MEASURED DOSE DISTRIBUTIONS OF IODINE-125 SOURCES AND THE COMPUTERISED OPTIMISATION OF THEIR POSITIONS IN BRACHYTHERAPY PLANNING

William Ian Duncombe Rae

Department of Medical Physics, Groote Schuur Hospital and University of Cape Town, Observatory, Cape Town

Supervisor:

Professor David Shackleton

Department of Medical Physics, Groote Schuur Hospital and University of Cape Town, Observatory, Cape Town

April, 1987

This thesis is presented to the University of Cape Town for the degree of MSc Med(Medical Physics).

CONTENTS

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to all those without whom this work could not have been completed, and especially to my supervisor Professor D Shackleton who has been an encouragement and an expert advisor at all times.

I extend grateful thanks to Mrs M Bezuidenhout for typing this manuscript and for all the help she has given to me during the past year.

To Mr R Norris, who has patiently instructed me in the use of FORTRAN and computers, and to Mr J Korrubel who has helped in constructing the equipment I required, I would like to extend my heartfelt thanks and appreciation.

Also to all the other members and colleagues of the Department of Medical Physics of Groote Schuur Hospital and the University of Cape Town, I extend my gratitude, for their friendship, and for their advice on many aspects of this work.

To Groote Schuur Hospital and the University of Cape Town, which institutions made available their resources, I am deeply indebted.

My thanks go to the Medical Research Council for awarding me the Post Internship Scholarship which made this research possible.

Finally, I would like to express my gratitude to Kerry, my wife, for her love and tolerance during this time.

W. Rae

ABSTRACT

The use of I-125 seeds in brachytherapy is widespread and becoming increasingly varied. The spatial dose distributions around two types of I-125 seeds in general use, were measured using a Geiger-Muller chamber. Seeds with the I-125 adsorbed onto resin spheres had a 10% less anisotropic dose distribution than seeds containing a silver wire with the I-125 adsorbed onto it. An interpolative method was developed for fast dose calculations taking this anisotropy into account. An elliptical model using parametric representations was then developed to represent the positions and orientations of sources in space. Using the model and the interpolative method an optimisation procedure was performed to minimise the difference between the desired and achieved dose distributions by varying source positions. A sequential augmented Lagrangian technique (implemented by a mathematical library routine from the Numerical Algorithms Group, NAG*LIB.E04UAF) was used to perform the optimisation on a Sperry 1100/81 computer. The results are dependent on the problem conditioning and the initial estimates provided to the optimisation routine. The model is generalisable, but is best applied to small volume tumours requiring relatively few sources where precise positioning is desired and possible. The results of the optimisation can be applied clinically using documented techniques. With the use of parallel processing and graphics techniques this model could be widely applied to brachytherapy optimisation problems.

CHAPTER 1

INTRODUCTION

The major objectives of treatment planning in radiotherapy are, (a) to provide a complete specification of the distribution of absorbed dose produced in a given region of an irradiated patient and, (b) to determine the arrangement of the internal radiation sources or the orientation of the externally applied beams such that the optimal dose distribution is achieved (Laughlin et al, 1963). These are, in essence, the same as those specified for brachytherapy, that is radiotherapy with sources placed in or near the volume to be treated, by Meredith (1949). These objectives remain valid and applicable today, as then, to both brachytherapy and teletherapy, that is external radiation beam therapy. Much work has been done towards achieving these goals with the aim of ultimately optimising the treatment of cancer using radiotherapeutic methods.

Optimisation in radiotherapy involves in general the related disciplines of radiobiology, clinical radiotherapy, radiotherapy planning and medical physics. The mathematical formulation of many of the models used in the above disciplines is given in the text of Swan (1981), who approaches the entire subject from the point of view of optimising the effect and administration of the best practicable radiation dose to the patient. Specifically with regard to radiotherapy treatment planning, it is possible to divide the optimisation techniques into two fields. Firstly, "clinical planning" is done to specify the preferred dose

distribution. This involves the consideration of the best treatment modality or combination of modalities, the fractionation scheme, the localisation of the tumour and extent of tumour spread, the required dose to the area of treatment, and the tolerance doses to surrounding organs. Secondly, "physical planning" is done, entailing the development of the treatment plan which best approximates the preferred idealised distribution as given above (McDonald et al, 1977). This aspect of treatment planning optimisation involves the specific mathematical methods known as optimisation methods. Effectively these can only be applied once the clinical criteria for the optimal dose distribution and type of radiation are quantitatively specified (Bjarngard, 1977). Thus "optimisation" can be used in both contexts, generally (or clinically) and mathematically. Here the term will be used in the mathematical sense only.

The first of the major goals of treatment planning, that is to provide a complete specification of the distribution of the absorbed dose to the patient, has been contributed to greatly by improved calculation techniques of doses in tissue and the improved specification of the dose distribution delivered by the various treatment modalities. In brachytherapy planning the dose distribution calculations are frequently done on commercially available computer systems, although significant differences still exist even between "state of the art" systems for calculating the doses around radium sources (Tolbert et al, 1981). Many errors are inherent in the calculation of doses

around linear sources in brachytherapy and even limiting systematic errors to a minimum allows agreement to ±6%. This discrepancy is due to the calculation of tissue attenuation and build up factors, and the calculation of filtration in the source capsule. This does not include errors of reconstruction from radiographs and dosimetric uncertainties due to changes in the spectral quality with depth (Jayaraman et al, 1983). The exposure rate of Ir-192 sources has been calculated and it is dependent on the model used in the calculation. This is therefore a source of error in the calculation of doses around these sources (Glasgow, 1981).

The spatial dose distribution around implant sources has been measured and updated over the years as necessary. For I-125 sources (Krishnaswamy, 1978), (Krishnaswamy, 1979), (Dale, 1982), (Dale, 1983), (Hartmann et al, 1983), (Ling et al, 1983), (Ling et al, 1985), for Cs-137 sources (Krishnaswamy, 1972), (Diffey et al, 1975), for Au-198 sources (Dale, 1976) and for Ir-192 wire (Kwan et al, 1983), (Murphy et al, 1984), (Kline et al, 1985), both distribution measurements and calculations of dose distributions have been done. Young et al (1964), first calculated the dose distribution for Ra-226 using computer techniques. The Sievert integral has since been evaluated by a Monte Carlo technique for Ra-226 and Ir-192 sources, and the results agreed with experimentally obtained values (Williamson et al, 1983). The effects of positioning of the crossing needle on dose distribution from planar implants of Ra-226 has also been calculated with computer techniques and this has assisted in

better planning of treatment schemes (Doss et al, 1979).

The calculation of the treatment doses delivered has also improved and interactive programs have been developed (Bulski et al, 1983), (Rosen, et al, 1980), (Schultz et al, 1984), (van der Leije et al, 1983). Several algorithms have been developed for the calculation of the positions of sources in implants from radiographs taken after the implant (Biggs et al, 1983), (Siddon et al, 1985). These are very useful, but do not take the orientation of I-125 sources into account. In implants using I-125 adsorbed onto silver rods and encapsulated in titanium, it is possible to see the orientation from radiographs, hence this could be used in the calculation of doses. The rate of calculation of dose distributions in brachytherapy has also improved. Batten (1968) developed a method using lookup tables of precalculated values, and automatic source location, to give calculation times of the order of 4 seconds for four sources calculated at 1600 points. Recently Boyer et al, (1986) applied a fast Fourier transform technique that has improved calculation times further, dependant on the number of sources used. This technique allows fast dose calculations of acceptable accuracy around radioactive sources in space and will greatly decrease the time required to obtain isodose plots. An array processor was used in the implementation of this technique. The specification and calculation of the spatial dose distribution around brachytherapy sources has thus improved greatly over the previous 2 decades due to greater use of computers and mathematical techniques that are now widely available.

The second goal of treatment planning has also been approached. The seeking of the best arrangement of radioactive sources in brachytherapy planning can be termed the optimisation of the position and activity of the radioactive sources. Newton (1974) refers to the discussion as to the need and place for further use of mathematical optimisation methods in a field where some work has been done and where there are many radiobiological and clinical uncertainties. Optimisation is concluded to be worthwhile in some clinical situations and can be implemented practically. Although immediate benefits are not expected, they should be seen clinically after some time (Newton, 1974). Computers were envisaged as being the tool to use to develop the techniques and models required to achieve the optimal dose distribution in planning (Bjarngard, 1977). This is only useful to give the solution that best matches the clinician's assessment and this is an area where quantification is needed as often clinical specifications are uncertain. Computer tomography is of use in quantifying the tumour volume to be treated and has improved the clinician's assessment of tumours to some extent (Jose et al, 1983). The mathematical tools are thus unable to be used to their full potential because of clinical uncertainties, but this situation may be improved in the future.

The objective of optimisation is to quantitatively assess the goodness of fit of an achieved calculated dose distribution to an idealised desired dose distribution (Starkschall, 1984). This is done by developing a parameter to measure this deviation from ideality and then mathematically finding the minimum of this

parameter by adjusting the variables in the function giving the parameter. Criteria for describing the dose distribution quantitatively have been listed (Hope et al, 1967), and are:

(a) dose gradient across the tumour

(b) dose to the tumour relative to the maximum incident dose

(c) integral dose (total energy deposited in a volume)

(d) shape of the treated area relative to the desired treatment area

(e) dose to particular vulnerable organs

(f) dose in regions of possible direct or lymphatic extension.

Much work has been done in optimising external beam therapy planning since mathematical optimisation methods were pioneered for radiotherapy use with a 4 MeV Linear Accelerator (Hope et al, 1965). Linear Programming techniques were developed to minimise an objective function describing integral dose over vulnerable regions subject to linear constraints on the variables in the problem of multiple external beams being used in planning (Bahr et al, 1968). Starkschall (1984) used a non-negative linear least squares optimisation method (Lawson et al, 1974) to determine the best beam weights for known beam sizes and orientations.

Quadratic programming techniques were used to solve the second order objective function problem developed to minimise the variation from preselected doses to points in the tumour volume (Redpath et al, 1975), (Redpath et al, 1976). The algorithm used was the Beale Algorithm of the Numerical Algorithms Group FORTRAN

Library which allowed them the use of linear inequality constraints on the variable parameters used (Numerical Algorithms Group, 1983). Recently a microcomputer has been used by Legras et al (1986), to determine the optimal dose distribution in external beam radiotherapy using non-linear optimisation. The computer had an arithmetic coprocessor which reduced the calculation times fivefold.

In brachytherapy planning, Rosenstein (1977) developed a simple algorithm for iteratively determining the best loading of a cervical afterloading device using sources of the activities available. Optimisation of source dwell times has been done using non-negative least squares techniques as for the beam weighting done in teletherapy (Starkschall, unpublished work). A point source of Ir-192 with an assumed isotropic inverse square distribution was used and the dose to specific interest points was calculated and a fitting to desired doses was done. Similarly this technique has been used with a fourth degree polynomial describing the dose distribution around an afterloading source used in the treatment of cervical tumours (Pistorius et al, 1984). Tai et al (1979), used linear programming for optimising the loading of Cs-137 sources in the treatment of cervical carcinoma with constraints placed on the doses to several points of interest. In the solution of this problem it was not always possible to achieve the desired doses at all the points of interest. The points of interest required and the approach to optimisation of cervical cancer have also been described (Maruyama et al, 1976). Martin et al (1986) used a

similar optimisation procedure for treatment of endometrial carcinoma. The routine developed on a personal computer took the anisotropic dose distribution of the 10 Ci Ir-192 source into account and calculated the dwell times for sources positioned in three dimensional space.

The optimal positioning of radioactive sources had long been determined by the use of the Manchester System (Meredith, 1949) and the Paris System (Pierquin et al, 1978). These systems aim to fulfil the criteria for optimal dose distribution by applying firstly, a set of "distribution rules" for determining positions of sources, and secondly, a set of tables to calculate the activity of sources to use. These systems were developed from clinical experience with radium needles and Ir-192 wire implants respectively. A nomograph has been used by the Memorial Sloan-Kettering Cancer Centre for the calculation of the spacing of I-125 sources to give a uniform dose distribution and the activities required in number of sources of a given strength (Anderson, 1976) This was improved upon by Rao et al (1981), by the preparation of graphs of maximum central and minimum peripheral doses as a function of the number and spacing of the sources.

There have been attempts at solving the problem of determining by computerised optimisation methods the best positions of the sources in brachytherapy. This showed marked improvements in dose distributions in the treatment of brain tumours with a small number of high activity permanently implanted sources (Bauer et al, 1984). The I-125 sources used in this technique were

positioned in the brain tumours using a stereotactic device. To limit trauma the number of sources was minimised and the activities were high and optimised using available seed activities. The spatial orientation of the sources was fixed depending on the point of entry of the stereotactic probe into the skull. The method of optimising the positions was an iterative gradient method and fitted the achieved dose to a desired dose shape determined by surgeons from multiple computer tomography images.

Progress has thus been made toward the goal of positioning sources of available activity to give the best fit dose distribution to some idealised distribution. The orientation of sources with anisotropic dose distributions is critical in determining the dose absorbed. The surgical positioning of sources in the patient must also be precise to make the optimisation procedure worthwhile. The present methods of temporary implants usually use I-125 sources on plaques (Weaver, 1986), (Sealy et al, 1980) or encased in plastic tubing (Hering, 1986), (Sealy, work in progress). These both lend themselves to accurate spacing and predetermined orientation of the sources in the implant. Thus the use of an optimisation procedure that determines the best positions of sources subject to constraints on the spacing between sources and the orientation of the sources so that they lie on lines, would give a practical solution. This would also be a generalisable and flexible solution to the problem of optimising positions of sources in tumours.

Chapter 1 Introduction

The present good methods of determining where best to position sources in brachytherapy are limited mostly to regular geometries, or to special cases of linear source positions, or to having a small number of sources accurately positioned. In head and neck tumours irregular shaped masses with irregular extensions have been treated with implantation, but the best position of the sources has not been immediately obvious. This necessitates repeated "visual optimisation". Alternate plotting of calculated dose distributions and adjusting proposed source positions is done. A method of automatically optimising the positions and orientations of brachytherapy sources would therefore be useful, and the development of such a method has been done in this work. Specific emphasis has been placed on the optimisation of the positions of I-125 sources because of their favourable physical and clinical characteristics, as described in Chapter 2, but the techniques and model developed are not limited to any one brachytherapy source type.

CHAPTER 2

SPATIAL DOSE DISTRIBUTION OF I-125 SEEDS

2.1 REVIEW OF PREVIOUS WORK

I-125 Seeds (Medical Products Division/3M, 1982) are used extensively in brachytherapy to treat a variety of tumours and are applied into the tumours in a variety of ways. They have been placed on plaques and applied to the surface of the eye (Sealy et al, 1980), (Packer et al, 1980). I-125 seeds have been placed intraoperatively into the prostate gland (Hilaris, 1975), into the pterygo-palatine fossa (Goffinet et al, 1983), and into brain tumours as individual unattached sources (Bauer et al, 1984). They have also been used in plastic tubing as removable implants in head and neck tumours (Sealy, work in progress), as boosters to teletherapy treatments. It is thus important to assess the clinical response of tumours to specific doses delivered so that differing protocols may be compared and the best treatment may thus be used in each clinical situation. In order to do this the dose distribution around an I-125 seed must be known and used in the calculation of doses in tissue.

The spatial dose distribution around I-125 seeds is known to be anisotropic in the plane of the long axis and has been measured for various models of seeds. The spatial dose distribution around Model 6701 was measured using LiF thermoluminescent dosimeters, (Krishnaswamy, 1978), (Hartmann et al, 1983) and by using a silicon diode detector (Ling et al, 1983). The spatial dose

distribution around Model 6711 was measured using both a silicon diode, and thermoluminescent dosimeters. These methods yielded identical results (Ling et al, 1985). Data supplied by the manufacturer indicate some differences between the dose distribution of Models 6702 and 6701 at small deviations from the long axis. Weaver (1986) is presently measuring the spatial dose distribution of seeds of Model 6702 using LiF thermoluminescent dosimeters. The dose distribution of all three models calculated using a Monte Carlo technique is expected to be presented in the near future (Chiu-Tsao et al, 1986). The design of the three different Models is shown in fig.2.1 (Ling et al, 1983), (Weaver, 1986). Owing to the titanium welds present in the seeds' general configuration there will always be an anisotropic dose distribution until a new design of seeds is produced with uniform filtering in all directions (Ling et al, 1979).

The I-125 in the seeds decays by electron capture and gamma-emission to stable Te-125. Three major peaks of photon energies in the photon spectrum are detected, 27.4 keV , 31.4 keV , and 35.5 keV in all the models of I-125 seeds. Model 6711 Seeds contain two additional peaks at 22.1 keV and 25.2 keV due to fluorescence resulting from interaction with the silver wire (Ling et al, 1983), (Dillman, 1969). The Auger electrons produced are all stopped by the titanium capsule and do not contribute to the dose delivered (Medical Products Division/3M, 1982). The dose deposition in tissue has been calculated from tabulated build up factors (Kornelsen et al, 1981) and dose deposition tables for energies in the range of the photons of I-125 (Berger, 1968). The

characteristics of this low energy spectrum and the half life of I-125 being 60.2 days have caused it to be widely used in brachytherapy. It is easy to shield and hence radiation protection is relatively simple. Its half-life is longer than that of Rn-222 (3.823 days), or Au-198 (2.693 days), and thus its shelf life is longer. I-125 Seeds are suitable for both permanent and temporary implants because of their intermediate length half-life. The radio-biological effect (RBE) of I-125 has been shown to be approximately 1.5 . Thus an I-125 source delivering the same dose to tissue as an Ir-192 source, gives a greater radio-biological effect (Marchese et al, 1985). To include the effect of differing radio-biological effects the dose to tissue is given in Gy equivalent (GyEquiv) which is the dose delivered by Ir-192 photons which would give the same effect as that of the isotope in question.

The calculation of dose at points around I-125 Seeds has been done by several authors using various mathematical models. Different mathematical models are used for different models of seed. Generally the dose rate, D(t), (cylindrically symetrical around long axis), at distance, r, from the source's centre and angle, t, from the source's long axis, for activity, A, of an isotope with a specific dose constant, SDC, can be given by:

$$D(t) = A.SDC.g(r,t)/(r^2)$$

where the spatial distribution function (SDF) is given by g(r,t). Calculation using an assumed point source distribution modified by an anisotropic factor has been done (Hartmann et al, 1983). The theoretical Monte Carlo calculations done by Dale (1983)

assume a point source, but he does mention in conclusion that for calculations of doses an anisotropic factor must be included. Ling et al (1985), use a matrix fit technique to fit an analytic function to experimentally obtained data, and a good fit was achieved with the formula;

$$D(r,t) = e^{-ur}.(a + b.r + c.r^2 + d.r^3)$$

where D(r,t) is the relative dose distribution and u, a, b, c, d, are t dependent parameters. A matrix of fitted values for the parameters is given with error estimates on each parameter. This means for each point, the calculation of dose entails, first determining the distance and relative angle between the source and the point of calculation, then looking up parameters for the final calculation of the dose using the analytical formula. The fitted values of the parameters would be different for different models of seed.

When optimisation of dose distributions is done many dose calculations at many interest points are necessary. A look-up table with an adequate range of values would be a faster more efficient way of calculation than an analytical formula, although requiring a larger computer memory. In this work the dose distributions around I-125 Seeds (Model 6702 and 6711) were measured using a Geiger-Muller chamber. The data for Model 6711 were compared to published data obtained using other methods of measurement and the results were found to be the same within acceptable confidence limits. A table was created to allow fast dose calculation around Model 6702 Seeds (other similar tables could be created for other models of seed from published data).

2.2 METHOD

Four I-125 Seeds of Models 6702 and 6711 in the activity range of 27.4 - 39.2 MBq (0.74 - 1.06 mCi), (The order of magnitude used in clinical applications) were used for the measurement of the relative spatial dose distribution. All measurements were done with the individual seeds balanced on the tip of a polystyrene jig (See fig. 2.2a) in a Therados RFA 3 water phantom under 10 cm of water. The angular orientation of the seed in a horizontal plane could be varied with an accuracy of 0.5 degrees and the position of the centre of the seed could be moved around the tank in the plane with an accuracy of 0.1 mm. Rotation was about the centre of the seed. For a diagram of the experimental apparatus, see fig. 2.3.

All readings were taken using a Geiger-Muller probe 1.6 mm in diameter and 16 mm long encased in a 0.6 mm thick lead tube with a tapered aperture 1.1 mm wide extending around half the circumference of the lead shield (See fig. 2.2b). The apparent aperture was thus 1.1 x 1.6 mm allowing scatter detection up to 60 degrees from horizontal. (The HVL in lead for I-125 photons is 0.025 mm). The thickness of the wall of the chamber was 0.05 mm. The probe was connected to a MIM-EON Counter to register the events detected. The response range of the chamber from the initial response to a continuously discharging state was determined by varying the voltage over its entire range while being exposed to an I-125 Seed. A voltage in the centre of this response region was applied to the detector and was kept constant for all the measurements . The timer scale was set to 30 seconds

for all readings. If the counts were statistically too few after 30 seconds then the readings were repeated until an aggregate count was such that the relative standard deviation was less than 3%. 1000 counts was taken to be the lower limit for counts detected.

The dead time of the chamber was measured using 2 seeds at a distance of 1 cm from the detector. Seeds were counted together then individually and the sequence was repeated. This was done on 2 separate occasions.

The scatter component of the radiation from an I-125 seed was measured both to determine the direction of the scatter detected and the amount of scatter detected. Two collimating lead shields were constructed for this purpose. A shield was constructed 6.6 mm in outside diameter with 2.4 mm thick walls and a 1 mm x 1 mm aperture in one face. The chamber was placed in this shield, 1 cm from a seed and rotated about its central axis thus detecting the relative number of counts from different angles about the detector. The other collimator was constructed with a band 1 mm wide between the two separate segments of collimator and the edges were tapered so that all radiation up to 60 degrees from the horizontal plane in all directions could be detected by the band of chamber visible. A removable shadow shield was interposed at different distances between the detector and the source. The shadow shield was 1 mm thick gold plate identical in size to the I-125 Seed. Counts were taken with and without the shield in place half way between the detector and the seed, at

distances of 1 cm and 2 cm from the detector to the source.

Absorption coefficients of some materials were measured using the same experimental configuration. Sheets of varying thicknesses of the materials were interposed between the detector and the seed. This was done for wax, water and acrylic.

The effect of a gold sheet immediately behind an I-125 Seed was measured. A holder that allowed a gold foil 0.3 mm thick to be placed immediately behind the seed without disturbing its position, was used to support the seed in the tank. Counts were taken at various angles to the gold foil with and without the foil in place. This was done for 4, I-125 Seeds Model 6702 at angles of -20 , 0 , and +20 degrees from the perpendicular bisector of the seed in the horizontal plane of the long axis of the seed perpendicular to the plane of the gold foil. It was also done at 45 degrees from the plane of the foil in the plane of the perpendicular bisector of the seed. (See fig. 2.3a)

The 2-dimensional dose distribution of Model 6711 seeds was measured. Counts were measured at distances of 0.5, 1, and 2 cm from the centre of each seed at angles of 0, 10, 20, 30, 40, 50, 70, 90 degrees from the axis of each seed in the horizontal plane in all quadrants.

The 2-dimensional dose distribution of Model 6702 I-125 Seeds was measured for 4 seeds. Counts were taken at distances of 0.5, 0.75, 1.0, 1.5, 2.0, 2.5, and 3 cm and at the same angles mentioned above. All measurements for an individual seed were done on the same day without switching off or readjusting the

counter. Daily checks on the stability of the instrument were done to determine if a sequence of random counts measured could with confidence be said to be normally distributed. A test on the consistency of means described by R T Birge (Worthing et al, 1944), was done to check that the counts came from the same normal distribution.

2.3 RESULTS

The experimental setup was found to be stable and to render results within expectations for random counters. The daily checks confirmed this reproducibility. The value of H, ($H=\sqrt{n}(pe/pi-1)$ where n is the sample size and pe and pi are the external and internal consistencies respectively), for 8 sample counts was less than 1.14 on all testing days (Accept the test if $H \leq 1.83$), (Worthing et al, 1944). The instrument was stable on the 30 second time scale. The dead time was found to be 0.425 ms $\pm$ 0.066 ms after 4 sets of data were obtained from measurements taken on 2 separate occasions. This implies a count rate of 471 c.p.s. or 14130 counts per 30 seconds, for a 20% count loss. This was adequate for all readings down to 0.75 cm from the source. At 0.5 cm from the source some of the readings on higher activity seeds exceeded this 20% loss limit. The highest reading was 26208 which implied a true count of 44465 in 30 seconds. In only 2 of 16 quadrants at 0.5cm did readings exceed 20 000 which, implies a 31% count loss.

The scatter component was found to be 16.4 $\pm$ 2.8% at 1 cm and 35.3 $\pm$ 5.7% at 2 cm. This is the percentage difference between the counts measured with and without the gold shadow shield in place half way between the detector and the source. The direction from which the scatter component was measured by the detector was found to be predominantly within $\pm$ 60 degrees of directing the collimator aperture toward the source. Outside the penumbral region of the collimated chamber, which was $\pm$ 30

degrees , the scatter counts measured were found to be 3% of the counts measured on the perpendicular bisector of the seed. At ± 60 degrees from the direction facing the seed the counts had dropped to 1% of the counts on the perpendicular bisector. Beyond 90 degrees of rotation of the collimated chamber the lead shield acted as a shadow shield and hence the counts measured beyond this rotation were artificially low.

The linear attenuation factors measured were found to be different for substances often taken to be equivalent at higher energies. See table 1 for measured values.

Substance	Linear Attenuation Coefficient
Water	0.261/cm ± 0.003/cm
Wax	0.218/cm ± 0.002/cm
Acrylic	0.173/cm ± 0.002/cm

TABLE 1 LINEAR ATTENUATION COEFFICIENTS OF I-125 PHOTONS WITH THE STANDARD ERROR ON THE ESTIMATE

The relative dose distribution at the distances and angles around the seeds mentioned above, was measured. The readings were taken from 4 seeds of Model 6702 over 4 quadrants per seed. The counts were corrected for dead time losses. At each distance the readings were normalised in each quadrant to the average of the reading at that distance in that quadrant, to eliminate irregularities with respect to the loading activities of the resin spheres. The means and standard deviations of the readings at each distance and angle in the 16 quadrants were calculated. The results were then all normalised to the reading at 1 cm and at 90 degrees to the seed axis. The normalised readings were then

multiplied by the distance squared to remove the inverse square dependence and are plotted in fig.2.4 . The error bars shown are the standard errors on the mean readings. The greatest standard error on the mean was found to be 8.8%. The largest errors were all found in the measurements along the axis of the seed. The dose distribution was found to be markedly anisotropic and the dose along the axis of the seed was found to be between 43% and 62% of that at 90 degrees to the axis, dependent on the distance from the source.

The dose distribution around I-125 Seeds Model 6711 was measured at the angles and distances mentioned in the method above to compare its agreement with published data. The data was analysed as above and nine points were compared to the matrix fit method as proposed by Ling et al (1985). The errors quoted on the published data were calculated from the individual parameter errors quoted, using a standard error combination formula (Topping, 1962). A Chi Squared fit was performed with 5 degrees of freedom to test the four parameter model. The results agree to a 0.05 confidence level. A Kolmogorov-Smirnov non-parametric test was performed on the nine paired data points and the results again agreed to a 0.05 confidence level (Siegel, 1956). The method used was thus shown to give with confidence values comparable to published values that were obtained using other techniques for measurement of the dose distribution for Model 6711 Seeds. This method was thus validly used for measurement of the dose distribution around Model 6702 Seeds.

2.4 DISCUSSION

The consistency and stability checks done initially showed that the MIM-EON Counter was suitable for counting events from the Geiger-Muller chamber used, at the settings used.

The chamber used was a thin-walled, Geiger-Muller chamber that was shielded to have a small responsive volume (2.2 mm^3). This was suitable for measuring counts in a small volume in space. The dead time was long, but as most of the counts measured had a less than 20% loss this was suitable for the activities chosen and the counts registered. The photon energy spectrum of I-125 has been shown not to vary significantly with increasing depth in water due to the marginal predominance of the Compton interaction over the photoelectric effects at those energies. The loss of energy in Compton interactions is small. The competition is thus essentially between total absorption of photons, and a change in the direction of travel of the photon (Dale, 1983). This accounts also for the large scatter component in the detected radiation. The depth in water at which 50% of the dose from I-125 is deposited is due to scatter is 1.7 cm. These results by Dale were calculated using Monte Carlo techniques, the principles of which are reviewed by Raeside (1976). The shielding of the chamber used here allowed for detection of radiation from, $\pm$ 60 degrees from the horizontal in the vertical direction, and $\pm$ 110 degrees in the horizontal plane from the line joining the source and the detector. This thus allowed detection of most of the radiation both scattered and direct. The counts detected at different distances were all analysed relative to the counts on

the perpendicular bisector of the seed's axis at 1 cm distance. The invariance of the energy spectrum at different depths allows comparison of counts without concern for slight variation in energy response of the chamber used. The measurements, all being done at a depth of 10 cm water in the RFA-3 water tank, were effectively taken in a full scattering medium with the other absorbers and scatterers kept to a minimum. The large volume of the shielding used decreased the scattered component of the radiation by an equal proportion for all orientations of the source. The counts were all corrected for dead time losses and were always greater than 1000 counts so that the standard deviation intrinsic in measurements of random counts was kept to less than 3.2%. The background counts were negligible for the 30 second counting period used. No amplification above that already in the counter was needed, thus electronic noise was negligible.

The measurement of the scattered component of radiation did not agree with the theoretically calculated values. It was expected to detect a greater than 50% scatter contribution at 2 cm distance from the source. The measured value of 35.3 $\pm$ 5.7% was low due to the imperfect geometry used. This did not influence the relative spatial dose distribution measurements as the relative error was the same in all the dose measurements done and thus cancelled out during the analysis. The solid angle above and below the detector contained in the cones above and below 60 degrees from the horizontal were not available for counting and the scattered radiation excluded by the shadow shield was also not counted. The direction of the scatter was found to be

predominantly forward in direction and could thus be well detected by the collimated chamber used in all other measurements.

The attenuation of the substances tested was found to vary greatly. This emphasises the difficulties in calculating doses delivered by I-125 to different tissues. It is therefore also necessary to use a homogeneous medium for measuring the relative doses to points around I-125 Seeds, or to correct for any irregularities and inhomogeneities in the medium used, after measurement of the attenuations of the substances in the medium has been done.

The method used to measure the dose distribution around the I-125 Seeds (Model 6711) gave results which agreed within experimental error with the published data (Ling et al, 1985). They used thermoluminescent dosimeters in a lucite phantom and a silicon diode in an RFA-3 water phantom to measure the relative dose distribution around the seeds. These methods both have difficulties in the description of the energy response to low energy photons. The energy response of LiF is non-linear, and the LiF was calibrated using an ionisation chamber exposed to the I-125 spectrum (Weaver, 1984). As relative measurements are being done and the spectrum does not change with depth then these methods are both valid, but experimentally, possibly more time consuming.

The dose distribution around I-125 Seeds Model 6702 was found to be markedly anisotropic and about 10% more isotropic than Model

6711 Seeds (For comparison with a point source see isodoses in fig. 2.5). The distribution was fitted to two analytical models, but they both showed marked inconsistencies and unacceptable trends.

The mathematical model used to fit the relative dose distribution surface in polar coordinates, r, t described the spatial distribution function (SDF), (Cylindrically symetrical) and was;

$$SDF\ (r,t) = e^{-ur} .(a(t) + b(t)r + c(t)r^2)$$

The dose rate at a point was given by;

$$D = A.\ SDF(r,t).\ SDC\ /\ r^2$$

where A is the activity of the source and SDC is the Specific Dose Constant of I-125 in water.

The dependence of u on t was found to be minimal and u's value was calculated to be 0.3901 $\pm$ 0.0173 . This method is similar to the matrix fit method described by Ling et al (1985) and the fitted parameters with errors are given in Table 2.

t/degrees	a(t)	b(t)	c(t)
0	1.87$\pm$0.22	-1.46$\pm$0.35	0.38$\pm$0.11
10	1.91$\pm$0.22	-1.35$\pm$0.34	0.36$\pm$0.11
20	1.94$\pm$0.24	-1.05$\pm$0.38	0.29$\pm$0.12
30	2.07$\pm$0.16	-0.99$\pm$0.26	0.27$\pm$0.08
90	2.15$\pm$0.14	-0.84$\pm$0.23	0.23$\pm$0.07

TABLE 2 FITTED PARAMETERS FOR THE RELATIVE SPATIAL DOSE DISTRIBUTION FUNCTION OF AN I-125 SEED

The second fitted model used the same exponential term and the

squares Chebychev series.

$$SDF(r,t) = e^{-ur} \sum_{i=0}^{2} \sum_{j=0}^{3} A_{ij} T_i(XCAP) T_j(YCAP)$$

where $T_i(XCAP)$ is the Chebychev polynomial of the first kind of degree i, and $T_j(YCAP)$ is similarly defined. This fit was done using the Numerical Algorithms Group Library routine E02CAF and the fitted values were calculated using routine E02CBF. The fit was poor at the extremes of the surface and hence could not be validly used.

The method used for calculation of doses from the measured data was linear interpolation between the array values of a 101 by 101 data array. These array values were calculated using bicubic splines to fit the original data points. Details of the calculation will be given in the section on the application of this data to dose calculations (Chapter 2.5).

The I-125 Seeds Model 6702 has an anisotropic distribution and although it was available in higher activities than other seeds at one time it has now been superseded by the use of Model 6711 Seeds which also have high activities available, but are much more visible radiographically and thus have real advantage over other models of seeds (Ling et al, 1983). Both Models 6711 and 6702 were in use at Groote Schuur Hospital, Cape Town, during 1986 and thus the spatial dose distribution was required of both of these models.

2.5 APPLICATION

It is necessary to apply the measured dose distribution data to the calculation of doses at points in space delivered by one or many seeds. The analytical methods available require calculating doses using a function of distance and angle around the seeds axis. The parameters of the function are either fixed or variable and may be dependent on the angle around the seed (Ling et al, 1985). These methods although giving a good estimate of the dose delivered to a point do take some time to calculate. When repeated calculations of doses at many interest points delivered by many seeds at slightly different orientations or positions are done, they become time consuming. Thus when implementing an optimisation procedure that may require up to approximately $k \times N^3$ calculations of the dose at all the interest points from all the N sources, where k is a proportionality constant the speed of calculation of dose is limiting on the size of problems that can be addressed.

An array of values of dose per unit activity, D(r,t), where;

$$D(r,t) = SDF(r,t) \cdot SDC / r^2$$

and r = distance from source to interest point

t = the angle from the seeds major axis

SDC = Specific dose constant of isotope

SDF(r,t) = Spatial distribution function

was created so that values could be obtained after linear interpolation directly from the table. The inverse of distance and sin(t), are used as the variable axes for the array of values. A point of interest is represented as (xp, yp, zp) and a

source as (xs, ys, zs, ps, ts) where ps and ts are the spherical coordinates, given the orientation angles around its centre of the axis of the source. The square of the spatial distance between the point of interest and the source is;

$$dstsq = (xp - xs)^2 + (yp - ys)^2 + (zp - zs)^2$$

and the sin of the angle between the unit vector in the source axis and the vector between the source and point of interest is

$$\sin(t) = (1 - UR / dstsq)^{0.5}$$

where UR is the dot product of UVEC, the unit vector in the direction of the seeds axis, given by

UVEC = (cos(ts) x sin(ps), sin(ts) x sin(ps), cos(ps))

and RVEC, the vector between the source and the point of interest,given by;

RVEC = ((xs - xp), (ys - yp), (zs - zp))

The values of 1/r and sin(t), are scaled and used to look up four adjacent values from a table. The scaled values are the array element numbers (position markers) in the two axes. Two dimensional linear interpolation is done between these 4 points to give a value of the dose per unit activity at any distance and angular orientation. Rotation around seed's long axis immaterial.

Linear interpolation is adequate as there are 101 points in the sin(t) and 1/r axes. These points are obtained by fitting bicubic splines to the original data. The bivariate function surface obtained using these variables as axes is suitably flat and follows easily from the calculations that need to be performed to determine the relationship between the source and interest points. The value, D(r,t), thus obtained can be used to

calculate dose rate, DR, in Gy/hr for a known activity of seed, A, in MBq, as follows;

$$DR = A \times D(r,t)$$

for each value of D(r,t) interpolated from the table in units of Gy/(hxMBq). The table values are obtained at points of a 101 by 101 array with even step sizes in 1/r from 1/(maximum distance) to 1/(min distance), and sin(t), from 0 to 1, by multiplying the value of $1/(r^2)$, by the specific dose constant for I-125 in water, and Val, the value obtained from NAG*LIB.E01ACE, (which performs the bicubic spline fitting routine), as follows;

$$D(r,t) = Val \times SDC / (r^2)$$

The creation of the table takes approximately 1.0 minute of CPU time of the Sperry 1100/81, indicating each operation would take 0.001 seconds, which is prohibitively long if many calculations are done for each dose distribution calculation used in an optimisation procedure, and hence it is faster to use the linear interpolation method.

This array look up table method may be used with other isotopes and is easily generalised to any implant source. The dose distribution measured can easily be used to calculate the doses in planning without the use of fitting of an analytical formula to the data first, but rather by interpolation between known data points.

CHAPTER 3

OPTIMISATION IN BRACHYTHERAPY

I METHODS INVESTIGATED

To minimise the difference, mathematically, between some "ideal dose distribution" desired by the radiotherapist, and the dose distribution achieved by the planning staff, several parameters can be varied. These are generally:-

1. The number of sources used
2. The activities of the individual sources
3. The positions of the sources in tissue
4. The type of radioisotope used, and its form
5. The shielding around the sources

The desired dose rate and the total dose to be given with the treatment, are usually predetermined by the radiotherapist. Thus knowing the area of the region to be treated and the distance from the plane of the sources at which the desired dose is specified , an estimate of the activity required can be obtained from tables of mghrs (mgs Radium x hours implanted), for surface applicators and planar implants (Johns et al, 1983). The generalisation of these estimates from Ra-226 to other isotopes is not always exactly correct as the attenuation in tissue of other photons may differ from those of Ra-226 and thus the required activities to achieve the same dose at a distance may also differ. An estimate of the total activity can also be obtained from the average dimension of the volume to be treated, by using a nomogram (Anderson, 1976).

Chapter 3 Optimisation in Brachytherapy

The available activities of isotopes are usually fixed to a small number of activity ranges and thus with the total activity fixed usually the number and activities of the sources are also fixed to a few different discrete choices. This does not lend much scope to mathematical optimisation, unless a large source delivering a high dose rate is used and the position of this source is varied in precisely timed steps. In this case the dwell times of the source at various points in space can be varied to effectively give a series of variable activity sources positioned at predetermined positions as in the treatment of carcinoma of the cervix. The activities (or dwell times) can be varied mathematically to give the optimal dose distribution (Pistorius et al, 1984). The method used was a non-negative least squares method that is well described, and a FORTRAN program implementing it is available (Lawson et al, 1974). The same method has been used in the optimisation of external beam treatment plans (Starkschall, 1984). The method minimises a least squares objective function subject to non-negativity constraints. The algorithm is finitely convergent, and this allows fast solutions to be achieved.

This implementation was emulated on a Sperry 1100/81 using a simplified dose rate calculation to a point in space as follows:

$$DA_i = \sum_{j=1}^{N} A_j .SDC / (d_{ij})^2 \quad .....1$$

where DA_i is the dose achieved at the point i

N is the number of sources used

d_{ij} is the distance from the point, i, to the

source, j,

A_j is the activity of the source j.

This yielded good results for points of interest and sources described in 3-dimensional Euclidean Space. The activities were given accurately and thus accurate dwell times could be determined for the use of afterloading devices such as those used in the treatment of carcinoma of the cervix. The time to find the optimal dwell times of a source stopping at 7 points along a straight line to give the desired doses at 5 points of interest, was 1.9 seconds of Central Processor Unit (CPU) time. This method of optimisation is not readily applicable to I-125 Seeds as these are either placed as permanent implants, or as a plaque containing all the sources, or as groups of sources in plastic tubes. The sources thus cannot easily be removed individually to give the desired dwell times. I-125 Seeds are usually available in activity ranges and are not accurately calibrated. For these reasons optimising the activities of I-125 using a mathematical method is inappropriate.

The choice of the type of radioisotope to be used is usually done by the radiotherapist and this is usually part of the clinical planning aspect of the optimisation of treatments. The most appropriate isotope to apply optimisation techniques to is I-125, because of its clinical and physical properties. The dose distribution is well described. The low penetration of I-125 photons in tissue and the thus very localised distribution allow a very specific well delineated area to be treated. Where accurate doses are required in small areas this is the isotope of

choice. The clinical advantages of easy shielding and the increased radiobiological effect over other isotopes also contribute to this choice of I-125 as the favoured isotope (Marchese et al, 1985). Variation of this parameter is done clinically.

The shielding of radiation from surrounding structures is easily done for I-125, but is not as easy for isotopes with higher energy photon emissions. The half value layer of lead for I-125 photons is 0.025 mm. Thus inert materials such as gold and stainless steel can be used to provide adequate shielding with relatively thin sheets of these materials. Usually 0.3 mm sheets of pure gold are used for the plaques of intraorbital implants (Sealy et al, 1980). Thus areas where radiation is not required can easily be shielded and effectively directional implants can be constructed. The precise spatial relationship between the shielding, and the seeds, in an implant could be optimised, but due to the limitations usually present on the position where shielding may be used it is usually not done as shielding is either possible or not possible in clinical cases. Precise positioning of the shielding would make a difference to the final dose distribution, but no optimisation model has yet been developed to represent this problem.

The field thus most appropriate to apply optimisation techniques to, is the optimisation of source positions. This has been done by Bauer et al (1984) using an iterative gradient search method in 4 dimensional parameter space with 3 dimensions spatial and

the 4th dimension the number of sources. A weighted least squares objectives function was used to give a measure of the difference at a set of points in tissue between the desired doses at those points and the achieved doses at those points. No constraints were placed upon the problem. High activity seeds were used, and it was reported that good fits were obtained to desired isodose curves by varying the position of the sources in space. The anisotropic dose distribution of the I-125 seeds was taken into account when doing the calculations, but the orientation of the sources was not varied as this depended on the stereotactic device used to position the sources in the brain tumours for which purpose the technique was developed. To limit trauma to the brain the minimum number of seeds was desired and the method was thus applied only to a small number of seeds.

In this work initial attempts were made to reproduce the above results using the simplification of an isotropic dose distribution. A more powerful optimisation technique to alter the positions of the sources in 3 dimensional space was used. Many optimisation techniques are available and one was chosen that took multiple inequality and range constraints into account, and that had good convergence properties (Bunday, 1984). The optimisation technique used was a sequential augmented Lagrangian technique (see Appendix 1) which solved the minimisation sub-problems using a quasi-Newton technique (see Appendix 2). The dose rate, $D_{ij}(r)$, at a point, i, from a source, j, was given by:-

$$D_{ij}(r) = A_j . RDF(r_{ij}) . SDC . / r_{ij}^2 \qquad \ldots\ldots\ldots 2$$

where A_j is the activity of the jth source

SDC is the specific dose constant of I-125 in water.
(3.636×10^{-4} Gy cm^2/hr.MBq , (Dale, 1982)

r_{ij} is the distance between the source j and the point i

and $RDF(r) = a_0 + a_1 r + a_2 r^2 + a_3 r^3$ 3

where $a_0 = 0.97987$

$a_1 = 0.07962$

$a_2 = 0.07914$

$a_3 = 0.83326 \times 10^{-2}$ (Dale, 1982)

The technique was developed from the one dimensional case with 1 source on a line with a point of interest on the same line. No constraints were placed upon the problem. The objective function was given by an unweighted least squares function that was easily extended to more than one source and more than one interest point. Thus for n points of interest and m sources, where $D_{ij}(r)$ is as above, and DD_i, is the desired dose at a point i, the value of the objective function F was given by:-

$$F = \sum_{i=1}^{n} \left(DD_i - \sum_{j=1}^{m} D_{ij}(r)\right)^2 \quad .....4$$

This function is poorly behaved. When a source coincides with an interest point, a singularity occurs and when all the sources are far from the interest points the function tends toward a limit. In the region of a local minimum of equation 4, the function is convex and a solution can be found. The optimisation routine used to implement this was NAG*LIB.E04UAF (see Appendix 5) which attempts to solve problems of the form

Minimise $F(x)$; $x \in E^n$

where $g(x) = 0$ are equality constraints on the variable

and $h(x) > 0$ are inequality constraints on the variables

and F,g,h, are all continuous functions into E^n

(Numerical Algorithms Group, 1983)

The objective function above was found to achieve a solution if a good estimate of the source positions was given as a starting point. The number of iterations used and the number of calculations of the value of the function was strongly dependent on the starting point and the specific parameters used to control the calculations by the routine E04UAF. For the unweighted least squares function it was found that a small starting value of RHO, the penalty parameter, (See Appendix 1) and accurate minimisation of the subproblems, ($0<ETA<<1$, where ETA is the error limit of minimisation), gave good convergence to a local minimum. The convergence properties of the routine used are very good and are of the best presently available for optimisation routines (See Appendix 1 for discussion of the convergence properties of the sequential augmented Lagrangian technique) (Bertsekas, 1976).

The poor convergence with the objective function 4 was overcome by multiplying the function by the sum of the distances between an individual point and all the sources.

Thus $$F = \sum_{i=1}^{n} \left(\left(DD_i - \sum_{j=1}^{m} D_{ij} \right)^2 \times \sum_{j=1}^{m} d_{ij}^2 \right)$$

where, d_{ij}, is the distance between the ith point and the jth source. Where an exact solution to the problem is possible and all the doses desired can be achieved then this function is acceptable. When only an approximation is achieved then the

points are effectively weighted by the factor, $\sum_{j=1}^{m} d_{ij}^{2}$. This method yielded much faster convergence for soluble problems, but was unacceptable because of the weighting introduced.

A set of constraints was then introduced to limit the positions of the sources to be on ellipses. This necessitated the development of a model that allowed representation of sources on ellipses in space with suitable variables allowing optimisation of the doses at the points of interest. This newly developed model is described in detail in the following section.

3.2 ELLIPTICAL MODEL

A parametric model was developed based on the generalised ellipse in 3-dimensional space. The equation of a vector describing an ellipse in 3-dimensional space was given by;

$$r(t) = \sum_{i=1}^{3} (a_i \sin(t+b_i) + c_i)u_i \qquad \ldots\ldots\ldots 1$$

where t is the parameter defining angle around the ellipse from a fixed reference point given by the vector, r(0)

a_i defines the amplitude in the ith axis

b_i defines the phase shift in the ith axis

c_i defines the central shift in the ith axis and

u_i defines the unit vector in the ith axis.

Thus using this as a model the I-125 Seeds can be represented as lying on the ellipse orientated in the direction of the tangent to the ellipse by a single parameter variable, t. The ellipse itself is represented generally by 9 variables, $(a_{1,2,3}; b_{1,2,3}; c_{1,2,3})$, which for an individual seed on an individual ellipse allows totally general positioning and orientation of that seed. If several seeds are defined to lie on one ellipse then their positions are related and their orientations are limited to lie on elliptic arcs in space. These arcs can be generalised to most of the commonly used curve shapes. A straight line segment of length, 2||F||, is obtained when $b_1 = b_2 = b_3$ in equation 1. Elliptical line segments are obtained when no other special case is present. Circular line segments are obtained when the two half axes of the ellipse are equal in length. Approximations to parabolic curves are obtained

when the line segment is taken in the region of one of the axes of the ellipse. Thus with the use of several ellipses in space most smooth curves can be approximated by segments of the ellipses used, and with the use of straight line segments angular positioning of the sources can be readily be done. See fig. 3.1 for some examples of the generality of ellipses.

The seeds thus represented as parameter variables on an ellipse must be constrained to lie greater than one seed length apart on the ellipse. The length of arc between two points on an ellipse was thus constrained to be greater than 5 mm, the length of the seed being 4.8 mm (See Appendix 4). The calculation of the length of arc requires an increasing number of terms to achieve sufficient accuracy as the ellipse tends towards a straight line. The accuracy achieved by 10 terms in the series is acceptable (<0.5% over 1 quadrant), for the case where the ratio of the long to the short half axes is less than 10 to 1. The eccentricity of the ellipse or the ratio of the long to short half axes must thus be constrained. The inequality constraint function being;

$$A/B - 0.1 > 0$$

where A is the length of the short half axis

B is the length of the long half axis.

This constraint on the ellipse does not prevent the size of the ellipse being infinitely small, thus an absolute constraint is required on the circumference of the ellipse such that it can contain all the seeds that are required upon it. This implies the circumference of the ellipse is given by C then;

$$C > A.2.\pi \qquad \ldots\ldots 4$$

where A is the length of the short arm of the ellipse, and we thus have the inequality constraint;

$$A.2.\pi - 0,5.N > 0$$

where N is the number of sources on the ellipse, and π is given to 9 decimal places. This is the smallest possible length of C, and it is not necessarily the practicable length. This constraint is placed on the shape of the ellipse to prevent it being reduced to an infinitely small size by any optimisation procedure that changes the values of the 9 variables defining the ellipse in equation 1.

Ellipses can therefore under certain constraints be used to limit the positions of sources in brachytherapy implants. In most implants the positions of I-125 Seeds can easily be seen to lie on arcs of ellipses in space. Interstitial implant methods at present sometimes use temporary implants of I-125 Seeds in plastic tubing (Sealy et al, work in progress). Gold plaques with fixed seeds positioned usually in rows upon them (Sealy et al, 1980), (Weaver, 1986) are also used extensively for opthalmological tumours. Both these methods can easily be represented by a set of elliptical segments in space. Other implantation techniques using Ra-226, and Ir-192 sources, (Schultz, 1984), (Henschke et al, 1963), could also easily be represented by this model. This model can also be generalised to be used with any linear source type in an analogous way to the use for I-125 Seeds. It can also be simplified to ignore the orientation of a source upon the ellipse thus allowing the application to point sources in similar applicators as described

above.

The model to represent radioactive implant source positions in space as lying on ellipses is widely applicable and practicable (using conventional techniques). It has been used here to constrain source positions during optimisation of the doses at interest points in space.

3.3 IMPLEMENTATION OF THE MODEL

The elliptical model for optimisation of I-125 Seeds in space was implemented on a Sperry 1100/81 mainframe computer. The optimisation technique used was a sequential augmented Lagrangian technique (See Appendix 1). The initial values of the control parameters used in the problem allowed accurate minimisation of the subproblems and relatively weak initial constraints. This was achieved by small values of RHO and ETA. The other parameters were found to have very little influence in this model of the problem and intermediate values were chosen.

The variables in the problem were divided into two sets so as to reduce the number of variables and to allow variables of the same type to be varied simultaneously. The first set of variables was the source parameters describing the angle, from some reference point on the ellipse, at which the sources lie on the ellipse. The reference vector in space from which the parameter varied is given by substituting in the parametric equation of the ellipse the value, t = 0. These are grouped in sets of 10 variables representing 10 sources on each of 4 ellipses. The numbers of ellipses and sources were set to develop the program and can be increased. An implant using 40 seeds on 4 ellipses was assumed to be large enough to show the behaviour of the model and to be applicable to most I-125 implants. The second set was the ellipse variables that described the 4 ellipses using 9 ellipse variables each. These described the amplitudes of the ellipses in the three axes, the phase shifts relative to the other axes and, the translocation of the centre of the ellipse.

The optimisation of the positions of the sources to minimise the least squares unweighted objective function;

$$F(x) = \sum_{i=1}^{n} (AD_i - DD_i)^2$$

where, AD_i, is the achieved relative dose and, DD_i, is the desired relative dose, and n, is the number of interest points, was done by first varying one set of variables and then varying the other set of variables.

The source variables were varied by E04UAF on each ellipse in turn and then these parameters were fixed and the ellipse variables were used by E04UAF to attempt to find a minimum of F(x). The routine for calculation of dose at the interest points used the changed parameters returned by E04UAF to calculate the spatial positions of the sources and the orientation of the sources from the other parameters already set. The array used in the calculations was a composite array with 40 variables for the source parameters on 4 ellipses and 36 variables to describe the four ellipses. The relevant variables from the array were passed to E04UAF depending on whether ellipses as a group or sources on one of the ellipses were being varied in an attempt to minimise F(x). The new set of values of the variables once reset by E04UAF, were copied into the array. Then the routines for calculating the value of the objective function, and of the constraints, used the entire array. For details of the calculation see the program description in Chapter 4 and the program listing.

The constraints placed on the problem were of five types.

1. The NAG routine requires fixed bounds to be placed on all the variables. Each set of variables describes a different parameter of the problem. The source variables describe angle and were thus limited to vary in the range $\pm 4\pi$ radians, to prevent continuous cycling of these variables which may happen in a poorly constructed problem. The ellipse amplitudes were limited to vary from 0 to 10 cm. This allowed for much larger ellipses than are necessary in most situations. The phases were bound to the range $\pm\pi$ radians which allowed all geometries in space without limitation at the ends of the range as the ellipse geometry is cyclical in the phase variables with a periodicity of $\pi/2$, and most phase shifts are entered in the range 0 to $\pi/2$. The centre shift was limited to the cube $\pm$ 5 cm in all axes. This allowed a large volume of interest.

2. The constraints placed on the sources as they were varied on the ellipse limited them to remain a seed length apart so that the final result would be practicable. The distance of separation is that between the centres of the seeds. This was represented for the optimisation routine as an inequality constraint and was calculated as shown in Appendix 4.

3. The sources were limited to lie on ellipses by their intrinsic nature of being represented as parameters of an ellipse. This was not seen as a constraint by the optimisation routine. The parameter, as it varied, always lay on the ellipse represented by the ellipse variables.

4. The doses were calculated as relative doses and thus it was necessary to limit the absolute dose achieved (or absolute Time Dose Factor achieved (TDF)), at a specified point, to lie within some reasonable range. This prevents moving all the sources to the furthest possible distance to attain the most uniform relative doses at all the interest points. The absolute dose rate, in Gray per hour, was used. The range chosen was 0.2 - 0.8 Gy/hr (0.3-1.2 GyEquiv./hr) as most I-125 implants are treated within this range. This was implemented by using a range constraint function when optimising both the elliptical and source variables.

5. The ellipse shape was limited by two inequality constraints per ellipse which were active when the ellipses were being varied as a set. The first of these limited the ratio of the long to short arm of the ellipse to be greater than 20:1. This prevented excessive elongation of the ellipse as when elongation occurred the line integral calculating length of arc required more terms in its expansion to achieve the same accuracy. The second constraint prevented the total length of arc of the ellipse from getting shorter than the total length of all the seeds on the ellipse. This did not prevent overlapping of seeds and so optimisation, (or repeated optimisations) of the source positions with the inequality constraints activated was always necessary after optimisation of the ellipse variables.

The optimisation procedure was allowed to follow one of 3 options. Either the ellipse variables, or the source variables, or both, could be optimised. The objective function gave a

measure of the difference between desired doses relative to interest point 1, and the achieved doses relative to interest point 1, subject to the constraints above. As the parameters in the two cases were varied a minimum of the function was sought by the optimisation procedure. The behaviour of the method is such that a local minimum is sought (Appendix 6). The validity of not optimising alternate sets repeatedly was checked by allowing some cases to cycle through each routine 5-6 times. Improvement in the objective function value after one cycle of each routine was found to be less than 1% of the original starting value. In some cases an improvement may be achieved by cycling, but due to the function behaviour a local minimum is rapidly found and little further improvement is gained.

The timing of the procedure was dependent on two operations. Firstly the calculation of the function and constraint values by the user defined sub-routines, and secondly the optimisation itself. The first operation had a variable timing dependent upon whether it was doing calculations for sources being varied on ellipses or for ellipses being varied as a group. If sources on one ellipse were being varied then the calculation time was directly proportional to the number of sources, NS, on the ellipse, and the number of interest points, NP. If the ellipses were varied as a group then the calculation time was directly proportional to the total number of sources, NT, and NP. In the second operation the calculation time was roughly proportional to the square of the number of free variables N_{free};

where $N_{free} \leq N + MINEQ + MRNGE$

Where N= Number of variables passed to the optimisation routine, MINEQ=Number of inequality constraints,

MRNGE=Number of range constraints.

In the case of optimising sources where, $N_{free} < 2 \times NS + 1$, the timing is proportional to;

$$(4 \times NS^3 + 4 \times NS^2 + NS) \times NP$$

This becomes very large, very fast, with any increase in NS and thus only small numbers of sources on each ellipse are practical. For 1 ellipse with 6 sources and 10 interest points the time taken for the maximum allowed calculations to optimise the source positions on the ellipse once, is approximately 25 minutes of CPU time.

In the case of optimising ellipse parameters where;

$$N_{free} < 11 \times NE + 1,$$

where NE is the number of ellipses,

the timing is proportional to $(121 \times NE^2 + 22 \times NE + 1) \times NT \times NP$. This thus only allows optimisation of one or two ellipses within a practicable time.

The variables used in the optimisation of ellipses are not orthogonal. This may account in this implementation for poor convergence to a minimum, as the variables are interdependent. If these variables were made to be orthogonal by the use of polynomials, then the optimisation procedure would see orthogonal variables and may proceeded more swiftly to a minimum. Small computer programs are available for the calculation of orthogonal polynomials (Tyson et al, 1982). The use of orthogonal

polynomials would necessitate the interpretation of these polynomials by the calculation routines, which would take some extra calculation time and thus the benefit of time gained by their use may be lost by their implementation.

The model thus implemented was applied to test cases and to clinical examples (See Chapter 5). The application of the model of the clinical situation was found to be limited to certain appropriate cases.

3.4 APPLICABILITY OF THE MODEL

The elliptical constraint model being easily generalisable to many shapes and sizes could be widely used if the limitation on the number of ellipses and number of sources on the ellipses was removed. Regular shaped large volume implants requiring a large number of sources do not require accurate positioning of the sources as the positioning of sources becomes less critical with increased numbers of sources, but the uniformity improves. The mean peripheral dose and mean central dose are not sensitive to source position when the number of sources used is large, >24. (Waterman et al, 1983). This has been been verified clinically showing that doses achieved are not dependent on the exact positioning of the seeds (Rosemark et al, 1982). The use of optimisation techniques for permanent implants is not appropriate as the tumour volume changes thus altering the seeds positions. This has not influenced clinical results (Tokita et al, 1980). Optimisation for large volume regular tumours is not practical or necessary and would be time consuming. The model reduces the number of variables to a minimum and still the calculation time extends to hours with only 10 - 15 seeds depending on the type of procedure to be used.

The number of calculations, being dependent mainly on NS^3, NS is the number of sources, increases rapidly as the number of sources per ellipse increases. The timing is also directly proportional to the number of interest points in and around the tumour volume, and hence for irregular volumes with several interest points this is also a limiting factor. The model is thus best applied to

small volume tumours or tumours requiring few high activity sources so that the time used for calculation is practically short. In small irregular tumours where no standard geometric shape (such as a plane, sphere or cylinder) can be used as a model, it is difficult to give good estimates of the best source positioning using standard methods (Anderson, 1976).

In the case of temporary implants where very well defined areas need to be treated to an exactly specified dose the positioning of the seeds can be improved by the use of this technique. The model can thus best be applied to small volume tumours, close to sensitive areas, where the standard methods do not give an answer as to where to place the I-125 Seeds.

The model is thus used as follows:

1. The tumour volume is defined and critical points of interest are noted.

2. The desired dose rates or relative doses at these points are decided upon.

3. The number of sources required of the available activity is calculated (Johns et al, 1983), (Anderson, 1976)

4. A good estimate of the elliptical arcs required is made using model elliptical templates with major angles marked to allow good estimates of where on the ellipses the seeds are lying.

5. These source estimates, ellipse estimates, activities,

desired Time Dose Factor to a point, points of interest, and desired relative dose rates, are then entered during a real time run of the implementation program (See Program Description Chapter 4).

6. The optimisation procedure is done in batch mode and the results are interpreted by plotting them or by entering the new source positions into the isodose plotting routine which gives an output of an isodose plot in a plane in space as desired (See Chapter 5). Spatial and orientational coordinates of the final source positions are given and the final spacing between sources is also given.

7. The sources are then loaded into plastic tubing to best represent the graphic plot of their positions.

8. The surgeon or radiotherapist finally implants the tubes reproducing (as closely as possible) the ellipses drawn from the plotted computer printout. The source positioning is checked radiographically in the usual way and the achieved dose distribution is calculated.

This technique is limited firstly by a long calculation time, but techniques are becoming available that can cope with time consuming problems for example array processors (See Chapter 6), and secondly by the accuracy with which the sources can be placed in the tumours. It is thus best applied to easily accessible tumours allowing accurate placing or to use when gold plaque applicators are being used to hold the sources.

The model can easily be generalised to use with other isotopes and the spatial dose distribution tables can readily be created from data around any anisotropic or isotropic source. Thus with similar constraints placed upon the problem any discrete radioactive sources can be made to take up optimal positions along arcs of ellipses in space. If the same constraints hold as hold on I-125 Seeds (model 6702) then no modifications are necessary to the program. The program has been kept in a modular structure so that any changes to allow generalisation can easily be made.

The model is best used in the appropriate clinical situations (as described above) and in its place it should be very helpful in the planning of implants. It can be used with little understanding of optimisation techniques or geometric transformations. The implementation of the results is done using documented implantation techniques and is thus practicable in the clinical setting.

CHAPTER 4

PROGRAM DESCRIPTION

4.1 PROGRAM OUTLINE

The program was written in FORTRAN Programming Language (FORTRAN 11RIA) on a Sperry 1100/81 computer. The executive language used on the computer is Exec 8. The file containing the program was designated PROJ*TEST. and all development of the program was done under this file name. The program was developed to be run in both demand and batch modes. For a description of the execution and compilation of the program see Chapter 4.2.

The program was developed in a modular fashion with a fairly strict hierarchy of control (See Fig. 4.1). The executable program created made use of 4 direct access data files, 2 sequential access output files and two Numerical Algorithms Group FORTRAN Library routines (Numerical Algorithms Group, 1983), (See Appendix 5).

The program was divided into four major functional units which are controlled by TEST.CALC2 . These are:

1. Data input and modification for the use in the optimisation routine
2. Creation and update of files containing relative dose distributions around radioactive sources
3. Optimisation of the positions of radioactive sources in space
4. Output of monitoring information and results

Chapter 4 Program Description

The implementation of the model as described in Chapter 3.3 is carried out by the optimisation subroutine NAG*LIB.E04UAF. It is controlled by the subroutines .OPTPAR and .OPTELI which set up the variables required by the optimisation routine. The calculation of the value of the objective function is done by the routine .FSEV and the constraints are calculated by .CON1. The output of the results is done after each phase of the optimisation, that is each time control is returned from .E04UAF. TABLE writes the results and all other relevant information to the file NEWPF. which can then be scrutinised at a later stage.

For a description of the 32 Subroutines from TEST. used to compile the executable program see Chapter 4.3.

4.2 COMPILING AND EXECUTING THE PROGRAM

The executable element TEST.ABS2 was developed to run either in demand mode, that is interactively at a terminal, or in batch mode, that is without any interaction and gaining input from data files as necessary. This allows use by computer users and not only computer programmers.

The control file that was read by the program to determine whether to execute a demand mode run or a batch mode run was called CONT. This set a variable NSET in the control routine and if NSET was 0 it allowed keyboard input of all data requested. If NSET was 1 the control skipped all the keyboard input requests and read from the file CONT. the record numbers of records to read from DATA 4., which contained all the relevant information for initial input to the optimisation routine. A series of up to five records could thus be used in batch mode to allow optimisation of source positions. The output data was then printed from file NEWPF. and examined later. The demand run allowed stepping through all the various options and updating or initialising any records for later use. It was possible to do a demand mode optimisation, but as the complexity and timing of the problem was increased these runs became prohibitively long and most optimisations were done in batch mode.

The size of the program, as it was developed, became too large to execute outside normal hours. To allow data input and updating of files outside normal hours a copy of the data handling routines was made with minor modifications which allowed the

update of files outside normal hours so that batch runs could be started to run overnight. The executable element which did this was TEST.ABSI.

An executable element TEST.ABSSET was created to allow easy modification of the file CONT. so that if it was set to run in one or other mode then it could be reset without executing the entire batch run or demand run. This element also allowed quick changes to be made to the record numbers to be used for optimisation in batch mode.

The compilation of the executable element TEST.ABS2 was done using an extended mapping and all the elements were compiled using an 'O' option with the FORTRAN compiler. The absolute size of the executable element attained 130 kilo words, (a word on the Sperry uses 6 bytes of storage), but was not large enough to require a segmental mapping. The compilation included all elements used during development of the model and thus several elements could be excluded in a program specifically for use as a clinical tool. The absolute size of the executable element is largely due to the size of the arrays required for the optimisation routine and for the calculation of doses in space around an anisotropic source so reduction in the number of elements will not largely influence the size of the executable element. The size of the data files was set to allow up to 60 records of input data for optimisation. 5 records of data for dose calculation were available thus allowing use of different relative spatial distributions for different isotopes. The size of these files could be modified to reduce the storage

requirement of a computer executing the program.

The size of microcomputer memory is increasing rapidly and it is feasible, given that the NAG FORTRAN library may be available for compilation on microcomputers, that this program may be compiled to run on a microcomputer system. With the possibility of array processors decreasing the time of calculation it may be possible to use this model on an in house computer system with an array processor and achieve acceptable demand times in the execution of absolute elements.

For a listing of the compilation runstreams see Program Listing.

4.3 MODULE DESCRIPTION

CALC 2. This is the main program controlling flow and function of the entire executable element. Initially the file CONT. is read to determine if a batch or interactive run is to be done. This sets variables throughout the routine either reading values from the files only, or allowing keyboard input. Options are presented as menus requesting numerical input (See Fig. 4.2 for a presentation of the main menu). If records are to be modified the entire list of previously used records of data allowing execution of the optimisation routine is displayed (See Fig. 4.3). Choice of a file to be used is allowed, or if a batch run is being prepared then a list of up to 5 file records are requested, (the record numbers are stored in file CONT.) which will be executed serially in batch mode. Input of data into files is allowed using two subroutines. INELLI is used for the parameterised model data input. INPUT was used during the developmental stages for all other models attempted. The data input is stored in a direct access file DATA 4. and multiple updates of this file are allowed. The program then calls OPTIM or PAROPT which control the optimisation procedures for non-parameterised or parameterised procedures respectively. All normal exits take place from the main program.

OPENFL. This routine opens all the files for use during execution of the program. The direct access files open statements may be non-standard FORTRAN as these statements are Sperry 1100 specific and may need modification if this program is to be used on other systems. The sequential access files are rewound for re-use and

store output only until the next execution of the absolute element takes place.

INPUT. This routine reads all input from the keyboard for use during developmental runs. It allows modification of all relevant parameters controlling the execution of the routine NAG*LIB.E04UAF. It also reads data for interest point positions, source vector estimates, activities of sources and the type of source, (hence the objective function), to be used. It also allows selective changing of any data values.

INELLI. This routine reads all input from the keyboard for the optimisation of parameterised ellipses. It allows changes to the data files and presents a menu of allowed changes with the present data values displayed (See Fig. 4.4). It reads data for interest point positions and relative doses required at these points, parameter estimates of source positions, estimates of the ellipse size and shape, and source activities. A value of the TDF desired at the first point of interest is required. All angles entered from the keyboard are to be entered in degrees and are immediately converted to radians for later use.

OPTIM. This routine sets all the necessary parameters for the call of the subroutine NAG*LIB.E04UAF and calls the routine. Before calling the NAG routine it calls FUNCT1 to obtain an initial value of the objective function so as to assess the improvement achieved by the optimisation procedure. The results of the initial call and the starting values are written to NEWPF. by TABLE in a tabular form, and to MYPT., by WRITE and various

'WRITE' statements in a fully explanatory form.

PAROPT. This routine controls the various optimisation sequences allowed for optimising sources in space. It calls OPTPAR to optimise the positions of the sources on the ellipse and OPTELI to optimise the shapes and sizes of the ellipses. The parameters required by both routines for their respective calls of E04UAF are set here. The routine calls TABLE to write initial and final values of data to NEWPF.

OPTPAR. This routine sets the control parameters and variable values for a call of E04UAF to allow optimisation of the parameter values on the ellipse, and calls E04UAF.

OPTELI. This routine sets the control parameters and variable values for a call of E04UAF to allow optimisation of the ellipse variables, and then calls E04UAF.

WRITE. This routine writes the input values for developmental stage problems to the file MYPF. . It writes all the control variables to allow assessment of effect of these on the routines performance for a specific problem.

TABLE. The tabular output of this routine is written to NEWPF. and contains all the relevant information from the various stages of the optimisation procedure. The initial and final source parameters and spatial position and orientation are given. The interest points and the desired and achieved relative doses are given and the final dose rates and TDF to the first point of interest is given. The routine calculates TDF and time for

treatment assuming a relative biological effect of I-125 of 1.5 and using the equations;

$$Teq = TDF/((0.00473)x((150 \times DR)^{1.35}) \quad ....1$$

where TDF is the desired Time Dose Factor,

DR is the achieved dose rate in Gy/hr,

Teq is the Time equivalent

and $T = -(\ln(1-1.35 \times Teq \times HLAM))/(1.35 \times HLAM)$ ...2

where T is the total time of implantation,

HLAM is the decay constant of I-125 HLAM=0.00048055/hr

(Orton, 1974).

OUTPUT. The output from the developmental routines is written to MYPF. by this routine. The routine also calls TABLE which writes the final output to NEWPF. in tabular form.

AMONIT. The Monitoring routine called by E04UAF with frequency dependent on the user defined variable IPRINT. This routine monitors the progress of the minimisation procedure and gives an indication of when the convergence criteria have been achieved. The value of GLNORM, the Euclidean norm of the estimated gradient of the augmented Lagrangian in the free variables, gives an indication of convergence and should be small at the solution. The test for convergence being;

$$GLNORM/(1.+|F|) + \|D^{-0.5} r\| < XTOL$$

where F is the value of the objective function,

D is the matrix of the diagonal elements of $I+A^T A$

where A is the Jacobian of active constraints (Appendix 5),

r is the vector of residual active constraints,

XTOL is the required accuracy of the solution.

CNORM is the Euclidean norm of r and should also be small at the solution.

AMONIT calls TABLE to give a tabular output to the file NEWPF.

FUNCT1. This is the routine called by E04UAF and is used to pass control to the routines calculating the objective function. It is also used to monitor the timing by calling SU$P to prevent a system error termination because of the maximum time requested being exceeded. If the maximum time is being approached this routine resets IFLAG and allows a normal user initiated termination.

FTEST. The example function as given in the users manual. This was used as an initial test of the function of E0UAF.

FTWO. This calculates the objective function that attempts to find a uniform dose distribution by the function

$$F = (BIG - DDOS)^2 + (SMAL - DDOS)^2$$

where BIG is the maximum achieved dose at an interest point,

SMAL is the minimum achieved dose at an interest point,

and DDOS is the desired uniform dose.

An isotropic inverse square dose distribution was assumed. This routine was tested using three dimensional problems, but was poorly convergent due to discontinuities in the function.

FREE. This routine calculated the weighted least squares objective function described in Chapter 3.1 using an anisotropic dose distribution calculation and 5 variables per source. The individual operations were done in the subroutines DST3V, PHISIN

and LINPOL. This routine yielded acceptable answers, but because of the great number of variables it required a large number of calculations of the function value to achieve a solution.

FOUR. The same weighted least squares function is calculated as in FREE, but an isotropic dose distribution of the I-125 source is assumed and calculated analytically. This gave good solutions in one, two and three dimensional problems with rapid convergence.

FIVE. This routine used the same calculation methods and dose distributions as FTWO, but the objective function attempted was an unweighted least squares function. The convergence was poor because of the poor behaviour of this function.

FSIX. This only allowed a one dimensional case to check the dose calculation, and the calculations of the value for the objective function. The calculations were checked with an isotropic function and a purely inverse square function.

FSEV. This routine calculated the objective function for the optimisation of sources positioned on ellipses. The optimisation flow is such that if one ellipse is having sources varied on it the other ellipse's sources are stationary and their contribution to the dose at all the interest points is unchanged. The routine thus only calculates the dose contributions from the sources that are having their parameters changed by E04UAF. If all the ellipse parameters are being varied then all the dose contributions from all the sources are added at each interest

point. The routine directly calculates the distances and relative angles between the interest points and the sources to reduce duplication of function calls. The objective function is an unweighted least squares function.

IGRATE. Calculates the line integral along the arc of an ellipse to an angle T as described in Appendix 4.

TFORM. Calculates the length of the long and short half axes of the ellipse and the angle of transformation to convert the values of the parameters from their arbitrary zero point to zero on the long axis of the ellipse. See Appendix 3.

PARA. From the source parameter value and the values of the ellipse parameters this routine calculates the spatial and orientational coordinates of the sources. The results of this routine are written out by TABLE to allow interpretation of the parameter values. See Appendix 3.

LINPOL. This routine linearly interpolates on a two dimensional grid to obtain a function value. The grid from which it is interpolating is created by UPDAT and the axes of the grid are the inverse of the distance and the sin of phi, the relative angle between the axis of the source and the vector joining the source to the interest point. The arguments passed to LINPOL are scaled by the scaling factors calculated previously to allow direct reading of data points from the array of function values. All out of range possibilities are checked and are assumed to have an inverse square dose dependence. The data range is from 0.5 - 3.0 cm, but extrapolated data points outside this range are

also used to create the array and hence valid interpolation can take place in the range 0.1 - 5.0 cm and only outside this range is an inverse square law assumed.

DST3V. The development of the program required a spatial distance between a three dimensional interest point and a variable dimensional source vector and this program calculates such distances and returns a value of the spatial distance squared between an interest point and a source vector.

PHISIN. This routine calculates the sin of the angle phi between the unit vector in the direction of the source and the vector between the centre of the source and the interest point. It makes use of the identity that for two vectors r and u, where, p is the angle between them:

$$r \times u = \sin(p).\ \|r\|\ .\ \|u\|$$
$$= ((u.u)(r.r)-(u.r)^2)^{0.5}$$
$$= (r.r - (u.r)^2)^{0.5} \quad \text{for u a unit vector.}$$

CON1. This is the constraint routine required by E04UAF to return values for the constraint functions. The routine selects which constraints to use dependent on a variable parameter MTYP. For MTYP from 2 to 6 various constraints were attempted including, constraints limiting isotropic sources to lie on ellipses, range constraints only, constraints on the orientations only, and constraints limiting the spatial coordinates of sources to lie on ellipses and to have their orientations limited. For MTYP equal to 1 the constraint function is identical to that given in the user's manual example for E04UAF. For MTYP equal to

7 the constraints are those limiting sources to lie greater than 1 seed length apart, (See Appendix 4) and the constraint limiting the dose at the first interest point to be in the range 0.2 - 0.8 Gy/hr . (0.3 - 1.2 Gy equiv./hr). For MTYP equal to 8 the constraints limit the ratio of short to long half axes of the ellipse to be greater than 1:20. The absolute length of the short half axis is limited to be greater than that required to allow the number of sources on any one ellipse to fit onto the circumference of that ellipse. The range constraint on the dose rate to be delivered to point of interest 1 is still applied when the ellipses are being varied.

UPDAT. This routine allows input and modification of dose distribution data in file DATFL., and controls the interpolation between the data points that is done by the routine E01ACE which is called from the routine INTPOL. The function values returned are the relative doses at a point given by (1/r;sin(p)) where, p is the angle between the sources axis and the interest point at distance r. These values are multiplied by the specific dose constant of I-125. The value used was that for water (Dale, 1983). It also allows test interpolation both linearly from arrays, or using bicubic splines directly on the data. The scaling factors for the look up array are written with the array to a data file MTLK.

INTPOL. This routine sets up the variable size, two dimensional array for E01ACE and calls the routine which after having fitted bicubic splines to the data returns a function value at a desired

point.

SUBLIN. This routine allows linear interpolation from already created look up tables at any desired point around a source.

PAG. This routine uses escape functions specific for the terminal used, to clear the screen and start writing again at the top. It is used purely for ease of reading the displayed data.

CHAPTER 5

ILLUSTRATIVE EXAMPLES

5.1 SINGLE LAYER PLANAR IMPLANT

A theoretical planar implant consisting of 4 elongated ellipses (elongation factor 4:1), each containing 4 I-125 Seeds, was used to show the effect of only varying the relative positions of the seeds on fixed ellipses. The seeds lay on the straightest parts of the ellipses which were spaced 0.75 cm apart. The tumour area to be treated was taken to be 3.5 cm by 2.5 cm and 1 cm thick. This area required 235 mg.hrs Ra equivalent to achieve a dose of 10 Gy to the volume (Johns et al, 1983). For a TDF of 100 and an implant time of about 4 days, a dose rate of about 0.55 Gy/hr was desired at the specified points of interest demarcating the volume of interest. The activity of I-125 required (RBE assumed to be 1.5), was about 1980 MBq. Thus with 150 MBq seeds about 13 seeds were required for the implant. To allow a more uniform dose distribution an equal number of sources on each ellipse was desired and hence 16 sources were used. These were evenly spaced on the four ellipses (See fig. 5.1a). Four points of interest defining the volume of treatment were placed 0.5 cm above the plane of the implant and between the rows of seeds. Two other points were placed at the level of the end of the rows of seeds in the plane of the implant. The last two points were placed 0.25 cm outside the outermost ellipses in the plane of the implant. These effectively define the points at which the doses are required to be uniform. The relative doses were taken to be 1.0

at the first four points and 0.7 at the ends of the ellipses and 0.2 outside the outermost ellipses.

The optimisation procedure moved the parameter variables on each of the ellipses in turn, and completed the cycle of four ellipses twice. The objective function was reduced from 0.2658 to 0.0675 in 52.7 minutes of CPU time. The final positions of the sources are shown in fig. 5.1b. The isodose curves before and after optimisation are plotted in the plane of the interest points, 0.5 cm above the source plane. The TDF to the first interest point was 100 after 44.2 hrs at an initial dose rate of 0.65 Gy/hr (0.97 GyEquiv./hr). The total dose to the point was 28.6 Gy (42.9 GyEquiv.), (See Chapter 2). The required relative dose rates were achieved at all the interest points within $\pm$ 19%. The initial range of variation before optimisation was $\pm$ 29%. The isodose curves did not show any marked improvements in uniformity as the implant had good initial estimates and no optimisation was required in this case, although at specific points the relative desired dose rates were achieved (See fig. 5.2 for isodoses).

The limitation on the sources to lie in a plane on specific ellipses is not realistic. It is practical to prescribe exact spacing of the sources in the tubes, and this can be implemented to better than 1 mm accuracy, but to position four parallel lines of sources at exact spacing is not. The implementation of these results would entail the exact positioning of the sources in the tubes and then placing the tubes into the tissue in parallel lines at 0.75 cm separation.

5.2 ELLIPSOID IMPLANT

A theoretical implant, consisting of two circles in perpendicular planes, was used to demonstrate the effect of varying first the ellipse variables and then the source variables. The volume to be implanted was assumed to be a disc 1.0 cm thick and 2.5 cm in diameter. The sources used were 150 MBq I-125 Seeds. The number of sources required to achieve a TDF of 50, (The implant was assumed to be used as a booster to teletherapy) to a point on the periphery of the disc, at a dose rate of 0.4 GyEquiv/hr, was 8.8. To maintain symmetry 8 seeds were used, 2 on the central circular ellipse, and 6 on the outer circular ellipse (See fig. 5.3). The desired relative dose rates were chosen to be different at different points around the volume of interest. Four points around the periphery were chosen to have a relative dose rate of 1.0 relative to that at the first point of interest. Two points inside the volume required a relative dose rate of 2.0, and four points 0.5 cm from the plane of the sources required a relative dose rate of 1.5. This was to demonstrate the ability of the program to attempt to achieve various relative dose rates. The source positions and the relative dose rates achieved, before and after the optimisation, are shown in fig. 5.4. The optimisation procedure took 56.7 minutes and the objective function value improved from 180.9 to 0.041. 31766 calls to calculate the objective function were required and 865 iterations were done by the optimisation routine.

The isodoses in the plane 0.5 cm from the plane of the large circle are shown before and after optimisation in fig. 5.4. The

distribution was less uniform after optimisation although the desired doses had been achieved. By defining more interest points it is possible to force a more uniform dose distribution after optimisation, but the calculation time is increased. The time required to deliver a TDF of 50 at the first interest point was 54.1 hrs and the dose rate was 33.0 Gy/hr (50.0 GyEquiv./hr). The interest points were all within $\pm$ 16% of their desired relative doses. Before optimisation the achieved doses at two central points were 5.5 times their required relative doses.

5.3 U-SHAPED IMPLANT

A hypothetical U-shaped volume for implantation was used to demonstrate the optimisation of the sources and the ellipses to achieve desired doses at points in space. As in the planar implant (Chapter 5.1) the optimisation commenced by altering the positions of the sources on the ellipses. Then the procedure followed in the ellipsoid implant was executed (Chapter 5.2). The U-shaped volume was delimited by 12 interest points (See fig. 5.6). The total volume to be treated to a TDF of 100 was 5 cm^3. The activity required to deliver this TDF at a rate of 0.4 Gy/hr was 1040 MBq. This required seven 150 MBq Seeds. An even number of seeds would best deliver the desired relative doses at the interest points. 6 sources were used. The achieved dose rates were low and it would have been necessary to leave the implant in situ for 114.1 hrs to achieve a TDF of 50 at a dose rate of 0.20 Gy/hr (0.29 GyEquiv./hr) at the first interest point. The original estimates and the final positions of the sources are shown in fig. 5.5. A marked improvement in the dose distribution at the points of interest was achieved and the objective function decreased from 0.724 to 0.080. The deviations of the achieved doses from the desired doses before optimisation were at most 2.3 times and after optimisation were at most $\pm$ 36%. The constraints were all fulfilled and no further large source movements were necessary after the optimisation of the ellipses to prevent overlapping of the sources at the solution.

The initial estimate was good and the final result showed an improvement. The largest deviation of an achieved dose from a

desired dose was 36%. This is large and may have been due to poor initial estimates or unrealistic desired doses at the interest points. This was not a clinical example and all the desired doses and point positions were hypothetical, and hence may have caused poor conditioning of the problem. See fig. 5.6 for isodose distributions before and after optimisation.

5.4 CARCINOMA OF THE FLOOR OF MOUTH AND TONGUE

A clinical example was used to show the ease with which the method can be applied to clinical situations. The tumour to be treated was a recurrence of a previously treated carcinoma of the floor of the mouth. The size of the nodule to be given a full course of treatment to a TDF of 100 was 2 cm by 1 cm by 1 cm. The radiotherapist wanted two tubes of sources 6 cm long to be placed on either side of the nodule. The arcs described were easily approximated by two ellipses lying in parallel planes 1 cm apart (See fig. 5.7). The tumour volume was demarcated by 12 interest points. The available I-125 Seeds had an activity of 120 MBq and 6 Seeds were to be used in each tube. This gave an expected dose rate in the desired range. For isodose distribution see fig. 5.8.

The optimisation procedure followed was the same as that for the U-Shaped implant. The entire procedure required 2.07 hrs of CPU time. The optimisation program called the routine to calculate the objective function 48128 times and performed 1361 iterations. The objective function decreased from 0.6593 to 0.0259 and the maximum deviation from the desired doses decreased from 51% to 12%. The time required to achieve a desired TDF of 100 was 103 hrs at an initial dose rate of 0.35 Gy/hr (0.53 GyEquiv./hr). The total dose to the first interest point was 36.2 Gy (54.3 GyEquiv).

The implant could easily be loaded as the exact spacing of the sources is known, but because of the situation of the tumour it is difficult to achieve the exact positioning required. Thus assessment of the suitability for optimisation must be done.

CHAPTER 6

DISCUSSION

The increasingly wide use of I-125 Seeds in the treatment of a wide variety of tumours has been due to the many advantages of the use of this isotope. The biological and physical attributes of this isotope are very suitable for its use in small tumours where previously surgery would have been the treatment of choice. The results have been very good thus far (Marchese et al, 1985). With the improved knowledge of the spatial dose distribution and the improved methods for the calculation of dose in space, it has become feasible to attempt to optimise the positions of I-125 Seeds in the treatment of tumours. This has allowed the automatic calculation of the best positions for seeds with constraints on their relative position, and on absolute doses at certain desired points. Certain problems still exist in this optimisation though and these will be discussed here.

As shown in the examples of Chapter 5 the results of the optimisation procedure are strongly dependent on the initial positioning of the ellipses and the sources. The order in which the optimisation subproblems are solved also greatly influences the final result. The large number of variables required if all the parameters were to be varied simultaneously by the optimisation routine would cause the routine to take a prohibitively long time to solve the problem. The subproblems of first varying one set of parameters and then varying the other use much less time. The order in which the parameters are varied

influences the final result. The decision to vary first one or the other set is made depending on the specific clinical problem. If the ellipse positions can be varied easily in the implant then it may be chosen to use even initial spacing of the sources and to vary the ellipse variables first. This will give an easily implementable result. If the ellipse positions are fixed as on plaques then only the source variables need be varied to obtain the best dose distribution.

Anatomical limits have not yet been applied to the optimisation problem. The sequential augmented Lagrangian technique used here to implement the optimisation allows for the use of general constraint functions. These constraint functions could be applied to limiting the sources to lie within or outside some defined anatomical space. The space could be defined as a flagged volume inside which source variables may not lie. This method of defining a function requires large amounts of computation time and hence was not implemented here as the time of calculation was already at the limit of practicability in most cases, with the conventional computer facilities used.

The major difficulty encountered in implementing an optimisation procedure on the type of problem considered here is that of the definition and calculation of the objective function. The poor behaviour of the objective function seems to be an inherent problem in the definition of deviation from ideality of a generalised dose distribution around several sources in or on a tumour. By starting the optimisation procedure close to an estimated minimum (optimal solution) an improvement in the

positioning can be achieved without convergence problems (the minimum achieved is not necessarily the global minimum). An attempt was made in Chapter 5 to use good initial estimates. The number of calculations of the objective function is not always the maximum number allowed, as a minimum of the objective function is often achieved to the desired accuracy. By reducing the number of variables by use of the parametrised elliptical model, this maximum number is substantially reduced.

The calculation of the objective function is time consuming. (See Chapter 5.4 for an example taking 2.07 hrs of CPU time). The total time required for the optimisation procedure is dependent on the number of sources cubed. The calculation time however could be reduced by at least an order of magnitude by the use of array processors. This would allow implementation of much larger problems and more involved constraint calculations could be developed to include anatomical constraints. Array processors and parallel computation techniques have been applied to many large computation problems (Rodrigue, 1982). These computers and the specialised programming techniques required to operate them could be applied to many aspects of the program developed here. Fast Fourier transforms using array processors have already been used to calculate the dose distribution around radioactive sources for use in implants (Boyer et al, 1986). If an anisotropic dose distribution is to be used then interpolation techniques can be facilitated by array processors. Multiple calculations using the same function on many vectors of data simultaneously, would reduce the time dependence on the number of

sources to be proportional to the number of sources squared. The constraint functions could all be calculated simultaneously and the series required to calculate the constraint functions could be pipelined to reduce the time required for their calculation. The optimisation procedure itself is dependent on array reduction, and array manipulations are well suited to processing by parallel processors. Future developments using these relatively new techniques are thus very promising and may easily yield very acceptable calculation times for larger optimisation problems.

The large memory availability on most computers means that the size of this program is not a major limitation, requiring of the order of 1 MByte of memory space at most (that is inclusive of data files of variable size).

The problems with the description by the user of the ellipses in space could be overcome by use of a computer linked to a graphics planning system. Estimates of the positions of the sources, interest points, and the ellipses could be read directly as the output of a sonic digitiser or graphic mouse. This would allow direct input of data from computer tomography scans or from radiographs. This data could be mathematically converted to suitable estimates for the variables describing the ellipses and the sources on each ellipse. These estimates could then be used by the optimisation procedure to achieve an optimal solution of where best to position the sources. This would greatly facilitate the use of the optimisation procedure. This would

allow input of a great variety of problems which would have to be automatically scaled so that all input is well conditioned before it is used in the optimisation procedure. In its present form the user is required to judge how best to enter the problem so that the variables each lie in the range (-1,+1) at the solution point. This implies that the Hessian matrix is well conditioned at the solution, and the optimisation takes less time.

The NAG library routines are widely available and can be used on many computer systems. If a suitable hardware system was available on which the NAG routines required could be implemented, and which allowed implementation of the above mentioned improvements, a very widely useful and easily applied optimisation method could be developed. The parameterised elliptical model is flexible and could be applied to much larger and more intricate problems.

The method of positioning the sources in the tumour (or on the surface of the treated area), is by the conventional techniques used and thus no other procedures need be adapted. The clinically achieved positioning of the sources will need to be compared to the theoretically defined optimal positions to determine with what accuracy the optimised positions can be clinically applied. Assuming such clinical reproducibility in suitably selected cases this model should be a useful contribution to the accuracy with which brachytherapy treatments can be planned. An understanding of the limitations and required conditions for the use of the developed method is required for its effective application to clinical problems. It is however

even in its present form a practical method that can be used in the planning of brachytherapy treatments to patients with small volume tumours requiring accurately defined dose distributions from a small number of I-125 Seeds.

CHAPTER 7

SUMMARY

The increased use of I-125 seeds in brachytherapy has led to an increased need to accurately define the dose distribution around these sources in space. The spatial dose distributions around two types of I-125 Seeds (Models 6702 and 6711), (Medical Products Division/3M, 1982) were measured using a Geiger-Muller chamber as described in Chapter 2.2. The dose distribution of Model 6702 was found to be 10% less anisotropic than that of Model 6711. A fast and accurate interpolative method was developed to calculate the doses at points in space around I-125 Seeds. This method used linear interpolation between points in a 101 by 101 array that was created from the measured data points by fitting them to bicubic splines. A mathematical library routine, NAG*LIB.E04UAF, was used to do the fitting (Numerical Algorithms Group, 1983).

A parametrically represented elliptical model was developed to define the spatial and orientational coordinates of radioactive sources in 3-dimensional Euclidian space. The model constrained the sources to lie at points on ellipses in space and to be oriented along the tangent to the ellipse at those points. The coordinates of each source were given by a single parameter variable. Constraints were placed upon the relative positioning of the sources in space. The model was then used in the optimisation of the positions of radioactive sources in brachytherapy planning. The optimisation routine used was a sequential augmented Lagrangian technique implemented by the

Chapter 7 Summary

FORTRAN library routine NAG*LIB.E04UAF (Numerical Algorithms Group, 1983). A least squares objective function that defined the difference between the desired and achieved doses at interest points in and around the tumour volume was minimised by varying the parameters describing the source positions and the ellipses.

The model can best be applied to clinical situations where the positioning of the sources is not immediately obvious, and where accurate doses are required to be delivered at certain interest points. The calculation time is dependent on the number of variables passed to the optimisation routine, cubed. This strong time dependence on the number of variables at present allows effective use of the model only in small problems requiring few sources as otherwise the routine takes prohibitively long. With the use of array processors and graphics systems the optimisation technique could also be applied to many larger brachytherapy problems and could be widely used in the clinical setting.

Examples of the use of the optimisation routine are given in Chapter 5. The local minima that are achieved depend on the initial estimates of the source positions and the suitability of the interest points chosen to define the problems.

The model developed is easily generalised to many tumour shapes and could be used with most radioactive source types. The model allows easy application of the results of the optimisation by use of documented techniques. With the appropriate improvements this model should thus be very useful in brachytherapy planning.

APPENDIX 1

SEQUENTIAL AUGMENTED LAGRANGIAN METHOD OF OPTIMISATION

This is a sequential method of iteratively solving non-linear, equality and inequality constrained, multivariate optimisation problems (Rockafellar, 1973). It was developed from the solution to the equality constrained problem by the use of so-called multiplier methods developed by Powell (1969).

The method is applicable to problems of the form:

Minimise $f(x)$; $x \in E^n$ (x is a vector in n dimensional Euclidian Space)

Subject to $g_i(x) \geq 0$ for $i = 1,\ldots,m$

$h_j(x) = 0$ for $j = 1,\ldots,p$

where f, $g_{1..m}$, $h_{1..p}$ are real valued functions in E^n ..1

The augmented Lagrangian, $M(x,u,w,q)$, for the above problem is given by;

$$M(x,u,w,q,) = f(x) + 1/4\left(\sum_{i=1}^{m} (\{\max[u^i - 2qg_i(x),0]\}^2 - (u^i)^2)\right) + w^T h(x) + q\, h(x)^T h(x) \quad \ldots 2$$

where $x \in E^n$

$u \in (u^1,\ldots u^m)$; Lagrange multipliers for inequality constraints,

$w \in (w^1,\ldots w^p)$; Lagrange multipliers for equality constraints,

$q > 0$;is a scalar that is large enough but not so large as to make $M(x,u,w,q)$ ill conditioned, and f,g,h, are as given above.

An iterative procedure is followed where M (x,u,w,q) is minimised using a standard unconstrained minimisation technique where a local minimum for a vector, **x**, is found for each step of the procedure. The multipliers are updated from step k, as follows;

$$u^{i}_{k+1} = \max[u^{i}_{k} - 2q_k\, g_i(X_{k+1}),\ 0] \quad \text{for } i=1,\ldots m \qquad \ldots 3$$

$$w^{i}_{k+1} = w^{i}_{k} + 2\, h_j(x_{k+1})\, q_k \quad \text{for } j=1,\ldots p \qquad \ldots 4$$

$q_{k+1} > q_k$ if $\|h(x_{k+1})\|$ is not sufficiently $<$ $\|h(x_k)\|$,

or $q_{k+1} = q_k$ if $\|h(x_{k+1})\|$ is sufficiently $<$ $\|h(x_k)\|$,

by some rule such that q_k tends to infinity if the rule is applied infinitely often,

where x_{k+1}, is the local unconstrained minimiser of the problem at step k.

If x_{k+1} is sufficiently close, (this being any desired value), to a local minimiser of the constrained problem then the iterations are stopped, otherwise u_{k+1}, w_{k+1}, q_{k+1}, are inserted and k is set to k+1 and the local minimiser x_{k+2} is sought. The function, M (x,u,w,q), once u,w,q, are set, becomes a function of the vector x, the local minimum of which can be found using any quasi-Newton method. See Appendix 2 . This should be well behaved because of the choice of the Lagrange Multipliers used and should have a positive definite Hessian matrix (Appendix 2) and thus a minimum can be found. It is not a necessary requirement that an exact minimum be found at each cycle for global convergence of the constrained function. Therefore

depending on the speed at which the function value can be calculated it may be decided to use more iterations with poor unconstrained minimisation between each iteration or, very accurate minimisation with fewer iterations (Bertsekas, 1976). The quasi-Newton methods require several function evaluations to create the Hessian matrix used and to update this matrix to find the minimum. With slow calculation of the function values, it is therefore more efficient to use more iterations and less function calculations and the accuracy with which minimisations are done is thus kept low (Rockafellar, 1973)

Essentially inequality constraints are only considered, due to the updating formula for u_i's if at the points x_{k+1},

$$g_i(x_{k+1}) \leq u_i^k / 2q_k \quad \text{... from equation 3.}$$

Two sided inequalities can be considered to be two separate inequality constraints thus necessitating the addition of two slack variables to the unconstrained minimisation problem. A more efficient method is by using for the two sided problem given by

$$\min f(x)$$

$$a_j \leq g_j(x) \leq b_j$$

$$j = 1, \ldots, r$$

the following problem

$$\min f(x)$$

$$a_j \leq g_j(x) - u_j \leq b_j$$

$$u_j = 0; \qquad j = 1, \ldots, r$$

This is an equivalent problem involving additional variables $u_1,..u_r$. This problem once minimised yields the values of $u_1,..u_r$ as functions of x, the Lagrange multipliers, and the scalar penalty function. This allows direct updating of the Lagrange multipliers by the iteration

$$
\begin{aligned}
w^{j}_{k+1} &= y^{j}_{k} + q\,\phi'[g_j(x_k) - b_j] && \text{if } y^{j}_{k} + c\,\phi'[g_j(x_k) - b_j] \geq 0 \\
&= y^{j}_{k} + q\,\phi'[g_j(x_k) - a_j] && \text{if } y^{j}_{k} + c\,\phi'[g_j(x_k) - a_j] \leq 0 \\
&= 0 && \text{otherwise.}
\end{aligned}
$$

where ϕ is the penalty function used and is a function of t,

such that $\phi(t) \geq 0 \quad \forall\, t$, $\phi(t) = 0$ if and only if $t=0$

and ϕ' is the first derivative of ϕ with respect to t

(Bertsekas, 1976).

The convergence properties of this method are superior to penalty function methods and it is the best method available for problems with non-linear constraints in the absence of special structures. It is not a necessary requirement for convergence that the function be strictly convex and hence it can be applied to a great variety of functions with good results. Convergence is super-linear in uniextremal functions with less stringent assumptions placed upon the problem (Bertsekas, 1976).

APPENDIX 2

QUASI-NEWTON METHODS OF OPTIMISATION

A quasi-Newton method is any algorithm that generates a sequence of points which tend to find a local minimiser for f(x) in some open set by means of the following general equation:

$$x_{k+1} = x_k - H_k \nabla f(x_k)\, t_k \qquad k = 0,1,\ldots, \qquad \ldots.1$$

where $x \in R$, an open set in E^n space and where H_k is an approximation to the inverse Hessian (See below) at a presumed local unconstrained minimiser and t_k is a step size scalar (usually chosen using an optimal step size procedure).

This is a class of problems which use different algorithms to compute H_k. Typical quasi-Newton methods do not replace old information at every iteration but rather update the information. Some methods which use this technique are the Rank 1 Method, the Davidon-Fletcher-Powell Method, and the Broyden-Fletcher-Goldfarb-Shanno Method. There are families of updating formulae, but as a general rule the results are very much the same for all the available methods. The updating formulae are usually of the form of a function of H_k, δ_k, y_k and I.

The general class of algorithms follows an iterative procedure to estimate the Hessian matrix, H_k, and maintain the matrix in a positive definite form. A Hessian matrix is the n x n matrix of second partial derivatives of the function f such that the i, jth term at x_0, is $d^2 f(x_0)/dx_j dx_i$ assuming f is twice continuously

differentiable in the open set defined. This matrix is said to be positive definite if $Z^T HZ > 0$ for all $Z \neq 0$

Iterations are done to estimate H_{k+1} from H_k.

$$\delta_k = x_{k+1} - x_k \quad \text{and} \quad y_k = \nabla f(x_k)$$

are defined and thus by definition;

$$y_k = G_k \delta_k \qquad .....2$$

where $G_k = \int_0^1 \nabla^2 f(x_k + (x_{k+1} - x_k)t)dt.$

Thus H_{k+1} must behave in a way that G_k^{-1} does, thus satisfying

$$H_{k+1} y_k = \delta_k \qquad .....3$$

This is like trying to estimate $(\nabla^2 f)^{-1}$ using information from points x_k and x_{k+1}, if f(x) is a positive definite form. This can be accomplished by adding another matrix to H_k to satisfy 3. The different methods use different updating schemes.

Generally the following steps are used:

Iteration k + 1, k > 0 : Use the information about the inverse Hessian matrix (that is assumed to exist or to have an estimated approximate at the isolated local unconstrained minimiser that is being sought) that is gained by the difference in the gradient vector, (given above as y_k), to get H_{k+1} from an updating formula, so that H_{k+1} satisfies 3. S_{k+1} is then set: if this is a decent direction then;

$$S_{k+1} = -H_{k+1} \nabla f(x_{k+1})$$

or otherwise

$$S_{k+1} = +H_{k+1} \nabla f(x_{k+1})$$

x_{k+1} is obtained using a step-size procedure and if convergence is not achieved then a further iteration is done. With most estimating formulae, and f in a positive definite quadratic form, the methods converge to the global minimiser in, less than or equal to, n steps if an optimal step-size procedure is used.

Optimal step-size procedures estimate the optimal value of the parameter t, say t_k. The parameter, t, defines a directed curve $y_k(t)$ with the properties that $y_k(0) = x_k$, and for t positive and small, $f(y_k(t)) \leq f(x_k)$. The value t_k is estimated to give;

$$x_{k+1} = y_k(t_k).$$

The curve $y_k(t)$ is generally of the form $y_k(t) = x_k + s_k t$ where s_k is an n dimensional vector direction of search. These step-size procedures generally give the same answers if the problem is strictly convex and they all give the same rate of convergence although some are more easily computed than others (McCormick, 1983).

APPENDIX 3

PARAMETRIC REPRESENTATION AND TRANSFORMATION OF ELLIPSES IN THREE DIMENSIONAL SPACE

The equation for an ellipse in a plane parallel to the x-y plane in space can be given in Cartesian coordinates as;

$$x^2/a_1^2 + y^2/a_2^2 = 1 \quad ; \quad z = a_3 \qquad \ldots 1$$

where a_1, a_2, a_3, are constants the first two of which define the half axes lengths in the x and y directions respectively.

This equation can be given in the parameter variable t, where t is the angle of arc around the ellipse from some reference point, in this case on the x-axis, as follows:

$$\underline{r} = a_1 \sin(t)\underline{i} + a_2 \cos(t)\underline{j} + a_3\underline{k}$$

where $\underline{r}$ is the vector to a point on the ellipse given by the value of t and $\underline{i}$, $\underline{j}$, $\underline{k}$, are unit vectors in the direction of the x,y,z axes respectively (Kreyszig, 1979). This is the special case of the generalised ellipse in three dimensional space given by:

$$\underline{r} = (a_1 \sin(t+b_1) + c_1)\underline{i} + (a_2 \sin(t+b_2) + c_2)\underline{j} + (a_3 \sin(t+b_3) + c_3)\underline{k} \qquad \ldots\ldots 2$$

where a_1, a_2, a_3, represent the amplitudes of oscillation in the x,y,z directions respectively and b_1, b_2, b_3, represent the phase shifts in the x,y,z directions respectively and c_1, c_2, c_3, represent the centre shift of the ellipse in the x,y,z,

directions respectively, when

$a_3 = 0,\ b_1 = 0,\ b_2 = \pi/2\ ,\ c_1 = 0,\ c_2 = 0.$

It is necessary to transform the generalised parametric equation to the standard form of the ellipse in the x-y plane so that calculations to determine arc lengths and positions on the ellipse are simplified. Only the lengths of the half axes in the x and y directions are required for the transformation. The angle t_m at which the rate of change of the length of the vector $\underline{r}'$, is zero will give one half axis ($\underline{r}_m$) where $\underline{r}' = \underline{r} - \underline{c}$ and $\underline{c}$ is the vector to the centre of the ellipse from the origin. The other half axis ($\underline{r}_n$) will be given by $t_m + \pi/2$. The lengths of these half axes can then be used to give the required equation. The value of t_m is required to transform the points on the generalised ellipse given by $t_1, t_2, ..t_l$ to the points on the standardised ellipse by the following transformation;

$$t'_i = t_i + t_m \quad (i=1, ...l)$$

where there are l points and t'_i is the point on the standardised ellipse corresponding to the point t_i on the generalised ellipse.. t_m is thus the relative phase shift in space from the one set of coordinate axes to the other. Therefore with;

$$\underline{r} = \underline{r}' + \underline{c} \quad .....3$$

where $\underline{c} = c_1 \underline{i} + c_2 \underline{j} + c_3 \underline{k}$

the length of $\underline{r}'$ will be given by $f(\underline{r}') = (\underline{r}'.\underline{r}')^{0.5}$ ...4

and $df(\underline{r}')/dt = 0$ at the extrema.

Thus from 2, 3 and 4

$$f(\underline{r}') = (a_1 \sin(t+b_1))^2 + (a_2 \sin(t+b_2))^2 + (a_3 \sin(t+b_3))^2)^{0.5}$$

and

$$df(\underline{r}')/dt = (a_1^2 \sin(t+b_1)\cos(t+b_1) + a_2^2 \sin(t+b_2)\cos(t+b_2) + a_3^2 \sin(t+b_3)\cos(t+b_3))/f(\underline{r}')$$

therefore at extrema

$$a_1^2 \sin 2(t+b_1) + a_2^2 \sin 2(t+b_2) + a_3^2 \sin 2(t+b_3) = 0$$

Separating out the variable t the required condition for an extremum is;

$$M\sin 2t + N\cos 2t = 0 \quad \ldots 5$$

$$\text{where } M = a_1^2 \cos 2b_1 + a_2^2 \cos 2b_2 + a_3^2 \cos 2b_3$$

$$\text{and } N = a_1^2 \sin 2b_1 + a_2^2 \sin 2b_2 + a_3^2 \sin 2b_3$$

This simplifies to $(M^2 + N^2)^{0.5} .\sin(2t \pm d) = 0$ where $d=\text{atan}(\pm N/M)$

therefore, as $M^2 + N^2 = 0$ only in the trivial case,

$$2t_m + d = 0 \quad \text{or} \quad t_m = -0.5 \text{ atan}(N/M) \qquad \text{(Kreyszig, 1979)}$$

Thus having t_m, the vectors in the directions of the 2 half axes, $\underline{r}_m$ and $\underline{r}_n$, and their lengths, $|\underline{r}_m|$ and $|\underline{r}_n|$, can be calculated as follows;

$$\underline{r}_m = (a_1 \sin(t_m + b_1))\underline{i} + (a_2 \sin(t_m + b_2))\underline{j} + (a_3 \sin(t_m + b_3))\underline{k}$$

$$\underline{r}_n = a_1 \sin(t_n + b_1 + \pi/2)\underline{i} + a_2 \sin(t_n + b_2 + \pi/2)\underline{j} + a_3 \sin(t_n + b_3 + \pi/2)\underline{k}$$

Let the constant for the unit vectors in the vector $\underline{r}_m$ be given by

$$A_i = a_i \sin(t_m + b_i) \qquad (i=1,2,3)$$

and in the vector r_n be

$$B_i = a_i \sin(t_n + \pi/2 + b_i) \qquad (i=1,2,3)$$

thus

$$|\underset{m}{r}| = (A_1^2 + A_2^2 + A_3^2)^{0.5} = K$$

and

$$|\underset{n}{r}| = (B_1^2 + B_2^2 + B_3^2)^{0.5} = L$$

and finally if $K \geq L$

$$\underline{v} = K\sin(t+t_m)\underline{i} + L\sin(t+t_m + \pi/2)\underline{j}$$

is the parameterised equation of an ellipse with the long half axis in the x direction and the short half axis in the y direction.

If $K < L$ then;

$$\underline{v} = L\sin(t+t_m - \pi/2)\underline{i} + K\sin(t+t_m)\underline{j}$$

is the required equation in space.

From equation 2 the Cartesian coordinates of the n points given by $t_1, \ldots t_n$ in space are simply;

$$x_i = a_{1\,1}\sin(t_i + b_{1\,1}) + c_1$$

$$y_i = a_{2\,1}\sin(t_i + b_{2\,1}) + c_2$$

$$z_i = a_{3\,1}\sin(t_i + b_{3\,1}) + c_3 \qquad \text{for } i=1,\ldots n.$$

The orientation of the tangent to the ellipse can be described by two angles in spherical coordinates about the point x_i, y_i, z_i.

The tangent at the point given by t_i is

$$\underline{s} = (a_{1\,1}\cos(t_i + b_{1\,1}))\underline{i} + (a_{2\,1}\cos(t_i + b_{2\,1}))\underline{j} + (a_{3\,1}\cos(t_i + b_{3\,1}))\underline{k}$$

Let $s_{ij} = (a_{i\,1}\cos(t_i + b_{i\,1}))$; $j=1,2,3$; s_{ij} constant for each

$$\therefore \underline{s}_i = s_{1i}\underline{i} + s_{2i}\underline{j} + s_{3i}\underline{k}$$

thus

$$|\underline{r}_m| = (A_1^2 + A_2^2 + A_3^2)^{0.5} = K$$

and $$|\underline{r}_n| = (B_1^2 + B_2^2 + B_3^2)^{0.5} = L$$

and finally if $K \geq L$

$$\underline{v} = K\sin(t + t_m)\underline{i} + L\sin(t + t_m + \pi/2)\underline{j}$$

is the parameterised equation of an ellipse with the long half axis in the x direction and the short half axis in the y direction.

If $K < L$ then;

$$\underline{v} = L\sin(t + t_m - \pi/2)\underline{i} + K\sin(t + t_m)\underline{j}$$

is the required equation in space.

From equation 2 the Cartesian coordinates of the n points given by $t_1, \ldots t_n$ in space are simply;

$$x_i = a_{i1}\sin(t_1 + b_{i1}) + c_1$$

$$y_i = a_{i2}\sin(t_2 + b_{i2}) + c_2$$

$$z_i = a_{i3}\sin(t_3 + b_{i3}) + c_3 \qquad \text{for } i=1,\ldots n.$$

The orientation of the tangent to the ellipse can be described by two angles in spherical coordinates about the point x_i, y_i, z_i.
The tangent at the point given by t_i is

$$\underline{s} = (a_1\cos(t_i + b_{i1}))\underline{i} + (a_2\cos(t_i + b_{i2}))\underline{j} + (a_3\cos(t_i + b_{i3}))\underline{k}$$

Let $s_{ij} = (a_i\cos(t_i + b_{ij}))$; $j=1,2,3$; s_{ij} constant for each

$$\therefore \underline{s}_i = s_{1i}\underline{i} + s_{2i}\underline{j} + s_{3i}\underline{k}$$

$$|\underline{s}_i| = (s_{1i}^2 + s_{2i}^2 + s_{3i}^2)^{0.5}$$

The unit vector in the direction of the tangent is thus

$$\underline{u} = (s_{1i}/|\underline{s}_i|)\underline{i} + (s_{2i}/|\underline{s}_i|)\underline{j} + (s_{3i}/|\underline{s}_i|)\underline{k}$$

Thus the spherical angles θ and ϕ can be given by

$$\theta_i = \arccos(s_{3i}/|s_i|)$$

$$\phi_i = \arcsin(s_{2i}/\sin(\theta_i)$$

Thus a five dimensional vector $(x_i, y_i, z_i, \theta_i, \phi_i)$ is defined, giving the position of a point on a generalised ellipse (defined by t_i) as x_i, y_i, z_i, and the orientation of the tangent at that point as θ_i and ϕ_i.

APPENDIX 4

THE LENGTH OF ARC OF AN ELLIPSE

The length of arc of a parameterised ellipse in the x - y plane given by $\underline{r} = (a_1 \sin t)\underline{i} + (a_1 \cos t)\underline{j}$ 1

from t1 to t2 is given by the integral

$$s = \int_{t1}^{t2} (\underline{r}.\underline{r})^{0.5} \, dt \qquad .....2$$

(Kreyszig, 1979 pp 377)

$$= \int_{0}^{t2} (\underline{r}.\underline{r})^{0.5} \, dt - \int_{0}^{t1} (\underline{r}.\underline{r})^{0.5} \, dt$$

To find the distance between points on the ellipse given by parameters ti it is necessary only to solve

$$s = \int_{0}^{ti} (\underline{r}.\underline{r})^{0.5} \, dt \qquad .....3$$

for each point ti and the differences follow.

from 1 and 3 we have

$$s = \int_{0}^{ti} (a_1^2 \sin^2 t + a_2^2 \cos^2 t)^{0.5} \, dt$$

$$= a_1 \int_{0}^{ti} (1 - e^2 \cos^2 t)^{0.5} \, dt \quad \text{where } e^2 = (a_1^2 - a_2^2)/a_1^2$$

(Dolp et al, 1960)

Using the substitution $z = e^2 \cos^2 t$ we can expand

$f(t) = (1 - e^2 \cos^2 t)^{0.5}$ by a MacLaurin series around zero

to give $f(z) = (1 - z)^{0.5}$

$$= \sum_{n=0}^{N} f^{(n)}(0).z^n/n! \quad \text{(Kreyszig, 1979 pp 694)}$$

where $f^{(n)}(0)$ is the nth derivative of f(z) evaluated at 0,

and where the sum is to N terms.

Resubstituting $e^2 \cos^2 t = z$ we get

$$s = a\int_0^{ti} (1 - (e^2\cos^2 t)/2 - (e^4\cos^4 t)/2!4 - (e^6\cos^6 t)3/3!8....)\,dt$$

This can be calculated to N terms with a remainder R(z) which is a measure of the error on the final result.

The range for which the expansion is convergent is $|z| \leq 1$ which is uniquely determined for $0 \leq t \leq \pi/2$

Thus s can be calculated over the range of t given, and multiples of the length of the one quadrant must be added to any values of t outside the range $0 \leq t \leq \pi/2$. The integral is solved by means of the sum of integrals in $\cos^2 t$. To solve the integral of $(\cos^2 t)^n$ another expansion is used.

$$\int_0^t \cos^{2n} t\,dt = [(\sin(t))(\cos^{n-1}(t))]_0^t + (n-1)/n \int_0^t \cos^{2n-2} t\,dt$$

Thus in the first quadrant of the ellipse the calculation is

$$s = a\,[\sum_{n=1}^{N} (f^{(n)}(0)/n!).[\int_0^t \cos^{2n} t\,dt]]$$

where N is the number of terms in the Maclaurin expansion to be used. Thus using the value of s for 1 quadrant, that is with $t = \pi/2$ the values of s can be calculated in other quadrants as follows;

In odd numbered quadrants that is $\pi.m \leq t < (\pi.m) + \pi/2$

where m is any integer

$s_t = (n-1)s_1 + s_{t^*}$ where s_t is the length of arc to angle t, s_1 is the length of 1 quadrant and s_{t^*} is the length from 0 to t*, where $t^* = t\ (n-1).\ \pi/2$ and n is the

number of the quadrant and is odd.

In even numbered quadrants that is $\pi/2.(2m-1) \leq t < \pi.m$ where

m is any integer

$s_t = n.s_1 - s_{t^*}$ where the symbols have the same meaning except that $t^* = n.\pi - t$

and n is the number of the quadrant and is even.

Then finally the length of arc, s_{12}, between 2 angles, t1, and t2, is given by;

$s_{12} = s_{t1} - s_{t2}$ where $t1 > t2$ and s_{t1} is the length from zero to angle t1,

and s_2 is the length from zero to angle t2.

APPENDIX 5

NUMERICAL ALGORITHMS GROUP FORTRAN LIBRARY ROUTINES

These are algorithms written in the programming language FORTRAN from a library that is available from the Numerical Algorithms Group (Numerical Algorithms Group, 1983). They are implemented at the University of Cape Town on a Sperry 1100/81 mainframe computer. The use of the subroutines is described in user's manuals which are extensive and detailed.

The subroutines called during the development and implementation of this work were, NAG*LIB.E01ACE, NAG*LIB.E02CAF, NAG*LIB.E02CBF, NAG*LIB.E04UAF.

NAG*LIB.E01ACE, is a routine to give function values at a desired point given by two variables. This is done by fitting bicubic splines to data points in the two variables. The data points must be regularly spaced and at strictly increasing intervals in both variables. The cubic spline is fitted first in one variable and then a cubic spline is fitted through these splines in the other variable to obtain desired function values. This process is repeated starting with the second variable first and then fitting a spline through these fitted values. This then gives the user the opportunity to check that the fitting is consistent in both axes and gives the same function value irrespective of the order of fitting. The values returned are reliable only within the range of the data values and an error message is returned if the value required lies outside the range of the data values. (A single call, within a subroutine, of E01ACE takes

approximately 20×10^{-3} s).

NAG*LIB.E02CAF, forms an approximation to the weighted least-squares Chebyshev series surface fit to data arbitrarily distributed on lines parallel to one independent coordinate axis. It determines a bivariate polynomial approximation of user defined degrees in both variables to a set of weighted data points. The series is represented in double Chebyshev series form with arguments XCAP and YCAP, related to the original variables X and Y by the transformation.

$$XCAP = (2X-(XMAX+XMIN))/(XMAX-XMIN)$$

$$YCAP = (2Y-(YMAX+YMIN))/(YMAX-YMIN)$$

and lying in the range -1 to +1. (XMAX and XMIN, and YMAX and YMIN, are the maxima and minimal desired in the X and Y axes). The double Chebyshev series can be written as

$$F(X,Y) = \sum_{i=0}^{K} \sum_{j=0}^{L} A_{ij} T_i(XCAP) T_j(YCAP)$$

where $T_i(XCAP)$ is the Chebyshev polynomial of the first kind of degree i up to a maximum of degree K and T_j is the Chebyshev polynomial of the first kind of degree j up to a maximum of degree L, and A_{ij} is the ij th coefficient.

The fit of the polynomial surface is poor at the edges and is unreliable if K and L are too large for the number of data points supplied in both axes. The calculation of fitted values using the coefficients supplied can be done using the subroutine E02CBF. This allows easy and efficient calculation of function values with the fitted polynomials.

NAG*LIB.E04UAF, attempts to find a minimum of a function of several variables subject to fixed bounds on the variables and to general inequality and/or equality constraints. A sequential augmented Lagrangian method (Appendix 1) is used, the minimisation subproblems being solved by a quasi-Newton method (Appendix 2). No derivatives are required although the functions should be continuous with continuous first and second derivatives although it will usually work even with occasional discontinuities. It is applicable to problems of the form, (using the NAG manuals variable names and conventions),

$$\text{minimise} \quad F(X_1 \ldots\ldots X_N)$$

subject to fixed bounds

$$L_J \leq X_J \leq U_J \qquad (J= 1,2 \ldots\ldots N),$$

equality constraints

$$C_R(X_1 \ldots\ldots X_N) = 0 \qquad (R= 1,2, \ldots\ldots MEQ),$$

inequality constraints

$$C_{MEQ+S}(X_1 \ldots\ldots X_N) \geq 0 \qquad (S= 1,2, \ldots\ldots MINEQ),$$

and/or range constraints

$$LB_T \leq C_{MEQ+MINEQ+T}(X_1 \ldots X_N) \leq UB_T \qquad (T= 1, \ldots MRNGE)$$

where

(MEQ, MINEQ, MRNGE $\geq$ 0 and M = MEQ + MINEQ + MRNGE $\geq$ 1),

and UB_T and LB_T are the upper and lower bounds of the Tth range constraints respectively, and U_J an L_J are the fixed upper and lower bounds of the variable X_J.

The user is required to supply subroutines FUNCT1, to supply the values of the function at any point X, CON1, to supply the value of the constraint at any point X, and AMONIT, to monitor the

progress of the minimisation. The solution is found as follows: slack variables are added to convert inequality to equality constraints, the augmented Lagrangian is constructed using FUNCT1 and CON1 and the penalty scaler RHO, and estimates of the Lagrange Multipliers. This is minimised by a call of E04JBF by E04UAF subject to the fixed bounds on the variables. This minimisation using a quasi-Newton method is done to an accuracy of ETA, and using an initial step-size for the step-size procedure of STEPMX. An iteration step follows completion of the minimisation and better estimates are obtained of the Lagrange Multipliers and the value of RHO. Then follows a sequence of iterations and minimisations until termination conditions are met and the values of GLNORM (the Euclidean norm of the gradient of the augmented Lagrangian) and CNORM (the Euclidean norm of the residual of the active constraints) are small. The precise text for convergence to the minimum is

$$GLNORM/(1.+|F|) + \|D^{-.5} r\| < XTOL$$

where D is a diagonal matrix whose elements are the diagonal elements of $(I + A^T A)$ where A is the Jacobian, that is the determinant of the matrix of first derivatives, of active constraints, |F| = the absolute function value, r is the residual of active constraints, and XTOL is a preset desired value prescribing the accuracy required.

Suggested values and conditioning of the problem are given in the manual. The timing of the routine depends on the behaviour of FUNCT1 and CON1. The number of calls of these routines is roughly proportional to N_{FREE}^2, $(N_{FREE} \leq N + MINEQ + MRNGE)$, where

N_{FREE} is the number of variables of the augmented Lagrangian not on their bounds, plus each iteration makes $N_Z + 1$ calls of FUNCT1 and CON1 where N_Z is the number of variables not on their bounds ($0 \leq N_Z \leq N$). The time taken by the routine is dominated by time spent in FUNCT1 and CON1.

The routines are portable to other computer systems as they are written in FORTRAN, but they are large and compilation of the absolute elements to be executed may exceed generally available computers' memory space. It is thus necessary to run these routines on large computer systems. They are stable routines and are efficient in their computation with most time being used by the user supplied programs.

APPENDIX 6

THE COMPLEXITY AND EFFICIENCY OF THE OPTIMISATION PROBLEM

The efficiency of an optimisation method applied to a certain class of problems can be characterised by its laboriousness and error, or rather, by the upper bounds (over the problems of a class) for the number of steps in its work on the problem and by the error of the results. The complexity of a class of problems can be given as a function of the error. Thus methods which achieve the estimated potential lower bound for laboriousness, which is the complexity of a class of problems, are the least laborious for that class.

The class from which the problem; min F(X)

$$\text{where } F(X) = \sum_{j=1}^{n} (g_j(X) - c_j)^2$$

$$\text{where } g_j(X) = \sum_{i=1}^{m} (d_i / (X_i - b_j)^2)$$

and c_i, d_i, b_i, are constants and X is an array m x p where p is the dimension of the space being used and m is the number of points described in that space with X_i the ith point, is derived is known as the multiextremal, non smooth, non convex problems. The problem can be seen to be smooth except for occasional asymptotic points. The lower bound of the function giving the complexity of the problem behaves catastrophically when the relative error tends to zero or when the number of variables tends to infinity. This behaviour is so marked that solutions of problems of this class with only 3 extrema cannot be generally obtained within acceptable bounds on laboriousness (Nemirovsky et

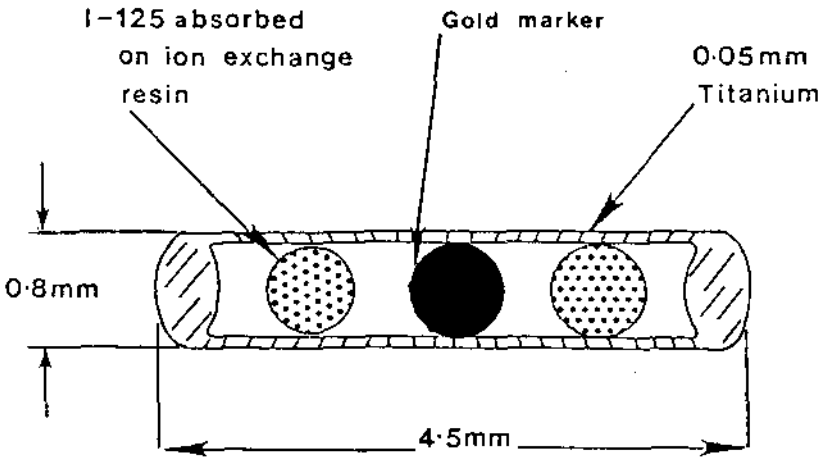
al, 1983).

The complexity of smooth, uniextremal, convex problems with constraints and dimensionality greater than 3 can thus be considered. Dependent on the degree of smoothness of the class of functions the upper and lower bounds to the complexity of these problems can be defined. The Quasi-Newton methods of optimisation when applied to a strongly convex problem do not give with certainty a good solution to the problem. They are dependent on the scale of the specific problem and thus if the problem is well scaled then a good solution that may approach the optimal bound on complexity can be obtained. If Quasi-Newton methods are applied to variable general functions in the same class of strongly convex problem then the method used may become excessively laborious due to this sensitivity to scale. Everything is not yet known about the least laborious method of finding the minimum of a strongly convex problem. Quasi-Newton methods are good and if well constructed yield acceptably laborious results. The solution of poorly convex, or non convex problems is much more laborious.

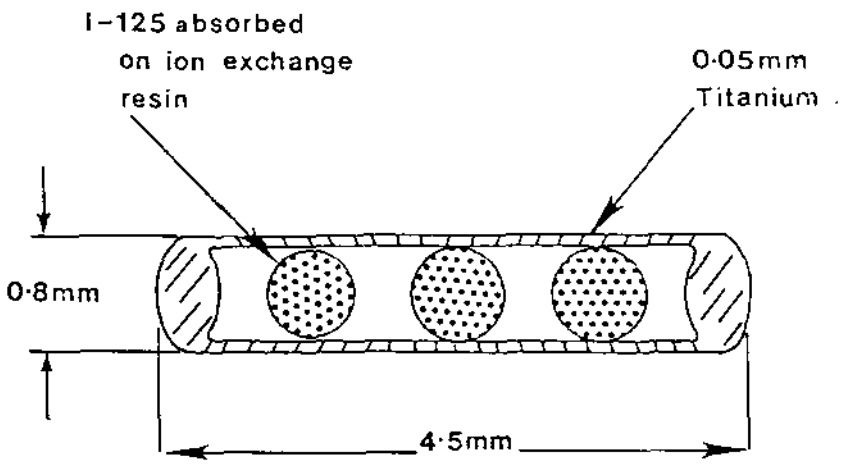
In general therefore optimisation of problems of the type described in Chapter 3 is mathematically laborious and it is meaningless to pose the question of constructing universal methods of solving such problems. This would also hold for uniextremal, but non convex problems, thus the problem's generality is severely limited (Nemirovsky et al, 1983).

The complexity of the general problem is thus excessive and the

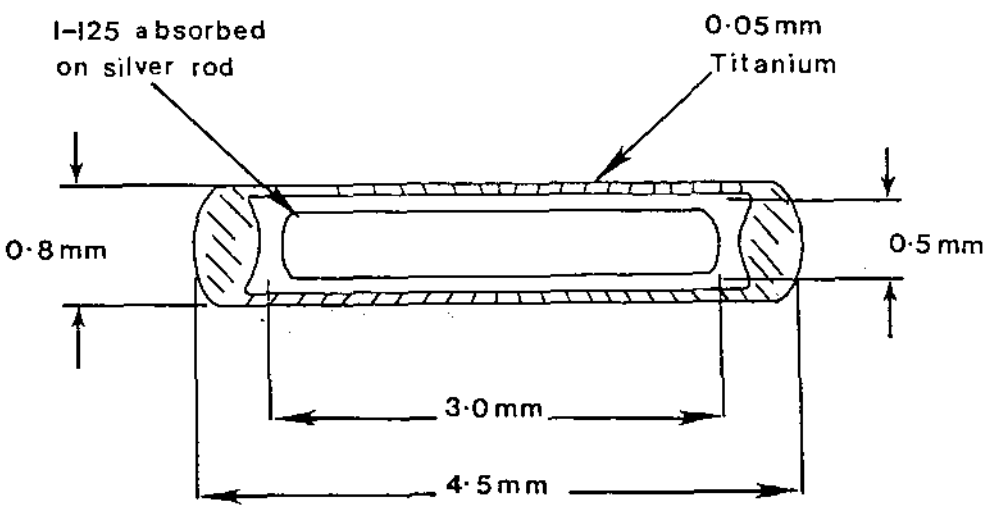
problem is limited to finding the nearest extremum from a "reasonable" guess at the correct answer. If in the region from the "reasonable" guess, say X_R to the extremum X_E the function is smooth and convex then in general a solution can be found.



MODEL 6701



MODEL 6702



MODEL 6711

FIGURE 2.1 I-125 ENCAPSULATION DESIGN

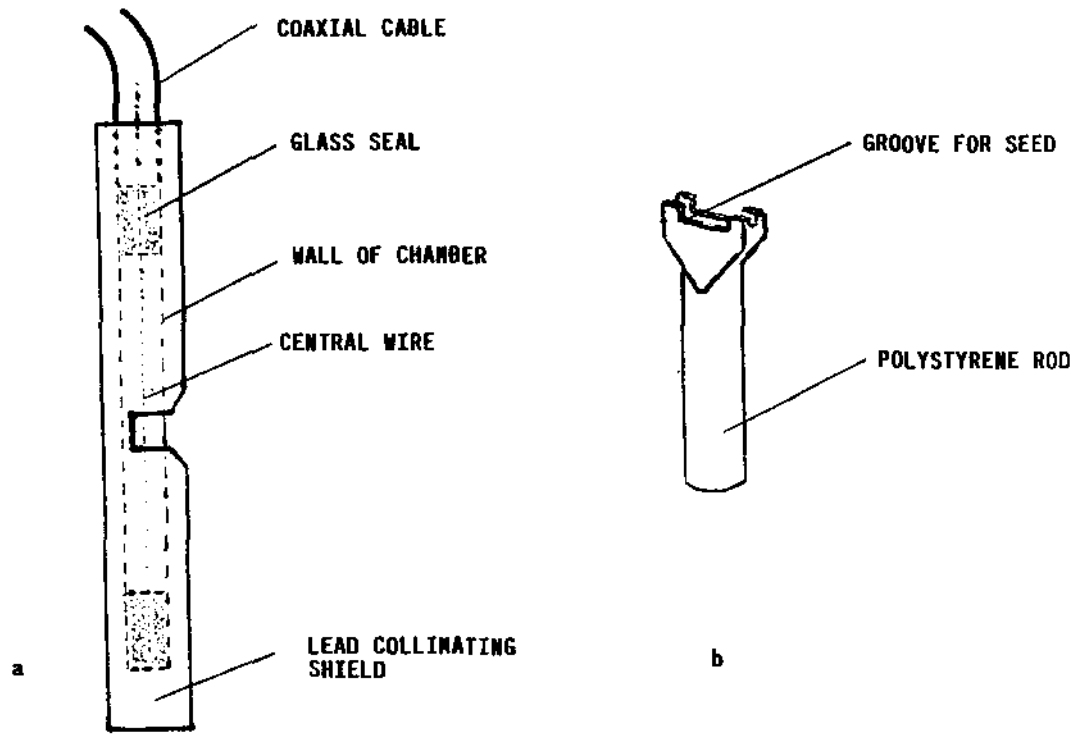


FIGURE 2.2 a COLLIMATED GEIGER MULLER CHAMBER SHOWING COMPONENTS. SCALE 4:1

b POLYSTYRENE TIP TO HOLD THE I-125 WITH MINIMAL INTER-POSITION OF POLYSTYRENE BETWEEN THE SOURCE AND THE DETECTOR. SCALE 2:1

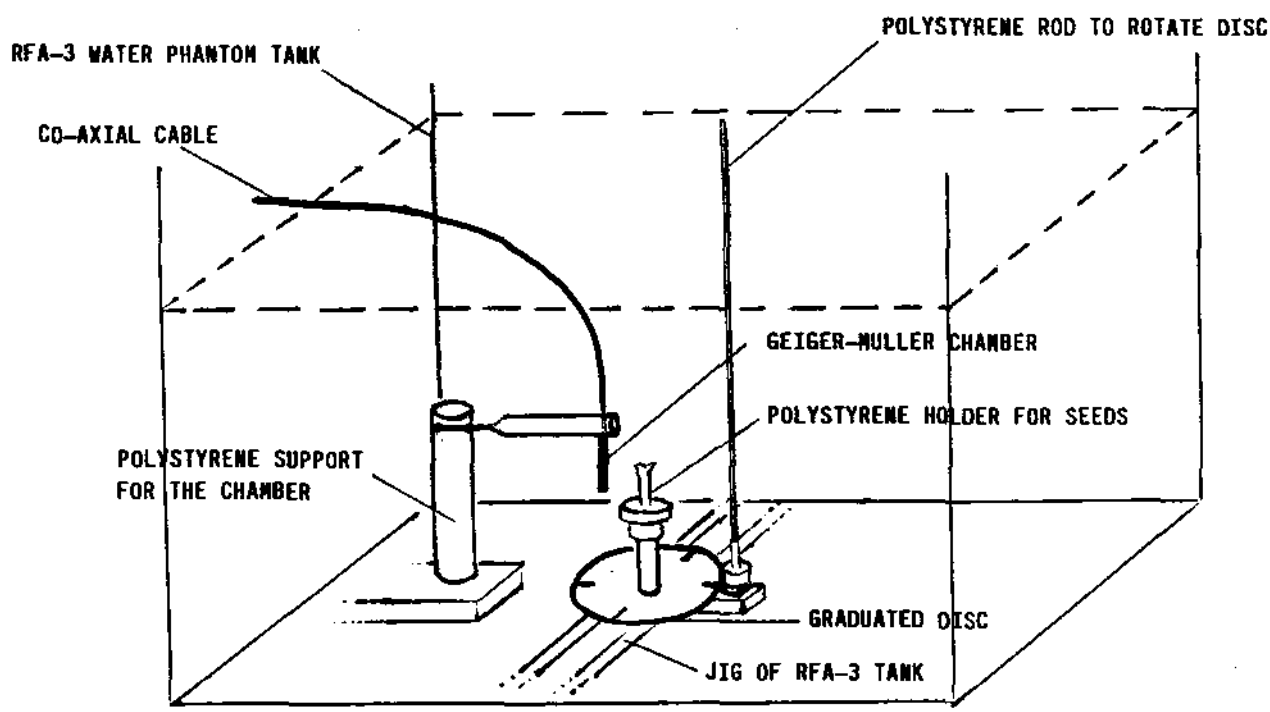
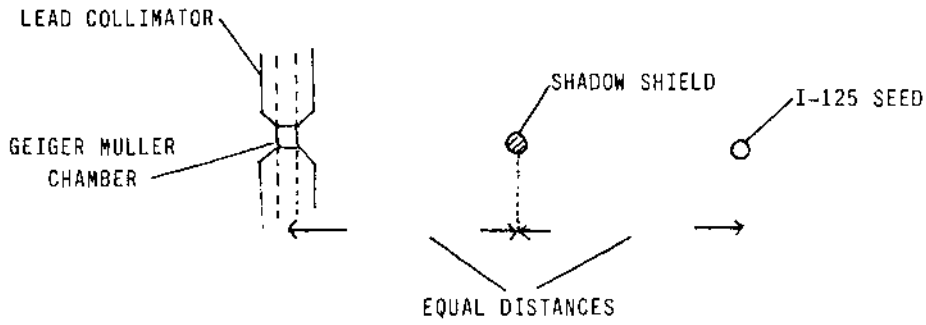
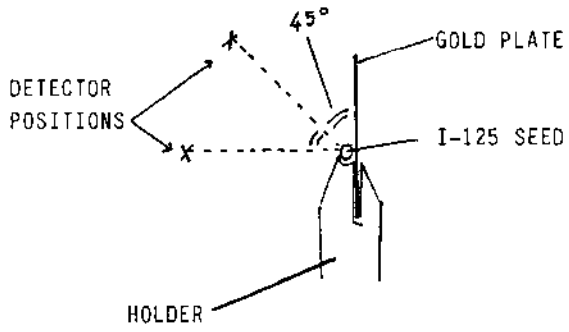
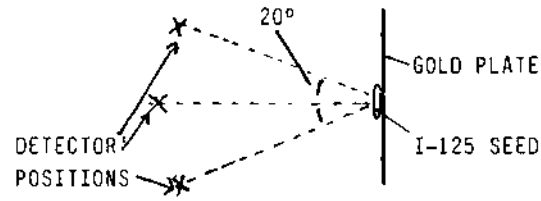


FIGURE 2.3 PERSPECTIVE DIAGRAM OF THE EXPERIMENTAL SET-UP USED IN THE MEASUREMENT OF THE RELATIVE SPATIAL DOSE DISTRIBUTION AROUND I-125 SEEDS.





SIDE VIEW



TOP VIEW

2.3.b TOP - EXPERIMENTAL ARRANGEMENT FOR MEASUREMENT OF SCATTER CONTRIBUTION

BOTTOM - ANGLES OF MEASUREMENT WITH AND WITHOUT THE GOLD PLATE IN PLACE

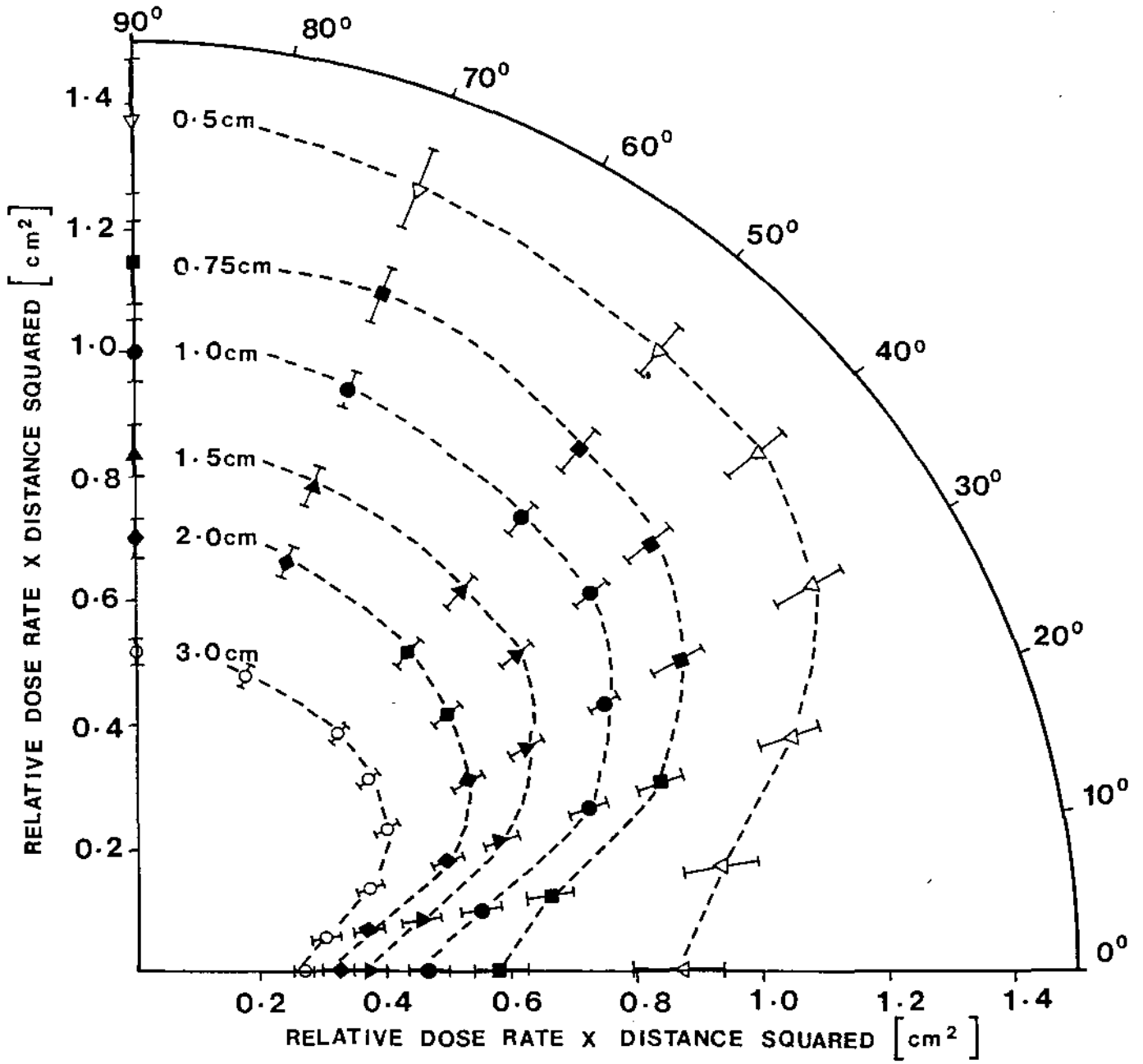
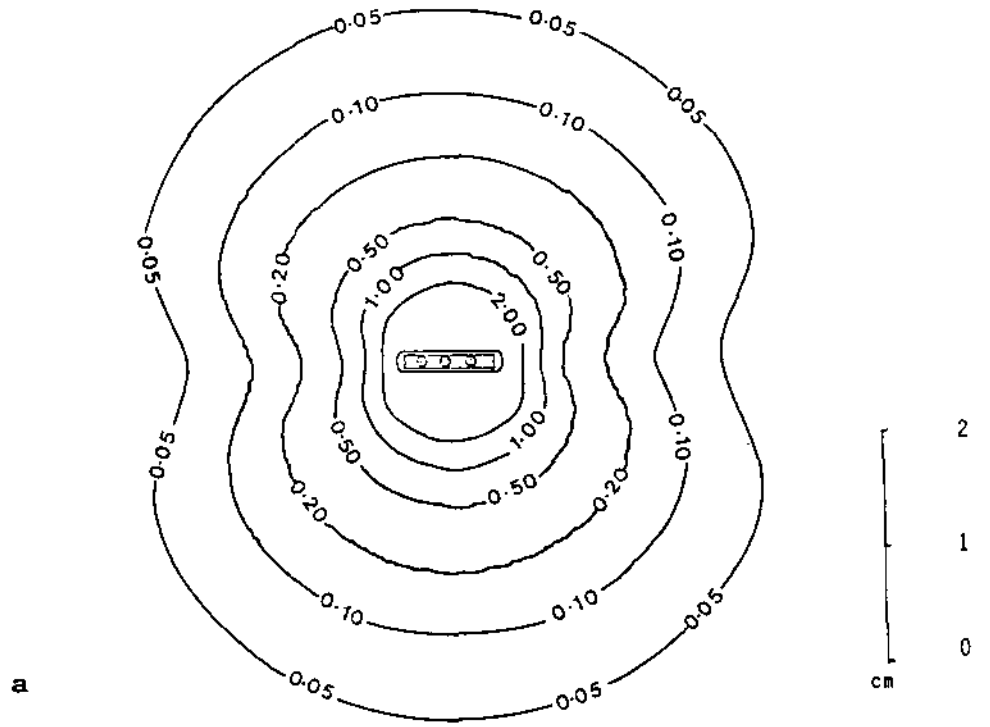
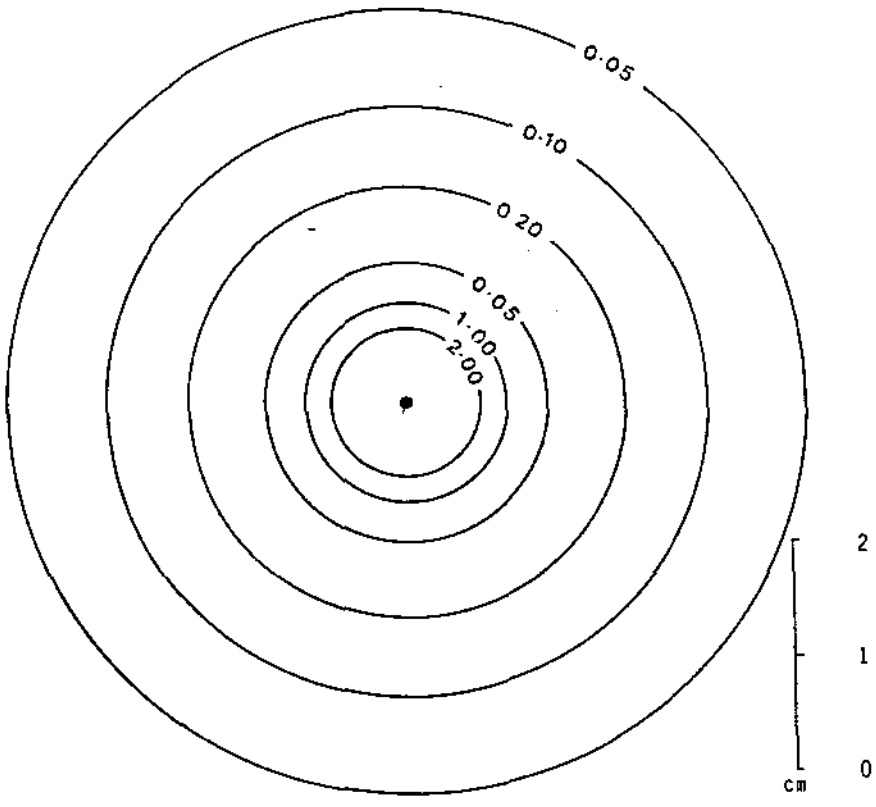


FIGURE 2.4 EXPERIMENTAL RESULTS SHOWING THE RELATIVE SPATIAL DOSE DISTRIBUTION IN ONE QUADRANT OF AN I-125 SEED MODEL 6702. THE DATA POINTS GIVE THE RELATIVE DOSE RATE X DISTANCE SQUARED AT THE DISTANCES SHOWN, AND ARE NORMALISED TO THE VALUE AT 90° AND 1 CM. THE STANDARD ERROR ON MEAN IS SHOWN.



a

I-125 SEED 1000MBQ ON X-AXIS



b

I-125 POINT 1000 MBQ

FIGURE 2.5 a. THE ISODOSE DISTRIBUTION OF AN I-125 SEED MODEL 6702 AS INTERPOLATED FROM THE MEASURED DATA USING THE LOOK UP TABLE DESCRIBED IN CHAPTER 2.4

b. THE ISODOSE DISTRIBUTION OBTAINED USING A POINT SOURCE OF I-125 AND CALCULATED USING THE 3RD DEGREE POLYNOMIAL VALUES CALCULATED BY DALE (DALE 1982)

ISODOSES GIVEN IN Gy/hr

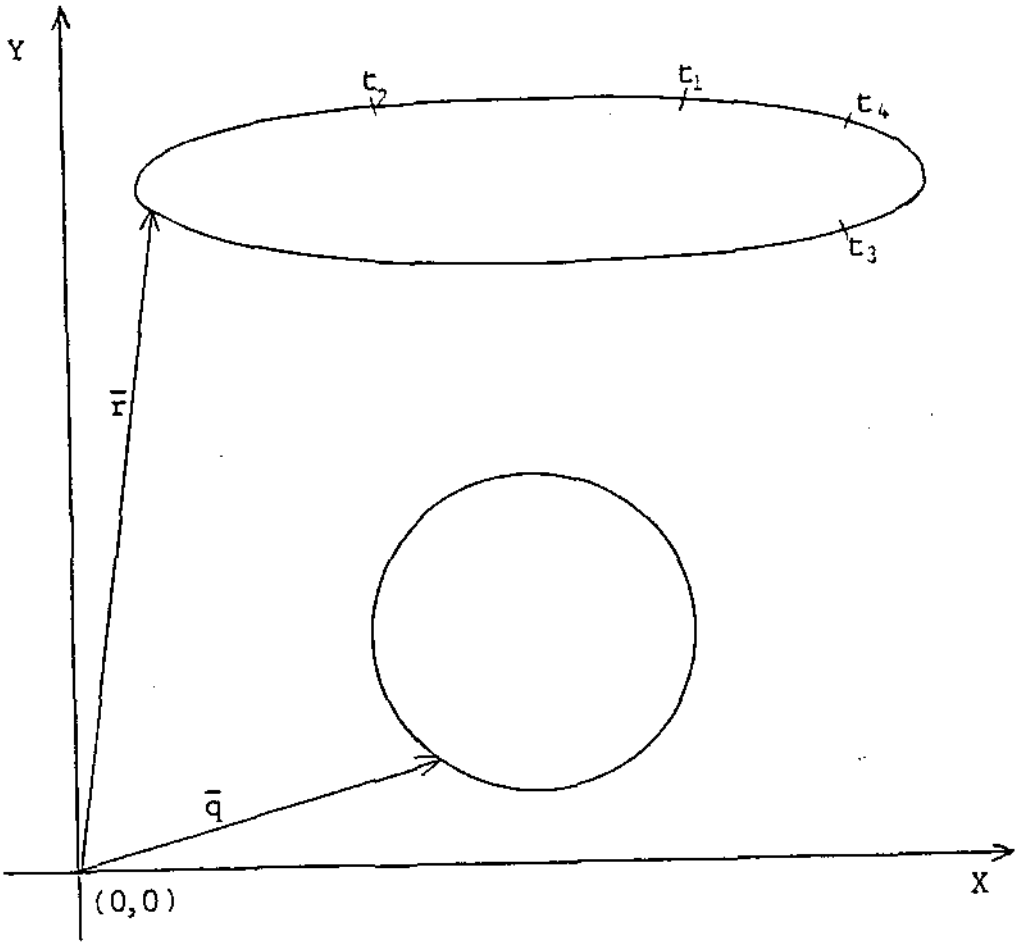


FIGURE 3.1 TWO ELLIPSES IN THE X Y PLANE DESCRIBED BY;

$$\bar{r}(t) = (5\cos t + 6)\bar{i} + (\sin t + 9)\bar{j} \; ;$$
$$0° \leq t \leq 360°$$

and

$$\bar{q}(t) = (2\cos t + 6)\bar{i} + (2\sin t + 3)\bar{j} \; ;$$
$$0° \leq t \leq 360°$$

THE ARC DESCRIBED FROM t_1 TO t_2 ON $\bar{r}(t)$ IS APPROXIMATELY A STRAIGHT LINE SEGMENT, 4cm LONG, AND THE ARC DESCRIBED FROM t_3 TO t_4 IS APPROXIMATELY A PARABOLIC ARC.

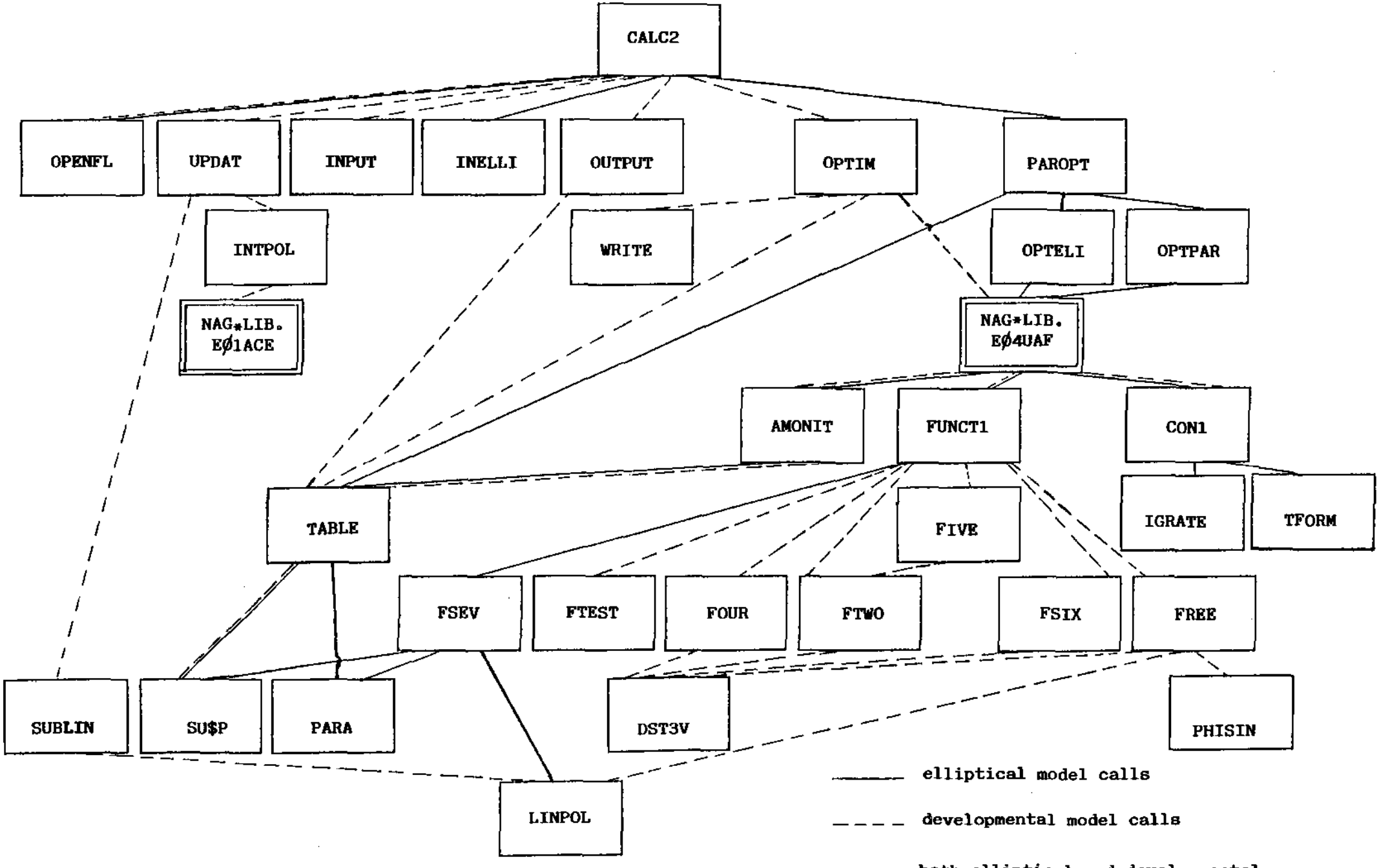


FIGURE 4.1 BLOCK DIAGRAM OF THE SUBROUTINES OF THE IMPLEMENTATION PROGRAM

```
*****  TEST.CALC     *****
**************************

A: Do you wish to calculate optimal positions
       for sources with old data records...(1)
                   with new data records...(2)
                   batch run (upto 5 recs).(3)
B: Modify dose distribution files.....(4)
C: Or exit CALC2.....(Ø)
>
```

FIGURE 4.2 **MAIN MENU PRESENTED BY CALC2 AS APPEARS ON THE COMPUTER TERMINAL VDU DURING AN INTERACTIVE PROGRAM EXECUTION**

```
Record numbers  and titles so far used;
 1 test                  2 test.bench            3 5D ELI 4S 3P
 4 SOURCE 1D             5 5D ELI 5S 4P          6 SM1ØØØ,E.9,R1,MCHCK
 7 SM1;E.ØØ1;R.1         8 SOURCE 2D             9 SOURCE 5D NT3
1Ø TEN                  11 TESTPNTINT           12 SOURCE 5D NT3 LOWACT
13 5D 1S 1P NT3         14 3D ELI 2S 2P NT4     15 5D 1S  3P
16 2DS 3DP              17 5D 1S 1P NT4 ELIREP  18 5D 1S1PNT4 ELI TRY
19 5D 1S1PNT4 ELITRY    2Ø NEW TRY              21 SEVEN
22 SOURCE 5D NEW POS    23 2D TYPE 2            24 SMALL ETA NEW START
25 23 NEW ETA SMALL     26 3S                   27 Concent. circ.
28 5d 4s 3p nt4 ag      29 5D 3S 3P GOOD NT2    3Ø U-Shaped Ca Implant
31 5D 1S 1P NEW POS     32 3D 4S 3P NT4         33 PARA2
34 LIN CONSTRAINT       35 LIN CON 2 2EQ        36 para sources
37 para works           38 Planar Implant       39 PARA TEST
4Ø plane 2              41 Ca Base Tongue. Rec. 42 plane
 Enter rec. no. for calc. or (Ø) to end batch.
```

FIGURE 4.3 **PRESENTATION OF THE TITLES OF THE RECORDS USED, ALLOWING CHOICE OF A RECORD TO BE OPTIMISED, UPDATED, OR NOTED IN A FILE FOR LATER OPTIMISATION IN BATCH MODE**

```
 Enter rec: no. for calc. or (Ø) to end batch.
›27
  Enter parameter MCHCK, (Unsure or none=Ø)
›Ø
Enter also title of record.
›
 Do you wish to change any values? Yes(1)
›1
  ***INELLI***
  ***********

 Enter selection number only;
NTYP..(1)   NELLI.(2)
X est.(3)   ACT...(4)
PNTINT(5)   EXIT..(6)
  Title of file ;Concent. circ.        Rec. no;27
  NTYP = 9 There are 2 ellipses with
 2 , 6 ,
  Estimates of parameters for ellipse 1
  For source   1 T= .157+ØØ1 Act=    15Ø.ØØ
  For source   2 T= .471+ØØ1 Act=    15Ø.ØØ
  Ellipse 1 has Amplitudes   .ØØØ        .5ØØ+ØØØ   .5ØØ+ØØØ
                Phase shift  .157+ØØ1    .ØØØ       .ØØØ
                Center at    .ØØØ        .ØØØ       .ØØØ
  Estimates of parameters for ellipse 2
  For source   1 T= .ØØØ      Act=    15Ø.ØØ
  For source   2 T= .1Ø5+ØØ1 Act=    15Ø.ØØ
  For source   3 T= .2Ø9+ØØ1 Act=    15Ø.ØØ
  For source   4 T= .314+ØØ1 Act=    15Ø.ØØ
  For source   5 T= .419+ØØ1 Act=    15Ø.ØØ
  For source   6 T= .524+ØØ1 Act=    15Ø.ØØ
  Ellipse 2 has Amplitudes   .1ØØ+ØØ1   .ØØØ       .1ØØ+ØØ1
                Phase shift  .157+ØØ1    .ØØØ       .ØØØ
                Center at    .ØØØ        .ØØØ       .ØØØ
›
```

FIGURE 4.4 THE SCREEN DISPLAY OF THE CURRENT DATA STORED IN A RECORD, AND THE OPTIONS AVAILABLE FOR CHANGING THE DATA IF REQUIRED

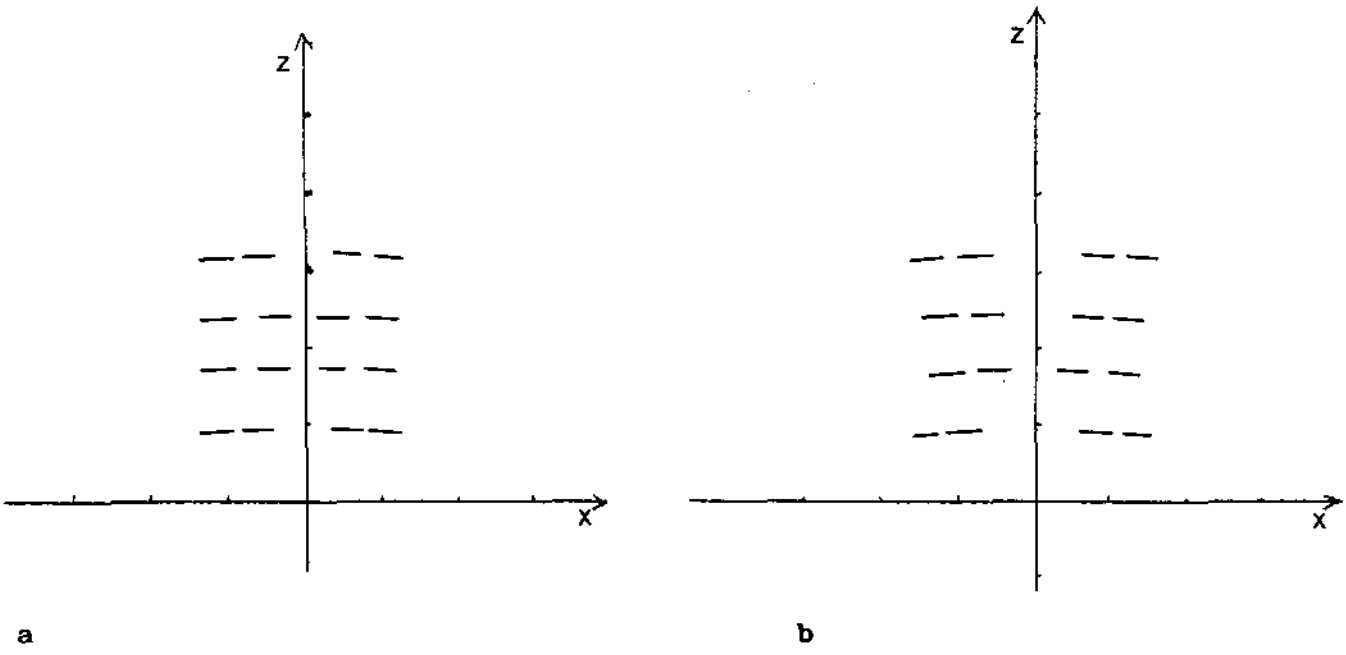


FIGURE 5.1 THE INITIAL (a) AND FINAL (b) POSITIONS OF THE I-125 SEEDS USED IN THE PLANAR IMPLANT, PROJECTED ONTO THE X Z PLANE. THE AXES ARE GRADUATED IN 1 CM STEPS. THE ELLIPSES ON WHICH THE SOURCES LIE HAVE NOT BEEN ALTERED. THE OBJECTIVE FUNCTION IMPROVED FROM 0.2658 TO 0.0675. THE CHANGES IN THE ISODOSE PATTERNS ARE SHOWN IN FIGURE 5.2 AND THE MAXIMUM DEVIATION FROM THE DESIRED DOSES DECREASED FROM 29% TO 19%.

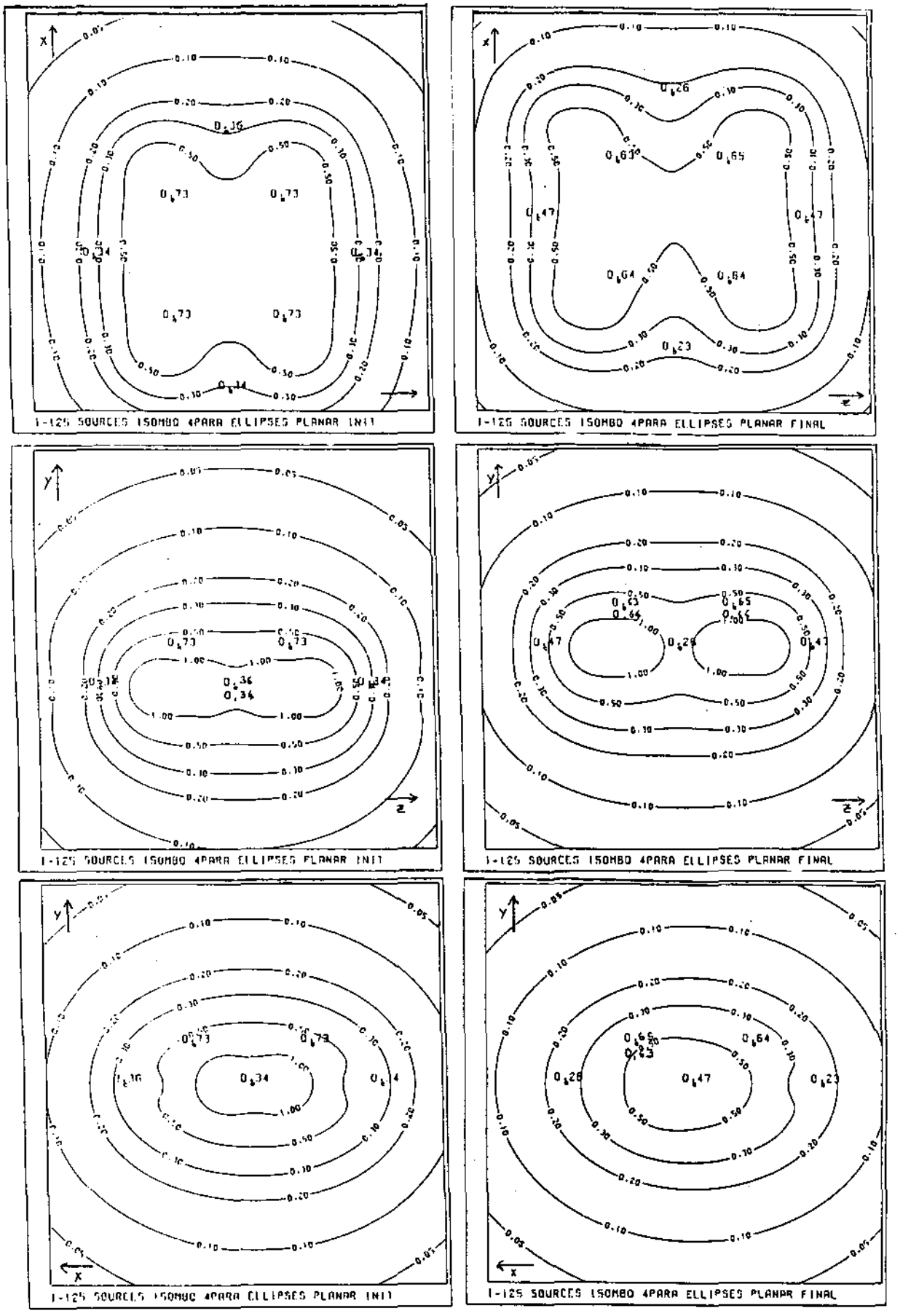


FIGURE 5.2 THE ISODOSE DISTRIBUTIONS IN THREE PERPENDICULAR PLANES BEFORE (Lt) AND AFTER (Rt) OPTIMISATION. ISODOSES GIVEN IN Gy/hr.

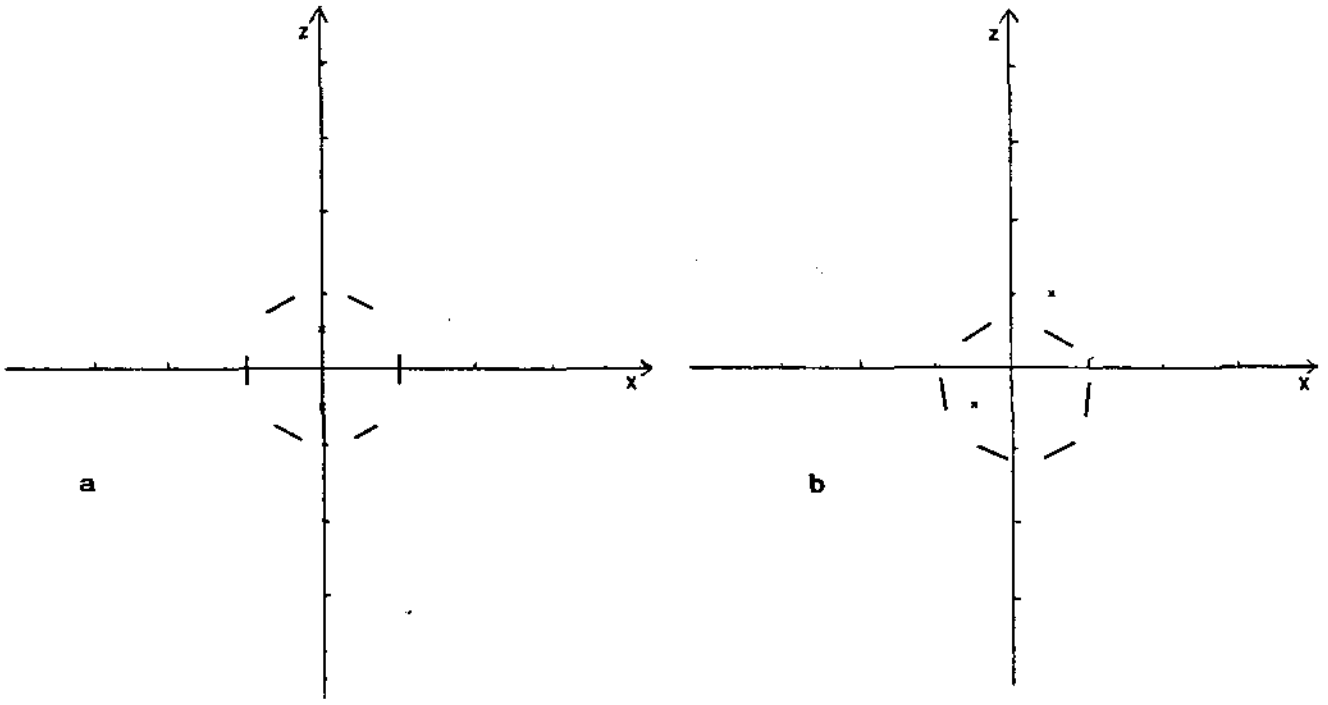


FIGURE 5.3 THE INITIAL (a) AND FINAL (b) SOURCE POSITIONS OF THE I-125 SEEDS USED IN THE ELLIPSOID IMPLANT. THE CROSSES DENOTE SEEDS PERPENDICULAR TO THE PLANE OF THE PROJECTION (THE X Z PLANE). THE POSITIONS OF THE ELLIPSES HAVE CHANGED DURING OPTIMISATION. THE OBJECTIVE FUNCTION IMPROVED FROM 9.503 TO 0.145. THE ISODOSES ARE SHOWN IN FIGURE 5.4. THE MAXIMUM DEVIATION FROM THE DESIRED DOSES DECREASED FROM 550% TO 16%.

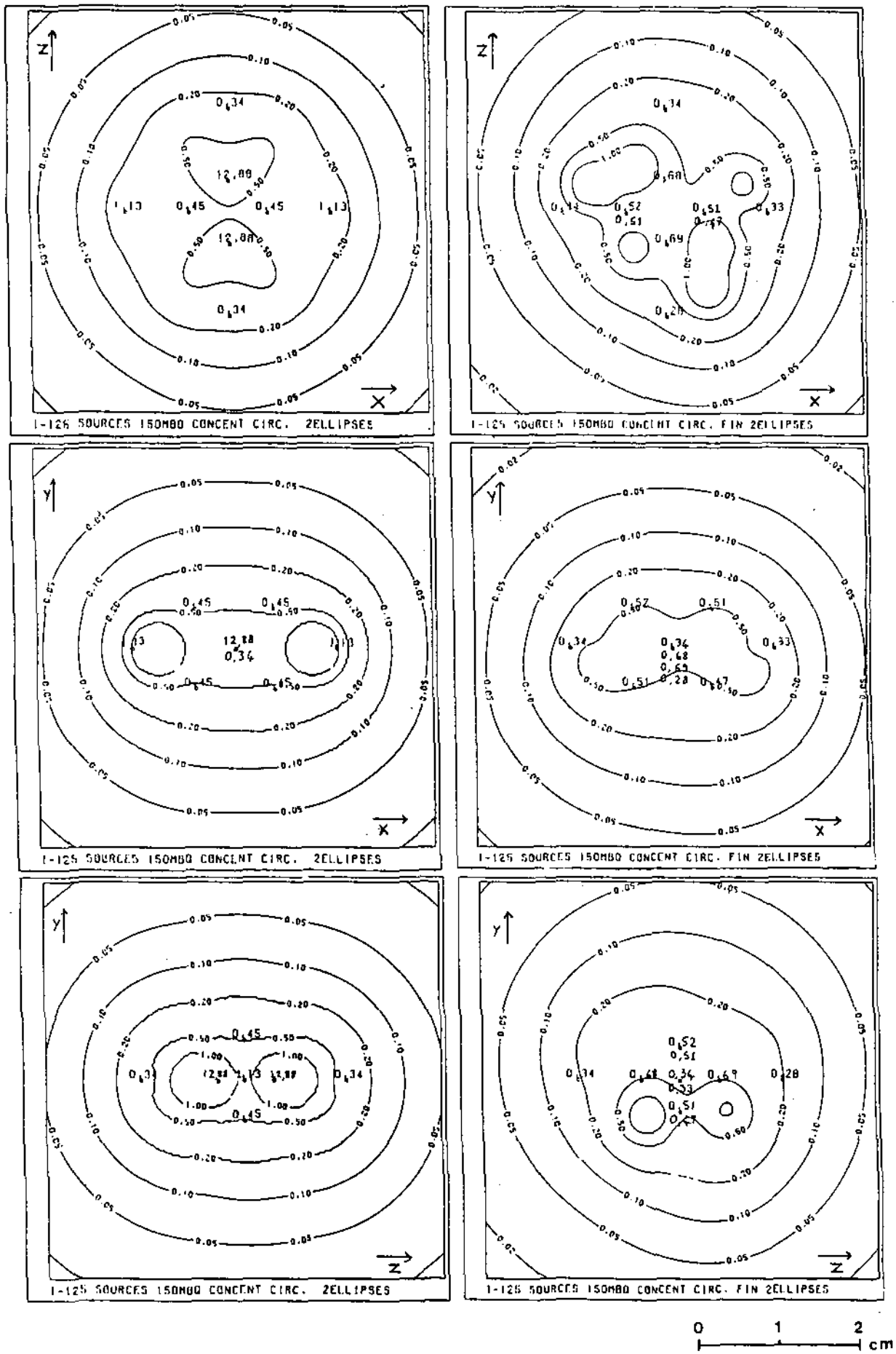


FIGURE 5.4 THE ISODOSE DISTRIBUTION IN THREE PERPENDICULAR PLANES BEFORE (Lt) AND AFTER (Rt) OPTIMISATION. ISODOSES GIVEN IN Gy/hr.

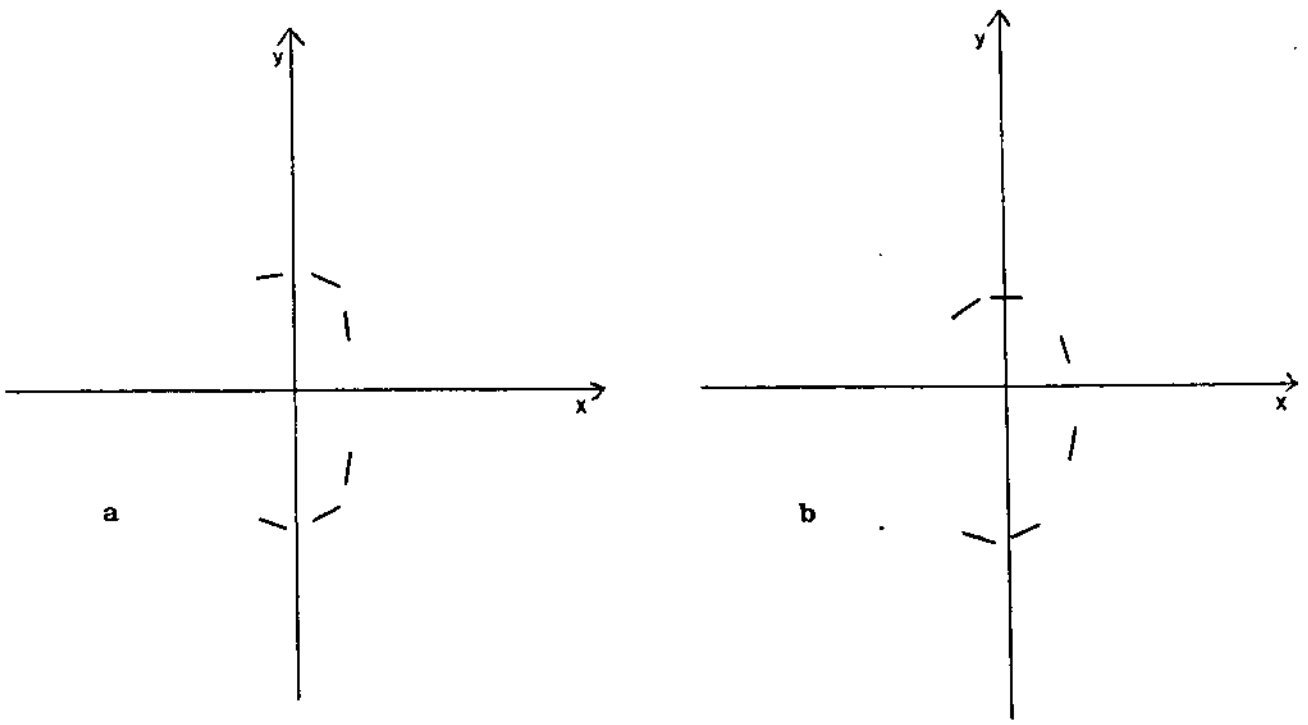


FIGURE 5.5 THE INITIAL (a) AND FINAL (b) SOURCE POSITIONS OF THE I-125 SEEDS USED IN THE U-SHAPED IMPLANT. THE ELLIPSES AND SOURCE POSITIONS HAVE BEEN VARIED TO ACHIEVE A REDUCTION IN MAXIMUM DEVIATION FROM THE DESIRED DOSES FROM 230% TO 36%. THE IMPROVEMENT IN THE ISODOSES IS SHOWN IN FIGURE 5.6

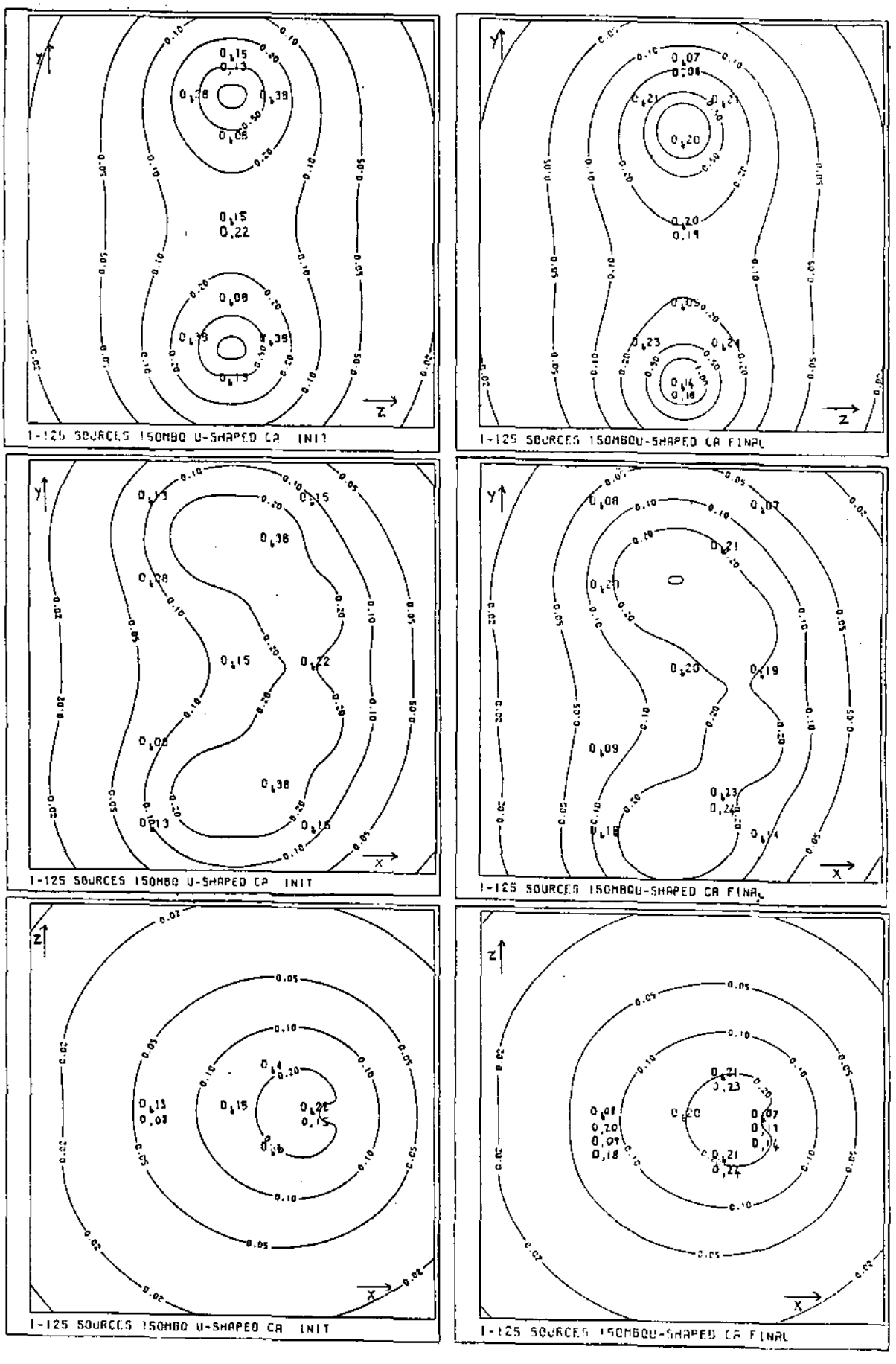


FIGURE 5.6 THE ISODOSE DISTRIBUTIONS IN THREE PERPENDICULAR PLANES BEFORE (Lt) AND AFTER (Rt) OPTIMISATION. ISODOSES GIVEN IN Gy/hr.

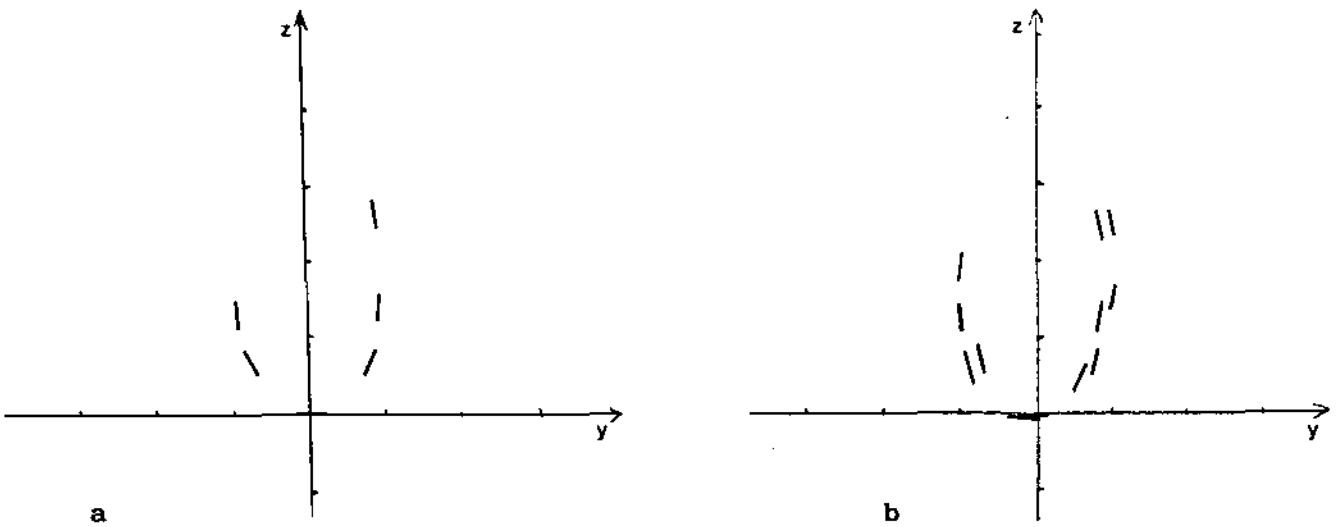


FIGURE 5.7 THE INITIAL (a) AND FINAL (b) SOURCE POSITIONS OF THE CLINICAL EXAMPLE. THE TWO J-SHAPED ELLIPTIC ARCS ARE SUPERIMPOSED IN THE INITIAL CASE AS THE POSITIONS SHOWN ARE PROJECTIONS ONTO THE Y X PLANE. THE MAXIMUM DEVIATION FROM THE DESIRED DOSES DECREASED FROM 51% TO 12%. THE ISODOSES ACHIEVED ARE SHOWN IN FIGURE 5.8

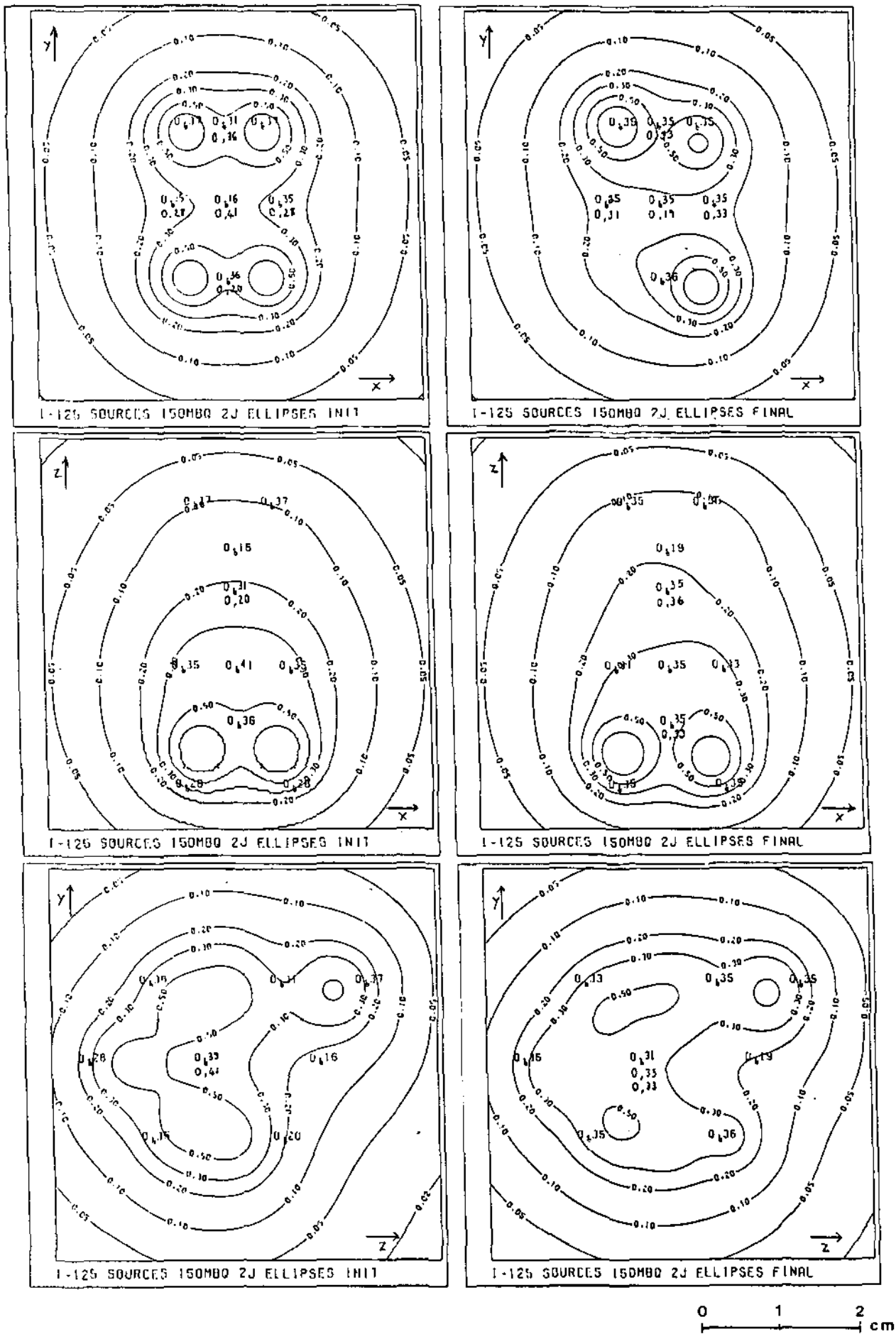


FIGURE 5.8 THE ISODOSE DISTRIBUTIONS IN THREE PERPENDICULAR PLANES BEFORE (Lt) AND AFTER (Rt) OPTIMISATION. ISODOSES GIVEN IN Gy/hr.

REFERENCES

1. Anderson L L. Spacing nomograph for interstitial implants of I-125 Seeds. Med. Phys. 1976; 3(1): 48-51

2. Bahr G K, Kereiakes J G, Horwitz H, Finney R, Galvin J, Goode K. The method of linear programming applied to radiation treatment planning. Radiology 1968; 91: 686-693

3. Batten G W. The M.D. Anderson Method for the computation of isodose curves around interstitial and intracavitary radiation sources. II. Mathematical and computational aspects. Am. J. Roentgenology 1968; 102(3): 673-676

4. Bauer B, Schlegel W, Scharfenberg H, Sturm V, Lorenz W J. Computerised optimisation of treatment planning in the brachytherapy of brain tumours. Proc. 8th Int. Conf. Use of Computers in Radiation Therapy 1984; 395-398

5. Berger M J. Energy deposition in water by photons from point isotropic sources. J. Nucl. Med. M.I.R.D. Pamphlet No 2 1968; Supplement 1: 15-25

6. Bertsekas D P. Multiplier Methods; A Survey. Automatica. 1976; 12: 133-145

7. Biggs P J, Kelly D M. Geometric Reconstruction of Seed Implants using a three-film technique. Med. Phys. 1983; 10(5): 701-704

8. Bjarngard B E. Optimisation in Radiation Therapy. Int. J. Radiation Oncology Biol. Phys. 1977; 2: 381-382

9. Boyer A L, Mok E C. Brachytherapy seed dose distribution calculation employing the Fast Fourier transform. Med. Phys. 1986; 13(4): 525-529

10. Bulski N, Dade M, Kawczynska M. A computer system for dose calculations in brachytherapy. Radiobiol. Radiother. 1983; 24(H5): 638-648

11. Bunday B D. Basic Optimisation Methods. Edward Arnold 1984

12. Chiu-Tsao S, O'Brien K, Sanna R, Tsao H, Vialotti C, Chang Y, Rotman M, Packer S. Monte Carlo dosimetry for I-125 and Co-60 in eye plaque therapy. Med. Phys. 1986; 13(5): 678-682

13. Dale R G. Calculation by computer of dose distributions for superficial gold-198 implants and the derivation of optimised distribution rules. Br. J. Radiology 1976; 49: 533-539

14. Dale R G. A Monte Carlo derivation of parameters for use in the tissue dosimetry of medium and low energy nuclides. Br. J. Radiology 1982; 55: 748-757

15. Dale R G. Some theoretical derivations relating to the tissue dosimetry of brachytherapy nuclide with particular reference to Iodine-125. Med. Phys. 1983; 10(2): 176-183

16. Diffey B L, Klevenhagen S C. An experimental and calculated dose distribution in water around CDC K-type Caesium-137 sources. Phys. Med. Biol. 1975; 20(3): 446-454

17. Dillman L T. Radionuclide decay schemes and nuclear parameters for use in radiation dose estimation. J. Nucl. Med.

1969; 10, M.I.R.D. Suppl. Pamphlet 4:25 6-32.

18. Dolp H, Netto E. Differential und Integralrechnung. 23 Auflage. Verlag Alfred Topelmann, Berlin 1960; pg. 154 and 198

19. Doss L L, Richman M S. The influence of location of the crossing needle on dose distributions from planar implants. Br. J. Radiology 1979; 52: 810-815

20. Glasgow G P. Exposure rate constants for filtered Ir-192 Sources. Med. Phys. 1981; 8(4): 502-503

21. Goffinet D R, Martinez A, Pooler D, Fee W, Levine P A. Intraoperative Pterygo-Palatine interstitial I-125 Seed implants. Int. J. Radiation Oncology Biol. Phys. 1983; 9(1): 103-106

22. Hartmann G H, Schlegel W, Scharfenberg H. The three-dimensional dose distribution of I-125 Seeds in tissue. Phys. Med. Biol. 1983; 28(6): 693-699

23. Henschke U K, Hilaris B S, Mahan G D. Afterloading in interstitial and intracavitary radiation therapy. Am. J. Roentgenology 1963; 90(2): 386-395

24. Hilaris B S. Afterloading 20 yrs of Experience 1955-1975. Proceedings of the 2nd International Symposium on Radiation Therapy 1975 New York, New York. Memorial Sloan Kettering Cancer Centre, New York, New York.

25. Hering E R. Brachytherapy. S A J Science 1986; 82: 406-409

26. Hope C S, Orr J S. Computer Optimisation of 4 MeV Treatment Planning. Phys. Med. Biol. 1965; 10(3): 365-373

27. Hope C S, Laurie J, Orr J S, Halnan K E. Optimization of X-ray Treatment Planning by Computer Judgement. Phys. Med. Biol. 1967; 12(4): 531-542

28. Jayaraman S, Lanzl L H, Agarwal S K. An overview of errors in line source dosimetry for gamma-ray brachytherapy. Med. Phys. 1983; 10(6): 871-875

29. Johns H E, Cunningham J R. The Physics of Radiology IV Edition C C Thomas 1983 Springfield Illinois

30. Jose B, Chu A M, Sharma S C, Tobin D A, D A, Scott R M. Computed Tomography and Radiotherapy in the Treatment of Cancer. Journal of Surgical Oncology. 1983;23:83-92

31. Kline R W, Gullin M T, Grimm D F, Niroomand-Rad A. Computer dosimetry of Ir-192 wire. Med. Phys. 1985;12(5):634-638

32. Kornelsen R O, Young M E J. Brachytherapy build-up factors. British J. Radiology 1981;54:136

33. Kreyszig E. Advanced Engineering Mathematics IV Edition 1979. John Wiley and Sons, New York, New York.

34. Krishnaswamy V. Dose distributions about Cs - 137 sources in tissue. Radiology 1972;105:181-184

35. Krishnaswamy V. Dose distribution around on I-125 seed source in tissue. Radiology 1978;126:489-491

36. Krishnaswamy V. Dose tables for I-125 seed implants. Radiology 1979;132:727-730

37. Kwan D K, Kagan A R, Olch A J, Chan P Y M, Hintz B L, Wollin M. Single- and double-plane iridium-192 interstitial implants: Implantation guidelines and dosimetry. Med. Phys. 1983;10(4):456-461

38. Laughlin J S, Siler W M, Holodny E I, Ritter F W. A dose description system for interstitial radiation therapy. Am. J. Roentgenol 1963;89(3):470-474

39. Lawson C L, Hanson R J. Solving Least Squares Problems. 1974, Prentice-Hall Inc. Englewood Cliffs, New Jersey.

40. Legras J, Legras B, Lambert J P, Aletti P. The use of a microcomputer for non-linear optimisation of doses in external radiotherapy. Phys. Med. Biol. 1986; 31(12): 1353-1359

41. Ling C C, Anderson L L, Shipley W U. Dose Inhomogeneity in interstitial implants using I-125 seeds. Int.J. Radiation Oncology Biol. Phys. 1979; 5(3): 419-425

42. Ling C C, Yorke E D, Spiro I J, Kubiatowicz D, Bennett D. Physical dosimetry of I-125 seeds of a new design for interstitial implant. Int.J.Radiation Oncology Biol. Phys. 1983; 9(11): 1747-1752

43. Ling C C, Schell M C, York E D, Palos B B, Kubiatowicz D. Two-dimensional dose distribution of I-125 seeds. Med. Phys. 1985;12(5):652-655

44. Marchese M J, Hall E J, Hilaris B S. Clinical, Physical and Radiobiological Aspects of Encapsulated Iodine-125 in Radiation Oncology. Endocurie, Hypertherm, Oncol. 1985;1:67-82

45. Martin M, Kuhnle H, Frischkorn R. Calculation of dose distribution for the local irradiation of the endometrium-carcinoma. Strahlentherapie und Onkologie 1986;162(5):308-315

46. Maruyama Y, Van Nagell J R, Wrede D E , Coffey II C, Utley J F, Avila J. Approaches to Optimisation of Dose in Radiation Therapy of Cervix Carcinoma. Radiology 1976;120:389-398

47. McCormick G P. Nonlinear Programming (Theory Algorithms and Applications) 1983 John Wiley and Sons, New York, New York.

48. McDonald S C, Rubin P. Optimization of External Beam Radiation Therapy. Int. J. Radiation Oncology Biol. Phys. 1977;2:307-317

49. Medical Products Division/3 M - I-125 Seeds Models 6702 and 6711 Specification Sheets. Nov,1982. 3M Center, St. Paul, MN 55144

50. Meredith W J. Radium dosage. The Manchester System. 1949 The Williams and Wilkins Co. Baltimore

51. Murphy Jr. D J, Doss L L. Small computer algorithms for comparing therapeutic performances of single-plane iridium implants. Med. Phys.. 1984;11(2):193-196

52. Nemirovsky A S, Yudin D B. Problem Complexity and Method Efficiency in Optimisation. 1983 John Wiley and Sons, New York.

53. Newton C M. What next in radiation treatment optimisation? 5th International Conference in the use of Computers in Radiation Therapy. Hanover N H 1974. Editor E S Stemick. New England University Press.

54. Numerical Algorithms Group. FORTRAN Library Manual. Mark 10. 1983. N.A.G. Central Office, Mayfield House, 256 Banbury Rd, Oxford OX27DE, United Kingdom.

55. Orton C G. Time-dose factors (TDFs) in brachytherapy. Br. J. Radiology 1974;47(561):603-607

56. Packer S, Rotman M, Fairchild R G, Albert D M, Atkins H L, Chan B. Irradiation of Choroidal Melanoma with I-125 Opthalmic Plaque. Arch Opthalmol 1980;98:1453-1457

57. Pierquin B, Dutreix A, Paine C H, Chassagne D, Marinello G, Ash D. The Paris System in Interstitial Radiation Therapy. Acta Radiologica Oncology 1978;17(1):33-48

58. Pistorius S, Groenewald W A. The determination of source position time in Interactivity Radiotherapy. Proc. 8th Int. Conf. Use of Computers in Radiation Therapy. 1984

59. Powell M J D. A method for Nonlinear Constraints in Minimization Problems. Optimisation (Symposium of the Institute of Mathematics and its applications.) Edited by R. Fletcher, Academic Press, London, England, 1969

60. Raeside D E. Monte Carlo Principles and Applications. Phys. Med. Biol. 1976;21(2):181-197

61. Rao G U V, Kan P T, Howells R. Interstitial volume implants with I-125 seeds. Int.J. Radiation Oncology Biol. Phys. 1981;7(3):431-438

62. Redpath A T, Vickery B L, Wright D H. A Set of FORTRAN Subroutines for Optimizing Radiotherapy Plans. Comp. Progs. Biomed 1975; 5: 158-164

63. Redpath A T, Vickery B L, Wright D H. A New Technique for Radiotherapy Planning using Quadratic Programming. Phys. Med. Biol. 1976; 21(5): 781-791

64. Rockafellar R T. The Multiplier Method of Hestenes and Powell Applied to Convex Programming. J. Optim. Theor. Applic. 1973; 12(6): 555-562

65. Rodrigue G. Parallel Computations. Editor G. Rodrigue. Academic Press 1982. New York

66. Rosemark P J, Tobochnik N, Herman M W, Whiting J S, Thompson R W, Weisenburger T H, Metcalf D R, Greenfield M A. A Dose Table Describing Fractions of Peripheral Volume for I-125 Volume Implants. Radiology 1982; 142: 517-520

67. Rosen I I, Lane R G, Kelsey C A. Computation of Dose Distributions for Radioactive seed implants. Acta Radiologica Oncology 1980; 19(1): 41-44

68. Rosenstein L M. A single computer program for optimisation of source loading in cervical intracavitary applicators. Br.

J. Radiology 1977; 50(590): 119-122

69. Schultz U, Busch M, Bamberg M, Schmitt G. The interstitial afterloading therapy. Principles, practical application and first clinical experiences. Strahlentherapie 1984; 160(2): 85-89

70. Sealy R, Buret E, Cleminshaw H, Stannard C, Hering E, Shackleton D, Korrubel J, le Roux P L M, Sevel D, van Oldenborgh M, van Selm J. Progress in the use of iodine therapy for tumours of the eye. Br. J. Radiology. 1980; 53(635): 1052-1060

71. Siddon R L, Chin L M. Two-film brachytherapy reconstruction algorithm. Med. Phys. 1985; 12(1): 77-83

72. Siegel S. Non Parametric Statistics for the behavioural sciences. Mc Graw- Hill Book Company Inc. New York. 1956

73. Starkschall G. A constrained least-squares optimization method for external beam radiation therapy treatment planning. Med. Phys. 1984; 11(5): 659-665

74. Swan G W. Lecture Notes in Biomathematics: Optimization of Human Cancer Radiotherapy. 1981 Springer Verlag, Berlin, Heidelberg, New York.

75. Tai D T, Maruyama Y. Application of Linear Programming to dose optimization in intracavitary implant therapy. Acta Radiologica Oncology 1979; 18(4): 357-366

76. Tokita N, Kim J H, Hilaris B S. Time-dose-volume considerations in Iodine-125 interstitial brachytherapy. Int. J. Radiation Oncology Biol. Phys. 1980; 6(12): 1745-1749

77. Tolbert D D, Reed S A. An examination of the consistency and accuracy of computerised brachytherapy dose predictions. Int. J. Radiation Oncology Biol. Phys. 1981; 7(5): 675-682

78. Topping J. Errors of observation and their treatment. 3rd Edition. 1962 pg82 Chapman & Hall Ltd.

79. Tyson H, Fieldes M A. A BASIC Program for orthogonal polynomials and retrieval of regression coefficients for the original model. Comp. Progs. Biomed. 1982; 15: 151-154

80. van der Leije B A, van Kleffens H J. SIRAD: A program for automatic dosimetry and data transfer for brachytherapy planning. Comp. Progs. Biomed. 1983; 17: 243-248

81. Waterman F M, Strubler K A. Absorbed dose determination for interstitial I-125 boost therapy. Med. Phys. 1983; 10(2): 155-158

82. Weaver K A. Response of LiF powder to I-125 photons. Med. Phys. 1984; 11(6): 850-854

83. Weaver K A. The dosimetry of I-125 Seed eye plaques. Med. Phys. 1986; 13(1): 78-83

84. Williamson J F, Morin R L, Khan F M. Monte Carlo evaluation of the Sievert integral for brachytherapy dosimetry. Phys. Med. Biol. 1983; 28(9): 1021-1032

85. Worthing J F, Geffner J. Treatment of experimental data. 1944. John Wiley & Sons. New York, New York.

86. Young M E J, Batho H F. Dose tables for linear radium sources calculated by an electronic computer. Br. J. Radiology. 1964; 37(433): 38-44

PROGRAM LISTING

An alphabetical listing of the subroutines used to compile the executable element as described in Chapter 4 follows on the subsequent pages. The developmental programs are included, along with the final elliptical model implementation programs. On the first page are the runstreams used in the compilation of the executable element.

****** TRUN1 ******

```
@RUN,W/N WILLUM,AC619-R001,PROJ,3,20C
@ADD TEST.USEAS
@SETC,P
@PRT,S TEST.TRUN1
@PRT,S TEST.USEAS
@PRT,S TEST.FLIST
@FTN,SO TEST.TABLE
@PACK,P TEST.
@MAP,SE TEST.MAP3,.ABS2
@EOF
```

****** USEAS ******

```
@ASG,A DATA4.,FSO
@ASG,A MYPT.,FSO
@ASG,A CONT.,FSO
@ASG,A DATFL.,FSO
@ASG,A MTLK.,FSO
@ASG,A NEWPF.,FSO
@ASG,A TEST.
@ASG,A NAG*LIB.
@USE 18,MTLK.
@USE 19,DATFL.
@USE 20,DATA4.
@USE 21,MYPT.
@USE 22,CONT.
@USE 23,NEWPF.
```

****** FLIST ******

```
@FTN,SO TEST.CALC2
@FTN,SO TEST.OPENFL
@FTN,SO TEST.INPUT
@FTN,SO TEST.INFLLI
@FTN,SO TEST.OPTIM
@FTN,SO TEST.PAROPT
@FTN,SO TEST.OPTPAR
@FTN,SO TEST.OPTFLI
@FTN,SO TEST.WRITE
@FTN,SO TEST.TABLE
@FTN,SO TEST.OUTPUT
@FTN,SO TEST.FUNCT1
@FTN,SO TEST.FTEST
@FTN,SO TEST.FTWO
@FTN,SO TEST.FREE
@FTN,SO TEST.FOUR
@FTN,SO TEST.FIVE
@FTN,SO TEST.FSIX
@FTN,SO TEST.FSEV
@FTN,SO TEST.IGRATE
@FTN,SO TEST.TFORM
@FTN,SO TEST.PARA
@FTN,SO TEST.LINPOL
@FTN,SO TEST.DST3V
@FTN,SO TEST.PHISIN
@FTN,SO TEST.CON1
@FTN,SO TEST.AMONIT
@FTN,SO TEST.UPDAT
@FTN,SO TEST.INTPOL
@FTN,SO TEST.SUBLIN
@FTN,SO TEST.PAG
@MASM,SO TEST.SUBP
```

```
      SUBROUTINE AMONIT (N,M,X,F,C,NITER,NF,GLNORM,COND,POSDEF,
     1  RHO,RLAM)
C
C     This routine is called by E04UAF.
C     It is a monitoring routine supplied by user.
C     Called with frequency dependent on value of IPRINT.
C      It monitors the progress of E04UAF in minimising the Fn.
C      by presenting controlling parameters & CNORM, GLNORM.
C
C     AUTHOR:W.I.D.RAE
C
C     VERSION:19/6/86
C
C      Declaration of variables
C           COND.....Reflects conditioning of problem,better small.
C           F........Value of F(X).
C           GLNORM...Norm. of est. grad. of augmented Lagrangian.
C           RHO......Contains current value of rho if NITER=-1.
C           C(M).....Contains current value of constraints.
C           RLAM(M)..Current set of Lagrange multipliers.
C           X(N).....Contains value of current vector points.
C           CL( )....Lower bounds on Tth range constraint.
C           CU( )....Upper bounds on Tth range constraint.
C           CNORM,CTEMP,DUM,  temporary variables.
C           POSDEF...E04UAF calls AMONIT with POSDEF .TRUE. .
C           NF.......Total calls of FUNCT1 by E04UAF.
C           NITER....No. of iterations in current call of E04JBF.
C           DESDOS...Desired dose at PNTINT.
C           ACT......Activity of sources.
C           SUM......Array of doses at interest points.
C           NINTP....Number of interest points.
C           PNTINT...Interest points for calc.
C           NSRCE....Number of sources.
C           NTYP.....Type of function used by FUNCT1.
C           MPF......File for output.
C           ETA......Accuracy to which the subproblems are minimised.
C           TA.......Dummy array to pass vector of current values.
C           NOUT.....Output unit number.
C           I,L,T,S,R..Local counting integers.
C
C     Common blocks passed from CALC,OPTIM,OPTPAR,PAROPT,OPTELI.
C
      COMMON/ISO/NTYP,PNTINT,NINTP,ACT,DESDOS,NSRCE,MTYP,NDIM
      COMMON/MON/CL,CU,MEQ,MINEQ,MRNGE
      COMMON/DOS/SUM,TITLE
      COMMON/PARCON/XC,NELLI,MELL,LL
C
      DOUBLE PRECISION COND,F,GLNORM,RHO,C(M),RLAM(M),X(N)
      DOUBLE PRECISION CU(20),CL(20)
      DOUBLE PRECISION CNORM,CTEMP,DUM
      DOUBLE PRECISION ETA,STEPMX
      DOUBLE PRECISION TA(40)
      DOUBLE PRECISION XC(80)
      LOGICAL POSDEF
      INTEGER I,NOUT,L,R,S,T,NSRCE,MTYP,NTYP,NINTP
      INTEGER M,N,NF,NITER,MEQ,MINEQ,MRNGE,NDIM
      INTEGER NELLI(5),MELL,LL
      REAL SUM(20),DESDOS(20),ACT(40),PNTINT(60)
      CHARACTER*20 TITLE
C
```

```
      ******  AMONIT  ******


C       Data declaration of unit numbers used.
C
      DATA NOUT/6/,MPF/21/
C
C     Title presentation.
C
      CALL PAG
C
      WRITE(NOUT,100) TITLE,NSRCE
      WRITE(MPF,100) TITLE,NSRCE
100   FORMAT(1X,/,1X,'*****  BRACHY.AMONIT  *****'/
     1 1X,'*****************************',//
     2 1X,' FILE : ',A20,' NSRCE =',I3)
C
      IF(NITER.GE.0) GOTO 200
C
C     Monitoring at end of a cycle of E04UAF with
C     NITER = -1.
C     Calc. of length of Euclidian norm of residual active C( ).
C
C
      KFLAG=2
      DUM=0.0D+0
      IF(MEQ.EQ.0) GOTO 120
      DO 110 R=1,MEQ
      DUM=DUM+C(R)**2
110   CONTINUE
120   IF(MINEQ.EQ.0) GOTO 140
      DO 130 S=1,MINEQ
      L=MEQ+S
      CTEMP=C(L)
      IF(CTEMP.LT.0.0D+0) DUM=DUM+CTEMP**2
130   CONTINUE
140   IF(MRNGE.EQ.0) GOTO 160
      DO 150 T=1,MRNGE
      L=MEQ+MINEQ+T
      CTEMP=C(L)
      IF(CTEMP.LT.CL(T)) DUM=DUM+(CL(T)-CTEMP)**2
      IF(CTEMP.GT.CU(T)) DUM=DUM+(CTEMP-CU(T))**2
150   CONTINUE
160   CNORM=DSQRT(DUM)
C
C     Output of results of AMONIT
C
      WRITE(NOUT,1000) GLNORM,CNORM,RHO
      WRITE(MPF,1000) GLNORM,CNORM,RHO
1000  FORMAT(/,' The end of a cycle of E04JAF'/
     1 1X,' GLNORM =',D12.5,
     1 1X,' CNORM =',D12.4,1X,' RHO =',D12.4)
C       DO 1001 I=1,M
C     WRITE(NOUT,1002) I,RLAM(I)
C     WRITE(MPF,1002) I,RLAM(I)
1002  FORMAT(1X,'The Lagrange Multiplier Number',I3,' =',D12.6)
1001  CONTINUE
C       DO 253 I=1,NINTP
C     WRITE(NOUT,254) I,DESDOS(I),SUM(I)
C     WRITE(MPF,254) I,DESDOS(I),SUM(I)
254   FORMAT(1X,' At int. point ',I3,' Desired dose=',E10.4,
     1 1X,' Calc. dose=',E10.4)
253   CONTINUE
      GOTO 300
C
```

```
      ******  AMONIT  ******

C       Monitoring within routine with NITER
C        giving no of calls of E04JBF.
C
200     CONTINUE
C
        KFLAG=3
        WRITE(NOUT,1500) NF,F
        WRITE(MPF,1500) NF,F
1500    FORMAT(1X,//,' After ',I5,' evaluations ', /
     1  1X,' where the users Function value is ',D12.4)
C
CC        DO 1498 I=1,NINTP
CC      WRITE(NOUT,254) I,DESDOS(I),SUM(I)
C       WRITE(MPF,254) I,DESDOS(I),SUM(I)
1498    CONTINUE
C
        DO 201 I=1,NSRCE
C       WRITE(NOUT,1501) I,ACT(I)
C       WRITE(MPF,1501) I,ACT(I)
1501    FORMAT(1X,'Estimate for Source(',I3,') with ',
     1  1X,' Activity= ',F9.3)
C
        DO 201 J=1,NDIM
        K=(I-1)*NDIM+J
C       WRITE(NOUT,1508) J,X(K)
C       WRITE(MPF,1508) J,X(K)
1508    FORMAT(25X,' X(',I2,')=',D12.6)
201     CONTINUE
CC
CC        DO 202 I=1,M
CC      WRITE(NOUT,1502) I,C(I)
C       WRITE(MPF,1502) I,C(I)
1502    FORMAT(1X,'The constraint value of C(',I3,') is',D12.6)
202     CONTINUE
        WRITE(NOUT,1505) NITER,GLNORM
C       WRITE(MPF,1505) NITER,GLNORM
1505    FORMAT(1X,' There have been ',I3,' iterations'/
     1  1X,' and the norm of the projected gradient',/
     2  1X,' of the augmented Lagrangian is ',D12.4)
        IF(COND.EQ.0.0D+0) RETURN
        IF(COND.LT.1.0D+6) GOTO 210
        WRITE(NOUT,1600)
C       WRITE(MPF,1600)
1600    FORMAT(1X,' The estimated condition number of'/
     1  1X,' its projected Hessian exceeds 1.0D+6',/)
        GOTO 220
C
CC      Conditioning of Hessian.
C
210     WRITE(NOUT,1700) COND
C       WRITE(MPF,1700) COND
1700    FORMAT(1X,'The estimated condition no. of its'/
     1  1X,' projected Hessian is ',D9.2)
C
CC      This is included for generalisation for cases where
CC         the projected Hessian is not pos.def.
C
220     IF(.NOT.POSDEF) WRITE(NOUT,1800)
        IF(.NOT.POSDEF) WRITE(MPF,1800)
1800    FORMAT(1X,' The projected Hessian is not pos. def.')
C
300      CONTINUE
C
CC        Call of routine to give a tabulated output of the results.
C
        CALL TABLE(IFAIL,NITER,MCALL,NINTP,N,M,NTYP,MTYP,ETA,
     1        STEPMX,POSDEF,XC,F,C,PNTINT,ACT,GLNORM,CNORM,
     2    COND,RLAM,RHO,NF,NSRCE,TITLE,KFLAG,SUM,DESDOS,SPECDC,TA)
C
        RETURN
C
        END
```

```
      PROGRAM CALC2
C
C     This routine does the calculations and output for TEST.
C     It makes use of NAG*LIB. E04UAF in routines
C      TEST.OPTIM,.OPTPAR and .OPTELI .
C     These call FUNCT1 which calculates objective fn value.
C     These call CON1 to calculate constraint function values.
C     Allows for various types of sources.
C     Allows optimisation of parameterised equations.
C
C     AUTHOR:W.I.D.RAE
C
C     VERSION:1/9/86
C
C     MAIN PROGRAM BLOCK
C
C      Declaration of variables as listed here below..
C           ETA......Specifies accuracy of linear minimisation.
C           F........Contains value of F(X) on exit.
C           RHO......Current value of parameter rho in Lagrangian.
C           STEPMX...Estimate of Euclidian distance to min.
C           CL( )....Array of dim >or= MRNGE,lower constraint bound.
C           CU( )....Upper bound on constraint.
C           C( ).....Array of dim >or= M, constraint Fn value.
C           RLAM( )..Initial estimates of Lagrange multipliers.
C           W( ).....Array of dim >or= p ,for workspece.
C           XL( )....Array of dim >or= N, has fixed lower bounds.
C           XL( )....Array contains fixed upper bounds of X( ).
C           X( ).....Array contains value of constrained minimum.
C           XTOL.....Accuracy required for solution.
C           MEQ......Equality constraint number.
C           MINEQ....Inequality constraint number.
C           MRNGE....Number of range constraints.
C           M........MEQ+MINEQ+MRNGE.
C           MAXCAL...Limits calls of FUNCT1&CON1 by E04UAF.
C           N........Number of independent variables.
C           NOUT.....Specifies unit for output of data.
C           I........Integer.
C           IBOUND...Type of bounds used see notes.
C           IFAIL....Indicator of type of failure in optimisation.
C           IPRINT...Regulates calling of AMONIT by E04UAF.
C           LCLU.....Actual length of CL,CU declared in CON1.
C           LIW......Actual length of IW.
C           LW.......Actual length of W.
C           IW( )....Integer array dim >or= N+MINEQ+MRNGE+M+12.
C           PNTINT...Points  of interest to calculate uniformity.
C           NTYP.....Type of isotope chosen to  regulate function.
C           ACT......Activity of sources in group.
C           DESDOS...Desired dose of points of interest.
C           NINTP....Number of interest points.
C           MTYP.....Type of constraint function used.
C           NDIM.....Dimension of source variables.
C           SUM......Dose at interest points from all sources.
C           LL.......Control variable for array LREC.
C           LREC.....Array containing rec. nos. for batch use.
C           NSET.....Control variable allowing batchruns.
C
C      Declaration of common blocks.
C
      COMMON/ISO/NTYP,PNTINT,NINTP,ACT,DESDOS,NSRCE,MTYP,NDIM
```

```
******  CALC2  ******

      COMMON/MON/CL,CU,MEQ,MINEQ,MRNGE
      COMMON/DOS/SUM,TITLE
      COMMON/VARCON/STEPMX,ETA,IBOUND,RHO,RLAM,MAXCAL,XTOL,M,XU,XL
      COMMON/WORKS/IPRINT,IFAIL,LAMSET,LCLU,IW,LIW,W,LW
      COMMON/LINCOM/MM,CX1,CX2,CY1,CY2
      COMMON/OPT/MCNTS,MCALL,MFLAG,MITER,MCHCK,XCD
      COMMON/CONCOM/CENTEL,AXEL
      COMMON/PARCON/X,NELLI,MELL,LM
      COMMON/RATE/DRATE,TDOSE,CON
C
CC     Declaration of variables used ;
C
      DOUBLE PRECISION ETA,F,RHO,STEPMX,XTOL
      DOUBLE PRECISION CL(20),CU(20),C(50),W(9000),XL(80),XU(80)
      DOUBLE PRECISION X(80),XCD(80),RLAM(80)
      DOUBLE PRECISION DSQRT
      INTEGER MEQ,MINEQ,MRNGE,I,IBOUND,IFAIL,IPRINT,LCLU,LIW,LW
      INTEGER M,MAXCAL,N,IW(160)
      INTEGER NOUT,NSRCE,NDIM
      INTEGER NTYP,NINTP,MTYP
      INTEGER NSET,LREC(5),LL,MCH(5)
      INTEGER NCOL(60)
      INTEGER MELL,LM
      INTEGER NELLI(5)
      REAL PNTINT(60),ACT(40),DESDOS(20),SUM(20)
      REAL TDOSE,DRATE,CON(40)
      REAL MM(101,101),CX1,CX2,CY1,CY2
      REAL CENTEL(3),AXEL(2)
      DOUBLE PRECISION ELLI(9,4)
      CHARACTER*20 TITLE,TITL
      CHARACTER*20 COLTIT(60)
C
CC     Declaration of logical variables.Allows restart.
C
      LOGICAL LAMSET
C
CC     DATA declaration of output unit numbers.
C
      DATA NOUT /6/
      DATA MPF /21/
      DATA NDAT /20/
      DATA NCON /22/
C
CC     Opening of files used in CALC.
C
      CALL OPENFL
C
CC     Set up of variables if batchrun.
C
      READ(NCON,REC=1) NSET,LREC,MCH
      LL=0
      IF(NSET.EQ.1) THEN
      NCW=1
      GOTO 469
      END IF
C
CC     Title statement.
C
100   CALL PAG
C
      WRITE(NOUT,200 )
200   FORMAT(1X,/'     *****  TEST.CALC     *****'/
```

```
      1 1X,'   **************************'///
      2 1X,'A: Do you wish to calculate optimal positions'/
      3 1X,'       for sources with old data records...(1)'/
      4 1X,'                   with new data records...(2)'/
      5 1X,'                   batch run (upto 5 recs).(3)'/
      6 1X,'B: Modify dose distribution files.........(4)'/
      7 1X,'C: Or exit CALC2.........................(0)')
        READ(5,300 ,ERR=100) NOW
300     FORMAT(I1)
        IF(NOW.GE.5.OR.NOW.LT.0) GOTO 100
C
C        Directs control to STOP if NOW=0.
C
        IF(NOW.EQ.0) GOTO 1000
C
C        Allows call of MODDAT for modification of dose table data.
C         Also interpolation on this data.
C
        IF(NOW.EQ.4) THEN
C
        CALL UPDAT
        GOTO 100
C
        END IF
C
C        Reads all records and prints only those occupied.
C        If NOW =2 then only next empty rec. no. is given.
C        If now not equal to an allowed choice ,treated as =2.
C
490     IF(NOW.NE.1.AND.NOW.NE.2.AND.NOW.NE.3) NOW=2
        IC=0
        DO 500  I=1,60
        READ(NDAT,REC=I,ERR=470) NTYP,NSRCE,ACT,N,MEQ,MINEQ,MRNGE,
     1       IPRINT,STEPMX,ETA,CU,CL,IBOUND,XU,XL,RHC,X,
     2       NINTP,PNTINT,DESDOS,IFAIL,TITLE,MTYP,NDIM,CENTEL,AXEL,
     3     NELLI,ELLI
        IF(N.EQ.0.AND.NOW.EQ.2) THEN
        NREC=I
        WRITE(NOUT,201 ) I
201     FORMAT(1X,' Next new record is no. ',I2,//
     1 1X,' Enter title for new record.')
        READ(5,305) TITL
C
C        Directs control to read all values sequentially.
C
        GOTO 105
        END IF
C
C        Skips blank records or full records depending on NOW.
C
        IF(N.EQ.0.OR.NOW.EQ.2) GOTO 500
        IC=IC+1
        CCLTIT(IC)=TITLE
        NCCOL(IC)=I
500     CONTINUE
C
C        Presents in column form the record numbers and titles used.
C
        CALL PAG
C
        WRITE(NOUT,261)
261     FORMAT(1X,//,'  Record numbers  and titles so far used;')
```

```
      DO 501 I=1,IC,3
      WRITE(NOUT,260) (NOCOL(I+J),CCLTIT(I+J),J=0,2)
260   FORMAT(1X,3(I2,1X,A20))
501   CONTINUE
C
C      Allows choice of record to be used.
C      If in batch run then file numbers read from CONT. are used.
C
469   IF(NSET.EQ.1) THEN
      LL=LL+1
      IF(LREC(LL).LE.0.OR.LREC(LL).GT.100) LREC(LL)=0
      NREC=LREC(LL)
      MCHCK=MCH(LL)
      IF(NREC.EQ.0) GOTO 900
      NYES=0
      GOTO 471
      END IF
C
470   WRITE(NOUT,203 )
203   FORMAT(1X,' Enter rec. no. for calc. or (0) to end batch.')
      READ(5,*,ERR=470) NREC
      IF(NREC.LT.0.OR.NREC.GE.61) GOTO 470
      IF(NREC.EQ.0.AND.NOW.NE.3) GOTO 470
C
CC     Resets NSET to 1 in CONT. and continues run.
C
      IF(NREC.EQ.0.AND.NOW.EQ.3) THEN
      NSET=1
      WRITE(NCON,REC=1) NSET,LREC,MCH
      NSET=0
      GOTO  100
      END IF
C
C      Allows entry of MCHCK which allows iteration of E04UAF
C       every MCHCK calls of FUNCT1 if convergence not fast.
C
475   WRITE(NOUT,220)
220   FORMAT(1X,'  Enter parameter MCHCK, (Unsure or none=0)')
      READ(5,*,ERR=475) MCHCK
C
CC     Writes values to CONT. after setting NSET to 1.
C
      IF(NOW.NE.3) GOTO 474
      LL=LL+1
      LREC(LL)=NREC
      MCH(LL)=MCHCK
      IF(LL.EQ.5) THEN
      NSET=1
      WRITE(NCON,REC=1) NSET,LREC,MCH
      NSET=0
      GOTO 100
      END IF
      GOTO 470
C
C      Allows change of record title.
C
474   WRITE(NOUT,204 )
204   FORMAT(1X,'Enter also title of record.')
      READ(5,305) TITL
305   FORMAT(A20)
C
C      If NOW=1 specific changes are allowed if desired.
```

```
C
476     WRITE(NOUT,205 )
205     FORMAT(1X,' Do you wish to change any values? Yes(1)')
        READ(5,*,ERR=476) NYES
C
C        File is read to give all values as at last usage.
C
471     READ(NDAT,REC=NREC,ERR=900) NTYP,NSRCE,ACT,N,MEQ,MINEQ,MRNGE,
     1          IPRINT,STEPMX,ETA,CU,CL,IBOUND,XU,XL,RHO,X,
     2        NINTP,PNTINT,DESDOS,IFAIL,TITLE,MTYP,NDIM,CENTEL,AXEL,
     3       NELLI,ELLI
C
C
C        Directs control to just before call of E04UAF.
C
        IF(NYES.NE.1) GOTO 401
C
C        Allows for entry of parameterised data into new records.
C
105     IF(NOW.EQ.2) THEN
        WRITE(NOUT,207)
207     FORMAT(1X,' Enter type of source to be used.1..9.')
        READ(5,*,ERR=105) NTYP
        IF(NTYP.LT.1.OR.NTYP.GT.10) GOTO 105
C
        END IF
C
        IF(NTYP.LT.7) THEN
C
        CALL INPUT(TITL,TITLE,NREC,NOW,N,IPRINT,X,STEPMX,
     1              ETA,IBOUND,RHO,XU,XL,IFAIL)
C
        ELSE
C
C
        CALL INELLI(TITLE,NREC,X,NELLI)
C
        N=NELLI(1)
C
C
        END IF
C
        IF(NOW.EQ.1) THEN
C
C         Gives choice of file to be used for present data.
C        writes to desired file.
C
465     WRITE(NOUT,238) TITL
238     FORMAT(1X,' Into which record must ',A20,'be copied?')
        READ(5,*,ERR=465) NREC
        IF(NREC.LE.0.OR.NREC.GT.100) GOTO 465
        TITLE=TITL
        END IF
C
432     WRITE(NDAT,REC=NREC) NTYP,NSRCE,ACT,N,MEQ,MINEQ,MRNGE,IPRINT,
     1          STEPMX,ETA,CU,CL,IBOUND,XU,XL,RHO,X,
     2        NINTP,PNTINT,DESDOS,IFAIL,TITLE,MTYP,NDIM,CENTEL,AXEL,
     3        NELLI,ELLI
C
C        Allows repeat changes.
C
401     IF(NSET.EQ.1) GOTO 405
```

```
      ******  CALC2  ******


      WRITE(NOUT,239)
239   FORMAT(1X,' Do you want to change more? ....(1)',/
     1 1X,                          restart?.........(2)',/
     1 1X,                          cont. optimising.(0)')
      READ(5,*,ERR=401) NYES
      IF(NYES.EQ.2) GOTO 100
      IF(NYES.EQ.1) THEN
      NCW=1
      GOTO 105
      ELSEIF(NYES.NE.0) THEN
      GOTO 401
      END IF
C
C     Call of optimisation routine. Parametric Eq's use PAROPT
C
405   NCIRC=0
407   NCIRC=NCIRC+1
      IF(NTYP.GE.7) THEN
      NREP=NELLI(1)
      LM=1
406   CALL PAROPT(N,C,F)
C
      IF(NTYP.EQ.8.AND.NREP.GE.2) THEN
C
      IF(NSET.EQ.1) THEN
      NEXT=1
      GOTO 242
      END IF
C
241   WRITE(NOUT,240)
240   FORMAT(1X,'Will you optimise Seeds on next ellipse?Yes=1')
      READ(5,*,ERR=241) NEXT
242   IF(NEXT.EQ.1) THEN
      LM=LM+1
      NREP=NREP-1
      GOTO 406
C
      END IF
      END IF
C
      IF(NTYP.EQ.8.AND.NCIRC.EQ.1) GOTO 407
C
      ELSE
C
      CALL OPTIM(N,X,C,F)
C
C      Final presentation of results by OUTPUT .
C
420   CALL OUTPUT(IFAIL,TITLE,F,M,N,X,C,NOUT,MPF,RHO,RLAM)
C
      END IF
C
      IF(NSET.EQ.1) GOTO 469
C
480   WRITE(NOUT,253 )
253   FORMAT(1X,' Will you  cont.(1), or exit(0)?')
      READ(5,*,ERR=1000) NSTOP
      IF(NSTOP.NE.1) GOTO 1000
      GOTO 100
C
900   IF(NSET.EQ.1) NSET=0
950   WRITE(NCON,REC=1) NSET,LREC,MCH
C
1000  STOP 'END OF TEST'
C
      END
```

```
      SUBROUTINE CON1(IFLAG,N,M,XC,CC)
C
CCCCC
CCCCC This is a user supplied routine to evaluate
CCCCC constraint functions for E04UAF in CALC2.
CCCCC
CCCCC AUTHOR:W.I.D.RAE
CCCCC
CCCCC VERSION:11/8/86
CCCCC
CCC    Declaration of variables.
CCC
CCC    Declaration of common block.
C
      COMMON/ISO/NTYP,PNTINT,NINTP,ACT,DESDOS,NSRCE,MTYP,NDIM
      COMMON/CONCOM/CENTEL,AXEL
      COMMON/PARCON/X,NELLI,MELL,LL
      COMMON/RATE/DRATE,TDOSE,CON
C
      DOUBLE PRECISION X(80)
      DOUBLE PRECISION CC(M),XC(N)
      REAL ACT(40),DESDOS(20),PNTINT(60)
      REAL CENTEL(3),AXEL(2)
      REAL RLEN(10),PIB2,TTR,A,B,E2,QUAD,GAP,RAT
      REAL DRATE,TDOSE,CON(40)
      DOUBLE PRECISION AX,AY,DCENT(3)
      INTEGER IFLAG,M,N
      INTEGER NTYP,NINTP,NSRCE,MTYP,NDIM
      INTEGER NELLI(5)
      INTEGER MELL,LL
      DOUBLE PRECISION DPIB2
      DOUBLE PRECISION CONST1,CONST2
      DOUBLE PRECISION T(40)
      DOUBLE PRECISION DSQRT
C
CCC    Data for PIB2
C
      DATA DPIB2/1.5707963270+0/
      DATA PIB2/1.5707963270/
      DATA GAP/0.5/
      DATA RAT/.05/
C
10    IF(MTYP.EQ.1) THEN
CCC
CCC    Constants used in trial run only.
C
      CONST2=DSQRT(2.D+0)
      CONST1=-2.0D+0-3.0D+0*CONST2
      CONST2=2.0D+0-CONST2-CONST2
C
CCC    Constraints for trial run  only.
C
      CC(1)=XC(1)+XC(2)**2+XC(3)**3+CONST1
      CC(2)=XC(2)-XC(3)**2+XC(4)+CONST2
      CC(3)=XC(5)*XC(1)
C
CCC    Constraints for MTYP=2, attempts ellipse.
CCC
20    ELSEIF(MTYP.EQ.2) THEN
C
      DO 11 I=1,NSRCE
```

```
      ****** CON1 ******


      II=5*I-3
      IJ=5*I-4
      IK=3*I-2
      AX=DBLE(AXEL(1)**2)
      AY=DBLE(AXEL(2)**2)
      DCENT(1)=DBLE(CENTEL(1))
      DCENT(2)=DBLE(CENTEL(2))
      DCENT(3)=DBLE(CENTEL(3))
      CC(3*I)=XC(5*I)
      CC(3*I-1)=XC(5*I-2)-DCENT(3)
      CC(IK)=(XC(II)-DCENT(2))**2/AY+(XC(IJ)-DCENT(1))**2/AX-1
11    CONTINUE
C
CC     Constraints only if THETX&PHIX are to be zero.
C
30    ELSEIF(MTYP.EQ.3) THEN
31    DO 32 I=1,NSRCE
      K=(I-1)*3+1
      CC(I)=XC(K)-XC(K+1)
32    CONTINUE
C
CC     Constraints effectively range limiting only.
C
40    ELSEIF(MTYP.EQ.4) THEN
      DO 41 I=1,M
      CC(I)=1000-XC(I)*5
41    CONTINUE
C
CC     Constraints on angular orientation only.
C
50    ELSEIF(MTYP.EQ.5) THEN
C
      DO 51 I=0,NSRCE-1
      CC(I*2+1)=XC(I*5+4)
      CC(I*2+2)=XC(I*5+5)
51    CONTINUE
C
CC     Constrained to ellipse and range constraints on angles.
C
60    ELSEIF(MTYP.EQ.6) THEN
C
      DO 61 I=1,NSRCE*2,2
      II=5*I-3
      IJ=5*I-4
      AX=DBLE(AXEL(1)**2)
      AY=DBLE(AXEL(2)**2)
      DCENT(1)=DBLE(CENTEL(1))
      DCENT(2)=DBLE(CENTEL(2))
      DCENT(3)=DBLE(CENTEL(3))
      CC(I)=XC(5*I-2)-DCENT(3)
      CC(I+1)=(XC(II)-DCENT(2))**2/AY+(XC(IJ)-DCENT(1))**2/AX-1
61    CONTINUE
      DO 62 I=0,NSRCE*2,2
      CC(I+NSRCE*2+1)=XC(I*5+4)
      CC(I+NSRCE*2+2)=XC(I*5+5)
62    CONTINUE
C
CC     This section does constraint value calculation for EC4UAF
CC      in the case of parametrerised elliptical equations.
C
      ELSEIF(MTYP.EQ.7) THEN
C
```

```
      ******  CON1  ******


C       Calc of long and short half axes.
C
        CALL TFORM(A,B,PIB2,TTR)
C
        E2=((A*A)-(B*B))/(A*A)
C
C       Calc of the length of arc of 1 Quadrant.
C
        CALL IGRATE(DPIB2,A,E2,QUAD)
C
        DO 70 I=1,NELLI(LL+1)
        T(I)=XC(I)+TTR
        NQU=0
        NODD=-1
C
C       Calc of length of arc around the ellipse to T(i).
C
72      IF(T(I).LT.0.) THEN
        NQU=NQU+1
73      IF(T(I).LT.-PIB2) THEN
        NQU=NQU+1
        NODD=NODD*(-1)
        T(I)=T(I)+PIB2
        GOTO 73
        ENDIF
        IF(NODD.EQ.1) THEN
        T(I)=PIB2+T(I)
        CALL IGRATE(T(I),A,E2,RLEN(I))
        RLEN(I)=-QUAD*NQU+RLEN(I)
        ELSE
        T(I)=-T(I)
        CALL IGRATE(T(I),A,E2,RLEN(I))
        RLEN(I)=-QUAD*(NQU-1)-RLEN(I)
        ENDIF
        GOTO 7C
        END IF
        IF(T(I).GE.0) THEN
        NQU=NQU+1
74      IF(T(I).GT.PIB2) THEN
        NQU=NQU+1
        NODD=NODD*(-1)
        T(I)=T(I)-PIB2
        GOTO 74
        END IF
        IF(NODD.EQ.1) THEN
        T(I)=PIB2-T(I)
        CALL IGRATE(T(I),A,E2,RLEN(I))
        RLEN(I)=QUAD*NQU-RLEN(I)
        ELSE
        CALL IGRATE(T(I),A,E2,RLEN(I))
        RLEN(I)=QUAD*(NQU-1)+RLEN(I)
        END IF
        END IF
70      CONTINUE
C
        DO 76 I=1,(NELLI(LL+1)-1)
        CC(I)=DBLE(RLEN(I+1)-RLEN(I)-GAP)
        CCN(I+(LL-1)*10)=CC(I)
76      CONTINUE
C
        CC(NELLI(LL+1))=DBLE(RLEN(1)-RLEN(NELLI(LL+1))+4*QUAD-GAP)
        CCN(NELLI(LL+1)+(LL-1)*10)=CC(NELLI(LL+1))
```

```
      ******  CON1  ******
C
C      Setting of range constraint.
C
      CC(NELLI(LL+1)+1)=DRATE
C
C      This section does the calc. for E04UAF of constraint
C       function for the ellipse parameters.
C
      ELSE
C
      DO 81 J=1,9*NELLI(1)
      X(J+40)=XC(J)
81    CONTINUE
C
      DO 80 I=1,NELLI(1)
      LL=I
      IT=I*2
      CALL TFORM(A,B,PIB2,TTR)
      CC(IT-1)=DBLE(B/A-RAT)
      CC(IT)=DBLE(B*PIB2*B-NELLI(I+1))
80    CONTINUE
C
C      Setting of range constraint value.
C
      CC(NELLI(1)*2+1)=DRATE
      LL=0
      END IF
C
      RETURN
C
      END
```

```
      SUBROUTINE DST3V(PNTINT,NSRCE,NINTP,NDIM,W,XC,I,J,DSTSQ)
C
C      This is a routine to calculate distance between an
C      (Ith)interest point and a (Jth) Source position for FUNCT1.
C       Only allows 3-dim in interest point and variable in NDIM.
C
C      AUTHOR:W.I.D.RAE
C
C      VERSION:11/3/86
C
C       Declaration of variables.
C            XC.......Contains point at which F(X) is required.
C            EX.etc...Variables used for calculations of DIST.
C            DSTSQ....Square of distance DIST.
C            I,J......Control variables used in calculation.
C
C
C       Declarations
C
      DOUBLE PRECISION XC(NDIM*NSRCE)
      REAL W(NDIM),XP,YP,ZP
      REAL DSTSQ(NINTP*NSRCE),PNTINT(NINTP*3)
C
C       First calculates value for DSTSQ.
C
      I3=I*3
20    XP=PNTINT(I3-2)
      YP=PNTINT(I3-1)
      ZP=PNTINT(I3)
C
      KJ=(J-1)*NDIM
      DO 10 K=1,NDIM
      W(K)=XC(KJ+K)
10    CONTINUE
C
      IF(NDIM.LE.1) W(2)=0.
C
      IF(NDIM.LE.2) W(3)=0.
C
      L=(I-1)*NSRCE+J
      DSTSQ(L)=(XP-W(1))*(XP-W(1))+(YP-W(2))*(YP-W(2))+
     1          (ZP-W(3))*(ZP-W(3))
C
      IF(DSTSQ(L).LT.0.0001) THEN
      PNTINT(I3-2)=PNTINT(I3-2)+0.02
      GOTO 20
      END IF
C
      RETURN
C
      END
```

```
      SUBROUTINE FIVE(IFLAG,N,XC,FC)
C
C     This is a user supplied routine calculating values for
C     objective function for  E04-UAF.
C     Same calculation as in FTWO but a sum of squared diffs is used.
C
C     AUTHOR:W.I.D.RAE
C
C     VERSION:11/8/86
C
C      Declaration of variables.
C           IFLAG....Set to 0, if neg. then stops E04UAF immediately.
C           XC.......Contains point at which F(X) is required.
C           FC.......Value of objective function at point XC.
C
C      Passed common block allowing control of source type.
C
      COMMON/ISO/NTYP,PNTINT,NINTP,ACT,DESDOS,NSRCE,MTYP,NDIM
      COMMON/DOS/SUM
C
C      Declarations
C
      INTEGER IFLAG,N,NTYP,NSRCE,NINTP,MTYP,NDIM
      DOUBLE PRECISION FC,XC(N)
      REAL SUM(20)
      REAL PNTINT(60),ACT(40),DESDOS(20)
C
C
C     Calculation of dose for point srce of I-125,
C      with same calc. as in NTYP=2, but using
C      Least  Squares fit to many different DESDOS's.
C
      CALL FTWO(IFLAG,N,XC,FC)
C
5001  FC=0.
      DO 5010 I=1,NINTP
      FC=FC+(SUM(I)-DESDOS(I))**2
5010  CONTINUE
C
      RETURN
8050  END
```

```
      SUBROUTINE FTEST(IFLAG,N,XC,FC)
C
C     This is a user supplied routine calculating values for
C     objective function for  E04-UAF.
C     This is the test  subroutine supplied in the user manual.
C
C     AUTHOR:W.I.D.RAE
C
C     VERSION:11/9/86
C
C      Declaration of variables.
C           IFLAG....Set to 0, if neg. then terminates immediately.
C           XC.......Contains point at which F(X) is required.
C           FC.......Value of objective function at point XC.
C      Declarations
C
      INTEGER IFLAG,N
      DOUBLE PRECISION FC,XC(N)
C
1000  CONTINUE
      FC=(XC(1)-100+C)**2+(XC(1)-XC(2))**2+(XC(2)-XC(3))**3
     1 +(XC(3)-XC(4))**4+(XC(4)-XC(5))**4
C
      RETURN
C
8050  END
```

```
      SUBROUTINE FOUR(IFLAG,N,XC,FC)
C
CCCC  This is a user supplied routine calculating values for
CCCC  a weighted least squares objective function for  EC4-UAF.
CCCC  Does calculation for point source of I-125.
CCCC
CCCC  AUTHOR:W.I.D.RAE
CCCC
CCCC  VERSION:11/9/86
CCCC
CCCC   Declaration of variables.
CCCC        IFLAG....Set to 0, if neg. then terminates immediately.
CCCC        XC.......Contains point at which F(X) is required.
CCCC        FC.......Value of objective function at point XC.
CCCC        NTYP.....Passes to section of FUNCT1 for source type.
CCCC        DOSE.....Gives value  of dose at point from source.
CCCC        ECKS.etc.Variables used for calculations of DIST.
CCCC        DIST(IK).Distance from PNTINT(I) to sources (J).
CCCC        SQDST....Square of distance DIST.
CCCC        AC,etc...Constants for RDF calculation.
CCCC        SDC......Spec dose const for I125 in H20.(Gy.cm2/hr.MBq)
CCCC                 Reference:Dale;Med.Phys.10(2);Mar1983.
CCCC        RDF......Radial distribution function.
CCCC        SUM......Sum of all doses for each point.
CCCC        DESDOS...Desired dose passed from CALC.
CCCC
C      Passed common block allowing control of source type.
C
      COMMON/ISO/NTYP,PNTINT,NINTP,ACT,DESDOS,NSRCE,MTYP,NDIM
      COMMON/DOS/SUM
C
CCC    Declarations
C
      INTEGER IFLAG,N,NTYP,NSRCE,NINTP,MTYP,NDIM
      DOUBLE PRECISION FC,XC(N)
      REAL DOSE
      REAL SUMDIS(20)
      REAL SUM(20),DSTSQ(100),DIST(100)
      REAL RDF,SDC,PNTINT(60),ACT(40),DESDOS(20)
      REAL XX(10)
C
CCC    Data for calculation of RDF.
C
      DATA AC/0.97987/,A1/0.079621/,A2/-0.079138/,A3/0.008326/
      DATA SDC/3.636E-4/
C
CCCC  For point sources of I-125 with DESDOS(I) equal at all NINTP.
CCCC  Sums over all sources for each point.
CCCC   First calculates value for DIST and SQDST.
CC
2101  DO 210C I=1,NINTP
      SUM(I)=0
      SUMDIS(I)=0.
      DO 2201 J=1,NSRCE
      IK=((I-1)*NSRCE)+J
      CALL DST3V(PNTINT,NSRCE,NINTP,NDIM,XX,XC,I,J,DSTSQ)
C
      SUMDIS(I)=SUMDIS(I)+DSTSQ(IK)
      DIST(IK)=SQRT(DSTSQ(IK))
C
CC     Calculation of RDF using data and DIST values.
C
      RDF=AC+A1*DIST(IK)+A2*DSTSQ(IK)+A3*DIST(IK)**3
      DOSE=ACT(J)*SDC*RDF/DSTSQ(IK)
      SUM(I)=SUM(I)+DOSE
2201  CONTINUE
210C  CONTINUE
C
CC     Calculation of FC the objective function.
C
      FC=0.
      DO 2000 I=1,NINTP
      FC=FC+SUMDIS(I)*(DESDOS(I)-SUM(I))**2
2000  CONTINUE
C
5001  RETURN
C
      END
```

```
      SUBROUTINE FREE(IFLAG,N,XC,FC)
C
CCC   This is a user supplied routine calculating values for
CCC    a least squares objective function for E04-UAF.
CCC    For I-125 seeds in 3-dim.
CCCCCC
CCC   AUTHOR:W.I.D.RAE
CCCCCC
CCC   VERSION:11/9/86
CCCCCC
CCC    Declaration of variables.
CCC         IFLAG....Set to 0, if neg. then terminates immediately.
CCC         XC.......Contains point at which F(X) is required.
CCC         FC.......Value of objective function at point XC.
CCC         NTYP.....Passes to section of FUNCT1 for source type.
CCC         DOSE.....Gives value  of dose at point from source.
CCC         ECKS.etc.Variables used for calculations of DIST.
CCC         DIST(I,J).Distance from PNTINT(I) to sources (J).
CCC         SQDST....Square of distance DIST.
CCC         SUM......Sum of all doses for each point.
CCC         DESDOS...Desired dose passed from CALC.
CCC         MM.......Array containing dose distribution values.
CCC         CX1,CX2..Scaling factors for values passed to LINPOL.
CCC         CY1,CY2..Scaling factors as above in Y-axis.
CCC
CCC    Passed common block allowing control of source type.
C
      COMMON/ISO/NTYP,PNTINT,NINTP,ACT,DESDOS,NSRCE,MTYP,NDIM
      COMMON/DOS/SUM
      COMMON/LINCOM/MM,CX1,CX2,CY1,CY2
C
CCC    Declarations
C
      INTEGER IFLAG,N,NTYP,NSRCE,NINTP,MTYP,NDIM
      DOUBLE PRECISION FC,XC(N)
      REAL DOSE
      REAL SUMDIS(20)
      REAL SUM(20),SINPHI,DSTSQ(100)
      REAL PNTINT(60),ACT(40),DESDOS(20)
      REAL MM(101,101),CX1,CX2,EX,CY1,CY2,WI
      REAL XX(10)
C
C
      FC=0.DD+0
      DO 2012 I=1,NINTP
      SUM(I)=0.
      SUMDIS(I)=0.
      DO 2015 J=1,NSRCE
      IK=(I-1)*NSRCE+J
C
      CALL DST3V(PNTINT,NSRCE,NINTP,NDIM,XX,XC,I,J,DSTSQ)
C
      SUMDIS(I)=SUMDIS(I)+DSTSQ(IK)
      EX=1/SQRT(DSTSQ(IK))
C
      CALL PHISIN(XC,PNTINT,NSRCE,NINTP,I,J,DSTSQ,SINPHI)
C
      WI=SINPHI
C
      CALL LINPOL(CX1,CX2,CY1,CY2,MM,EX,WI,RES)
C
      DOSE=ACT(J)*RES
      SUM(I)=SUM(I)+DOSE
2015  CONTINUE
      FC=FC+SUMDIS(I)*(SUM(I)-DESDOS(I))**2
C
2012  CONTINUE
C
      RETURN
C
805C  END
```

```
      SUBROUTINE FSEV(IFLAG,N,T,FC)
C
C     This is a user supplied routine calculating values for
C     objective function for  E04-UAF.
C      This uses parameter variables for the optimisation.
C       and variables in 3-dim with two sperical coordinate
C       angles describe the spatial position& orientation of the
C       I-125 seeds , thus creating a 5-dim vector.
C
C     AUTHOR:W.I.D.RAE
C
C     VERSION:21/10/86
C
C      Declaration of variables.
C           IFLAG....Set to 0, if neg. then terminates immediately.
C           T........Array of parameters being optimised.
C           FC.......Value of objective function at point XC.
C           NTYP.....Passes to section of FUNCT1 for source type.
C           DOSE.....Gives value  of dose at point from source.
C           ECKS.etc.Variables used for calculations of DIST.
C           DIST(I,J).Distance from PNTINT(I) to sources (J).
C           SQDST....Square of distance DIST.
C           SUM......Sum of all doses for each point.
C           DESDOS...Desired dose passed from CALC.
C           MM.......Array containing dose distribution values.
C           CX1,CX2..Scaling factors for values passed to LINPOL.
C           CY1,CY2..Scaling factors as above in Y-axis.
C            EX.......Inverse distance used as array counter.
C            WI.......Sin of relative seed inclination.
C            RES......The resultant function value.
C            DRATE....The achieved dose rate at PNTINT(1).
C           DDOS.....Desired dose if uniform over all NINTP.
C            ACT......The activity of the seeds used.
C            UVEC.....Unit vector in the direction of seed.
C            RVEC.....Vector from seed to interest point.
C            DUMDOS...Dummy dose array to hold component doses.
C
C      Passed common block allowing control of source type.
C
      COMMON/ISO/NTYP,PNTINT,NINTP,ACT,DESDOS,NSRCE,MTYP,NDIM
      COMMON/DOS/SUM
      COMMON/LINCOM/MM,CX1,CX2,CY1,CY2
      COMMON/PARCON/X,NELLI,MELL,LL
      COMMON/DUMSUM/DUMDIS,DUMDOS
      COMMON/RATE/DRATE,TDOSE,CON
C
C      Declarations
C
      INTEGER IFLAG,N,NTYP,NSRCE,NINTP,MTYP,NDIM
      INTEGER MELL,LL
      INTEGER NELLI(5)
      DOUBLE PRECISION FC,XW(5)
      DOUBLE PRECISION T(N),TEM
      DOUBLE PRECISION X(80)
      REAL DOSE
      REAL DRATE,TDOSE,CON(40)
      REAL DUMDIS(4,20)
      REAL DUMDOS(4,20)
      REAL SUM(20),DSTSQ
      REAL PNTINT(60),ACT(40),DESDOS(20)
```

```
      ******  FSEV  ******


      REAL MM(101,101),CX1,CX2,EX,CY1,CY2,WI
      REAL UVEC(3),RVEC(3)
C
      LDIM=5
      LT=LL
C
      FC=0.0D+0
C
C      Calculation if all the ellipses are to be optimised.
C
      IF(LL.EQ.0) THEN
C
      IF(MELL.EQ.-1) GOTO 112
C
      DO 110 I=1,NELLI(1)
      IK=9*(I-1)
      DO 110 J=1,9
      X(40+IK+J)=T(J+IK)
C
110   CONTINUE
C
112   DO 10 I=1,NINTP
      DO 10 L=1,NELLI(1)
C
C      Setting of dummy arrays.
C
      SLM(I)=0.
      LL=L
      DLMDOS(L,I)=0.
      LJ=(L-1)*10
      NS=NELLI(L+1)
      DO 20 J=1,NS
      TEM=X(LJ+J)
      CALL PARA(XW,TEM,NS,LDIM)
C
      C4=COS(SNGL(XW(4)))
      C5=COS(SNGL(XW(5)))
      S4=SIN(SNGL(XW(4)))
      S5=SIN(SNGL(XW(5)))
C
C      Calc. of the unit vector in the direction of the seeds axis.
C
      UVEC(1)=C5*S4
      UVEC(2)=S4*S5
      UVEC(3)=C4
C
C        Calculation of the dot product of UVEC&RVEC.
C
      I3=I*3
      UR=0.
      DO 300 K=1,3
      RVEC(K)=XW(K)-PNTINT(I3-3+K)
      UR=UVEC(K)*RVEC(K)+UR
300   CONTINUE
C
C      Calc. of the length of vector from point to source.
C
305   DSTSQ=RVEC(1)*RVEC(1)+RVEC(2)*RVEC(2)+RVEC(3)*RVEC(3)
C
      IF(DSTSQ.LT.0.0001) THEN
      PNTINT(I3-2)=PNTINT(I3-2)+0.02
      RVEC(1)=XW(1)-PNTINT(I3-2)
```

```
      ******  FSEV  ******


      DSTSQ=RVEC(1)*RVEC(1)+RVEC(2)*RVEC(2)+RVEC(3)*RVEC(3)
      END IF
C
      ECK=1/DSTSQ
      EX=SQRT(ECK)
      FA=UR*UR*ECK
C
      IF(FA.GE.1) FA=1.
C
C      Calc of sin of relative inclination of source to Intpnt.
C
      WI=SQRT(1.-FA)
C
C      Linear interpolation to obtain function value required.
C
      CALL LINPOL(CX1,CX2,CY1,CY2,MM,EX,WI,RES)
C
      DOSE=ACT(J+LJ)*RES
      SUM(I)=SUM(I)+DOSE
C
20    CONTINUE
C
      DUMDOS(L,I)=SUM(I)
C
10    CONTINUE
C
      GOTO 116
C
C      Calc of doses delivered by one ellipse of sources.
C
      ELSEIF(LL.GT.0) THEN
C
      LI=(LT-1)*10
      DO 111 I=1,NELLI(LL+1)
      X(LI+I)=T(I)
111   CONTINUE
C
      DO 114 I=1,NINTP
C
      SUM(I)=0.
C
      DO 100 K=1,NELLI(LT+1)
C
      TEM=X(LI+K)
      CALL PARA(XW,TEM,NELLI(LT+1),LDIM)
C
      C4=COS(SNGL(XW(4)))
      C5=COS(SNGL(XW(5)))
      S4=SIN(SNGL(XW(4)))
      S5=SIN(SNGL(XW(5)))
C
C      Calc. of the unit vector in the direction of the seeds axis.
C
      UVEC(1)=C5*S4
      UVEC(2)=S4*S5
      UVEC(3)=C4
C
C        Calculation of the dot product of UVEC&RVEC.
C
      I3=I*3
      UR=0.
      DO 320 KK=1,3
```

```
      ******  FSEV  ******


      RVEC(KK)=XW(KK)-PNTINT(I3-3+KK)
      UR=UVEC(KK)*RVEC(KK)+UR
320   CONTINUE
C
315   DSTSQ=RVEC(1)*RVEC(1)+RVEC(2)*RVEC(2)+RVEC(3)*RVEC(3)
C
      IF(DSTSQ.LT.0.0001) THEN
      PNTINT(I3-2)=PNTINT(I3-2)+0.02
      RVEC(1)=XW(1)-PNTINT(I3-2)
      DSTSQ=RVEC(1)*RVEC(1)+RVEC(2)*RVEC(2)+RVEC(3)*RVEC(3)
      END IF
C
      ECK=1/DSTSQ
      EX=SQRT(ECK)
      FA=UR*LR*ECK
C
      IF(FA.GE.1) FA=1.
C
      WI=SQRT(1.-FA)
C
C
      CALL LINPOL(CX1,CX2,CY1,CY2,MM,EX,WI,RES)
C
      DOSE=ACT(K+LI)*RES
      SUM(I)=SUM(I)+DOSE
C
100   CONTINUE
      DUMDOS(LT,I)=SUM(I)
C
C
114   CONTINUE
C
      END IF
C
116   CONTINUE
C
      DO 200 I=1,NINTP
      SUM(I)=0.
      DO 200 II=1,NELLI(1)
      SUM(I)=DUMDOS(II,I)+SUM(I)
200   CONTINUE
C
C      Dose rate at pntint(1) used as a constraint and a
C       normalisation constant.
C
      DRATE=SUM(1)
C
C      Calc. of least squares objective function.
C
      DO 220 I=1,NINTP
      SUM(I)=SUM(I)/DRATE
      FC=FC+(SUM(I)-DESDOS(I))**2
220   CONTINUE
C
      LL=LT
C
      RETURN
C
8050  END
```

```
      SUBROUTINE FSIX(IFLAG,N,XC,FC)
C
CCCCC This is a user supplied routine calculating values for
CCCCC a least squares objective function for  E04-UAF.
CCCCC
CCCCC AUTHOR:W.I.D.RAE
CCCCC
CCCCC VERSION:11/9/86
CCCCC
CCCCC  Declaration of variables.
CCCCC       IFLAG....Set to 0, if neg. then terminates immediately.
CCCCC       XC.......Contains point at which F(X) is required.
CCCCC       FC.......Value of objective function at point XC.
CCCCC       DOSE.....Gives value  of dose at point from source.
CCCCC       DSTSQ....Square of distance from PNTINT(I) to SOURCE
CCCCC       SDC......Spec dose const for I125 in H20.(Gy.cm2/hr.MBq)
CCCCC                Reference:Dale;Med.Phys.10(2);Mar1983.
CCCCC       RDF......Radial distribution function, assumed 1. here.
CCCCC       SUM......Sum of all doses for each point.
CCCCC       DESDOS...Desired dose passed from CALC.
CCCCC
CCCCC  Passed common block allowing control of source type.
C
      COMMON/ISO/NTYP,PNTINT,NINTP,ACT,DESDOS,NSRCE,MTYP,NDIM
      COMMON/DOS/SUM
C
CCC    Declarations
C
      INTEGER IFLAG,N,NTYP,NSRCE,NINTP,MTYP,NDIM
      DOUBLE PRECISION FC,XC(N)
      REAL DOSE
      REAL SUM(20),XX(10),DSTSQ(800)
      REAL RDF,SDC,PNTINT(60),ACT(40),DESDOS(20)
C
CCC    Data for calculation of Dose.
C
      DATA RDF/1./
      DATA SDC/3.636E-4/
C
CCC    For 1 Dim case to check dose calc.
C
      FC=0.
      DO 6001 I=1,NINTP
      SUM(I)=0
      DO 6002 J=1,NSRCE
      IK=((I-1)*NSRCE)+J
C
      CALL DST3V(PNTINT,NSRCE,NINTP,NDIM,XX,XC,I,J,DSTSQ)
C
      DOSE=ACT(J)*SDC*RDF/DSTSQ(IK)
      SUM(I)=DOSE+SUM(I)
6002  CONTINUE
      FC=(DESDOS(I)-SUM(I))**2+FC
6001  CONTINUE
C
      RETURN
C
8050  END
```

```
      SUBROUTINE FTWC(IFLAG,N,XC,FC)
C
CCC   This is a user supplied routine calculating values for
CCC   objective function for  E04-UAF.
CCC   Does calculation for point source of I-125.
CCC
CCC   AUTHOR:W.I.D.RAE
CCC
CCC   VERSION:11/9/86
CCC
CCC    Declaration of variables.
CCC         IFLAG....Set to 0, if neg. then terminates immediately.
CCC         XC.......Contains point at which F(X) is required.
CCC         FC.......Value of objective function at point XC.
CCC         NTYP.....Passes to section of FUNCT1 for source type.
CCC         DOSE.....Gives value  of dose at point from source.
CCC         ECKS.etc.Variables used for calculations of DIST.
CCC         DIST(IK).Distance from PNTINT(I) to sources (J).
CCC         SQDST....Square of distance DIST.
CCC         AO,etc...Constants for RDF calculation.
CCC         SDC......Spec dose const for I125 in H2O.(Gy.cm2/hr.MBq)
CCC                  Reference:Dale;Med.Phys.10(2);Mar1983.
CCC         RDF......Radial distribution function.
CCC         SUM......Sum of all doses for each point.
CCC         DESDOS...Desired dose passed from CALC.
CCC         DOS......Desired dose if uniform over all NINTP.
CCC
CCC    Passed common block allowing control of source type.
C
      COMMON/ISO/NTYP,PNTINT,NINTP,ACT,DESDOS,NSRCE,MTYP,NDIM
      COMMON/DOS/SUM
C
CCC    Declarations
C
      INTEGER IFLAG,N,NTYP,NSRCE,NINTP,MTYP,NDIM
      DOUBLE PRECISION FC,XC(N)
      REAL DOSE
      REAL SUM(20),DSTSQ(100),DIST(100)
      REAL RDF,SDC,PNTINT(60),ACT(40),DESDOS(20)
      REAL XX(10)
C
CCC    Data for calculation of RDF.
C
      DATA AO/0.97987/,A1/0.079621/,A2/-0.079138/,A3/0.008326/
      DATA RDF/1./
      DATA SDC/3.636E-4/
C
CCC   For point sources of I-125 with DESDOS(I) equal at all NINTP.
CCC   Sums over all sources for each point.
CCC    First calculates value for DIST and SQDST.
C
2101  DO 2100 I=1,NINTP
      SUM(I)=0
      DO 2201 J=1,NSRCE
      IK=((I-1)*NSRCE)+J
      CALL DST3V(PNTINT,NSRCE,NINTP,NDIM,XX,XC,I,J,DSTSQ)
C
      DIST(IK)=SQRT(DSTSQ(IK))
C
CCC    Calculation of RDF using data and DIST values.
CCC
C     RDF=AO+A1*DIST(IK)+A2*DSTSQ(IK)+A3*DIST(IK)**3
      DOSE=ACT(J)*SDC*RDF/DSTSQ(IK)
      SUM(I)=SUM(I)+DOSE
2201  CONTINUE
2100  CONTINUE
      IF(NTYP.EQ.5) GOTO 5001
C
CCC    Calculation of FUNC the objective function.
C
      DOS=DESDOS(1)
      SMAL=SUM(1)
      BIG=SUM(1)
      DO 2500 I=1,NINTP
      IF(BIG.LT.SUM(I)) BIG=SUM(I)
      IF(SMAL.GT.SUM(I)) SMAL=SUM(I)
2500  CONTINUE
      FUNC=((BIG-DOS)**2+(SMAL-DOS)**2)
      FC=FUNC
C
5001  RETURN
C
      END
```

```
      SUBROUTINE FUNCT1(IFLAG,N,XC,FC)
C
CCC   This is a user supplied routine calculating values for
CCC   objective function for  E04-UAF.
CCC   Allows many choices of function calculation.
CCCCC
CCC   AUTHOR:W.I.D.RAE
CCCCC
CCC   VERSION:11/8/86
CCCCC
CCC    Declaration of variables.
CCC         IFLAG....Set to 0, if neg. then terminates immediately.
CCC         XC.......Contains point at which F(X) is required.
CCC         FC.......Value of objective function at point XC.
CCC         NTYP.....Passes to section of FUNCT1 for source type.
CCC         ITABLE...Integer array with sups used in 200e-6s units.
CCC         MFLAG....Integer flag to control cycling of E04UAF.
CCC         MCALL....The dummy number of total agregate calls done.
CCC         FMON.....Function convergence monitor.
CCC         F1.......Previous initial function value.
CCC         MITER....Agregate number of iterations done.
CCC         MCHCK....The number of calls allowed in each cycle.
C
CC     Passed common blocks allowing control of source type.
C
      COMMON/ISO/NTYP,PNTINT,NINTP,ACT,DESDOS,NSRCE,MTYP,NDIM
      COMMON/OPT/MCNTS,MCALL,MFLAG,MITER,MCHCK,XCD
      COMMON/DUMSUM/DUMDIS,DUMDOS
C
CCC    Declarations
C
      INTEGER IFLAG,N,NTYP,NSRCE,NINTP,MTYP,NDIM
      INTEGER ITABLE(7)
      DOUBLE PRECISION FC,XC(N),XCD(80),FMON,F1
      REAL PNTINT(60),ACT(40),DESDOS(20)
      REAL DUMDIS(4,20)
      REAL DUMDOS(4,20)
C
CCCC    Allows exernal control of calls with re-starts if desired, at
CCCC     intervals of MCHCK.
C
      IF(NTYP.GE.7) THEN
      CALL SUSP(ITABLE)
      NDIF=ITABLE(2)-ITABLE(1)
      IF(NDIF.LT.1200000) IFLAG=-1
      END IF
C
      IF(MCHCK.NE.0) THEN
      MFLAG=0
      MCALL=MCALL+1
      IF(MCALL.EQ.1) F1=FC
      IF(MCALL.EQ.MCNTS*MCHCK.AND.MCNTS.LT.6) THEN
      FMON=F1-FC
      F1=FC
      MCNTS=MCNTS+1
      IF(FMON.LT.FC) THEN
      MITER=MITER+1
      MFLAG=-1
      IFLAG=MFLAG
      IF(NTYP.LT.7) THEN
```

```
      DO 400 I=1,N
      XCD(I)=XC(I)
400   CONTINUE
      END IF
      GOTO 8000
      END IF
      END IF
      END IF
C
C      Selects type of function to be used.
C
950   IF(NTYP.EQ.1) THEN
C
      CALL FTEST(IFLAG,N,XC,FC)
C
      ELSEIF(NTYP.EQ.2) THEN
C
      CALL FTWO(IFLAG,N,XC,FC)
C
      ELSEIF(NTYP.EQ.3) THEN
C
      CALL FREE(IFLAG,N,XC,FC)
C
      ELSEIF(NTYP.EQ.4) THEN
C
      CALL FOUR(IFLAG,N,XC,FC)
C
      ELSEIF(NTYP.EQ.5) THEN
C
      CALL FIVE(IFLAG,N,XC,FC)
C
      ELSEIF(NTYP.EQ.6) THEN
C
      CALL FSIX(IFLAG,N,XC,FC)
C
      ELSEIF(NTYP.GT.6) THEN
C
      CALL FSEV(IFLAG,N,XC,FC)
C
      END IF
C
8000  CONTINUE
C
 8020  FORMAT(1X,' POSTFUNCT FC=',D12.4)
C
      RETURN
C
8050  END
```

```
      SUBROUTINE IGRATE(T,A,E2,RLEN)
C
CCC    This routine does integration over arcs of an ellipse for CON1.
CCC    It uses a Maclaurin series expansion summed to NSUM terms.
CCCCC
CCC    AUTHOR:W.I.D.RAE
CCCCC
CCC    VERSION:6/11/86
CCCCC
CCC    Declaration of variables,
C
      DOUBLE PRECISION T
      REAL A,E2,RLEN
      REAL G,E2N,ST,CT,H,CT2,CT1
C
CCC    Setting up of variables for integration.
C
      NSUM=8
      G=0.
      E2N=1.
      ST=SIN(SNGL(T))
      CT=COS(SNGL(T))
      CT2=CT*CT
      FAC=1.
C
CCC    Calculation of coefficients of the Maclaurin expansion of
CCC                Fn ( Z ) = ( 1 - Z ) ** .5
C
      DO 100 I=1,NSUM
      M=I-1
      IF(I.EQ.1) M=1
      FAC=FAC*(2.*M-1.)/(I*2)
      E2N=E2N*E2
      JMX=2*I
      CT1=CT
      H=T
C
CCC       Calc. of the integral from 0 to T of (Cos(t))**(2*I)
C
      DO 200 J=2,JMX,2
      H=H*(J-1)/J+(ST/J)*CT1
      CT1=CT1*CT2
C
200   CONTINUE
CCC
CCC    Summing of terms of Maclaurin series.
C
      G=G-H*E2N*FAC
C
100   CONTINUE
C
      RLEN=A*(T+G)
C
      RETURN
C
      END
```

```
      SUBROUTINE INELLI(TITLE,NREC,X,NELLI)
C
CCCCCCCCCC
C      This routine allows entry of data for parameterised problem.
C      It only takes the variables that may be changed as this problem
C        is more rigidly defined than previous optimisations.
C
C      AUTHOR:W.I.D.RAE
C
C      VERSION:4/11/86
C
C      Declaration of common block.
C
      COMMON/ISO/NTYP,PNTINT,NINTP,ACT,DESDOS,NSRCE,MTYP,NDIM
      COMMON/RATE/DRATE,TDOSE,CON
C
CCC
C      Declaration of variables:
C
      DOUBLE PRECISION X(80)
      REAL PNTINT(60),ACT(40),DESDOS(20)
      REAL DRATE,TDOSE,CON(40)
      INTEGER NELLI(5)
      INTEGER NN,J,I
      CHARACTER*20 TITLE
C
CCC
C      Declaration of unit number for output.
C
      DATA NOUT/6/
C
CCC
C      Presentation of variable values on the screen.
C
111   CALL PAG
C
      WRITE(NOUT,110)
110   FORMAT(1X,'   ***INELLI***',//,'    ************')
      WRITE(NOUT,100) TITLE,NREC
100   FORMAT(1X,//,'  Enter selection number only;',/
     1  1X,'NTYP..(1)',2X,'NELLI.(2)',/
     2  1X,'X est.(3)',2X,'ACT...(4)',/
     3  1X,'PNTINT(5)',2X,'EXIT..(6)',//
     4  1X,'  Title of file ;',A20,' Rec. no;',I2)
      WRITE(NOUT,199) NTYP,NELLI(1),(NELLI(J),J=2,NELLI(1)+1)
199   FORMAT(1X,'  NTYP =',I2,' There are ',I1,' ellipses with',/
     1 1X,4(I2,1X,','),//,'  Sources respectively')
C
      DO 180 I=1,NELLI(1)
      WRITE(NOUT,181) I
181   FORMAT(1X,'  Estimates of parameters for ellipse ',I1)
      DO 182 J=1,NELLI(I+1)
      WRITE(NOUT,183) J,X(J+(I-1)*10),ACT(J+(I-1)*10)
183   FORMAT(1X,'  For source ',I2,' T=',D9.3,' Act= ',F8.2)
182   CONTINUE
      DO 185 J=NELLI(I+1)+1,10
      X(J+(I-1)*10)=0.0+0
185   CONTINUE
C
      WRITE(NOUT,184) I,(X(40+(I-1)*9+K),K=1,9)
184   FORMAT(1X,'  Ellipse ',I1,' has Amplitudes ',3(D9.3,1X),/
     1 1X,'                 Phase shift',3(D9.3,1X),/
     2 1X,'                 Center at  ',3(D9.3,1X))
C
```

```
180     CONTINUE
C
C        Allows jump to point to input/change the values desired.
        READ(5,*,ERR=111) NOW
        GOTO(200,300,400,450,500,600)NOW
C
200     WRITE(NOUT,201) NTYP
201     FORMAT(1X,' NTYP=',I2,//,'  Do you want to change?Yes=1')
        READ(5,*,ERR=200) NCH
        IF(NCH.EQ.1) THEN
C
        WRITE(NOUT,202)
202     FORMAT(1X,'    Enter NTYP only in range 7 to 9')
        READ(5,*,ERR=200) NTYP
        IF(NTYP.LT.7.OR.NTYP.GT.9) THEN
        WRITE(NOUT,203)
203     FORMAT(1X,' NTYP out of range ! Please EXIT then COPY',/
     1 1X,'  to file then CHANGE MORE if you want to use this type')
        GOTO 111
        END IF
C
        END IF
C
        GOTO 111
C
300     CONTINUE
        WRITE(NOUT,310)
310     FORMAT(1X,' Enter number of ellipses to use.1...4')
        READ(5,*) NELLI(1)
C
        DO 330 I=1,NELLI(1)
        WRITE(NOUT,311) I
311     FORMAT(1X,' Enter number of sources on ellipse ',I1)
        READ(5,*) NELLI(I+1)
330     CONTINUE
C
        DO 331 I=NELLI(1)+1,4
        NELLI(I+1)=0
331     CONTINUE
        GOTO 111
C
400     CONTINUE
        WRITE(NOUT,410)
410     FORMAT(1X,' Do you want to change all ? Yes=1')
        READ(5,*) NA
        IF(NA.EQ.1) THEN
C
        DO 440 I=1,NELLI(1)
        DO 420 J=1,NELLI(I+1)
        WRITE(NOUT,421) I,J
421     FORMAT(1X,' On ellipse ',I1,' enter good est. for source ',I1,/
     1     1X,'    in degrees , it will be converted to radians')
        READ(5,*) X((I-1)*10+J)
        X((I-1)*10+J)=X((I-1)*10+J)*0.017453293
420     CONTINUE
        WRITE(NOUT,422) I
422     FORMAT(1X,' On ellipse ',I1,' enter amplitudes in each axis')
        READ(5,*) X((I-1)*9+41)
        READ(5,*) X((I-1)*9+42)
        READ(5,*) X((I-1)*9+43)
        WRITE(NOUT,423) I
423     FORMAT(1X,' On ellipse ',I1,' enter centers shift in each axis')
```

```
      READ(5,*) X((I-1)*9+47)
      READ(5,*) X((I-1)*9+48)
      READ(5,*) X((I-1)*9+49)
C
      WRITE(NOUT,424) I
424   FORMAT(1X,' On ellipse ',I1,' enter phase shift in each axis',/
     1       1X,' in degrees,it will be converted to radians')
      READ(5,*) X((I-1)*9+44)
      READ(5,*) X((I-1)*9+45)
      READ(5,*) X((I-1)*9+46)
C
      X((I-1)*9+44)=X((I-1)*9+44)*0.C17453293
      X((I-1)*9+45)=X((I-1)*9+45)*0.C17453293
      X((I-1)*9+46)=X((I-1)*9+46)*0.C17453293
C
440   CONTINUE
C
      GOTO 111
C
      ELSE
C
      WRITE(NOUT,460)
460   FORMAT(1X,'  On which ellipse will you make changes?')
      READ(5,*) NE
      NN=(NE-1)*9
462   WRITE(NOUT,461)
461   FORMAT(1X,'  what will you change? Sources..1',/
     1 1X,'                          Amplitudes...2',/
     2 1X,'                          Center shift.3',/
     3 1X,'                          Phase shift..4',/
     4 1X,'                          No more......5')
      READ(5,*,ERR=462) NW
      IF(NW.EQ.5) GOTO 111
      IF(NW.EQ.1) THEN
      WRITE(NOUT,465)
465   FORMAT(1X,'  Which source will you change?')
      READ(5,*) NS
      WRITE(NOUT,466) NS,NE
466   FORMAT(1X,' What is new est. for source',I2,' on ellipse',I1,/
     1  1X,' in degrees, it will be converted to radians.')
      READ(5,*) X((NE-1)*10+NS)
      X((NE-1)*10+NS)=X((NE-1)*10+NS)*0.017453293
C
      ELSEIF(NW.EQ.2) THEN
      WRITE(NOUT,470) NE
470   FORMAT(1X,' Enter est. for amplitudes of ellipse',I1,',all axes'
      READ(5,*) X(NN+41)
      READ(5,*) X(NN+42)
      READ(5,*) X(NN+43)
C
      ELSEIF(NW.EQ.4) THEN
      WRITE(NOUT,471) NE
471   FORMAT(1X,' Enter est. for phases of ellipse',I1,',all axes',/
     1       1X,'  in degrees, it will be converted to radians')
      READ(5,*) X(NN+44)
      READ(5,*) X(NN+45)
      READ(5,*) X(NN+46)
C
      X(NN+44)=X(NN+44)*0.017453293
      X(NN+45)=X(NN+45)*0.017453293
      X(NN+46)=X(NN+46)*0.017453293
C
```

```
      ELSE
      WRITE(NOUT,472) NE
472   FORMAT(1X,' Enter est. for center of ellipse',I1,',all axes')
      READ(5,*) X(NN+47)
      READ(5,*) X(NN+48)
      READ(5,*) X(NN+49)
C
      END IF
      GOTO 462
      END IF
450   CONTINUE
      WRITE(NOUT,601)
601   FORMAT(1X,'On which ellipse do you want to change activities?')
      READ(5,*) NE
      WRITE(NOUT,602)
602   FORMAT(1X,'   Do you want to change all, yes=1?')
      READ(5,*) NA
      IF(NA.EQ.1) THEN
      DO 620 J=1,NELLI(NE+1)
      WRITE(NOUT,603) NE,J
603   FORMAT(1X,' Enter new act. for ellipse',I1,' source ',I2)
      READ(5,*) ACT((NE-1)*10+J)
620   CONTINUE
      ELSE
C
      WRITE(NOUT,621)
621   FORMAT(1X,'   which source will you change?')
      READ(5,*) NS
      WRITE(NOUT,622) NE,NS
622   FORMAT(1X,' For ellipse ',I1,' enter new act for source',I2)
      READ(5,*) ACT((NE-1)*10+NS)
C
      END IF
      GOTO 111
500   CALL PAG
C
      IF(NINTP.GT.20) NC=20
      NC=NINTP
      WRITE(NOUT,241)
241   FORMAT(1X,' Int.Pnt.',/,'  Pnt.No.',4X,'X',8X,'Y',8X,'Z',6X,
     1  1X,' CESDCS')
      WRITE(NOUT,205)(I,PNTINT(I*3-2),PNTINT(I*3-1),PNTINT(I*3),
     1  CESDCS(I),I=1,NC)
205   FORMAT(20(2X,I2,4X,4(1X,F8.4),/))
206   WRITE(NOUT,270)
270   FORMAT(1X,'   Do you wish to change values? None..0',/,
     1         33X,' All...1',/,33X,' Some..2',/,33X,' No.pnt.3')
      READ(5,*,ERR=206) NCH
      IF(NCH.EQ.0) GOTO 111
      IF(NCH.EQ.1.OR.NCH.EQ.3) THEN
255   WRITE(NOUT,232)
232   FORMAT(1X,'Enter no. of points of interest.')
      READ(5,*,ERR=255) NINTP
      IF(NINTP.GT.20) GOTO 255
      IF(NCH.EQ.3) GOTO 500
C
C
      DO 535 I=1,NINTP
      WRITE(NOUT,233) I
233   FORMAT(1X,'Enter interest point',I3,' in (cm); X,Y,Z')
      READ(5,*) PNTINT(3*I-2)
      READ(5,*) PNTINT(3*I-1)
```

```
      READ(5,*) PNTINT(3*I)
      IF(I.EQ.1) THEN
      WRITE(NOUT,251) I
251   FORMAT(1X,' At PNTINT(1) Relative Dose Rate=1.',/
     1    1X,' Enter the desired TDF to be given to PNTINT(1)')
      DESDOS(1)=1.0
      READ(5,*) X(80)
      GOTO 535
      END IF
      WRITE(NOUT,234) I
234   FORMAT(1X,'Enter desired relative dose rate at PNTINT(',I2,')')
      READ(5,*) DESDOS(I)
535   CONTINUE
      GOTO 500
      END IF
C
      IF(NCH.EQ.2) THEN
240   WRITE(NOUT,235)
235   FORMAT(1X,'  Which point number do you want to change?')
      READ(5,*,ERR=240) IK
      WRITE(NOUT,243) IK
243   FORMAT(1X,'Enter interest point',I3,' in (cm); X,Y,Z')
      READ(5,*) PNTINT(3*IK-2)
      READ(5,*) PNTINT(3*IK-1)
      READ(5,*) PNTINT(3*IK)
      IF(IK.EQ.1) THEN
      WRITE(NOUT,245)
245   FORMAT(1X,'  Relative dose rate at PNTINT(1)=1.0',/
     1   1X,'  Enter desired TDF to be given here')
      DESDOS(1)=1.0
      READ(5,*) X(80)
      GOTO 500
      END IF
      WRITE(NOUT,244) IK
244   FORMAT(1X,'Enter desired relative dose rate at PNTINT',I2,
     1  1X,' in Gy/hr')
      READ(5,*) DESDOS(IK)
C
      GOTO 500
C
      END IF
600   CONTINUE
C
      RETURN
C
      END
```

```
      SUBROUTINE INPUT(TITL,TITLE,NREC,NOW,N,IPRINT,X,
     1                 STEPMX,ETA,IBOUND,RHO,XU,XL,IFAIL)
C
C     This subroutine reads input for use in CALC.
C
C     AUTHOR:W.I.D.RAE
C
C     VERSION:1/9/86
C
C
C     Declaration of variables as listed here below..
C          ETA......Specifies accuracy of linear minimisation.
C          F........Contains value of F(X) on exit.
C          RHO......Current value of parameter rho in Lagrangian.
C          STEPMX...Estimate of Euclidian distance to min.
C          CL( )....Array of dim >or= MRNGE,lower constraint bound.
C          CU( )....Upper bound on constraint.
C          C( )..... Array of dim >or= M, holds constraint value.
C          RLAM( )..Initial estimates of Lagrange multipliers.
C          XL( )....Array of dim >or= N, holds fixed lower bounds.
C          XU( )....Array contains fixed upper bounds of X( ).
C          X( )..... Array contains value of constrained minimum.
C          MEQ......Equality constraint number.
C          MINEQ....Inequality constraint number.
C          MRNGE....Number of range constraints.
C          M........MEQ+MINEQ+MRNGE.
C          N........Number of independent variables.
C          NX.......Integer =N+MINEQ+MRNGE.
C          NOUT.....Specifies unit for output of data.
C          I........Integer.
C          IBOUND...Type of bounds used see notes.
C          IFAIL....Indicator of type of failure in optimisation.
C          IPRINT...Regulates calling of AMONIT by E04UAF.
C          PNTINT...Points  of interest to calculate uniformity.
C          NTYP.....Type of isotope chosen to  regulate functior.
C          ACT......Activity of sources in group.
C          DESDOS...Desired dose of points of interest.
C          NINTP....Number of interest points.
C          MTYP.....Type of constraint function used.
C          NDIM.....Dimension of source variables.
C          XM,XV....Dummy variables for XL,XU respectively.
C
C     Declaration of common blocks.
C
      COMMON/ISO/NTYP,PNTINT,NINTP,ACT,DESDOS,NSRCE,MTYP,NDIM
      COMMON/MON/CL,CU,MEQ,MINEQ,MRNGE
      COMMON/CONCOM/CENTEL,AXEL
C
      DOUBLE PRECISION ETA,RHO,STEPMX
      DOUBLE PRECISION CL(20),CU(20),XL(80),XU(80)
      DOUBLE PRECISION X(80),RLAM(80),XM(80),XV(80)
      INTEGER MEQ,MINEQ,MRNGE,I,IBOUND,IFAIL,IPRINT
      INTEGER M,N
      INTEGER NOUT,NSRCE,NDIM
      INTEGER NTYP,NINTP,MTYP
      REAL PNTINT(60),ACT(40),DESDOS(20)
      REAL CENTEL(3),AXEL(2)
      CHARACTER*20 TITLE,TITL
C
C     Declaration of logical variables.Allows restart.
```

```
C
      LOGICAL LAMSET
C
CCC    DATA declaration of output unit numbers.
C
      DATA NOUT /6/
C
CCC    Menu of allowed changes with present values below it.
C
105   CALL PAG
      IF(NOW.EQ.2) GOTO 400
      WRITE(NOUT,206 ) TITL,NREC
206   FORMAT(1X,'***SELECTION MENU***',//,' ********************',//
     1 1X,'Enter selection number only',//
     2 1X,'NTYP..(1)       NSRCE.(2)       ',//
     3 1X,'ACT...(3)       CONSTR(4)',/
     4 1X,'IPRINT(5)       STEPMX(6)',/
     5 1X,'ETA...(7)       IBOUND(8)',/
     6 1X,'IFAIL.(9)       RHO...(10)      ',/
     7 1X,'NINTP.(11)      LAMSET(12)      Nomore(13)',/
     8 1X,//,'    Title: ',A20,'    Record no.: ',I3,//)
C
      WRITE(NOUT,207 ) NTYP,NSRCE,N,MEQ,MINEQ,MRNGE,
     1    IPRINT,STEPMX,ETA,IBOUND,RHO,NINTP,IFAIL,MTYP,NDIM
207   FORMAT(1X/,' NTYP=',I3,'; NSRCE=',I3,'; N=',I3,'; MEQ=',I3,/
     1 1X,'MINEQ=',I3,'; MRNGE=',I3,'; IPRINT=',I3,'; STEPMX=',D8.2,/
     2 1X,'ETA=',D8.2,'; IBOUND=',I1,'; RHO=',D8.2,'; NINTP=',I3,/
     3 1X,' IFAIL=',I1,'; MTYP=',I2,'; NDIM=',I2)
C
CCC    Presentation of range constraints.
C
      DO 505  I=1,MRNGE
      WRITE(NOUT,208 ) I,CU(I),CL(I)
208   FORMAT(1X,'Range const',I3,' has CU=',D8.2,';CL=',D8.2)
505   CONTINUE
C
CCC    Reads NSEL which allows specific changes of parameters.
C
      READ(5,*) NSEL
C
      GOTO(400,110,415,420,425,430,435,440,445,450,455,460,465)NSEL
C
CCC    Entry or change of data to optimise and regulate program.
C
400   WRITE(NOUT,209)
209   FORMAT(1X,' What type of source is this? 1..7')
      READ(5,*)NTYP
      IF(NOW.EQ.1) GOTO 105
      IF(NOW.NE.1) GOTO 410
C
110   CALL PAG
C
      DO 510  I=1,NSRCE
      WRITE(NOUT,210 ) I,ACT(I)
210   FORMAT(1X,'Source ',I2,' with activity ',F6.2,' has')
      DO 510  J=1,NDIM
      K=NDIM*(I-1)+J
C
CCC    Allows representation of bounds of each X(),
CCC     dependant on value of IBOUND.
C
      IF(IBOUND.EQ.0) THEN
```

```
      XM(K)=XL(K)
      XV(K)=XU(K)
      END IF
C
      IF(IBOUND.EQ.1) THEN
      XM(K)=-100000000
      XV(K)=100000000
      END IF
C
      IF(IBOUND.EQ.2) THEN
      XM(K)=0.0
      XV(K)=100000000
      END IF
C
      IF(IBOUND.EQ.3) THEN
      XM(K)=XL(1)
      XV(K)=XU(1)
      END IF
C
      WRITE(NOUT,211 ) XM(K),X(K),XV(K)
211   FORMAT(1X,1PD10.3,'<',1PD10.3,'<',1PD10.3)
510   CONTINUE
C
      WRITE(NOUT,212 )
212   FORMAT(1X,//,'  Do you want to change any values?Yes(1)')
      READ(5,*) NCH
      IF(NCH.NE.1.AND.NOW.EQ.1) GOTO 105
      IF(NCH.NE.1.AND.NOW.NE.1) GOTO 415
C
410   CONTINUE
      WRITE(NOUT,213)
213   FORMAT(1X,'How many sources are used?')
      READ(5,*) NSRCE
      WRITE(NOUT,214) NSRCE
214   FORMAT(1X,'NSRCE=',I3)
      WRITE(NOUT,215)
215   FORMAT(1X,'How many variables per source?')
      READ(5,*) NDIM
      N=NDIM*NSRCE
700   WRITE(NOUT,216) N
216   FORMAT(1X,'Number of variables is',I3)
705   DO 515 I=1,NSRCE
      DO 515 J=1,NDIM
      K=(I-1)*NDIM+J
      WRITE(NOUT,217) K,I
217   FORMAT(1X,'Enter est. (cm) for X(',I3,') in source ',I3)
      READ(5,*) X(K)
515   CONTINUE
      GOTO 110
C
415   IF(NSRCE.LE.1) THEN
      DO 520 I=1,NSRCE
      WRITE(NOUT,204) I
204   FORMAT(1X,'Enter activity (MBq) of source..',I3)
      READ(5,*) ACT(I)
520   CONTINUE
      ELSE
      WRITE(NOUT,201)
201   FORMAT(1X,'  Do you wish to optimise activities? Yes(1)')
      READ(5,*) NYES
      IF(NYES.EQ.1) THEN
      WRITE(NOUT,202)
```

```
      ******  INPUT  ******


202   FORMAT(1X,' Enter only max activity ;')
      READ(5,*) ACT(1)
      WRITE(NOUT,203)
203   FORMAT(1X,' Enter min activity ;')
      READ(5,*) ACT(2)
      DO 521 I=3,NSRCE,1
      ACT(I)=ACT(1)
521   CONTINUE
      READ(5,*) ACT(1)
      ELSE
      DO 522 I=1,NSRCE
      WRITE(NOUT,218) I
218   FORMAT(1X,'Enter activity (MBq) of source..',I3)
      READ(5,*) ACT(I)
522   CONTINUE
      END IF
      END IF
      IF(NOW.EQ.1) GOTO 105
C
420   WRITE(NOUT,220)
220   FORMAT(1X,' What set of constraints is to be used?1..6')
      READ(5,*,ERR=420) MTYP
      IF(MTYP.LE.0.OR.MTYP.GE.7) GOTO 420
C
      IF(MTYP.EQ.2) THEN
      DO 600 I=1,3
      WRITE(NOUT,601) I
601   FORMAT(1X,' Enter value for CENTEX(',I1,')')
      READ(5,*,ERR=420) CENTEL(I)
600   CONTINUE
C
      DO 602 I=1,2
      WRITE(NOUT,603) I
603   FORMAT(1X,' Enter axis of ellipse, axis(',I1,')')
      READ(5,*,ERR=420) AXEL(I)
602   CONTINUE
C
      END IF
C
606   WRITE(NOUT,219)
219   FORMAT(1X,'Enter no. of equality constraints')
      READ(5,*) MEQ
710   WRITE(NOUT,221)
221   FORMAT(1X,'Enter no. of inequality constraints')
      READ(5,*) MINEQ
715   WRITE(NOUT,222)
222   FORMAT(1X,'Enter no. of range constraints')
      READ(5,*) MRNGE
      DO 525 I=1,MRNGE
      WRITE(NOUT,223) I
223   FORMAT(1X,'Enter upper bound for range constraint C (',I3,')')
      READ(5,*) CU(I)
      WRITE(NOUT,224) I
224   FORMAT(1X,'Enter lower bound for range constraint C(',I3,')')
      READ(5,*) CL(I)
525   CONTINUE
      IF(NOW.EQ.1) GOTO 105
C
425   WRITE(NOUT,225)
225   FORMAT(1X,'Enter no. to regulate calls of AMONIT,eg.10')
      READ(5,*) IPRINT
      IF(NOW.EQ.1) GOTO 105
```

```
C
430     WRITE(NOUT,226)
226     FORMAT(1X,'Enter est. of Euclidian dist. (cm) to minimum.')
        READ(5,*) STEPMX
        IF(NOW.EQ.1) GOTO 105
C
435     WRITE(NOUT,227)
227     FORMAT(1X,'Enter required accuracy for linear min. 0.0<X<1.0')
        READ(5,*) ETA
        IF(NOW.EQ.1) GOTO 105
C
440     WRITE(NOUT,223)
228     FORMAT(1X,'Enter value for IBOUND,0=all supplied by user',/
     1  1X,'                          1=no bounds',/
     2  1X,'                          2=all of form 0<or=X',/
     3  1X,'                          3=all L equal & all U equal',/
     1  1X,' Enter no. 0,1,2,or3.')
        READ(5,*) IBOUND
        IF(IBOUND.EQ.0) GOTO 422
        IF(IBOUND.EQ.3) GOTO 422
        IF(NOW.EQ.1) GOTO 105
        GOTO 450
C
422     DO 530 I=1,N
        WRITE(NOUT,229) I
229     FORMAT(1X,'Enter upper bounds for variable',I3)
        READ(5,*) XU(I)
        WRITE(NOUT,230) I
230     FORMAT(1X,'Enter lower bounds for variable',I3)
        READ(5,*) XL(I)
        IF(IBOUND.EQ.3) GOTO 590
530     CONTINUE
590     IF(NOW.EQ.1) GOTO 105
C
450     WRITE(NOUT,231)
231     FORMAT(1X,'Enter value for Rho,eg.1.0 or 100.0')
        READ(5,*) RHO
        IF(NOW.EQ.1) GOTO 105
C
455     CALL PAG
        IF(NINTP.GT.20) NC=20
        NC=NINTP
        WRITE(NOUT,241)
241     FORMAT(1X,' Points of interest  ;',//,9X,'X',8X,'Y',8X,'Z')
        WRITE(NOUT,205)(PNTINT(I*3-2),PNTINT(I*3-1),PNTINT(I*3),I=1,NC)
205     FORMAT(20(3(1X,F8.4),/))
        WRITE(NOUT,270)
270     FORMAT(1X,'    Do you wish to change any values? Yes=1')
        READ(5,*) NCH
        IF(NCH.NE.1) GOTO 535
255     WRITE(NOUT,232)
232     FORMAT(1X,'Enter no. of points of interest.')
        READ(5,*,ERR=255) NINTP
        IF(NINTP.GT.20) GOTO 255
C
        DO 535 I=1,NINTP
        WRITE(NOUT,233) I
233     FORMAT(1X,'Enter interest point',I3,' in (cm); X,Y,Z')
        READ(5,*) PNTINT(3*I-2)
        READ(5,*) PNTINT(3*I-1)
        READ(5,*) PNTINT(3*I)
        WRITE(NOUT,234) I
```

```
          ******  INPUT  ******


234       FORMAT(1X,'Enter desired dose rate at PNTINT.(Gy/hr)',I2)
          READ(5,*) DESDOS(I)
535       CONTINUE
          IF(NOW.EQ.1) GOTO 105
C
445       WRITE(NOUT,235)
235       FORMAT(1X,'Set IFAIL =1 ; allow restart or 0 ; dump on err.')
          READ(5,310) IFAIL
310       FORMAT(I1)
          IF(NOW.EQ.1) GOTO 105
C
CC          Allows restart with reset of LAMSET and RLAM.
C
460       WRITE(NOUT,236)
236       FORMAT(1X,' If restarting set LAMSET=.TRUE. and',/
         1 1X,'  set RLAM to suggested values.',/
         2 1X,'  Enter 1 if LAMSET=.TRUE.')
          READ(5,*) LAM
          LAMSET=.FALSE.
          IF(LAM.EQ.1) LAMSET=.TRUE.
          M=MEQ+MINEQ+MRNGE
          IF(LAM.NE.1) GOTO 485
          DO 540 I=1,M
          WRITE(NOUT,237) I
237       FORMAT(1X,' Enter RLAM(',I2,')')
          READ(5,*) RLAM(I)
540       CONTINUE
485       CONTINUE
          IF(NOW.EQ.1) GOTO 105
C
CC          Sets TITLE=TITL for rewriting to file.
C
          TITLE=TITL
C
CC        DEBUG UNIT(6),INIT(NDIM,N)
CC        AT 100
C         TRACE ON
C         AT 1000
C         TRACE OFF
C
465       RETURN
C
          END
```

```
      SUBROUTINE INTPOL(X,Y,FF,F,A,B,VAL,NFAIL,M1,AM,XX,WORK,D,N1,IG1)
C
CCC    Calling subroutine for NAG*LIB.E01ACE, which uses bicubic
CCC      splines to interpolate between data points in a plane.
CCC    This Routine sets up arrays and array sizes to allow the call.
CCC
CCC    AUTHOR:W.I.D.RAE
CCC
CCC    VERSION:13/12/86
CCC
CCC    Declaration of  variables.
C
      REAL X(N1),Y(M1),F(N1,M1),FF(150),AM(IG1),XX(IG1),
     1           WORK(IG1),D(IG1),A,B,VAL,VALL
C
      INTEGER NFAIL,IFAIL,N1,M1,IG1
C
      IFAIL=1
C
CCC    Conversion of 1 dim array to two dim array.
C
      DO 60 I=1,N1
      DO 60 J=1,M1
      K=(I-1)*M1+J
      F(I,J)=FF(K)
60    CONTINUE
C
CCC    Call of bi-cubic spline interpolation routine.
C
      CALL E01ACE(A,B,X,Y,F,VAL,VALL,IFAIL,XX,WORK,
     1      AM,D,IG1,M1,N1)
C
CCC    Calc. of % difference between values calc'ed using 1st
CCC     X direction and using 1st Y direction.
C
      NFAIL=IFAIL
      IF(VAL.EQ.VALL) THEN
      DIF=0
      ELSE
      DIF=200*(VAL-VALL)/(VAL+VALL)
      END IF
C
CCC    Sets IFAIL if the routine fails in the calculation or if
CCC     the % difference between the axes values is greater than .5 %.
C
      IF(DIF.GT.1) NFAIL=NFAIL+5
C
      RETURN
C
      END
```

```
      SUBROUTINE LINPOL(CX1,CX2,CY1,CY2,MM,EX,WI,RES)
C
CC     This routine does linear interpolation to give interpolated
CC      values from Array MM for routine FUNCT1.
CC
CC      AUTHOR: W.I.C.RAE
CC
CC      VERSION: 4/9/86
CC
CC      Declaration of variables used.
CC      WI......Y-value at which function value needed.
CC      EX......X-value at which function value needed.
CC      MM......2-Dim array of function values to be interpolated.
CC      CX1,CX2.Scale factors for conversion to used step size.
CC      CY1,CY2.Scale factors as above in Y-axis.
CC      FAA,etc.Function values at points  AA,AB,etc.
CC      NX,NY...Scaled input values to nearest previous integer.
CC      NN......Passing integer value for  POLINT.
CC      XS,YS...Scaled input variables to matrix scale.
CC      POLINT..Linear interpolation function for 1-dim.
CC      SCALE...Scaling function for input conversion.
CC      RES.....The function value at the desired point
C
      REAL MM(101,101),CX1,CX2,CY1,CY2,EX,WI,POLINT,SCALE
      REAL RES,FAA,FAB,FBA,FBB,XS,YS,D,H
      INTEGER NX,NY
C
CC     Scaling  of input variables to matrix MM's scale.
C
      XS=SCALE(CX1,CX2,EX)
      YS=SCALE(CY1,CY2,WI)
C
CC     Conversion to integer value to allow direct access to MM.
C
      NX=IFIX(XS)
      NY=IFIX(YS)
C
CC        Setting of RES dependent on variable out of range.
C
      IF(NX.LT.1) THEN
        RES=MM(1,1)*EX*CX1/(1-CX2)
        GOTO 10
C
        ELSEIF(NX.GT.100) THEN
          RES=MM(101,101)*EX*CX1/(1C1-CX2)
          GOTO 10
C
        ELSEIF(NY.LT.1) THEN
          FAA=MM(NX,1)
          FAB=MM(NX+1,1)
          RES=POLINT(NX,XS,FAA,FAB)
          GOTO 10
C
        ELSEIF(NY.GT.100) THEN
          FAA=MM(NX,101)
          FAB=MM(NX+1,101)
          RES=POLINT(NX,XS,FAA,FAB)
          GOTO 10
C
        ELSE
```

```
C
CC          Accessing MM for values to use when interpolating.
C
            FAA=MM(NX,NY)
            FBB=MM(NX+1,NY+1)
            FBA=MM(NX,NY+1)
            FAB=MM(NX+1,NY)
C
CC          Linear interpolation in 1-Dim between two points,
CC        first in X-direction then in Y-direction.
C
            D=POLINT(NX,XS,FAA,FAB)
            H=POLINT(NX,XS,FBA,FBB)
            RES=POLINT(NY,YS,D,H)
C
          ENDIF
C
10        RETURN
C
          END
          REAL FUNCTION SCALE(E,F,G)
C
CC         Real function to scale G using F and E.
C
          REAL E,F,G
C
          SCALE=G*E+F
C
          RETURN
C
          END
          REAL FUNCTION POLINT(NN,B,F1,F2)
C
CC         Real function to do linear interpolation in 1-Dim,
CC          between values F1&F2 where B (between NN&NN+1) is
CC          the point at which a function is required and NN is
CC          the integer array point at which F1 occurs.
C
          REAL F1,F2,B
          INTEGER NN
C
          POLINT=(F2-F1)*(B-NN)+F1
C
          RETURN
C
          END
```

```
      SUBROUTINE OPENFL
C
C     This routine opens files for CALC2.
C
C     AUTHOR:W.I.D.RAE
C
C     VERSION:1/9/86
C
C
C      Declaration of variables as listed here below..
C           NDAT.....Number of data file, DATA3.
C           MPF......Number of print file, MYPT.
C           NCON.....Number of control file, CONT.
C           MDAT.....Number of data file used in UPDAT.
C           NPF......Number of data file for  tabular output.
C           NDOS.....Number of file used for interpolation tables.
C
C      DATA declaration of output unit numbers.
C
      DATA MPF /21/
      DATA NDAT /20/
      DATA NDOS /19/
      DATA MDAT /18/
      DATA NCON /22/
      DATA NPF/23/
C
C      Opening of files used in CALC2.
C
      OPEN(NDAT,ACCESS='DIRECT',STATUS='OLD',RECL=810,
     1     FILE='DATA4',RCDS=60)
C
      OPEN(MPF,FILE='MYPT',STATUS='OLD',ACCESS='SEQUENTIAL',
     1     FORM='FORMATTED',BLANK='ZERO')
C
      OPEN(NCON,FILE='CONT',STATUS='OLD',RECL=11,RCDS=1,
     1     ACCESS='DIRECT')
C
      OPEN(NDOS,FILE='DATFL',STATUS='OLD',ACCESS='DIRECT',
     1     RECL=200,RCDS=5)
C
      OPEN(MDAT,FILE='MTLK',STATUS='OLD',ACCESS='DIRECT',
     1     RECL=10403,RCDS=5)
C
      OPEN(NPF,FILE='NEWPF',STATUS='OLD',ACCESS='SEQUENTIAL',
     1     FORM='FORMATTED',BLANK='ZERO')
C
C      Rewind of sequential access file to start.
C
      REWIND(MPF)
C
      REWIND(NPF)
C
      RETURN
C
C
      END
```

```
      SUBROUTINE OPTELI(XTOL)
C
C      This subroutine controls parameters used in calling E04UAF
C        when optimisation of parameters on ellipse is being done.
C      It sets control variables and selects variables to optimise.
C
C      AUTHOR:W.I.D.RAE
C
C      VERSION:12/11/86
C
      COMMON/ISO/NTYP,PNTINT,NINTP,ACT,DESDOS,NSRCE,MTYP,NDIM
      COMMON/PARCON/X,NELLI,MELL,LL
      COMMON/LINCOM/MM,CX1,CX2,CY1,CY2
      COMMON/DUMSUM/DUMDIS,DUMDOS
      COMMON/MON/CL,CU,MEQ,MINEQ,MRNGE
C
C      Declaration of variables:
C
      DOUBLE PRECISION ETA,F,RHO,STEPMX,XTOL
      DOUBLE PRECISION CL(20),CU(20),C(50),W(9000),XL(80),XU(80)
      DOUBLE PRECISION X(80),RLAM(80)
      DOUBLE PRECISION T(40)
      REAL MM(101,101),DUMDIS(4,20),DUMDOS(4,20)
      REAL PNTINT(60),ACT(40),DESDOS(20)
      INTEGER MEQ,MINEQ,MRNGE,I,IBOUND,IFAIL,IPRINT,LCLU,LIW,LW
      INTEGER M,MAXCAL,N,NX,IW(160)
      INTEGER NELLI(5)
      LOGICAL LAMSET
C
C      Program called by E04UAF.
C
      EXTERNAL EC4WAY
C
C      Setting of parameters for call of E04UAF
C
      N=NELLI(1)*9
      DO 100 I=1,N
      T(I)=X(I+40)
100   CONTINUE
      NSRCE=NELLI(1)
      NDIM=9
      MEQ=0
      MINEQ=NELLI(1)*2
      MRNGE=1
      CL(1)=0.2D+0
      CU(1)=0.8D+0
      M=MEQ+MINEQ+MRNGE
      IPRINT=0
      NX=N+MINEQ+MRNGE
      MAXCAL=100*(N+5)*NX
      ETA=.0010+0
      STEPMX=50
      DO 200 I=1,NELLI(1)
      DO 300 II=1,3
      XL((I-1)*9+II)=10.D+0
      XL((I-1)*9+II)=0.D+0
300   CONTINUE
      DO 305 JJ=1,3
      XL((I-1)*9+JJ+3)=-3.141592653589D+0
      XL((I-1)*9+JJ+3)=+3.141592653589D+0
305   CONTINUE
      DO 310 KK=1,3
      XL((I-1)*9+KK+6)=5.D+0
      XL((I-1)*9+KK+6)=-5.D+0
310   CONTINUE
200   CONTINUE
      LCLU=40
      MTYP=8
      IFAIL=1
      IBOUND=0
      LAMSET=.FALSE.
      RHO=1.D+0
      LIW=160
      LW=9000
C
125   CALL EC4UAF(N,MEQ,MINEQ,MRNGE,M,EC4WAY,IPRINT,
     1 MAXCAL,ETA,XTOL,STEPMX,CL,CU,LCLU,IBOUND,
     2 XL,XU,LAMSET,T,RHO,RLAM,F,C,IW,LIW,W,LW,IFAIL)
C
      RETURN
C
      END
```

```
      SUBROUTINE OPTIM(N,X,C,F)
C
C     This routine does the calculations and optimisation for TEST.
C     It makes use of NAG*LIB. E04UAF
C     It uses FUNCT1 which calculates objective fn value.
C     It uses CON1 to calculate constraint function values.
C     Allows for various types of sources.
C
C     AUTHOR:W.I.D.RAE
C
C     VERSION:1/9/86
C
C
C      Declaration of variables as listed here below..
C          ETA......Specifies accuracy of linear minimisation.
C          F........Contains value of F(X) on exit.
C          RHO......Current value of parameter rho in Lagrangian.
C          STEPMX...Estimate of Euclidian distance to min.
C          CL( )....Array of dim >or= MRNGE,lower bound on constraint.
C          CU( )....Upper bound on constraint.
C          C( )......Array of dim >or= M, contains constraint value.
C          RLAM( )..Initial estimates of Lagrange multipliers.
C          W( )......Array of dim >or= p ,for workspace.
C          XL( )....Array of dim >or= N, contains fixed lower bounds.
C          XU( )....Array contains fixed upper bounds of X( ).
C          X( )......Array contains value of constrained minimum.
C          XTOL.....Accuracy required for solution.
C          MEQ......Equality constraint number.
C          MINEQ....Inequality constraint number.
C          MRNGE....Number of range constraints.
C          M........MEQ+MINEQ+MRNGE.
C          MAXCAL...Limits calls of FUNCT1&CON1 by E04UAF.
C          N........Number of independent variables.
C          NX.......Integer =N+MINEQ+MRNGE.
C          I........Integer.
C          IBOUND...Type of bounds used see notes.
C          IFAIL....Indicator of type of failure in optimisation.
C          IPRINT...Regulates calling of AMONIT by E04UAF.
C          LCLU.....Actual length of CL,CU declared in CON1.
C          LIW......Actual length of IW.
C          LW.......Actual length of W.
C          IW( )....Integer array dim >or= N+MINEQ+MRNGE+M+12.
C          PNTINT...Points  of interest to calculate uniformity.
C          NTYP.....Type of isotope chosen to  regulate function.
C          ACT......Activity of sources in group.
C          DESDOS...Desired dose of points of interest.
C          NINTP....Number of interest points.
C          MTYP.....Type of constraint function used.
C          NDIM.....Dimension of source variables.
C          SUM......Dose at interest points from all sources.
C
C      Declaration of common blocks.
C
      COMMON/ISO/NTYP,PNTINT,NINTP,ACT,DESDOS,NSRCE,MTYP,NDIM
      COMMON/MON/CL,CU,MEQ,MINEQ,MRNGE
      COMMON/DOS/SUM,TITLE
      COMMON/VARCON/STEPMX,ETA,IBOUND,RHO,RLAM,MAXCAL,XTOL,M,XU,XL
      COMMON/WORKS/IPRINT,IFAIL,LAMSET,LCLU,IW,LIW,W,LW
      COMMON/OPT/MCNTS,MCALL,MFLAG,MITER,MCHCK,XCD
      COMMON/LINCOM/MM,CX1,CX2,CY1,CY2
```

```
C
      DOUBLE PRECISION ETA,F,RHO,STEPMX,XTOL,FC
      DOUBLE PRECISION CL(20),CU(20),C(50),W(9000),XL(80),XU(80)
      DOUBLE PRECISION X(80),XCD(80),RLAM(80)
      DOUBLE PRECISION T(40)
      DOUBLE PRECISION X02AAF,DSQRT
      DOUBLE PRECISION GLNORM,COND,CNORM
      INTEGER MEQ,MINEQ,MRNGE,I,IBOUND,IFAIL,IPRINT,LCLU,LIW,LW
      INTEGER M,MAXCAL,N,NX,IW(160)
      INTEGER NSRCE,NDIM
      INTEGER NTYP,NINTP,MTYP
      REAL PNTINT(60),ACT(40),DESDOS(20),SUM(20)
      REAL MM(101,101),CX1,CX2,CY1,CY2
      LOGICAL POSDEF
      CHARACTER*20 TITLE
C
C       Program called by E04UAF.
C
      EXTERNAL EC4WAY
C
C       Declaration of logical variables.Allows restart.
C
      LOGICAL LAMSET
C
C       DATA declaration of output unit numbers.
C
      DATA NOUT/6/
      DATA MDAT/13/
      DATA MPF /21/
C
      IF(NTYP.EQ.3) READ(MDAT,REC=4) MM,CX1,CX2,CY1,CY2,SPECDC
C
100   CONTINUE
      MCALL=0
      MCNTS=1
      MITER=0
      M=MEQ+MINEQ+MRNGE
      NX=N+MINEQ+MRNGE
      MAXCAL=100*(N+5)*NX
      LCLU=40
      LIW=160
      LW=9000
      XTOL=100.0D+0*DSQRT(X02AAF(XTOL))
C
C          Call of subroutine to write present data to MPF.
C
400   IF(IFAIL.LT.0) IFAIL=1
      CALL WRITE(TITLE,STEPMX,ETA,IBOUND,IFAIL,RHO,IPRINT,
     1             XU,XL,N,X)
C
      KFLAG=0
C
      CALL TABLE(IFAIL,NITER,MCALL,NINTP,N,M,NTYP,MTYP,ETA,
     1   STEPMX,POSDEF,X,FC,C,PNTINT,ACT,GLNORM,
     2   CNORM,COND,RLAM,RHO,NF,NSRCE,TITLE,KFLAG,SUM,DESDOS,SPECDC,T)
C
C
115   CALL FUNCT1(IFLAG,N,X,FC)
C
C
      KFLAG=1
C
```

```
      ******  OPTIM  ******


        CALL TABLE(IFAIL,NITER,MCALL,NINTP,N,M,NTYP,MTYP,ETA,
     1    STEPMX,PCSDEF,X,FC,C,PNTINT,ACT,GLNORM,
     2    CNORM,COND,RLAM,RHO,NF,NSRCE,TITLE,KFLAG,SUM,DESDOS,SPECDC,T)
C
        WRITE(MPF,242)
242     FORMAT(1X,//,'  *** FUNCT1 CALL ***',//,'   ******************')
        WRITE(MPF,243) FC
243     FORMAT(1X,//,'  Initial value of FC=',D12.6,/)
        DO 565 I=1,NINTP
        WRITE(MPF,244) I,DESDOS(I),SUM(I)
244     FORMAT(1X,' At interest point',I3,' Desired dose=',F9.3,
     1 1X,'  Initial calculated dose=',F9.3,/)
565     CONTINUE
C
C        Calling from CPTIM of E04UAF ,the optimisation routine used.
C
C
125     CALL E04UAF(N,MEQ,MINEQ,MRNGE,M,E04WAY,IPRINT,
     1 MAXCAL,ETA,XTOL,STEPMX,CL,CU,LCLU,IBOUND,
     2 XL,XU,LAMSET,X,RHO,RLAM,F,C,IW,LIW,W,LW,IFAIL)
C
C
        IF(IFAIL.LT.0) THEN
        DO 570 I=1,N
        X(I)=XCD(I)
570     CONTINUE
        WRITE(NOUT,250) IFAIL,MCNTS,MITER,MCALL
        WRITE(MPF,250) IFAIL,MCNTS,MITER,MCALL
250     FORMAT(1X,' Ifail=',I1,' After ',I2,' checks and ',I2,/
     1 1X,'       restarts and a total of ',I3,' calls of FUNCT1')
        GOTO 4CO
        END IF
C
        RETURN
C
        END
```

```
      SUBROUTINE OPTPAR(F,XTOL)
C
CC     This subroutine controls parameters used in calling E04UAF
CC        when optimisation of parameters on ellipse is being done.
CC     It sets control variables and selects variables to optimise.
CC
CC     AUTHOR:W.I.D.RAE
CC
CC     VERSION:12/11/86
C
      COMMON/ISO/NTYP,PNTINT,NINTP,ACT,DESDOS,NSRCE,MTYP,NDIM
      COMMON/PARCON/X,NELLI,MELL,LL
      COMMON/LINCOM/MM,CX1,CX2,CY1,CY2
      COMMON/DUMSUM/DUMDIS,DUMDOS
      COMMON/MON/CL,CU,MEQ,MINEQ,MRNGE
C
CC     Declaration of variables:
C
      DOUBLE PRECISION ETA,F,RHO,STEPMX,XTOL
      DOUBLE PRECISION CL(20),CU(20),C(50),W(9000),XL(80),XU(80)
      DOUBLE PRECISION X(80),RLAM(80)
      DOUBLE PRECISION T(40)
      REAL MM(101,101),DUMDIS(4,20),DUMDOS(4,20)
      REAL DESDOS(20),ACT(40),PNTINT(60)
      INTEGER MEQ,MINEQ,MRNGE,I,IBOUND,IFAIL,IPRINT,LCLU,LIW,LW
      INTEGER M,MAXCAL,N,NX,IW(160)
      INTEGER NELLI(5),MELL,LL
      LOGICAL LAMSET
C
CC     Program called by E04UAF.
C
      EXTERNAL E04WAY
C     Setting of control parameters for call of E04UAF.
C
      DO 100 I=1,NELLI(LL+1)
      T(I)=X(I+(LL-1)*10)
100   CONTINUE
      N=NELLI(LL+1)
      NSRCE=N
      MEQ=0
      MINEQ=NELLI(LL+1)
      MRNGE=1
      CL(1)=0.2D+0
      CL(1)=0.8D+0
      M=MEQ+MINEQ+MRNGE
      IPRINT=0
      NX=N+MINEQ+MRNGE
      MAXCAL=100*(N+5)*NX
      ETA=.0010+0
      STEPMX=50
      XL(1)=-12.6D+0
      XL(1)=+12.6D+0
      LCLU=40
      MTYP=7
      IFAIL=1
      IBOUND=3
      LAMSET=.FALSE.
      RHO=1.D+0
      LIW=160
      LW=9000
C
125   CALL E04UAF(N,MEQ,MINEQ,MRNGE,M,E04WAY,IPRINT,
     1 MAXCAL,ETA,XTOL,STEPMX,CL,CU,LCLU,IBOUND,
     2 XL,XU,LAMSET,T,RHO,RLAM,F,C,IW,LIW,W,LW,IFAIL)
C
      RETURN
C
      END
```

```
      SUBROUTINE OUTPUT(IFAIL,TITLE,F,M,N,X,C,NOUT,MPF,RHO,RLAM)
C
C     This routine writes the final output for test when NTYP<7.
C
C     ALTHOR:W.I.D.RAE
C
C     VERSION:1/9/86
C
C
C      Declaration of variables as listed here below..
C          F.........Contains value of F(X) on exit.
C          RHO.......Current value of parameter rho in Lagrangian.
C          STEPMX....Estimate of Euclidian distance to min.
C          CL( ).....Array of dim >or= MRNGE,lower bound on constraint.
C          CU( ).....Upper bound on constraint.
C          C( ).....Array of dim >or= M, contains constraint value.
C          RLAM( )..Initial estimates of Lagrange multipliers.
C          XL( ).....Array of dim >or= N, contains fixed lower bounds.
C          X( ).....Array contains value of constrained minimum.
C          XTOL......Accuracy required for solution.
C          MEQ.......Equality constraint number.
C          MINEQ.....Inequality constraint number.
C          MRNGE.....Number of range constraints.
C          M.........MEQ+MINEQ+MRNGE.
C          N.........Number of independent variables.
C          NX........Integer =N+MINEQ+MRNGE.
C          NOUT......Specifies unit for output of data.
C          I.........Integer.
C          IFAIL.....Indicator of type of failure in optimisation.
C          PNTINT....Points  of interest to calculate uniformity.
C          NTYP......Type of isotope chosen to  regulate function.
C          ACT.......Activity of sources in group.
C          DESDOS....Desired dose of points of interest.
C          NINTP.....Number of interest points.
C          MTYP......Type of constraint function used.
C          NDIM......Dimension of source variables.
C          SUM.......Dose at interest points from all sources.
C
C      Declaration of common blocks.
C
      COMMON/ISO/NTYP,PNTINT,NINTP,ACT,DESDOS,NSRCE,MTYP,NDIM
      COMMON/MON/CL,CU,MEQ,MINEQ,MRNGE
C
      DOUBLE PRECISION F,RHO
      DOUBLE PRECISION CL(20),CU(20),C(50)
      DOUBLE PRECISION RLAM(80),X(80)
      DOUBLE PRECISION CNORM,GLNORM,COND,ETA,STEPMX
      DOUBLE PRECISION T(40)
      INTEGER MEQ,MINEQ,MRNGE,I,IFAIL
      INTEGER M,N
      INTEGER NOUT,NSRCE,NDIM
      INTEGER NTYP,NINTP,MTYP
      REAL ACT(40),SUM(20),DESDOS(20),PNTINT(60)
      LOGICAL POSDEF
      CHARACTER*20 TITLE
C
C      Final presentation of results.
C      Since IFAIL set=1 before entry ,check if non-zero.
C
720   IF(IFAIL.NE.0) THEN
```

```
      ******  OUTPUT  ******


      KFLAG=5
      WRITE(NOUT,245) IFAIL
C     WRITE(MPF,245) IFAIL
245   FORMAT(///,' IFAIL= ',I3,' ERROR EXIT')
      END IF
      IF(IFAIL.EQ.1) GOTO 1000
130   CALL PAG
      WRITE(NOUT,246 )
C     WRITE(MPF,246 )
246   FORMAT(1X,//,1X,' ***RESULTS***',//,'  ************')
      WRITE(NOUT,247 ) TITLE,F
      WRITE(MPF,247 ) TITLE,F
247   FORMAT(//,' TITLE : ',A20,///,' FUNCT VALUE= ',D12.4)
C
C      Presentation of optimal results.
C
      IF(IFAIL.NE.2) THEN
      KFLAG=4
      NSC=0
      DO 570 I=1,N,NDIM
      NSC=NSC+1
      DO 570 J=1,NDIM
      WRITE(NOUT,248 ) NSC,J,X(I+J-1)
      WRITE(MPF,248 ) NSC,J,X(I+J-1)
248   FORMAT(1X,' Source ',I3,' has X(',I3,') =',1PD12.4)
570   CONTINUE
      END IF
C
C      Non optimal results, suggest restart.
C
      IF(IFAIL.EQ.2) THEN
      WRITE(NOUT,254 )
C     WRITE(MPF,254 )
254   FORMAT(1X,//,' IFAIL=2')
      NSC=0
      DO 575 I=1,N,NDIM
      NSC=NSC+1
      DO 575 J=1,NDIM
      WRITE(NOUT,255 ) NSC,J,X(I+J-1)
      WRITE(MPF,255 ) NSC,J,X(I+J-1)
255   FORMAT(1X,//,' Srce(',I3,') has X(',I3,')=',F14.5)
575   CONTINUE
      END IF
C
C      Presentation of constraint values on exit.
C
      WRITE(NOUT,249 )
C     WRITE(MPF,249 )
249   FORMAT(//,' Constraints are')
      DO 580  I=1,M
      WRITE(NOUT,250 ) I,C(I)
C     WRITE(MPF,250 ) I,C(I)
250   FORMAT(3H C(, I1, 4H)=  , D12.5)
580   CONTINUE
C
      IF(IFAIL.NE.2) GOTO 1000
      WRITE(NOUT,251 ) RHO
C     WRITE(MPF,251 ) RHO
251   FORMAT(/,1X,' If restarting set RHO= ',D12.4,'and RLAM = ')
      WRITE(NOUT,252 ) (RLAM(I),I=1,M)
C     WRITE(MPF,252 ) (RLAM(I),I=1,M)
252   FORMAT(1H , D12.4)
C
C      Call of TABLE to give tabulated results.
C
1000  CALL TABLE(IFAIL,NITER,MCALL,NINTP,N,M,NTYP,MTYP,ETA,
     1  STEPMX,POSDEF,X,F,C,PNTINT,ACT,GLNORM,CNORM,
     2  COND,RLAM,RHO,NF,NSRCE,TITLE,KFLAG,SUM,DESDOS,SPECDC,T)
C
      RETURN
C
      END
```

```
      SUBROUTINE PARA(XW,T,NSRCE,NDIM)
C
CC     This subroutine sets variables XC from parameterised
CC      variable T.
CC     It gives spatial positions and orientations.
CC
CC     AUTHOR: W.I.D.RAE
CC
CC     VERSION:16/10/86
CC
CC      Common block passed
C
      COMMON/PARCON/X,NELLI,MELL,LL
C
CC      Declaration of variables;
C
      INTEGER NELLI(5),MELL,LL
      DOUBLE PRECISION XW(5),T
      DOUBLE PRECISION X(80)
      REAL XA,Y,Z,TANG
      REAL XE(9)
C
CC     Calc. of the spatial coordinates at a point T.
C
      L=40+(LL-1)*9
C
      DO 10 J=1,9
      XE(J)=SNGL(X(L+J))
10    CONTINUE
C
      XW(1)=DBLE(XE(1)*SIN(XE(4)+T)+XE(7))
      XW(2)=DBLE(XE(2)*SIN(XE(5)+T)+XE(8))
      XW(3)=DBLE(XE(3)*SIN(XE(6)+T)+XE(9))
C
CC     Tangent vector , first partial derivative XW(i),i=1,3
C
      XA=XE(1)*COS(T+XE(4))
      Y=XE(2)*COS(T+XE(5))
      Z=XE(3)*COS(T+XE(6))
      TANG=SQRT(XA*XA+Y*Y+Z*Z)
C
      IF(TANG.EQ.0.) TANG=0.0001
C
CC     Calc of angle of depression from Z-axis.
C
      XW(4)=DBLE(ACOS(Z/TANG))
      SS=SIN(SNGL(XW(4)))
      YTS=Y/(TANG*SS)
C
      IF(SS.EQ.0.) SS=0.0001
      IF(YTS.GT.1.0) YTS=1.0
      IF(YTS.LT.-1.0) YTS=-1.0
C
CC     Calc. of angle of rotation from x-axis of projection
C       of the vector onto X-Y plane.
      IF(Y.EQ.0.) THEN
      XW(5)=0.
      GOTO 30
      END IF
C
      XW(5)=DBLE(ASIN(YTS))
C
30    IF(XA.LT.0.) XW(5)=3.1415926535-XW(5)
C
      RETURN
C
      END


      SUBROUTINE PAG
C
CC     This routine moves cursor to top of screen and clears it.
CC      It is for ICL terminals (catalogue serial number 6402/00)
CC      It contains only CHAR functions.
CC
CC      AUTHOR:W.I.D.RAE
CC
CC      VERSION:13/12/86
C
      WRITE(6,140)CHAR(30),CHAR(27)//CHAR(89)
140   FORMAT(1X,A1,A2)
C
      RETURN
C
      END
```

```
      SUBROUTINE PARCPT(N,C,F)
C
C     This routine uses parametric equations for optimisation.
C     This routine does the calculations and optimisation for TEST.
C     It makes use of NAG*LIB. E04UAF
C     EC4UAF uses FUNCT1 which calculates objective fn value.
C     EC4UAF uses CON1 to calculate constraint function values.
C     May allow for various types of sources.
C
C     AUTHOR:W.I.D.RAE
C
C     VERSION:5/11/86
C
C
C      Declaration of variables as listed here below..
C          ETA......Specifies accuracy of linear minimisation.
C          F........Contains value of F(X) on exit.
C          RHO......Current value of parameter rho in Lagrangian.
C          STEPMX...Estimate of Euclidian distance to min.
C          CL( )....Array of dim >or= MRNGE,lower bound on constraint.
C          CU( )....Upper bound on constraint.
C          C( )..... Array of dim >or= M, contains constraint value.
C          RLAM( )..Initial estimates of Lagrange multipliers.
C          W( )..... Array of dim >or= p ,for workspace.
C          XL( )....Array of dim >or= N, contains fixed lower bounds.
C          XU( )....Array contains fixed upper bounds of X( ).
C          X( )..... Array contains value of constrained minimum.
C          XTOL.....Accuracy required for solution.
C          MEQ......Equality constraint number.
C          MINEQ....Inequality constraint number.
C          MRNGE....Number of range constraints.
C          M........MEQ+MINEQ+MRNGE.
C          MAXCAL...Limits calls of FUNCT1&CON1 by E04UAF.
C          N........Number of independent variables.
C          I........Integer.
C          IBOUND...Type of bounds used see notes.
C          IFAIL....Indicator of type of failure in optimisation.
C          IPRINT...Regulates calling of AMONIT by E04UAF.
C          LCLU.....Actual length of CL,CU declared in CON1.
C          LIW......Actual length of IW.
C          LW.......Actual length of W.
C          IW( )....Integer array dim >or= N+MINEQ+MRNGE+M+12.
C          PNTINT...Points  of interest to calculate uniformity.
C          NTYP.....Type of isotope chosen to  regulate function.
C          ACT......Activity of sources in group.
C          DESDOS...Desired dose of points of interest.
C          NINTP....Number of interest points.
C          MTYP.....Type of constraint function used.
C          NDIM.....Dimension of source variables.
C          SUM......Dose at interest points from all sources.
C
C      Declaration of common blocks.
C
      COMMON/ISO/NTYP,PNTINT,NINTP,ACT,DESDOS,NSRCE,MTYP,NDIM
      COMMON/MON/CL,CU,MEQ,MINEQ,MRNGE
      COMMON/DOS/SUM,TITLE
      COMMON/VARCON/STEPMX,ETA,IBOUND,RHO,RLAM,MAXCAL,XTOL,M,XU,XL
      COMMON/WORKS/IPRINT,IFAIL,LAMSET,LCLU,IW,LIW,W,LW
      COMMON/OPT/MCNTS,MCALL,MFLAG,MITER,MCHCK,XCD
      COMMON/LINCOM/MM,CX1,CX2,CY1,CY2
```

```
      ******  PAROPT  ******


      COMMON/PARCON/X,NELLI,MELL,LL
C
C      Declaration of variables:
C
      DOUBLE PRECISION ETA,F,RHO,STEPMX,XTOL
      DOUBLE PRECISION CL(20),CU(20),C(50),W(9000),XL(80),XU(80)
      DOUBLE PRECISION XX(80),X(80),XCD(80),RLAM(80)
      DOUBLE PRECISION T(40)
      DOUBLE PRECISION X02AAF,DSQRT
      DOUBLE PRECISION GLNORM,COND,CNORM
      INTEGER MEQ,MINEQ,MRNGE,I,IBOUND,IFAIL,IPRINT,LCLU,LIW,LW
      INTEGER M,MAXCAL,N,IW(160)
      INTEGER NELLI(5)
      INTEGER MELL,LL
      INTEGER NSRCE,NDIM
      INTEGER NTYP,NINTP,MTYP
      REAL PNTINT(60),ACT(40),DESDOS(20),SUM(20)
      REAL MM(101,101),CX1,CX2,CY1,CY2
      LOGICAL POSDEF,LAMSET
      CHARACTER*20 TITLE
C
C      DATA declaration of output unit numbers.
C
      DATA NOUT/6/
      DATA MDAT/18/
      DATA MPF /21/
C
C      Read of file MTLK to get lookup table for calculation of
C       dose by FSEV.
C
      READ(MDAT,REC=4) MM,CX1,CX2,CY1,CY2,SPECDC
C
C      Setting of values required in both OPTELI and OPTPAR.
C
      DO 101 I=1,80
      XX(I)=X(I)
101   CONTINUE
C
      N=NELLI(1)*9
      DO 102 I=1,N
      T(I)=X(I+40)
102   CONTINUE
C
100   CONTINUE
      MCALL=0
      MCNTS=1
      MITER=0
      LCLU=40
      LIW=160
      LW=9000
      XTOL=10000.0D+0*DSQRT(X02AAF(XTOL))
C
C         Call of subroutine to write present data to NPF.
C
400   IF(IFAIL.LT.0) THEN
      IFAIL=1
      ELSE
      IFAIL=0
      ENDIF
C
      KFLAG=0
C
```

```
      CALL TABLE(IFAIL,NITER,MCALL,NINTP,N,M,NTYP,MTYP,ETA,
     1   STEPMX,POSDEF,XX,F,C,PNTINT,ACT,GLNORM,
     2  CNORM,COND,RLAM,RHO,NF,NSRCE,TITLE,KFLAG,SUM,DESDOS,SPECDC,T)
C
      LT=LL
      LL=0
      MELL=-1
      CALL FUNCT1(IFLAG,N,T,F)
      MELL=0
C
      KFLAG=1
C
      CALL TABLE(IFAIL,NITER,MCALL,NINTP,N,M,NTYP,MTYP,ETA,
     1   STEPMX,POSDEF,XX,F,C,PNTINT,ACT,GLNORM,
     2  CNORM,COND,RLAM,RHO,NF,NSRCE,TITLE,KFLAG,SUM,DESDOS,SPECDC,T)
C
      LL=LT
      NCUM=NTYP
C
C       Calling from CALC of E04UAF ,the optimisation routine used.
C
      ITS=0
300   CONTINUE
      IF(NTYP.EQ.7) THEN
C
      MELL=1
      LL=1
C
C       This section cyclically calls routines to optimise
C         positions and ellipse variables alternately.
C
      DO 200 I=1,NELLI(1)
      KFLAG=1
      CALL TABLE(IFAIL,NITER,MCALL,NINTP,N,M,NTYP,MTYP,ETA,
     1   STEPMX,POSDEF,XX,F,C,PNTINT,ACT,GLNORM,
     2  CNORM,COND,RLAM,RHO,NF,NSRCE,TITLE,KFLAG,SUM,DESDOS,SPECDC,T)
      CALL OPTPAR(F,XTOL)
      MELL=MELL+10
      LL=LL+1
      KFLAG=4
C
      IF(IFAIL.EQ.0) KFLAG=5
C
      CALL TABLE(IFAIL,NITER,MCALL,NINTP,N,M,NTYP,MTYP,ETA,
     1   STEPMX,POSDEF,X,F,C,PNTINT,ACT,GLNORM,
     2  CNORM,COND,RLAM,RHO,NF,NSRCE,TITLE,KFLAG,SUM,DESDOS,SPECDC,T)
200   CONTINUE
C
      LL=0
      IF(ITS.EQ.1) GOTO 301
C
      CALL OPTELI(XTOL)
C
      ITS=ITS+1
      GOTO 300
C
      ELSEIF(NTYP.EQ.8) THEN
C
C       This calls a routine to do optimisation of parameters of source
C
      CALL OPTPAR(F,XTOL)
C
```

```
      ELSEIF(NTYP.EQ.9) THEN
C
C      This calls a routine to do optimisation of ellipse variables
C
      LL=0
C
      CALL OPTELI(XTOL)
C
      ITS=1
      NCUM=NTYP
      NTYP=7
      GOTO 300
C
      END IF
C
301   CONTINUE
      NTYP=NCUM
C
C      Final call of TABLE to write results.
C
C
      KFLAG=4
C
      IF(IFAIL.EQ.0) KFLAG=5
C
      CALL TABLE(IFAIL,NITER,MCALL,NINTP,N,M,NTYP,MTYP,ETA,
     1   STEPMX,POSDEF,X,F,C,PNTINT,ACT,GLNORM,
     2  CNORM,COND,RLAM,RHO,NF,NSRCE,TITLE,KFLAG,SUM,DESDOS,SPECDC,T)
C
      IF(IFAIL.LT.0) THEN
      WRITE(NOUT,250) IFAIL,MCNTS,MITER,MCALL
      WRITE(MPF,250) IFAIL,MCNTS,MITER,MCALL
250   FORMAT(1X,'Ifail=',I1,' After ',I2,' checks and ',I2,/
     1 1X,'        restarts and a total of ',I3,' calls of FUNCT1')
      GOTO 400
      END IF
C
      RETURN
C
      END
```

```
      SUBROUTINE PHISIN(XC,PNTINT,NSRCE,NINTP,I,J,DSTSQ,SINPHI)
C
CC     This subroutine calculates the value of sine phi, the
CC       angle between the Jth seeds axis and the perpendicular
CC       bisector of that axis from the Ith point of interest.
CC
CC     AUTHOR:W.I.D.RAE
CC
CC     VERSION:24/8/86
CC
CC     Declaration of variables as below,
CC          XC......Current value of X.
CC          PNTINT..Point of current interest.
CC          I,J.....Control variables used in calculation.
CC          DSTSQ...Distance between source and interest point.
CC          SINPHI..Sine of relative angle of seeds axis.
CC          UR......Dot product of UVEC and RVEC.
CC          UVEC....Unit vector in direction of seed axis.
CC          RVEC....Vector between source and interest point.
CC          J5,J4,JO.Dummy counters.
CC          C4,C5,CO.Dummy cos variables.
CC          S4,S5,SO.Dummy sin variables.
C
      REAL SINPHI,DSTSQ(NINTP*NSRCE),PNTINT(NINTP*3)
      REAL UR,UVEC(3),RVEC(3)
      INTEGER I,J
      DOUBLE PRECISION XC(NSRCE*5)
C
      J5=J*5
      J4=J5-1
      IC=I*3-3
      JC=J5-5
C
      C4=COS(SNGL(XC(J4)))
      C5=COS(SNGL(XC(J5)))
      S4=SIN(SNGL(XC(J4)))
      S5=SIN(SNGL(XC(J5)))
C
CC     Calc. of the unit vector in the direction of the seeds axis.
C
      UVEC(1)=C5*S4
      UVEC(2)=S4*S5
      UVEC(3)=C4
C
CC       Calculation of the dot product of UVEC&RVEC.
C
      UR=0.
      DO 20 K=1,3
      RVEC(K)=XC(JO+K)-PNTINT(IO+K)
      UR=UVEC(K)*RVEC(K)+UR
20    CONTINUE
C
      FA=UR*UR/DSTSQ((I-1)*NSRCE+J)
      IF(FA.GE.1) FA=1.
C
CC       Calculation of the SIN of PhI by a cross product property.
C
      SINPHI=SQRT(1.-FA)
C
      RETURN
C
      END
```

```
      SUBRCUTINE SUBLIN
C
C     This subroutine accepts new data for UPDAT and allows
C       linear interpolation at a point around a source
C         defined by X in cm and Y in degrees,
C     by LINPOL .
C
C     AUTHOR :W.I.D.RAE
C
C     VERSION :13/12/86
C
C     Declaration of variables.
C       RES......Result of linear interpolation by LINPOL.
C       EX.......Input variable in X axis, 1/r2.
C       WI.......Input variable in Y axis, radians.
C       CX1,etc..Scale factors passed to LINPOL.
C       MATFIL...File for interpolated matrix from E01ACE.
C       DRCF.....Conversion factor from degrees to radians.
C       MM.......File for array of cubic spline fitted values.
C       NOREC....Number of file to use.
C       NYES.....Control variable allowing interpolation.
C       NOUT.....Number of output unit.
C       MDAT.....Number of interpolated data file for output.
C
      INTEGER NYES,NOREC,NOUT,MDAT
      REAL DRCF,MM(101,101),CX1,CX2,CY1,CY2,RES,EX,WI
C
C     Data for I/O units.
C
      DATA MDAT/18/
      DATA NOUT/6/
C
C     Data for DEGREES to RADIANS conversion.
C
      DATA DRCF/0.01745329252/
C
C     Asks for and receives data from the keyboard.
C
20    CALL PAG
C
      WRITE(NOUT,30 )
30    FORMAT(1X,//,'  Do you wish to interpolate linearly from',/
     1  1X,'     an already created table? Yes(1)')
      READ(5,*) NYES
C
C     Allows exit.
      IF(NYES.NE.1) GOTO 10
C
      WRITE(NOUT,40 )
40    FORMAT(1X,//,'  Which record made the table for use now?')
      READ(5,*) NOREC
      READ(MDAT,REC=NOREC) MM,CX1,CX2,CY1,CY2,SPECDC
C
      WRITE(NOUT,50 )
50    FORMAT(1X,//,'  Enter value in cm. for interpolation in X.')
      READ(5,*) EX
      EX=1/EX
C
      WRITE(NOUT,55 )
55    FORMAT(1X,//,'  Enter value in deg. for interpol. in Y.')
      READ(5,*) WI
      WI=SIN(WI*DRCF)
C
C     Call of LINPOL to do the linear interpolation off table
C       for values of 1/(d) and SIN(PHI).
C
      CALL LINPOL(CX1,CX2,CY1,CY2,MM,EX,WI,RES)
C
C     Writes output to the screen
C
      WRITE(NOUT,60 ) EX,WI,RES
60    FORMAT(1X,//,' At EX=',F6.2,'/cm & WI=',F6.2,'rad ,F=',E10.4)
C
      PAUSE 'PRESS RETURN TO CONTINUE'
C
C     Allows restart.
C
      GOTO 20
C
10    RETURN
      END
```

```
      SUBROUTINE TABLE(IFAIL,NITER,MCALL,NINTP,N,M,NTYP,MTYP,ETA,
     1         STEPMX,POSDEF,XC,FC,CC,PNTINT,ACT,GLNORM,CNORM,
     2    COND,RLAM,RHO,NF,NSRCE,TITLE,KFLAG,SUM,DESDOS,SPECDC,T)
C
CC       This routine prints a table of current values to file NEWPF.
CC
CC       AUTHOR:W.I.D.RAE
CC
CC       VERSION:22/9/86
CC
CC       Declaration of variables used in tabulation columns.
CC             VARC....Variable column title.
CC             DRATE...Dose rate achieved at PNTINT(1)
CC             TIME....Time to achieve the desired TDF
CC             DRFIN...Final dose rate achieved in GyEquiv.
CC             ...COL...Suffix denoting column of prefixed variable.
CC             TDOSE...Desired TDF at PNTINT(1)
CC             TDFIN...Total dose equivalent achieved.
CC             TOTD....Total dose equivalent at PNTINT(1)
C
      COMMON/CONCOM/CENTEL,AXEL
      COMMON/PARCON/X,NELLI,MELL,LL
      COMMON/RATE/DRATE,TDOSE,CON
C
      CHARACTER*20 TITLE
      CHARACTER*10 COMMNT(50)
      CHARACTER*5 VARC
      CHARACTER*1 POSCOL(50)
      REAL XCOL(50),FCOL(50),CCOL(50),CONCOL(50),PNTCOL(50)
      REAL TDOSE,DRATE,TIME,DRFIN
      REAL RHOCOL(50),SUMCOL(50),GLCCL(50)
      DOUBLE PRECISION RLACCL(50)
      REAL SPECDC
      INTEGER ITCOL(50),NFCOL(50)
      INTEGER IFAIL,MCALL,NINTP,N,M,NTYP,MTYP,NITER,NF
      INTEGER NELLI(5),ITABLE(7)
      INTEGER MELL,LL
      REAL PNTINT(60),DESDOS(20),SUM(20),ACT(40)
      REAL CENTEL(3),AXEL(2)
      DOUBLE PRECISION COND,CNORM,GLNORM
      DOUBLE PRECISION ETA,STEPMX,XC(N),FC,CC(M),RHO,RLAM(M)
      REAL ELLI(9,4),CON(40)
      DOUBLE PRECISION T(40),TCOL(50),X(80)
      DOUBLE PRECISION TEM,XW(5)
      LOGICAL POSDEF
C
      DATA NPF/23/
C
CC       Initialisation of variable column heading.
C
      IF(NTYP.GE.7) THEN
      VARC='PARAM'
      LT=LL
      ELSE
      VARC='CNORM'
      END IF
C
CC       Initialisation of all data cclumns.
C
      DO 10 I=1,50
```

```
      CCMMNT(I)=' '
      POSCOL(I)=' '
      XCOL(I)=0.0
      FCOL(I)=0.
      CCOL(I)=0.
      CONCOL(I)=0.
      PNTCOL(I)=0.
      ITCOL(I)=0
      NFCOL(I)=0
      SUMCOL(I)=0.
      GLCOL(I)=0.
      RLACOL(I)=0.
      RHOCOL(I)=0.
      TCOL(I)=0.
10    CONTINUE
C
C       Writing of heading and all control variables used.
C        Only done on first call.
C
      IF(KFLAG.EQ.0) THEN
      WRITE(NPF,100) TITLE,NTYP,MTYP,ETA,STEPMX,
100   FORMAT(1X,'TITLE:',A20,'NTYP=',I1,' MTYP=',I1,' ETA=',F10.4,
     1 1X,' STEPMX=',E10.3)
      IF(IFAIL.NE.0) WRITE(NPF,105) IFAIL
105   FORMAT(1X,' IFAIL=',I2)
      WRITE(NPF,101) (DESDOS(I),I=1,NINTP)
101   FORMAT(1X,' DESDOS=',10(1X,F8.4))
      IF(NTYP.LT.7) WRITE(NPF,102) (ACT(I),I=1,NSRCE)
102   FORMAT(1X,' ACT=',10(1X,F8.3))
      IF(MTYP.EQ.2) THEN
      WRITE(NPF,103) AXEL,CENTEL
103   FORMAT(1X,'  Axes of ellipse are ',2F8.3,' cent. at',3F8.2)
      END IF
      IF(NTYP.EQ.3) THEN
      WRITE(NPF,104) SPECDC
104   FORMAT(1X,'  The specific dose constant used was=',F8.3)
      END IF
C
C       Writing of input values of the ellipse and sources.
C
      IF(NTYP.GE.7) THEN
      DO 119 K=1,NELLI(1)
      DO 119 L=1,9
      ELLI(L,K)=SNGL(X(40+L+(K-1)*9))
119   CONTINUE
      DO 120 I=1,NELLI(1)
      WRITE(NPF,109) I,ELLI(1,I),ELLI(2,I),ELLI(3,I),ELLI(4,I),
     1          ELLI(5,I),ELLI(6,I),ELLI(7,I),ELLI(8,I),ELLI(9,I)
109   FORMAT(12X,' ELLIPSE ',I1,/,1X,' AMPLITUDE',3X,3(1X,D10.3),/
     1     1X,' PHASES',6X,3(1X,D10.3),/,1X,' CENTER',6X,3(1X,D10.3))
C
      WRITE(NPF,111) (ACT((I-1)*10+J),J=1,NELLI(I+1))
111   FORMAT(1X,' Source activities are ',10(F8.2))
C
120   CONTINUE
      WRITE(NPF,210)
210   FORMAT(1X,/,' Source Data',/,/,18X,'ACT',8X,'X',8X,'Y',8X,'Z',4X,
     1       5X,'Theta',4X,'Phi')
      DO 203 I=1,NELLI(1)
      WRITE(NPF,211) I
211   FORMAT(1X,/,2X,'Ellipse(',I1,')')
C
```

```
      ****** TABLE ******


      LL=I
C
      DO 204 J=1,NELLI(I+1)
      TEM=X(J+(I-1)*10)
      CALL PARA(XW,TEM,NSRCE,NDIM)
      WRITE(NPF,312) J,ACT(((I-1)*10)+J),XW
204   CONTINUE
203   CONTINUE
C
      END IF
C
      WRITE(NPF,110) VARC
110   FORMAT(1X,///,'COMMENT',9X,' XC',3X,'PNTINT',1X,
     2 1X,' Function',4X,'Dose',4X,'Constnt',2X,' RLAM  ',2X,
     3 1X,' Rho',5X,'GLNORM',3X,A5,2X,'Conditn',1X,
     4 3X,'Calls',1X,'Iter',2X,'PD')
      GOTO 500
      END IF
C
C      Setting of comment column.
C
      IF(NTYP.GE.7.AND.KFLAG.EQ.1.AND.LL.EQ.0) THEN
      DO 300 II=1,40
      COM(II)=0.
300   CONTINUE
C
      END IF
C
      IF(NTYP.GE.7.AND.LL.NE.0) COMMNT(2)='ELLIPSE '//CHAR(LL+48)
      IF(NTYP.GE.7.AND.LL.EQ.0) COMMNT(2)='VARY ELLI'
      IF(KFLAG.EQ.1) THEN
      COMMNT(1)='Initial'
      ELSEIF(KFLAG.EQ.2) THEN
      COMMNT(1)='End cycle'
      ELSEIF(KFLAG.EQ.3) THEN
      COMMNT(1)='Intermit'
      ELSEIF(KFLAG.EQ.4) THEN
      COMMNT(1)='Results'
C     WRITE(6,*) MTYP,NTYP
      ELSE
      COMMNT(1)='I fail'
      ITCOL(2)=IFAIL
      ENDIF
C
      MAXNO=N
      IF(NINTP*3.GT.N.AND.NTYP.LT.7) MAXNO=NINTP*3
      IF(M.GT.MAXNO) MAXNO=M
      IF(MAXNO.LT.NINTP) MAXNO=NINTP
C
C      Setting of all column variables used.
C
      DO 20 I=1,M
      CCOL(I)=SNGL(CC(I))
      RLACOL(I)=RLAM(I)
20    CONTINUE
C
      DO 30 I=1,N
      XCOL(I)=SNGL(XC(I))
30    CONTINUE
C
      DO 40 I=1,NINTP
      SUMCOL(I)=SUM(I)
```

```
      ******  TABLE  ******

40    CONTINUE
C
      IF(NTYP.LT.7) THEN
      DO 50 I=1,NINTP*3
      PNTCOL(I)=PNTINT(I)
50    CONTINUE
      END IF
C
      IF(NTYP.GE.7.AND.LL.NE.0) THEN
C
      LK=(LL-1)*10
      DO 55 I=1,NELLI(LL+1)
      TCOL(I)=XC(LK+I)
55    CONTINUE
C
      ELSEIF(NTYP.GE.7.AND.LL.EQ.0) THEN
C
      DO 56 J=1,NELLI(1)*9
      TCOL(J)=XC(40+J)
56    CONTINUE
C
      ELSE
C
      TCOL(1)=CNORM
C
      END IF
C
      IF(POSDEF) POSCOL(1)='T'
      IF(.NOT.POSDEF) POSCOL(1)='F'
C
      FCOL(1)=SNGL(FC)
      CONCOL(1)=SNGL(COND)
      ITCOL(1)=NITER
      NFCOL(1)=NF
      GLCOL(1)=SNGL(GLNORM)
C
      IF(NTYP.GE.7) GLCOL(2)=SNGL(CNORM)
C
      RHOCOL(1)=SNGL(RHO)
C
      DO 60 I=1,MAXNO
      WRITE(NPF,200) COMMNT(I),XCOL(I),PNTCOL(I),FCOL(I),SUMCOL(I),
     1    CCOL(I),RLACOL(I),RHOCOL(I),GLCOL(I),TCOL(I),CONCOL(I),
     2    NFCOL(I),ITCOL(I),POSCOL(I)
200   FORMAT(1X,A10,1X,F8.4,1X,F8.4,1X,E10.4,1X,E8.2,1X,E8.2,1X,D8.2,
     1       1X,E8.2,1X,E8.2,1X,D9.3,1X,E8.2,1X,I6,1X,I3,1X,A1)
60    CONTINUE
C
      IF(NTYP.GE.7) THEN
C
      IF(KFLAG.GE.4) THEN
      CALL SU$P(ITABLE)
      WRITE(NPF,321) ITABLE(1),ITABLE(2),ITABLE(4)
321   FORMAT(1X,'  Time used total =',I12,'sups  Run card Time =',
     1 1X,I12,'sups CPU Time=',I12,'sups')
C
      WRITE(NPF,310)
310   FORMAT(1X,//,' Source Data',//,18X,'ACT',8X,'X',8X,'Y',8X,'Z',4X,
     1          5X,'Phi',6X,'Theta',4X,'Gap to Next Source')
      DO 303 I=1,NELLI(1)
      WRITE(NPF,311) I
311   FORMAT(1X,/,2X,'Ellipse(',I1,')')
```

```
      ******  TABLE  ******


C
      LL=I
C
      DO 304 J=1,NELLI(I+1)
      TEM=XC(J+(I-1)*10)
      CALL PARA(XW,TEM,NSRCE,NDIM)
      WRITE(NPF,312) J,ACT(((I-1)*10)+J),XW,CON(J+(I-1)*10)
312   FORMAT(1X,'  Source(',I2,')',1X,F8.2,3X,6(1X,F8.4))
304   CONTINUE
303   CONTINUE
C
      END IF
C
      IF(KFLAG.EQ.1.OR.KFLAG.GE.4) THEN
C
C      Calc. of the time required to achieve desired TDF.
C      Calc. of total dose equivalent to PNTINT(1)
C
      HLAM=0.000480551
      TDOSE=SNGL(X(80))
      TIME=TDOSE/((4.76E-3)*((150*DRATE)**1.35))
      IF(TIME.GT.1500) TIME=1500
      TIME=-(ALOG(1-1.35*TIME*HLAM))/(1.35*HLAM)
      TOTD=TIME*DRATE*1.5
C
      WRITE(NPF,313) TIME,TDOSE,TOTD
313   FORMAT(1X,' Interest Point Data . Time for Rx=',F8.2,'hrs',
     1  1X,'to TDF =',F9.3,' & dose =',F9.3,'Gy equiv. at PNTINT(1)',/
     1  1X,' No.  TOT Gy Equiv. GyEquiv./hr  X',8X,'Y',8X,'Z')
C
C      Convertion to dose equivalent for I-125 photons.
C
      DO 330 K=1,NINTP
      DRFIN=DRATE*SUM(K)*1.5
      TDFIN=DRFIN*TIME
      WRITE(NPF,329) K,TDFIN,DRFIN,PNTINT((K-1)*3+1),
     1                PNTINT((K-1)*3+2),PNTINT((K-1)*3+3)
329   FORMAT(1X,I2,2X,F10.2,2X,F10.2,3(1X,F8.3))
330   CONTINUE
C
      END IF
C
      END IF
C
500   LL=LT
C
      RETURN
C
      END
```

```
      SUBROUTINE TFORM(A,B,PIB2,TTR)
C
CC     This routine calculates the long and short half axes of the
CC      parameterised ellipse No.LL and the angle of transformation
CC      of T to a coordinate system with axis A in X direction,
CC      and axis B in the  Y direction.
CC
CC     AUTHOR: W..I.C.RAE
CC
CC     VERSION: 6/11/86
CC
C     Declaration of variables;
C
      DOUBLE PRECISION X(80)
      REAL A,B,PIB2,TTR,TEMP
      INTEGER NELLI(5),MELL,LL
      REAL AD,AE
      REAL A1,A2,A3
C
CC     Common block used in CALC2,PAROPT,OPTPAR,OPTELI,TABLE,TFORM,
CC                          PARA,CON1,FSEV.
C
      COMMON/PARCON/X,NELLI,MELL,LL
C
CC     Calculation of the parameter giving a vector in the direction
CC      of maximum change of the tangent vector to the ellipse.
CC      This is in the direction of the half axis.
C
      NV=40+(LL-1)*9
C
      X1=SNGL(X(NV+1))
      X2=SNGL(X(NV+2))
      X3=SNGL(X(NV+3))
      X4=SNGL(X(NV+4))
      X5=SNGL(X(NV+5))
      X6=SNGL(X(NV+6))
      A1=X1*X1
      A2=X2*X2
      A3=X3*X3
C
      AD=A1*SIN(2.*X4)+
     1    A2*SIN(2.*X5)+
     2    A3*SIN(2.*X6)
C
      AE=A1*COS(2.*X4)+
     1    A2*COS(2.*X5)+
     2    A3*COS(2.*X6)
C
      IF(AD.LT.0.0001) THEN
C
      TMAX=0.0
C
      ELSE
C
      IF(AE.EQ.0) AE=0.0001
C
      TMAX=.5*ATAN(AD/AE)
C
      END IF
CC     Calc. of half axes of new ellipse using above result.
C
      A=(SQRT((X1*SIN(TMAX+X4))**2+
     1       (X2*SIN(TMAX+X5))**2+
     1       (X3*SIN(TMAX+X6))**2))
C
      TP=(TMAX+PIB2)
C
      B=(SQRT((X1*SIN(TP+X4))**2+
     1       (X2*SIN(TP+X5))**2+
     1       (X3*SIN(TP+X6))**2))
C
CC      Set long axis to be A and in the direction of TTR.
C
      TEMP=A
      IF(B.GT.A) THEN
      A=B
      B=TEMP
      TTR=(TP)
      ELSE
C
CC       Setting of angle transformation.
C
      TTR=(TMAX)                                                  P62
C
      END IF
      RETURN
      END
```

```
      SUBROUTINE UPDAT
C
CC    This subroutine accepts new raw data for
CC      interpolation by E01ACE to create the table MM(101,101)
CC      using bicubic splines.
CC    It also allows interpolation on these tables and updating
CC      of the values presently in the data file.
CC
CC    AUTHOR :W.I.D.RAE
CC
CC    VERSION :28/7/86
CC
CC    Declaration of variables.
CC     DATITL...Title of data array for dose data.2-dim.
CC     MTFL...File for interpolated matrix from E01ACE.
CC     DATFL...File for original data input.
CC     A.......Value for interpolation by E01ACE  in X-axis.
CC     B.......Value for interpolation as above in Y-axis.
CC     X......Array for distance,(DIS) values stored for INTPOL.
CC     XINT....Intermediate array for use when calculations use X.
CC     Y.......Array for PSI values stored for interpolation.
CC     VAL.....Value returned as interpolated by E01ACE.
CC     F.......2-Dim Array for input function values.
CC     M1......Number of PSI values.
CC     N1......Number of DIS values input.
CC     IG1.....Size of working arrays.
CC     FF......1-Dim array to pass to INTPOL
CC     SPECDC...Specific dose constant to use in calc.
CC     DRCF.....Conversion factor from degrees to radians.
CC     STPX.....Step size between successive points for matrix MM.
CC     NOREC....Number of file to use.
CC     NFAIL....Converted NAG error indicator.
CC     NDOS.....Number of data file input.(DATFL.)
CC     MDAT.....Number of interpolated data file for output.(MTFL.)
CC     AM,XX,D,WORK..Working arrays used by E01ACE.
CC     MM......Array for storage and lookup of interpolated values.
CC     XMX,XMN..Max and min values on X axis.
CC     STPX.....Step size on X axis.
C
      INTEGER M1,NOREC,NFAIL,NDOS,MDAT,IG1,N1
      REAL A,B,X(10),XINT(10),Y(15),F(10,15),AM(25),XX(25),
     1    D(25),WORK(25),VAL,FF(150),MM(101,101)
      REAL XMX,XMN,YMX,YMN,STPX,STPY,SPECDC,DRCF,CX1,CX2,CY1,CY2
      CHARACTER*20 DATITL
C
CC     Declaration of common block.
C
      COMMON/LINCOM/MM,CX1,CX2,CY1,CY2
C
CC
CC      Declaration of file unit numbers.
C
      DATA MDAT/18/
      DATA NDOS/19/
      DATA NOUT/6/
      DATA DRCF/0.01745329252/
C
CC      Presentaion of options allowed and title.
C
10    CALL PAG
```

```
      ******  UPDAT  ******


C
      WRITE(NOUT,71)
71    FORMAT(1X,' ***BRACHY.UPDAT***',//,'  ******************',//)
12    WRITE(NOUT,80)
80    FORMAT(1X,' Do you wish to use new records,.....(1)?',/
     1 1X,'                      update old records.....(2)?',/
     1 1X,'                      interpolate on data.....(3)?',/
     1 1X,'                      interpolate for table.....(4)?',/
     1 1X,'                      interpol.linearly off table(5)?',/
     1 1X,'                      exit.....(0)?')
      READ(5,*,ERR=10) NOLD
C
C      Allows exit.
C
      IF(NOLD.EQ.0) GOTO 400
C
C      Lists data already present.
C
      CALL PAG
C
      DO 88 I=1,10
      READ(NDOS,REC=I) DATITL
      IF(DATITL.EQ.'  ') GOTO 88
      WRITE(NOUT,89) I,DATITL
89    FORMAT(1X,' Record ',I2,' Title : ',A20)
88    CONTINUE
      WRITE(NOUT,87)
87    FORMAT(1X,/)
C
C      Allows input of new data records.
C
      IF(NOLD.EQ.1) THEN
C
99    WRITE(NOUT,100)
100   FORMAT(1X,/,1X,' Enter title for new Data Array')
      READ(5,101,ERR=99) DATITL
101   FORMAT(A20)
C
106   WRITE(NOUT,107)
107   FORMAT(1X,' Enter record no to be used.')
      READ(5,*,ERR=106) NOREC
C
119   WRITE(NOUT,120)
120   FORMAT(1X,' No. of DIS vals. to enter, strictly increasing.')
      READ(5,*,ERR=119) N1
C
121   WRITE(NOUT,122)
122   FORMAT(1X,' No. of PSI vals. to enter, strictly increasing.')
      READ(5,*,ERR=121) M1
C
      DO 200 I=1,N1
127   WRITE(NOUT,123) I
123   FORMAT(1X,' Enter DIS(',I3,')')
      READ(5,*) X(I)
      IF(I.GE.2)THEN
      IF(X(I).LE.X(I-1)) GOTO 127
      END IF
      WRITE(NOUT,126)
      READ(5,*,ERR=127) NCON
      IF(NCON.NE.1) GOTO 127
200   CONTINUE
C
```

```
      ****** UPDAT ******


      DO 221 J=1,M1
128   WRITE(NOUT,124) J
124   FORMAT(1X,' Enter PSI(',I3,')')
      READ(5,*) Y(J)
      IF(J.GE.2) THEN
      IF(Y(J).LE.Y(J-1)) GOTO 128
      END IF
      WRITE(NOUT,126)
      READ(5,*,ERR=128) NCON
      IF(NCON.NE.1) GOTO 128
221   CONTINUE
C
      DO 220 I=1,N1
      DO 220 J=1,M1
129   WRITE(NOUT,125) X(I),Y(J),I,J
125   FORMAT(1X,' DIS=',1PE9.3,' &PSI=',1PE9.3,' enter F(',I2,I2,')')
      READ(5,*) F(I,J)
      WRITE(NOUT,126)
126   FORMAT(1X,' To confirm press 1 if correct')
      READ(5,*,ERR=129) NCON
      IF(NCON.NE.1) GOTO 129
220   CONTINUE
C
      WRITE(NDOS,REC=NOREC) DATITL,X,Y,F,M1,N1
C
      GOTO 10
      END IF
C
C      Allows update of old records and allows copy to any record.
C
      IF(NOLD.EQ.2) THEN
149   WRITE(NOUT,150)
150   FORMAT(1X,//,'  Which record do you wish to change?')
      READ(5,*,ERR=149) NREC
C
      READ(NDOS,REC=NREC,ERR=149) DATITL,X,Y,F,M1,N1
C
161   CALL PAG
      WRITE(NOUT,370)
370   FORMAT(1X,'    Do you want to write to the same record,Yes=1')
      READ(5,*) NSAM
      IF(NSAM.EQ.1) GOTO 162
      WRITE(NOUT,371)
371   FORMAT(1X,' Enter new record number')
      READ(5,*) NREC
162   WRITE(NOUT,311) DATITL,(X(I),I=1,N1)
      DO 163 I=1,M1
      WRITE(NOUT,313) Y(I),(F(J,I),J=1,N1)
163   CONTINUE
      WRITE(NDOS,REC=NREC) DATITL,X,Y,F,M1,N1
      WRITE(NOUT,152)
152   FORMAT(1X,//,'  Enter number of change only. Exit=0',/
     1 21X,'    If N1 or M1 are changed then update DIS & PSI',//
     1 21X,'    Title..1    DIS....2',//
     1 21X,'    PSI....3    F......4',//
     1 21X,'    N1.....5    M1.....6',///)
      READ(5,*) NCH
C
      IF(NCH.LT.1.OR.NCH.GT.6) GOTO 10
C
      IF(NCH.EQ.1) THEN
      WRITE(NOUT,309)
```

```
309     FORMAT(1X,' Enter new Title.')
        READ(5,151) DATITL
151     FORMAT(A20)
        GOTO 161
        END IF
C
        IF(NCH.EQ.2)THEN
230     WRITE(NOUT,327)
327     FORMAT(1X,' Enter number of DIS value to change.')
        READ(5,*,ERR=161)  II
        WRITE(NOUT,305) II
305     FORMAT(1X,' Enter value for DIS(',I2,').')
        READ(5,*) X(II)
        IF(II.GE.2) THEN
        IF(X(II).LE.X(II-1)) GOTO 230
        END IF
        GOTO 161
        END IF
C
        IF(NCH.EQ.3)THEN
235     WRITE(NOUT,307)
307     FORMAT(1X,' Enter number of PSI value to change.')
        READ(5,*,ERR=161)  II
        WRITE(NOUT,308) II
308     FORMAT(1X,' Enter value for PSI(',I2,').')
        READ(5,*) Y(II)
        IF(II.GE.2) THEN
        IF(Y(II).LE.Y(II-1)) GOTO 235
        END IF
        GOTO 161
        END IF
C
        IF(NCH.EQ.4)THEN
        WRITE(NOUT,320)
320     FORMAT(1X,' Enter number of DIS value of F to change.')
        READ(5,*,ERR=161)  JJ
        WRITE(NOUT,323)
323     FORMAT(1X,' Enter number of PSI value of F to change.')
        READ(5,*,ERR=161)  II
        WRITE(NOUT,322) JJ,II
322     FORMAT(1X,' Enter value for F(',I2,I2,').')
        READ(5,*) F(JJ,II)
        GOTO 161
        END IF
C
        IF(NCH.EQ.5) THEN
        WRITE(NOUT,340)
340     FORMAT(1X,' Enter new value for N1')
        READ(5,*,ERR=161) N1
        GOTO 161
        END IF
C
        IF(NCH.EQ.6) THEN
        WRITE(NOUT,341)
341     FORMAT(1X,' Enter new value for M1')
        READ(5,*,ERR=161) M1
        GOTO 161
        END IF
        GOTO 161
        END IF
C
C        Allows interpolation using bicubic splines on raw data.
```

```
C
      IF(NOLD.EQ.3) THEN
299   WRITE(NOUT,300)
300   FORMAT(1X,' Do you want to interpolate ? YES=1')
      READ(5,*) INT
      IF(INT.NE.1) GOTO 10
C
      WRITE(NOUT,351)
351   FORMAT(1X,' With which record will you calc?')
      READ(5,*) NOREC
      READ(NDOS,REC=NOREC) DATITL,X,Y,F,M1,N1
      IG1=M1+N1
C
      WRITE(NOUT,311) DATITL,(X(I),I=1,N1)
      DO 312 I=1,M1
      WRITE(NOUT,313) Y(I),(F(J,I),J=1,N1)
312   CONTINUE
      DO 321 I=1,N1
      DO 321 J=1,M1
      K=(I-1)*M1+J
      FF(K)=F(I,J)
      F(I,J)=0
321   CONTINUE
C
      WRITE(NOUT,303)
303   FORMAT(1X,' Enter point in DIS for calculation')
      READ(5,*) A
      WRITE(NOUT,304)
304   FORMAT(1X,' Enter point in PSI for calc.')
      READ(5,*) B
C
      CALL INTPOL(X,Y,FF,F,A,B,VAL,NFAIL,M1,AM,XX,WORK,D,N1,IG1)
C
      IF(NFAIL.NE.0) THEN
      WRITE(NOUT,310) NFAIL
      END IF
C
      WRITE(NOUT,302) A,B,VAL
302   FORMAT(1X,' At point ',1PE12.4,',',1PE12.4,' Value=',1PE12.4)
C
      PAUSE 'PRESS RETURN TO CONTINUE'
C
      CALL PAG
C
      GOTO 299
C
      END IF
C
C      Allows calculation of full array in real time.(7-10minutes).
C
      IF(NOLD.EQ.4) THEN
C
      WRITE(NOUT,301)
301   FORMAT(1X,' With which record will you calc?')
      READ(5,*) NOREC
      READ(NDOS,REC=NOREC) DATITL,X,Y,F,M1,N1
C
      WRITE(NOUT,311) DATITL,(X(I),I=1,N1)
311   FORMAT(1X,' TITLE : ',A20,/,1CX,10(1X,1PE8.2))
      DO 352 I=1,M1
      WRITE(NOUT,313) Y(I),(F(J,I),J=1,N1)
313   FORMAT(1X,1PE8.2,1X,10(1X,1PE8.2))
```

```
352     CONTINUE
C
        WRITE(NOUT,363)
363     FORMAT(1X,//,'    What value for spec. dose. const. in',//
     1 1X,'  Gy.cm2/hr.MBq  will you use?')
        READ(5,*) SPECDC
C
        DO 353 I=1,N1
        DO 353 J=1,M1
        K=(N1-I)*M1+J
        FF(K)=F(I,J)
        F(I,J)=0
353     CONTINUE
C
C        Setting up of values on the axes used for interpolation.
C
        DO 713 I=1,N1
        XINT(I)=1/(X(N1+1-I))
713     CONTINUE
        DO 721 I=1,N1
        X(I)=XINT(I)
721     CONTINUE
C
        DO 714 I=1,M1
        Y(I)=SIN(Y(I)*DRDF)
714     CONTINUE
C
        IG1=N1+M1
C
C        Set up of step sizes and scaling factors.
C
        XMN=X(1)
        XMX=X(N1)
        YMN=Y(1)
        YMX=Y(M1)
        STPX=(XMX-XMN)/100
        STPY=(YMX-YMN)/100
        CX1=1/STPX
        CX2=1-XMN/STPX
        CY1=1/STPY
        CY2=1-YMN/STPY
C
C
        DO 333 I=1,101
        DO 334 J=1,101
        A=(XMN+(I-1)*STPX)
        B=YMN+(J-1)*STPY
C
C        Interpolation routine that calls E01ACE to interpolate
C          using cubic spline fitting.
C
        CALL INTPOL(X,Y,FF,F,A,B,VAL,NFAIL,M1,AM,XX,WORK,D,N1,IG1)
C
        IF(NFAIL.NE.0) THEN
        WRITE(NOUT,310) NFAIL
310     FORMAT(1X,' NFAIL=',I3)
        END IF
C
C        Calc. of dose rate in Gy/hr.MBq for storage in Array MM.
C
        MM(I,J)=VAL*SPECDC*A*A
334     CONTINUE
C
C        Indication of every 101 calls of E01ACE.
C
        WRITE(NOUT,401) I
401     FORMAT(1X,'  Counter =',I3)
333     CONTINUE
C
        WRITE(MDAT,REC=NOREC) MM,CX1,CX2,CY1,CY2,SPECDC
C
        GOTO 10
C
        END IF
C
        IF(NOLD.EQ.5) THEN
        CALL SUBLIN
        GOTO 10
        END IF
C
400     PAUSE 'EXIT FROM TEST.UPDAT'
        RETURN
        END
```

```
      SUBROUTINE WRITE(TITLE,STEPMX,ETA,IBOUND,IFAIL,RHO,IPRINT,
     1                 XU,XL,N,X)
C
C     This routine writes initial values for OPTIM in TEST.
C
C     AUTHOR:W.I.D.RAE
C
C     VERSION:1/9/86
C
C
C      Declaration of variables as listed here below..
C           ETA......Specifies accuracy of linear minimisation.
C           F........Contains value of F(X) on exit.
C           RHO......Current value of parameter rho in Lagrangian.
C           STEPMX...Estimate of Euclidian distance to min.
C           CL( )....Array of dim >or= MRNGE,lower bound on constraint.
C           CU( )....Upper bound on constraint.
C           XL( )....Array of dim >or= N, contains fixed lower bounds.
C           XU( )....Array contains fixed upper bounds of X( ).
C           X( )......Array contains value of constrained minimum.
C           MEQ......Equality constraint number.
C           MINEQ....Inequality constraint number.
C           MRNGE....Number of range constraints.
C           M........MEQ+MINEQ+MRNGE.
C           N........Number of independent variables.
C           NX.......Integer =N+MINEQ+MRNGE.
C           I........Integer.
C           IBOUND...Type of bounds used see notes.
C           IFAIL....Indicator of type of failure in optimisation.
C           IPRINT...Regulates calling of AMONIT by E04UAF.
C           LCLU.....Actual length of CL,CU declared in CON1.
C           PNTINT...Points  of interest to calculate uniformity.
C           NTYP.....Type of isotope chosen to  regulate function.
C           ACT......Activity of sources in group.
C           DESDOS...Desired dose of points of interest.
C           NINTP....Number of interest points.
C           MTYP.....Type of constraint function used.
C           NDIM.....Dimension of source variables.
C           XM,XV....Dummy variables for XL,XU respectively.
C
C      Declaration of common blocks.
C
      COMMON/ISO/NTYP,PNTINT,NINTP,ACT,DESDOS,NSRCE,MTYP,NDIM
      COMMON/MON/CL,CU,MEQ,MINEQ,MRNGE
C
C      Declaration of variables.
C
      DOUBLE PRECISION ETA,RHO,STEPMX
      DOUBLE PRECISION CL(20),CU(20),XL(80),XU(80)
      DOUBLE PRECISION X(80),XM(80),XV(80)
      DOUBLE PRECISION DSQRT
      INTEGER MEQ,MINEQ,MRNGE,I,IBOUND,IFAIL,IPRINT
      INTEGER N
      INTEGER NSRCE,NDIM
      INTEGER NTYP,NINTP,MTYP
      REAL PNTINT(60),ACT(40),DESDOS(20)
      CHARACTER*20 TITLE
C
C      DATA declaration of output unit numbers.
C
```

```
      ******  WRITE  ******


      DATA MPF /21/
C
C       writing of all the initial input used in call of E04UAF
C
472   WRITE(MPF,240 ) TITLE
240   FORMAT(1X,///,' **INPUT**',//,' *********',///,' TITLE :',A20)
      WRITE(MPF,207 ) NTYP,NSRCE,N,MEQ,MINEQ,MRNGE,IPRINT,
     1          STEPMX,ETA,IBOUND,RHO,NINTP,IFAIL,MTYP,NDIM
207   FORMAT(1X/,' NTYP=',I3,'; NSRCE=',I3,'; N=',I3,'; MEQ=',I3,/
     1 1X,'MINEQ=',I3,'; MRNGE=',I3,'; IPRINT=',I3,'; STEPMX=',D8.2,/
     2 1X,'ETA=',D8.2,';IBOUND=',I1,'; RHO=',D8.2,';NINTP=',I3,/
     3 1X,' IFAIL=',I1,';MTYP=',I2,'; NDIM=',I2)
C
      DO 545  I=1,NSRCE
      WRITE(MPF,210 ) I,ACT(I)
210   FORMAT(1X,' Source ',I2,' with activity ',F6.2,' has')
      DO 545  J=1,NDIM
      K=NDIM*(I-1)+J
C
      IF(IBOUND.EQ.0) THEN
      XM(K)=XL(K)
      XV(K)=XU(K)
      END IF
C
      IF(IBOUND.EQ.1) THEN
      XM(K)=-1000000000
      XV(K)=1000000000
      END IF
C
      IF(IBOUND.EQ.2) THEN
      XM(K)=0.0
      XV(K)=1000000000
      END IF
C
      IF(IBOUND.EQ.3) THEN
      XM(K)=XL(1)
      XV(K)=XU(1)
      END IF
C
      WRITE(MPF,211 ) XM(K),X(K),XV(K)
211   FORMAT(1X,1PD10.3,'<',1PD10.3,'<',1PD10.3)
545   CONTINUE
      DO 550  I=1,MRNGE
      WRITE(MPF,208 ) I,CU(I),CL(I)
208   FORMAT(1X,' Range constr. ',I3,' has CU=',D8.2,'; CL=',D8.2)
550   CONTINUE
      DO 555  I=1,NINTP
      K=1+(I-1)*3
      WRITE(MPF,241 ) I,PNTINT(K),PNTINT(K+1),PNTINT(K+2)
241   FORMAT(1X,'  Nintp.',I2,' X=',F10.4,/,11X,' Y=',F10.4,/
     1 1X,10X,' Z=',F10.4,/)
555   CONTINUE
C
      RETURN
C
      END
```