMAGS=0.D0
C
C      CALC.  GRADIENT IN   G.
       DO 300 J=1,K
       G(J)=0.D0
       DO 110 I=1,N
       DO 110 L=1,NS
110    G(J)=G(J)+W(I,L)**2*(Y(I,L)-FN(I,L))*DF(I,J,L)
       G(J)=-2.D0*G(J)
       GMAGS=GMAGS+G(J)**2
300    CONTINUE
C
       GMAG=DSQRT(GMAGS)
C*************************************************************
C      PLOT PHY FOR LINES
       IF(MIN.LT.3) GO TO 320
        STEP=1.D-5
       NSTEP=5
C
       T(1)=PARM(1)-.00015
       T(2)=PARM(2)+.0005
       T(3)=PARM(3)+2.0005
       T(4)=PARM(4)-.0002
       T(5)=PARM(5)-.0005
       GD=0.D0
       DO 360 J=1,K
360    GD=GD+T(J)**2
       DO 370 J=1,K
370    T(J)=-T(J)/DSQRT(GD)
C
       CT=0.D0
       DO 391 J=1,K
391    CT=CT-T(J)*G(J)/GMAG
       DCT=DACOS(CT)/PIO
       WRITE(3,392)CT,DCT
392    FORMAT(' ****COS,THET= ',2D15.8)
C
C
       DO 330 I=1,300
       ALPHA=STEP*I
       DO 340 J=1,K
340    GTEMP(J)=PARM(J)-ALPHA*G(J)/GMAG
       CALL FCTN(N,M,K,NS,X,GTEMP,FN,KOUNT1)
       CALL PHI(N,NS,Y,FN,W,FY1,WORST,IWORST)
       DO 380  J=1,K
380    GTEMP(J)=PARM(J)+ALPHA*T(J)
       CALL FCTN(N,M,K,NS,X,GTEMP,FN,KOUNT1)
       CALL PHI(N,NS,Y,FN,W,FY2,WORST,IWORST)
330     WRITE(3,390)I,FY1,FY2
390    FORMAT(1X,' I,FY1,FY2=',I5,2D20.10)
C
       MIN=2
320    CONTINUE
C****  .......................................................
C
C      CALC. SCALE,DESCENT VECTOR T AND  SCALE PARMS
        TMAG=0.D0
       DO 310 J=1,K
       SCALE(J)=DABS(GMAG/G(J))
       T(J)=-G(J)*SCALE(J)
        TMAG=TMAG+T(J)**2
       PARM(J)=PARM(J)/SCALE(J)
310    CONTINUE
C
C
C      NORMALISE T
       TMAG=DSQRT(TMAG)
       DO 400 J=1,K
400    T(J)=T(J)/TMAG
C
C
C     CALC. ETOL STEP MAX-STEP
       STEP=PHY/MEG
       ETOL=STEP/1.D5
       STEPM=STEP*100.D0
C
       IF(NPR.GE.1)WRITE(3,510)ITER,STEP,STEPM,ETOL,MEG,BEE
510    FORMAT(' ITER,STEP,MAX,E,MEG,BEE=',I4,5D13.6)
C
C      MINIMIZE PHY ALONG LINE T
C
       GO TO(601,602),MIN
601   _CALL LMIN(N,M,K,NS,X,Y,W,PHY,FN,PARM,T,GTEMP,SCALE,ALPHA,
      1ETOL,STEP,STEPM,IT,NPR,KOUNT1)
       GO TO 610
C
602    CONTINUE
      _CALL GMIN(N,M,K,NS,X,Y,W,PHY,FN,PARM,T,GTEMP,SCALE,ALPHA,
      1ETOL,STEP,STEPM,IT,NPR,KOUNT1)
C
610    CONTINUE
C
C      TEST FOR CONVERGENCE
900    ITEST=0
       DO 910 J=1,K
       TEST=DABS(ALPHA*T(J))/(DABS(PARM(J))+TAU)
       IF(TEST.GT.EPS)ITEST=1
```

```
910   CONTINUE
      IF(ITEST.EQ.0)GO TO 1000
C
C
C     PRINT
1100  CONTINUE
C
C     DESCALE PARMS
      DO 1200 J=1,K
1200  PARM(J)=PARM(J)*SCALE(J)
C
      RMERR=DSQRT(PHY/N)
      RMS=RMERR/SEC
      WRITE(3,1110)ITER,IT,KOUNT1,KOUNT2,ALPHA,PHY,RMS
1110 _FORMAT(' ITER+LMIN IT=',2I5,' KOUNT1,2=',2I4,' ALPHA,PHY,RMS=',
     1,3D15.8,' PARMS FOLLOW')
      WRITE(3,1120)PARM
1120  FORMAT(/2X,5D15.8)
      IF(NPR.GE.1)WRITE(3,1130)T
1130  FORMAT(' T= ',5D15.8)
      ITER=ITER+1
      PHIO=PHY
C*************************************************
      IF(KOUNT1.GT.300)RETURN
C*************************************************
      GO TO 120
C
C
C     EXIT*$**$*$*$*$*$*$*$*$*$*$*$*$*$*$*$*$*$*$*$*$$**$*$$$*$*$$*$*$*$$*
1000  CONTINUE
      RMS=DSQRT(PHY/N)/SEC
      WRITE(3,1020)ITER,IT,KOUNT1,KOUNT2,ALPHA,PHY,RMS
1020 _FORMAT(//' GRAD2 CONVERGED AT ITER/IT=',2I4,' KOUNT1,2=',2I4,
     1' ALPHA,PHY,RMS= ',3D15.8,' PARMS FOLLOW..')
      DO 1040 J=1,K
C     ( DESCALE PARMS)
      PARM(J)=PARM(J)*SCALE(J)
      DB=PARM(J)/PIO
      SB=PARM(J)/SEC
1040  WRITE(3,1030)J,PARM(J),DB,SB
1030  FORMAT(' PARM ',I4,D15.8,' RAD ',F12.6,' DEG ',F12.6,' SEC')
      RETURN
C
C     READ FAIL
4010  WRITE(3,4011) NVAR
4011  FORMAT(///' READ ROUTINE FAIL',I5,'  VARIABLES WERE READ.')
      RETURN
      END
```

GRADNT

```
C     ELT,PROG=GRADNT                              MODIFICATION NO.=7
     _SUBROUTINE GRADNT(N,M,K,NS,NOPT,X,Y,PARM,FN,DF,W,G,GTEMP,T,SCALE,
     1MASK)
C     PROGRAM  4
C     MODIFIED STEEPEST DESCENT ALGORITHM FOR PARM. S ESTIMATION.
C     MOD. 1
C
C
C     ------------------------------------------------------OPTIONS
C     IF NOPT(1)=0 READS PARM,MASK FROM  K  CARDS.
C
C     NOPT(2) GE 1 ENABLES PRINT**
C     -----------------------------------------------------------
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION    DATA(2),NOPT(4)
      DIMENSION    X(N,M),Y(N,NS),FN(N,NS),W(N,NS),DF(N,K,NS)
      DIMENSION PARM(K),T(K),G(K),GTEMP(K),MASK(K),SCALE(K)
C
C     INITIALIZE
      PI=3.14159265360D0
      PIO=PI/180.D0
      SEC=PIO/3600.D0
      ITER=1
      ALPHA=1.D-2
      D1=0.5D0
      D2=1.D0
      GAMMA=80.D0*PI/180.D0
      TAU=1.D-15
      EPS=1.D-5
      KOUNT1=0
      KOUNT2=0
      COSGAM=DCOS(GAMMA)
      NPR=NOPT(2)
      NAL=0
C
C     *****************************************************. DATA INPUT S
C
      IF(NOPT(1).EQ.0) GO TO 6
6     CONTINUE
      WRITE(3,7)K
7     FORMAT(' *HEY BUGGERLUGS ENTER ',I3,'  PARMS.')
      DO 10 J=1,K
      CALL READ(NVAR,DATA,NERR)
      IF(NVAR.NE.2.OR.NERR.EQ.1) GO TO 4010
      PARM(J)= DATA(1)
      MASK(J)= DATA(2)
10    CONTINUE
C
C     ******************************************************* START*
C     CALC CURRENT PHI
      CALL FCTN(N,M,K,NS,X,PARM,FN,KOUNT1)
      CALL WT(N,M,NS,X,W)
      CALL PHI(N,NS,Y,FN,W,PHIO,WORST,IWORST)
      RMS=DSQRT(PHIO/K)/SEC
C
C     PRINT
      WRITE(3,50) PHIO,RMS
50    FORMAT(////' STEEPEST DESCENT ROUTINE,INITIAL PHI,RMS IS..',
     1D15.8,F12.6/'  INITIAL PARMS(MASKS) ARE..')
      DO 60 J=1,K
      GTEMP(J)=PARM(J)
60    WRITE(3,70) PARM(J),MASK(J)
70    FORMAT(2X,D15.8,'      (',I2,' )')
      PHY=PHIO
      GO TO 120
C
C     *$*$*$$*$$$*$**$*$*$*$*$*$*$*$*$*$*$*$*$$*$*$*$*$$$*$*$*$$$  RE
100   CONTINUE
C     DESCALE PARMS
      DO 130 J=1,K
130   GTEMP(J)=GTEMP(J)*SCALE(J)
120   CALL DERV(N,M,K,NS,X,GTEMP,DF,KOUNT2)
C
C     CALC. GRADIENT IN GTEMP
      DO 300 J=1,K
      GTEMP(J)=0.D0
      DO 110 I=1,N
      DO 110 L=1,NS
110   GTEMP(J)=GTEMP(J)+W(I,L)**2*(Y(I,L)-FN(I,L))*DF(I,J,L)
      GTEMP(J)=-2.D0*GTEMP(J)
C
C
300   CONTINUE
C
C     CALC. SCALE, CORRECTION VECTOR T AND SCALE PARMS
      R=K
      R=DSQRT(1.D0/R)
      DO 310 J=1,K
      SCALE(J)=R/GTEMP(J)
      T(J)=-R
      PARM(J)=PARM(J)/SCALE(J)
310   CONTINUE
      GO TO 400
C
C
C
600   ALPHA=ALPHA/4.D0
      NAL=NAL+1
C     CREATE TEMPORY DESCALED  PARM VECTOR IN GTEMP....
400     DO 410 J=1,K
410   GTEMP(J)=(PARM(J)+ALPHA*T(J))*SCALE(J)
      CALL FCTN(N,M,K,NS,X,GTEMP,FN,KOUNT1)
      CALL PHI(N,NS,Y,FN,W,PHY,WORST,IWORST)
      RMS=DSQRT(PHY/K)/SEC
C
CC
C     TEST PHY
500   IF(PHY.LT.PHIO) GO TO 700
      GO TO 600
C
C
C
C
C     CALC. ANGLE BETWEEN SUCCESSIVE CORR. VECTORS
700   CONTINUE
      IF(ITER.EQ.1) GOTO 900
      CTHETA=0.D0
      DO 710 J=1,K
710   CTHETA=CTHETA+T(J)*G(J)
C
C     IF ANGLE THETA GT GAMMA REDUCE ALPHA
800   IF(CTHETA.GT.COSGAM) GO TO 900
C
C     CALC NEW ALPHA
      ALPHA=ALPHA/4.D0
      DO 850 J=1,K
850   GTEMP(J)=PARM(J)+ALPHA*T(J)
900   CONTINUE
      DO 810 J=1,K
810   PARM(J)=PARM(J)+ALPHA*T(J)
C
C
C     TEST FOR CONVERGENCE
      ITEST=0
      DO 910 J=1,K
      TEST=DABS(ALPHA*T(J))/(DABS(PARM(J))+TAU)
      IF(TEST.GT.EPS) ITEST=1
910   CONTINUE
      IF(ITEST.EQ.0) GO TO 1000
C
C     PRINT
1100  WRITE(3,1110)ITER,PHY,RMS,CTHETA,ALPHA,NAL
1110 _FORMAT(' END ITER NO.',I5,'  PHI,RMS=',D15.8,F12.6,
     1' COSTHETA,ALPHA=',2D15.6,' NAL=',I5,' PARMS FOLLOW')
      WRITE(3,1120) PARM
1120  FORMAT(/2X,5D15.8)
      IF(NPR.GE.1) WRITE(3,1130)T
1130  FORMAT(/'   T=',5D15.8)
C
C     PREPARE FOR NEXT ITER.
1200  ITER=ITER+1
C
C     SAVE OLD UNIT VECTOR T IN G. AND CALC. NEW ALPHA & PARMS.
      NAL=0
      DO 1210 J=1,K
      PARM(J)=GTEMP(J)*SCALE(J)
1210  G(J)=T(J)
      PHIO=PHY
      GO TO 100
C
C     EXIT*$**$*$*$*$*$*$*$*$*$*$*$*$*$*$*$*$*$*$$**$*$$$*$*$$*$*$*$$*
1000  DO 1010 J=1,K
1010   PARM(J)=GTEMP(J)
      WRITE(3,1020) ITER,PHY,CTHETA,ALPHA,KOUNT1,KOUNT2,NAL
1020 _FORMAT(//' M.S.D. CONVERGED ITER NO.',I6,'  FINAL PHI=',D15.8/'  C
     1OSTHETA,ALPHA,KOUNT1,2,NAL=',2D15.8,3I6,'  PARMS FOLLOW..')
      WRITE(3,1030)PARM
1030   FORMAT(/2X,D15.8)
      RETURN
C
C     READ FAIL
4010  WRITE(3,4011) NVAR
4011  FORMAT(///' READ ROUTINE FAIL',I5,'  VARIABLES WERE READ.')
      RETURN
      END
```

IBD

```
C     ELT=B     IBD ......MOD 1
      INTEGER FUNCTION IBD(I,K,J)
C     TAKES K BITS FROM WORD J STARTING AT THE I-TH FROM RIGHT
C     (COUNTING 0,1,2..) AND CONVERTS BCD TO DECIMAL.
      INTEGER FLD
      M=36
      MM=M-K-I
      JJ=FLD(MM,K,J)
      LL=(K-1)/4
      IBD=0
      DO 5  L=0,LL
      MM=M-4*(L+1)
      KK=FLD(MM,4,JJ)
      IBD=IBD+KK*10**L
5     CONTINUE
      RETURN
      END
```

LMIN

```
C     ELT=LMIN POWELL LINE MINIMIZATION.      MOD 10
      SUBROUTINE LMIN(N,M,K,NS,X,Y,W,PHY,FN,B,T,BNEW,SCALE,ALPHA,ETOL,
     1STEP,STEPM,IT,NPR,K1)
C
C
C     ETOL=ACCURACY
C     ETC..................
C
      DIMENSION X(N,M),W(N,NS),FN(N,NS),Y(N,NS),B(K),T(K),BNEW(K)
      DIMENSION SCALE(K)
      DIMENSION D(3),FY(3),TEST(3),COMP(3),I(3)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
      IT=1
C
      WRITE(3,40)ETOL,STEP,STEPM
40    FORMAT(' POWELL LMIN.  ETOL,STEP,MAX=',3D15.8)
      D(1)=0.D0
      FY(1)=PHY
      D(2)=STEP
      DO 50    J=1,K
50    BNEW(J)=B(J)+D(2)*T(J)
      CALL FCTN(N,M,K,NS,X,BNEW,FN,K1)
      CALL PHI(N,NS,Y,FN,W,FY(2),WZ,IWZ)
C
C     GET THIRD PNT ON LINE
100   CONTINUE
      D(3)=-STEP
      IF(FY(1).GT.FY(2)) D(3)=2.D0*STEP
300   DO 310 J=1,K
310   BNEW(J)=B(J)+D(3)*T(J)
      CALL FCTN(N,M,K,NS,X,BNEW,FN,K1)
          CALL PHI(N,NS,Y,FN,W,FY(3),WZ,IWZ)
C
C     CALC TURNING PNT
350   CONTINUE
      BC=D(2)-D(3)
      CA=D(3)-D(1)
      AB=D(1)-D(2)
      BCS=D(2)+D(3)
      ABS=D(1)+D(2)
      CAS=D(3)+D(1)
      BCP=BC*FY(1)
      CAP=CA*FY(2)
      ABP=AB*FY(3)
      DMIN=0.5D0*(BCS*BCP+CAS*CAP+ABS*ABP)/(BCP+CAP+ABP)
C     CALC SECOND DERV AS CHECK
      D2=(BCP+CAP+ABP)/(AB*BC*CA)
C
500   IF(D2.LT.0.D0)GO TO 600
      LAM=2
      GO TO 750
C
C     TURNING PNT IS A MAXIMUM OR REQUIRED STEP IS TOO LARGE.
C     WHICH OF D(1) D(3) D(2) IS FURTHEST FROM DMIN
700   LAM=1
750   DO 760 J=1,3
760   TEST(J)=DABS(DMIN-D(J))
      CALL SORT(TEST,COMP,I,3)
      INDEX=I(3)
C
C
C     STEP IN DIRN. OF DECREASING PHY AND DISCARD OLD POINT FURTHEST
C         FROM DMIN
C
800   ESTEP=STEPM
      IF(DMIN.LT.0D0)ESTEP=-STEPM
C
C     ENSURE ESTEP DOES NOT DUPLICATE ANY EXISTINGPOINT
805   DO 830 J=1,3
      IF(DABS(ESTEP-D(J)).LT.ETOL*1.D-2) GO TO 850
830   CONTINUE
      GO TO 840
850   ESTEP=ESTEP+STEP
      IF(DMIN.LT.0.D0)ESTEP=ESTEP-2.D0*STEP
      GO TO 805
C
840   DO 810 J=1,K
810   BNEW(J)=B(J)+ESTEP*T(J)
      CALL FCTN(N,M,K,NS,X,BNEW,FN,K1)
      CALL PHI(N,NS,Y,FN,W,PHIN,WZ,IWZ)
      D(INDEX)=ESTEP
      FY(INDEX)=PHIN
C
      WRITE(3,820)IT,LAM,FY,D,DMIN,INDEX
820  _FORMAT(' LMIN/IT,LAM=',2I4,'PHY,D=',6D15.8,/' DMIN,REPLACED=',
     1,D15.8,I4)
      GO TO 950
C
C
C
C     CHECK THAT STEP DMIN IS NOT TOO LARGE
600   IF(DABS(DMIN).GT.STEPM)GO TO 700
C
C     CHECK IF DMIN WITHIN ETOL OF D(1) D(2) OR D(3)
      LAM=0
      DO 610 J=1,3
      ERR=DABS(DMIN-D(J))
      IF(ERR.LT.ETOL) GO TO 620
610   CONTINUE
      GO TO 900
C
620   ALPHA=D(J)
      PHY=FY(J)
      GO TO 650
C
C
C
C     USE DMIN AS NEW POINT
900   DO 910 J=1,K
910   BNEW(J)=B(J)+DMIN*T(J)
      CALL FCTN(N,M,K,NS,X,BNEW,FN,K1)
      CALL PHI(N,NS,Y,FN,W,PHIN,WZ,IWZ)
C
C     FIND LARGEST OF FY(1) FY(2) FY(3) AND REPLACE WITH PHIN UNLESS
C        TWO OF D(1) D(2) D(3) BRACKET DMIN.
C
      CALL SORT(FY,TEST,I,3)
      INDEX=I(3)
C
      CALL SORT(D,TEST,I,3)
      IF(DMIN.GT.TEST(1).AND.DMIN.LT.TEST(2)) INDEX=I(3)
      IF(DMIN.GT.TEST(2).AND.DMIN.LT.TEST(3)) INDEX=I(1)
```

```
920   FY(INDEX)=PHIN
      D(INDEX)=DMIN
C
      WRITE(3,930)IT,LAM,FY,D,DMIN,INDEX
930  _FORMAT(' LMIN/IT,LAM=',I4,I2,' FY,D=',6D12.5,' DMIN,REPLACED='
     1,D12.5,I4)
C
C
950   IT=IT+1
      GO TO 350
C
C     EXIT
650   DO 640 J=1,K
640   B(J)=B(J)+ALPHA*T(J)
C
      CALL FCTN(N,M,K,NS,X,B,FN,K1)
C     K1=K1-1
C
      WRITE(3,660)IT,LAM,FY,D,PHY,ALPHA
660  _FORMAT(' LMIN,EXIT/IT,LAM=',2I4,' FY,D=',6D15.8,/
     1'          PHY,ALPHA ',2D15.8)
      RETURN
      END
C
```

MARQDT

```
C     ELT=C  MARQUARDT ALGORI,HM WITH PARAMETER MASKING.  MOD# 23A.
     _SUBROUTINE MARQRT(N,M,K,NS,NOPT,X,Y,PARM,FN,DF,W,A,ATEMP,G,GTEMP,T
     1,SCALE,MASK)
C
C     MARQUARDT ALGORITHM FOR NON LIN. PARM. ESTIMATION.
C
C
C     ---------------------------------------------------------OPTIONS
C     IF NOPT(1)=0 READS PARM(J)&MASK(J)(=0 IF PARM FROZEN) FROM K  CARD
C
C     NOPT(2) GE 1 ENABLES PRINT..
C
C     NOPT(3)=0 EPS=1.D-5 CUTOFF AFTER ITEND ITERATIONS,
C            =1 EPS=1.D-3 NO CUTOFF,=2 EPS=1.D-5 NO CUTOFF.
C
C     ----------------------------------------------------------------
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION    DATA(2),NOPT(4)
      DIMENSION    X(N,M),Y(N,NS),FN(N,NS),W(N,NS),DF(N,K,NS)
      DIMENSION    PARM(K),T(K),G(K),GTEMP(K),SCALE(K),MASK(K)
      DIMENSION  A(K,K),ATEMP(K,K),WB(2)
      REAL EPSLON,FRAT,FISH
C     INITIALIZE.
      PI=3.141592653589793
      PIO=PI/180.D0
      ASEC=PI/(180.DC*3600.D0)
      ITER=1
      MSUM=K
      ALAMDA=1.D-2
      ANU=10.D0
      TAU=1.D-15
      EPS=1.D-5
      IF(NOPT(3).EQ.1)EPS=1.D-3
      ITEND=25
      COSGMO=DCOS(PI/4.D0)
      KOUNT1=0
      KOUNT2=0
      NPR=NOPT(2)
      EPSLON=1.E-15
C
C
C     ********************************,****************** DATA INPUT
C
      IF(NOPT(1).EQ.0) GO TO 6
6     CONTINUE
      WRITE(3,11)K
11    FORMAT(' ENTER ',I5,' PARMS & MASKS ( 1 PAIR PER LINE)')
      DO 10 J=1,K
      CALL READ(NVAR,DATA,NERR)
      IF(NVAR.NE.2.OR.NERR.EQ.1) GO TO 4010
      PARM(J)= DATA(1)
      MASK(J)= DATA(2)
10    CONTINUE
C
C     ****************************************************** START.
C
C     CALC CURRENT PHI
      CALL FCTN(N,M,K,NS,X,PARM,FN,KOUNT1)
      CALL WT(N,M,NS,X,W)
      CALL PHI(N,NS,Y,FN,W,PHID,WORST,IWORST)
      RMS=DSQRT(PHID/N)/ASEC
      WORST=DSQRT(WORST)/ASEC
C
C     PRINT
      WRITE(3,7) PHID,RMS,WORST,IWORST
7    _FORMAT(//' MARQUARDT NON-LIN. PARM.ESTIMATION ROUTINE,INITIAL PHI
     1IS',D15.8,' RMS ERRS=',2F12.4,I5/' INITIAL PARMS ARE ')
      DO 9 J=1,K
9     WRITE(3,8) PARM(J),MASK(J)
8     FORMAT(2X,D15.8,I6)
C
C
C
C     ...................................................RE-ENTRY
100   CONTINUE
      CALL DERV(N,M,K,NS,X,PARM,DF,KOUNT2)
C     GENERATE SCALED MATRICES A , G
      SCALE(1)=0.D0
      CALL MATEQN(N,M,K,NS,Y,FN,W,DF,A,G,SCALE)
C
C
C     COMPRESS A,G,SCALE  IF PARMS MASKED & SOME ARE TO BE FROZEN.
      CALL CMPRES(K,MASK,A,G,SCALE,MSUM)
      MSQ=MSUM**2
      ENCODE(192,V)MSUM
192   FORMAT('(1X,',I3,'F5.3)')
      IF(ITER.NE.1.OR.NPR.LT.1) GO TO 196
       WRITE(3,V)(A(MS),MS=1,MSQ)
196   NUU=2*N-MSUM
C
C
      IF(ITER.EQ.1.AND.MSUM.NE.K)WRITE(3,195)MSUM
195  _FORMAT(/' ***ONLY ',I3,' PARMS INVOLVED IN ITERATIONS,THE',
     1' REST ARE FROZEN AT START VALUES.')
C
C
C
190   IF(ALAMDA.LT.1.D-13) GO TO 170
      LAMDA=1
      DLAMDA=ALAMDA/ANU
      GO TO 180
C
170   LAMDA=2
      DLAMDA=ALAMDA
C     SOLVE EQN. (A + LAM. I ) T = G FOR T
180   DO 181 J=1,MSUM
      GTEMP(J)=G(J)
      INDEX=J+(J-1)*MSUM
       ATEMP(INDEX)=A(INDEX)+DLAMDA
      DO 181 KK=1,MSUM
      INDEX=J+(KK-1)*MSUM
      IF(J.NE.KK) ATEMP(INDEX)=A(INDEX)
181   CONTINUE
C
      CALL DGELG(GTEMP,ATEMP,MSUM,1,EPSLON,IER)
      IF(IER.NE.0) GO TO 4000
C
      DO 182 J=1,MSUM
182   T(J)=GTEMP(J)/SCALE(J)
C
C
C     EXPAND PARMS  IF MASKED
      IF(NPR.GE.3)WRITE(3,193)T
193   FORMAT(' T BEFORE EXPAND= ',5(D18.8))
      CALL EXPAND(K,MASK,T,MSUM)
      IF(NPR.GE.3)WRITE(3,194)T
194   FORMAT(' T AFTER EXPAND= ',5(D18.8))
C
C     CALC. GTEMP =TEMPORARY PARM VECTOR
220   DO 230 J=1,K
230   GTEMP(J)=PARM(J)+T(J)
C
C     CALC. FN VALUES.
      CALL FCTN(N,M,K,NS,X,GTEMP,FN,KOUNT1)
C     CALC. PHI
      CALL PHI(N,NS,Y,FN,W,PHY,WORST,IWORST)
      SIGSQ=PHY/NUU
      RMS=DSQRT(PHY/N)/ASEC
      WORST=DSQRT(WORST)/ASEC
C
C
C     TEST FOR CONVERGENCE.+ EXIT IF NO. OF ITERS EXCESSIVE.
      ITEST=0
      TSTMAX=0.D0
      DO 290 J=1,K
      IF(MASK(J).EQ.0) GO TO 290
      TEST=DABS(T(J))/(DABS(PARM(J))+TAU)
      IF(TEST.GT.EPS) ITEST=1
      IF(TSTMAX.GE.TEST) GO TO 290
      TSTMAX=TEST
      MAXTST=J
290   CONTINUE
      IF(ITEST.EQ.0) GO TO 1500
      IF(NOPT(3).EQ.C.AND.ITER.GE.ITEND.AND.PHY.LE.PHID)GO TO 400
C
C
      GO TO (1000,2000),LAMDA
C
C
400   WRITE(3,410)ITER,TSTMAX,MAXTST,T
410  _FORMAT(/' TOO MANY ITERATIONS MARQRT AXED AT ITER=',I4,' TEST=',
     1D20.10,' ON PARM ',I4,/' CURRENT T=',5(D20.10))
      GO TO 1500
C
C     LAMDA = ONE
1000  IF(PHY.GT.PHID) GO TO 170
      ALAMDA=ALAMDA/ANU
      GO TO 2100
C
C     LAMDA = TWO
2000  IF(PHY.LE.PHID) GO TO 2100
C
C     LAMDA  = THREE
      LAMDA=3
      STEP=1.D0
3200  ALAMDA=ALAMDA*ANU
      DLAMDA=ALAMDA
      DO 3210 J=1,MSUM
      INDEX=J+(J-1)*MSUM
      GTEMP(J)=G(J)
      ATEMP(INDEX)=A(INDEX)+DLAMDA
      DO 3210 KK=1,MSUM
      INDEX=J+(KK-1)*MSUM
      IF(J.NE.KK) ATEMP(INDEX)=A(INDEX)
3210   CONTINUE
C     SOLVE AQN (A + LAM* I) T = G
      CALLDGELG(GTEMP,ATEMP,MSUM,1,EPSLO..,IER)
      IF(IER.NE.0) GO TO 4000
C
C
C     CALC. COSGAM= COSINE ANGLE GAMMA FROM T* , G* .
      ANUM=0.D0
      DENOM=0.D0
      DO 3220 J=1,MSUM
      ANUM=ANUM+GTEMP(J)*G(J)
      IF(NPR.GE.1) WRITE(3,3221) J,GTEMP(J),G(J)
3221  FORMAT(' MARQRT 3221 J,T*,G*=',I6,2D15.8)
      DO 3220 KK=1,MSUM
      DENOM=DENOM+(GTEMP(J)*G(KK)*SCALE(KK)/SCALE(J))**2
3220  CONTINUE
      COSGAM=ANUM/DSQRT(DENOM)
      IF(NPR.GE.1) WRITE(3,3222) ANUM,DENOM, COSGAM
3222  FORMAT(' MARQRT 3222 ',3D15.8)
C
C     CALC. T FROM T*
      DO 3230 J=1,MSUM
3230  T(J)=GTEMP(J)/SCALE(J)
C
C     EXPAND T IF PARMS MASKED.
      IF(NPR.GE.3)WRITE(3,193)T
      CALL EXPAND(K,MASK,T,MSUM)
      IF(NPR.GE.3)WRITE(3,194)T
C
C     CALC. GTEMP TEMPORARY PARM VECTOR.
3300  DO 3310 J=1,K
3310  GTEMP(J)=PARM(J)+STEP*T(J)
C
      CALL FCTN(N,M,K,NS,X,GTEMP,FN,KOUNT1)
      CALL PHI(N,NS,Y,FN,W,PHY,WORST,IWORST)
      SIGSQ=PHY/NUU.
      RMS=DSQRT(PHY/N)/ASEC
      WORST=DSQRT(WORST)/ASEC
C
      IF(PHY.LE.PHID) GO TO 2100
C
C     IF GAMMA GE. GAMMAO KEEP MULTIPLYING UP LAMDA
C     IF GAMMA LT. GAMMAO REDUCE STEP SIZE.
      IF(COSGAM.LE.COSGMO) GO TO 3200
3400  STEP=STEP/ANU
      GO TO 3300
C
C     PREPARE FOR NEXT ITERATION
2100  PHID=PHY
      DO 2110 J=1,K
2110  PARM(J)=GTEMP(J)
C
C     PRINT
     _WRITE(3,1210)ITER,PHID,ALAMDA,LAMDA,KOUNT1,KOUNT2,RMS,IWORST,
     1WORST,SIGSQ,NUU
      IF(NPR.GE.1) WRITE(3,1250) PARM
1250  FORMAT('  CURRENT PARMS=',5(D15.8))
      IF(LAMDA.EQ.3) WRITE(3,1220) STEP
      ITER=ITER+1
      GO TO 100
C
C
C     EXIT             ***************,**********************,****
1500  CONTINUE
C
C     SAVE N,NUU,PHY,SIGSQ IN W  FOR TESTS
C
      W(1,1)=N
      W(2,1)=NUU
      W(3,1)=PHY
      W(4,1)=SIGSQ
C
C
      DO 1510 J=1,K
1510  PARM(J)=GTEMP(J)
      PHID=PHY
C
C     PRINT             ******************************,**,*,*****
C
     _WRITE(3,1210)ITER,PHID,ALAMDA,LAMDA,KOUNT1,KOUNT2,RMS,IWORST,
     1WORST,SIGSQ,NUU
      IF(LAMDA.EQ.3) WRITE(3,1220) STEP
1210  FORMAT(///' ITER,PHID,ALAM,LAM,KOUNT1,2=',I6,2(3X,D15.8),3I6,
     1/' RMS ERROR ARCSEC=',F12.4,' WORST OBSVN & ERROR=',I6,2X,F12.4,
     2' SIGSQ(NU)=',D16.6,I4)
1220  FORMAT('  STEP SIZE= ',F20.15)
C
C     GET Y AVERAGE
      RESAVG=0.D0
      DO 1260 I=1,N
1260  RESAVG=RESAVG+Y(I,1)-X(I,1)+(Y(I,2)-X(I,2))*DCOS(X(I,1))
      RESAVG=RESAVG/(2.D0*N)
C
C     PRINT A MATRIX
      IF(NPR.GE.1)WRITE(3,V)(A(MS),MS=1,MSQ)
C
C
C     GET SUMSQURE DUE REGRESS,AND CORR FOR MEAN
      SSE=0.D0
      SSR=0.D0
      SSM=0.D0
      WB(1)=1.D0
      DO 1270 I=1,N
      WB(2)=DCOS(X(I,1))**2
      DO 1270 L=1,2
      SSE=SSE+(FN(I,L)-Y(I,L))**2*WB(L)
      SSR=SSR+(FN(I,L)-X(I,L)-RESAVG)**2*WB(L)
      SSM=SSM+(Y(I,L)-X(I,L)-RESAVG)**2*WB(L)
1270  CONTINUE
C
C     CALC MULT REGRESS COEFF-RSQ,MEAN SQR RATIO-SQRAT
      NUU1=MSUM-1
      NU2=NUU
      RSQ=SSR*100.D0/SSM
      SQRAT=SSR/(SIGSQ*NUU1)
      FRAT=SQRAT
C*****FISH PRANGS EASILY
      IF(FRAT.LT.10.0)PROB=100.D0*(1.D0-FISH(FRAT,NUU1,NU2))
C
C
      WRITE(3,1290)RESAVG
1290  FORMAT(//' ************CONVERGENCE** RESID AVGE=',D13.5)
C
      WRITE(3,1280)SSE,SSR,SSM,SQRAT,NUU1,NU2,PROB,RSQ
1280 _FORMAT(//' PHI,SSR,SSM=',3D14.6,' MEANSQRAT=',F13.5,2I5,
     1' PROB,MULTREGCOEFF=',F10.5,F8.4)
C
C
C
      CALL SHIFT(PARM(3))
      DO 1530 J=1,K
      GTEMP(J)=PARM(J)/PIO
      G(J)=PARM(J)/ASEC
1530  WRITE(3,1240)J,PARM(J),GTEMP(J),G(J)
1240 _FORMAT(//' FINAL PARM  ',I3,6X,D17.10,' QV ',F15.8,' DEGREES OR ',
     1F12.4,' ARCSEC.')
      RETURN
C
C
C
C
C
C
C     DGELG FAIL
4000  WRITE(3,4001) IER
4001  FORMAT(///' MATRIX EQN SOLN. ROUTINE FAIL,IER(DGELG)=',I4)
      RETURN
C
C
C     READ FAIL
4010  WRITE(3,4011) NVAR
4011  FORMAT(///' READ ROUTINE FAIL',I5,'  VARIABLES WERE READ.')
      RETURN
      END
```

MATEQN

```
      SUBROUTINE MATEQN(N,M,K,NS,Y,FN,W,DF,A,G,SCALE)
C     GIVEN DATA  + FN VALUES ,DERIVATIVES WRT PARMS SETS UP EQN.A*.T* =
C     ALSO GIVES SCALE VECTOR
C     A,G ARE SCALED ONLY IF ON INPUT SCALE(1)=0.0
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Y(N,NS),FN(N,NS),W(N,NS),DF(N,K,NS)
      DIMENSION  A(K,K),G(K),SCALE(K)
C
      NPR=0
      ISCALE=9
      IF(SCALE(1).EQ.0.D0) ISCALE=0
C
      DO 21 J=1,K
      G(J)=0.D0
      DO 20 KK=1,K
      A(J,KK)=0.D0
      IF(J.LE.KK) GO TO 31
      A(J,KK)=A(KK,J)
      GO TO 20
C
C     CALC. A BY SUMMING OVER DATA PTS AND FN. SECTIONS.
31    DO 30 L=1,NS
      DO 30 I=1,N
30    A(J,KK)=A(J,KK)+W(I,L)**2*DF(I,J,L)*DF(I,KK,L)
      IF(J.NE.KK) GO TO 20
C     CALC. G BY SUMMING OVER DATA PTS AND FN. SECTIONS.
      DO 32 L=1,NS
      DO 32 I=1,N
32    G(J)=G(J)+W(I,L)**2*(Y(I,L)-FN(I,L))*DF(I,J,L)
20    CONTINUE
C     CALC. SCALE (J)  =SQRT OF A(J,J)
      SCALE(J)=DSQRT(A(J,J))
      IF(NPR.EQ.1)WRITE(3,621) (A(J,JJ),JJ=1,K),G(J),SCALE(J)
621   FORMAT(' 621  MATEQN ',5D13.6,D22.16,D13.6)
21    CONTINUE
C
C
      IF(ISCALE.NE.0) GO TO 41
C     SCALE A AND  G
      DO 40 J=1,K
      G(J)=G(J)/SCALE(J)
      DO 40 KK=1,K
      A(J,KK)=A(J,KK)/(SCALE(J)*SCALE(KK))
40    CONTINUE
      IF(NPR.EQ.1) WRITE(3,622)((A(J,JJ),JJ=1,K),G(J),SCALE(J),J=1,K)
622   FORMAT(' 622  MATEQN ',7D15.8)
41    RETURN
      END
```

MATMPY

```
      SUBROUTINE MATMPY(A,B,C,N,M,L)
C     C=A*B WHERE A IS N*M, B  IS  M*L AND C IS  N*L.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION  A(N,M),B(M,L),C(N,L)
      DO 20 I=1,N
      DO 20 K=1,L
      C(I,K)=0.D0
      DO 20 J=1,M
20    C(I,K)=C(I,K)+A(I,J)*B(J,K)
      RETURN
      END
```

MOTEST

```
C     ELT=T    MOTEST   MOD NO. = 9.  *************************************
      SUBROUTINE MOTEST(N,M,K,NS,NOPT,X,Y,FN,W,PARM,MASK)
C
C       ...PROGRAM FIGHT...TESTS ACCURACY OF MODELS FIT
C
C     NOPT(1) GT 0 ENABLES PRINT,=9 PLOTS RESIDUALS,=8 BOTH.
C     NOPT(4)=9 SUPPRESSES COMPARISON..ONLY SUM GENERATED.
C
      IMPLICIT  REAL*8(A-H,O-Z)
      DIMENSION NOPT(4),X(N,M),Y(N,NS),FN(N,NS),W(N,NS),PARM(K),MASK(K)
      DIMENSION NDAT(4)
C     CAUTION TEMPORARILY DIMENSIONED VARIABLES..
      REAL DEC(160),HA(160),DD(160),DH(160),DR(160),FRATIO,FISH
      PI=3.14159265358979300
      PIO=PI/180.D0
      SEC=3600.D0/PIO
      NPR=NOPT(1)
      IF(NOPT(1).EQ.9)NPR=0
      IDATA=NOPT(3)
      NDAT(1)=NPR
      NDAT(2)=0
      NDAT(3)=0
      NDAT(4)=0
C
C     EXTRACT NFIT,NU2,PHIFIT,SIG2 FROM W.
C
      NFIT=W(1,1)
      NU2=W(2,1)
      PHIFIT=W(3,1)
      SIG2=W(4,1)
C
C
C
C
C     ****************************************************************
C     READ IN NEW TELESCOPE DATA FROM FILE 14.
200   CONTINUE
C
      READ(14)NOBS
      WRITE(3,210)NOBS
210  _FORMAT(' N=NO. OF DATA PTS HAS BEEN RESET TO ',I5,
     1' FOR FOLLOWING FIT TESTS.')
      N=NOBS
      DO 220 J=1,NOBS
      READ(14)X(J,1),X(J,2),Y(J,1),Y(J,2)
230   FORMAT(' XY FILE 14 ENTRY # ',I4,5X,4D20.10)
      CALL SHIFT2(X(J,2))
      CALL SHIFT2(Y(J,2))
      IF(NPR.GT.1)WRITE(3,230)J,X(J,1),X(J,2),Y(J,1),Y(J,2)
220   CONTINUE
C     ****************************************************************
400   CONTINUE
      CALL FCTN(N,M,K,NS,X,PARM,FN,IFFFF)
      CALL WT(N,M,NS,X,W)
      CALL PHI(N,NS,Y,FN,W,PHY,ERRMAX,JMAX)
C     ****************************************************************
C     OPTIONALLY SUPRESS COMPARISON.
500   IF(NOPT(4).NE.9) GO TO 700
C     RUBBARA.        :
600   RETURN
C     ****************************************************************
700   CONTINUF
      RMS=DSQRT(PHY/N)*SEC
      NUU=2*N
      SIGSQ=PHY/NUU
      ERRMAX=DSQRT(ERRMAX)*SEC
       WRITE(3,940) PHY,RMS,ERRMAX,JMAX,SIGSQ,NUU
940  _FORMAT(' ON-SKY PHY=',D15.8,' RMS(ARCSEC)=',D15.8,
     1' MAXIMUM ON-SKY ERR.=',D15.8,' AT OBS#',I5,
     2/'      SIGSQ(NU)=',D16.6,I4)
C
C     FIT ASSESMENT
C
      NU1=2*(N-NFIT)
      SIG1=(PHY-PHIFIT)/NU1
      FRATIO=SIG1/SIG2
      PROB=(1.D0-FISH(FRATIO,NU1,NU2))*100.D0
C
      WRITE(3,950)PHY,PHIFIT,SIG1,SIG2,FRATIO,NU1,NU2,PROB
950   FORMAT(/' PHY,PHIFIT,SIG1,2= ',4D13.5,' F,NU1,2,PROB= ',F7.4,
     12I4,F11.5/)
C     ****************************************************************
C     PLOT&LIST RESIDUALS.........
      IF(NOPT(1).NE.9.AND.NOPT(1).NE.8)GO TO 800
      DO 810 J=1,NOBS
      DEC(J)=X(J,1)/PIO
      HADES=X(J,2)
      CALL SHIFT2(HADES)
      HA(J)=HADES*12.D0/PI
      DD(J)=(Y(J,1)-FN(J,1))*SEC
      DH(J)=(Y(J,2)-FN(J,2))*DCOS(X(J,1))*SEC
      DR(J)=SQRT(DD(J)**2+DH(J)**2)
      IF(NPR.GT.0)WRITE(3,820)J,DEC(J),HA(J),DD(J),DH(J),DR(J)
820  _FORMAT(' DEC,HA FOR PNT.,',I4,2F10.4,8X,'RESIDUALS DD,DH*COS=',
     12F10.4,'    DR= ',F10.4)
810   CONTINUE
C
       CALL APLOT(N,DEC,' DEC ',DD,' DDEC',1)
       CALL APLOT(N,DEC,' DEC ',DH,'DH*COS',2)
       CALL APLOT(N,HA ,' HA  ',DD,'DDEC ',3)
       CALL APLOT(N,HA ,' HA  ',DH,'DH*COS',4)
       CALL APLOT(N,DEC,' DEC ',DR,'DRES  ',5)
       CALL APLOT(N,HA ,' HA  ',DR,'DRES  ',6)
C     ****************************************************************
800   RETURN
C
C
      END
```

MXV

```
C     ELT=W    MXV MOD=1
      SUBROUTINE MXV(C,A,B)
C     MULTIPLIES MATRIX BY VECTOR (3D)
      REAL*8 A(3,3),B(3),C(3)
      DO 5 J=1,3
      C(J)=0.D0
      DO 5 I=1,3
5     C(J)=C(J)+A(J,I)*B(I)
      RETURN
      END
```

NUTATE

```
C     ELT=J    NUTATE  MOD=1
      SUBROUTINE NUTATE(RM,TJ,RT,KSHORT)
C     NUTATION(RECTANGULAR)ALGORITHM.
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 L,LD,RM(3),RT(3),S(3,3)
      PI=3.14159265358979300
      PIO=PI/180.D0
C*****PARAMETERS W.R.T. MEAN EQUINOX OF DATE.
      T=(TJ-2415020.D0)/36525.D0
      ON=259.183275000-1934.142008 3*T+0.20778D-2*T*T+0.22D-5*T**3
      OS=279.69667800+36000.768925*T+0.3025D-3*T**2
      OM=270.43416390O+481267.8831417*T-0.11333D-2*T**2+0.1889D-5*T**3
      G=(6189.03*T+1.63*T*T+0.012*T*T*T)/3600.D0
      G=281.220833+G
      GD=(-37.17*T*T-0.045*T*T*T)/3600.D0
      GD=GD+334.329556+4069.03403333*T
      ON=ON*PIO
      OS=OS*PIO
      OM=OM*PIO
      G=G*PIO
      GD=GD*PIO
      L=OM-GD
      LD=OS-G
      F=OM-ON
      D=OM-OS
C       **************************************************************
C*****SERIES FOR DPHI,DEPS INCLUDES ALL TERMS WITH COEFFS > 0.01 ARCSEC.
C
C      INITIALIZE KSHORT GT ZERO TO SKIP SHORT PERIOD TERMS.
       DPHI=-(17.2327+0.01737*T)*DSIN(ON)+0.2088*DSIN(2.D0*ON)-1.2729*DSI
      CN(F+F-D-D+ON+ON)+0.1261*DSIN(LD)-0.0497*DSIN(LD+F+F-D-D+ON+ON)+0.0
      C214*DSIN(F+F-D-D-LD+ON+ON)+0.0124*DSIN(F+F-D-D+ON)
C
       IF(KSHORT.GT.0) GO TO 1
       DPHI=DPHI-0.2037*DSIN(F+F+ON+ON)+0.0675*DSIN(L)-0.0342*DSIN(F+F+ON
      C)-0.0261*DSIN(L+F+F+ON+ON)-0.0149*DSIN(L-D-D)+0.0114*DSIN(F+F-L+ON
      C+ON)
1      CONTINUF
C
       DEPS=(9.210+0.00091*T)*DCOS(ON)-0.0904*DCOS(ON+ON)+0.5522*DCOS(F+F
      C-D-D+ON+ON)+0.0216*DCOS(LD+F+F-D-D+ON+ON)
C
       IF(KSHORT.GT.0)GO TO 2
       DEPS=DEPS+0.0884*DCOS(F+F+ON+ON)+0.0183*DCOS(F+F+ON)+0.0113*DCOS(F
      C+F+ON+ON+L)
2      CONTINUE
C
       EPSO=23.452294D0-0.0130125D0*T-1.64D-6*T**2+5.03D-7*T**3
       DPHI=DPHI*PIO/3600.D0
       DEPS=DEPS*PIO/3600.D0
       EPSO=EPSO*PIO
       EPS=EPSO+DEPS
C      CALCULATE NUTATION TO SECOND ORDER IN DPHI,DEPS.
       S(1,1)=1.D0
       S(1,2)=-DPHI*DCOS(EPSO)
       S(1,3)=-DPHI*DSIN(EPSO)
       S(2,1)=DPHI*DCOS(EPS)
       S(2,2)=DCOS(DEPS)
       S(2,3)=-DSIN(DEPS)
       S(3,1)=DPHI*DSIN(EPS)
       S(3,2)=-S(2,3)
       S(3,3)=S(2,2)
C
       CALL MXV(RT,S,RM)
       RETURN
       END
```

ORGPOL

```
C     ELT=E  ORGPOL  MOD NO.=6
     _SUBROUTINE ORGPOL(N,J,K,KD,NOPT,IW,X,W,P,ALPHA,JS,JT,JG,JQ,JL,
     1JRA,JRB)
C      SEQENTIALLY COMPUTES 2D FORSYTHE POLYNOMIALS TO 2D RANDOMLY DISTRIB
C      DATA X    WEIGHTS=W
C      TO GET J-TH ORDER ORGPOL MUST BE CALLED J+1 TIMES WITH J=0,1,2..J
C      REQUIRES=SUBROUTINE ANABEL,IW=0 IF W(I,1)=W(I,2)...IW=1 IF NOT
C      RETURNS==INDEX CONSTANTS JL,JQ,JG AND ALPHA,P FOR ALL I
C      NOTE CASE J=0 STORED WITH INDEX J=KD
C
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION NOPT(4),JL(KD),JQ(KD),JG(KD),X(N,2),W(N,2),P(N,KD,2)
       DIMENSION ALPHA(KD,KD,2)
        DIMENSION  JS(KD),JT(KD),JRA(KD),JRB(KD )
C
C
C::::START ON P(N,J,1)
       NPR=NOPT(1)
       M=1
       IF(J.NE.0) GO TO 100
       M=2
       CALL ANABEL(K,KD,JS,JT,JG,JQ,JL,JRA,JRB)
       DO 50 I=1,N
       P(I,KD,1)=1.D0
50     P(I,KD,2)=1.D0
       GO TO 600
C
C
100    CONTINUE
C
C
200    KQ=JQ(J)
       IQ=KQ
       IF(IQ.EQ.0) IQ=KD
       L=JL(J)
       JA=JRA(J)
       JB=JRB(J)
240    DO 250 JR=JA,JB
       A=0.D0
       B=0.D0
C      NOTE IQ,L ARE SPECIAL INDICES
       JRC=JR
       IF(JR.EQ.0)JRC=KD
       DO 260 I=1,N
       A=A+X(I,L)*P(I,IQ,M)*P(I,JRC,M)*W(I,M)**2
       B=B+(P(I,JRC,M)*W(I,M))**2
260    CONTINUE
       ALPHA(J,JRC,M)=A/B
250    CONTINUE
C
C
C::::COMPUTE POLYN. VALUES P(I,J,M)
300    DO 310 I=1,N
       A=0.D0
       DO 320 JR=JA,JB
       JRC=JR
       IF(JR.EQ.0) JRC=KD
       A=A+ALPHA(J,JRC,M)*P(I,JRC,M)
320    CONTINUE
310    P(I,J,M)=X(I,L)*P(I,IQ,M)-A
C      TEST FORMATS....
991    FORMAT('  XXX1')
992    FORMAT(' XXX2')
993    FORMAT(' XXX3')
994    FORMAT(' XXX4')
995    FORMAT(' XXX5')
C
C
C::::IF W(I,1).NE.W(I,2)REPEAT 240 DOWN FOR M=2
       IF(IW.NE.0) GO TO 600
500    DO 510 I=1,N
510    P(I,J,2)=P(I,J,1)
       DO 511 JR=JA,JB
       JC=JR
       IF(JR.EQ.0)JC=KD
511    ALPHA(J,JC,2)=ALPHA(J,JC,1)
       M=2
       IF(NPR.NE.0)WRITE(3,520)
520    FORMAT(' WEIGHTS EQUAL P(I,J,1)=P(I,J,2).')
C
C
C::::PRINT
600    JJ=J
       IF(J.EQ.0) JJ=KD
       IF(NPR.GE.1)WRITE(3,610) J,M,JG(JJ),JS(JJ),JT(JJ),JQ(JJ),JL(JJ),
      1JRA(JJ),JRB(JJ)
610    FORMAT(I4,'TH ORDER POLS. IN X(',I1,') COMPLETE.G,S,T,Q,L,
      1JRA,JRB=',7I5)
       IF(NPR.EQ.0)GO TO 700
       DO 640 JR=JA,JB
       JJ=JR
       IF(JR.EQ.0) JJ=KD
640    WRITE(3,620) J,JR,ALPHA (J,JJ,1),ALPHA(J,JJ,2)
620    FORMAT(' ALPHA(',I2,1X,I2,' M)= ',2D15.8)
```

```
      IF(NPR.LT.2) GO TO 700
      JJ=J
      IF(J.EQ.0) JJ=K0
      WRITE(3,630) (I,J,M,P(I,JJ,M),I=1,N)
630   FORMAT(' P ',4(' (',I2,1X,I2,1X,I2,')=',D15.8))
C
C
C:::::HAVE BOTH VARIABLES BEEN TREATED
700   IF(M.EQ.2) GO TO 800
900   M=2
      GO TO 240
C
C
800   RETURN
      END
```

PHI

```
C     ELT=L PHI                        MOD=1
      SUBROUTINE PHI(N,NS,Y,FN,W,PHY,WORST,IWORST)
C     PHY IS SUM OF RESIDUALS SQUARED TIMES WEIGHTING FN W
C     SUMMED OVER N DATA POINTS + OVERNS SECTIONS OF FN.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION Y(N,NS),FN(N,NS),W(N,NS)
      NPR=0
      WORST=0.D0
      PHY=0.D0
      DO 21 I=1,N
      DELTA=0.D0
      DO 20 L=1,NS
      DELTA=DELTA+(Y(I,L)-FN(I,L))**2*W(I,L)**2
20    CONTINUE
      IF(WORST.GT.DELTA)GO TO 21
      WORST=DELTA
      IWORST=I
21    PHY=PHY+DELTA
      IF(NPR.EQ.1)WRITE(3,624) PHY,WORST,IWORST
624   FORMAT('  624 PHY,WORST,IWORST= ',2D15.8,I6)
      RETURN
      END
```

PRECES

```
C     ELT=H     PRECES MOD=1
      SUBROUTINE PRECES(R1,TJ1,TJ2,R2)
C     LUNISOLAR PRECESSION CORRECTION ROUTINE (RECTANGULAR).
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION S(3,3),R1(3),R2(3)
      PI=3.14159265358979300
      PIO=PI/180.D0
      TO=(TJ1-2415020.31300)/36524.219879
      T=(TJ2-TJ1)/36524.219879
      IF(TJ2.GE.TJ1)GO TO 1
      TO=(TJ2-2415020.31300)/36524.219879
      T=(TJ1-TJ2)/36524.219879
1     CONTINUE
      ZETA=(0.64006944+3.8777778D-4*TO)*T+8.3888888889D-5*T*T+5.D-6*T**3
      ZED=ZETA+2.197222D-4*T*T
      AJAY=(0.55685611-2.3694D-4*TO)*T-1.18333D-4*T*T-1.1666667D-5*T**3
      ZETA=ZETA*PIO
      ZED=ZED*PIO
      AJAY=AJAY*PIO
      IF(TJ2.GE.TJ1)GO TO 2
      ZETA=-ZED
      ZED=-ZETA
      AJAY=-AJAY
2     CONTINUE
C     ****************************************************************
      CZ=DCOS(ZED)
      SZ=DSIN(ZED)
      CZO=DCOS(ZETA)
      SZO=DSIN(ZETA)
      CT=DCOS(AJAY)
      ST=DSIN(AJAY)
C
      S(1,1)=CZO*CT*CZ-SZO*SZ
      S(1,2)=-SZO*CT*CZ-CZO*SZ
      S(1,3)=-ST*CZ
      S(2,1)=CZO*CT*SZ+SZO*CZ
      S(2,2)=-SZO*CT*SZ+CZO*CZ
      S(2,3)=-ST*SZ
      S(3,1)=CZO*ST
      S(3,2)=-SZO*ST
      S(3,3)=CT
C
      CALL MXV(R2,S,R1)
      RETURN
      END
```

PTRX

```
C     ELT=G  PTRX MOD=1
      SUBROUTINE PTRX(RA,DEC,R)
C     POLAR TO RECTANGULAR TRANSFORM.
      DOUBLE PRECISION R(3),RA,DEC
      R(1)=DCOS(RA)*DCOS(DEC)
      R(2)=DSIN(RA)*DCOS(DEC)
      R(3)=DSIN(DEC)
      RETURN
      END
```

REFRAK

```
C     ELT=N REFRAK MOD=1
      SUBROUTINE REFRAK(HA,DEC,BAR,TEMP,HAO,DECO,PHI)
C     ATMOSPHERIC REFRACTION CORRECTION ROUTINE.
      IMPLICIT REAL*8 (A-H,O-Z)
      PI=3.14159265358979300
      PIO=PI/180.D0
      CALL ZANGLE(HA,DEC,ZA,PHI)
      UM1=1.0533667D-4*BAR/(273.D0+TEMP)
      HO=0.0012541D0
      TZ=DTAN(ZA)
      R=UM1*(1.D0-HO)*TZ-UM1*(HO-0.5D0*UM1)*TZ**3
C     ****************************************************************
      HDIF=-R*DCOS(PHI)*DSIN(HA)/(DCOS(DEC)*DSIN(ZA))
      DDIF=R/DSIN(ZA)*(DSIN(PHI)*DCOS(DEC)-DCOS(PHI)*DSIN(DEC)*DCOS(HA))
      HAO=HDIF+HA
      DECO=DEC+DDIF
      ZA=ZA-R
      RETURN
      END
```

RDATIN

```
C     ELT=E . RDATIN(TELSCOPE DATA INPUT ROUTINE) MOD # 6.
      SUBROUTINE RDATIN(N,M,K,NS,NOPT,X,Y)
C     PROGRAM # 7
C     RDATIN READS NO. OF OBSVNS (NOBS) & XY DATA FROM FILE 13 (SEE TA)
C     CHECKS NOBS VS N & FILLS ARRAYS X,Y.
C
C     NOPT(1)=1,2,3 PRINT OPTIONS.
C
C     NOPT(2)=0 REPLACE N=NO OF PTS BY NOBS READ IN.
C            =1      USE N AS NO. OF PTS IF N<NOBS.
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION NOPT(4),X(N,M),Y(N,NS)
      NPR=NOPT(1)
C     READ FILE 13 FOR NOBS
      READ(13) NOBS
C
      IF(NOPT(2).NE.0) GO TO 20
      N=NOBS
      IOBS=NOBS
      WRITE(3,11) NOBS
11    FORMAT(' 999N=NO. OF DATA PTS IS RESET TO SUIT NO. OF DATASET',
     1' ENTRIES=',I4)
      GO TO 30
20    IF(N.GT.NOBS) GO TO 40
      IOBS=N
      WRITE(3,12) N,NOBS
12    FORMAT(' NO. OF DATA PTS=',I6,' DATA SET OF',I6,' ENTRIES INPUT')
      GO TO 30
40    WRITE(3,13)
13    FORMAT(' 999 OPTION CONFLICT NOBS<N. NOBS IS ASSUMED.')
      IOBS=NOBS
      N=NOBS
C
C
C     READ FILE 13 FOR X,Y DATA.
30    DO 100 J=1,IOBS
      READ(13)X(J,1),X(J,2),Y(J,1),Y(J,2)
110   FORMAT(' XY FILE ENTRY# ',I4,5X,4D20.10)
      CALL SHIFT2(X(J,2))
      CALL SHIFT2(Y(J,2))
      IF(NPR.GE.2)WRITE(3,110)J,X(J,1),X(J,2),Y(J,1),Y(J,2)
100   CONTINUE
C
C
C     FIND PHI RMS ETC..
      DMAX=0.D0
      PHI=0.D0
      PI=3.14152653589793D0
      ASEC=PI/(180.D0*3600.D0)
      DO 200 J=1,IOBS
      DELTA=(Y(J,1)-X(J,1))**2+(Y(J,2)-X(J,2))**2*DCOS(X(J,1))**2
      PHI=PHI+DELTA
      IF(DMAX.GT.DELTA) GO TO 200
      DMAX=DELTA
      JMAX=J
200   CONTINUE
      RMS=DSQRT(PHI/IOBS)/ASEC
      NUU=2*N
      SIGSQ=PHI/NUU
      DMAX=DSQRT(DMAX)/ASEC
      WRITE(3,210)IOBS,PHI,RMS,DMAX,JMAX,SIGSQ,NUU
210   FORMAT(' PHI FOR',I5,' OBS= ',D20.10,' RMS ERR(ARCSEC)=',F10.2,
     1' WORST-CASE ERR & OBS NO.= ',F10.2,4X,I5,
     2/'      SIGSQ(NU)= ',D16.6,I4)
C
      RETURN
      END
```

RTPX

```
C     ELT=K RTPX MOD=2
      SUBROUTINE RTPX(RA,DEC,R)
C     RECTANGULAR TO POLAR TRANSFORM.
      DOUBLE PRECISION R(3),RA,Q,DEC
      Q=DSQRT(R(1)**2+R(2)**2)
      RA=DATAN2(R(2)/Q,R(1)/Q)
      DEC=DATAN(R(3)/Q)
      CALL SHIFT(RA)
      RETURN
      END
```

SHIFT

```
C     ELT=L SHIFTS ARG TO INTERVAL 0-2PI .MOD=1
      SUBROUTINE SHIFT(A)
      IMPLICIT REAL*8(A-H,O-Z)
      PI=3.14159265358900
      IF(A.LT.0.D0) GO TO 1
      N=A*0.5D0/PI
      A=A-N*2.D0*PI
      RETURN
1      N=-A*0.5D0/PI
      A=A+(N+1)*2.D0*PI
      RETURN
      END
```

SHIFT2

```
C     ELT=M SHIFT2,SHIFTS ARG INTO INTERVAL -PI,PI .MOD=1
      SUBROUTINE SHIFT2(A)
      IMPLICIT REAL*8 (A-H,O-Z)
      PI=3.14159265358900
      IF(A.LT.-PI) GO TO 1
      IF(A.GT.PI) GO TO 2
      RETURN
2     N=(A+PI)*0.5D0/PI
      A=A-N*2.D0*PI
1     N=(PI-A)*0.5D0/PI
      A=A+N*2.D0*PI
      RETURN
      END
```

SORT

```
C     ELT=SORT   MODIFICATION NO.  3
      SUBROUTINE SORT(A,B,I,N)
C
C    SORTS N REAL*8 VARS A INTO B IN ASCENDING ORDER AND RECORDS ORDER IN
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(N),B(N),I(N)
C
C
C
      DO 10 J=1,N
      I(J)=J
10    B(J)=A(J)
C                          SORT
      NM=N-1
      DO 20 K=1,NM
      NK=N-K
      DO 20 J=1,NK
      IF(B(J).LE.B(J+1)) GO TO 20
      SAVE=B(J+1)
      B(J+1)=B(J)
      B(J)=SAVE
      ISAVE=I(J+1)
      I(J+1)=I(J)
      I(J)=ISAVE
20    CONTINUE
C
      RETURN
      END
```

SPIRAL

```
C     ELT,PROG = SPIRAL                    MODIFICATION NO. = 6
      SUBROUTINE SPIRAL(N,M,K,NS,NOPT,X,Y,PARM,FN,DF,W,A,ATEMP,G,GTEMP,T,
     1,SCALE,MASK)
C      PROGRAM  4
C      JONES SPIRAL ALGORITHM FOR PARM ESTIMATION.
C
C     ................<.<.<...................................OPTIONS
C     IF NOPT(1)=0 READS PARM,MASK FROM  K  CARDS.
C
C     NOPT(2) GE 1 ENABLES PRINT..
C     ................<.<.<...........................................
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION    DATA(2),NOPT(4)
      DIMENSION    X(N,M),Y(N,NS),FN(N,NS),W(N,NS),DF(N,K,NS)
      DIMENSION    PARM(K),T(K),G(K),GTEMP(K),SCALE(K),MASK(K)
      DIMENSION    A(K,K),ATEMP(K,K)
C
C     INITIALIZE.
      PI=3.141592653589793
      PIO=PI/180.D0
      SEC=PIO/3600.D0
      ITER=1
      IS=1
      ALAMDA=1.D-2
      AMU1=0.1D0
      AMU2=0.9D0
      EPS=1.D-5
      TAU=1.D-15
      MSUM=K
      NPR=NOPT(2)
      KOUNT1=0
      KOUNT2=0
C
C     ..................................................... DATA INPUT..
C
      IF(NOPT(1).EQ.0) GO TO 6
6     CONTINUE
      WRITE(3,7)
7     FORMAT(' ::::: ENTER PARMS +MASKS PLEASE')
      DO 10 J=1,K
      CALL READ(NVAR,DATA,NERR)
      IF(NVAR.NE.2.OR.NERR.EQ.1) GO TO 4010
      PARM(J)= DATA(1)
      MASK(J)= DATA(2)
10    CONTINUE
C
C     ................................................................
C
C     CALC. CURRENT.PHI.
100   CALL FCTN(N,M,K,NS,X,PARM,FN,KOUNT1)
      CALL WT(N,M,NS,X,W)
      CALL PHI(N,NS,Y,FN,W,PHIO,WORST,IWORST)
C
C     PRINT....
      WRITE(3,110) PHIO
110   FORMAT(///' JONES SPIRAL ALGORITHM , INITIAL PHI=',D15.8,' INITIAL
     1 PARMS (MASKS) ARE..')
      DO 130 J=1,K
130   WRITE(3,120) PARM(J),MASK(J)
120   FORMAT(2X,D15.8,'(',I4,'  )')
C
200   CONTINUE
      CALL DERV(N,M,K,NS,X,PARM,DF,KOUNT2)
      SCALE(1)=0.D0
      CALL MATEQN(N,M,K,NS,Y,FN,W,DF,A,G,SCALE)
C     STORE GRADIENT IN GTEMP.
      DO 210 J=1,K
```

```
210   GTEMP(J)=G(J)*SCALE(J)
C
C     SOLVE EQN.  A*T=G.
      CALL DGELG(G,A,K,1,1.E-15,IER)
      IF(IER.NE.0) GO TO 4000
C     IF MATRIX A SINGULAR DO MARQUARDT'S TRICK
240   CONTINUE
C
C     DESCALE T , GET T,G MAGNITUDES.
      GMAG=0.D0
      TMAG=0.D0
      DO 220 J=1,K
      T(J)=G(J)/SCALE(J)
      TMAG=TMAG+T(J)**2
220   GMAG=GMAG+GTEMP(J)**2
      TMAG=DSQRT(TMAG)
C     WRITE(3,5000)
      GMAG=DSQRT(GMAG)
C     CALC. GAMMA ,COS,SIN.
      COSGAM=0.D0
      DO 230 J=1,K
      G(J)=GTEMP(J)
230   COSGAM=COSGAM+T(J)*G(J)
      COSGAM=COSGAM/(TMAG*GMAG)
      SINGAM=DSQRT(1.D0-COSGAM**2)
      GAMMA=DACOS(COSGAM)
      IF(COSGAM)4020,280,280
C     NOW HAVE TAYLOR VECTOR T , GRADIENT G , AND SIN , COS GAMMA.
C
C
C     CALC. TAYLOR POINT.  USE GTEMP AS TEMPORARY PARM VECTOR.
280   DO 290 J=1,K
290   GTEMP(J)=PARM(J)+T(J)
      CALL FCTN(N,M,K,NS,X,GTEMP,FN,KOUNT1)
      CALL PHI(N,NS,Y,FN,W,PHY,WORST,IWORST)
C
C     TEST FO R EPSILON CONVERGENCE.
C
300   ITEST=0
      DO 310 J=1,K
      TEST=DABS(T(J))/(DABS(PARM(J))+TAU)
      IF(TEST.GT.EPS) ITEST=1
310   CONTINUE
      IF(ITEST.EQ.0) GO TO 500
C     TEST PHI OF TAYLOR POINT.
      IF(PHY.LT.PHI0) GO TO 2000
C     IS T ORIGINAL TAYLOR POINT.?
600   IF(IS.EQ.1) GO TO 800
C     INTERPOLATE ALONG T ?
700   IF(PHY.GE.PHYT) GO TO 800
C
C     TAYLOR DIRN. INTERP.
C
C**********TEMPORARY CODING.
      WRITE(3,2210)IS
2210  FORMAT(' TAYLOR INTERP. REQUIRED SPIRAL NO.',I3)
      GO TO 800
C*********************
C
C     GENERATE SPIRAL.
C
800   AMU=AMU1
      IP=1
C     SAVE OLD T AS A( ,K).OLD PHI AS PHYT.
      DO 810 J=1,K
810   A(J,K)=T(J)
      PHYT=PHY
C     RESCALE GRADIENT VECTOR TO SAME MAGNITUDE AS CURRENT TAYLOR VECTOR
      GMAG=0.D0
      TMAG=0.D0
      DO 820 J=1,K
      GMAG=GMAG+G(J)**2
820   TMAG=TMAG+T(J)**2
      GMAG=DSQRT(GMAG)
      TMAG=DSQRT(TMAG)
      DO 830 J=1,K
830   G(J)=G(J)*TMAG/GMAG
C     CALC. SPIRAL POINT.
900   BETA=GAMMA/2.D0
      THETA=AMU*SINGAM/(1.D0-AMU+AMU*COSGAM)
      THETA=DATAN(THETA)
      RONXI=THETA/GAMMA
      RONXI=1.D0+THETA*DCOS(BETA)*(RONXI-1.D0)-RONXI**2
      RONXI=RONXI*DSIN(THETA)/(AMU*SINGAM)
      DEE=RONXI*AMU
      TEE=RONXI*(1.D0-AMU)
C     FORM TEMPORARY PARM VECTOR GTEMP=PARM+S
      DO 910 J=1,K
910   GTEMP(J)=PARM(J)+DEE*G(J)+TEE*T (J)
      CALL FCTN(N,M,K,NS,X,GTEMP,FN,KOUNT1)
      CALL PHI(N,NS,Y,FN,W,PHY,WORST,IWORST)
C     SAVE LAST TWO SPIRAL POINTS + PHI VALUES.
950   IF(IP.EQ.1) GO TO 980
      IF(IP.EQ.2) GO TO 970
      SAVMU1=SAVMU2
      PHIMU1=PHIMU2
970       SAVMU2=SAVMU3
      PHIMU2=PHIMU3
980   SAVMU3=AMU
      PHIMU3=PHY
C     TEST CURRENT PHI
990   IF(PHY.LT.PHI0) GO TO 2000
C
C     CHECK IF INTERP. ALONG SPIRAL REQUIRED.
C
1000  IF(IP.LT.3) GO TO 1100
C     TEST=PHIMU1+(PHIMU3-PHIMU1)*(SAVMU2-SAVMU1)/(SAVMU3-SAVMU1)
C     IF(PHIMU2.GE.TEST) GO TO 1100
      IF(PHIMU2.GE.PHIMU1.OR.PHIMU2.GE.PHIMU3) GO TO 1100
C*****INTERPOLATE ALONG SPIRAL
C
      Q1=(SAVMU2-SAVMU3)*PHIMU1
      Q2=(SAVMU3-SAVMU1)*PHIMU2
      Q3=(SAVMU1-SAVMU2)*PHIMU3
      P1=(SAVMU2+SAVMU3)
      P2=(SAVMU3+SAVMU1)
      P3=(SAVMU1+SAVMU2)
C     CALC TURNING PNT OF QUADRIC
      DMU=0.5D0*(Q1*P1+Q2*P2+Q3*P3)/(Q1+Q2+Q3)
C     GET INTERPOLATED POINT
      BETA=GAMMA/2.D0
      THETA=DMU*SINGAM/(1.D0-DMU+DMU*COSGAM)
      THETA=DATAN(THETA)
      RONXI=THETA/GAMMA
      RONXI=1.D0+THETA*DCOS(BETA)*(RONXI-1.D0)-RONXI**2
      RONXI=RONXI*DSIN(THETA)/(DMU*SINGAM)
      DEE=RONXI*DMU
      TEE=RONXI*(1.D0-DMU)
      DO 1210 J=1,K
1210  GTEMP(J)=PARM(J)+DEE*G(J)+TEE*T(J)
      CALL FCTN(N,M,K,NS,X,GTEMP,FN,KOUNT1)
      CALL PHI(N,NS,Y,FN,W,PHY,WORST,IWORST)
       WRITE(3,1201)DMU,PHY
C
C     TEST CURRENT PHI
      IF(PHY.LT.PHI0)  GO TO 2000
C
1201  FORMAT( ' INTERPOLATION  ALONG SPIRAL. MU,PHY= ',2D15.8)
C*********************
C
C     CHECK IF ALL POINTS ON SPIRAL ARE DONE.
C
1100  IF(AMU.GE.AMU2) GO TO 1400
C     IF NOT CALC. NEW MU.
1300  IP=IP+1
      AMU=2.D0*AMU/(1.D0+AMU)
      GO TO 900
C
C     CHECK IF ALL SPIRALS DONE.
C
1400  IF(IS.EQ.4) GO TO 1600
C     IF NOT HALVE TAYLOR VECTOR + RETURN.
1500  DO 1510 J=1,K
1510  T(J)=T(J)/2.D0
      IS=IS+1
      GO TO 280
C
C     FINALLY SEARCH STEEPEST DESCENT DIRN.
C
1600  DO 1610 J=1,K
1610  GTEMP(J)=PARM(J)+G(J)
      CALL FCTN(N,M,K,NS,X,GTEMP,FN,KOUNT1)
      CALL PHI(N,NS,Y,FN,W,PHY,WORST,IWORST)
C     TEST PHI.
      IF(PHY.LT.PHI0) GO TO 2000
1700  ITEST=0
      DO 1710 J=1,K
      TEST=DABS(G(J))/(DABS(PARM(J))+TAU)
      IF(TEST.GE.EPS) ITEST =1
1710  CONTINUE
      IF(ITEST.EQ.0) GO TO 1800
1900  DO 1910 J=1,K
1910  G(J)=G(J)/2.D0
      GO TO 1600
C
C     ****************************************************************
C
C     STEEPEST DESCENT CONVERGENCE.
1800  WRITE(3,1810)
1810  FORMAT(///' SPIRAL STEEPEST DESCENT CONVERGENCE')
      GO TO 2100
C
C     EPSILON CONVERGENCE.
500   WRITE(3,510)
510   FORMAT(///' SPIRAL...EPSILON CONVERGENCE.')
C
C     EXIT SECTION FINALIZE + PRINT.
2100  CONTINUE
      WRITE(3,2110) ITER,PHY,IS,IP,KOUNT1,KOUNT2
2110 _FORMAT(/' ITER NO.',I5,'    PHI=',D15.8,'  IS,IP=',2I4,' KOUNT1,2=
     1',2I5,' FINAL PARM VECTOR FOLLOWS')
C
      CALL SHIFT(GTEMP(3))
      DO 2130 J=1,K
      DPARM=GTEMP(J)/PI0
      SPARM=GTEMP(J)/SEC
      WRITE(3,2120)J,GTEMP(J),DPARM,SPARM
2130  CONTINUE
2120  FORMAT(' PARM NO.',I4,D15.8,' RAD ',F12.5,' DEG ',F11.4,' SEC')
      RETURN
C      ***************************************************************
C     MOVE TO NEW BASE POINT.
2000  DO 2010 J=1,K
2010  PARM(J)=GTEMP(J)
C     PRINT.
      WRITE(3,2020) ITER,IS,IP,KOUNT1,KOUNT2,SINGAM,PHY
2020 _FORMAT(' END ITER NO.',I5,'  IS,IP,KOUNT1,2=',2I4,2I5,' SIN GAM=',
     1D12.5,'  PHI=',D15.8,' PARMS FOLLOW')
      WRITE(3,2030) PARM
2030  FORMAT(3X,D15.8)
      IS=1
      ITER=ITER+1
      PHI0=PHY
      GO TO 200
C
C     ****************************************************************
C
C     DGELG FAIL
4000  WRITE(3,4001) IER
4001  FORMAT(///' MATRIX EQN SOLN. ROUTINE FAIL,IER(DGELG)=',I4)
      WRITE(3,4002)ALAMDA
4002  FORMAT(' *******TRY AGAIN WITH LAMDA= ',D15.8)
      SCALE(1)=0.D0
      CALL MATEQN(N,M,K,NS,Y,FN,W,DF,A,G,SCALE)
      DO 4005 J=1,K
      GTEMP(J)=G(J)*SCALE(J)
      A(J,J)=A(J,J)+ALAMDA
4005  CONTINUE
      CALL DGELG(G,A,K,1,1.E-15,IER)
      IF(IER.EQ.0)GO TO 240
      WRITE(3,4001)IER
      RETURN
C
C
C     READ FAIL
4010  WRITE(3,4011) NVAR
4011  FORMAT(///' READ ROUTINE FAIL',I5,'  VARIABLES WERE READ.')
      RETURN
C
C
C     GAMMA FAIL.
4020  WRITE(3,4021) GAMMA,COSGAM,SINGAM
4021  FORMAT(///' ANGLE GAMMA GT PI/2, ANGLE,COS,SIN= ' ,3D15.8)
      RETURN
      END
```

STIME

```
C     ELT=X     STIME MOD =3
      SUBROUTINE STIME(TJ,ELONG,ST,EQUIN,MEAN)
C
C     COMPUTES LOCAL SID. TIME (MEAN IF MEAN=0,APPARENT OTHERWISE)
C     IN RAD. FOR ELONG = EAST LONG. (RAD), AT UT. TJ=JULIAN DATE.
C     ALSO GIVES EQUIN=EQUATION OF EQUINOXES IN RADIAN.
C
      IMPLICIT REAL*8(A-H,O-Z)
      PI=3.141592653589793D0
      PI0=PI/180.D0
C
      IT=TJ
      TU=IT
      TREF=TU+0.5D0
      TU=(TREF-2415020.D0)/36525.D0
      STG0=23925.836D0+8640184.542D0*TU+0.0929D0*TU*TU
      STG0=STG0*PI/43200.D0
      DTHETA=1.0027379092650D0+0.589D-10*TU
      STG=STG0+(TJ-TREF)*DTHETA*PI*2.D0
      ST=STG+ELONG
C
      IF(MEAN.EQ.0) GO TO 10
C
C*****COMPUTE EQUATION OF EQUINOXES.
C*****PARAMETERS W.R.T. MEAN EQUINOX OF DATE.
      T=(TJ-2415020.D0)/36525.D0
      ON=259.183275D0-1934.142008*T+0.20778D-2*T*T+0.22D-5*T**3
      OS=279.696678D0+36000.768925*T+0.3025D-3*T**2
      OM=270.434164D0+481267.8831417*T-0.11333D-2*T**2+0.1889D-5*T**3
      G=(6189.03*T+1.63*T*T+0.012*T*T*T)/3600.D0
      G=281.220833+G
      GD=(-37.17*T*T-0.045*T*T*T)/3600.D0
      GD=GD+334.329556+4069.03403333*T
      ON=ON*PI0
      OS=OS*PI0
      OM=OM*PI0
      G=G*PI0
      GD=GD*PI0
      L=OM-GD
      LD=OS-G
      F=OM-ON
      D=OM-OS
C      ***************************************************************
C*****SERIES FOR DPHI. INCLUDES ALL TERMS WITH COEFFS > 0.01 ARCSEC.
C
      DPHI=-(17.2327+0.01737*T)*DSIN(ON)+0.2088*DSIN(2.D0*ON)-1.2729*DSI
     CN(F+F-D-D+ON+ON)+0.1261*DSIN(LD)-0.0497*DSIN(LD+F+F-D-D+ON+ON)+0.0
     C214*DSIN(F+F-D-D-LD+ON+ON)+0.0124*DSIN(F+F-D-D+ON)
C
      DPHI=DPHI-0.2037*DSIN(F+F+ON+ON)+0.0675*DSIN(L)-0.0342*DSIN(F+F+ON
     C)-0.0261*DSIN(L+F+F+ON+ON)-0.0149*DSIN(L-D-D)+0.0114*DSIN(F+F-L+ON
     C+ON)
C
      EPS0=23.452294D0-0.0130125D0*T-1.64D-6*T**2+5.03D-7*T**3
      DPHI=DPHI*PI0/3600.D0
      EPS0=EPS0*PI0
      EQUIN=DPHI*DCOS(EPS0)
      ST=ST+EQUIN
10    N=ST*0.5D0/PI
      ST=ST-N*2.D0*PI
      RETURN
      END
```

TDATIN

```
C     ELT=J . TDATIN(TELSCOPE DATA INPUT ROUTINE) MOD = 5.
      SUBROUTINE TDATIN(N,K,KD,NOPT,X,Y)
C     PROGRAM # 4
C     TDATIN READS NO. OF OBSVNS (NOBS) &  XY DATA FROM FILE 13 (SEE TA)
C     CHECKS NOBS VS N & FILLS ARRAYS X,Y.
C
C     NOPT(1)=1,2,3 PRINT OPTIONS,=9 PLOT DEC/HA.
C
C     NOPT(2)=0 REPLACE N=NO OF PTS BY NOBS READ IN.
C            =1      USE N AS NO. OF PTS IF N<NOBS.
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION NOPT(4),X(N,2),Y(N,2)
C     CAUTION TEMPORARILY DIMENSIONED VARIABLES FOLLOW..
      REAL XH(160),YH(160)
C
      PI=3.141592653589793D0
      NPR=NOPT(1)
C     READ FILE 13 FOR NOBS
      READ(13) NOBS
C
      IF(NOPT(2).NE.0) GO TO 20
      N=NOBS
      IOBS=NOBS
      WRITE(3,11) NOBS
11   _FORMAT(' @@@N=NO. OF DATA PTS IS RESET TO SUIT NO. OF DATASET',
     1' ENTRIES=',I6)
      GO TO 30
20    IF(N.GT.NOBS) GO TO 40
      IOBS=N
      WRITE(3,12) N,NOBS
12    FORMAT(' NO. OF DATA PTS=',I6,' DATA SET OF',I6,' ENTRIES INPUT')
      GO TO 30
40    WRITE(3,13)
13    FORMAT(' @@@ OPTION CONFLICT NOBS<N. NOBS IS ASSUMED.')
      IOBS=NOBS
      N=NOBS
C
C
C     READ FILE 13 FOR X,Y DATA.
30    DO 100 J=1,IOBS
      READ(13)X(J,1),X(J,2),Y(J,1),Y(J,2)
      IF(NPR.GE.2)WRITE(3,110)J,X(J,1),X(J,2),Y(J,1),Y(J,2)
110   FORMAT(' XY FILE ENTRY# ',I4,5X,4D20.10)
100   CONTINUE
C
C
C     FIND PHI RMS ETC..
      DMAX=0.D0
      PHI=0.D0
      PI=3.141526535897930D0
      ASEC=PI/(180.D0*3600.D0)
      DO 200 J=1,IOBS
      XXJ=X(J,2)
      CALL SHIFT2(XXJ)
      XH(J)=XXJ*12.D0/PI
      YH(J)=X(J,1)*180.D0/PI
```

```
      DELTA=(Y(J,1)-X(J,1))**2+(Y(J,2)-X(J,2))**2*DCOS(X(J,1))**2
      PHI=PHI+DELTA
      IF(DMAX.GT.DELTA) GO TO 200
      DMAX=DELTA
      JMAX=J
200   CONTINUE
      RMS=DSQRT(PHI/IOBS)/ASEC
      DMAX=DSQRT(DMAX)/ASEC
      WRITE(3,210)IOBS,PHI,RMS,DMAX,JMAX
210   FORMAT(' PHI FOR',I5,' OBS= ',D20.10,' RMS ERR(ARCSEC)=',F10.2,
     1' WORST-CASE ERR & OBS NO.= ',F10.2,4X,I5)
C
C      ***********************************************
C     PLOT
      IF(NPR.EQ.9)CALL APLOT(IOBS,XH,'HA',YH,'DEC',0)
C     ***************************************************
      RETURN
      END
```

WT

```
      SUBROUTINE WT(N,M,NS,X,W)
C     CALCULATES WEIGHTS FOR TELESCOPE POINTING ERROR FN.
C     X(J,1)=DEC,X(J,2)=HA...RETURNS W(J,1)=1.0,W(J,2)=COSINE DEC(J)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION  X(N,2),W(N,2)
      NPR=0
      DO 20 I=1,N
      W(I,1)=1.D0
      W(I,2)=DCOS(X(I,1))
      IF(NPR.EQ.1)WRITE(3,623) X(I,1),X(I,2),W(I,1),W(I,2)
623   FORMAT('  623  WT ', 4D15.8)
20    CONTINUE
      RETURN
      END
```

ZANGLE

```
C     ELT=0  ZANGLE MOD=1
      SUBROUTINE ZANGLE(HA,DEC,ZA,PHI)
C     CALCULATES ZENITH ANGLE GIVEN TOPOCENTRIC H.A.,DEC.
      IMPLICIT REAL*8 (A-H,O-Z)
      ZA=DCOS(PHI)*DCOS(DEC)*DCOS(HA)+DSIN(PHI)*DSIN(DEC)
      ZA=DACOS(ZA)
      RETURN
      END
```

APPENDIX E; reprinted from Proc. I.R.E.E. Aust. March 1973.

Summary: A method of converting a standard frequency (referred to either the mean solar or atomic second) to the equivalent frequency referred to the mean sidereal second is described. A voltage-controlled oscillator on the sidereal frequency is mixed with the standard and the difference frequency phase-locked to a signal derived by division of the standard. A long-term conversion accuracy of better than 50 ms per year is possible.

A Phase-Locked Solar to Sidereal Frequency Converter

G. R. HOVEY*

The production of sidereal time for the use of astronomical observatories and deep space tracking stations usually involves the generation of frequencies referred to the mean sidereal second from standard frequencies which are referred to either the atomic second or the mean solar second. Existing methods for converting a " solar " frequency to the same nominal " sidereal " frequency usually employ integral dividers or mix the standard with a separate free-running oscillator and do not have a conversion accuracy comparable to the long-term stability of the oscillator they convert.

Consider the difference frequency, f_3, obtained by mixing the standard (solar) oscillator of frequency f_1 with a voltage-controlled crystal oscillator (VCXO) which is on the required sidereal frequency, f_2.

$$f_3 = f_2 - f_1 \tag{1}$$

Assuming the solar frequency, f_1, is exact, the fractional error of f_2 is given by :

$$\left|\frac{\Delta f_2}{f_2}\right| = \left|\frac{\Delta(f_1+f_3)}{f_1+f_3}\right| = \left|\frac{\Delta f_3}{f_1+f_3}\right| \simeq \left|\frac{\Delta f_3}{f_1}\right| = \left|\frac{\Delta f_3}{f_3}\right| . \frac{f_3}{f_1} \tag{2}$$

Thus the fractional error of the required sidereal frequency is less than that of the beat frequency, f_3, by the factor f_3/f_1

With the adoption in January 1972[1] of the atomic second as the mean rate used to generate solar time the ratio of sidereal frequency to atomic frequency is given by :

$$f_2/f_1 = 1.002\ 737\ 939\ 3 \tag{3}$$

while the value for the pre-1972 solar rate is[2]

$$1.002\ 737\ 909\ 3$$

Thus the factor f_3/f_1 is independent of the actual frequency f_1 and is

$$f_3/f_1 = (f_2 - f_1)/f_1 \simeq 2.74 \times 10^{-3} \tag{4}$$

*Department of Engineering Physics, The Australian National University, Canberra, A.C.T.

Manuscript received by The Institution September 29, 1972.

U.D.C. number 621.314.26.

Clearly, if we phase-lock the beat frequency, f_3, to a reference frequency, f'_3, obtained by division from the standard f_1, the eventual error in f_2 (due to the fact that we can only divide f_1 integrally) is less than if we were to obtain f_2 directly. Optimum conversion accuracy is obtained by deriving f'_3 from as high a frequency, f_1, as possible but employing a low frequency at which to mix the signals and therefore a low difference frequency. This is the basis of the system shown in fig. 1 in which f_1 is 5.0 MHz. The reference frequency, f'_3, is generated with a fractional error of 5.26×10^{-7} (5.33×10^{-7})† and thus from equation 2 above, the fractional error in f_2 is 1.44×10^{-9} (1.47×10^{-9})†.

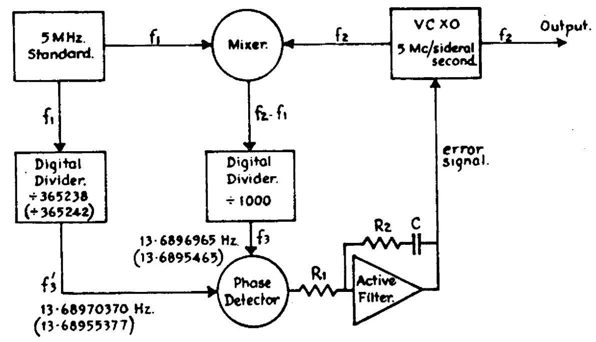


Figure 1.—Phase-locked solar to sidereal frequency converter. Frequencies shown for 5 MHz atomic frequency standard with values for pre-1972 solar rate in parentheses.

For the frequencies shown (fig. 1) the system has a conversion accuracy better than 1.5 parts in 10^9 or in terms of cumulative error approximately 45 ms per year. This is substantially better than the ageing rate of the standard, which is in our case a HP 105B oscillator (ageing rate 5 parts in 10^{10} per day, approximately 5 seconds per year).

In the prototype tested the phase detector is a TTL logic bistable and the input to the active filter is a 13.7 Hz squarewave with a mark-space ratio dependent on the phase difference between f_3 and f'_3. As well as defining the loop performance[3] this filter serves to integrate the squarewave. With the loop locked the control voltage to the VCXO has a 13.7 Hz component of approximately 140 mV peak to peak amplitude causing a frequency swing of 100 Hz (peak to peak) in f_2. This is not considered important since the projected use demands only long-term accuracy.

The loop damping with the filter time constants used is quite poor and an improvement in damping factor unfortunately necessitates an increase in the amplitude of the 13.7 Hz component in the output. This can be avoided by using a higher frequency for f_3, for example,

$$f_3 = 136.89 \text{ Hz},$$

in which case f_1 would be divided by 36524. The compromise is reduced long-term conversion accuracy, since now

$$\frac{|\Delta f_2|}{|f_2|} = 1.35 \times 10^{-8}$$

and this would represent a cumulative conversion error of approximately 400 ms per year.

Acknowledgments

Acknowledgment is due to Evan Muir (ex Mt. Stromlo Observatory) for numerous helpful ideas and most of the design work and to John Williams of Mt. Stromlo Observatory for design, construction of the prototype and measurements thereon.

References

1. *Trans. Int. Astronomical Union*, Vol. XIV(b), (Proceedings 1970).
2. Explanatory Supplement to the " Astronomical Ephemeris ", H.M.S.O. (1961).
3. Gardner, F. M., " Phaselock Techniques ", John Wiley (1966).

†Results for pre-1972 solar rate shown in parentheses.

APPENDIX F.
BIBLIOGRAPHY

A.A.T. (1970) Anglo-Australian Telescope Project Office, (private communication).

ABDEL-GAWAD, KHAIRY, M. (1969) Analysis for Redesign of 150" Stellar Telescope Serrurier Truss Structure; A.U.R.A. Engineering Technical Report No. 9.

ANSCOMBE, F.J. (1960) Rejection of Outliers; Technometrics, Vol. 2, No. 2, p. 123, May 1960.

ANSCOMBE, F.J. and TUKEY, J.W. (1963) The Examination and Analysis of Residuals; Technometrics, Vol. 5, No. 2, p. 141, May 1963.

AREND, S. (1951) Theorie de l'équatorial visuel et de l'équatorial photographique; Monographies No. 2, Observatoire Royal de Belgique, 1951.

ATKINSON, R. d'E. and SADLER, D.H. (1951) On the Use of Mean Sidereal Time; Monthly Notices of the Royal Astronomical Society, Vol. 111, No. 11, p. 619, 1951.

AUS, H.M. and KORN, G.A. (1969) Table Lookup/Interpolation Function Generation for Fixed-Point Digital Computations; I.E.E.E. Transactions on Computers, August 1969.

AYRE, J.K. (1967) Controlling an Astronomical Radio Telescope; Industrial Electronics, Vol. 5, p. 240, June 1967.

BAHNER, K. (1967) Teleskope; in Handbuck der Physik, Flugge, S. (editor), Vol. 29, Optische Instrumente; p.227, 1967.

BAIN, R.W. (1961) Preparation of Steam Tables with the aid of a Digital Computer; Journal of Mechanical Engineering, Vol. 3, No. 4, p.289, 1961.

BAKER, J.G. (1969) On Improving the Effectiveness of Large Telescopes; I.E.E.E. Transactions on Aerospace and Electronic Systems, Vol. AES-5, No. 2, p. 261, March 1969.

BARR, L.D. (1969) Mechanical Drive Considerations for the 150" Stellar Telescope; A.U.R.A. Engineering Technical Report No. 12.

BAUSTIAN, W.W. (1965), in I.A.U. Symposium No. 27, 1965.

BEALE, E.M.L. (1960) Confidence Regions in Nonlinear Estimation; Journal of the Royal Statistical Society, Series B, Vol. 22, p.41, 1960.

BERTIN, B. (1971) I.N.A.G. Report on the French 2 metre and 3.6 metre Telescopes; in E.S.O./C.E.R.N. conference proceedings, 1971.

BERZTISS, A.T. (1964) Least Squares Fitting of Polynomials to Irregularly Spaced Data; S.I.A.M. Revue, Vol. 6, p.203, 1964.

BIRKHOFF, G. and GARABEDIAN, H.L. (1960) Smooth Surface Interpolation; J. Math. Phys., Vol. 39, p.258, 1960.

BIRKHOFF, G. and DEBOOR, C.R. (1965) Piecewise Polynomial Interpolation and Approximation; in Garabedian 1965, p.164.

BOWEN, I.S. (1967) Astronomical Optics; in Annual Review of Astronomy and Astrophysics, Vol. 5, 1967. Goldberg, L. (editor).

BOX, G.E.P. and COUTIE, G.A. (1956) The Application of Digital Computers in the Exploration of Functional Relationships; Proc. I.E.E., part B, Supplement, Vol. 103, No. 1, p. 100, 1956.

BOX, G.E.P. and HUNTER, W.G. (1965) The Experimental Study of Physical Mechanisms; Technometrics, Vol. 7, No. 1, p. 23, 1965.

BROWN, R.R., DENNIS, J.B. and KINGSLEY, C. (1956) Design of Systems using Digital Computers; Wright Air Development Centre Technical Note 56-383.

CADWELL, J.H. and WILLIAMS, D.E. (1961) Some Orthogonal Methods of Curve and Surface Fitting; Computer Journal, Vol. 4, p. 260, 1961.

CLARKE, B.G. (1970) Information Processing Systems in Radio Astronomy and Astronomy; Annual Review of Astronomy and Astrophysics, Vol.8, 1970, Goldberg, L. (editor).

CLARKE, D. (1971) Telescope Drives and Guidance by Stepping Motors; Observatory, Vol.91, No. 985, p.215, 1971.

CLENSHAW, C.W. (1960) Curve Fitting with a Digital Computer; Computer Journal, Vol.2, p. 170, 1960.

CLENSHAW, C.W. and HAYES, J.G. (1965) Curve and Surface Fitting; J. Inst. Maths. Applics., Vol. 1, p.164, 1965.

COONS, S.A. (1967) Surfaces for Computer Aided Design of Space Forms; M.I.T. Report MAC-TR-41.

CRAWFORD, D.L., MEINEL, A.B. and STOCKTON, M.W. (editors) (1966) Symposium on the Support and Testing of Large Mirrors; K.P.N.O. and University of Arizona, Tucson, 1966.

DAVIDON, W.C. (1959) Variable Metric Method for Minimization; A.E.C. Research and Development Report, ANL-5990 (rev), 1959.

DEJAIFFE, R.J. (1970) Power Spectrum in Meridian Astronomy - Spectral Analysis of the Division Errors of Meridian Circles; Astronomy and Astrophysics, Vol. 7, p. 169, 1970.

DISNEY, M. (1973) Telescopes for Tomorrow; New Scientist, Vol. 58, No. 842, p. 147, April 19th 1973.

DRAPER, N.R. and SMITH, H. (1966) APPLIED REGRESSION ANALYSIS; John Wiley and sons Inc. (Applied Probability and Statistics Series).

Endeavour (1970) Automation in Optical Astronomy; (editorial) Endeavour, Vol. 24, No. 107, May, 1970.

E.S.O./C.E.R.N. (1971) Proceedings of E.S.O./C.E.R.N. Conference on Large Telescope Design; held Geneva, March 1971, West, R.M. (editor), published E.S.O./C.E.R.N. June 1971.

Explanatory Supplement to the Astronomical Ephemeris; H.M.S.O., 1961.

FERGUSON, J. (1964) Multivariable Curve Interpolation; J.A.C.M., Vol. 11, p. 221, 1964.

FINDLAY, J.W. (1971) Filled-aperture Antennas for Radio Astronomy; Annual Review of Astronomy and Astrophysics, Vol. 9, 1971.

FLETCHER, R. (1965) Function Minimization without Calculating Derivatives - A Review; Computer Journal, Vol. 8, p. 33, 1965.

FORSYTHE, G.E. (1957) Generation and Use of Orthogonal Polynomials for Data Fitting with a Digital Computer; J. Soc. Indust. Appl. Math., Vol. 5, No. 2, P. 74, 1957.

FOSTH, D.C. (1969) A Survey of Precision Pointing Systems; in N.A.S.A. SP233, Optical Telescope Technology Workshop, p. 73, 1969.

FRICKE, W. (1972) Fundamental Systems of Positions and Proper Motions; Annual Review of Astronomy and Astrophysics, Vol. 10, 1972.

GARABEDIAN, H.L. (editor) (1965) Approximation of Functions; Proceedings of Symposium at G.M.H. Laboratories Michigan 1964, Amsterdam, Elsevier.

GARDNER, F.M. (1966) PHASELOCK TECHNIQUES; John Wiley, 1966.

GASCOIGNE, S.C.B. (1970) Optical Telescopes; Journal of Scientific Instruments (J. Phys. E.), Vol. 3, No. 3, p. 165, 1970.

GOLUB, G.H. (1965) Numerical Methods for Solving Linear Least Squares Problems; Numer. Math., Vol. 7, p. 206, 1965.

GROENVELD, D.G.S. (1969) Considerations in the Design of Primary Worm Gear Drives for Astronomical Telescopes; Proceedings of the Astronomical Society of Australia, Vol. 1, p. 245, 1969.

GRUBBS, F.E. (1950) Sample Criteria for Testing Outlying Observations; Ann. Math. Stat., Vol. 21, p. 27, 1950.

GUTTMAN, I. and MEETER, D.A. (1965) On Beale's Measures of Nonlinearity; Technometrics, Vol. 7, No. 4, p. 623, November 1965.

HARBAUGH, J.W. and MERRIAM, D.F. (1968) COMPUTER APPLICATIONS IN STRATIGRAPHIC ANALYSIS; John Wiley, New York, 1968.

HARDIE, R.H. and BALLARD, C.M. (1962) On Reducing the Periodic Error in a Telescope Drive; Publication of the Astronomical Society of the Pacific, Vol. 74, p. 242, 1962.

HARRIS, P.C. and LARGE, M.I. (1967) The Reduction of Radio Source Positions to Standard Epoch; Proceedings of the Astronomical Society of Australia, Vol. 1, No. 1, p. 31, January 1967.

HAYES, J.G. (editor) (1970) Numerical Approximations to Functions and Data; Proceedings of a Conference of the Institute of Mathematics and its Applications, at Canterbury 1967, University of London, Athlone Press, 1970.

HOAG, A.A. (1965), in I.A.U. Symposium No. 27, 1965.

HOVEY, G.R. (1973) A Phase-Locked Solar to Sidereal Frequency Converter; Proc. I.R.E.E. Australia, Vol. 34, No. 2, p. 59, March 1973.

I.A.U. (1965) The Construction of Large Telescopes; I.A.U. Symposium No. 27, Tucson Arizona April 1965. Crawford, D.L. (editor), Acedemic Press, 1966.

IRWIN, J.O. (1925) Detection of Outliers; Biometrika, Vol. 17, p. 238, 1925.

JONES, A. (1970) SPIRAL - A New Algorithm for Nonlinear Parameter Estimation using Least Squares; Computer Journal, Vol. 13, No. 3, p. 301, August 1970.

KLOCK, B.L., GELLER, R,Z, and DACHS, M.A. (1969) Inductosyn Angular Readout System of the U.S. Naval Observatory 6 inch Transit Circle; Proceedings of Electro-optical Systems Design Conference 1969, U.S. Naval Observatory Reprint No. 107.

KOWALIK, J. and OSBORNE, M.R. (1968) METHODS FOR UNCONSTRAINED OPTIMIZATION PROBLEMS; No. 12 of Modern Analytic and Computational Methods in Science and Mathematics Series, Bellman, R. (editor), American Elsevier Publishing, New York 1968.

KRON, G.E. (1960) Periodic Errors in Worm and Gear Telescope Drives; Publications of the Astronomical Society of the Pacific, Vol. 72, p. 505, 1960.

KUHNE, C. (1957) The Choice of Mounting for a Schmidt Telescope; Astronomical Journal, Vol. 62, p. 267, 1957.

KUHNE, C. (1971), in E.S.O./C.E.R.N. Conference Proceedings, 1971.

KUIPER, G.P. (editor) (1960) Telescopes; Vol. 1 of Stars and Stellar Systems series, University of Chicago Press 1960.

LAUSTEN, S. and MALM, B. (1971), in E.S.O./C.E.R.N. Proceedings, 1971.

LEVENBERG, K. (1944) A Method for the Solution of Certain Nonlinear Problems in Least Squares; Quarterly Applied Mathematics, Vol. 2, p. 164, 1944.

MALVICK, A.J. (1972) Theoretical Elastic Deformations of the Steward Observatory 230 cm and Optical Sciences Centre 154 cm Mirrors; Applied Optics, Vol. 11, No. 3, 1972.

MAR, J.W. and LIEBOWITZ, H. (editors) (1969) STRUCTURES TECHNOLOGY FOR LARGE RADIO AND RADAR TELESCOPES; M.I.T. International Symposium October 1967, M.I.T. Press 1969.

MARAN, S.P. (1967) Telescopes and Automation; Science, Vol. 158, No. 3803, November 17th 1967.

MARQUARDT, D.W. (1959) Solution of Nonlinear Chemical Engineering Problems; Chemical Engineering Progress, Vol. 55, No. 6, p. 65, June 1959.

MARQUARDT, D.W. (1963) An Algorithm for Least-squares Estimation of Nonlinear Parameters; J. Soc. Indust. Appl. Math., Vol. 11, No. 2, p. 431, June 1963.

MARTIN, L.D. (1967) Large Contact Ratios Minimize Effects of Gear Errors; Machinery 1967, p. 97.

MATARA, L.B. (1969) Electrical Drive Considerations for the 150 inch Stellar Telescope; A.U.R.A. Engineering Technical Report No. 14, 1969.

McCORD, T.B., SNELLEN, G. and PAAVOLA, S. (1972) The M.I.T. Automated Astrophysical Observatory; Symposium of Advanced Electronic Systems for Astronomy Santa Cruz 1971, Publications of the Astronomical Society of the Pacific, Vol. 84, No. 497, p. 220, 1972.

MEEKS, M.L., BALL, J.A. and HULL, A.B. (1968) The Pointing Calibration of the Haystack Antenna; I.E.E.E. Transactions on Antennas and Propagation, Vol. AP-16, No. 6, p. 746.

MEINEL, A.B. (1971) A 1.8 metre Lightweight Doubly Asymmetric Equatorial Telescope Design; Applied Optics, Vol. 10, No. 2, p.249, 1971.

MERTZ, L. (1971) Analysis of Telescope Costs; Bull. Amer. Astronomical Soc., Vol. 3, p. 386, 1971.

MIKHELSON, N.N. (1970) Some Problems of the Theory of the Altazimuth Mounting of a Telescope; Izv. Glav. Astron. Obs. Pulkove, No. 185, p.279, 1970.

MINNETT, H.C., YABSLEY, D.E. and PUTTOCK, M.J. (1967) Structural Performance of the Parkes 210 foot Paraboloid; in MAR and LIEBOWITZ (1967).

MINNETT, H.C. (1971) Progress on the 150 inch Anglo-Australian Telescope; Proceedings of the Astronomical Society of Australia, Vol. 2, No. 1, July 1971.

N.A.S.A. (1969) Optical Telescope Technology Workshop; held Alabama May 1969, N.A.S.A. SP233.

PEARSON, E.T. (1972) Fourier Transform Spectrometer Hydrostatic Bearing Design; A.U.R.A. Engineering Technical Report No. 46, K.P.N.O. 1972.

POPE, J.D. (1971) in E.S.O./C.E.R.N. Conference Proceedings, 1971.

PORTER, J.G. and SADLER, D.H. (1953) The Accurate Calculation of Apparent Places of Stars; Monthly Notices of the Royal Astronomical Society, Vol. 113, No. 4, p. 455, 1953.

POWELL, M.J.D. (1964) An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives; Computer Journal, Vol. 7, No. 2, p. 155, July 1964.

POWELL, M.J.D. (1965) A Method for Minimizing a Sum of Squares of Non-linear Functions without Calculating Derivatives; Computer Journal, Vol. 7, No. 4, p. 303, January 1965.

Product Engineering (1971) Flexure Bearings Support Telescope in Sky-Lab Mission; Product Engineering, Vol. 42, p.15, 1971.

REDDISH, V.C. (1966), in Sky and Telescope, Vol. 32, p. 124, 1966.

ROSENBROCK, H.H. (1960) An Automatic Method for Finding the Greatest or Least Value of a Function; Computer Journal, Vol. 3, p. 175, 1960.

RULE, B.H. (1965), in I.A.U. Symposium No. 27, 1965.

RULE, B.H. (1971), in E.S.O./C.E.R.N. Conference Proceedings, 1971.

RUSSELL, A. (1966) An Absolute Digital Measuring System using an Optical Grating and Shaft Encoder; Instrument Review, Vol. 13, p. 174, 1966. also N.E.L. Report No. 233, July 1966.

RUSSELL, A. (1969) Machine Tool Control - an Absolute Digital System; Control and Instrumentation, June 1969.

SCHWESINGER, G. (1969) Comparative Assessment of Aberrations Originating in Telescope Mirrors from Edge Supports; Astronomical Journal, Vol. 74, No. 1, 1969.

SCOTT, F.P. (1964) A Method for Evaluating the Elliptic E-terms of the Aberration; Astronomical Journal, Vol. 69, No. 5, p. 372, 1964.

SCOTT, F.P. and HUGHES, J.A. (1964) Computation of Apparent Places for the Southern Reference Star Program; Astronomical Journal, Vol. 69, No. 5, June 1964.

SEIFERT, L.I. (1969) Star-tracker Systems; N.A.S.A. Optical Telescope Technology Workshop 1969, p. 457.

SISSON, G.M. (1965), in I.A.U. Symposium No. 27, 1965.

SMITH, C.S. (1962) The Automatic Computation of Maximum Likelihood Estimates; N.C.B. Scientific Department Report S.C. 846/MR/40, 1962.

SMITH, L.V. Jr. (1967) Pointing and Tracking Accuracy - Recommended Standards; in MAR and LIEBOWITZ (1967).

SOLF, J. (1971), in E.S.O./C.E.R.N. Proceedings 1971.

SPANG, H.A. (1962) A Review of Minimization Techniques for Nonlinear Functions; S.I.A.M. Review, Vol. 4, No. 4, p. 343, October 1962.

STRUVE, O., EMBERSON, R.M. and FINDLAY, J.W. (1960) The 140 foot Radio Telescope of the National Radio Astronomy Observatory; Publication of the Astronomical Society of the Pacific, Vol. 72, No. 429, p. 439.

SWANN, W.H. (1964) Report on the Development of a New Direct Search Method of Optimization; I.C.I. Ltd. Central Instrument Laboratory Research Note 64/3.

SYDENHAM, P.H. (1968) Linear and Angular Transducers for Positional Control in the Decametre Range;

THACHER, H.C. and MILNE, W.E. (1960) Interpolation in Several Variables; J. Soc. Indust. App. Math., Vol. 8, No. 1, p. 33.

THEILHEIMER, F. and STARKWEATHER, W. (1961) The Fairing of Ships Lines on a High Speed Computer; Math. Comp., Vol. 15, P. 338.

TRUMBO, D. (1965), in I.A.U. Symposium No. 27, 1965.

VASELEVSKIS, S. (1962) On the Flexure of Fork Mounted Telescopes; Astronomical Journal, Vol. 67, No. 7, p. 464, 1962.

VASELEVSKIS, S. (1965), in I.A.U. Symposium No. 27, 1965.

VOKAC, P.R. (1970) Conceptual Design of a Shaft Encoder Interface; A.U.R.A. Engineering Technical Report No. 22, 1970.

VON HOENER, S. (1967a) Design of Large Steerable Antennas; Astronomical Journal, Vol. 72, No. 1, p. 35, 1967.

VON HOENER, S. (1967b) Homologous Deformations of Tiltable Telescopes; in MAR and LIEBOWITZ (1967).

WEIDLINGER, P. (1967) Control of R.M.S. Surface Error in Large Antenna Structures; in MAR and LIEBOWITZ (1967).

WEISFELD, M. (1959) Orthogonal Polynomials in Several Variables; Numer. Math., Vol. 1, p. 38, 1959.

WETZ, J.M. (1964) Criteria for Judging Adequacy of Estimation by an Approximating Response Function; Ph.D. Thesis University of Wisconsin 1964.

WHITWELL, A.L. (1972) private communication.

WISCHNIA, F.H. (1969) Precision Pointing of a Large Aperture Diffraction Limited Space Telescope; in N.A.S.A. SP233 O.T.T.W., p. 465, 1969.

WOOLARD, E.W. and CLEMENCE, G.M. (1966) SPHERICAL ASTRONOMY; Academic Press, New York 1966.