

INVERSE RADIOSURGERY TREATMENT PLANNING THROUGH
DECONVOLUTION AND CONSTRAINED OPTIMIZATION

by

JOSEPH F. HARMON JR.

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL OF THE
UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1994

LD
1780
1994
. H288



ACKNOWLEDGMENTS

Many individuals have contributed to the completion of this research. First, I am grateful for the guidance and helpful comments provided by all of my committee members. Special thanks are due Dr. Frank Bova for his valuable insight and support while serving as committee chairman. I wish to also extend my thanks to Dr. Rodhe Mohan and Michael Lovelock at Memorial Sloan Kettering Cancer Center for providing me with the differential pencil beam data so critical to this endeavor. Finally, and most of all, I wish to thank my wife, Rose, and my children, Ashley, Sarah, and Taylor, for their patience and never ending support.

TABLE OF CONTENTS

	<u>page</u>
ABSTRACT.....	v
CHAPTERS	
1 INTRODUCTION.....	1
Overview.....	1
Stereotactic Radiosurgery.....	2
Inverse Treatment Planning.....	4
Treatment Implementation.....	6
2 REVIEW OF LITERATURE.....	10
Dosimetry Algorithms.....	10
Overview.....	10
Matrix Models.....	10
Separation of Primary and Scattered Components..	12
Monte Carlo Techniques.....	13
Energy Deposition Kernels.....	14
Inverse Radiotherapy Planning Approaches.....	23
Analytical Methods.....	24
Iterative Methods.....	28
3 DOSIMETRY CALCULATIONS.....	32
Introduction.....	32
Energy Deposition Kernel.....	32
TERMA.....	41
Convolution.....	42
4 BREMSSTRAHLUNG SPECTRA.....	48
Introduction.....	48
Spectrum Model.....	49
Optimization Algorithm.....	56
Results.....	59
5 DOSIMETRY MEASUREMENTS.....	66
Overview.....	66
Radiochromic Film.....	67
TLD Sheets.....	77

6 INTENSITY MODULATION.....	83
Overview.....	83
Optimal Intensity Modulation.....	83
Deconvolution.....	83
Backprojection.....	88
Single Beam Intensity Modulated Dosimetry.....	93
7 RADIOSURGERY PLANNING.....	97
Overview.....	97
Coordinate Transformation.....	98
Multiple Beam Optimization.....	102
Annealing Parameters.....	104
Example Results.....	107
8 EXTENSION TO GENERAL RADIOTHERAPY.....	135
Overview.....	135
Small Field Approximations.....	136
Large Field Dosimetry verification.....	137
Example Inverse Radiotherapy Case.....	149
9 SUMMARY AND CONCLUSIONS.....	153
APPENDIX.....	156
REFERENCES.....	226
BIOGRAPHICAL SKETCH.....	233

Abstract of Dissertation Presented to the Graduate School of
the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

INVERSE RADIOSURGERY TREATMENT PLANNING THROUGH
DECONVOLUTION AND CONSTRAINED OPTIMIZATION

By

Joseph F. Harmon Jr.

April 1994

Chairman: Frank J. Bova, Ph.D.
Major Department: Nuclear Engineering Sciences

An inverse treatment planning approach is presented for radiosurgery which permits conformal dose distributions to small, irregularly shaped targets. Traditionally, such targets would require somewhat complex, multiple isocenter plans, arrived at after much trial and error effort. In contrast, the approach presented here requires only one isocenter location and arrives at an optimal plan automatically. The single isocenter is made possible by allowing complete beam intensity modulation for multiple beam's eye views (BEVs) of the target. The automation is due to incorporation of a deconvolution procedure to determine the intensity modulation of each BEV and a simulated annealing algorithm to optimize their respective weighting. Additionally, this method is fully 3-D and accurately models phantom scatter by incorporating Monte Carlo generated energy deposition kernels into the dosimetry process. The

extension of this methodology to larger targets for general radiotherapy cases is also addressed.

CHAPTER 1 INTRODUCTION

Overview

The goal of this research was development of an automated conformal 3-D treatment planning tool for radiosurgery sized targets incorporating photon beam intensity modulation and phantom scatter. Development of such a tool is considered a necessary step to eventually permit improved treatment of irregularly shaped radiosurgery sized target lesions by sparing dose to adjacent normal and critical structure tissue.

Automation of the planning procedure should reduce excessive trial and error treatment planning times associated with such radiosurgery cases. Also, due to the single isocenter approach of this method, the treatment delivery times should also be significantly reduced. Finally, the nature of the approach taken will permit a very flexible beam arrangement thus allowing a wide variety of mechanical treatment delivery methods.

This chapter will present a brief introduction to radiosurgery, inverse treatment planning, and potential methods of conformal treatment delivery. Subsequent chapters will present details of the specific approach taken and how it differs from other inverse treatment planning approaches.

Stereotactic Radiosurgery

Stereotactic radiosurgery was originally introduced in 1951 as a method of carefully focusing several small beams of ionizing radiation to destroy tumors in the brain (Leksell 1951). Over the years, the idea has been implemented with a wide range of radiation sources including 200-300 kV x-rays, multiple ^{60}Co sources, higher energy photons from linear accelerators, and heavy particles such as protons (Friedman & Bova 1989a). Of these possibilities, producing photons with a linac is the least expensive and thus most widely available option. Recently, this option has been made even more attractive for radiosurgery due to development of a system which quickly couples to an existing linac to improve positional accuracy. This system incorporates precision machined bearings and a gimbal device to ensure an isocenter accuracy of 0.2 ± 0.1 mm (Friedman & Bova 1989b).

The attractiveness of radiosurgery is due to the ability to treat lesions in regions of the brain that are otherwise inaccessible (Winston & Lutz 1988). Examples of the clinical applications include treatment of arteriovenous malformations (AVMs), acoustic schwannomas, malignant brain neoplasms, and meningiomas (Friedman, Bova & Spiegelmann 1992). The majority of the lesions treated are less than 30 mm in size and are subjected to single fraction doses ranging from 1500 to 3000 cGy (Friedman, Bova & Spiegelmann 1992).

In conventional radiation therapy, treatment is broken into many fractions in an attempt to take advantage of repair of normal tissues. This is accomplished through the differential biological sensitivity of healthy and tumor tissue to radiation (Hendrickson & Withers 1991). In stereotactic radiosurgery however, the radiation is usually delivered in a single fraction due to the difficulty in accurately localizing the lesion and positioning the patient over several treatments. In fact, the developments in radiosurgery have been slowed partly due to the need for more precise localization of the target (Leksell 1983). Because of the high, single fraction doses involved, it is obviously more important than ever to ensure lesion coverage while sparing healthy tissue. Thus, a highly accurate method of dose delivery is needed.

Depending on the specific case, target localization is accomplished with angiography, CT, MRI or a combination of these techniques. For example, due to the limitations of angiography, all AVM radiosurgery planning is supplemented with CT to ensure adequate target identification (Spiegelmann, Friedman & Bova 1992). The reference frame for localization is a precisely built Brown-Roberts-Wells (BRW) ring, mounted to the patient's head. An additional localizing frame is then attached to the top of the head ring. A different localizer is used for each of the possible diagnostic procedures mentioned earlier. They each, however, contain reference points which allow one to locate the

lesion in BRW stereotactic coordinates (anterior-posterior, lateral, and vertical). During treatment planning, a trial and error method is used to arrive at the number and direction of multiple beam arcs needed for treatment. Unlike conventional radiotherapy, radiosurgery treatment planning is not accomplished with the aid of images from a treatment simulator. Instead, the geometrical relationship between the treatment fields and internal anatomy is accomplished solely from images used from target localization. Also, as part of treatment planning, an appropriate diameter opening of a small bore (5-40 mm) circular collimator is chosen. This will be used, down stream from the standard rectangular field defining jaws, to obtain a circular shaped field. Treatment is then carried out by delivering the radiation in multiple arcs to the patient making use of the head ring still attached to the patient's head as illustrated in figure 1-1.

Inverse Treatment Planning

For radiation therapy one typically obtains an approximation to the ideal dose distribution through an iterative trial and error method. Various combinations of modified (e.g. wedges, irregular field shapes, compensators, etc.) and unmodified shaped beam combinations are simulated on a treatment planning workstation until an acceptable solution is found. Ideally, one wants as uniform coverage as possible in the target volume with a rapid fall off in dose outside the target volume. Standard empirical solutions

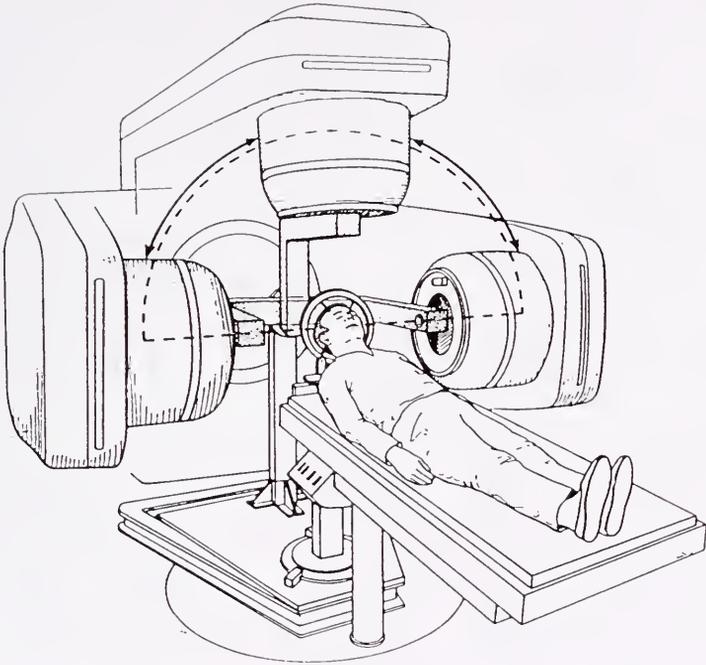


Figure 1-1. A schematic diagram indicating gantry rotation for a given couch position using the stereotactic radiosurgery attachment developed at the University of Florida (Bova 1990a).

exist for the more common cases (e.g. parallel opposed, four field box, etc.) but those involving multiple targets, targets of irregular shape or targets near critical radiosensitive structures, such as the spinal cord for example, can be difficult to plan for.

An alternate way of approaching the problem is to calculate the desired "optimal plan" directly from the ideal dose distribution and is referred to as inverse radiotherapy treatment planning. An optimal inverse solution will specify not only the number and direction of beams required but also the field shape and intensity modulation of each individual beam comprising the plan. Also, one must limit the solution to physically real solutions. For example, a solution resulting in negative fluence values may provide an ideal dose distribution in theory but is impossible to achieve in practice. There are several approaches one may take to arrive at a solution to the inverse planning problem and these are discussed later in the review of literature.

Treatment Implementation

Regardless of which approach is used to obtain the treatment planning solution, a method for implementation must be chosen. Specifically, a method of shaping and modulating the intensity of each incident field must be selected. Although hardware concerns are beyond the scope of this research, it is of interest to briefly present the techniques currently used or proposed to implement such a plan.

Other than hand made cerrobend blocks, the most predominant technique for general radiotherapy field shaping is the use of a multileaf collimator (Brahme 1988, Moss 1992). In this approach, the incident beam is shaped to conform to the beam's eye view of the target (lesion) contour at each orientation of the gantry (beam projection). A similar approach uses multiple independent vanes (Leavitt et al. 1991) to shape the borders of the field. However, these methods are currently limited to treating convex shaped targets for general radiotherapy since intensity modulation across the beam is not included (Brahme 1988). It also excludes the possibility of optimum treatment when an organ or region at risk is included within the target volume. Regardless of these limitations it is believed that field shaping alone would offer an improvement over current nonconformal techniques used in radiosurgery (Sixel, Podgorsak & Souhami 1993). In fact, studies have shown that approximately half of all stereotactic radiosurgery cases would receive a smaller radiation dose to normal brain tissue if conformal field shaping treatment options were available (Leavitt et al. 1991). Recently, the potential usefulness of multileaf collimators has been extended to include treatment of concave shaped targets and avoidance of enclosed critical structures (Galvin, Chen & Smith 1993). In this approach, the multileaf collimator is used to define a series of field shapes that are superimposed for a fixed

beam's eye view (BEV) to produce the desired modulated fluence pattern instead of simply shaping the field(s).

A second approach is to construct a customized intensity modulation device (compensator) to shape and attenuate the field as needed for each incident beam. Compensators are used in select general radiotherapy cases but usually take considerable time to create. Thus, they are only used when a small number of beams are present. This technique, as well as the multileaf approach, is not used for linac based stereotactic radiosurgery since the treatments are currently achieved by multiple continuous arcs. However, these techniques may prove useful for a treatment plan based on a reasonable number of highly modulated incident beams as proposed by this research.

Another possible approach is to scan a pencil beam of photons across the target volume (Näfstadius, Brahme & Nordell 1984). In principle this seems attractive but is limited to facilities equipped with such specialized equipment. Also, the intrinsic bremsstrahlung process and multiple scattering within the photon producing target result in a photon beam too broad for radiosurgery applications.

A fourth approach is irradiation of the target with a slit field incorporating intensity modulation along the length of the slit (Carol et al. 1993; Mackie et al. 1993). The common theme in this approach is the irradiation of multiple, contiguous slices with a temporally modulated

slit. In order to treat a 3-D region, both parallel slices (PeacockTM) and spiral slices (tomotherapy) have been proposed. Thus, this technique is analogous to image reconstruction in computed tomography where measured transmission profiles are back-projected to obtain a 2-D density distribution. However, the difference here is that a "measured" intensity modulation function (IMF) is "back projected" to obtain a dose distribution. Possible problems with this approach include the extreme geometrical accuracy needed to avoid "hot and cold" spots in the dose distribution between adjacent slices.

Regardless of the specific approach, implementing an inverse solution involves complete beam compensation (field shaping and intensity modulation across the field) to achieve nonuniform fields related to a desired dose distribution instead of achieving uniform fields at depth through incorporation of patient contours (Goitein 1990). The steep, often irregular dose profile encountered in radiosurgery represents a difficult inverse problem to solve. To date, no treatment planning methodology has been applied specifically for such a solution for stereotactic radiosurgery that includes complete beam compensation. Thus, the following chapters will address this issue.

CHAPTER 2
REVIEW OF LITERATURE

Dosimetry Algorithms

Overview

Regardless of how one approaches the inverse radiotherapy planning problem, an accurate algorithm for calculating dose distributions must be incorporated. There are several models to choose from although they do not all work equally well for a particular situation. The following sections will review the models available and point out their weaknesses and strengths.

Matrix Models

A simple and straightforward method of dose calculation is to measure and store isodose curves or beam profiles in an array. For completeness, this would be accomplished over the range of square field sizes and depths one expects to use. For rectangular field sizes, an equivalent square field can be used. Multidimensional interpolation is used to obtain data at positions not measured. Thus, a significant number of measurements may be required if one wants to reduce interpolation errors. For common field sizes and shapes, one can obtain off axis ratios, $f(d,x)$ and $g(d,y)$, that can be multiplied by central axis depth dose data, $d(d)$, to obtain the dose, $D(d,x,y)$ at any point (Horton 1987):

$$D(d,x,y) = f(d,x)g(d,y)$$

This model is commonly referred to as the Milan-Bentley algorithm and has been applied to stereotactic radiosurgery through measurements of small circular fields (Podgorsak et al. 1988). In fact, the dose model for stereotactic radiosurgery at the University of Florida uses a similar approach (Suh 1990). In this case, the dose at position p , D_p , for a circular field, is found from an off-axis-ratio (OAR), tissue maximum ratio (TMR), and a relative output factor (ROF) obtained from measured data such that:

$$D_p(C,STD,d,r) = D_{ref} ROF(C) TMR(w,d) OAR(C,STD,d,r) (SAD/STD)^2$$

where

C = field size at SAD (source to axis distance)

STD = source to target distance

d = depth at point p

r = off-axis distance

w = field size at point p

D_{ref} = reference dose

A closely related technique is that of storing the measured dose information in the form of decrement lines instead of a grid of points or off-axis ratios for all depths of interest. These lines, usually expressed as a second order polynomial, represent the distance from the central axis to a given off-center ratio. The off-axis data can then be represented by a small number of constants for each field size and can be combined with central axis

percent depth dose data to calculate dose at any given point.

These matrix type algorithms are fast and simple but require a large number of measurements and are not suitable for irregular field shapes or intensity modulated beams since each variation would require a complete set of measurements.

Separation of primary and scattered components

A commonly used technique for dose calculation is separation of the beam into primary and scattered components. In this model, the scattered photons are those defined to have undergone at least one interaction in the irradiated object before they contribute, in a subsequent interaction, to the dose at the point of interest (Bjarngard & Petti 1988). This technique, proposed by Clarkson and elaborated by Cunningham (Horton 1987) defines the primary dose as

$$D_{\text{prim}} = D_a(d) f(x,y) \text{TAR}(d,0)$$

where

$D_a(d)$ = dose in air for a given field size at depth, d .

$\text{TAR}(d,0)$ = tissue air ratio for zero area field size at depth, d .

$f(x,y)$ = a function dependent on penumbra (collimator transmission, source size) and type of filter.

The function $f(x,y)$ is found by comparing measured and calculated data in an iterative fashion.

The scattered component of dose, unlike the primary component, depends on field size and shape. It can be calculated by dividing the field into wedge shaped sectors and summing the scatter air ratio (SAR) values for each sector to obtain the average SAR for the field (Johns & Cunningham 1983):

$$D_{\text{scatt}} = \frac{\Delta\theta}{2\pi} \sum_i \text{SAR}_i(d,r)$$

Thus, the total dose is the sum of that from primary and scatter:

$$D_{\text{tot}} = D_{\text{prim}} + D_{\text{scatt}}$$

Although this method is conceptually appealing, it does have limitations. It has been pointed out (Mohan & Chui 1986) that this method assumes absorbed dose is equal to KERMA. This is a good approximation in many cases but not for high photon energies or near beam edges where electron equilibrium does not exist. Also, the concept of zero area TAR is flawed since the primary dose actually becomes zero as the field size becomes zero (Nizin & Chang 1991; Mohan & Chui 1986).

Monte Carlo Techniques

Analytically, it is virtually impossible to accurately include the effects of electron transport such that one can compute absorbed dose from KERMA (Bjarngard & Cunningham 1986). However, one can simulate electron interactions with Monte Carlo techniques. In fact, Monte Carlo techniques provide the most complete description of dose distributions

in both homogeneous and nonhomogeneous materials (Horton 1987). Unfortunately, the computer time required to follow the hundreds of thousands, or even millions of photons, required to ensure high accuracy is currently prohibitive. For this reason, Monte Carlo techniques currently find little application in routine treatment planning.

Energy Deposition Kernels

Although Monte Carlo techniques are too time consuming to use on a routine basis, they can be used to resolve the dose vs KERMA dilemma. Photon beam algorithms (Mackie et al. 1988, Boyer & Mok 1985, Mohan, Chui & Lidofsky 1986, Ahnesjo, Saxner & Trepp 1992) have recently been introduced that calculate dose by combining the distribution of photons originating from the linac head with a kernel representing the interaction of these photons in the phantom. Specifically, the kernel describes the complete history of a given photon from the point of initial interaction to the point of complete energy deposition of the resulting electrons. These algorithms are attractive due to their sound physical basis, accuracy, and versatility. Once a kernel has been established, intensity modulated beam dosimetry can be accomplished by simply determining the effect on the primary photons.

The kernels can be divided into three types. The most basic type is the single voxel interaction kernel. It represents the fraction of energy deposited in a voxel at point r that is released by primary photons interacting in a

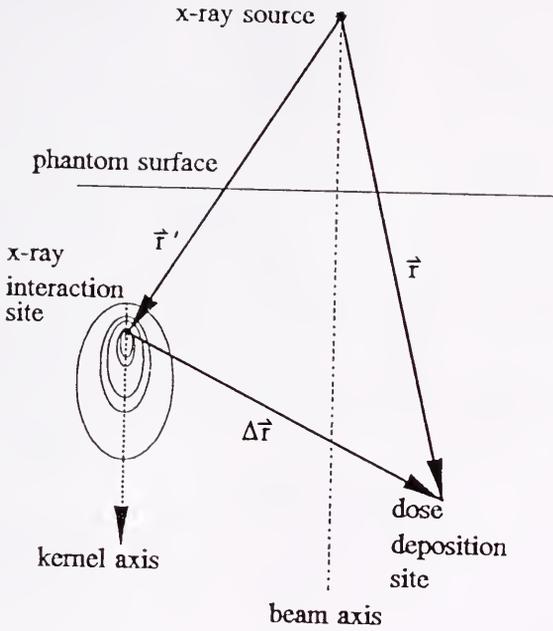


Figure 2-1 Relationship between the interaction site (\vec{r}') and dose deposition site (\vec{r}) using energy deposition kernels (Sharpe & Battista 1993).

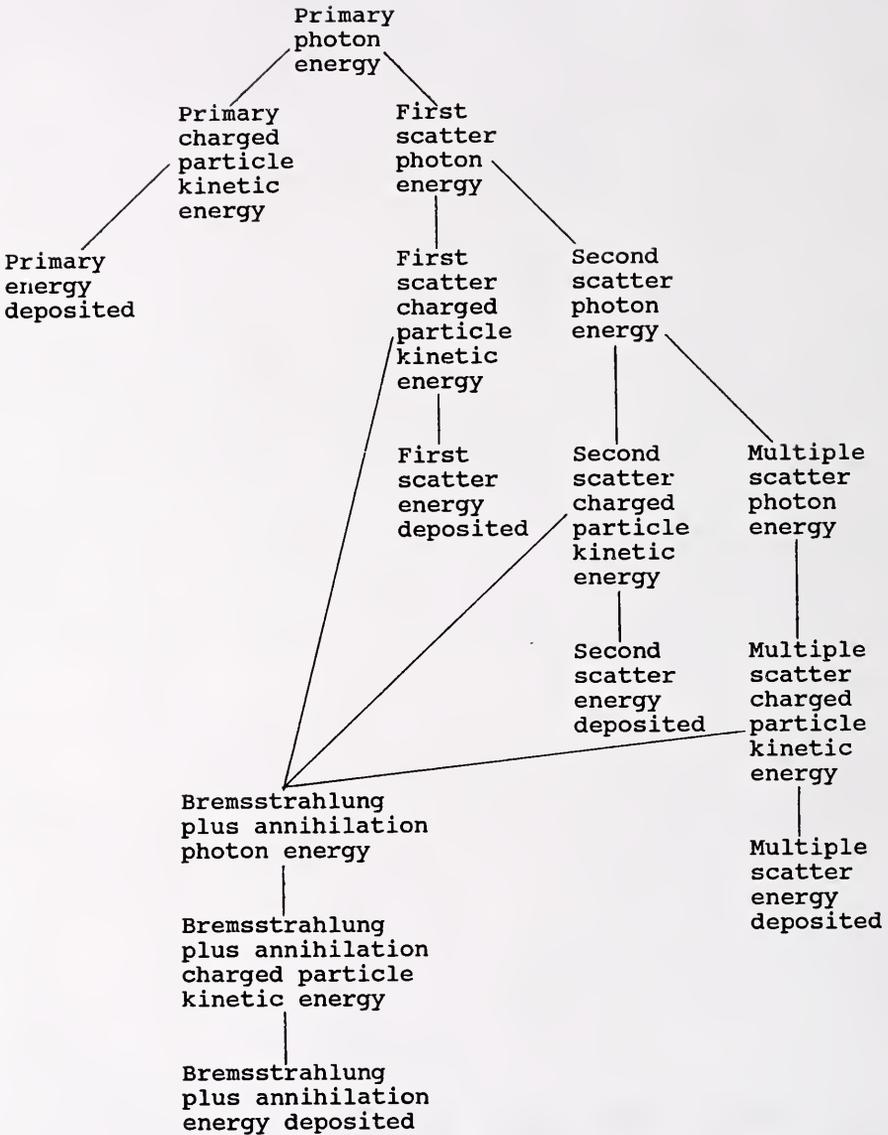


Figure 2-2. Energy deposition partitioning used in kernel development (Mackie et al. 1988).

voxel at point r' (see figure 2-1). Since it is experimentally impossible to make photons interact in a single voxel, the kernels are usually obtained through Monte Carlo modeling (Mackie et al. 1988, Mohan, Chui & Lidofsky 1986) or analytical approximations (Boyer & Mok 1985, 1986). However, the analytical models only follow scattered photons and not electrons thus assuming KERMA is equal to absorbed dose. Also, the analytical approach is limited to lower energies (e.g. ^{60}Co) since they incorporate Klein-Nishina cross sections for the Compton interaction but neglect other effects. Higher energy effects are more effectively modeled with the Monte Carlo method. For example, Mackie et al., 1988, has calculated energy deposition kernels using the Electron Gamma Shower (EGS) Monte Carlo code and includes the effects shown in figure 2-2. An example of such a kernel is shown in figure 2-3. This type of kernel is different from others mentioned in the literature. The pencil beam kernel terminology often encountered represents a dose distribution about a very narrow beam of radiation interacting with the phantom. This type of kernel can be obtained from Monte Carlo measurements (Ahnesjo, Saxner & Trepp 1992), depth dose measurements (Chui & Mohan 1988), or by convolving a single voxel interaction kernel with an attenuation term along a ray line through the phantom (Brahme 1988). An example of a pencil beam kernel is shown in figure 2-4. If several pencil beams interact at a voxel (point), a dose distribution referred to as a convergent

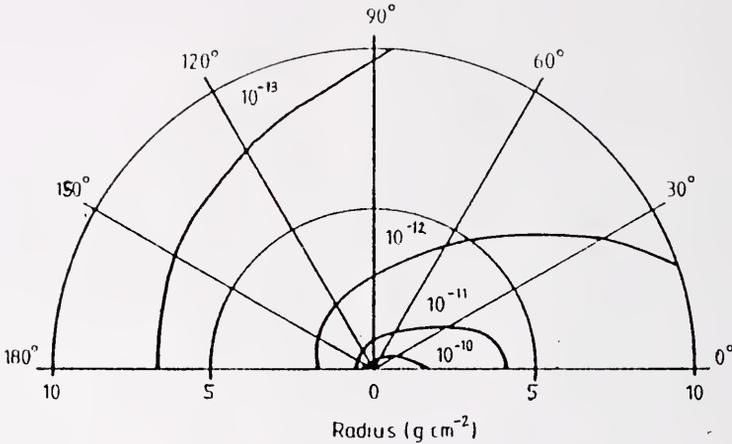


Figure 2-3. Isoline format of a total energy deposition kernel for 1.25 MeV photons in water. The units are $\text{cGy MeV}^{-1} \text{ photon}^{-1}$ (Mackie et al. 1988).

point irradiation kernel is found. If the irradiation of the voxel is from a 2π rotation in a plane or from all directions (4π) in three-dimensions, a very smooth kernel is formed representing the steepest dose gradient possible about the voxel. An example of a kernel for the 2-D rotation case is shown in figure 2-5. In the next section a method for solving the inverse problem in radiation therapy using these convergent point irradiation kernels will be

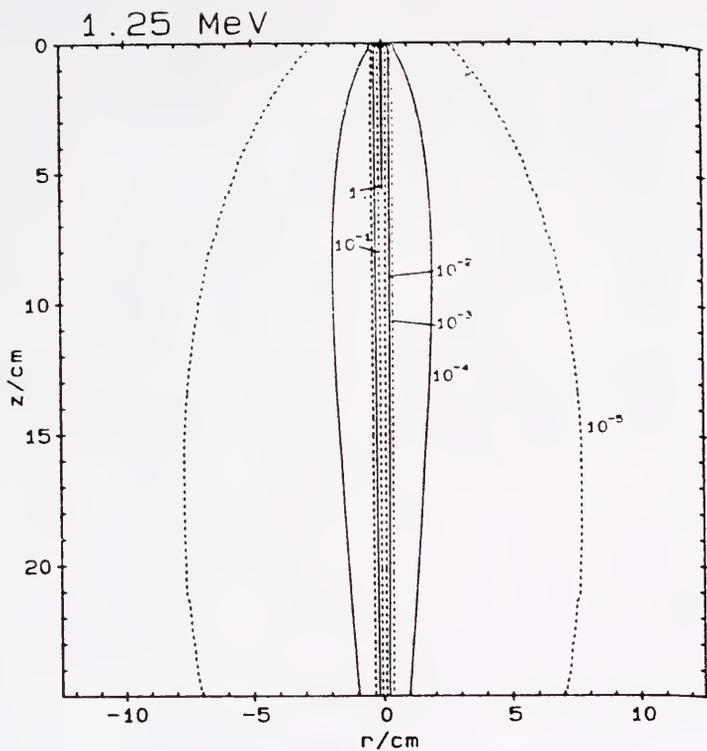


Figure 2-4. Isoline format for a 1.25 MeV pencil beam kernel. Units are fractional energy deposited per unit volume (cm^3) (Eklof, Ahnesjo & Brahme 1990).

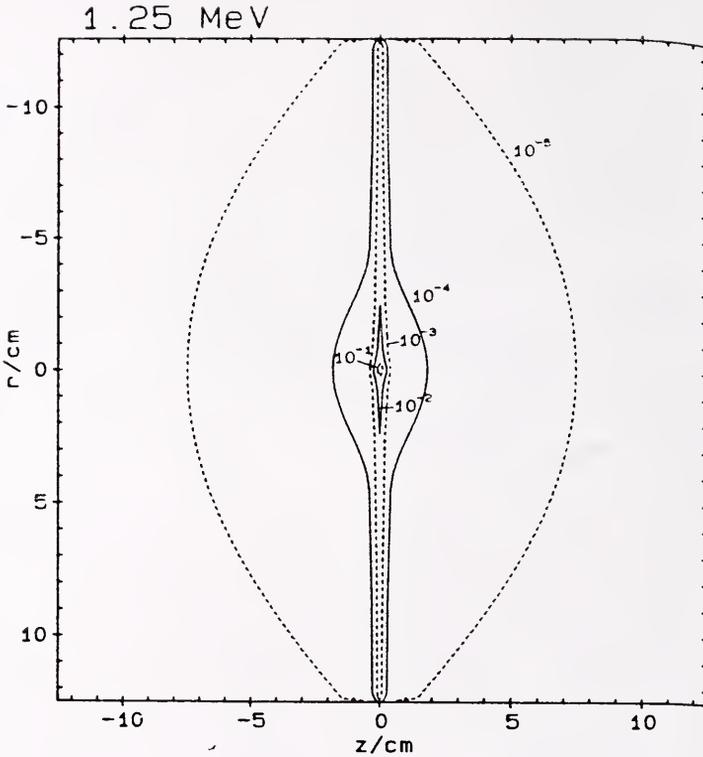


Figure 2-5. Isoline format of a convergent point irradiation kernel from a 360° rotation of a pencil beam in a single plane perpendicular to this page (Eklof, Ahnesjo & Brahme 1990).

presented. However, as will be shown, computational problems arise when this approach is applied to the stereotactic radiosurgery case.

The algorithm of interest for my research makes use of the single voxel interaction kernel. The relationship for the absorbed dose at r , $D(r)$, can be expressed as

$$D(r) = \int T(r') k(r-r') d^3r'$$

where $T(r')$ represents the distribution of total energy released to matter (TERMA) in each voxel of the phantom, $k(r-r')$ is the kernel representing the deposition of energy about a single interaction voxel, and d^3r' represents integration over all interaction sites in three dimensions. $T(r')$ is found from the primary energy fluence distribution, $\Psi(r')$, and the mass attenuation coefficient μ/ρ . For a homogeneous phantom, the TERMA can be expressed as

$$T(r') = \mu/\rho \Psi(r')$$

The first step in dose calculation is determining the primary energy fluence at each point in the phantom. One method of accomplishing this is to first obtain the in-air photon fluence distribution for the open field. Exponential attenuation is then applied along ray lines considering beam modification and patient anatomy (Mohan, Chui & Lidofsky 1986). The TERMA is then found as described above and convolved with the energy deposition kernel. If the kernel is invariant, the convolution can be achieved much more quickly by employing discrete Fourier transforms (DFT).

Situations resulting in noninvariant kernels include phantom heterogeneities, significant beam divergence, and beam hardening as a function of off-axis distance. However, if one is modeling a small diameter beam in a homogeneous media, the kernel can easily be treated as invariant. The assumption of an invariant kernel for larger fields will be discussed in chapter 8.

As mentioned, a convenient feature of this algorithm is its ability to incorporate a wide variety of complicating factors while retaining its simple physical foundation. Much of its flexibility lies in the reliance on primary energy fluence data which can easily be modified. In fact, even first order corrections for heterogeneities in a phantom can be accounted for by considering their attenuation effect on the energy fluence and then multiplying this fluence by the local attenuation coefficient to determine the number of primary photon collisions in the scattering voxel (Mohan, Chui & Lidofsky 1986). Further corrections may be included from density scaling along the path between the scattering voxel and the dose computation point. Another strength of the algorithm is the ability to model the buildup region and field edges where electron nonequilibrium exist. This is especially critical when trying to accurately model the steep dose gradient from multiple overlapping beams. It has been shown that for a 15 MV, 10X10 cm field, the lateral transport of secondary electrons alone contribute as much to penumbra as all other factors (geometry, photons scattered

from beam defining system, and photons scattered in phantom) combined (Mohan, Chui & Lidofsky 1986).

As a result of the features just mentioned, this model has a promising future and has even been used for traditional stereotactic treatment planning (Kusbad et al. 1990). However, it has been not been used for inverse treatment planning in stereotactic radiosurgery.

Inverse Radiotherapy Planning Approaches

Inverse radiotherapy planning is a specific case of the more general inverse problem in radiation transport which is applicable to nuclear engineering, oil well logging, atmospheric studies, medical imaging, etc. (McCormick 1992). While direct transport problems are concerned with estimating the particle distribution within and emerging from a prescribed medium, inverse transport problems are concerned with estimating the properties of an obscured medium with which particles have interacted and then emerged from. In the case of radiation therapy, for example, a direct transport problem would be the calculation of an absorbed dose distribution in a phantom from a photon beam that has been modified by a compensating filter. An inverse transport problem example would be estimation of the compensating filter properties given an absorbed dose distribution in a phantom.

Although there are several types of radiative transfer inverse problems, there are two general approaches for solving them: explicit and implicit (McCormick 1992).

Explicit approaches are analytical in nature and are typically applied to problems involving significant simplifications (e.g. plane-parallel illumination, plane-geometry, one spatial variable, etc.) (McCormick 1992). Regardless of these simplifications, the explicit solutions are suitable for a wide variety of realistic problems and may also serve as initial estimates for implicit methods. Implicit approaches are iterative in nature. Each iteration is simply a solution to the direct radiation transfer problem where the number of iterations performed depends on the desired level of accuracy.

In the following two sections, a summary of methods used to approach the inverse radiation therapy problem is presented. The first section will present explicit (analytical) methods while the second will present implicit (iterative) methods.

Analytical Methods

An analytical approach to inverse treatment planning is possible if simplifications are made. Early techniques assumed simple attenuation of a parallel incident photon beam and calculated the intensity modulation function (IMF) for 2-D circularly (Brahme, Roos & Lax 1982) and noncircularly (Cormack 1987; Cormack & Cormack 1987) symmetric dose distributions inside a cylindrical phantom, for arc therapy, with the axis of rotation coinciding with the origin of the phantom. A similar approach was later used to obtain the IMF for a 2-D circularly symmetric dose

distribution inside an arbitrarily shaped phantom with rotation about any axis (Barth 1990). Having accomplished this, one can consider 2-D nonsymmetric dose distributions by considering them as composed of adjacent, nonintersecting, radially symmetric dose distributions as shown in figure 2-6. Analytical solutions of this type are quickly obtained but do not account for photon scatter or electron transport (Kooy & Barth 1990). The solutions also assume rotational therapy thus excluding the option of a finite number of projections.

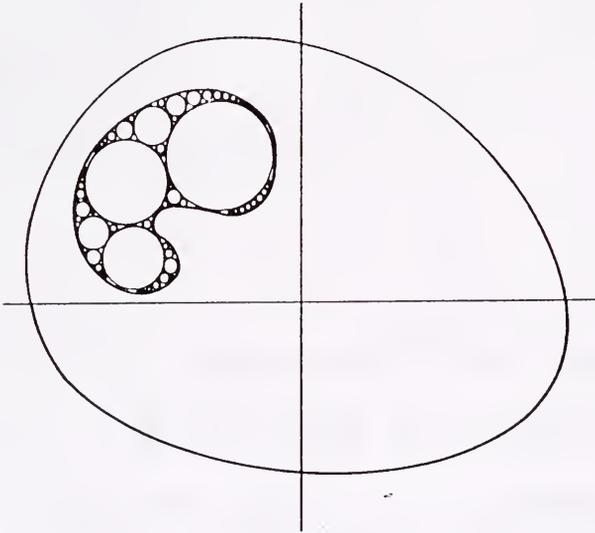


Figure 2-6. A nonsymmetric dose distribution composed of adjacent radially symmetric dose distributions located inside an arbitrarily shaped phantom (Barth 1990).

Another, quite different approach to obtaining IMFs for given beam projections was proposed by Mackie, Scrimger and

Battista (1985). It was proposed that by deconvolving a Monte Carlo generated single voxel energy deposition kernel from an ideal dose distribution in the phantom, the resulting primary energy fluence distribution could be obtained. This idea makes use of the kernels (discussed previously in Dosimetry Algorithms) which include effects of secondary-particle transport.

A related idea was actually investigated (Brahme 1988) with encouraging results. The problem was approached by dividing the target into many small voxels. For each voxel, the dose distribution resulting from a pencil beam kernel, rotated 360° while intersecting the voxel, was found. This rotated pencil beam dose distribution represents the convergent point irradiation kernel discussed earlier. As long as the symmetric, 360° irradiation by pencil beams about each voxel in the target is present, a single, invariant kernel can be used to calculate dose throughout the target volume and represents a Fredholm integral equation of the convolution type. The next step, thus, consisted of deconvolving the rotated pencil beam kernel from the desired dose distribution to obtain a 2-D function representing the number of convergent point irradiation kernels needed in each target voxel. This density function was then projected, as in CT, to obtain a 1-D modulation function for all incident beam directions. An example of this method, extended to include 3-D cases (Boyer, Desobry & Wells 1991), is shown in figure 2-7. As can be seen, this

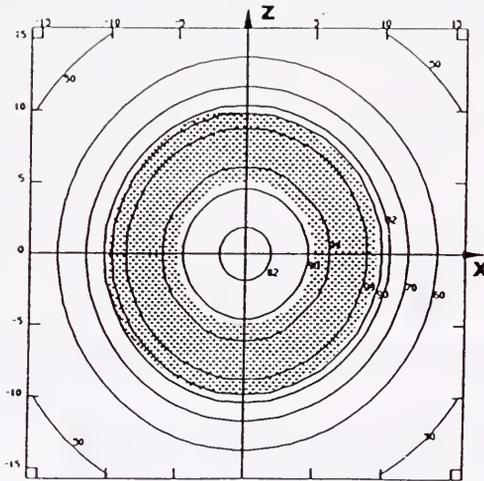
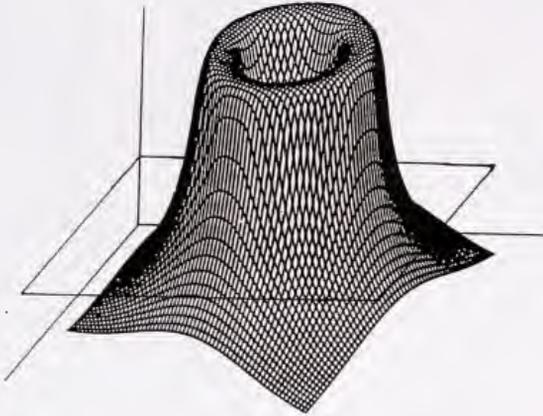


Figure 2-7. Resultant dose distribution for a torus shaped target (Boyer, Desobry & Wells 1991). The isodose lines shown at the bottom of the page represent a horizontal slice through the surface rendering above.

method produces good results which include higher order effects such as electron transport etc. and has been proposed as a treatment planning tool for tomotherapy (Holmes & Mackie 1994). However, situations involving nonsymmetric irradiation become difficult to solve since the rotated pencil beam kernel is no longer invariant. More realistic situations involving nonsymmetric irradiation can be crudely approximated by deconvolving symmetric irradiation kernels but projecting the resulting density function for only those incident beam projections chosen. Methods of choosing appropriate beam projections are discussed in Soderstrom and Brahme (1992). Otherwise, more cumbersome numerical techniques are required to solve the Fredholm integral equation of the first kind resulting from non - invariant point irradiation kernels (Lind & Brahme 1992).

Iterative Methods

One implicit method of obtaining an IMF is through algorithms similar to those used in computed tomography. Recall, that in CT, a 1-D transmission function is measured and represents the projection of a 2-D density distribution of that slice of the patient for a given projection angle. Analogously, a 1-D IMF for a given projection angle can be determined by first projecting the ideal 2-D dose distribution. This comparison is illustrated in figure 2-8.

Due to much similarity, it has been proposed (Bortfeld et al. 1990) that CT algorithms can be transferred to this

new application. However, one significant difference between CT and radiation therapy "backprojection" lies in the distribution of values along each ray line. In CT the cumulative transmission value obtained from attenuation through the patient slice is evenly redistributed along the same ray line during backprojection. Of course, this is not possible when backprojecting a physical beam due to radiation interactions (attenuation and scattering).

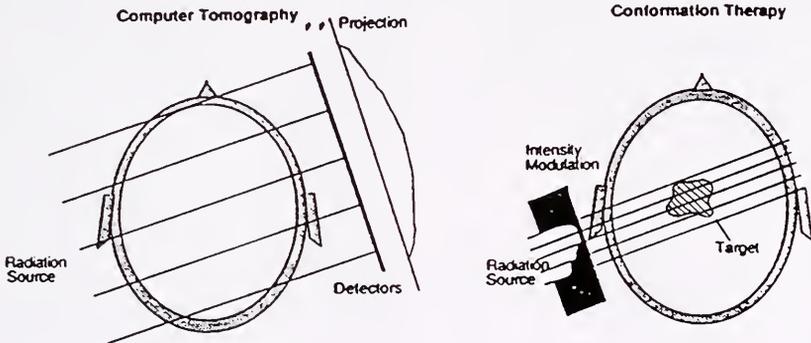


Figure 2-8. Sketch comparing computed tomography and intensity modulated radiotherapy (Bortfeld et al. 1990).

Unlike the analytical approaches just mentioned, a fixed number of projection angles can be determined thus giving more flexibility than rotational therapy. Of the algorithms used in CT, the filtered backprojection followed

by iterative optimization has been investigated (Bortfeld et al. 1990). Filtered backprojection provides a good first guess which is then used as input to an iterative algorithm. The results seem to indicate that the expense of using more than seven or nine beams was not justified with improved results. Unfortunately, no consideration was given to inhomogeneties and scattering was only approximately included (Bortfeld et al. 1990).

Another type of implicit method relies on direct transport and numerical optimization alone. Of the many treatment plan optimization studies accomplished, only a select few have truly addressed the inverse radiotherapy problem by including intensity modulation of the incident beam (Soderstrom & Brahme 1993; Webb 1989, 1991a, 1991b). These studies approach the problem by dividing each incident macroscopic beam into multiple elemental beam segments. Multidimensional optimization routines are then used to vary the intensity of each elemental beam while keeping score of the cumulative absorbed dose distribution in the phantom. The number of variables involved in such an optimization problem can be quite large even for 2-D cases. For example, allowing ten incident beams with each beam being composed of ten (2-D case) or 100 (3-D case) elemental beams would result in 100 and 1000 variables respectively.

It has been shown that the solution space to such an optimization problem is filled with local minima traps (Soderstrom & Brahme 1993). The most effective approach to

such a problem is using a simulated annealing algorithm which will be discussed in chapter 4. This approach, used by Webb, is the most comprehensive and has been extended to 3-D geometries although the inclusion of scatter is currently limited to the 2-D case. An advantage of this type of implicit approach is the degree of complexity allowed in the planning process. However, the price paid is the extensive computing resources needed.

A compromise between the two inverse problem approaches would be an analytic 3-D deconvolution approach to arrive at any individual BEV intensity modulation function followed by an iterative beam weight optimization. In this case, the number of optimization variables would be limited to the number of incident macroscopic beams. Such an approach is the topic of this report and has significant advantages. First of all, the deconvolution of single voxel interaction kernels and subsequent backprojection permit solutions for any desired number and direction of incident beams. Thus, significant planning flexibility is possible. Also, by using the energy deposition kernels, photon and electron transport are included resulting in realistic dosimetry based on fundamental principles. A third advantage results from optimizing macroscopic beam weights instead of individual beam elements. Due to the much fewer number of variables, a significant increase in speed can be expected. The details of this approach are discussed in the next several chapters and will be followed by several example test cases.

CHAPTER 3 DOSIMETRY CALCULATIONS

Introduction

This chapter describes the 3-D dosimetry algorithm used for the inverse treatment planning approach of this research. For clarity, the discussion will be limited to unmodulated, monoenergetic photon beams and flat, homogeneous water equivalent phantoms. Applications involving intensity modulation, polyenergetic beams, and irregular surfaces will be considered in subsequent chapters.

The dosimetry algorithm used was chosen for its sound physical basis, accuracy, and ease of incorporation into a 3-D inverse radiotherapy protocol. As described earlier, the absorbed dose is determined through convolution of an energy deposition kernel and a TERMA distribution resulting from the primary beam. I will first discuss the characteristics of the energy deposition kernel and TERMA. I will then illustrate how they are incorporated into a dosimetry algorithm through convolution.

Energy Deposition Kernels

The kernels used for my research are differential pencil beams (DPBs) developed at Memorial Sloan Kettering Cancer Center in New York (Mohan, Chui & Lidofsky L, 1986). These kernels were specifically created for photon based

dosimetry in a tissue equivalent medium and are available to anyone interested in their applications. These kernels are equivalent to those generated by Mackie et al. (1988) but are provided in ASCII format for machine compatibility. The DPBs, as received, have units of

$$\frac{\text{Energy deposited in calculation voxel (MeV) per cm}^3}{10^6 \text{ Monoenergetic photons at interaction point}}$$

associated with each and every annulus shaped voxel within a cylindrical water phantom. The initial interaction for every incident photon was made to occur at the center (interaction point) of the cylinder and the resulting location of energy deposition from secondary particles was determined via Monte Carlo methods. The bins were scaled in a logarithmic fashion such that the radial and height dimensions of each annulus is 10% larger than the previous one. This arrangement allows fine resolution near the interaction point and a coarse resolution further away. The DPBs, as received, were also divided into two components. A short range component included energy deposition from electrons produced in the initial photon interaction at the interaction point (due to Compton electrons, pair production electrons, and photoelectrons). A long range component included energy deposition resulting from subsequent interactions of Compton photons, Bremsstrahlung, and positron annihilation. This arrangement arose from a desire to decrease the size of the computation array when performing dose computations. With this arrangement, a small, fine grid for the rapidly varying

short range component can be overlaid on a larger, coarse grid for the slowly varying long range component.

Upon receiving the files, an algorithm was first developed to determine the total and component energies for each of the 22 incident photon energies included in the library. This program, DPBPROC, is included in the appendix and a summary of the entire DPB library is shown in figures 3-1 through 3-3. Note that, at higher energies, the summed energy total of both components is slightly less than that of the incident photon. This, of course, is due to the finite size of the cylindrical phantom which was approximately 2.5 meters in both height and diameter. Using DPBPROC, the components were then summed to produce a single, complete DPB for each photon energy in the library.

Next, a second algorithm, DPBVOX, was developed to convert the summed DPBs into a linearly scaled Cartesian coordinate format needed for the convolution process. This was accomplished by first obtaining a volume weighted average of all logarithmic scaled voxels lying within the limits of a given linearly scaled voxel. This provided an average energy deposition density (MeV/cm^3 per 10^6 incident photons) for each linearly scaled annulus shaped voxel. The linearly scaled annulus voxel format was then converted to that of equal sized cubes along the radial (x or y axis) and longitudinal (z axis) directions by multiplying the energy density at each position by the appropriate cube volume. Finally, the DPBs were normalized by dividing the deposited

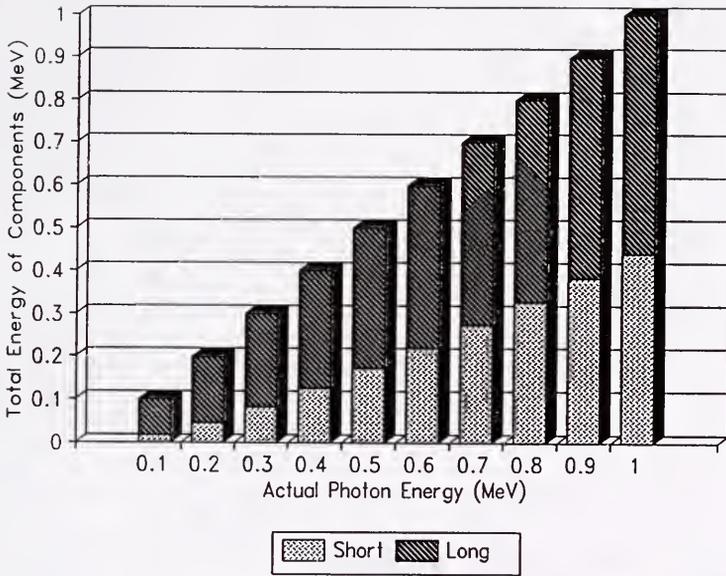


Figure 3-1. Differential pencil beam components, 0.1--1 MeV.

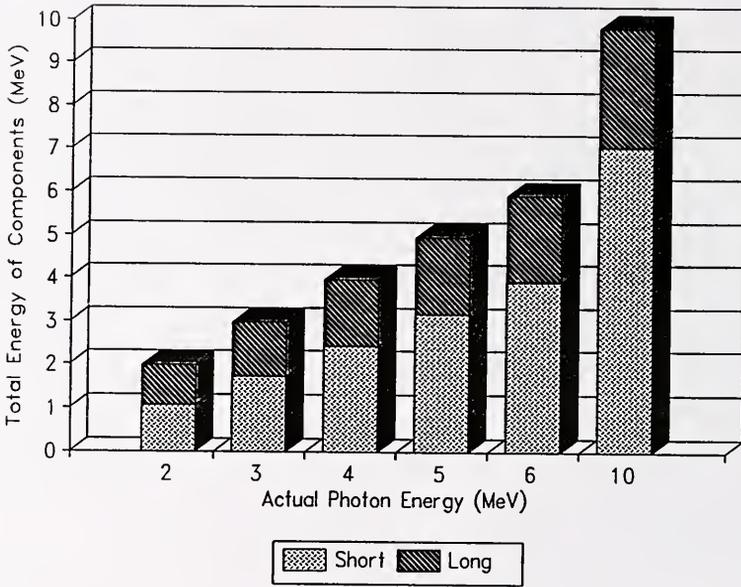


Figure 3-2. Differential pencil beam components, 2--10 MeV.

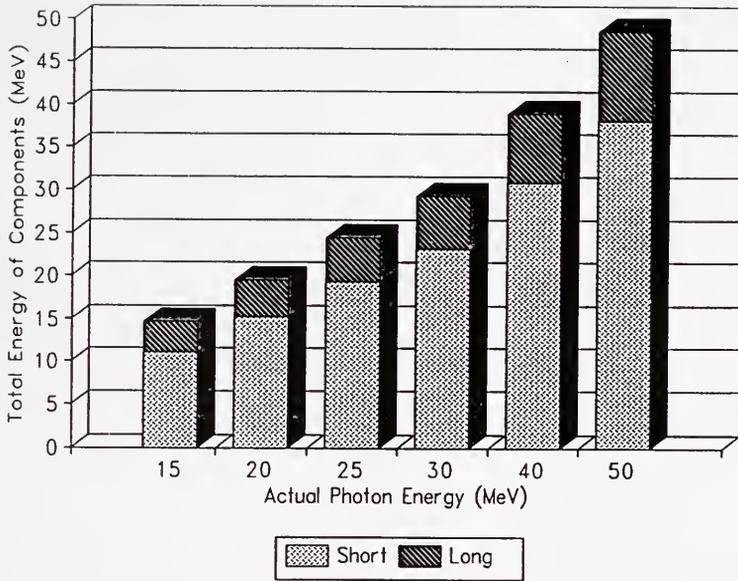


Figure 3-3. Differential pencil beam components, 15--50 MeV.

energy assigned to each voxel by the energy per incident photon (in MeV) and number of incident photons (10^6). Thus, the DPBs may now be expressed in units of

$$\frac{\text{Energy deposited in calculation voxel}}{\text{Energy released at interaction point}}$$

representing the fraction of incident photon energy that ends up being deposited in a particular calculation voxel. Having characterized the DPBs as needed in the x- (or y-)z plane, a spreadsheet template was used to produce a 3-D version. One simply enters the linearly scaled values for each radial (x or y axis) and depth (z axis). The spreadsheet then makes use of radial symmetry and assigns values to the remainder of the x-y plane for every depth. The radial and depth characteristics of the 2 MeV DPB in Cartesian format is illustrated in figure 3-4. In this example, the grid spacing is 2 mm. Other spacing can be chosen as needed. However, the uniformity of spacing cannot be relaxed due to restrictions imposed by the FFT convolution procedure. A comparison of the linearly scaled Cartesian coordinate format DPB data and the original logarithmically scaled cylindrical coordinate format is shown in figure 3-5. Note that the linearly scaled Cartesian coordinate curves are much smoother than the raw Monte Carlo data thus significantly reducing any high frequency noise during convolution/deconvolution.

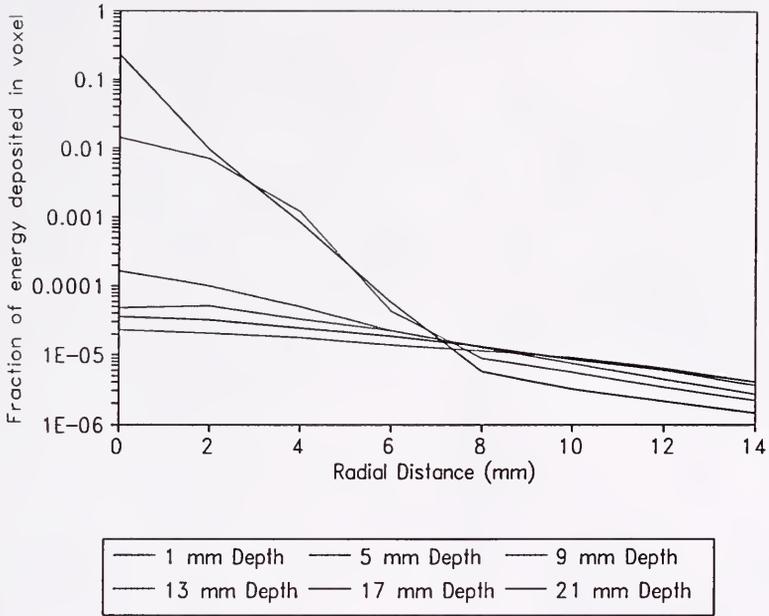
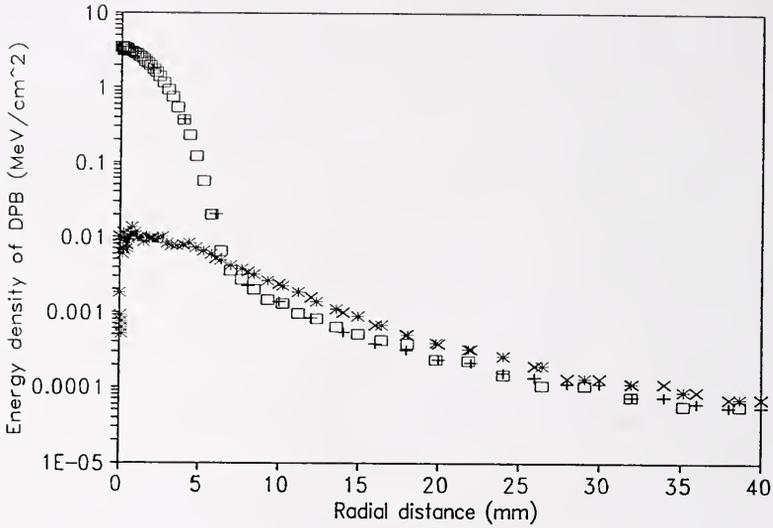


Figure 3-4. Uniformly scaled (2 MeV, 2x2x2 mm³ voxels) differential pencil beam data for selected depths.



□ Raw 5.23 mm + Avg 5 mm * Raw 14.91 m × Avg 15 mm

Figure 3-5. A comparison of the linearly scaled Cartesian coordinate (avg) and logarithmically scaled cylindrical coordinate (raw) format for the 2 MeV, $2 \times 2 \times 2 \text{ mm}^3$ case at depths of 5 and 15 mm.

TERMA

The energy released at any location within a phantom can be described in terms of TERMA. As mentioned earlier, the TERMA at a given point represents the total energy released to matter (water phantom) for a given number of incident photons at that point. It can be found by applying inverse square and attenuation to a calibrated source of photons:

$$\text{TERMA} = \left[\Psi_r \left(\frac{\text{MeV}}{\text{cm}^2} \right) \right] \left[\frac{\mu}{\rho} \left(\frac{\text{cm}^2}{\text{g}} \right) \right] \left(\frac{s}{r} \right)^2 e^{-\left(\frac{\mu}{\rho} \right) \rho (r - \text{ssd})}$$

where

Ψ_r = reference incident energy fluence for Monoenergetic photons at a specified distance from the source.

$\frac{\mu}{\rho}$ = mass attenuation coefficient of phantom for

monoenergetic photon energy.

s = distance from source to reference distance.

r = distance from source to TERMA calculation point.

ssd = distance from source to surface of phantom.

The TERMA at each interaction point is expressed in terms of energy released, E_{rel} , per mass, m, and per incident photon energy fluence (e.g. [pJ/kg]/[MeV/cm²]). Note that the energy per mass is not referred to as absorbed dose in this case since the energy is that released at the interaction point but not all deposited, E_{dep} , at that location. By choosing units as described above, it is seen

that the product of TERMA and a DPB in a convolution operation yields absorbed dose per reference energy fluence (D/Ψ_r):

$$\left(\frac{E_{\text{dep}}}{E_{\text{rel}}}\right) \otimes \left(\frac{E_{\text{rel}}}{m \cdot \Psi_r}\right) = \left(\frac{E_{\text{dep}}}{m}\right) / \Psi_r = \frac{D}{\Psi_r}$$

Thus, the absorbed dose at any point in the phantom can be found if the reference energy fluence is known for the incident beam. However, determination of energy fluence for an intense source of high energy photons is not easily accomplished. I have devoted all of chapter five to this topic. For the purpose of this chapter, I will simply present all output in relative units.

Convolution

The convolution approach used for dose calculation is very similar to that often used in image processing. Instead of convolving an image with, for example, a smoothing or edge enhancement kernel, a TERMA distribution is convolved with a DPB. The FFT is used to speed up the convolution process. As previously mentioned, this is especially useful when working with large 3-D arrays. Additionally, once the procedure for FFT convolution is established, deconvolution becomes an option. In fact, the deconvolution process is used to determine the desired intensity modulation for a given incident beam and will be described in chapter 6.

The FFT algorithm chosen for this work is a multi-dimensional routine (Press et al. 1992). As input, one

provides the number of dimensions involved, a dimension vector identifying the length of the array in each dimension, the array of data, and an integer (± 1) to indicate FFT or FFT⁻¹. The array of data is actually accessed as a 1-D array with alternating real and imaginary parts. Thus the dimension vector is used to interpret the arrangement of data within the 1-D array. Finally, the length of the data array in each dimension must be a power of two.

The FFT routine was incorporated into the dosimetry algorithm as illustrated in figure 3-6. The DPB is first read from a file and the FFT is computed. Next, the TERMA is computed and its FFT found. The convolution is then accomplished via the product of the two FFTs in frequency space and then performing an inverse FFT (Starkschall 1988). It has been shown (Boyer 1984) that the factor by which a Fourier transform of two functions decreases the calculation time over traditional convolution is equal to

$$N^3 / (3 \text{ Log}_2 N)$$

where N is the number of calculation points in each dimension of a cube. Thus, over a 50x50x50 voxel region, a time savings of 7,400 would be realized. In general, there is a linear relationship between calculation time and $N \text{Log}_2 N$ (Boyer, Wackwitz & Mok 1988). A comparison of the relative usefulness of Fourier vs direct processing convolution is shown in figure 3-7 for 2-D situations. Here it is shown that only when the dimensions of the impulse array are small

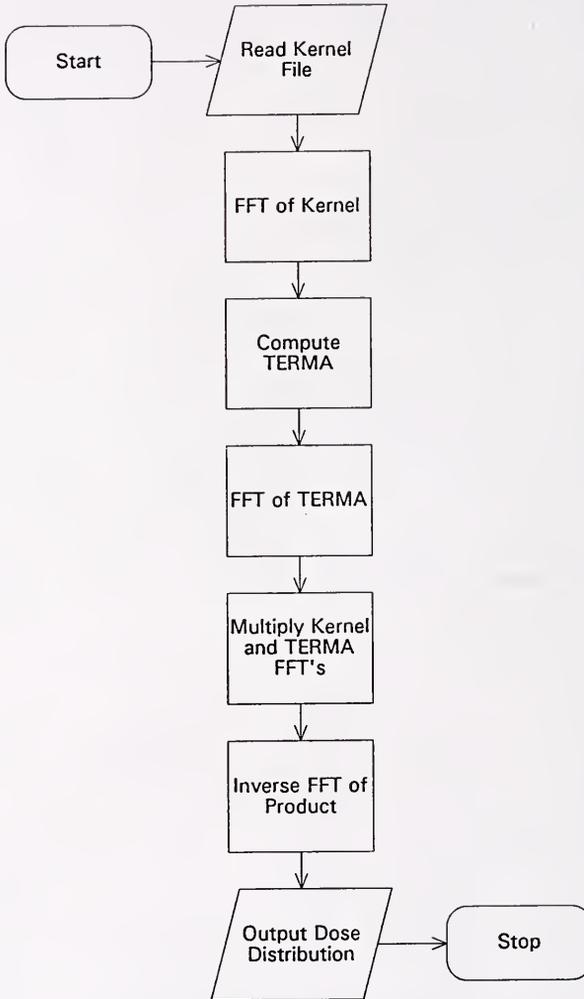


Figure 3-6. General flowchart of dosimetry algorithm.

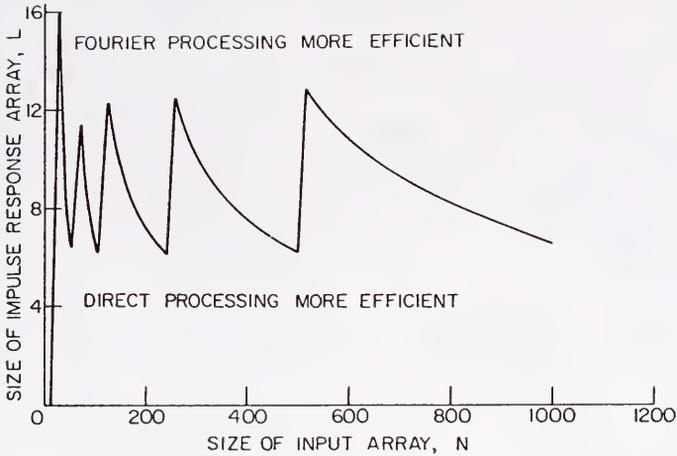


figure 3-7 Comparison of direct and Fourier domain processing for finite area convolution in two dimensions (Pratt 1978).

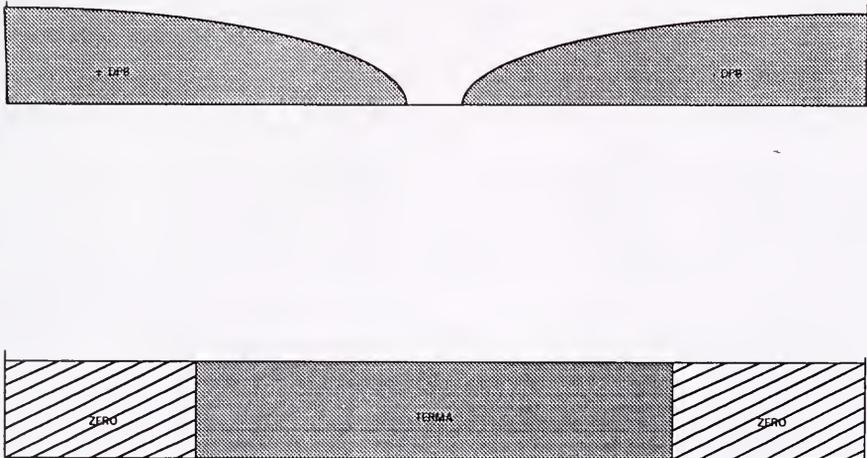


Figure 3-8. Example of wrap-around format for DPB and zero padded TERMA in one dimension.

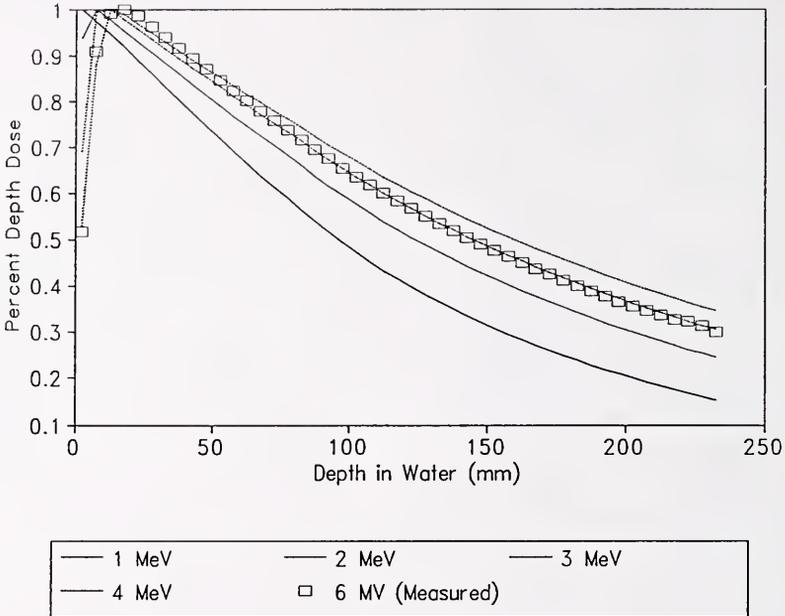


Figure 3-9. Comparison of measured percent depth dose to that calculated from monoenergetic differential pencil beam convolution (FS = $5 \times 5 \text{ cm}^2$, SSD = 100 cm, $5 \times 5 \times 5 \text{ mm}^3$ voxels).

relative to the input (signal) array is direct processing more efficient. For this research, since the range of the DPB arrays extend to the dimensions of the calculation volumes of interest, Fourier processing is more efficient. Extension of this trend to 3-D situations would be even more dramatic. Although the convolution process is relatively straightforward, one must take great care in the arrangement and format of the arrays to be convolved. The convolution theorem presumes two conditions which must be met (Press et al. 1992). First, the arrays must have the same dimensions. This criteria can be accomplished by zero padding the smaller array. A second, more difficult constraint, is the assumption that the signal (TERMA) and response (DPB) are periodic. If this is not the case, significant wrap-around artifacts will occur. This requirement may be met by zero padding the edges of the signal function and putting the response function in wrap-around order. The amount of zero padding required depends on the non-zero extent of the response function. An example of the zero padded TERMA and wrap-around format for the response function in a single dimension is shown in figure 3-8. Percent depth dose data produced by the dosimetry algorithm for 1, 2, 3, and 4 MeV photons are displayed in figure 3-9 and compared to measured 6 MV data. More accurate modeling requires a polyenergetic approach as described in the next chapter.

CHAPTER 4 BREMSSTRAHLUNG SPECTRA

Introduction

This chapter describes the method used to determine the effective beam spectrum for a given linac and collimating system. This spectrum is used to model the clinical x-ray beam in all treatment planning dosimetry as opposed to assuming a monoenergetic beam of energy equal to one third of the nominal accelerating potential.

Although an approximate estimation of a linac spectrum is possible using analytical methods (Desobry & Boyer 1992; Koch & Motz 1959), the complexity and variability of linac physical parameters limit the usefulness of such an approach. Thus, most estimates of linac spectra are based on Monte Carlo simulations or experimental measurements.

The Monte Carlo simulations have proven to be fairly accurate as long as sufficient detailed information of the linac head is available (Mohan, Chui & Lidofsky 1985; Kusbad et al. 1990; Lo 1992). However, such simulations require computing resources not readily available in most clinical settings. Also, regardless of their sophistication, Monte Carlo simulations should always be experimentally verified.

A variety of experimental techniques have been investigated and include off-axis Compton scattering spectroscopy (Faddegon, Ross & Rogers 1991, Levy et al.

1974, 1976), photoactivation (Nath & Schulz 1976), attenuation analysis (Dawson 1989; Huang, Kase & Bjarngard 1983, Archer, Almond & Wagner 1985), and numerical reconstruction from depth dose data (Sauer & Neumann 1990; Ahnesjo & Andreo 1989). Of these experimental techniques, only those based on numerical reconstruction of depth dose data rely on readily available measured clinical data. As will be shown, by using such measured data and the energy deposition kernels previously discussed, a link between the dosimetry model in question and the calibration of specific hardware (i.e. SL75/5 linac) can be achieved.

Spectrum Model

The algorithm developed for determination of the effective beam spectrum, SPECTRUM, is illustrated in figure 4-1. As input, two different sets of data are required. First, experimentally measured data are obtained for the open field (unmodulated) collimating system of interest. For example, a 5x5 cm² field size at SAD may consistently be used, in conjunction with a beam intensity modulator, for all treatments (further modulation of the beam by the attenuation device is considered in the dosimetry algorithm). The measured depth dose data requires units of absorbed dose per monitor unit (Gy/MU) and was obtained by applying calibrated output and field size factors (AAPM Task Group 21, 1983) to percent depth dose measurements for the field size of interest. For the small field sizes typical of radiosurgery, the beam quality does not vary

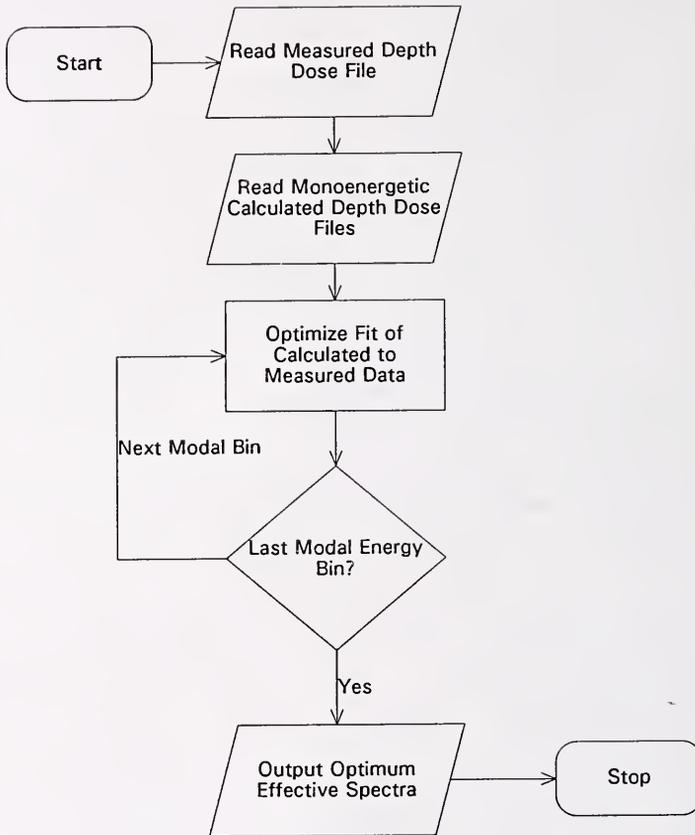


Figure 4-1. General flowchart of SPECTRUM algorithm.

significantly across the field. Thus, central axis data alone may be used to characterize the beam (The potential of extending this method to larger fields will be discussed in chapter 8).

The second set of data required is absorbed dose per reference energy fluence and per monitor unit for each photon energy to be included in the effective spectrum. The "reference" term used here indicates a point in space between the photon source and the surface of the phantom. Although any distance may be chosen, it is convenient to place the reference point at the location of the beam modulating device. For the purposes of this research, a value of 65 cm from source to center of modulating device was chosen. This data is created by the monoenergetic dosimetry model discussed in the previous chapter and is shown in Figures 4-2 and 4-3.

Obviously, the number of energy bins included in the spectrum is a compromise between speed and accuracy of dosimetry calculation. For example, it has been shown (Mackie, Scrimger & Battista 1985) that a monoenergetic approximation of 5 MeV duplicates measured depth dose values to within 5% up to depths of 20 cm for a 10 x 10 cm, 15 MV beam. Modeling their spectra with five energy bins reduced the error to within 0.5 %. Thus, based on these results, and the availability of DPB energies, seven energy bins were chosen for my research (0.5, 1, 2, 3, 4, 5 and 6 MeV). Any more energy bins than this would increase the computation

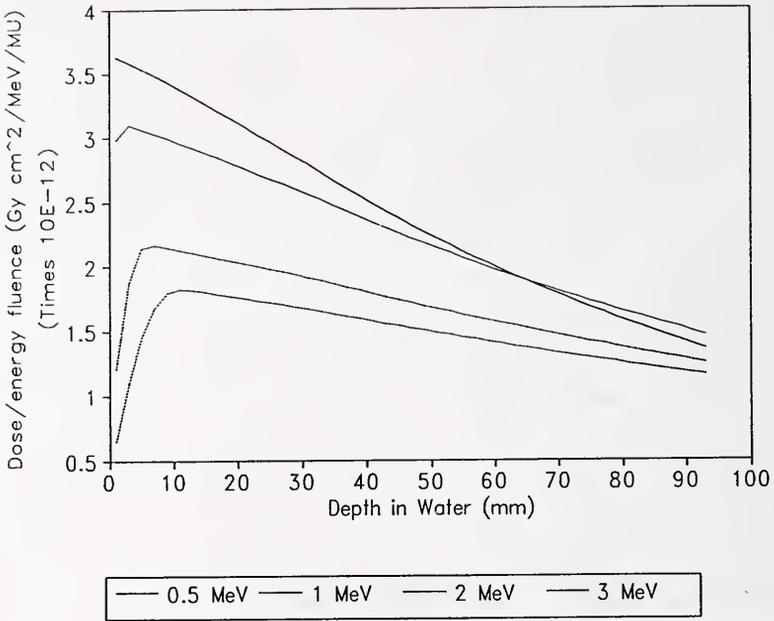


Figure 4-2. Absorbed dose per energy fluence for four monoenergetic photon energies.

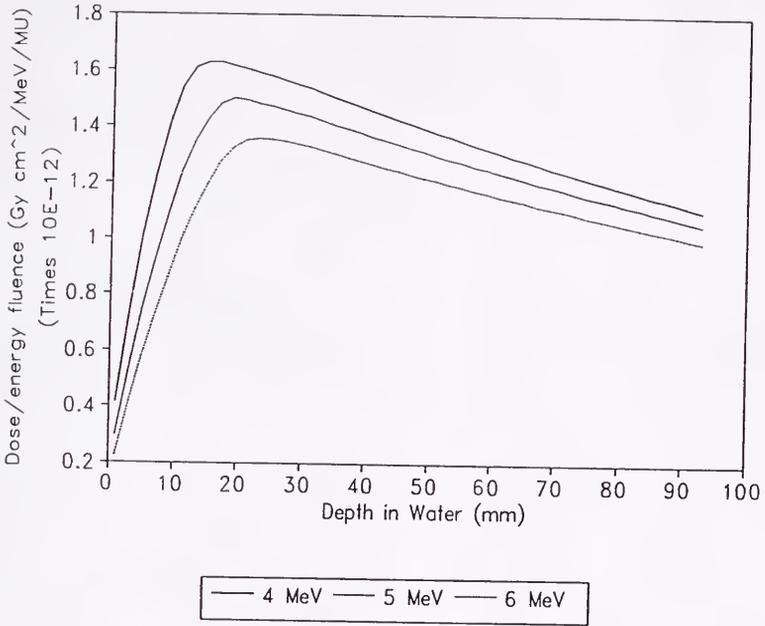


Figure 4-3. Absorbed dose per energy fluence for three monoenergetic photon energies.

time with no significant increase in dosimetry accuracy and would require interpolation between the energies provided in the kernel library.

Once the input data files are read, SPECTRUM then proceeds to optimize the fit between measured and calculated depth dose data. The objective function (a measure of the fit) to minimize is similar to that used by Ahnesjo (Ahnesjo & Andreo 1989):

$$\text{objfunc} = \sum_z \left| \sum_E d_{Ez} \Psi_r(E) - D_z \right|$$

where

E = photon energy of given spectral bin.

d_{Ez} = dose per reference energy fluence data ($\text{Gy MeV}^{-1} \text{ cm}^2 \text{ MU}^{-1}$) as a function of photon energy, E , and depth in the phantom, z .

$\Psi_r(E)$ = reference energy fluence (MeV cm^{-2}) for each photon energy, E .

D_z = measured dose per monitor unit data (Gy MU^{-1}) for depths, z .

z = Depth in phantom (cm) where difference between measured and calculated data is found.

Thus, at each depth, the difference is found between calculated and measured absorbed dose. The actual depth values are based on the voxel size used in the dose calculation. The total objective function, a sum of the absolute value of these differences, is then normalized for the total number of depths used. A good fit to measured data

is found by choosing the proper weighting of energy fluence for each photon energy bin comprising the spectrum. An optimum weighting is referred to as the "effective spectrum" as opposed to a physically accurate spectrum. This is due to both the discrete polyenergetic dosimetry modeling and the weak energy dependence of depth dose on the spectral shape. These conditions result in multiple "solutions" to the problem.

Due to such a weak energy dependence of depth dose data, two practical constraints were imposed to help guide the search through solution space towards more realistic solutions (Sauer & Neumann 1990). First of all, the energy fluence values for all energy bins are required to be equal to or greater than zero:

$$\Psi_n \geq 0 \text{ for all } n \text{ energy bins}$$

The second constraint limits the magnitude of energy fluence bins relative to their neighbors. This constraint assumes that a pure bremsstrahlung spectrum with no characteristic peaks should increase up to a modal energy (peak of energy fluence spectrum) and then steadily decrease. Thus, assuming a modal energy bin, n_{modal} ,

$$\Psi_n \leq \Psi_{n+1} \text{ for } n < n_{\text{modal}}$$

$$\Psi_n \geq \Psi_{n+1} \text{ for } n \geq n_{\text{modal}}$$

Since the modal energy bin is not known ahead of time, the optimization is automatically repeated n times letting each energy bin assume the modal bin role and the best fit is chosen from these solutions.

In addition to these constraints, the minimum depth, z_c , used in the objective function was limited to values beyond the depth of electron contamination. Thus, the fitted spectrum represents all primary and scattered photons which penetrate beyond z_c but excludes contaminant electrons and lower energy photons. Based on previous purging magnet experiments (Ahnesjo & Andreo 1989) an empirical relationship was found between the maximum depth of contamination, z_c (cm), and the nominal accelerating potential, E_0 :

$$z_c \leq 0.3E_0$$

Based on this relationship, the maximum depth of electron contamination would be approximately 1.8 cm for a 6 MV photon beam. For the SL75/5, 6 MV linac modeled in my research, a conservative minimum depth value of 2.25 cm was chosen.

Optimization Algorithm

The algorithm used within SPECTRUM for multi-dimensional (7 energy bins) optimization, AMEBSA (Press et al. 1992) is based on simulated annealing (Metropolis et al. 1953; Kirkpatrick, Gelatt & Vecchi 1983; Bohachevsky, Johnson & Stein 1986). This approach makes an analogy between the statistical behavior of large numbers of atoms in a liquid or solid system and multi-dimensional optimization problems. In such a system, there exist a large number of possible atomic configurations/energy states (local minima) although only one ground state (global

minimum) exists. The probability, $P(E)$, of a system being in a particular energy state, E , is described by the Boltzmann probability distribution:

$$P(E) \propto e^{-\left(\frac{E}{kT}\right)}$$

where k is Boltzmann's constant and T is the thermal equilibrium temperature. This relationship indicates that, even at low temperatures, a finite probability exists for the system to be in a higher energy state. The probability of a system changing from energy state E_1 to energy state E_2 is expressed as

$$P(\Delta E) \propto e^{-\left(\frac{(E_2 - E_1)}{kT}\right)}$$

This relationship indicates a high probability for $E_2 < E_1$ (downhill) and small but finite probabilities for $E_2 > E_1$ (uphill) changes. Using these statistical trends, the Metropolis based algorithm explores solution space, always allowing downhill steps and sometimes allowing uphill steps, as the temperature of the system is gradually decreased (see figure 4-4). As with real physical systems, if the temperature of the system is cooled too rapidly (quenched), the final resulting energy state will not be the ground state. However, if the temperature is cooled sufficiently slowly and the various system configurations are thoroughly sampled at each temperature step (by incorporating a random number generator), the global minimum is more likely to be found. The workhorse for AMEBSA is the downhill simplex algorithm, AMOEBA (Press et al. 1992). This method replaces

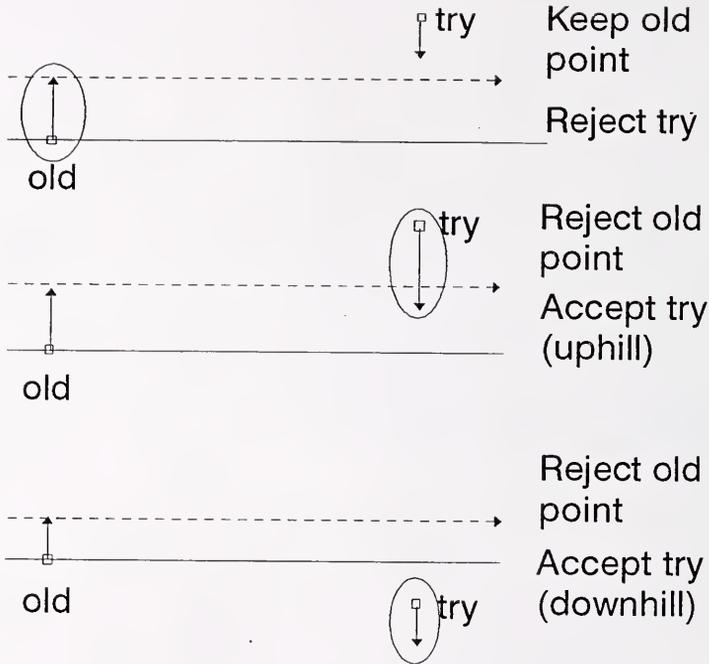


Figure 4-4. Example of accept or reject criteria used in the AMEBSA algorithm. One random number proportional to the current temperature is added to the objective function value associated with a point in solution space. A second random number is subtracted from the value associated with the potential replacement point. The resulting values are then compared.

a single point (solution array) description of the system with a simplex of $N+1$ points. The exploration of solution space is then accomplished by reflections, expansions and contractions of the simplex (see figure 4-5). As the temperature approaches zero, the AMEBSA algorithm reduces exactly to the downhill simplex method and converges to a local minimum.

Results

Figure 4-6 illustrates an "effective energy fluence" spectrum consisting of seven energy bins optimized for a 6 MV, $5 \times 5 \text{ cm}^2$ at SSD, 100 cm SSD beam from the SL 75/5 linac using $1 \times 1 \times 1 \text{ cm}^3$ calculation voxels. A comparison between the depth dose resulting from this spectrum and the measured depth dose is also shown in figure 4-6. It is obvious that the spectrum does not reflect the properties of a physically real Bremsstrahlung spectrum (see figure 4-7). The reason for this results from the relative insensitivity of depth dose data to spectral variations as previously mentioned. This trend is indicated in figures 4-8 and 4-9 where depth dose data is reconstructed using spectra more and less realistic than that shown in figure 4-6 with reasonable results. This trend is also supported by recent work (Zhu & Van Dyk 1993) indicating most changes in percent depth dose are caused by changes in the average energy of the beam rather than the specific spectral shape. For this research, the actual shape of the spectrum is not the critical issue. What is important is having a good fit to measured data with

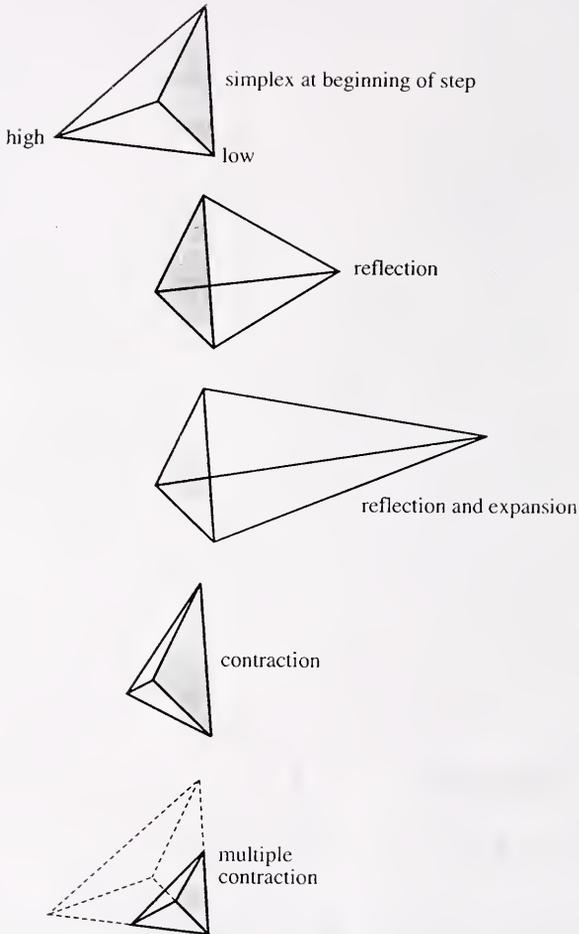


Figure 4-5. Possible outcomes for a step in the downhill simplex method. Such steps allow the simplex to change shape as needed and "crawl" through the solution space landscape.

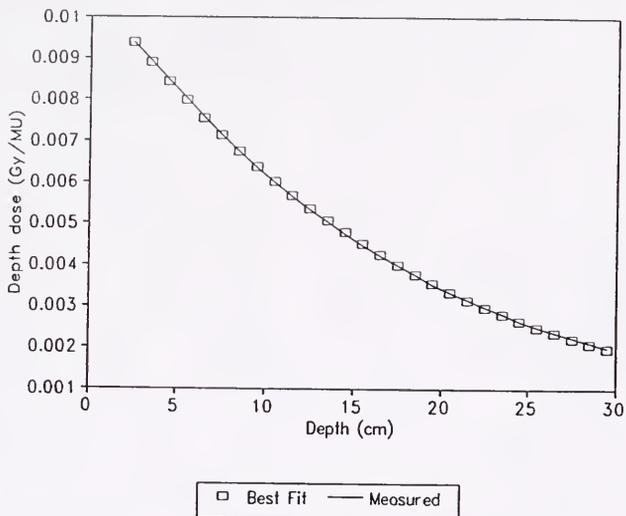
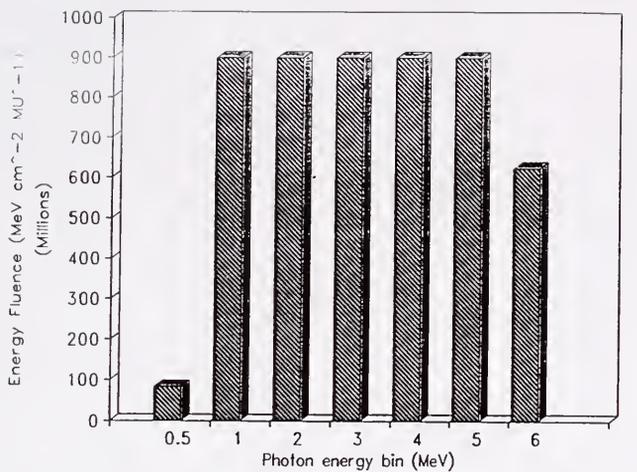


Figure 4-6. Above: Seven energy bin Bremsstrahlung spectrum output from SPECTRUM algorithm. Below: Comparison of calculated and measured depth dose data for 6 MV beam.

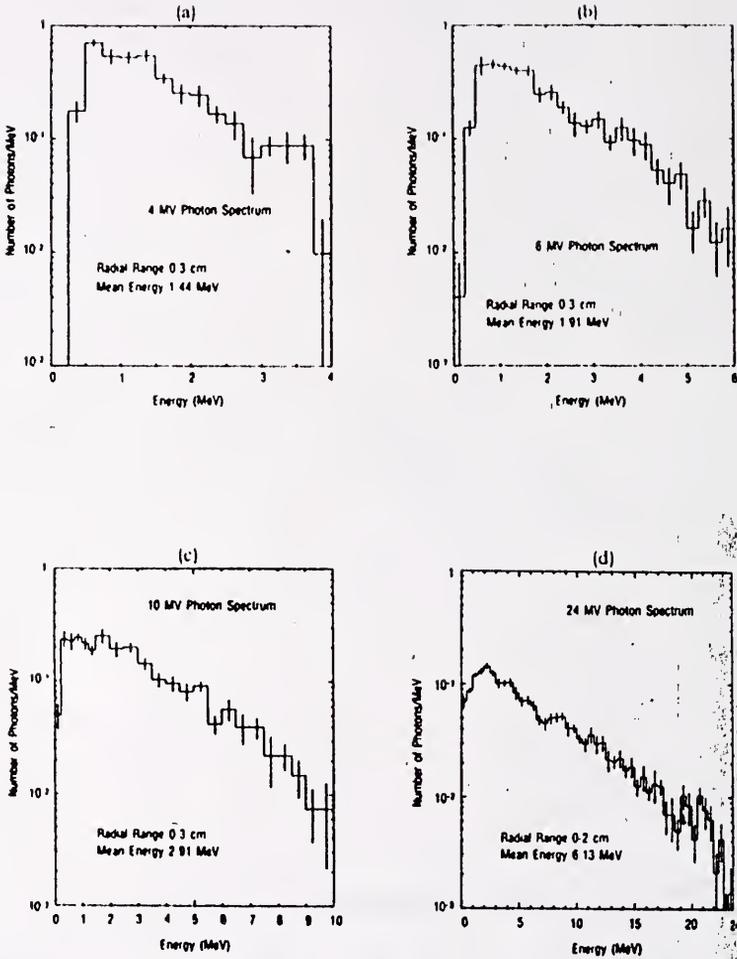


Figure 4-7. Example of realistic 6 MV spectra generated by Monte Carlo modeling for several linacs. a) Clinac 4 (4 MV), b) Clinac 6 (6 MV), c) Clinac 18 (10 MV), d) Clinac 2500 (24 MV) (Mohan, Chui & Lidofsky 1985).

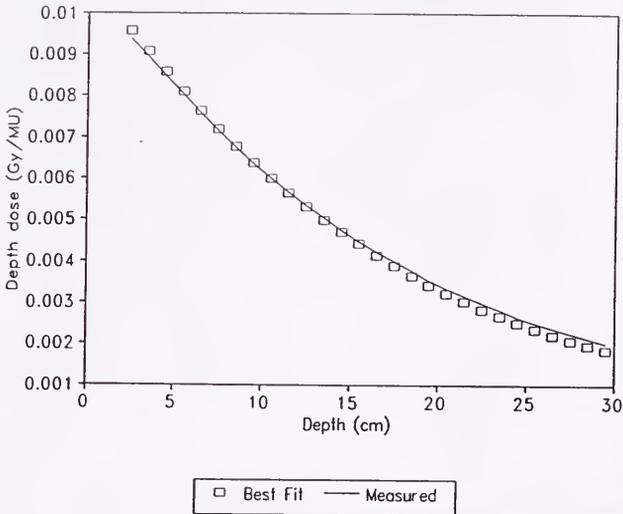
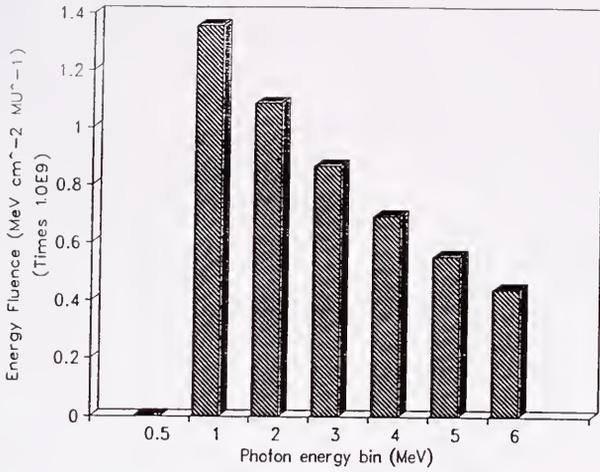
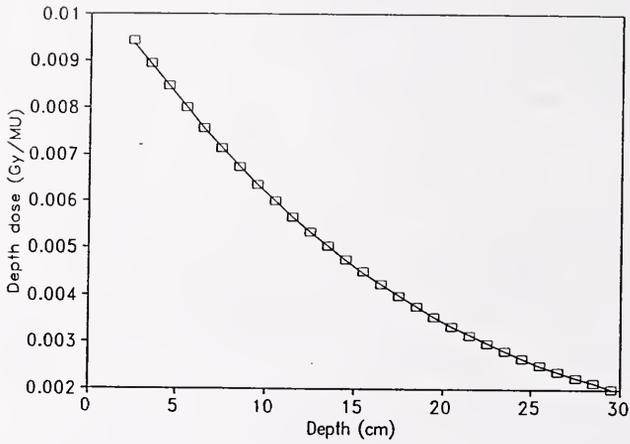
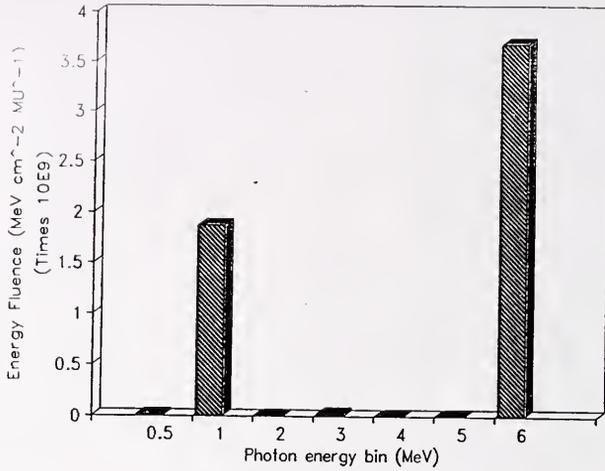


Figure 4-8. Depth dose data (below) resulting from using a more realistic Bremsstrahlung spectrum (above).



□ Best Fit — Measured

Figure 4-9. Depth dose data (below) resulting from using a very unrealistic Bremsstrahlung spectrum (above).

the total (summed over all energy bins) energy fluence providing a link to the calibrated linac output.

Having characterized a polyenergetic open field beam, the next task is to model the effect of partial attenuation (intensity modulation) on the dose distribution. This topic, and how it is incorporated into inverse radiotherapy planning for a single, static beam is discussed in chapter six. However, prior to this, chapter 5 will compare calculated narrow beam polyenergetic dose profiles with measurements.

CHAPTER 5
DOSIMETRY MEASUREMENTS

Overview

This chapter presents absorbed dose profiles resulting from narrow photon beams and the experimental methods used to achieve them. Such measurements are necessary to evaluate the accuracy of dosimetric calculations discussed in previous chapters. Of particular interest is the absorbed dose profile resulting from the extreme case of a 6 MV photon beam collimated by a $3 \times 3 \text{ mm}^2$ opening in a 10 cm thick lead attenuator. As will be discussed in chapter 6, beam intensity modulation is modeled as variable attenuation in adjacent variable thickness lead voxels (0 - 10 cm thick). The smallest feasible lead voxel dimension was chosen to be $3 \times 3 \text{ mm}^2$ as a compromise between resolution and realistic engineering constraints. Thus, it seems appropriate to compare measurements of this "building block" of full sized beams with theoretical calculations.

The convolution dosimetry method is ideal for producing dose distributions in situations where longitudinal and/or lateral equilibrium does not exist. However, measuring absorbed dose in such a situation is quite difficult. Requirements for such dosimetry measurements include high resolution and a flat energy response. Unfortunately, the

most common methods of clinical measuring devices (i.e. ion chambers, semiconductor diodes, and silver halide film) do not meet one or both of these requirements. The ion chamber has an ideal energy response but is typically not small enough relative to the conditions of the narrow photon beam. The silver halide film provides excellent resolution but over responds to the low energy components of the beam. The diode suffers from being both too large and having a non-linear energy response. Alternate methods of measurement include radiochromic film and high resolution TLD sheets. As will be shown, both of these materials better approximate the resolution and energy response criteria.

Radiochromic film

Radiochromic film dosimetry is based on an ionizing radiation induced polymerization process resulting in a color change without the need of post exposure processing (Muench et al. 1991; McLaughlin et al. 1991). The most widely available form of radiochromic film, GafChromicTM Dosimetry Media (Nuclear Associates), consists of a 7 micron sensitive layer on a 4 mil polyester base. The film is colorless and practically "grainless" with spatial resolution of > 1200 lines/mm (McLaughlin et al. 1991). Exposure to radiation at wavelengths below 300 nm result in an absorbed dose dependent change of color from clear to blue. The increase in absorbance or optical density is determined using a spectrophotometer or appropriate densitometer.

Radiochromic film has been used for a variety of measurements including National Institute of Standards and Technology source profiling (Walker et al. 1992), calibration of ophthalmic applicators (Soares 1991; Sayeg & Gregory 1991), brachytherapy dosimetry (Coursey et al. 1992; Muench et al. 1991), and a variety of electron and photon beam investigations (Galvin, Smith & Lally 1993; McLaughlin et al. 1991). The results of these studies have shown that the radiochromic materials have a response similar to that of tissue over a wide energy range (0.1 - 20 MeV for photons and 0.01 - 20 MeV for electrons) with reproducible response characteristics of approximately $\pm 5\%$ at a dose of 200 Gy (McLaughlin et al. 1991). It has also been shown that the material is dose rate independent and has nearly identical response (Δ absorbance per unit absorbed dose $\cong 5\%$) for electrons and photons. Thus, this dosimetry material has the resolution and energy response characteristics needed for the narrow beam dosimetry application at hand.

There are, however, a few difficulties associated with this type of material. Other than the relatively high cost of the material, one must be careful to avoid temperature variations between exposure and reading of more than a few degrees Celsius. However, this is typically not a problem in a controlled climate of a cancer clinic. If significant temperature differences are noted, simple correction factors can be applied (McLaughlin et al. 1991). Also, it has been suggested that one wait at least 24 hours between exposure

and reading in order to allow post irradiation density growth to plateau (McLaughlin et al. 1991; Muench et al. 1991). Data collected for my research support this recommendation and indicate a 10% increase in optical density during the first 12 hours with no further detectable change noted at a 36 hour reading. A third matter of concern is the low sensitivity of the material. For example, a change in optical density of 0.5 requires an absorbed dose of several thousand cGy (see figure 5-1). Finally, and most important of all, one must have access to a suitable spectrophotometer or densitometer. As shown in figure 5-2, the absorbance of the exposed radiochromic film depends on the wavelength used during reading. The series of spectra shown in figure 5-2 were made using a Beckman DU-64 spectrophotometer and agree well with published results (McLaughlin et al. 1991). Thus, due to the spectral sensitivity, one must be sure to evaluate all exposed films (including calibration shots) under the same spectral conditions. A variety of wavelengths (broadband, 400, 510, 540, 580, 600, 605, 633, 650, and 675 nm) have been evaluated (McLaughlin et al. 1991; Muench et al. 1991) although no specific recommendations are given as to the ideal wavelength to use. It appears most investigators are simply using whatever wavelength(s) are available to them. If one is interested in dose profiles, the most likely choice would be either a scanning laser (633 nm) densitometer or a solid state camera film digitization

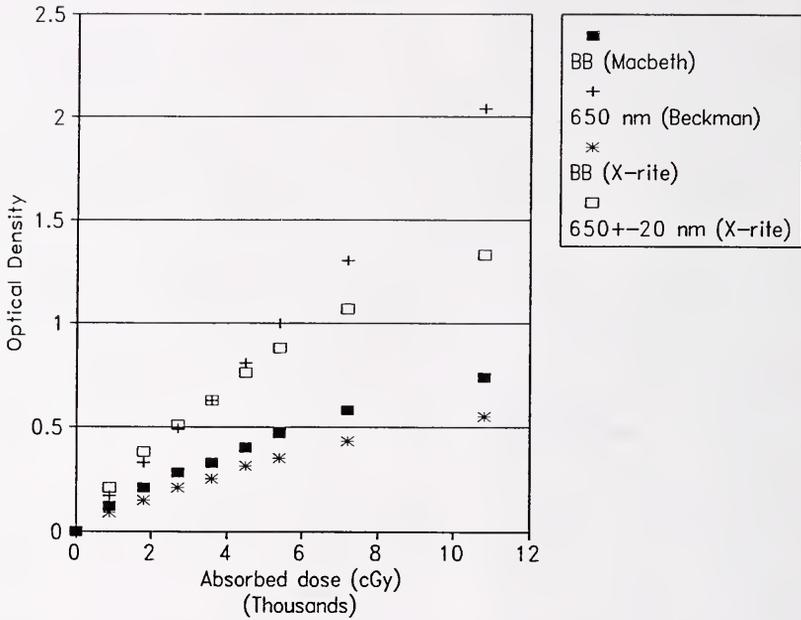


Figure 5-1. The relationship between absorbed dose and optical density for a variety of film reading devices. BB indicates broadband as defined by the white light source of the specific densitometer.

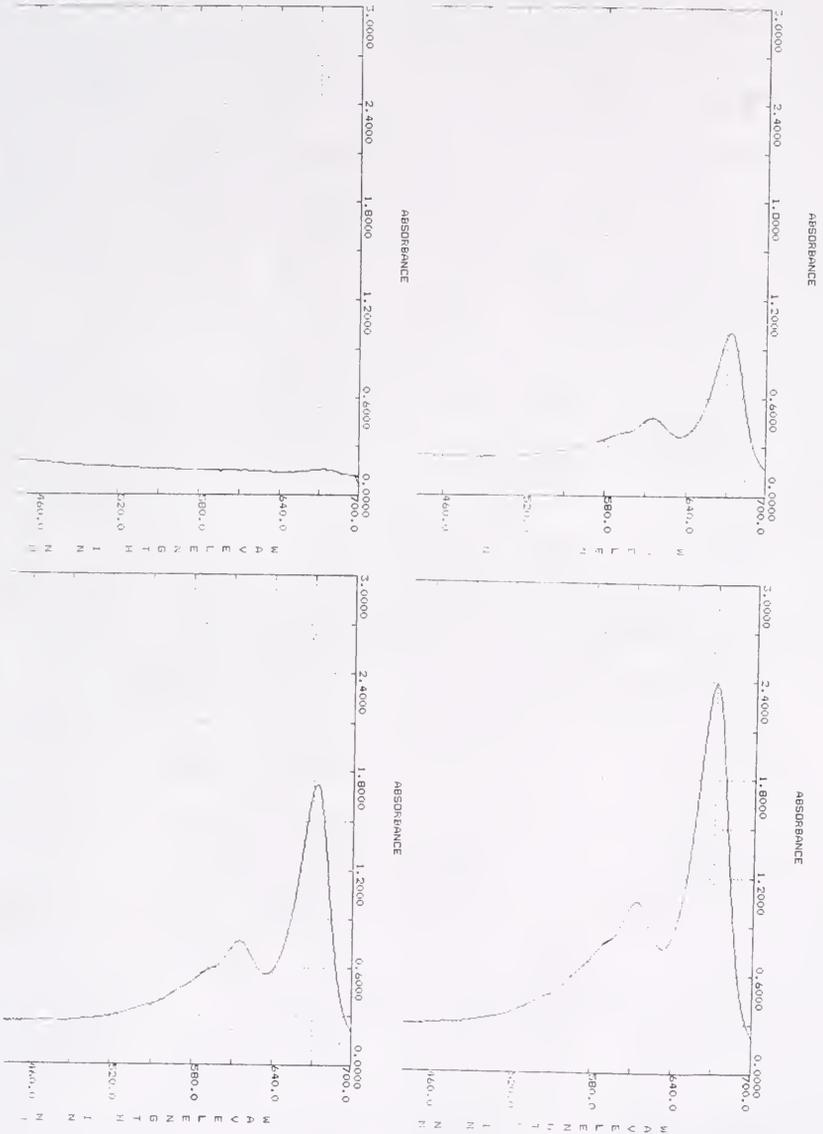


Figure 5-2. Series of absorption spectra obtained from GafChromic™ film (model 041) with a Beckman spectrophotometer. The absorbed dose delivered to each film, in order of increasing magnitude, are: 0, 1800, 3600, and 5400 cGy.

system. The later, of course, may allow some wavelength variability if narrow band optical filters are used. The dose profile capable densitometers currently available in the Shands Cancer Center include a Wellhofer WD-102 scanning diode system and a Charge Injection Device (CID) video system. Unfortunately, the exposed radiochromic film is transparent to the IR diode light source and detector system (950 ± 20 nm) of the Wellhofer system. Thus, all narrow beam dose profile data was collected using the CID video system which has a broadband light source. This system, with a 512×480 CID array, has a resolution based on the field of view (FOV) used during image capture. For the FOV used (10 cm), a resolution of approximately 0.21 mm was achieved.

Since the CID video system was previously shown to work well with the standard silver halide film (Dubois 1993), a comparison was first made between Kodak XV-2 (silver halide) and GafChromicTM for the case of 6 MeV electrons. Although there is a significant difference in the effective Z between the water equivalent radiochromic film and silver halide film, there should be little difference in response for 6 MeV electrons since their ratio of collision stopping power varies slowly with electron energy (Khan 1984). The results of each film type are shown in figure 5-3. Since most of the published measurements with radiochromic film were achieved with monochromatic light sources, these measurements were repeated using a 650 ± 20 nm interference filter in conjunction with the CID camera. The 650 nm filter was

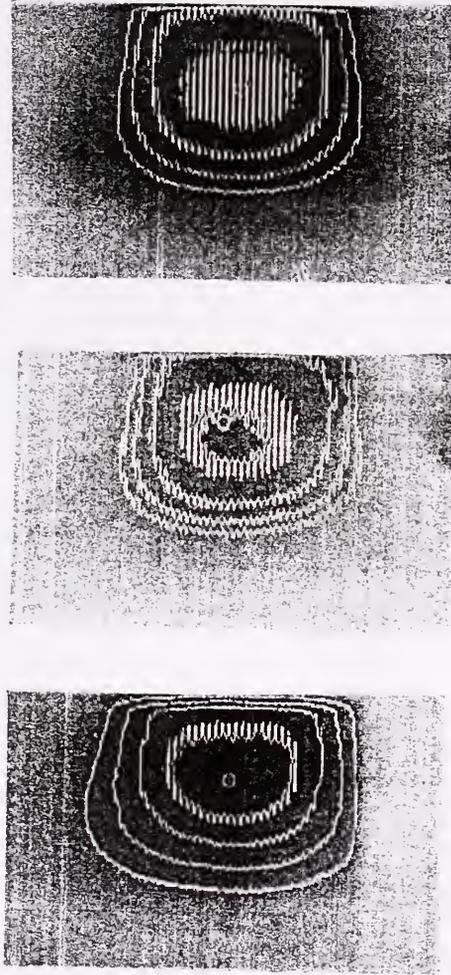


Figure 5-3. Comparison of 6 MeV e^- dose profiles from Kodak XV-2 film (top) and GafChromic™ film (model 041) using a broadband CID densitometer system (middle: no filter, bottom: 650 nm filter).

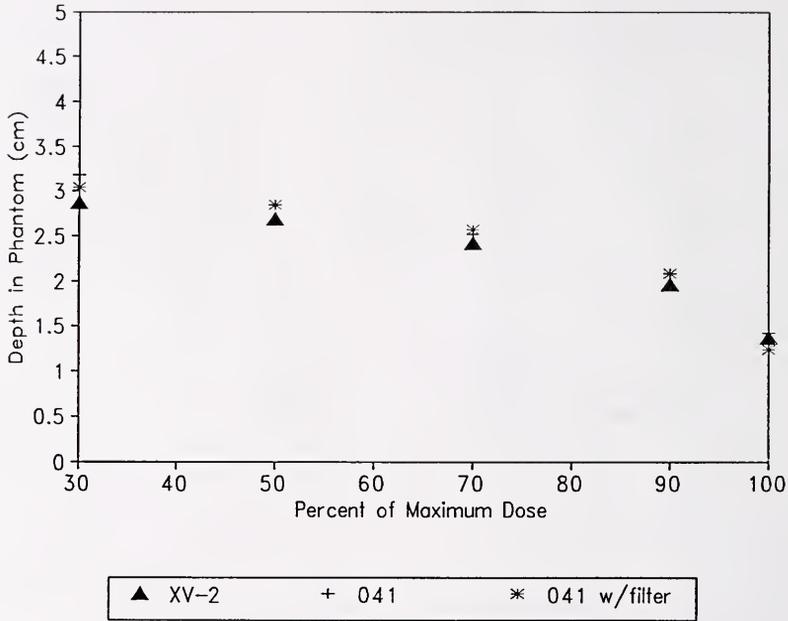


Figure 5-4. Comparison of 6 MeV e^- depth dose data (10×10 cm^2 cone, 3×3 cm^2 blocked field, and 100 cm SSD) obtained using broadband and filtered (650 ± 20 nm) light sources.

chosen since it provided the largest signal amplification as reported in the literature (McLaughlin et al. 1991) and since the CID camera has a peaked response at 650 nm. As shown in figure 5-4, no significant difference in response was observed between the broadband and filtered electron depth dose data. However, the filtered image was significantly blurred compared to the unfiltered image. Thus, the decision was made to proceed with narrow beam measurements using the broadband light source.

The 6 MV absorbed dose profiles for the 3x3 mm² collimator opening is shown in figure 5-5. The experimental data was obtained by carefully aligning the radiochromic film, placed inside of a solid water film phantom (Bova 1990b), parallel to the incident collimated photon beam. The parameters of interest include a collimator setting of 5x5 cm², source to surface distance of 100 cm, center of attenuator to surface distance of 27 cm, and an opening of 3x3 mm² within a 12.7 cm thick (10 cm lead equivalent) cerrobend attenuator. A total of 4500 monitor units were delivered resulting in a maximum absorbed dose reading of 2078 ± 81 cGy at a depth of 0.83 ± 0.05 cm compared to a calculated maximum value of 2246 cGy at a depth of 0.85 cm. These absorbed dose values differ by approximately 7.5% which is reasonable considering the experimental conditions (geometrical alignment) and degree of nonequilibrium present. Also, the data appear consistent with Monte Carlo

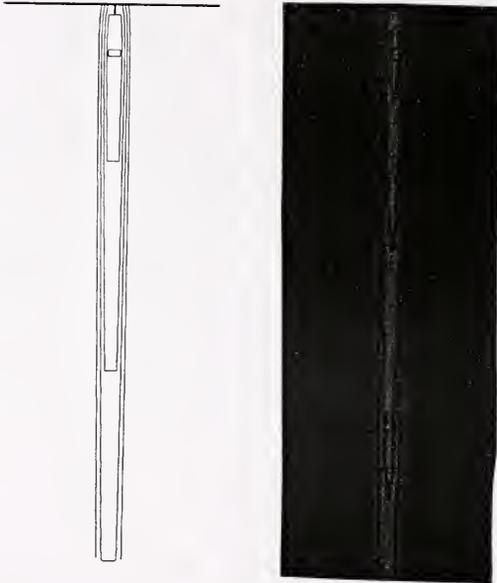


Figure 5-5. Comparison of calculated (left) and measured (right) 6 MV narrow beam ($3 \times 3 \text{ mm}^2$ blocked field) photon absorbed dose profiles. Measured data was obtained from GafChromicTM film using the CID densitometer system. The 90,70,50 and 30% isodose lines are shown.

predictions for narrow photon beams (Bjarngard, Tsai & Rice 1990).

TLD sheets

TLDs have been in use for quite some time for obtaining accurate integrated dosimetry data. The small size of available TLD chips or rods make them ideal for obtaining "point" measurements comprising a 2 or 3 dimension dose distribution. However, as each TLD requires its own calibration and is read individually (see figure 5-6), a tremendous amount of work is needed for large arrays. Also, the size of each TLD chip or rod prohibits high resolution measurements.

Recently, an alternative approach to TL dosimetry has become available (Jones 1993; Jones et al. 1992) whereby a nearly continuous layer of TLD material is exposed to ionizing radiation and read via a precision controlled CO₂ laser. The TLD material is currently available in two different forms. For standard sized field dosimetry, 1.5 mm diameter, 38 µm thick MgB₄O₇:Tm elements are arranged in a 3 mm x 3 mm grid on a 30 cm x 30 cm sized sheet totaling 10201 elements. For small field, high resolution work, a continuous 4.65 cm x 4.65 cm sheet of CaSO₄:Mn is read at 0.375 mm intervals for a total of 15,625 readings. These materials, while offering good resolution and measurement convenience, do have drawbacks. For example, the CaSO₄:Mn material used on the high resolution sheets exhibits significant fading (50 - 60% in the first 24 hours) and some

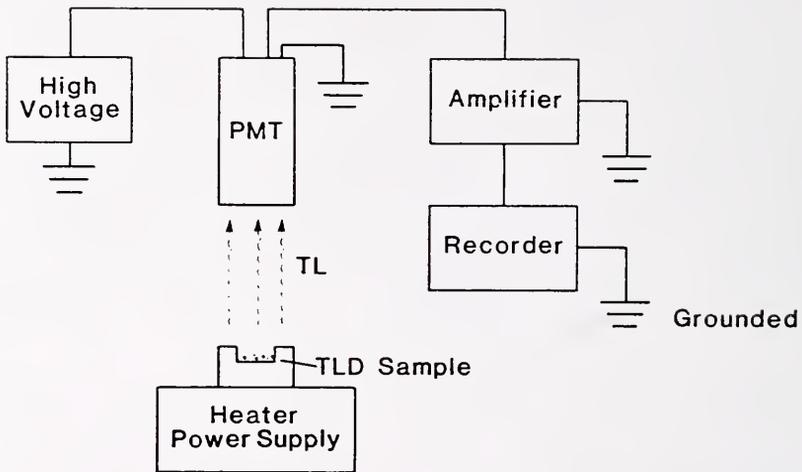


Figure 5-6. Schematic diagram showing typical apparatus for measuring thermoluminescence (Khan 1984).

sensitivity to ambient light (Khan 1984). Thus, the data sheets must be kept in the dark as much as possible and a standard exposure to reading delay should be chosen. Also, while the effective Z of $\text{CaSO}_4:\text{Mn}$ is only 15.3, an enhanced sensitivity to lower energy photons does exist. For example, the material is approximately 10 times more sensitive at 30 keV than at 1.25 MeV (Khan 1984). In contrast, however, silver halide film is approximately 25 times more sensitive at 25 keV than at 1.25 MeV (Johns & Cunningham 1983). The substrate for both types of material is a 0.125 mm thick polymer.

In the arrangement shown in figure 5-7, the CO_2 laser rapidly heats the TLD spot producing the characteristic glow curve which is detected by the photomultiplier tube. If careful calibration and annealing schedules are followed, accuracy to within 1% of the delivered dose is claimed (Jones 1993, coarse grid arrays). Unfortunately, no published data on the accuracy of the high resolution sheets is currently available. These sheets are still in the experimental phase of development for which the Shands Cancer Center is a test site. Ongoing tests have shown an accuracy of approximately 7%.

Prior to collecting narrow beam data for this research, a high resolution TLD sheet calibration was accomplished with 400 cGy from a Co^{60} unit. After the calibration reading, the sheet was annealed for 30 minutes at 170°C . As with the radiochromic film, the TLD sheet was placed

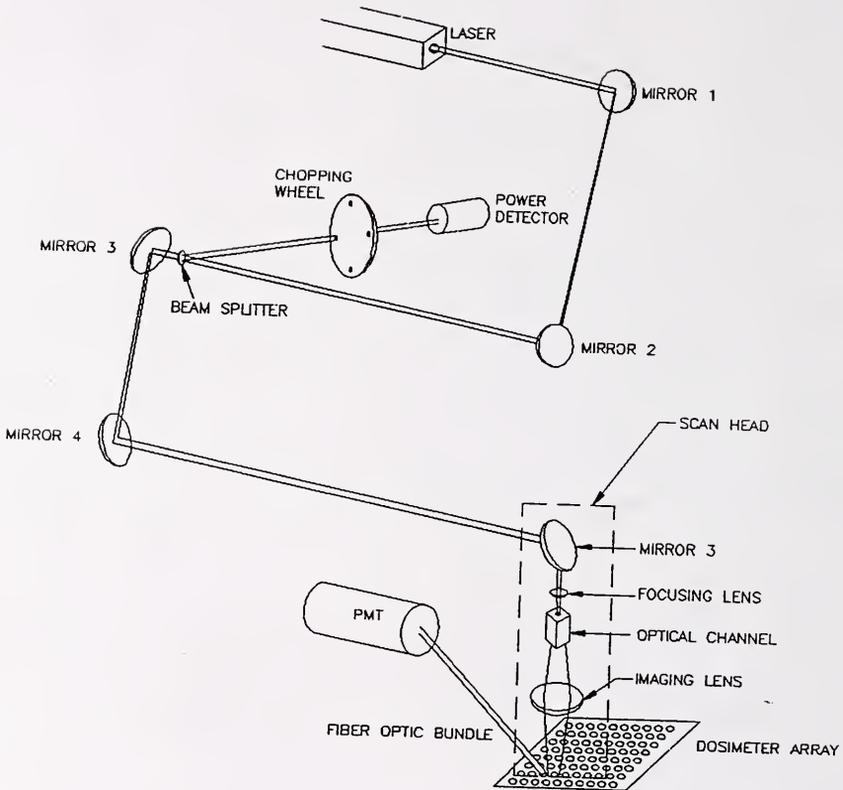


Figure 5-7. Schematic diagram of the laser/optical system of the TLD array reader. The 3 mm x 3 mm grid array type of sheet is shown (Jones 1993).

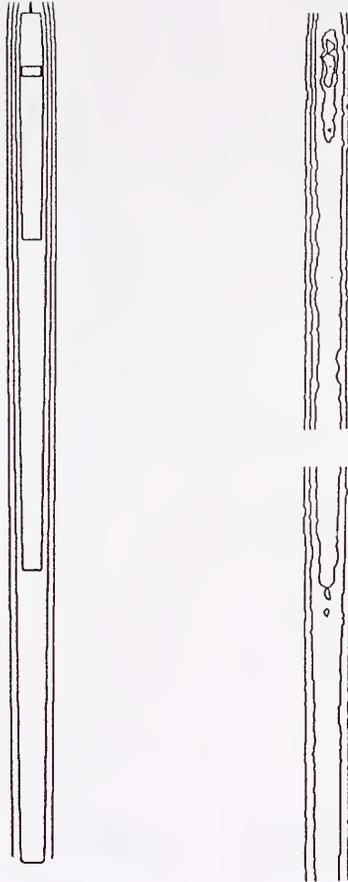


Figure 5-8. Comparison of calculated (left) and measured (right) 6 MV narrow beam ($3 \times 3 \text{ mm}^2$ blocked field) photon absorbed dose profiles. Measured data were obtained from two $\text{CaSO}_4:\text{Mn}$ high resolution TLD sheets placed adjacent to each other. The 90, 70, 50, and 30% isodose curves are shown for calculated results on the left. The same isodose percentages, as well as the 95% line, are shown for measured data on the right.

parallel to the incident 6 MV beam using the film phantom and same geometry as before but with a monitor unit value of 900. The TLD sheet was then read with the device shown in figure 5-7 using the previously stored calibration data to obtain absolute data. Experiments have shown that several readings can be accomplished without recalibrating the TLD sheets. However, for this research, the data were the first obtained following the calibration. A comparison of the TLD reading and calculated results are shown in figure 5-8. These results, as with the radiochromic film data, compare favorably with the calculated data with a measured maximum dose of 503 ± 35 cGy at a depth of 0.80 ± 0.05 compared to calculated values of 449 cGy at 0.85 cm.

The position of the calculated d_{\max} matches the measured values from both experimental methods. The calculated absorbed dose value appears to be 8% larger than that measured with radiochromic film and 11% smaller than that shown with the high resolution TLD sheets. Considering the experimental conditions and dosimetry materials involved, the calculated data appear to be sound and quite suitable for the current application.

CHAPTER 6 INTENSITY MODULATION

Overview

This chapter presents the method used to determine the optimal 2-D beam intensity modulation function (IMF) needed for any given beam's eye view of the target tissue. The optimal choice is chosen in such a way that each beam's eye view IMF is independent of any others. Determining the IMF is a two step process involving deconvolution followed by reverse ray tracing. Polyenergetic absorbed dose distributions resulting from the projection of photon beams through intensity modulation devices will also be presented.

Optimal Intensity Modulation

Deconvolution

As you recall, the absorbed dose can be calculated by convolving an energy deposition kernel, referred to as a differential pencil beam (DPB), with a TERMA distribution resulting from the primary photon fluence. In a similar manner, the TERMA can be found if an absorbed dose distribution and DPB are provided. The convolution theorem, which proved useful in performing the convolution, is equally useful for the deconvolution process. The TERMA can be found from

$$T(\bar{r}') = \mathfrak{F}^{-1} \left(\frac{\mathfrak{F}[D(\bar{r})]}{\mathfrak{F}[k(\bar{r}-\bar{r}')] } \right)$$

where

\mathfrak{F} = Fourier transform.

\mathfrak{F}^{-1} = Inverse Fourier transform.

$T(\bar{r}')$ = TERMA value associated with a tissue voxel centered at \bar{r}' .

$D(\bar{r})$ = prescribed dose distribution value associated with a tissue voxel centered at \bar{r} .

$k(\bar{r}-\bar{r}')$ = DPB describing deposition of energy at \bar{r} due to an initial photon, representing the average energy of the beam, interacting at \bar{r}' .

Thus, instead of multiplying two values in frequency space, a ratio is found. The inverse FFT of this ratio represents the TERMA distribution required to achieve the desired absorbed dose prescription. This prescription contains a value of absorbed dose, in Gy, for each cubical voxel of tissue in 3-D space. It is an ideal request that contains all zeros outside of the target volume and a single, uniform value inside the target volume. Critical structures may also be included in the dose prescription by assigning an arbitrary numerical value (chosen to be equal to 1) outside the dose range of interest to the voxel(s) comprising the region of interest. Voxels that have been "tagged" as critical structures are treated the same as zero dose voxels during the deconvolution process but are spared

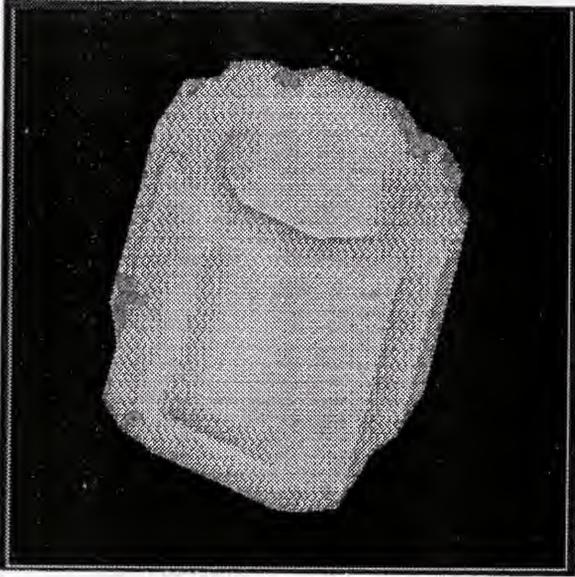


Figure 6-1. Isodose surface rendering of bowl shaped dose prescription containing a uniform value of 10 Gy.

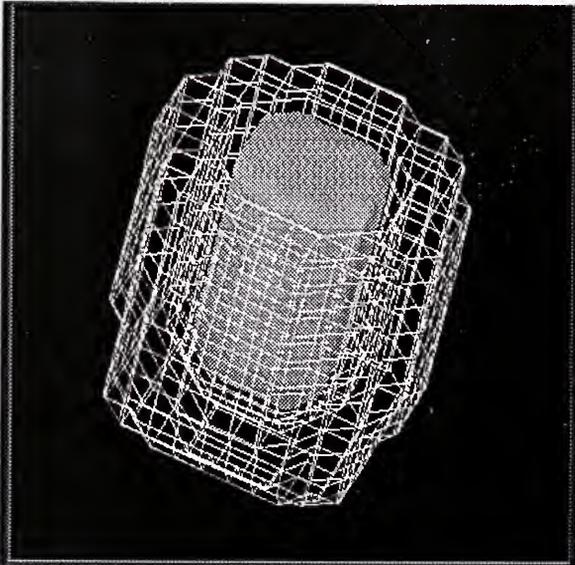


Figure 6-2. Isodose wire surface rendering of a bowl shaped dose prescription containing a uniform value of 10 Gy surrounding a region defined as a critical structure

dose during the backprojection process to be discussed in the next section. Examples of bowl shaped dose prescriptions with and without a critical structure are shown in figures 6-1 and 6-2.

Obviously, the TERMA distribution represents an ideal situation that can not strictly be achieved with a single photon beam or even multiple beams. The physics of photon interactions simply do not allow all values of the deconvolved ideal TERMA, especially those that are negative. However, the information resulting from the deconvolution does allow computation of the "best approximation" to the absorbed dose prescription for a given photon beam. Examples of the TERMA resulting from deconvolution using a 2 MeV DPB, a bowl shaped dose prescription, and $2 \times 2 \times 2 \text{ mm}^3$ voxels is shown in figures 6-3 and 6-4. The bowl shaped dose prescription shown in figure 6-1 was chosen to ultimately illustrate the ability of the full field intensity modulation technique to handle concave features. The values for the isoterma surface renderings in figures 6-3 and 6-4 were chosen to illustrate the difference between the TERMA and the dose prescription. This difference is due, of course, to the fact that energy released in one voxel is actually deposited in many voxels. Thus, while the entire bowl shaped absorbed dose prescription is assigned 10 Gy (10 J/kg deposited), a wide range of energy released values arranged in a nonuniform manner are required to achieve this prescription. Figure 6-3 shows only those voxels



Figure 6-3. Isoterma surface displaying 10 J/kg values.



Figure 6-4 Isoterma surface displaying 33 J/kg values.

containing 10 J/kg (released) and 6-4 shows only those voxels containing 33 J/kg (released). Figure 6-5 provides more detail by presenting a contour plot of the TERMA near the central axis.

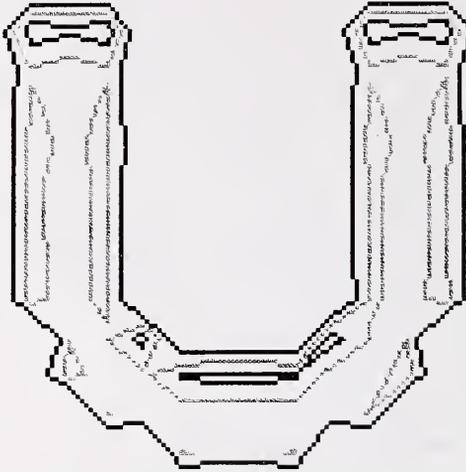


Figure 6-5. Isotherma contour plot parallel to central axis of beam. The curves shown represent 10, 20, and 30 J/kg respectively.

Backprojection

Once the TERMA is found, it can be used to calculate the physical characteristics of the intensity modulating device that the photon beam will pass through. Differential spatial attenuation will then result in an absorbed dose distribution that, within the physical limits of a single beam, will conform to that in the prescription.

For this work, lead was chosen as the attenuating material. The attenuating device was modeled as a uniform

40x40 voxel grid with each voxel having a variable dimension square face and variable thickness ranging from 0 to 10 cm. As discussed in chapter 5, a minimum value of 3x3 mm² for the voxel face was chosen as a convenient compromise between high resolution and construction limitations and is totally independent of the absorbed dose calculation voxel size in the phantom. The distance from the linac target to the center of the attenuator can be chosen to reflect any desired geometry. For the TERMA assigned to each cubical voxel within the dose calculation volume of the phantom, an appropriate lead thickness value in the lead grid can be determined by backprojection. The backprojection referred to is a simple ray trace from the center of a TERMA voxel towards the linac target as illustrated in figure 6-6. Included in the ray trace are inverse square, water phantom (W) attenuation, and lead (L) attenuation:

$$t_{xy} = \left(\ln \left[\frac{\text{TERMA}}{\Psi_r \left(\frac{s}{r} \right)^2 \left(\frac{\mu}{\rho} \right)_w 1.6E-10 e^{\left\{ - \left(\frac{\mu}{\rho} \right)_w 1.0(r-s) \right\}}} \right] \right) / \left(\frac{\mu}{\rho} \right)_L 11.3$$

where

t_{xy} = thickness of lead voxel, in cm, at grid position x, y.

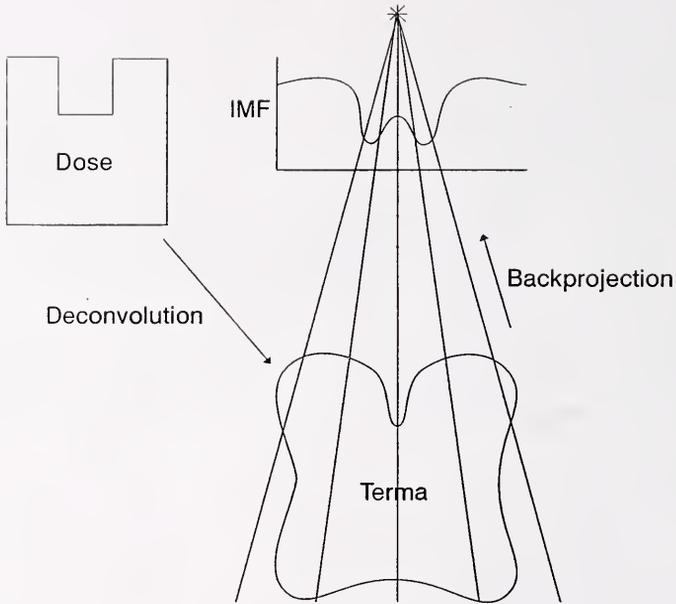


Figure 6-6. Illustration of two step process to determine IMF from absorbed dose prescription.

Ψ_i = reference incident energy fluence for monoenergetic photons at a specified distance from the source.

All other variables in the above equation were previously defined.

In order to exclude physically unrealistic solutions, the thickness calculation is only performed for those cubical voxels with TERMA values above a preset minimum. The minimum value was chosen to be that resulting from transmission through the thickest possible (10 cm) lead attenuator. Thus, for each cubical voxel containing a physically realistic TERMA value, a lead thickness value is computed. For voxels tagged as critical structures, a lead thickness value is assigned based on a weighting factor (ranging from 0 to 1.0) relative to that of target tissue. A default value of 1.0 results in a maximum lead thickness (10 cm) being assigned. This same tissue weighting factor will also play a role during beam weight optimization to be discussed in the next chapter. Next, the thickness values computed for all TERMA voxels (and those for critical structures) whose ray lines intersect a given lead attenuator voxel are averaged. When completed, the 40x40 voxel attenuator grid contains an average thickness value for each voxel which is representative of the beam's eye view of the TERMA distribution below it. Figure 6-7 illustrates the intensity modulating grid design for the case of a beam's eye view into the top of the bowl shaped absorbed dose prescription. This design was computed using a

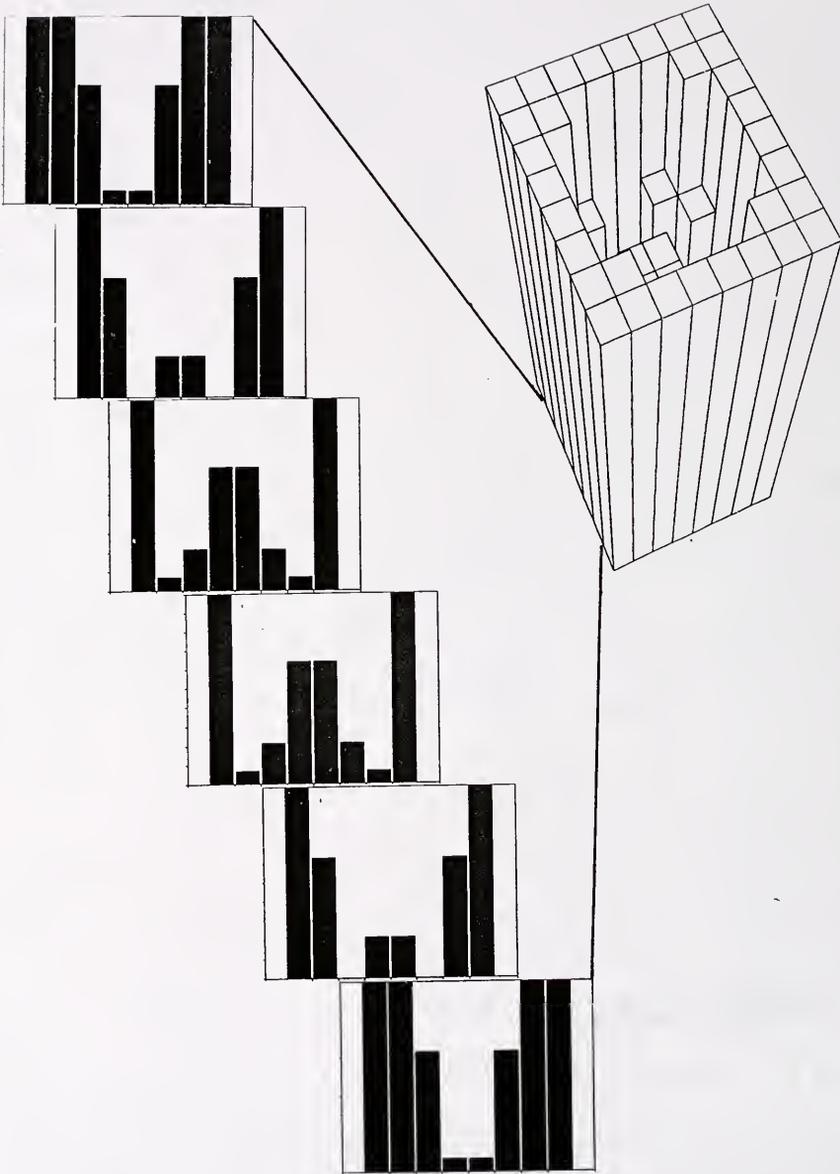


Figure 6-7. Example of beam's eye view intensity modulation device needed for a top view of the bowl shaped dose prescription surrounding a critical structure (cutaway provided for clarity).

single (2 MeV) energy DPB instead of a polyenergetic spectrum thus resulting in reasonable calculation times. The need for such short calculation times will be demonstrated in the next chapter where multiple beam treatment plans will be investigated.

Single Beam Intensity Modulated Dosimetry

A conformal, 3-D absorbed dose distribution can now be computed by modeling the effect of the lead grid on the incident energy fluence. At this point the versatility of the DPB convolution dosimetry algorithm becomes even more apparent. The differential attenuation of each energy component comprising the primary photon beam is accomplished with a simple exponential term for each energy bin. The term actually includes differential attenuation for the phantom as well as the lead. However, the phantom effect is minor compared to the lead. The attenuated incident fluence is then used to compute the TERMA which is convolved with the DPB(s) resulting in absorbed dose. Figure 6-8 illustrates relative isodose contour plots of the absorbed dose distribution using a polyenergetic beam and the intensity modulation function illustrated in figure 6-7. The dose sparing effect of averaging maximum lead thickness values for the critical structure (critical tissue weight = 1.0) regions is quite noticeable. If no critical structure is present, the resulting intensity modulating grid design projects the isodose contours shown in figure 6-9. In the next chapter, the use of such modeling will be extended to

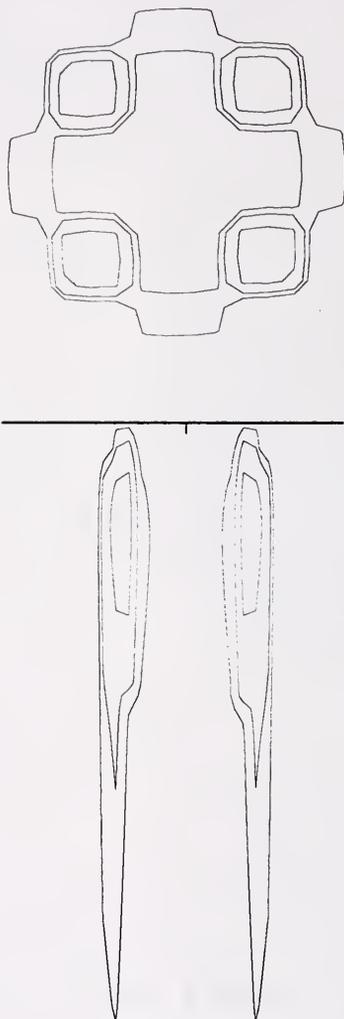


Figure 6-8. Relative isodose contour plots of absorbed dose resulting from the polyenergetic modeling of 6 MV photons passing through the optimal lead grid for a top view of the bowl shaped prescription with a critical structure. FS = 5x5 cm², SSD = 100 cm. Isodose lines shown are 70, 50, and 40% (above), and 60, 50, and 40% (below).

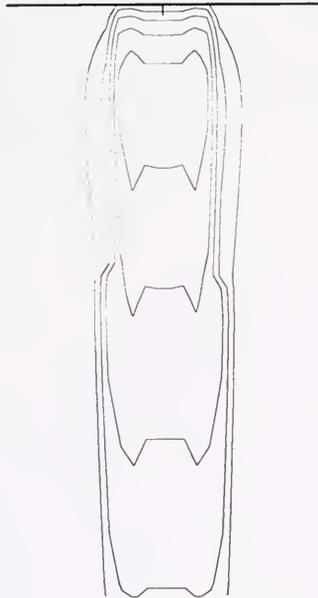
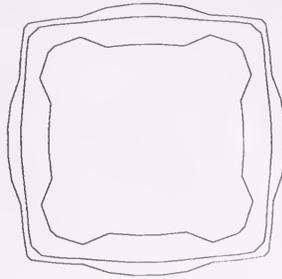


Figure 6-9. Relative isodose contour plots of absorbed dose resulting from the polyenergetic modeling of 6 MV photons passing through the optimal lead grid for a top view of the bowl shaped prescription without a critical structure. FS = $5 \times 5 \text{ cm}^2$, SSD = 100 cm. Isodose lines shown are 70, 50, and 40% (above), and 90, 80, 70, 60, and 50% (below).

multiple beam inverse treatment plans where the number of monitor units assigned to each beam will be determined through numerical optimization.

CHAPTER 7 RADIOSURGERY PLANNING

Overview

This chapter presents the steps taken to incorporate the previously discussed single beam conformal dosimetry into a multiple beam planning tool appropriate for radiosurgery. As mentioned previously, one of the primary goals of this research is to determine how well a limited number of highly modulated BEV fields can reproduce a desired dose distribution. Obviously, fewer fields result in a quicker and less error prone treatment. Producing such a plan, requires solving for the required intensity modulation and resulting absorbed dose distribution for any beam's eye view (BEV) of the target lesion. Thus, the first part of this chapter will concentrate on the method of coordinate transformation used to simulate table and gantry rotation.

Having arrived at the ideal intensity modulation for any BEV, the next task is to determine the optimum beam weighting assigned to each BEV field. This is a multi-dimensional optimization problem with the number of dimensions equal to the number of BEVs. The last half of this chapter will present an approach to this problem using the combined simulated annealing/downhill simplex optimization technique previously discussed.

Coordinate Transformation

As mentioned in chapter one, linac based stereotactic radiosurgery involves rotations of both the table and gantry. At the Shands Cancer Center, a typical single isocenter plan includes a continuous 100° gantry arc for each of five to nine table angles at 20° increments. All rotations are relative to the isocenter placed in the center of the target lesion.

Simulating these conditions within the PLAN algorithm can be accomplished by first choosing a point within the 3-D absorbed dose prescription array as the isocenter. For this research, the point does not necessarily have to be at the exact center of the lesion but should lie near the center of the target region to ensure the target does not rotate outside of the calculation volume boundaries. For example, in the case of a target region consisting of a lesion surrounding a critical structure, the isocenter may actually lie within the critical structure. This is, of course, not possible unless intensity modulation of the beam is available. Having identified the isocenter, the coordinate transformation is accomplished by translating it to the origin of the data set, rotating about the table (z) axis, rotating about the gantry (x) axis, and translating back to the original isocenter location. An example of such a procedure is shown in figure 7-1. If the 3-D volume of data is treated as a homogeneous coordinate system, all of the actions above can be combined into a single 4×4

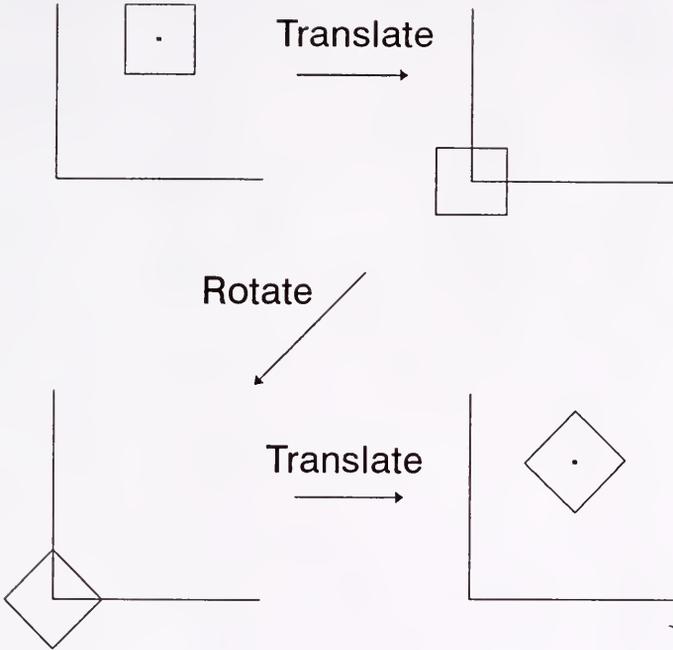


Figure 7-1. Illustration of translation, rotation, and translation in two dimensions.

transformation matrix. In a homogeneous coordinate system, each point (x, y, z) is expressed as (x, y, z, H) where H is a nonzero scale factor typically set equal to 1 (Park C. 1985). Care must be taken however, to insure the correct order of events since rotations about multiple axes are non commutative. The following four matrices were multiplied in proper order to obtain the complete transformation matrix with the table and gantry angles represented by θ and ϕ respectively:

Translate to origin:

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -l & -m & -n & 1 \end{vmatrix}$$

Rotate about z axis:

$$\begin{vmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Rotate about x axis:

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & \sin\phi & 0 \\ 0 & -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Translate to isocenter:

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & m & n & 1 \end{vmatrix}$$

Complete (translate, z rotation, x rotation, translate) transformation:

$$\begin{vmatrix} \cos\theta & \sin\theta\cos\phi & \sin\theta\sin\phi & 0 \\ -\sin\theta & \cos\theta\cos\phi & \cos\theta\sin\phi & 0 \\ 0 & -\sin\phi & \cos\phi & 0 \\ -l\cos\theta+m\sin\theta+1 & -l\sin\theta\cos\phi-m\cos\theta\cos\phi+n\sin\phi+m & -l\sin\theta\sin\phi-m\cos\theta\sin\phi-n\cos\phi+n & 1 \end{vmatrix}$$

The variables l , m , and n refer to the x , y , and z coordinates of the chosen isocenter in the prescription file. The table and gantry angles were chosen such that positive values correspond to positive angle rotations of the actual linac hardware.

After transforming the absorbed dose prescription data file, each appropriate BEV intensity modulation is determined through deconvolution and reverse ray tracing as discussed in the previous chapter. Each intensity modulation data set is then stored on disk and subsequently used to create its corresponding absorbed dose distribution. Each BEV absorbed dose distribution is computed with the appropriate intensity modulation through convolution and

then rotated about the previously determined isocenter location using the opposite transformation matrix (both sign and rotation order reversed) as that for determining the intensity modulation.

Multiple Beam Optimization

A multiple BEV absorbed dose distribution is obtained by simply summing the absorbed dose values, voxel by voxel, of each independently obtained BEV absorbed dose distribution. Each of these BEV distributions represents absorbed dose per monitor unit. Thus, the optimization variables considered are the number of monitor units assigned to each beam. For example, if three 100° arcs are approximated by discrete BEV intensity modulated fields at 20° intervals, a total of 18 variables would be involved. This number is significantly less than the hundreds or thousands needed if one were to allow the intensity of each finite pencil beam element of every BEV field to be a variable as mentioned in chapter 2.

Since the solution space is multi-dimensional and likely to have many minimum value "traps," the simulated annealing version of the downhill simplex used in chapter five is also used for this case. The objective function, O , was chosen to provide a measure of the root mean square difference between the desired dose distribution and the one obtained using deconvolution/optimization and is as follows:

$$O = \sqrt{\frac{\sum_{i=1}^N [(D_{Rx_i} - D_{calc_i}) w_i]^2}{\left(\sum_{i=1}^N w_i\right)^2}}$$

The variable i represents each voxel within the dose calculation volume consisting of N total voxels, D_{Rx} represents the absorbed dose prescription, and D_{calc} represents the total calculated absorbed dose:

$$D_{calc} = \sum_j (d_{calc_j} \times MU_j)$$

The variable j represents each BEV, d_{calc} represents the calculated dose per monitor unit value in a single voxel for a single BEV field, and MU represents the number of monitor units assigned to a given BEV field. The variable w_i is a weighting factor assigned to each type of voxel (normal, critical, or target) within the calculation volume. The values for the weighting factors can range from 0.0 to 1.0 with target tissue receiving a default value of 1.0. For critical structures, a value of 1.0 maximizes sparing of critical tissue while values less than 1.0 relax this condition. Likewise, for normal tissue, a large (close to 1.0) weighting factor of maximizes normal tissue sparing. This is accomplished by encouraging a larger number of incident beam directions from those available. Choosing a value equal to 1.0 forces equal beam weights while choosing a value of 0.0 permits some incident beam directions to be

removed from the final plan if deemed necessary. Of course the choice of weighting factors depend on the desired aggressiveness of treatment and anatomical conditions. For example, a target surrounding an island of normal tissue might be treated more aggressively than a target surrounding a critical structure. Of course, the terms "normal" and "critical" are relative when considering irradiation of the brain. finally, the other constraints imposed during optimization are that monitor unit values be greater than or equal to zero and that the maximum target dose be greater than or equal to the prescription dose. The first of these constraints is obvious since negative monitor unit values are physically impossible. The second of these constraints is required when critical structures near the target tissue are heavily weighted. Without such a constraint, the critical structure dose would be lowered at the expense of underdosing the target tissue.

Annealing Parameters

Prior to presenting and discussing the results of several example cases, a few words regarding the sensitivity of output to the optimization annealing schedule is required. The purpose of examining the annealing schedule is to identify the parameters that increase the likelihood of finding the global minimum while reducing unnecessary computational effort. Unfortunately, no exact method exists for determining these parameters. Thus, one must rely on experimentation with the type of objective function in

question. The annealing schedule was investigated by varying both the initial starting temperature, TEMPTR, and the number of iterations, IITER, accomplished prior to reducing the temperature by a predetermined reduction factor. Although a variety of temperature reduction factor schemes are possible (Press et al. 1992), a conservative reduction factor of 0.8 was chosen in order to reduce the sensitivity of the algorithm to subtle changes in the objective function due to increasing or decreasing the number of potential BEVs to be optimized. An example of the sensitivity of plan results to these parameters is shown in figure 7-2. Note that choosing too small a value of IITER clearly results in a non-global solution while choosing excessively large TEMPTR values increases the computation time significantly with little improvement in the objective function. Of these two variables, it appears that IITER is the most sensitive. The trend shown in this figure appears to hold for the other test cases considered and suggest that conservative values of 1 and 100 for TEMPTR and IITER be used for future cases. This parameter selection may not always guarantee a global solution, but do result in treatment plans being obtained in a somewhat reasonable time frame.

Since simulated annealing optimization techniques are known to consume significant CPU time, solutions for all example cases were also accomplished with the downhill simplex version of the annealing algorithm (TEMPTR=0). This version of the algorithm is "greedy" in the sense that it

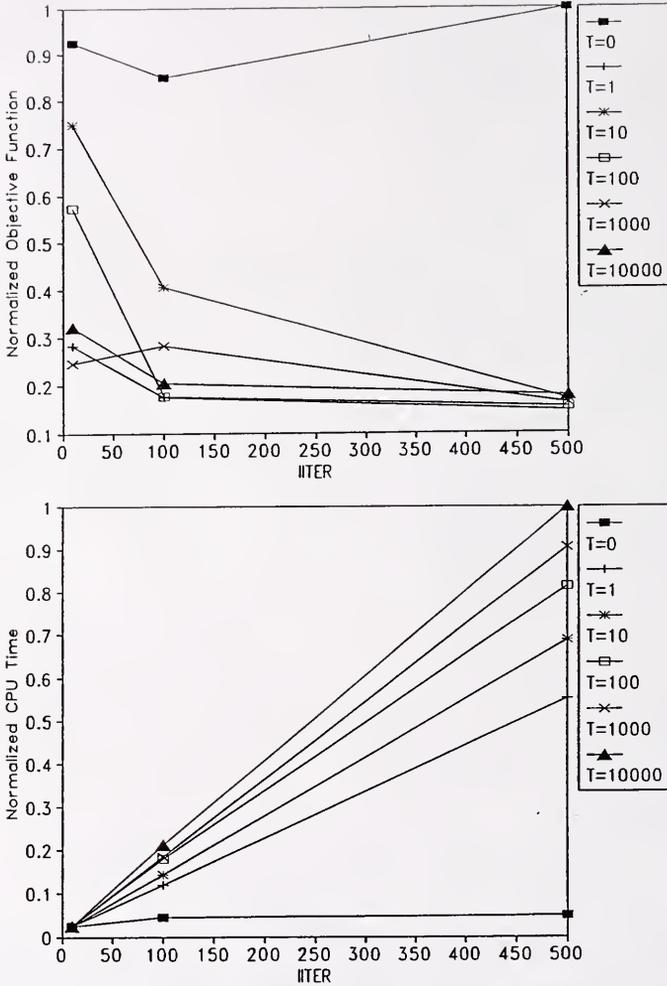


Figure 7-2. Effect of annealing parameters TEMPTR (T) and IITER on the objective function and planning time.

always looks for a downhill step. Thus, it is more likely to get trapped in a local minima. Experience with the example cases to be presented has indicated a solution space with many local minima. Surprisingly, the treatment plans resulting from these various minima do not appear significantly different. As a result, the downhill simplex optimization was terminated and the current minima accepted at approximately 1100 iterations resulting in significant CPU savings. A comparison of the convergence rates for a typical 18 BEV case study is shown in figure 7-3. Extension of CPU time to other cases with more or less BEVs is illustrated in figure 7-4. Obviously, the more variables (beams) to consider, the more advantageous it is, from a CPU standpoint, to use the truncated downhill simplex algorithm instead of the simulated annealing algorithm. The ultimate decision as to which algorithm to use must, of course, be based on the specific clinical goals as well as practical considerations such as planning time.

Example Results

In order to evaluate the effectiveness of this planning tool, multiple example cases were considered. The example case targets, shown in figures 7-5 through 7-12, consist of a single sphere, a cluster of three spherical targets, a rod shaped target, a hoop shaped target with and without a central critical structure, a notched cube shaped target with and without critical structure tissue filling the notch, and an irregular shaped AVM. These test cases, except

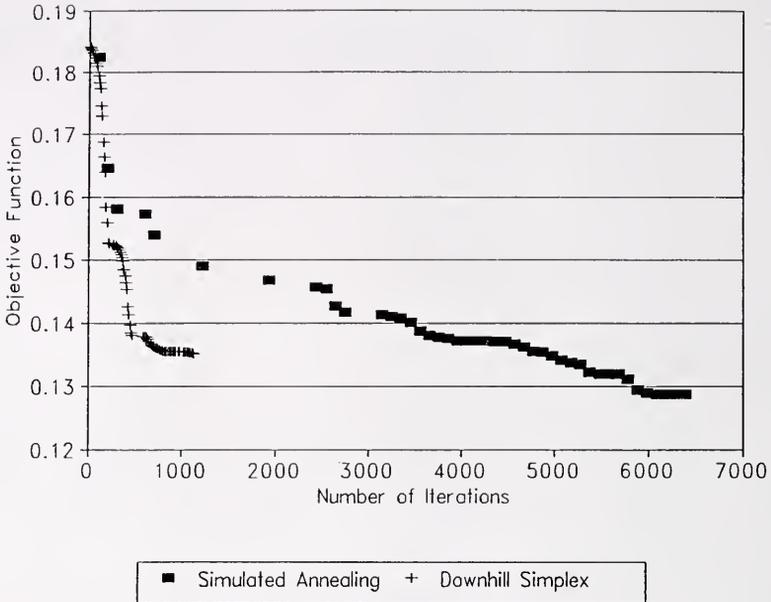


Figure 7-3. Convergence for a typical 18 BEV plan using simulated annealing (TEMPTR=1, IITER=100) and downhill simplex (TEMPTR=0).

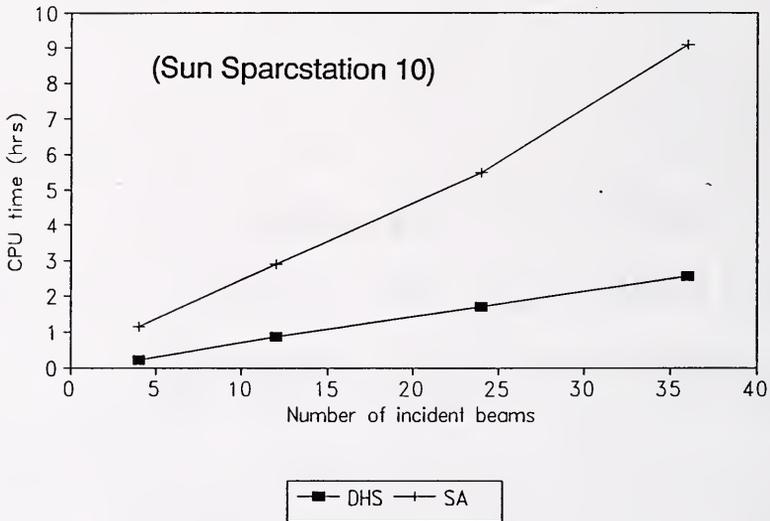


Figure 7-4. The effect of the number of incident beams of CPU time.

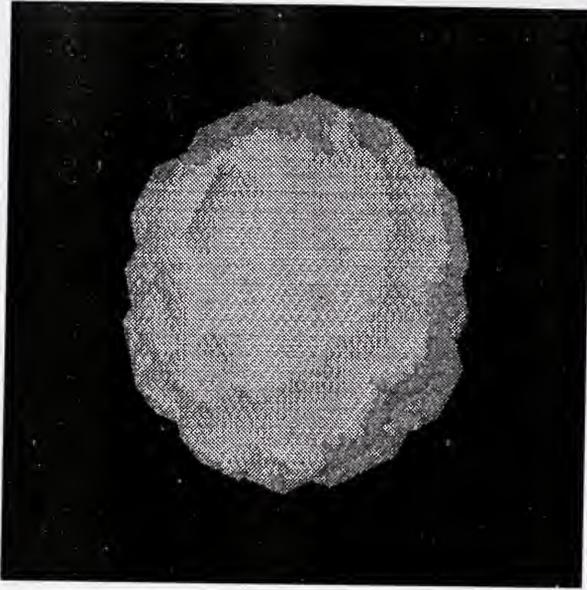


Figure 7-5. Single large sphere target

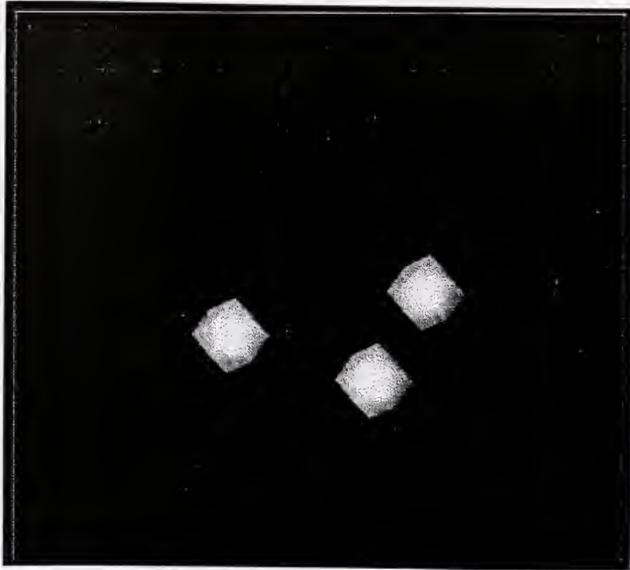


Figure 7-6. Multiple small sphere target

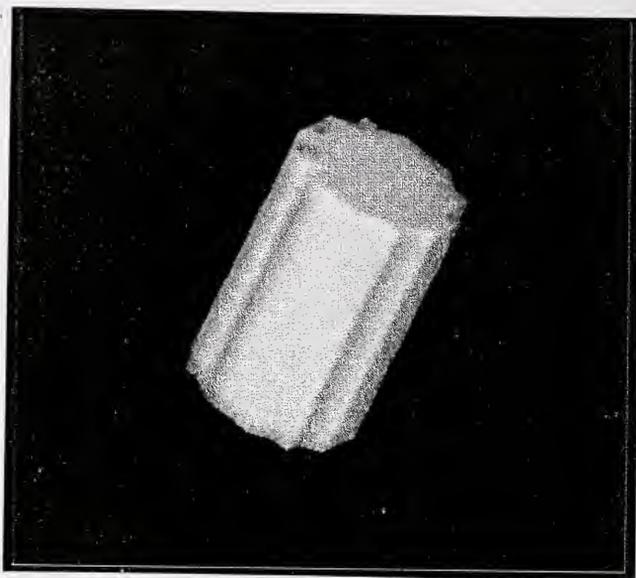


Figure 7-7. Rod shaped target

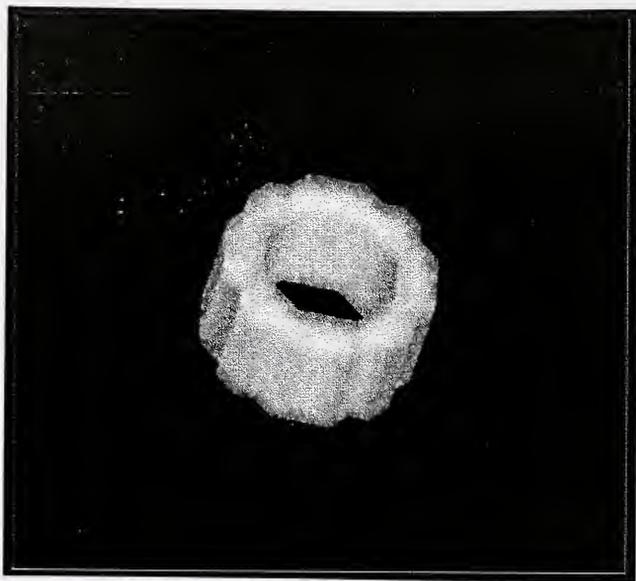


Figure 7-8. Hoop shaped target surrounding normal tissue.

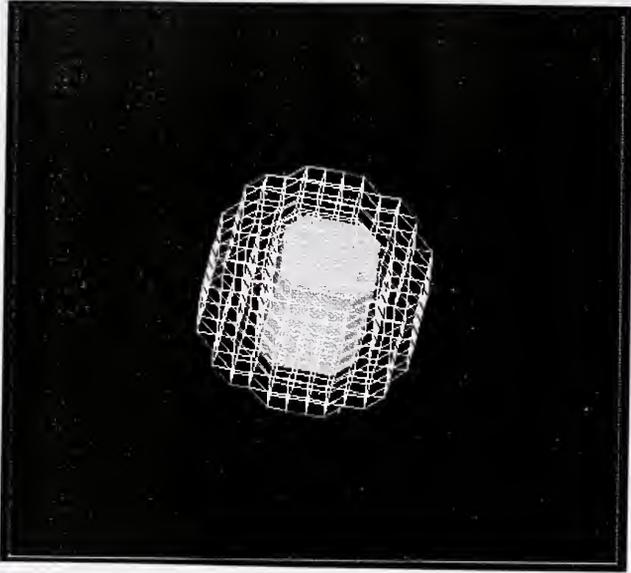


Figure 7-9. Hoop shaped target surrounding a critical structure.

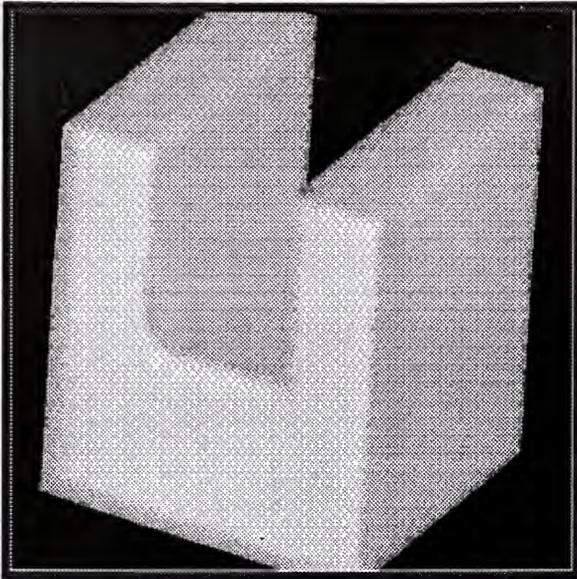


Figure 7-10. A notched cube shaped target with normal tissue in the notch.

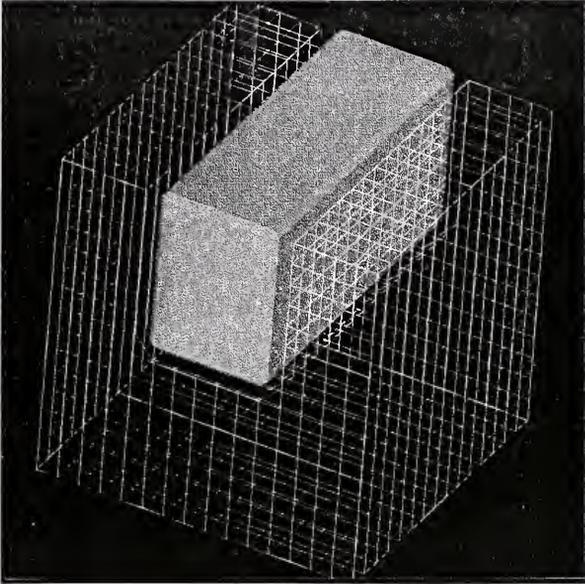


Figure 7-11. A notched cube shaped target with critical structure tissue in the notch.

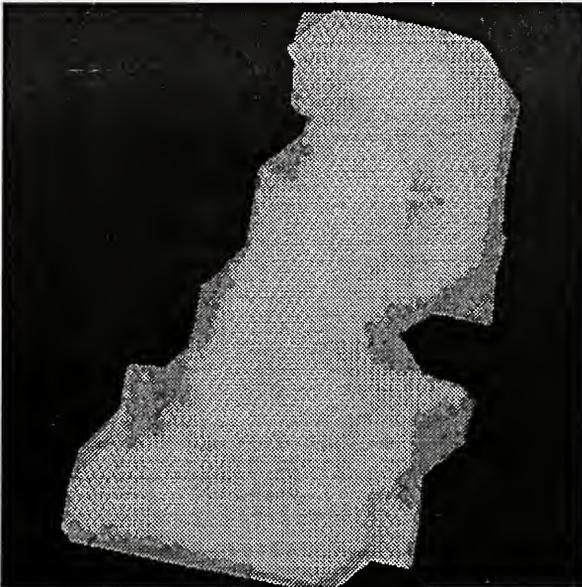


Figure 7-12. Irregular Shaped AVM target.

for the single sphere, were chosen for evaluation since they would have required multiple isocenters if planned and treated with conventional radiosurgery tools. Each isocenter would have required multiple arcs and possibly a unique collimator thus resulting in a potentially lengthy and complicated plan. In contrast, the targets chosen here were planned with a single isocenter and only three arcs. Similar to the current radiosurgery procedures at the Shands cancer clinic, each of the three arcs were limited to 100° (40° to 140°) but were approximated by six views separated by 20° gantry rotations. The table angles were chosen to be -45° , 0° , and $+45^\circ$ (see figure 7-13). Note also, that no special effort was made to insure that optimal views of the target were included. For example, none of the 18 views of the hoop shaped target could "see" a round opening in the center. These self imposed limitations arise from a goal of this research to investigate how conformal a plan can be achieved with a reasonable number of incident beams.

The output available for evaluation of each case includes 1) a numerical data window, 2) a differential dose volume histogram (dDVH) window, 3) a 3-D isodose surface rendering window, and 4) an orthogonal slice isodose contour window. This output provides both qualitative and quantitative comparison. Normally, this plan evaluation data is simultaneously accessible within a single user screen. From the first (upper left) window, the user may note the monitor units assigned to each table and gantry position,

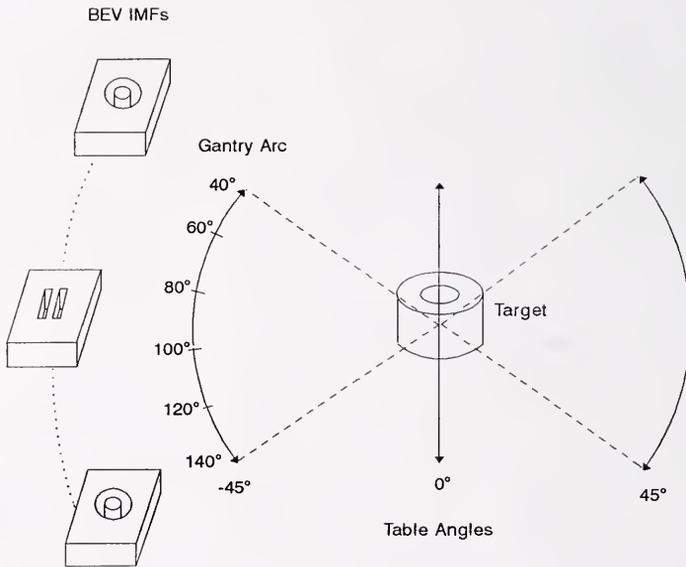


Figure 7-13 Illustration of geometry used for incident beam arrangement. Each target was irradiated by three 100° arcs (table angles: -45°, 0°, and 45°).

the objective function value, the prescribed dose, and the maximum and minimum absorbed dose values assigned to each tissue type. Using the dDVH (upper right) window, one may observe the distribution of dose between the maximum and minimum dose values for each tissue type. For the examples to be presented, the normal, target, and critical structure tissue curves are solid, dashed, and dotted respectively. Note that each tissue specific curve is normalized to the maximum dose of its own tissue and the bin widths are 1%. From these curves, one can choose an isodose level for a desired target coverage and visualize such coverage using the 3-D rendering capability of the third (bottom left) window. The isodose contour window (bottom right) will then permit slice by slice investigation of the resulting plan. Unless otherwise noted, for all example cases presented, the 90, 80, 70, 60, 50, and 10% isodose lines are shown in addition to an outline of the target structure. For simplicity, the results of each example case will be presented as a "snapshot" of the entire treatment plan evaluation screen with all four windows, and a control panel, shown simultaneously. In all cases, the results for the simulated annealing solution will appear at the top of the page while the results for the downhill simplex will appear at the bottom.

The results of the six test cases, under the conditions just described, are shown in figures 7-14 through 7-24. The first case, shown in figure 7-14, is a single sphere 3 cm in

diameter. This target would be easily treatable with a single isocenter using a circular collimator and five to nine arcs typical of linac based radiosurgery. Such a plan would result in complete target volume coverage with the 80-90% isodose line. Using three arcs and a finite ($\cong 5$ mm) IMF resolution at isocenter results in 65% and 60% isodose values for the simulated annealing (SA) and downhill simplex (DHS) optimization algorithms respectively. These values, although accurate, can be somewhat misleading. They represent the isodose value needed to ensure every single voxel comprising the target sphere receives at least the prescribed dose (10 Gy for all cases shown). Since the discrete nature of the voxels results in a "bumpy" sphere, the 100% target volume coverage isodose values are lower in order to include the corners of the voxels. A more realistic approach would be to choose an isodose value resulting in 90-95% target volume coverage and would be consistent with the current clinical judgment used at the Shands Cancer Clinic for radiosurgery cases. Using a conservative goal of $\geq 95\%$ target volume coverage, the isodose values for the sphere example increase to 70% (SA) and 65% (DHS). Except for a higher isodose coverage, the SA plan is very similar to the DHS plan. This is true in spite of the fact that the monitor unit values assigned to each BEV are different thus illustrating the multiple minima aspect of solution space.

The second case, shown in figure 7-15, is a cluster of

three small, approximately spherical targets with a single isocenter placed near the center of the cluster. All spheres are 10 mm in diameter and are located approximately 16 to 18 mm from one another (center to center). For this case, both the SA and DHS plans resulted in $\geq 95\%$ target volume coverage with 55% isodose surface values. In cases like this one, where multiple targets are not all visible in a single plane of the orthogonal slicer, the rotateable surface renderings appear to provide much more information. For this example only, the isodose lines shown in the orthogonal slicer window are 60, 50, 40, and 30%.

The third case, shown in figure 7-16, is a tall cylinder (rod) 20 mm in diameter, 30 mm long, and contains an isocenter near its geometric center. Using similar target coverage judgment as with the previous examples, a 65% isodose surface value can be chosen for both the SA and DHS plans. The end edge of the target not enclosed by this isodose surface represents the leading tail of the dDVH. Also note that the three table angles are revealed by including the 10% isodose line.

The fourth case, shown in figure 7-17, is a hoop 14 mm tall with inner and outer diameters of 12 and 24 mm respectively. The isocenter was placed near the center of the cylindrical hole. The resulting plan appears to be fairly conformal with $\geq 95\%$ target volume coverage achieved with the 65% isodose surface chosen for both the SA and DHS plans. A new plan was next accomplished for this same target

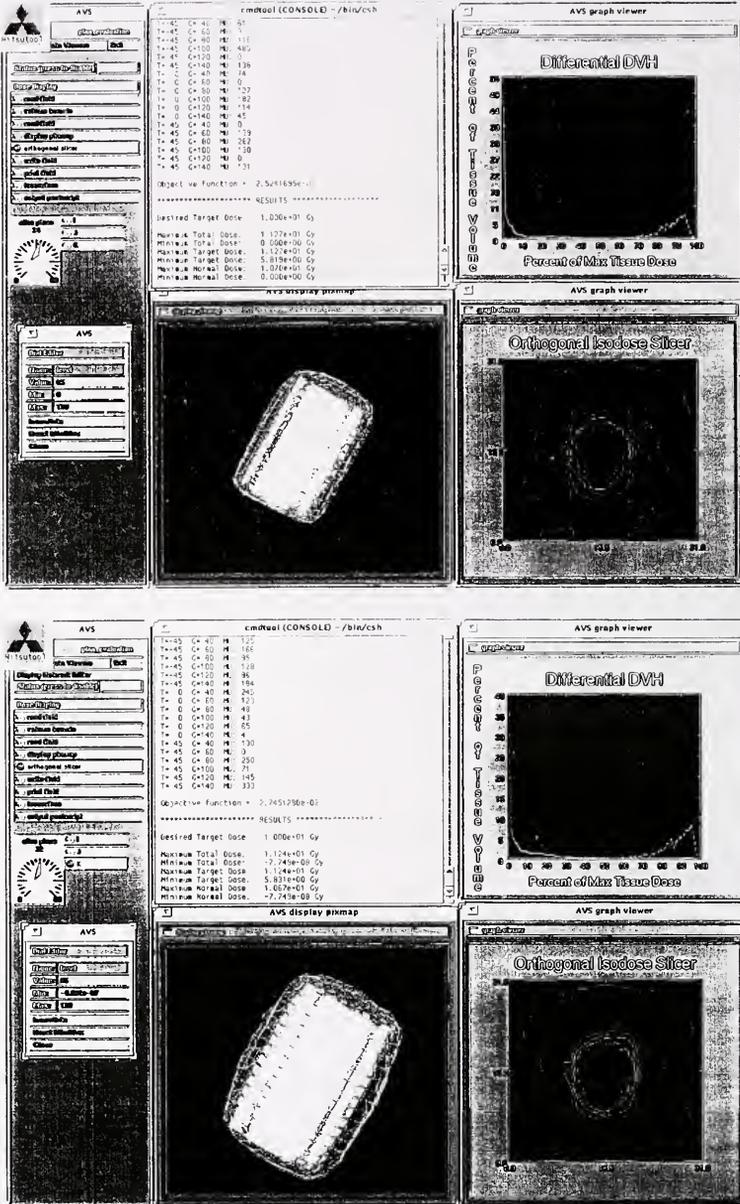


Figure 7-16. Plan results for rod shaped target. Top: SA, bottom: DHS.

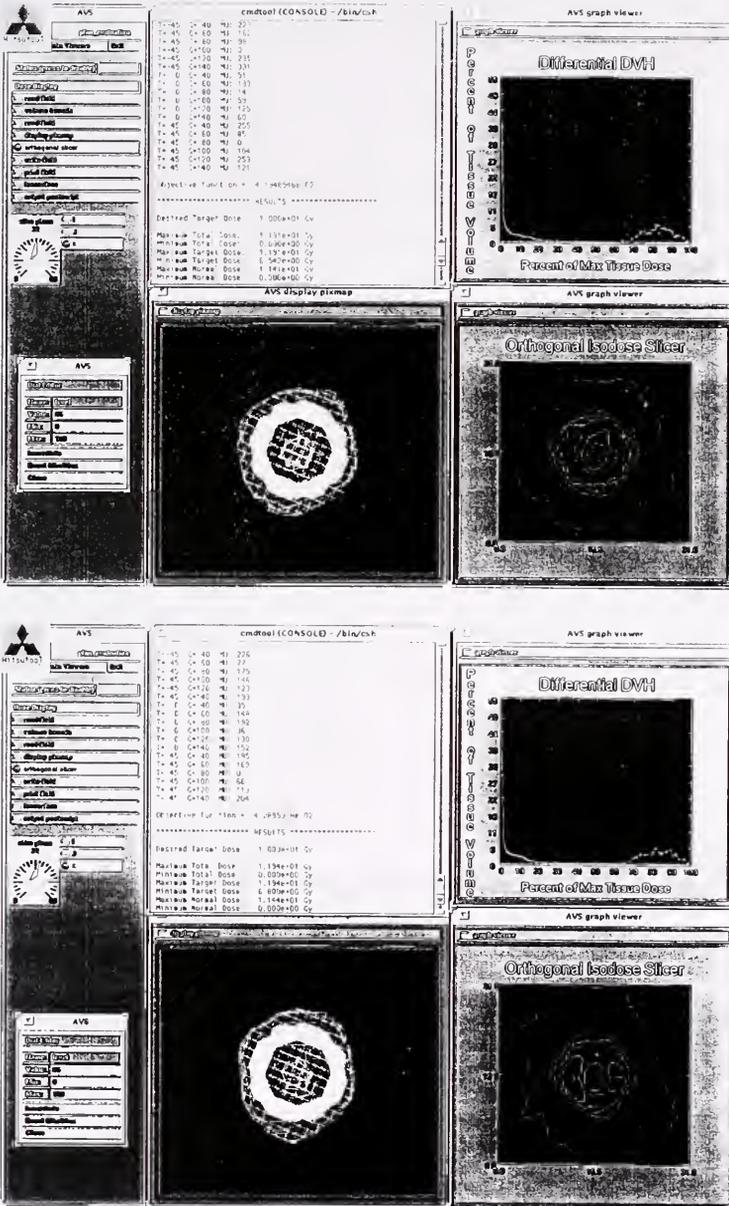


Figure 7-17. Plan results for hoop shaped target without a central critical structure. Top: SA, bottom: DHS.

after characterizing the central hole as critical tissue with a weighting of 1.0 (maximum). The results, shown in figure 7-18, clearly indicate a reduced dose to this central area. Comparing isodose contours for the two plans in question reveals a significant difference in dose to the central critical structure. This sparing results from the critical tissue being considered during both the formation of each BEV intensity modulation grid and during the beam weighting optimization procedure. The drawback to the critical structure sparing is the broadening of the target tissue dose range. By sparing the central core, a significant portion of the target tissue surrounding the core was underdosed reducing the $\geq 95\%$ target volume coverage isodose value to 55 and 50% for the SA and DHS plans respectively. This trend is most easily seen using the dDVH's or the orthogonal isodose slicer window. Also note that, as shown in the 3-D surface rendering, the 1.0 weighting value resulted in excessive dose to normal tissue in an effort to spare the critical structure and still irradiate the target. This effect is reduced by changing the critical structure weight to 0.5 as shown in figure 7-19.

The fifth case considered is shown in figure 7-20. This target is the largest of all with exterior dimensions of 32x32x32 mm and contains a notch 16 mm deep and 12 mm wide. Again, the isocenter was placed near the center of the target. The isodose values chosen for this target were 65 and 60% for the SA and DHS plans respectively. Similar to

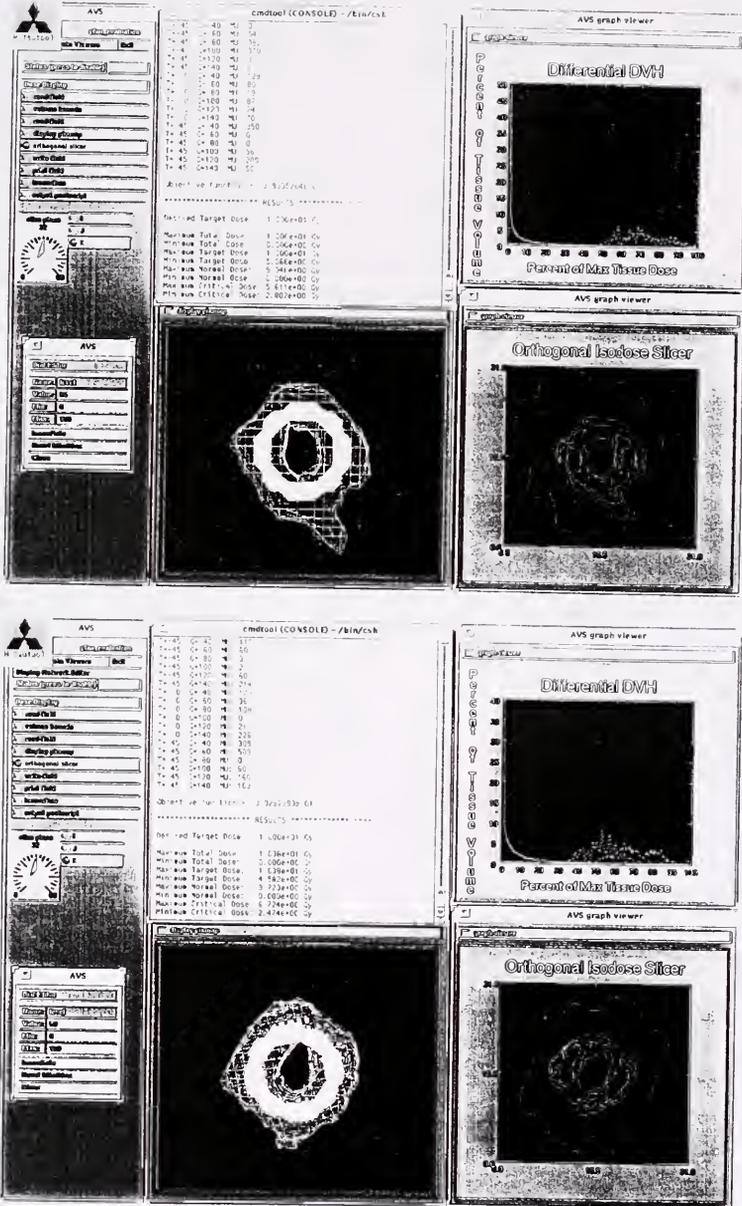


Figure 7-18. Plan results for hoop shaped target with a central critical structure assigned a weighting of 1.0. Top: SA, bottom: DHS.

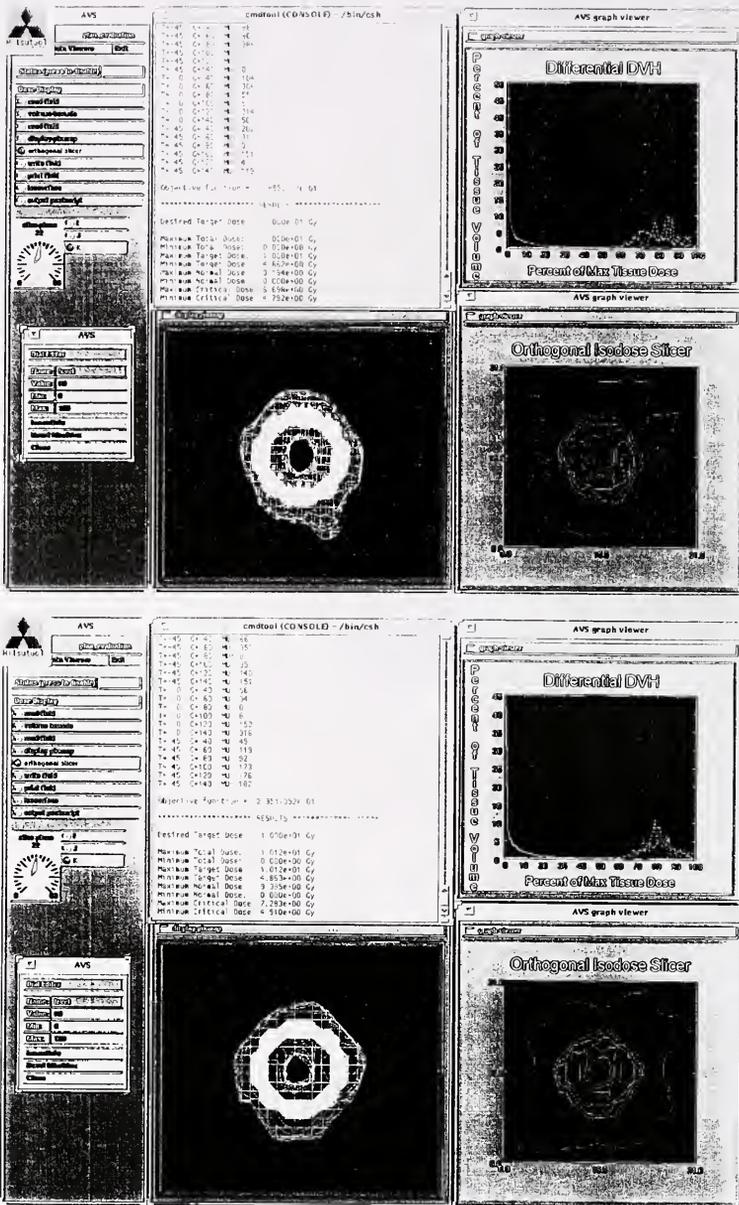


Figure 7-19. Plan results for hoop shaped target with a central critical structure assigned a weighting of 0.5. Top: SA, bottom: DHS.

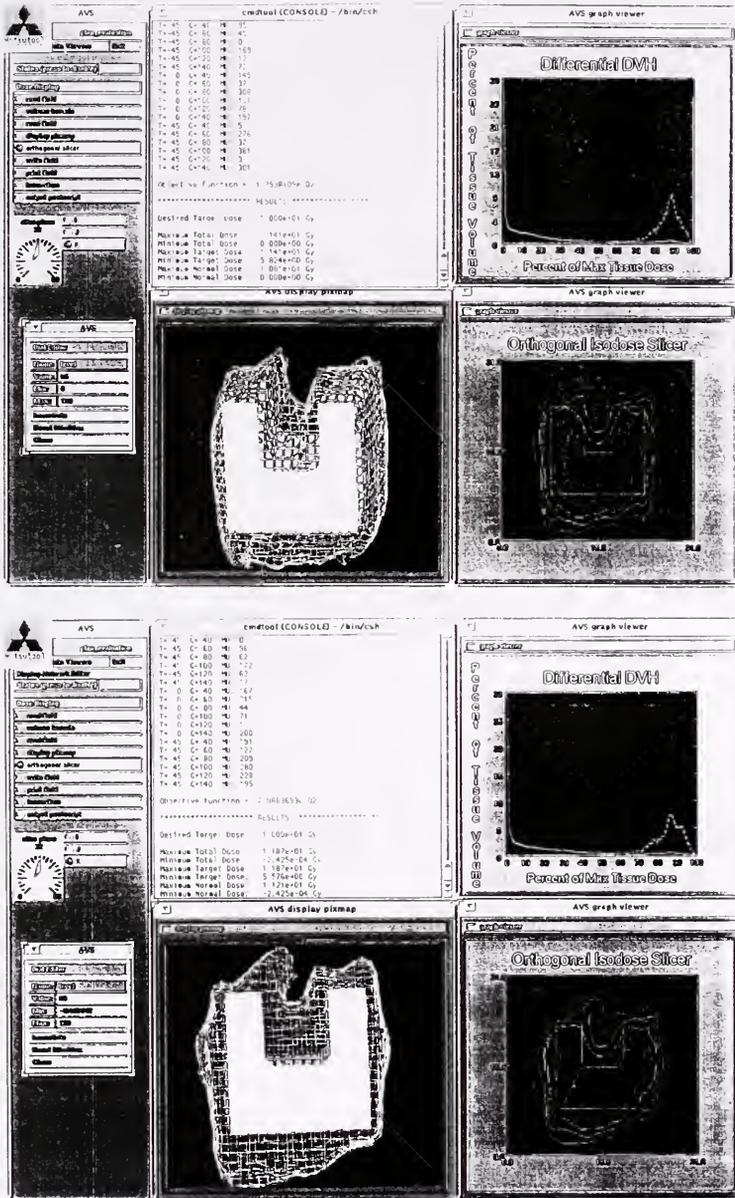


Figure 7-20. Plan results for notched cube shaped target without a critical structure. Top: SA, bottom: DHS.

the hoop shaped target, an attempt to spare critical structure tissue filling the notch (weight = 1.0) is successful but at the expense of reducing the $\geq 95\%$ target volume coverage isodose line to 55% for both the SA and DHS plans (see figure 7-21). The results of an attempt to increase the minimum target dose while still sparing the critical structure is shown in figure 7-22 where, again, the critical structure tissue weighting was decreased from 1.0 to 0.5. While the target dose did increase somewhat (from 55% to 60% for $\geq 95\%$ target volume coverage), the maximum critical structure dose also increased (from approximately 55% to 71% of maximum target dose). This effect is best seen with the dDVH curves which show a more peaked target curve and a critical structure curve shifted towards higher dose values.

The final radiosurgery sized example is a target of the same size and shape of a previously treated AVM. The highly irregular shaped target was treated to the 60% isodose line with three isocenters (each with five 100° arcs for a total of 15 arcs) and two circular collimators. The results of this plan are shown in figure 7-23 with the 60, 30, 12, and 6% lines displayed. In contrast, the single isocenter, three arc plan, shown in figure 7-24, results in a 65% isodose line treatment for both the SA and DHS approaches.

A summary of statistics for all example cases is presented in table 7-1. After reviewing the results of these small target plans, it should be apparent that a good degree

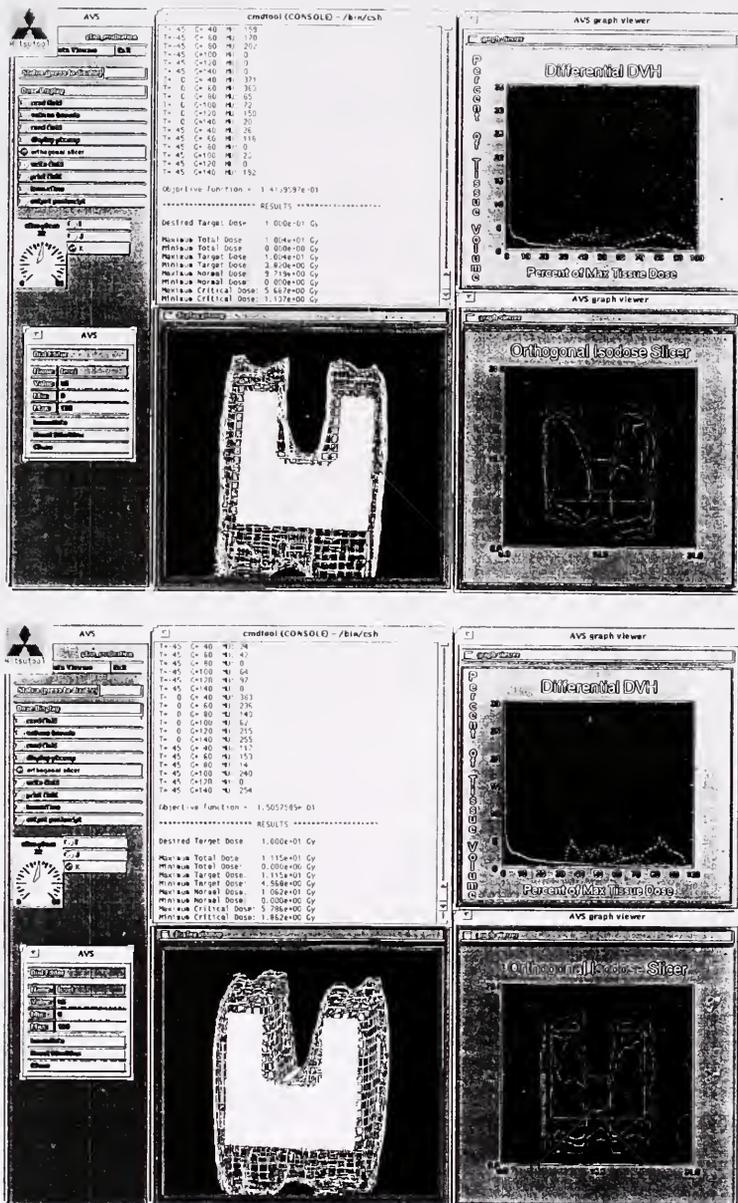


Figure 7-21. Plan results for notched cube shaped target with a critical structure assigned a weighting of 1.0. Top: SA, bottom: DHS.

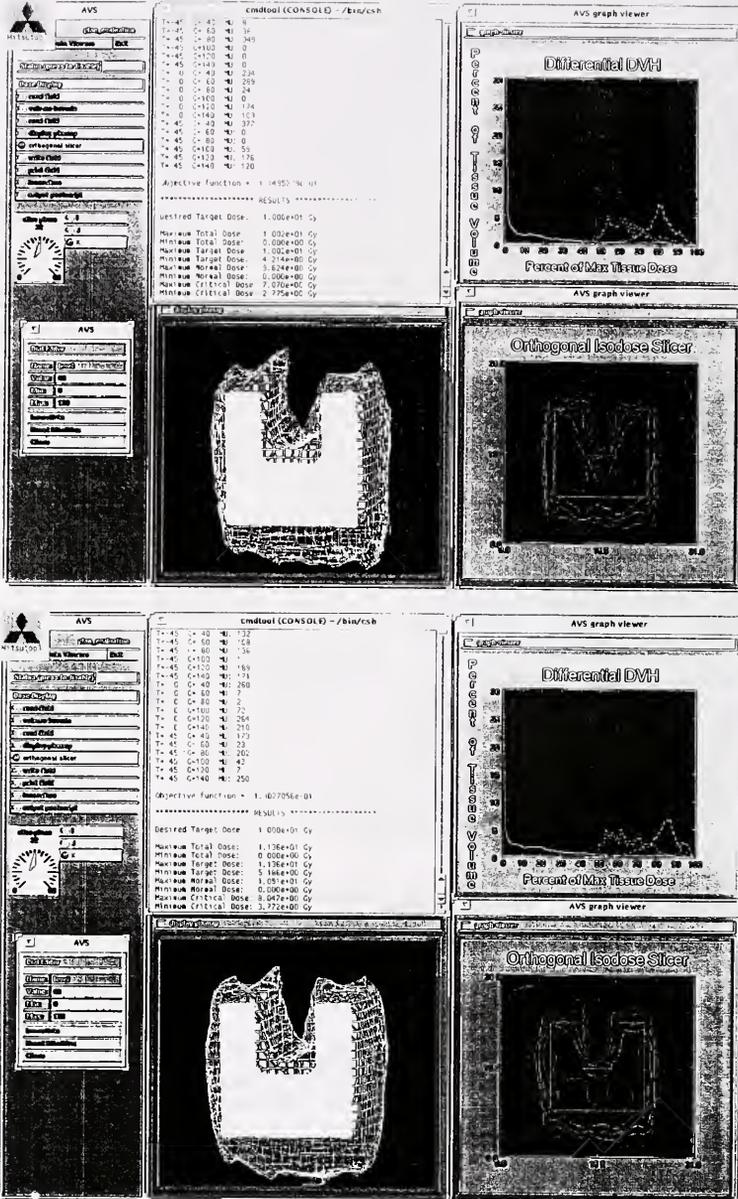


Figure 7-22. Plan results for notched cube shaped target with a critical structure assigned a weighting of 0.5. Top: SA, bottom: DHS.

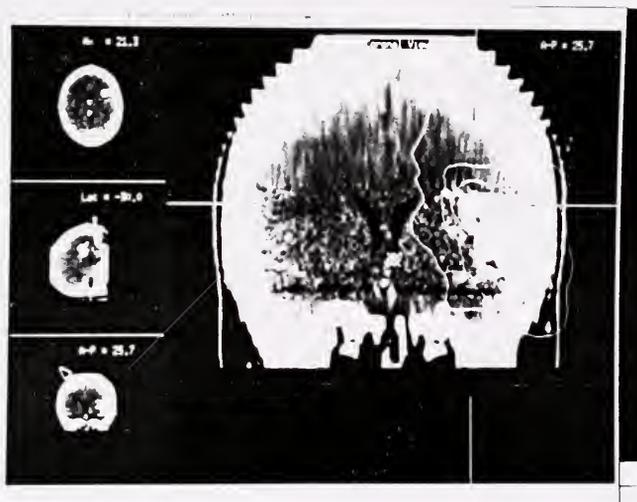


Figure 7-23. Three isocenter isodose distribution created for the AVM case by the radiosurgery planning system used at Shands Cancer Clinic.

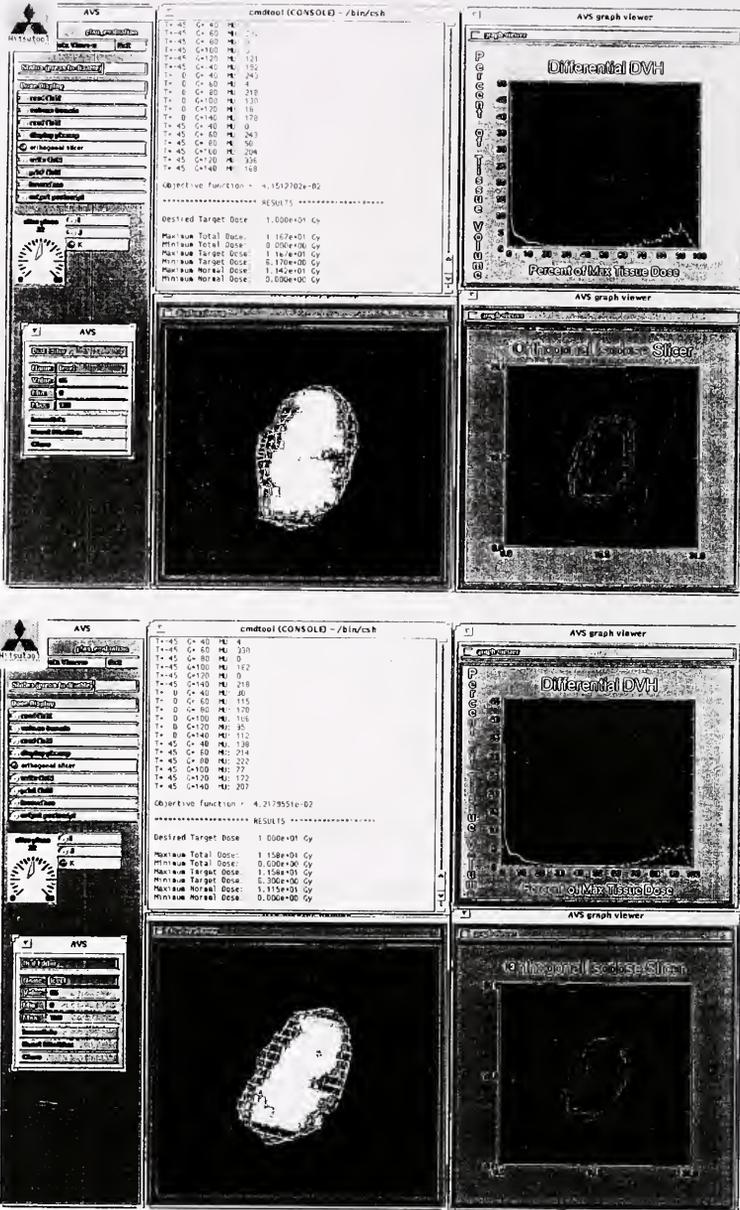


Figure 7-24. Plan results for irregular AVM target. Top: SA, bottom: DHS.

Table 7-1. Test Case Summary

Test Case	CPU (hrs)	Target Coverage 100%	Target Coverage >95%	Peak CS Dose (% of Max TD)	Obj Function
3 Balls\SA	3.61	48	55	na	0.1289
3 Balls\DHS	1.17	48	55	na	0.1352
Rod\SA	4.02	51	65	na	0.0252
Rod\DHS	1.18	52	65	na	0.0275
Sphere\SA	4.45	65	70	na	0.0183
Sphere\DHS	1.17	60	65	na	0.0320
Hoop\SA	3.91	55	65	na	0.0419
Hoop\DHS	1.18	57	65	na	0.0429
Hoop\cs\SA	3.41	51	55	56	0.3931
Hoop\cs\DHS	1.18	44	50	65	0.3826
Hoop\cs\5\SA	3.58	47	60	67	0.2896
Hoop\cs\5\DHS	1.17	48	60	72	0.2954
NC\SA	4.23	51	65	na	0.0179
NC\DHS	1.18	47	60	na	0.0209
NC\cs\SA	3.64	38	55	57	0.1414
NC\cs\DHS	1.18	45	55	52	0.1506
NC\cs\5\SA	3.73	42	60	71	0.1150
NC\cs\5\DHS	1.17	46	60	71	0.1303
NC\360arc\SA	2.75	63	65	na	0.0163
NC\360arc\DHS	0.83	57	60	na	0.0192
AVM\SA	3.95	53	65	na	0.0415
AVM\DHS	1.18	54	65	na	0.0422

of conformity ($\geq 60\%$ isodose line treatment for most cases) can be achieved with a relatively small number of BEVs. While this isodose treatment level is lower than the 80-90% used at Shands Cancer Center for single isocenter cases, it compares well with that used for multiple isocenter cases. It also appears that using the SA vs DHS algorithm makes little difference in the quality of the final plan. Thus, using the DHS approach provides reasonable results in approximately one hour which is comparable to conventional treatment planning times for most cases while difficult, multiple isocenter cases may take considerably longer.

As was shown in many of the example cases, local areas of underdosing result in a broadening of the target dDVH curve indicating an increase in target dose inhomogeneity. This inhomogeneity is approximately $39 \pm 5\%$ averaged over the example presented. For most cases, the areas of underdosing occur in sharp corners or edges of the target which are artifacts of the discrete voxel sizes rather than realistic features one might expect. In any case, the degree of target tissue coverage to be achieved is determined on a case by case basis and depends on the specific clinical goals. Also, recall that all example plan results are based on a limited number of arcs (3) and range per arc (100°). Thus, these cases represent what are considered extreme examples and serve to illustrate the physically attainable limits. If a more optimal arc set is allowed, which permits better characterization of target structure, much improved

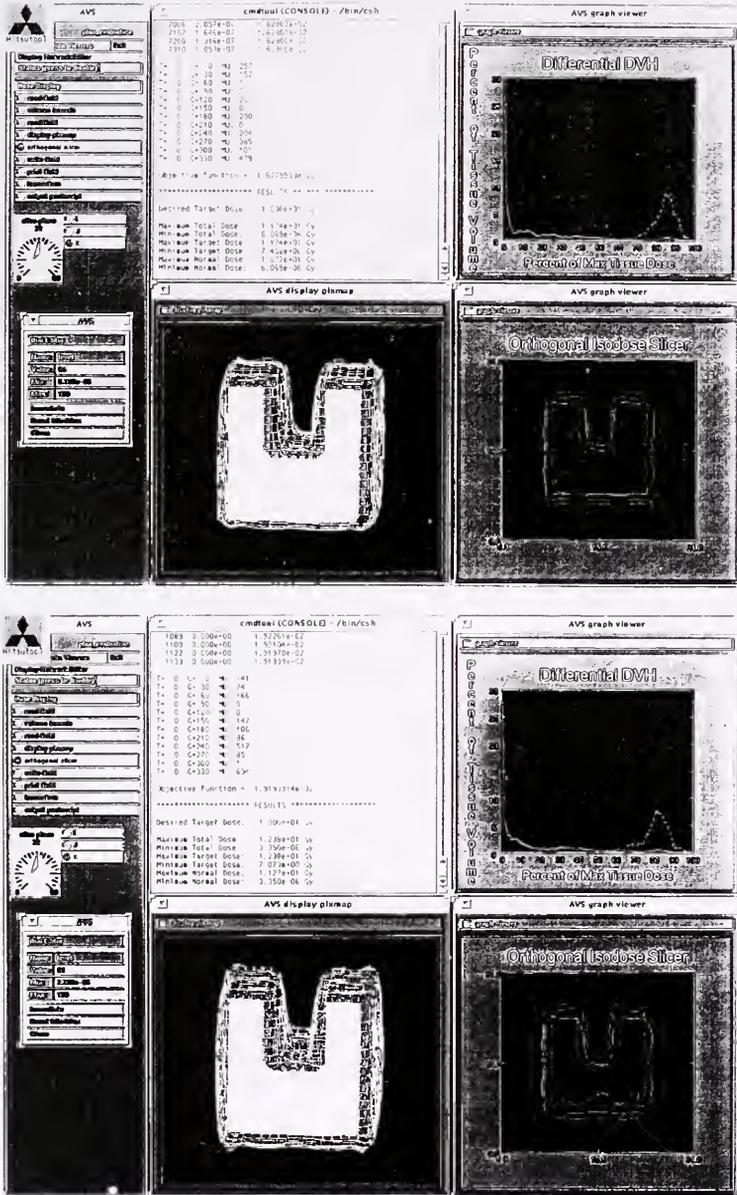


Figure 7-25. Plan results for notched cube shaped target (without critical structure) using a single 360° gantry arc.

plans can result. An example of this is illustrated in figure 7-25 where a single, properly chosen arc (table = 0° , gantry = 0° to 330° in 30° steps) produced a more conformal plan to those previously shown.

Having presented results for several radiosurgery sized targets, the next chapter will address the application of this approach to larger targets. Here, the flexibility of being able to choose any desired number and direction of incident beams will become evident.

CHAPTER 8
EXTENSION TO GENERAL RADIOTHERAPY

Overview

Although this research was focused primarily on the field sizes associated with radiosurgery, an effort was made to allow application to larger field situations. This chapter reviews the assumptions used to simplify the small field (i.e. less than 32 mm diameter) stereotactic treatment planning dosimetry and proposes modifications to allow improved larger field agreement. The interest in this extension lies in the fact that conventional larger field multiple beam plans (parallel opposed, four field box, etc.) may be improved or replaced by new arrangements if full field intensity modulation is available. In fact, research at the U. of Florida Shands Cancer Center is currently focused on assessing the advantages associated with intensity modulated conformal treatment for a variety of tumor sites in the body.

Since the results shown to date represent narrow field beams, it is of interest first to see how well the convolution dosimetry calculations model larger fields if small field approximations remain in place. If the dosimetry calculations carry over, then the entire inverse planning approach should also carry over as the remaining aspects of

the planning algorithm are either independent of field size or are easily scaled. After reviewing the small field dosimetry simplifications, such large field dosimetry data will be presented and a discussion of such approximations will be presented.

Small Field Approximations

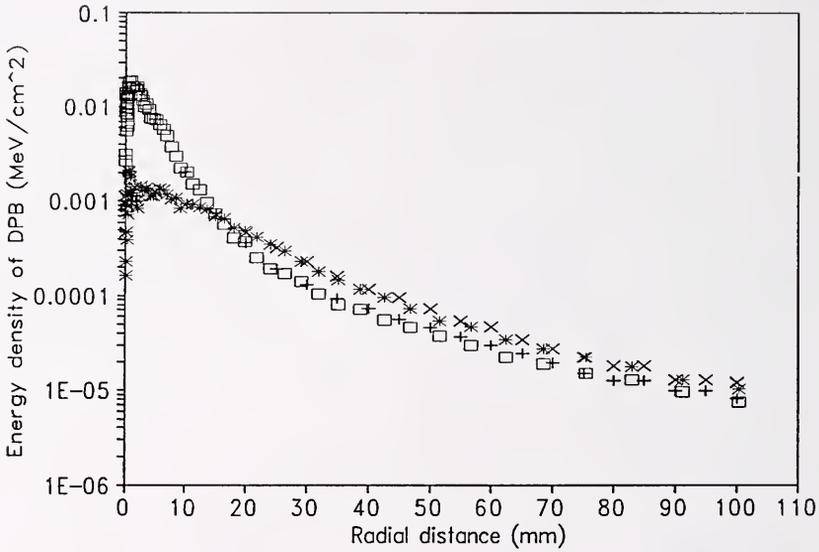
Due to the small size of the fields used for stereotactic radiosurgery, many complicating factors could be neglected. First, there was no need to tilt the energy deposition kernels along ray lines since the rays will be almost parallel. It has been shown that the difference in dose between tilted and non-tilted kernels is less than 1% for a 5x5 cm field at 100 cm SSD (Sharpe & Battista 1993). This speeds up the calculation process since the tilting results in noninvariant kernels which are not compatible with FFT convolution techniques. Second, it has been shown that the surface effects of photons scattered in small secondary collimators (Bjarngard, Tsai & Rice 1990) or electron contamination from Lipowitz compensators for small fields (Cardarelli, Rao & Cail 1991) is not significant. Because of this, no additional factors to account for surface effects were included. Third, it has been shown that the energy spectra of a photon beam is essentially constant for off-axis distances less than 3 cm at isocenter for the energies of interest (6 MV) (Mohan, Chui & Lidofsky 1985). Thus, no off axis spectral effects were included in the small field cases. Finally, the diameter of the narrow beams

relative to the curvature of the skull (assumed spherical) allowed the assumption a flat phantom surface which is ideal since the DPB kernels were generated with such a geometry.

Large Field Dosimetry Verification

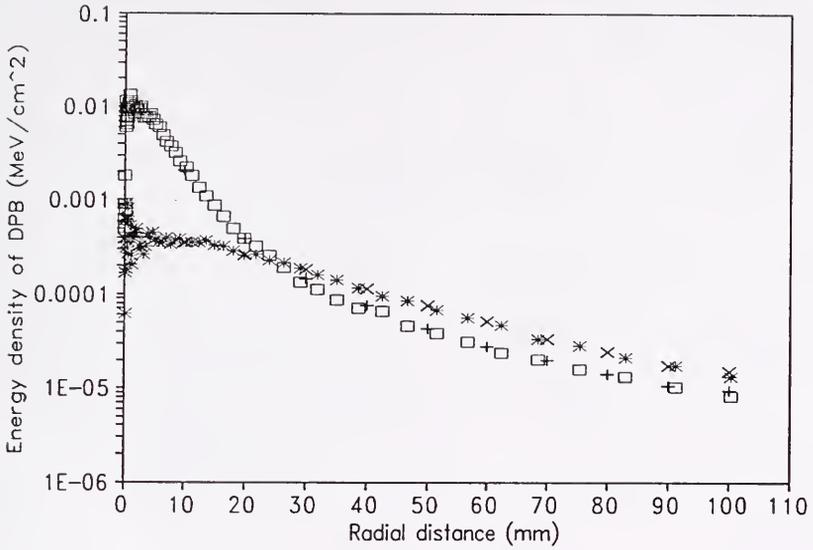
In order to plan a large field case, the voxel dimensions of the DPB and calculation volume must be increased. The dimensions for the DPB and calculation volume must be equal and were 2 mm in all dimensions for the small field case resulting in an approximate maximum beam diameter and calculation volume of 32 mm and 64x64x96 mm respectively (using a 32x32x64 array size). Recall that the full extent of the 32x32x64 array can not be used due to wrap-around effects. Increasing the DPB and calculation volume dimensions from 2 mm to 5 mm results in an approximate maximum beam diameter and calculation volume of 8 cm and 16x16x24 cm respectively. If the dimensions were further increased to 10 mm, the maximum beam diameter and calculation volume would be approximately 16 cm and 32x32x48 cm respectively. The DPBs needed for these dimensions have been created using the DPBVOX algorithm and examples are illustrated in figures 8-1 and 8-2.

The attenuation grid, set at a minimum size of 3 mm on edge for narrow field data, may also be increased to represent a value more practical for larger targets. An upper limit for this value, suggested by current multi-leaf collimator designs, would produce approximately 1 cm wide pencil beams at an SAD of 100 cm. With my current choice of



□ Raw 12.32 m + Avg 12.5 mm * Raw 42.5 mm × Avg 42.5 mm

Figure 8-1. Uniformly scaled (5x5x5 mm³ voxels) differential pencil beam data for selected depths.



□ Raw 14.91 m + Avg 15 mm * Raw 75.38 m × Avg 75 mm

Figure 8-2. Uniformly scaled ($10 \times 10 \times 10 \text{ mm}^3$ voxels) differential pencil beam data for selected depths.

attenuator distance of 65 cm, a maximum attenuator grid dimension of 6.5 mm would be appropriate. This value, or any other desired grid resolution may easily be incorporated into the inverse treatment planning algorithm presented here.

Also needed for each large field case is an effective spectrum. Recall from chapter 4 that such a spectrum was obtained from measured absolute depth dose per monitor unit data for a specific collimator setting and voxel dimension. Repeating this procedure for larger field sizes and voxel dimensions does not produce significant changes in the relative spectrum needed to properly fit measured data. However, the method is sensitive enough to detect a slight change in absolute output as field size and/or voxel dimensions change. Thus, appropriate depth dose data should be collected for the range of desired field and voxel sizes and processed using the SPECTRUM algorithm.

Isodose curves resulting from PLAN using larger DPB and calculation voxel dimensions and appropriate spectral output are shown in figures 8-3 through 8-6. Figures 8-3 and 8-4 compare the results for an 8×8 cm² (SSD = 100 cm) field in a plane parallel to and through the central axis of the beam. Figure 8-3 compares PLAN convolution results with measured data (obtained using a 0.1 cc ion chamber in a water phantom) and figure 8-4 compares that calculated by the THERAPLANTM treatment planning system used at the Shands Cancer Center with the measured data. This comparison is

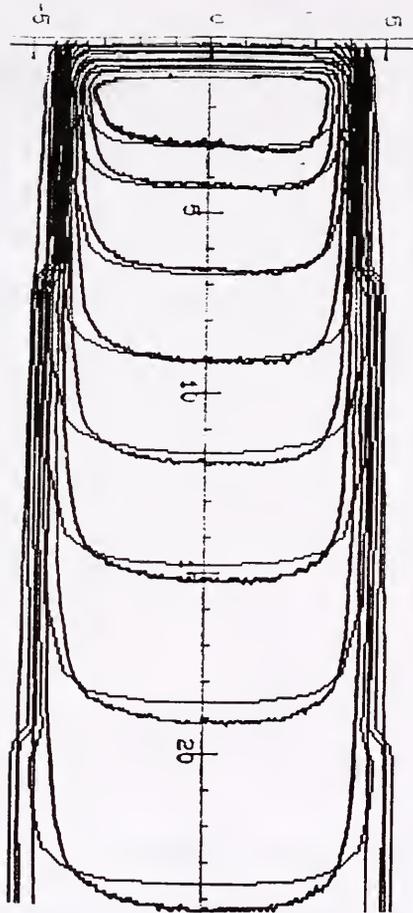


Figure 8-3. Comparison of PLAN calculated and measured isodose curves for an 8×8 cm², SSD = 100 cm field. The voxel dimensions for the calculation volume and attenuation grid are both 5 mm.

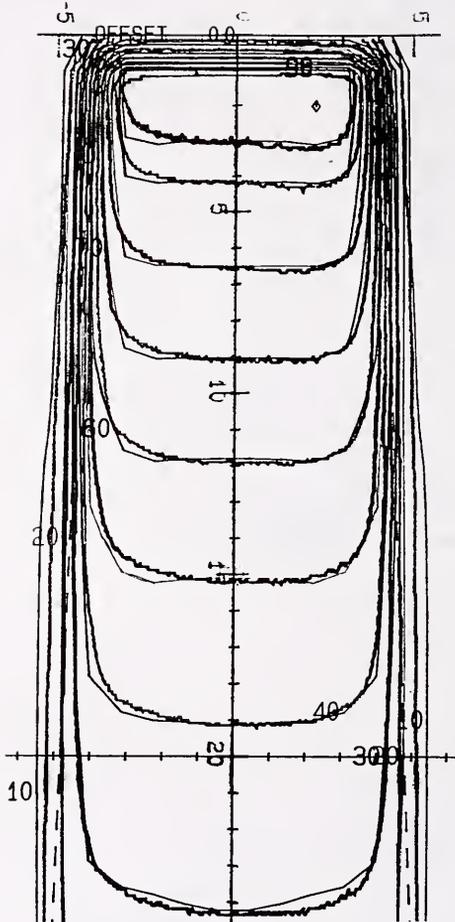


Figure 8-4. Comparison of THERAPLAN™ calculated and measured isodose curves for an 8x8 cm², SSD = 100 cm field.

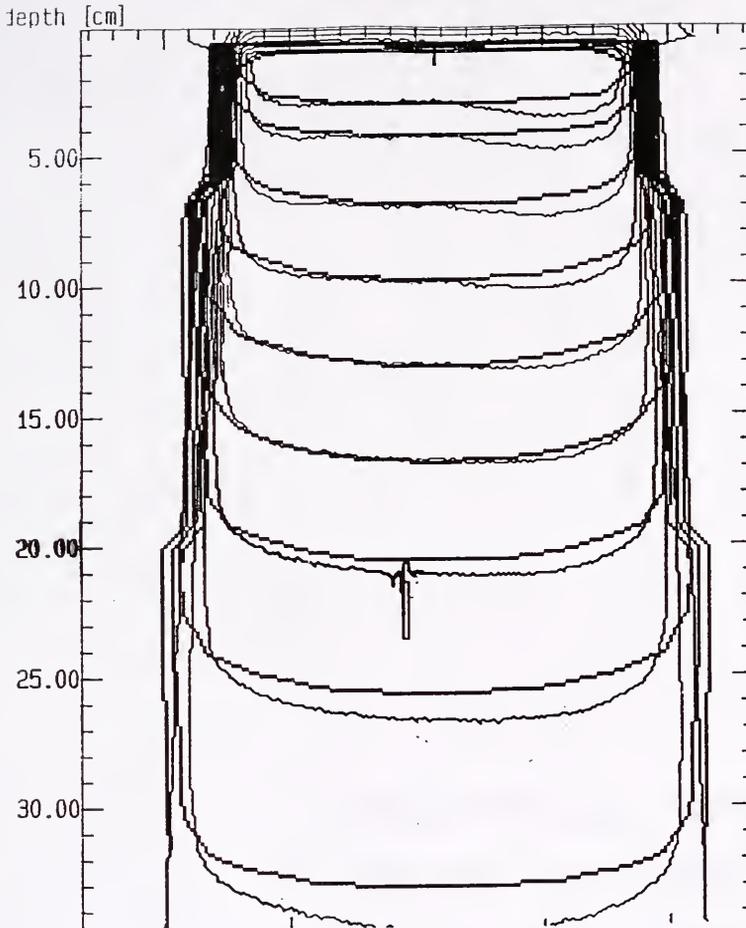


Figure 8-5. Comparison of PLAN calculated and measured isodose curves for an $16 \times 16 \text{ cm}^2$, $\text{SSD} = 100 \text{ cm}$ field. The voxel dimensions for the calculation volume and attenuation grid are 10 mm and 5 mm respectively.

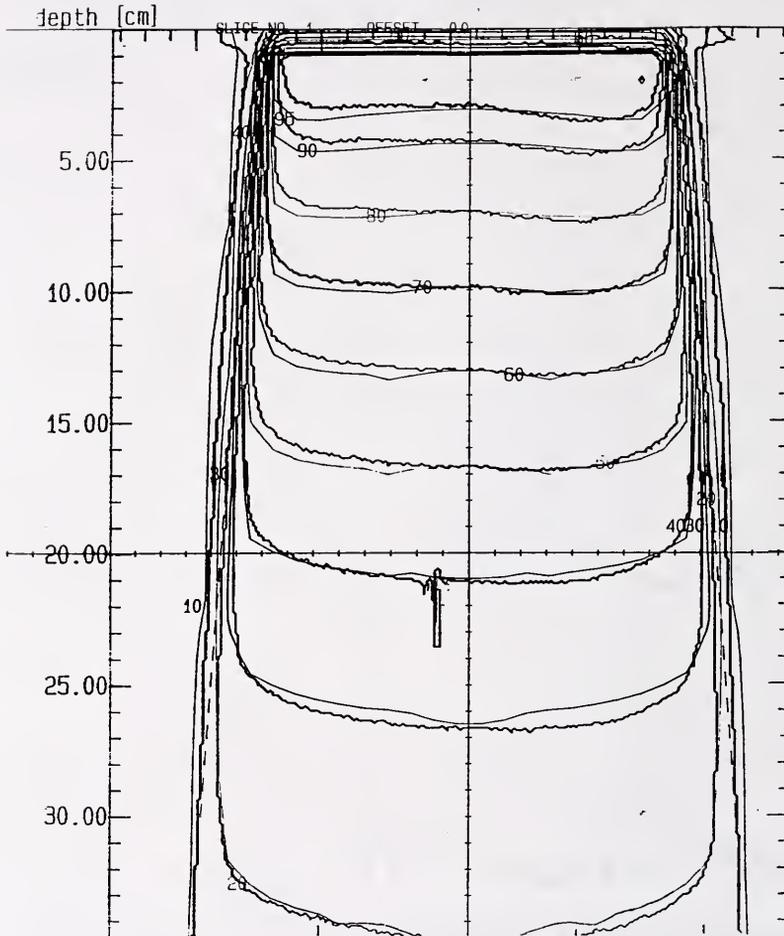


Figure 8-6. Comparison of THERAPLAN™ calculated and measured isodose curves for an 16x16 cm², (SSD = 100 cm) field.

repeated in figures 8-5 and 8-6 for the case of a 16x16 cm² field.

These comparisons indicate that neither the THERAPLANTM nor the convolution data exactly match the measurements. While both dosimetry models match measured data well along the central axis, they both differ somewhat from measurements in the penumbra region. This difference appears to arise primarily from the discrete voxel sizes used in the calculations. Although the difference in the convolution calculations are also due to exclusion of the geometric penumbra component. Additionally, the convolution data do not model the "horns" present in the measured profiles resulting from the flattening filter as THERAPLANTM does.

Having examined large field dosimetry using the small field approximations, the effect, if any, of removing such approximations should be addressed. The first simplification, non-tilting of DPB kernels, has recently been investigated (Sharpe & Battista 1993). Their work has shown that tilting the kernels to be parallel to the divergent ray lines of the incident beam is unnecessary for most clinical situations encountered in external beam radiotherapy. The degree of error between tilting and not tilting, of course, depends on several factors such as SSD, Field size, and beam energy. In general, the more divergent the ray lines are, the more one tends to overdose the central axis relative to off-axis when assuming non-tilted kernels. Their results, based on SSD values from 50 to 100

cm, field size values from 5x5 to 30x30 cm², and beam energy values from 2 to 10 MeV, indicate a difference of less than 3% over most of the irradiated volume if the SSD is greater than 80 cm. Based on these data, it appears there is no need to remove the non-tilted kernel approximation for the moderate field sizes considered here (i.e. ≤ 16 cm).

The second simplification involved the surface dose effects resulting from photons and electrons scattered from tertiary collimators and/or compensators. While the effect has been shown to be negligible for narrow fields (Bjarngard, Tsai & Rice 1990), the effect on larger fields may be more significant. Research has shown (Cardarelli, Rao & Cail 1991) that the surface dose can either increase or decrease relative to the open field depending on the SSD, field size, and thickness of compensator. The trend observed was an increase in the relative surface dose (RSD) as SSD decreased and field size increased, just as one would expect. The effect of the compensator thickness was more interesting with an increase in RSD for thin compensators and a decrease in RSD for thicker compensators. The choice of "thin" or "thick" in this case is determined by the range of the scattered electrons in the compensator material. For 6 MV photons and a lead compensator, the maximum scattered electron range is approximately 3 - 4 mm. In the case of a highly (nontemporally) modulated beam, the RSD may vary across the phantom surface but leave the dose at depth relatively unaffected. Thus, it appears there is no real

need to consider removing such an approximation if one is only concerned with cumulative dose at depth from multiple intersecting beams.

The third simplification assumed the off-axis spectrum was the same as that along the central axis. It has been shown (Mohan, Chui & Lidofsky 1985) that the mean energy of photons gradually decreases as a function of radial distance from the central axis. This trend, due to the triangular shape of the flattening filter, results in the "horns" referred to earlier. An example of this trend is shown in figure 8-7 for a 6 MV beam from a Clinac 6 accelerator. For convolution dosimetry, the variable shown to be the most sensitive to spectral variation is the TERMA. As a result, it is possible to include spectral changes in the TERMA calculations while using a single energy weighted DPB (Sharpe & Battista 1993). Using a similar approach, the current version of the PLAN algorithm could be modified to include spectral variation by including a radial off-axis correction factor during TERMA calculations. Such an approach should not increase the calculation time significantly since the DPB remains invariant during the convolution.

The final simplification used for the small field cases, is the assumption of a flat phantom. The PLAN approximation of a sphere as a flat surface is appropriate when dealing with radiosurgery sized beams. One may also

assume a flat surface for many large field cases depending on the local anatomy. For cases where this approximation is

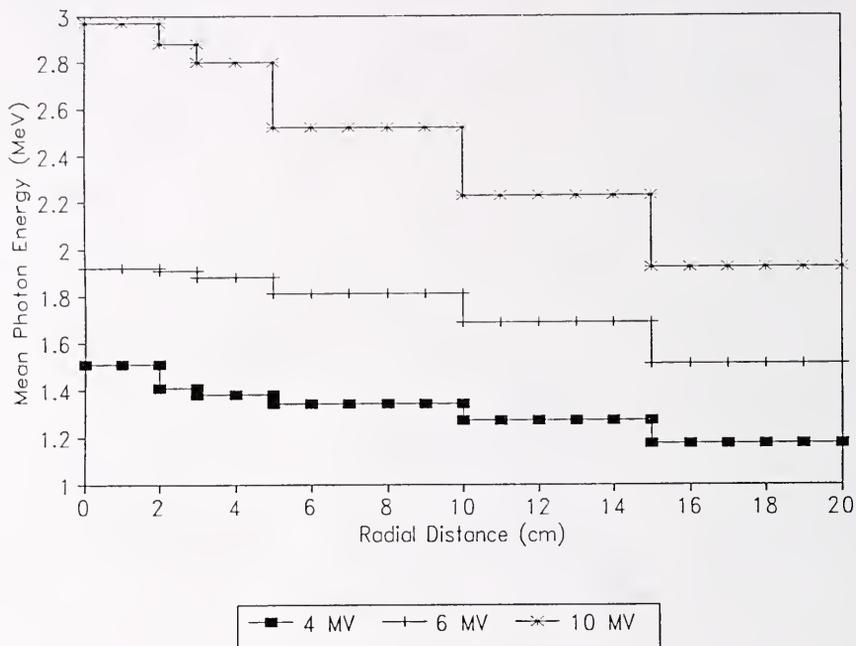


Figure 8-7. Photon mean energy as a function of radial distance from central axis (Data from Mohan, Chui & Lidofsky 1985).

not valid, a correction may be made for the addition of or lack of tissue. With some modifications, such a correction could be incorporated into PLAN by including a numerical "tissue" tag into the current 3-D prescription file. Identification of each calculation voxel as either tissue (water equivalent) or air would be used in both attenuation and voxel specific energy released computations. Incorporation of tissue tagging in this manner would also allow crude modeling of inhomogenities.

In short, with few modifications, the current convolution dosimetry would more accurately model measured data. However, even without such changes, the correlation with measurements is quite good considering its physical simplicity. Based on these observations, it seems reasonable to use the inverse planning algorithm as a tool to explore moderate field size cases without any significant changes.

Example Inverse Radiotherapy case

The target shape chosen as an example was proposed by Mackie et al. (1993) and represents a critical structure surrounded on three sides by target tissue. As shown in figure 8-8, the dimensions of the target are significantly larger than those of the radiosurgery cases presented earlier. The isodose lines shown in figure 8-8 assume 360° rotation in the plane of the paper using the tomotherapy approach discussed earlier. Although no specific timing data were reported for this particular target shape, similar cases are reported to have taken over 20 hours on DECstation 5000/200 using a 64x64x64 calculation array and 5 mm resolution (at isocenter) intensity modulation (Holmes et al. 1993). The isodose contours presented in figure 8-9 (90, 80, 70, 60, 50, and 10% shown) were accomplished in 0.17 (top) and 2.6 hours (bottom) respectively on a SPARCstation¹⁰ using a 32x32x64 calculation array and similar intensity modulation resolution (4.6 mm at isocenter). Note that the top of figure 8-9 presents isodose curves resulting from only four (0°, 90°, 180°, and 270°)

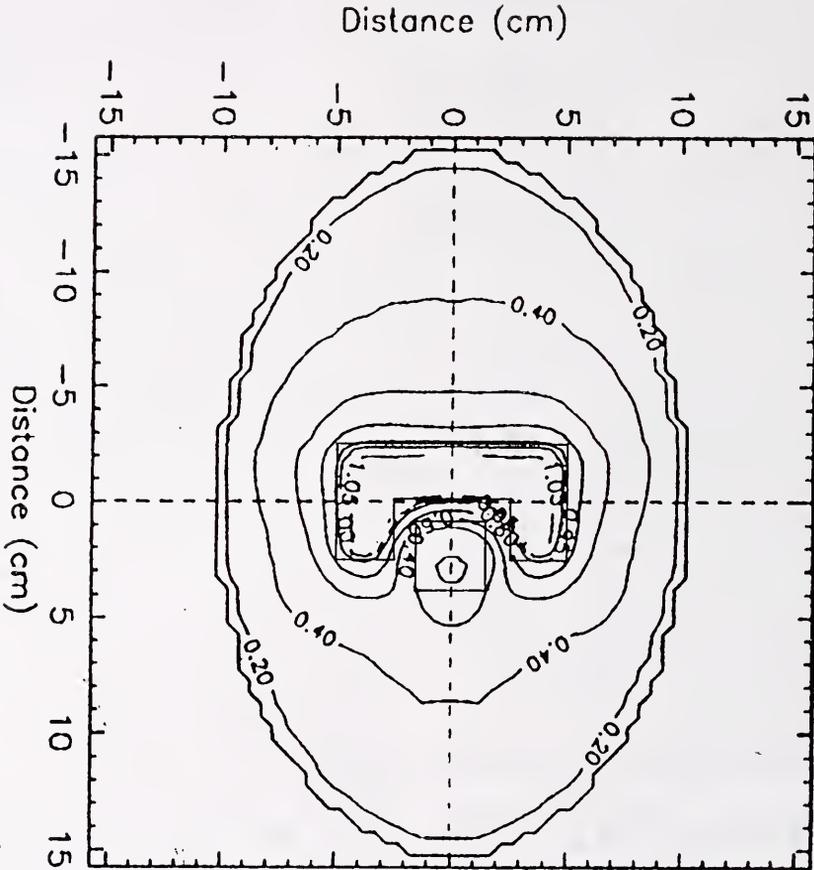


Figure 8-8. Mackie C target and resulting isodose distribution resulting from 360° tomotherapy type irradiation (Mackie et al. 1993).

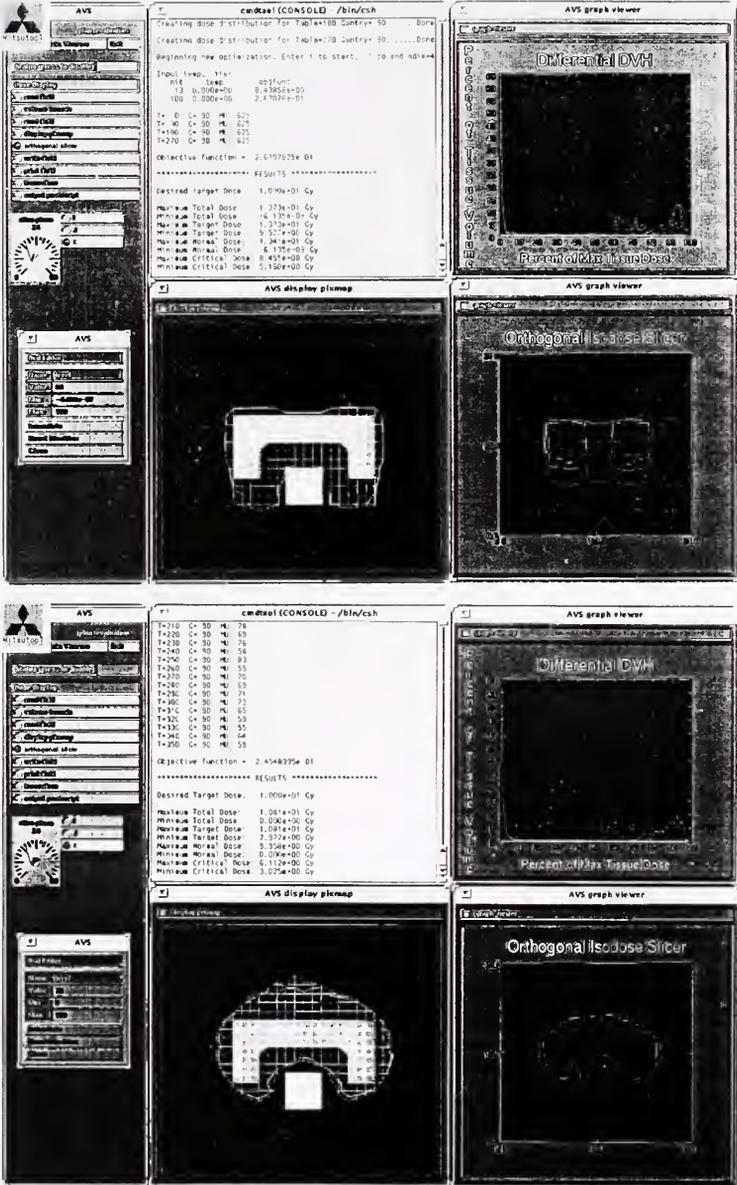


Figure 8-9. Isodose distributions for the Mackie C target generated by PLAN. Top: four beam plan, bottom: 36 beam plan.

beam directions yet provides complete target coverage with the 69% isodose line and limits the critical structure dose to 61%. Increasing the number of incident beams to 36 (one every 10°) increases the target coverage to 71% and decreases the maximum critical structure dose to 56%. While difficult to make accurate timing comparisons on this information alone, it appears the current approach offers a considerable time savings for comparable results. The current method also permits much more flexibility in incident beam arrangement which may prove useful in improving traditional few fixed field treatments.

CHAPTER 9
SUMMARY AND CONCLUSIONS

The inverse treatment planning approach presented here makes use of both deconvolution and simulated annealing techniques. It is an approach that employs the emerging, physically sound, convolution dosimetry and the most promising type of optimization algorithm available. These methods, combined in the manner previously discussed, have shown to provide an automated conformal 3-D treatment planning tool incorporating intensity modulation and phantom scatter. Useful features of this tool include 1) single isocenter planning, 2) variable tissue weighting, 3) variable dimensions of intensity modulation/dose calculation grids, 4) variable number, direction and weighting of incident beams, 5) easily modified objective function, 6) dosimetry linked to linac calibration and includes photon and electron transport, and 7) reasonable calculation times.

Although obviously not a complete treatment planning system, it is believed that this tool demonstrates the degree of conformity possible under the difficult conditions associated with the type of cases presented. These conditions include the small size and irregular shape of the targets, the location of the target near critical structures, the finite resolution of the intensity

modulating device and the limitation of the number and direction of BEVs. The size of the target, of course, plays a critical role when the dimensions of its components are irradiated with photon beams in lateral nonequilibrium states. As previously mentioned, for 6 MV beams, this state exists for any single beam less than 1 cm in diameter. For most of the cases presented here, the irregular shapes resulted in most of the target components being less than 1 cm in width. The small size and irregular shapes also resulted in artificial sharp corners which are difficult to conform to considering the finite resolution ($\cong 5 \times 5 \text{ mm}^2$ at isocenter) of the intensity modulation grid. Obviously, these difficulties can be overcome somewhat by allowing more BEVs in the planning process but at the expense of longer planning times.

While on the topic of planning times, it is worth noting that, other than using an FFT based convolution approach, no special effort was made to increase the speed of the planning algorithm. Thus, there is certainly room for improvements. One idea worth investigating is that of using coarse, random or region of interest sampling of the cumulative dose calculation array during the beam weight optimization. Currently, every single voxel of the $32 \times 32 \times 64$ array is sampled after a beam weight change. A more relaxed sampling may suffice with greatly improved speed. A second approach to increase speed involves so called "fast" or "very fast" annealing schedules. Unlike the currently used

approach, these schedules rely on more extensive objective function testing to arrive at customized cooling schedules. In fact, some methods actually assign unique annealing parameters to each variable dimension (Morril et al. 1993). In any case, annealing schedule optimization is an active area of research with much room for improvement.

Increasing the speed would also make it more practical to extend the calculation volume array size and include biological dose--response models in the objective function. Extending the array size would make it easier to incorporate CT data while the radiobiological models may eventually permit improved assessment of the biological implications of treatment through normal tissue complication and tumor control probabilities. Currently these models are not included in this research due to the extra computational burden and because of the uncertainty regarding their accuracy.

Finally, while designed primarily for radiosurgery sized targets, it is possible, as shown in chapter 8, to also address standard radiotherapy cases. In fact, such cases should be easier to plan since such a dramatic dose gradient is not typically needed and the resolution of the intensity modulating device becomes less of a limiting factor. Also, the number of beams involved in such a plan would be much fewer than with radiosurgery thus resulting in much faster planning times.

APPENDIX
COMPUTER PROGRAMS

The following pages contain a complete listing of all programs described in previous chapters. The programs (DPBPROC, DPBVOX, SPECTRUM and PLAN) contain documentation throughout the listing for all major functions. All code was written in ANSI C and compiled with the ANSI C compiler on a SUN SPARCstation™ 10. All isosurface, contour, and image figures presented in this work were produced using the Open Windows™ based Advanced Visual Systems (AVS) software package. The AVS network created for this work is shown in figure 1.

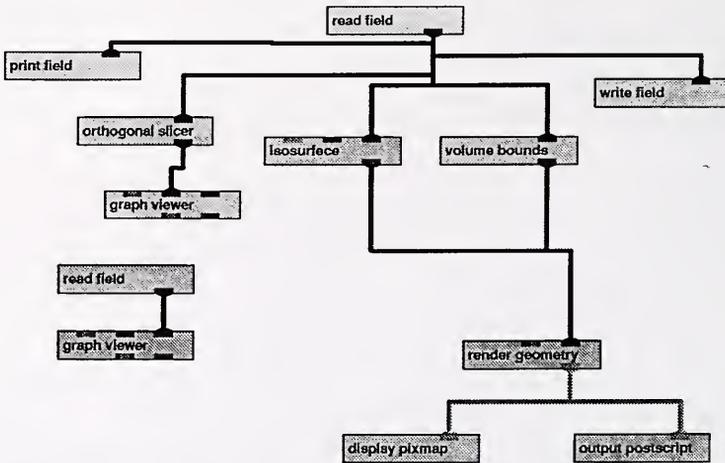


Figure 1. AVS network used for treatment plan evaluation.

```

/*****
This program, DPBPROC, processes the differential pencil beams
(DPB) received from Dr. Mohan via Michael Lovelock. Currently,
it first determines the energy for which each DPB was created.
This is accomplished by summing the product of energy deposited
per unit volume and voxel volume. It next sums the short and long
components to form a total DPB file. The DPBs are stored in
cylindrical coordinates. Thus, each voxel is actually an annulus.
The voxel binning is also logarithmic.
Last changed: 22 Feb 94
*****/

```

```

#include <stdlib.h>
#include <stdio.h>
#include <math.h>

#define MAX_RADII    99
#define MAX_DEPTHS  198

float dpb[40000];
float radii[100], depth[100], height[200];
float dpbsum[3], energy_s, energy_l, energy_tot, dist, vol,
dpbtot;
char infile[15], outfile[15];

void main(void)
{
    FILE *dpb_ptr;
    FILE *dpbtot_ptr;
    int j,r,d,c;
    long ctr1=0,ctr2=0;

/* This loop determines the radius and height associated with
each voxel */

    for (j=1;j<=MAX_RADII;j++)
        {
            radii[1]=0.01;
            depth[1]=0.01;
            dist=0.01*pow(1.1,j);
            radii[j+1]=dist;
            depth[j+1]=dist;
            height[99+j]=height[100-j]=depth[j]-depth[j-1];
        }

/* This section reads the DPB and determines the energy
associated with the short, long components
*/

    printf("Enter raw DPB filename (ex. 4mev.dat): \n");
    gets(infile);
    if ((dpb_ptr=fopen(infile, "r"))==NULL)
        {
            printf("*** Unable to open DPB file ***");

```

```

    exit(0);
}
for (c=0;c<=1;c++)
{
    for (r=1;r<=MAX_RADII;r++)
    {
        for (d=1;d<=MAX DEPTHS;d++)
        {
            vol=3.1415927*(radii[r]*radii[r]-radii[r-
            1]*radii[r-1])*height[d]; fscanf(dpb_ptr, "%f",
            &dpb[ctr1]); dpbsum[c]=dpbsum[c]+dpb[ctr1]*vol;
            ctr1=ctr1+1;
        }
    }
}
fclose(dpb_ptr);
energy_s=dpbsum[0]/1000000.0;
energy_l=dpbsum[1]/1000000.0;
printf("The short DPB energy is %10.3f MeV \n",
energy_s); printf("The long DPB energy is %10.3f MeV
\n", energy_l); printf("The total DPB energy is
%10.3f MeV \n", energy_s+energy_l);

/* This section creates a new DPB file that is the sum of the
short and long components
*/

    printf("Enter a filename for summed components DPB (ex.
4mev.sum): \n");

gets(outfile);
if ((dpbtot_ptr=fopen(outfile, "w"))==NULL)
{
    printf("*** Unable to open DPBTOT file ***");
    exit(0);
}
for(r=1;r<=MAX_RADII;r++)
{
    for(d=1;d<=MAX DEPTHS;d++)
    {
        vol=3.1415927*(radii[r]*radii[r]-radii[r-1]*radii[r-
        1])*height[d];
        dpbtot=dpbtot+(dpb[ctr2]+dpb[ctr2+19602])*vol;
        fprintf(dpbtot_ptr, "%10.3e",
        dpb[ctr2]+dpb[ctr2+19602]); ctr2=ctr2+1;
    }
}
fclose(dpbtot_ptr);
energy_tot=dpbtot/1000000.0;

printf("The summed component DPB energy is %10.3f MeV \n",
energy_tot);
return;
}

```

```

/*****
This file, DPBVOX, transforms the DPBs from a logarithmic
scaling to uniform scaling along the radial (X or Y
axis) and axial (Z axis) directions. The output file
containing the linearly scaled radially symmetric DPB is
in a format suitable for importing into a Quattro Pro
spreadsheet
Last modified: 22 Feb 1994.
*****/

```

```

#include <stdlib.h>
#include <stdio.h>
#include <math.h>

#define MAX_RADII    99
#define MAX_DEPTHS  198

void limits_01mm(void);
void limits_02mm(void);
void limits_05mm(void);
void limits_10mm(void);

float dpb[100][200];
float vox[101][101];
float
xmin[101],xmax[101],zmin[101],zmax[101],bigvox[
50][50]; float radii[101], depth[101],
height[200];
float dist,esum,vol,volsum,mev;
float voxsize;
int   rad_print,top_zprint,bot_zprint;
char  infile[15], outfile[15];

void main(void)
{
    FILE *dpb_ptr;
    FILE *out_ptr;

    int j,r,d,zvox,xvox;

    printf("Enter photon energy of DPB (in MeV):
");
    scanf ("%f",&mev);
    printf("\n");

    printf("Voxel dimension choices: 1, 2, 5, or 10
          mm. \n");
    printf("Enter desired voxel dimension: ");
    scanf("%f", &voxsize);
    printf("\n");

    if(voxsize == 1) {limits_01mm();}

```

```

if(voxsize == 2)
{limits_02mm();} if(voxsize
== 5) {limits_05mm();}
if(voxsize ==10)
{limits_10mm();}

/* Change units of voxsize from mm to cm.
*/

voxsize=voxsize/10;

/* This loop determines the radius and height, in cm,
associated
with each annulus shaped voxel
*/

for(j=1;j<=MAX_RADII;j++)
{
radii[0]=0.0;
radii[1]=0.01;
depth[0]=0.0;
depth[1]=0.01;
dist=0.01*pow(1.1,j);
radii[j+1]=dist;
depth[j+1]=dist;
height[99+j]=height[100-j]=depth[j]-depth[j-1];
}

/* This section reads the logarithmically scaled DPB and
normalizes all values to energy density (MeV/cm^3) per
incident photon. This is accomplished by dividing by the
number of incident photons (1E+06). */

if(mev==0.5)
{
if((dpb_ptr=fopen("0_5mev.sum", "r"))==NULL)
{
printf("*** Unable to open DPB file ***");
exit(0);
}
}

if(mev!=0.5)
{
sprintf(infile,"%dmev.sum", (int)mev);
if((dpb_ptr=fopen(infile, "r"))==NULL)
{
printf("*** Unable to open DPB file ***");
exit(0);
}
}

for(r=1;r<=MAX_RADII;r++)
{

```

```

    for(d=1;d<=MAX_DEPTHS;d++)
    {
        fscanf(dpb_ptr, "%f", &dpb[r][d]);
        dpb[r][d]=dpb[r][d]/1000000.0;
    }
    fclose(dpb_ptr);

/* This loop displays raw DPB data as a function of radial
position
                                for a selected depth.
                                */

/* d=71;
for(r=1;r<=MAX_RADII;r++)
{
    printf("%10d %10.3f %10.3f %10.7f \n",
        d, depth[d], radii[r], dpb[r][d+99]);
}
*/

/* This section determines the average energy (MeV)
assigned to each "voysize" cube along the X (or Y)
and Z axis ABOVE the interaction point. The limits
of each loop are obtained from the logarithmic
scaling of the original DPB and are contained in
limits_01mm(),
limits_02mm, limits_05mm or limits_10mm().
*/

printf("voysize=%10.2e \n", voysize);

    for(zvox=0;zvox<=20;zvox++)
    {
        for(xvox=0;xvox<=20;xvox++)
        {
            for(d=zmin[zvox];d<=zmax[zvox];d++)
            {
                for(r=xmin[xvox];r<=xmax[xvox];r++)
                {
                    vol=3.14159*(radii[r]*radii[r]-radii[r-
                        1]*radii[r-1])* height[100-d];
                    esum=esum+dpb[r][100-d]*vol;
                    volsum=volsum+vol;
                }
            }
            vox[zvox][xvox]=pow(voysize,3)*esum/volsum;
            esum=0.0;
            volsum=0.0;
        }
    }

/* This section prints the DPB in the X Z plane for
all voxels ABOVE the interaction point. The units

```

for each voxel are expressed in terms of fraction of incident photon energy. This is accomplished by dividing by the incident photon energy.
*/

```
    sprintf(outfile,"%dmev%dmm.dpb", (int)mev,
(int)(voysize*10));
```

```
    if ((out_ptr=fopen(outfile, "w"))==NULL)
    {
        printf("*** Unable to open file to create and write
***");
        exit(0);
    }
```

```
    printf("Enter number of radial values adjacent to origin to
output: ");
    scanf ("%d",&rad_print);
    fflush(stdin);
```

```
    printf("Enter number of axial values above origin to
output: ");
    scanf ("%d",&top_zprint);
    fflush(stdin);
```

```
    for(zvox=top_zprint-1;zvox>=0;zvox--)
    {
        for(xvox=0;xvox<=rad_print;xvox++)
        {
            fprintf(out_ptr,"%9.2e", vox[zvox][xvox]/mev);
        }
        fprintf(out_ptr,"\n");
    }
    fclose(out_ptr);
```

/* This section determines the average energy (MeV) assigned to each "voysize" cube along the X (or Y) and Z axis BELOW the interaction point. The limits of each loop are obtained from the logarithmic scaling of the original DPB and are contained in limits_01mm(), limits_02mm, limits_05mm or limits_10mm().
*/

```
    for(zvox=0;zvox<=20;zvox++)
    {
        for(xvox=0;xvox<=20;xvox++)
        {
            for(d=zmin[zvox];d<=zmax[zvox];d++)
            {
                for(r=xmin[xvox];r<=xmax[xvox];r++)
                {
                    vol=3.14159*(radii[r]*radii[r]-radii[r-1]*radii[r-1])* height[d+99];
                    esum=esum+dpb[r][d+99]*vol;
```

```

        volsum=volsum+vol;
    }
    vox[zvox][xvox]=pow(voxsize,3)*esum/volsum;
    esum=0.0;
    volsum=0.0;
}
}

/* This section prints the DPB in the X Z plane for
   all voxels BELOW the interaction point. The units
   for each voxel are expressed in terms of fraction
   of incident photon energy. This is accomplished
   by dividing by the incident photon energy.
*/

if ((out_ptr=fopen(outfile, "a"))==NULL)
{
    printf("*** Unable to open file to append ***");
    exit(0);
}

printf("Enter number of axial values below origin to
output: ");
scanf ("%d",&bot_zprint);
fflush(stdin);

for(zvox=0;zvox<=bot_zprint-1;zvox++)
{
    for(xvox=0;xvox<=rad_print;xvox++)
    {
        fprintf(out_ptr,"%9.2e", vox[zvox][xvox]/mev);
    }
    fprintf(out_ptr,"\n");
}
fclose(out_ptr);

return;
}

void limits_01mm(void)
{
    xmin[0]=1;      xmax[0]=18;      zmin[0]=1;
    zmax[0]=25;
    xmin[1]=19;     xmax[1]=30;      zmin[1]=26;
    zmax[1]=32;
    xmin[2]=31;     xmax[2]=35;      zmin[2]=33;
    zmax[2]=37;
    xmin[3]=36;     xmax[3]=38;      zmin[3]=38;
    zmax[3]=40;
    xmin[4]=39;     xmax[4]=41;      zmin[4]=41;
    zmax[4]=42;
    xmin[5]=42;     xmax[5]=43;      zmin[5]=43;
    zmax[5]=44;
}

```

```

    xmin[6]=44;      xmax[6]=45;      zmin[6]=45;
zmax[6]=46;      xmin[7]=46;      xmax[7]=46;      zmin[7]=47;
zmax[7]=47;      xmin[8]=47;      xmax[8]=48;      zmin[8]=48;
zmax[8]=48;      xmin[9]=49;      xmax[9]=49;      zmin[9]=49;
zmax[9]=49;      xmin[10]=50;     xmax[10]=50;     zmin[10]=50;
zmax[10]=50;     xmin[11]=51;     xmax[11]=51;     zmin[11]=51;
zmax[11]=51;     xmin[12]=52;     xmax[12]=52;     zmin[12]=52;
zmax[12]=52;     xmin[13]=53;     xmax[13]=53;     zmin[13]=53;
zmax[13]=53;     xmin[14]=53;     xmax[14]=53;     zmin[14]=54;
zmax[14]=54;     xmin[15]=54;     xmax[15]=54;     zmin[15]=54;
zmax[15]=54;     xmin[16]=55;     xmax[16]=55;     zmin[16]=55;
zmax[16]=55;     xmin[17]=55;     xmax[17]=55;     zmin[17]=56;
zmax[17]=56;     xmin[18]=56;     xmax[18]=56;     zmin[18]=56;
zmax[18]=56;     xmin[19]=57;     xmax[19]=57;     zmin[19]=57;
zmax[19]=57;     xmin[20]=57;     xmax[20]=57;     zmin[20]=57;
zmax[20]=57;
}

```

```
void limits_02mm(void)
```

```

{
    xmin[0]=1;      xmax[0]=25;      min[0]=1;
    zmax[0]=32;
    xmin[1]=26;     xmax[1]=37;      zmin[1]=33;
    zmax[1]=40;
    xmin[2]=38;     xmax[2]=43;      zmin[2]=41;
    zmax[2]=44;
    xmin[3]=44;     xmax[3]=46;      zmin[3]=45;
    zmax[3]=47;
    xmin[4]=47;     xmax[4]=48;      zmin[4]=48;
    zmax[4]=49;
    xmin[5]=49;     xmax[5]=50;      zmin[5]=50;
    zmax[5]=51;
    xmin[6]=51;     xmax[6]=52;      zmin[6]=52;
    zmax[6]=53;
    xmin[7]=53;     xmax[7]=54;      zmin[7]=54;
    zmax[7]=54;
    xmin[8]=55;     xmax[8]=55;      zmin[8]=55;
    zmax[8]=55;
    xmin[9]=56;     xmax[9]=56;      zmin[9]=56;
    zmax[9]=57;
}

```

```

xmin[10]=57;      xmax[10]=57;      zmin[10]=58;
zmax[10]=58;
xmin[11]=58;      xmax[11]=58;      zmin[11]=59;
zmax[11]=59;
xmin[12]=59;      xmax[12]=59;      zmin[12]=60;
zmax[12]=60;
xmin[13]=60;      xmax[13]=60;      zmin[13]=60;
zmax[13]=60;
xmin[14]=61;      xmax[14]=61;      zmin[14]=61;
zmax[14]=61;
xmin[15]=61;      xmax[15]=61;      zmin[15]=62;
zmax[15]=62;
xmin[16]=62;      xmax[16]=62;      zmin[16]=62;
zmax[16]=62;
xmin[17]=62;      xmax[17]=62;      zmin[17]=63;
zmax[17]=63;
xmin[18]=63;      xmax[18]=63;      zmin[18]=64;
zmax[18]=64;
xmin[19]=64;      xmax[19]=64;      zmin[19]=64;
zmax[19]=64;
xmin[20]=64;      xmax[20]=64;      zmin[20]=65;
zmax[20]=65;
}

```

```
void limits_05mm(void)
```

```

{
xmin[0]=1;      xmax[0]=35;      zmin[0]=1;
zmax[0]=42;
xmin[1]=36;      xmax[1]=46;      zmin[1]=43;
zmax[1]=49;
xmin[2]=47;      xmax[2]=52;      zmin[2]=50;
zmax[2]=54;
xmin[3]=53;      xmax[3]=55;      zmin[3]=55;
zmax[3]=57;
xmin[4]=56;      xmax[4]=58;      zmin[4]=58;
zmax[4]=59;
xmin[5]=59;      xmax[5]=60;      zmin[5]=60;
zmax[5]=61;
xmin[6]=61;      xmax[6]=61;      zmin[6]=62;
zmax[6]=62;
xmin[7]=62;      xmax[7]=63;      zmin[7]=63;
zmax[7]=64;
xmin[8]=64;      xmax[8]=64;      zmin[8]=65;
zmax[8]=65;
xmin[9]=65;      xmax[9]=65;      zmin[9]=66;
zmax[9]=66;
xmin[10]=66;     xmax[10]=66;     zmin[10]=67;
zmax[10]=67;
xmin[11]=67;     xmax[11]=67;     zmin[11]=68;
zmax[11]=68;
xmin[12]=68;     xmax[12]=68;     zmin[12]=69;
zmax[12]=69;
xmin[13]=69;     xmax[13]=69;     zmin[13]=70;
zmax[13]=70;
}

```

```

xmin[14]=70;      xmax[14]=70;      zmin[14]=71;
zmax[14]=71;      xmax[15]=71;      zmin[15]=71;
xmin[15]=71;      xmax[16]=72;      zmin[16]=72;
zmax[15]=71;      xmax[17]=72;      zmin[17]=73;
xmin[16]=72;      xmax[18]=73;      zmin[18]=73;
zmax[16]=72;      xmax[19]=73;      zmin[19]=74;
xmin[17]=72;      xmax[20]=74;      zmin[20]=74;
zmax[17]=73;
xmin[18]=73;
zmax[18]=73;
xmin[19]=73;
zmax[19]=74;
xmin[20]=74;
zmax[20]=74;
}

```

```
void limits_10mm(void)
```

```

{
xmin[0]=1;      xmax[0]=42;      zmin[0]=1;
zmax[0]=49;      xmax[1]=54;      zmin[1]=50;
xmin[1]=43;      xmax[2]=59;      zmin[2]=58;
zmax[1]=57;      xmax[3]=62;      zmin[3]=62;
xmin[2]=55;      xmax[4]=65;      zmin[4]=65;
zmax[2]=61;      xmax[5]=67;      zmin[5]=67;
xmin[3]=60;      xmax[6]=69;      zmin[6]=69;
zmax[3]=64;      xmax[7]=70;      zmin[7]=71;
xmin[4]=63;      xmax[8]=72;      zmin[8]=72;
zmax[4]=66;      xmax[9]=73;      zmin[9]=73;
xmin[5]=66;      xmax[10]=74;      zmin[10]=74;
zmax[5]=68;      xmax[11]=75;      zmin[11]=74;
xmin[6]=68;      xmax[12]=76;      zmin[12]=75;
zmax[6]=70;      xmax[13]=77;      zmin[13]=76;
xmin[7]=70;      xmax[14]=78;      zmin[14]=77;
zmax[7]=71;      xmax[15]=78;      zmin[15]=78;
xmin[8]=71;      xmax[16]=79;      zmin[16]=79;
zmax[8]=72;      xmax[17]=80;      zmin[17]=79;
xmin[9]=73;
zmax[9]=73;
xmin[10]=73;
zmax[10]=74;
xmin[11]=74;
zmax[11]=75;
xmin[12]=75;
zmax[12]=76;
xmin[13]=76;
zmax[13]=77;
xmin[14]=77;
zmax[14]=78;
xmin[15]=78;
zmax[15]=79;
xmin[16]=78;
zmax[16]=79;
xmin[17]=79;
zmax[17]=80;
}

```

```
xmin[18]=80;      xmax[18]=80;      zmin[18]=80;
zmax[18]=80;
xmin[19]=80;      xmax[19]=81;      zmin[19]=80;
zmax[19]=81;
xmin[20]=81;      xmax[20]=81;      zmin[20]=81;
zmax[20]=82;
}
```

```

/*****
This program, SPECTRUM, determines the effective beam
spectrum of a clinical linac. This is accomplished by
fitting monoenergetic depth dose data to the depth dose data
of a polyenergetic photon beam. The weighting of each energy
bin is found through a simulated annealing version of the
downhill simplex optimization method.
Last Modified: 22 Feb 94
*****/

```

```

Last Modified: 22 Feb 94

```

```

*****/

```

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

```

```

#define NR_END 1
#define FREE_ARG char*
#define IA 16807
#define IM 2147483647
#define AM (1.0/IM)
#define IQ 127773
#define IR 2836
#define NTAB 32
#define NDIIV (1+(IM-1)/NTAB)
#define EPS 1.2e-7
#define RNMIX (1.0-EPS)
#define GET_PSUM \

```

```

    for (n=1;n<=ndim;n++) {\
        for (sum=0.0,m=1;m<=mpts;m++) sum +=
p[m][n];\
        psum[n]=sum;}

```

```

void inreal(void);
void indose(void);
void optimize(void);
float funk(float *x);
void ambsa(float **p, float y[], int ndim, float pb[],
float *yb, float ftol,
float (*funk)(float []), int *iter, float temptr);
float amotsa(float **p, float y[], float psum[], int ndim,
float pb[],
float *yb, float (*funk)(float []), int ihi, float
*yi,
float fac);
float ran1(long *idum);
void nrerror(char error_text[]);
float *vector(long nl, long nh);
float **matrix(long nrl, long nrh, long ncl, long nch);
void free_vector(float *v, long nl, long nh);
void free_matrix(float **m, long nrl, long nrh, long ncl,
long nch);
void specout(void);

```

```

float  real[100], dose[100][10], finalx[10], normx[10],
esum[100][10];
float  modal_objfunc[10];
int    bin, modal_bin, best_modal, binmax=7, depths,
top_depth, fs;
long   idum=(-64);
float  tt, best_objfunc=1e+30, voxdim;

void main(void)
{

/*          Read measured depth dose file
*/

    inreal();

/*          Read dose per energy fluence files for each energy
bin          */

    indose();

/*          Perform optimization for all reasonable modal
values          */

    for(modal_bin=1;modal_bin<=4;modal_bin++)
    {

/*          Optimize fit of calculated to measured data
*/

        optimize();

        if(modal_objfunc[modal_bin]<best_objfunc)
        {
            best_objfunc=modal_objfunc[modal_bin];
            best_modal=modal_bin;
        }
    }

/*          Output the optimum "effective spectrum"
*/

    specout();

    return;
}

/*****
*****
This function, INREAL(), inputs the real (measured) % depth
dose from a file.
*****
*****/

```

```

void inreal(void)
{
    FILE *inreal_ptr;
    int i;
    char inrealfile[100];

    printf("Enter square calculation voxel size in cm (0.5 or
1.0): ");
    scanf("%f", &voxdim);
    printf("\n");

    printf("Enter desired square field size (e.g. 5 for a 5x5
cm2): ");
    scanf("%d", &fs);
    printf("\n");

    sprintf(inrealfile,"6mv%dfs.d%d", fs, (int)(10*voxdim));

    if(voxdim==0.5)
    {
        top_depth=5;
        depths=45;
    }

    if(voxdim==1.0)
    {
        top_depth=3;
        depths=30;
    }

    printf("Reading 6 MV data...");

    if ((inreal_ptr=fopen(inrealfile, "r"))==NULL)
    {
        printf("*** UNABLE TO OPEN 6MVREAL FILE *** \n");
        exit(0);
    }
    for(i=1;i<=depths;i++)
    {
        fscanf(inreal_ptr, "%f", &real[i]);
    }
    fclose(inreal_ptr);

    printf("Done. \n");

    return;
}

```

```

/*****
*****

```

This file, PINDOSE.CPP, inputs the dose per energy fluence data by reading a file for each energy bin being used for constructing the "effective

```

spectrum".
*****
*****/

void indose(void)
{
    FILE *indose0_ptr;
    FILE *indose1_ptr;
    FILE *indose2_ptr;
    FILE *indose3_ptr;
    FILE *indose4_ptr;
    FILE *indose5_ptr;
    FILE *indose6_ptr;

    int i;

    printf("Reading monoenergetic data sets...");

    if ((indose0_ptr=fopen("dose0.dat", "r"))==NULL)
    {
        printf("*** UNABLE TO OPEN DOSE0 FILE *** \n");
        exit(0);
    }
    for(i=1;i<=depths;i++)
    {
        fscanf(indose0_ptr, "%f", &dose[i][1]);
    }
    fclose(indose0_ptr);

    if ((indose1_ptr=fopen("dose1.dat", "r"))==NULL)
    {
        printf("*** UNABLE TO OPEN DOSE1 FILE *** \n");
        exit(0);
    }
    for(i=1;i<=depths;i++)
    {
        fscanf(indose1_ptr, "%f", &dose[i][2]);
    }
    fclose(indose1_ptr);

    if ((indose2_ptr=fopen("dose2.dat", "r"))==NULL)
    {
        printf("*** UNABLE TO OPEN DOSE2 FILE *** \n");
        exit(0);
    }
    for(i=1;i<=depths;i++)
    {
        fscanf(indose2_ptr, "%f", &dose[i][3]);
    }
    fclose(indose2_ptr);

    if ((indose3_ptr=fopen("dose3.dat", "r"))==NULL)
    {

```

```

    printf("*** UNABLE TO OPEN DOSE3 FILE *** \n");
    exit(0);
}
for(i=1;i<=depths;i++)
{
    fscanf(indose3_ptr, "%f", &dose[i][4]);
}
fclose(indose3_ptr);

if ((indose4_ptr=fopen("dose4.dat", "r"))==NULL)
{
    printf("*** UNABLE TO OPEN DOSE4 FILE *** \n");
    exit(0);
}
for(i=1;i<=depths;i++)
{
    fscanf(indose4_ptr, "%f", &dose[i][5]);
}
fclose(indose4_ptr);

if ((indose5_ptr=fopen("dose5.dat", "r"))==NULL)
{
    printf("*** UNABLE TO OPEN DOSE5 FILE *** \n");
    exit(0);
}
for(i=1;i<=depths;i++)
{
    fscanf(indose5_ptr, "%f", &dose[i][6]);
}
fclose(indose5_ptr);

if ((indose6_ptr=fopen("dose6.dat", "r"))==NULL)
{
    printf("*** UNABLE TO OPEN DOSE6 FILE *** \n");
    exit(0);
}
for(i=1;i<=depths;i++)
{
    fscanf(indose6_ptr, "%f", &dose[i][7]);
}
fclose(indose6_ptr);

printf("Done. \n");

return;
}

```

```

/*****
*****
This function, OPTIMIZE(), contains the starting values and
annealing
schedule required by AMEBSA().
*****
*****/

```

```

void optimize(void)
{
    float ftol=1.0E-6;
    int i,iiter,iter,j,jiter,nit;
    float temptr,yb,ybb;
    float **p,*x,*y,*pb;
    int ndim=binmax;
    static float xoff[]={0, 3.0, 2.5, 2.0, 1.5, 1.0, 0.5,
0.0};

    p=matrix(1,ndim+1,1,ndim);
    x=vector(1,ndim);
    y=vector(1,ndim+1);
    pb=vector(1,ndim);

    if(modal_bin==1)
    {
        printf("\n");
        printf("Input temp, iiter:");
        scanf("%f %d",&temptr,&iiter);
        printf("\n");
    }

    printf("Considering bin#:%d as modal energy...",
modal_bin);

    for(i=1;i<=ndim+1;i++)
    {
        for(j=1;j<=ndim;j++)
        {
            p[i][j]=0.0;
        }
    }

    for(j=2;j<=ndim+1;j++)
    {
        p[j][j-1]=0.5;
    }
    for(i=1;i<=ndim+1;i++)
    {
        for(j=1;j<=ndim;j++)
        {
            x[j]=(p[i][j] = p[i][j] + xoff[j]);
        }
        y[i]=funkt(x);
    }
    yb=1.0e30;
    ybb=1.0e30;
    nit=0;
    for(jiter=1;jiter<=100;jiter++)
    {
        iiter=iiter;
        temptr *= 0.9;
    }
}

```

```

amebsa(p,y,ndim,pb,&yb,ftol,funk,&iter,temptr);
nit += iiter-iter;
if(yb < ybb)
  {
  ybb=yb;
  for(j=1;j<=ndim;j++)
    {
    printf("%7.4e ",pb[j]);
    }
  }
if(iter > 0) break;
}

printf("Objective function =%10.3e\n\n",yb);
printf(" 0.5 MeV    1 MeV    2 MeV    3 MeV    4
5 MeV    6 MeV\n");
for(j=1;j<=ndim;j++)
  {
  printf("%10.3e",pb[j]);
  }
printf("\n\n");

modal_objfunc[modal_bin]=yb;

free_vector(pb,1,ndim);
free_vector(y,1,ndim+1);
free_vector(x,1,ndim);
free_matrix(p,1,ndim+1,1,ndim);
return;
}
/*****
*****
This function, AMEBSA(), performs a multidimensional
optimization of the
function FUNK through a combination of simulated annealing
and downhill
simplex.
*****
*****/

void ametsa(float **p, float y[], int ndim, float pb[],
float *yb, float ftol,
float (*funk)(float []), int *iter, float temptr)
{
float amotsa(float **p, float y[], float psum[], int
ndim, float pb[],
float *yb, float (*funk)(float []), int ihi,
float *yhi,
float fac);
float ran1(long *idum);
int i,ih,i,ilo,j,m,n,mpts=ndim+1;
float rtol,sum,swap,yhi,ylo,ynhi,ysave,yt,ytry,*psum;

psum=vector(1,ndim);

```

```

tt = -temptr;
GET_PSUM
for ( ;; ) {
    ilo=1;
    ihi=2;
    ynhi=ylo=y[1]+tt*log(ran1(&idum));
    yhi=y[2]+tt*log(ran1(&idum));
    if (ylo > yhi) {
        ihi=1;
        ilo=2;
        ynhi=yhi;
        yhi=ylo;
        ylo=ynhi;
    }
    for (i=3;i<=mpts;i++) {
        yt=y[i]+tt*log(ran1(&idum));
        if (yt <= ylo) {
            ilo=i;
            ylo=yt;
        }
        if (yt > yhi) {
            ynhi=yhi;
            ihi=i;
            yhi=yt;
        } else if (yt > ynhi) {
            ynhi=yt;
        }
    }
    rtol=2.0*fabs(yhi-ylo)/(fabs(yhi)+fabs(ylo));
    if (rtol < ftol || *iter < 0) {
        swap=y[1];
        y[1]=y[ilo];
        y[ilo]=swap;
        for (n=1;n<=ndim;n++) {
            swap=p[1][n];
            p[1][n]=p[ilo][n];
            p[ilo][n]=swap;
        }
        break;
    }
    *iter -- 2;
    ytry=amotsa(p,y,psum,ndim,pb,yb,funk,ihl,&yhi,-
1.0);
    if (ytry <= ylo) {

ytry=amotsa(p,y,psum,ndim,pb,yb,funk,ihl,&yhi,2.0);
    } else if (ytry >= ynhi) {
        ysave=yhi;

ytry=amotsa(p,y,psum,ndim,pb,yb,funk,ihl,&yhi,0.5);
    if (ytry >= ysave) {
        for (i=1;i<=mpts;i++) {
            if (i != ilo) {
                for (j=1;j<=ndim;j++) {

```

```

psum[j]=0.5*(p[i][j]+p[i1o][j]);
                    p[i][j]=psum[j];
                    }
                    y[i]=(*funk)(psum);
                }
            }
            *iter -= ndim;
            GET_PSUM
        }
    } else ++(*iter);
}
free_vector(psum,1,ndim);
}

/*****
*****
This function, AMOTSA(), is called by AMEBSA().
*****
*****/

float amotsa(float **p, float y[], float psum[], int ndim,
float pb[],
float *yb, float (*funk)(float []), int ihi, float
*yhi,
float fac)
{
float ran1(long *idum);
int j;
float fac1,fac2,yflu,ytry,*ptry;

ptry=vector(1,ndim);
fac1=(1.0-fac)/ndim;
fac2=fac1-fac;
for (j=1;j<=ndim;j++)
ptry[j]=psum[j]*fac1-p[ihi][j]*fac2;
ytry=(*funk)(ptry);
if (ytry <= *yb) {
for (j=1;j<=ndim;j++) pb[j]=ptry[j];
*yb=ytry;
}
yflu=ytry-tt*log(ran1(&idum));
if (yflu < *yhi) {
y[ihi]=ytry;
*yhi=yflu;
for (j=1;j<=ndim;j++) {
psum[j] += ptry[j]-p[ihi][j];
p[ihi][j]=ptry[j];
}
}
free_vector(ptry,1,ndim);
return yflu;
}
}

```

```

/*****
*****
This function, RAN1(), returns a uniform random deviate
between 0.0 and 1.0
(exclusive of endpoint values). It uses idum to initialize.
*****
*****/

```

```

float ran1(long *idum)
{
    int j;
    long k;
    static long iy=0;
    static long iv[NTAB];
    float temp;

    if (*idum <= 0 || !iy) {
        if (-(*idum) < 1) *idum=1;
        else *idum = -(*idum);
        for (j=NTAB+7;j>=0;j--) {
            k=(*idum)/IQ;
            *idum=IA*(*idum-k*IQ)-IR*k;
            if (*idum < 0) *idum += IM;
            if (j < NTAB) iv[j] = *idum;
        }
        iy=iv[0];
    }
    k=(*idum)/IQ;
    *idum=IA*(*idum-k*IQ)-IR*k;
    if (*idum < 0) *idum += IM;
    j=iy/NDIV;
    iy=iv[j];
    iv[j] = *idum;
    if ((temp=AM*iy) > RNMX) return RNMX;
    else return temp;
}

```

```

/*****
*****
The following functions are utilities needed by AMEBSA() and
AMOTRY().
*****
*****/

```

```

void nrerror(char error_text[])
/* Numerical Recipes standard error handler */
{
    fprintf(stderr,"4numerical Recipes run-time
error...\n");
    fprintf(stderr,"%s\n",error_text);
    fprintf(stderr,"...now exiting to system...\n");
    exit(1);
}

float *vector(long nl, long nh)

```

```

/* allocate a float vector with subscript range v[nl..nh] */
{
    float *v;

    v=(float *)malloc((size_t) ((nh-
nl+1+NR_END)*sizeof(float)));
    if (!v) nrerror("allocation failure in vector()");
    return v-nl+NR_END;
}

float **matrix(long nrl, long nrh, long ncl, long nch)
/* allocate a float matrix with subscript range
m[nrl..nrh][ncl..nch] */
{
    long i, nrow=nrh-nrl+1,ncol=nch-ncl+1;
    float **m;

    /* allocate pointers to rows */
    m=(float **)
malloc((size_t)((nrow+NR_END)*sizeof(float*)));
    if (!m) nrerror("allocation failure 1 in matrix()");
    m += NR_END;
    m -= nrl;

    /* allocate rows and set pointers to them */
    m[nrl]=(float *)
malloc((size_t)((nrow*ncol+NR_END)*sizeof(float)));
    if (!m[nrl]) nrerror("allocation failure 2 in
matrix()");
    m[nrl] += NR_END;
    m[nrl] -= ncl;

    for(i=nrl+1;i<=nrh;i++) m[i]=m[i-1]+ncol;

    /* return pointer to array of pointers to rows */
    return m;
}

void free_vector(float *v, long nl, long nh)
/* free a float vector allocated with vector() */
{
    free((FREE_ARG) (v+nl-NR_END));
}

void free_matrix(float **m, long nrl, long nrh, long ncl,
long nch)
/* free a float matrix allocated by matrix() */
{
    free((FREE_ARG) (m[nrl]+ncl-NR_END));
    free((FREE_ARG) (m+nrl-NR_END));
}

```

```

/*****
*****
This function, FUNK(), is the objective function called by
OPTIMUM(). It
represents the wellness of fit between the real (measured) %
depth dose and
that produced by the energy deposition kernel dosimetry
algorithm using an
"effective" spectrum.
*****
*****/

```

```

float funk(float *x)
{
    register int i, z;
    double      objfunc=0.0, neg_penalty=1.0,
modal_penalty=1.0;
    float      sumx=0.0;

    for(i=0;i<100;i++)
        {
            esum[i][modal_bin]=0;
        }

    for (bin=1;bin<=binmax;bin++)
        {
            if(x[bin]<0)
                {
                    neg_penalty=1000.0;
                }
        }

    for(bin=1;bin<=binmax;bin++)
        {
            if(bin<modal_bin)
                {
                    if(x[bin]>x[bin+1]) modal_penalty=1000.0;
                }
            if(bin>modal_bin)
                {
                    if(x[bin]>x[bin-1]) modal_penalty=1000.0;
                }
        }

    for (z=top_depth;z<=depths;z++)
        {
            for (bin=1;bin<=binmax;bin++)
                {
                    esum[z][modal_bin] += dose[z][bin]*x[bin]*1e+09;
                }
            objfunc=objfunc+fabs((double)esum[z][modal_bin]-
(double)real[z]);
        }
}

```

```

objfunc=objfunc*modal_penalty*neg_penalty/(depths-
top_depth+1);

return objfunc;
}

/*****
*****
This function, SPECOUT(), displays the optimum energy
fluence value assigned to
each energy bin as well as the calculated and measured depth
dose data.
*****/
*****/

void specout(void)
{
FILE *specout_ptr;
register int z;

printf("Best modal bin = %d \n", best_modal);
printf("Best objective function value = %10.3e \n",
best_objfunc);

specout_ptr=fopen("SPECFIT.DAT", "w");

for (z=top_depth;z<=depths;z++)
{
fprintf(specout_ptr,"%e,%e \n",
esum[z][best_modal],real[z]);
}

fclose(specout_ptr);

return;
}

```

```

/*****
*****

```

This program, PLAN, produces an optimized 3-D treatment plan and resulting dose distribution. Given a desired dose distribution, this program determines the intensity modulation required for any beam's eye view (BEV) of the target and the optimal beam weighting. After planning, a final dose distribution is generated for comparing with the prescription.

Last modified: 11 Apr 94.

```

*****
*****/

```

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

```

```

#define NR_END 1
#define FREE_ARG char*
#define IA 16807
#define IM 2147483647
#define AM (1.0/IM)
#define IQ 127773
#define IR 2836
#define NTAB 32
#define NDIV (1+(IM-1)/NTAB)
#define EPS 1.2e-7
#define RNMX (1.0-EPS)
#define GET_PSUM \
    for (n=1;n<=ndim;n++) {\
        for (sum=0.0,m=1;m<=mpts;m++) sum +=
p[m][n];\
        psum[n]=sum;}

```

```

#define SWAP(a,b) tempr=(a);(a)=(b);(b)=tempr

```

```

void inkernel(void);
void select(float kernel[]);
void indose();
void rotate_rx(double table,double gantry,float rx_dose[]);
void deconv3d(float rx_dose[],float kernel[],float
deconv[]);
void projection(float deconv[]);
void outthickness(void);
void terma3d(float terma[]);
void ndimfft(float data[],int isign);
void conv3d(float kernel[],float terma[],float conv[]);
void mono_sum(float mono_dose[]);
void rotate_dose(double table,double gantry);
void outdose(void);
void optimize(void);
float funk(float *bev_mu);

```

```

void amebesa(float **p, float y[], int ndim, float pb[],
float *yb, float ftol,
        float (*funk)(float []), int *iter, float temptr);
float amotsa(float **p, float y[], float psum[], int ndim,
float pb[],
        float *yb, float (*funk)(float []), int ihi, float
*yni,
        float fac);
float ran1(long *idum);
void nrerror(char error_text[]);
float *vector(long nl, long nh);
float **matrix(long nrl, long nrh, long ncl, long nch);
void free_vector(float *v, long nl, long nh);
void free_matrix(float **m, long nrl, long nrh, long ncl,
long nch);
void evaluate(void);
void outfina1(void);

float dpb[7][132000];
float raw_crit_struct[132000];
float crit_struct[51][132000];
float raw_rx_dose[132000];
float ctrx[132000];
float ctry[132000];
float ctrz[132000];
double thickness[50][50];
double sum_thickness[50][50];
double ave_thickness[50][50];
double sum_ctr[50][50];
float sum_dose[132000];
float dose[51][132000];
float mono_sum_dose[132000];
float unif_raw_rx_dose;

int ndim=3, view=0, ntot=1, iso, view_max;
int total_num, normal_num, crit_num=0, target_num=0;
double table, gantry;
long idum=(-64);
float tt, mu_est, ssd;

int table_start= -60; /* These variables contain the
desired table and */
int table_end= 60; /* gantry positions (in degrees)
for each beam's */
int table_delta= 30; /* eye view (BEV) of the target
tissue. */
int gantry_start= 40; /*
*/
int gantry_end= 140; /*
*/
int gantry_delta= 20; /*
*/
double eflu_ref=5.56e+09; /* reference energy fluence
(MeV/(cm^2 mu)) */

```

```

double refdist=      65.0; /* Source to middle of attenuator
dist (cm)           */
float  sq_field=     5.0; /* square field size at SSD (cm)
*/
float  grid=        0.3; /* Dimension of lead attenuator
voxel (cm)         */
float  voxdim=      0.2; /* Dimension of cubical
calculation voxel (cm) */
float  nn[]={0,64,32,32}; /* Dimensions for NULL, Z, Y, and
X respectively */
int    e=            2; /* Photon energy used to create
BEV IMF files      */
float  target_wt=   1.0; /* Tissue weighting factors
(0.....1)         */
float  normal_wt=   0.9; /*
*/
float  critical_wt= 1.0; /*
*/

/*Normalized energy fluence weighting for 0.5,1,2,3,4,5,and
6 MeV photons */

float e_wt[]={0.0311, 0.1615, 0.1615, 0.1615, 0.1615,
0.1615, 0.1615};

void main (void)
{
float rx_dose[132000];
float deconv[132000];
float kernel[132000];
float terma[132000];
float conv[132000];
int  isign, i;

for(i=1;i<=ndim;i++) {ntot *= nn[i];}

/*          First find the number of dimensions (BEV'S)
*/

view_max=(((table_end-table_start)/table_delta)+1)*
          (((gantry_end-gantry_start)/gantry_delta)+1);
printf("view_max=%d \n", view_max);

/*          Read all energy deposition kernel files
*/

for(e=0;e<=6;e++) {inkernel();}

/*          Select kernel for use in solving for IMF
*/

e=2;
select(kernel);

```

```

/*          Obtain FFT of energy deposition kernel
*/

    isign=1;
    ndimfft(kernel, isign);

/*          Read prescribed dose distribution file
*/

    indose();

/*          Create an IMF file for each BEV of the target
tissue          */

    view=0;

for(table=table_start;table<=table_end;table=table+table_del
ta)
    {

for(gantry=gantry_start;gantry<=gantry_end;gantry=gantry+gan
try_delta)
    {
        view=view+1;

/*          Rotate the prescribed rx_dose
distribution          */

        rotate_rx(table, gantry, rx_dose);

/*          Obtain FFT of rx_dose distribution
*/

        isign=1;
        ndimfft(rx_dose, isign);

/*          deconv[]=FFT(rx_dose[])/FFT(kernel[])
*/

        deconv3d(rx_dose, kernel, deconv);

        isign=-1;
        ndimfft(rx_dose, isign);

/*          Obtain inverse FFT of deconv[]
*/

        isign=-1;
        ndimfft(deconv, isign);

/*          Project 3-D onto 2-D surface
*/

        projection(deconv);

```

```

/*                                Output IMF results
*/

    outthickness();

    printf("IMF file for Table=%3.0f Gantry=%3.0f
created \n",
           table, gantry);
    }
}

/*    Create polyenergetic dose distribution for each BEV of
the target    */

    view=0;

for(table=table_start;table<=table_end;table=table+table_delta)
    {

for(gantry=gantry_start;gantry<=gantry_end;gantry=gantry+gantry_delta)
    {
        printf("Creating dose distribution for Table=%3.0f
Gantry=%3.0f",    table, gantry);

        view=view+1;
        for(e=0;e<=6;e++)
            {
                printf(".");

/*                                Select kernel for current energy
*/

                select(kernel);

/*                                Obtain FFT of energy deposition kernel
*/

                isign=1;
                ndimfft(kernel, isign);

/*                                Create monoenergetic TERMA distribution
*/

                terma3d(terma);

/*                                Obtain FFT of TERMA distribution
*/

                isign=1;
                ndimfft(terma, isign);

```

```

/*                                     conv[]=FFT(terma[])*FFT(kernel[])
*/

        conv3d(kernel,terma,conv);
        isign=-1;
        ndimfft(terma,isign);

/*      Inverse FFT of conv to find monoenergetic dose
distribution      */

        isign=-1;
        ndimfft(conv,isign);

/*      Sum monoenergetic results to obtain polyenergetic dose
distribution      */

        mono_sum(conv);
    }

/*                                     Rotate polyenergetic dosimetry results
*/

        rotate_dose(-table,-gantry);

/*                                     Output single beam dosimetry results
*/

        outdose();
        printf("Done \n");
    }

/*                                     Optimize the beam weighting for each BEV
*/

    for(;;)
    {
        printf("Beginning new optimization. Enter 1 to start,
-1 to end:");
        scanf("%d",&i);
        if(i==1) break;
        optimize();
        evaluate();
    }

/*      Output final beam weighted dosimetry
distribution      */

    printf("Optimized dose distribution will be saved to file
\n");
    outfinal();

    return;
}

```

```

/*****
*****
This function, INKERNEL(), recieves an empty array called
kernel[] and
fills it with a monoenergetic energy deposition kernel.
*****
*****/

```

```

void inkernel(void)
{
    FILE          *inkernel_ptr;
    register int  i, j;
    char          infile[15];

    sprintf(infile, "%e323264.k%d", e, (int)(10*voxdim));
    printf("Reading kernel: %s \n", infile);

    if ((inkernel_ptr=fopen(infile, "r"))==NULL)
    {
        printf("*** UNABLE TO READ %d MeV DPB FILE ***\n", e);
        exit(0);
    }

    for(j=1; j<=2*ntot-1; j=j+2)
    {
        fscanf(inkernel_ptr, "%f", &dpb[e][j]);
        dpb[e][j+1]=0.0;
    }
    fclose(inkernel_ptr);
}

```

```

/*****
*****
This function, SELECT(), puts the DPB of choice into an
array called
kernel[]. The choice is made by the current value of photon
energy, e.
*****
*****/

```

```

void select(float kernel[])
{
    register int ctr;

    for(ctr=1; ctr<=2*ntot-1; ctr=ctr+2)
    {
        kernel[ctr]=dpb[e][ctr];
        kernel[ctr+1]=0;
    }
    return;
}

```

```

/*****
*****
This function, INDOSE(), fills an array called raw_rx_dose[]
with the three
dimensional prescribed rx_dose distribution (Gy). This array
contains an
isocenter voxel and critical structure voxels (if any) as
well as the
desired rx_dose to the target voxels. The "raw" term
indicates the dose
prescription has not yet been rotated for a particular BEV.
*****/
*****/

```

```

void indose(void)
{
    FILE *rx_ptr;
    int i,j;
    float max_raw_rx_dose=0;
    char rx_file[25];

/* Read file containing 3-D prescribed rx_dose distribution
in units of Gy. */

    printf("Enter prescribed rx_dose filename (XXX.rx) : ");
    gets(rx_file);
    printf("\n");

    if ((rx_ptr=fopen(rx_file, "r"))==NULL)
        {
            printf("*** UNABLE TO READ RX FILE ***\n");
            exit(0);
        }

    for(j=1;j<=2*ntot-1;j=j+2)
        {
            fscanf(rx_ptr, "%f", &raw_rx_dose[j]);

/* Determine the number of voxels comprising the target
tissue */

            if(raw_rx_dose[j]!=0 && raw_rx_dose[j]!=1)
                {
                    target_num=target_num+1;
                }

/* Determine the number of voxels comprising normal tissue
*/

            if(raw_rx_dose[j]==0)
                {
                    normal_num=normal_num+1;
                }

```

```

/* Test the .rx file for presence of critical structures. If
voxel is
    within critical structure, remove the "1" identifier from
raw_rx_dose[]
    and place it in a crit_struct[] array.
*/

    if(raw_rx_dose[j]==1)
    {
        crit_num=crit_num+1;
        raw_rx_dose[j]=0;
        raw_crit_struct[j]=1;
        raw_crit_struct[j+1]=0;
    }

/* Find the uniform rx_dose prescription value among the
zero background by
    searching for the largest value in raw_rx_dose[].
*/

    if(raw_rx_dose[j]>max_raw_rx_dose)
    {
        max_raw_rx_dose=raw_rx_dose[j];
    }
/* printf("crit=%d max_raw_rx_dose=%f \n",j,max_raw_rx_dose);
*/

/* Test the .rx file for presence of a voxel tagged as the
isocenter. It
    is identified as the voxel other than 0 (background), 1
(critical
    structure), or unif_raw_rx_dose.
*/

    if(raw_rx_dose[j]!=0 && raw_rx_dose[j]!=1 &&
        raw_rx_dose[j]!=max_raw_rx_dose)
    {
        iso=j;
/* printf("iso=%d \n", iso); */
    }
}
/* printf("max_raw_rx_dose=%f \n",max_raw_rx_dose); */

    total_num=target_num+crit_num+normal_num;

/* printf("crit_num=%d target_num=%d normal_num=%d
total_num=%d\n",crit_num,target_num,normal_num,total_num);
*/

/* Determine an overestimate of the # of MU used to produce
each BEV .imf
    file.
*/

```

```

mu_est=max_raw_rx_dose*100*2;
/* Convert prescription to Gy/mu_est.
*/
for(j=1;j<=2*ntot-1;j=j+2)
{
raw_rx_dose[j]=raw_rx_dose[j]/mu_est;
raw_rx_dose[j+1]=0.0;
}

/* Determine the uniformly distributed dose prescription in
terms of Gy
per mu_est.
*/

unif_raw_rx_dose=max_raw_rx_dose/mu_est;

fclose(rx_ptr);
return;
}

/*****
*****
This function, ROTATE_RX(), rotates the desired
raw_rx_dose[] distribution
about the Z axis (table rotation) then the X axis (gantry
rotation) to
obtain a new beam's eye view distribution called rx_dose[].
The isocenter of rotation is identified by iso.
*****
*****/

void rotate_rx(double table, double gantry, float rx_dose[])
{
FILE *rotatedrx_ptr, *rotatedcrt_ptr;
int x, y, z, newx, newy, newz, r, c, X, Y, nbr;
int ctr, new_ctr, ctr_xyz[32][32][64];
double n, m, l, matrix[4][4];

/* Create a mapping between the 3_D XYZ voxel locations and
the 1-D ctr,
calculate the SSD from the isocenter location assuming an
SAD of 100. */

ctr=1;
for(z=0;z<=nn[1]-1;z++)
{
for(y=0;y<=nn[2]-1;y++)
{
for(x=0;x<=nn[3]-1;x++)
{
ctr_xyz[x][y][z]=ctr;
if(ctr==iso)

```

```

        {
            l=x;
            m=y;
            n=z;
            ssd=100-((n+1)*voxdim);
/* printf("l=%3.0f m=%3.0f n=%3.0f ssd=%3.0f \n",
l,m,n,ssd); */
        }
        rx_dose[ctr]=rx_dose[ctr+1]=0;

crit_struct[view][ctr]=crit_struct[view][ctr+1]=0;
        ctr=ctr+2;
    }
}

/* Create the transformation matrix. This matrix translates
the isocenter
    voxel to the origin, rotates the patient table, rotates
the gantry,
    and translates the isocenter voxel back to its original
position.    */

table=table/57.29577951;
gantry=gantry/57.29577951;

matrix[0][0]= cos(table);
matrix[0][1]= sin(table)*cos(gantry);
matrix[0][2]= sin(table)*sin(gantry);
matrix[0][3]= 0;
matrix[1][0]=-sin(table);
matrix[1][1]= cos(table)*cos(gantry);
matrix[1][2]= cos(table)*sin(gantry);
matrix[1][3]= 0;
matrix[2][0]= 0;
matrix[2][1]=-sin(gantry);
matrix[2][2]= cos(gantry);
matrix[2][3]= 0;
matrix[3][0]=-1*cos(table)+m*sin(table)+1;
matrix[3][1]=-1*(sin(table)*cos(gantry))-
m*(cos(table)*cos(gantry))
            +n*sin(gantry)+m;
matrix[3][2]=-1*(sin(table)*sin(gantry))-
m*(cos(table)*sin(gantry))
            -n*cos(gantry)+n;
matrix[3][3]= 1;

/* for(r=0;r<=3;r++)
    {
        for(c=0;c<=3;c++)
            {
                printf("%10.3e", matrix[r][c]);
            }
        printf("\n");
    } */

```

```

/* Operate on all voxels containing a rx_dose value,
including critical
structures. The mapper previously determined is provided
the new XYZ
values and the new_ctr position is determined. The
rx_dose value from
the old ctr position is given to the new_ctr position.
*/

ctr=1;
for(z=0;z<=nn[1]-1;z++)
{
for(y=0;y<=nn[2]-1;y++)
{
for(x=0;x<=nn[3]-1;x++)
{
if(raw_rx_dose[ctr]>0 ||
raw_crit_struct[ctr]==1)
{

newx=(int)(x*matrix[0][0]+y*matrix[1][0]+z*matrix[2][0]
+matrix[3][0]+0.5);

newy=(int)(x*matrix[0][1]+y*matrix[1][1]+z*matrix[2][1]
+matrix[3][1]+0.5);

newz=(int)(x*matrix[0][2]+y*matrix[1][2]+z*matrix[2][2]
+matrix[3][2]+0.5);
/*printf("x= %5d y= %5d z= %5d \n",x,y,z);
printf("newx=%5d newy=%5d newz=%5d \n",newx,newy,newz); */
new_ctr=ctr_xyz[newx][newy][newz];
/*printf("crt=%10d new_ctr=%10d \n",ctr,new_ctr); */
rx_dose[new_ctr]=raw_rx_dose[ctr];
rx_dose[new_ctr+1]=0;

crit_struct[view][new_ctr]=raw_crit_struct[ctr];
crit_struct[view][new_ctr+1]=0;
}
ctr=ctr+2;
}
}
}

/* Having rotated the dose prescription and critical
structure arrays, now
correct for "holes" due to the the discrete mapping.
*/

X=nn[3];
Y=nn[2];
for(ctr=1;ctr<=2*ntot;ctr=ctr+2)

```

```

{
nbr=0;
if(rx_dose[ctr] == 0 &&
   ctr > 2*(nn[2]*nn[3]+nn[3]+1) &&
   ctr < 2*(nn[3]*nn[2]*nn[1]-(nn[2]*nn[3]+nn[3]+1)))
{
  if(rx_dose[ctr-2*Y*X-2*X+2] != 0) nbr++;
  if(rx_dose[ctr-2*Y*X+2] != 0) nbr++;
  if(rx_dose[ctr-2*Y*X+2*X+2] != 0) nbr++;
  if(rx_dose[ctr-2*Y*X-2*X] != 0) nbr++;
  if(rx_dose[ctr-2*Y*X] != 0) nbr++;
  if(rx_dose[ctr-2*Y*X+2*X] != 0) nbr++;
  if(rx_dose[ctr-2*Y*X-2*X-2] != 0) nbr++;
  if(rx_dose[ctr-2*Y*X-2] != 0) nbr++;
  if(rx_dose[ctr-2*Y*X+2*X-2] != 0) nbr++;

  if(rx_dose[ctr-2*X+2] != 0) nbr++;
  if(rx_dose[ctr+2] != 0) nbr++;
  if(rx_dose[ctr+2*X+2] != 0) nbr++;
  if(rx_dose[ctr-2*X] != 0) nbr++;
  if(rx_dose[ctr+2*X] != 0) nbr++;
  if(rx_dose[ctr-2*X-2] != 0) nbr++;
  if(rx_dose[ctr-2] != 0) nbr++;
  if(rx_dose[ctr+2*X-2] != 0) nbr++;

  if(rx_dose[ctr+2*Y*X-2*X+2] != 0) nbr++;
  if(rx_dose[ctr+2*Y*X+2] != 0) nbr++;
  if(rx_dose[ctr+2*Y*X+2*X+2] != 0) nbr++;
  if(rx_dose[ctr+2*Y*X-2*X] != 0) nbr++;
  if(rx_dose[ctr+2*Y*X] != 0) nbr++;
  if(rx_dose[ctr+2*Y*X+2*X] != 0) nbr++;
  if(rx_dose[ctr+2*Y*X-2*X-2] != 0) nbr++;
  if(rx_dose[ctr+2*Y*X-2] != 0) nbr++;
  if(rx_dose[ctr+2*Y*X+2*X-2] != 0) nbr++;

  if(nbr >= 17)
  {
    rx_dose[ctr]=unif_raw_rx_dose;
  }
}
}

for(ctr=1;ctr<=2*ntot;ctr=ctr+2)
{
nbr=0;
if(crit_struct[view][ctr] == 0 &&
   ctr > 2*(nn[2]*nn[3]+nn[3]+1) &&
   ctr < 2*(nn[3]*nn[2]*nn[1]-(nn[2]*nn[3]+nn[3]+1)))
{
  if(crit_struct[view][ctr-2*Y*X-2*X+2] != 0) nbr++;
  if(crit_struct[view][ctr-2*Y*X+2] != 0) nbr++;
  if(crit_struct[view][ctr-2*Y*X+2*X+2] != 0) nbr++;
  if(crit_struct[view][ctr-2*Y*X-2*X] != 0) nbr++;
  if(crit_struct[view][ctr-2*Y*X] != 0) nbr++;
}
}

```

```

if(crit_struct[view][ctr-2*Y*X+2*X] != 0) nbr++;
if(crit_struct[view][ctr-2*Y*X-2*X-2] != 0) nbr++;
if(crit_struct[view][ctr-2*Y*X-2] != 0) nbr++;
if(crit_struct[view][ctr-2*Y*X+2*X-2] != 0) nbr++;

if(crit_struct[view][ctr-2*X+2] != 0) nbr++;
if(crit_struct[view][ctr+2] != 0) nbr++;
if(crit_struct[view][ctr+2*X+2] != 0) nbr++;
if(crit_struct[view][ctr-2*X] != 0) nbr++;
if(crit_struct[view][ctr+2*X] != 0) nbr++;
if(crit_struct[view][ctr-2*X-2] != 0) nbr++;
if(crit_struct[view][ctr-2] != 0) nbr++;
if(crit_struct[view][ctr+2*X-2] != 0) nbr++;

if(crit_struct[view][ctr+2*Y*X-2*X+2] != 0) nbr++;
if(crit_struct[view][ctr+2*Y*X+2] != 0) nbr++;
if(crit_struct[view][ctr+2*Y*X+2*X+2] != 0) nbr++;
if(crit_struct[view][ctr+2*Y*X-2*X] != 0) nbr++;
if(crit_struct[view][ctr+2*Y*X] != 0) nbr++;
if(crit_struct[view][ctr+2*Y*X+2*X] != 0) nbr++;
if(crit_struct[view][ctr+2*Y*X-2*X-2] != 0) nbr++;
if(crit_struct[view][ctr+2*Y*X-2] != 0) nbr++;
if(crit_struct[view][ctr+2*Y*X+2*X-2] != 0) nbr++;

if(nbr >= 17)
{
    crit_struct[view][ctr]=1;
}
}

/* rotatedrx_ptr=fopen("rotated.rx", "w");
for(ctr=1;ctr<=2*ntot-1;ctr=ctr+2)
{
    fprintf(rotatedrx_ptr, "%10.3e \n",
rx_dose[ctr]*mu_est);
}
fclose(rotatedrx_ptr);

rotatedcrt_ptr=fopen("rotated.crt", "w");
for(ctr=1;ctr<=2*ntot-1;ctr=ctr+2)
{
    fprintf(rotatedcrt_ptr, "%10.3e \n",
crit_struct[view][ctr]);
}
fclose(rotatedcrt_ptr); */
return;
}

/*****
*****
This function, DECONV3D(), divides one complex array, FFT1[]
(FFT of rx_dose[])

```

```

, by another, FFT2[] (FFT of kernel[]), and stores the
result in deconv[]
(FFT of the TERMA distribution).
*****
*****/

void deconv3d(float fft1[],float fft2[],float deconv[])
{
    FILE *spect_ptr;

    int i,ctr;
    float
fft1_spect[132000],fft2_spect[132000],deconv_spect[132000];
    deconv[0]=0;

/* Divide the rx_dose FFT by the kernel FFT and determine
the power spectrum
of all three distributions.
*/

    for (i=2;i<=2*ntot;i+=2)
    {
        deconv[i-1]=(fft1[i-1]*fft2[i-1] + fft1[i]*fft2[i])/
            (fft2[i-1]*fft2[i-1] + fft2[i]*fft2[i]);

        deconv[i]=(fft1[i]*fft2[i-1] - fft1[i-1]*fft2[i])/
            (fft2[i-1]*fft2[i-1] + fft2[i]*fft2[i]);

        fft1_spect[i-1]=fft1[i-1]*fft1[i-1]+fft1[i]*fft1[i];
        fft2_spect[i-1]=fft2[i-1]*fft2[i-1]+fft2[i]*fft2[i];

        deconv_spect[i-1]=deconv[i-1]*deconv[i-1]+deconv[i]*deconv[i];
    }

/* spect_ptr=fopen("spectrum.fft", "w");

    for(ctr=1;ctr<=2*ntot;ctr=ctr+2)
    {
        fprintf(spect_ptr, "%10d %10.3e \n", ctr,
1000*fft2_spect[ctr]);
    }

    fclose(spect_ptr);
*/
    return;
}

/*****
*****
This function, PROJECTION(), projects the 3-D TERMA
distribution (J/kg per

```

mu) onto a plane by tracing each calculation voxel back towards the source.
 The plane is a 40 x 40 voxel grid located "refdist" cm from the source.

 *****/

```
void projection(float deconv[])
{
  FILE *terma_ptr;
  double x,y,z,xs,ys,s,r,u_water[10],u_lead[10];
  double d,invsg,terma;
  float  xmax,ymax,depthmax,xref,yref,max_terma=0;
  int    xcoord,ycoord,ctr=1,i,j;

  depthmax=(nn[1]-0.5)*voxdim; /* z dist to center of
  deepest calc voxel (cm)*/
  xmax=(nn[3]/2-0.5)*voxdim; /*x dist to center of
  farthest calc voxel (cm)*/
  ymax=(nn[2]/2-0.5)*voxdim; /*y dist to center of
  farthest calc voxel (cm)*/

  /* total attenuation constants in units of cm2/g
  */

  u_water[0]=0.09665,
  u_lead[0]=0.1614;
  u_water[1]=0.07059,
  u_lead[1]=0.0708;
  u_water[2]=0.04932,
  u_lead[2]=0.0455;
  u_water[3]=0.03961,
  u_lead[3]=0.0417;
  u_water[4]=0.03396,
  u_lead[4]=0.0415;
  u_water[5]=0.03026,
  u_lead[5]=0.0424;
  u_water[6]=0.02764,
  u_lead[6]=0.0436;

  /* Find the maximum value of TERMA resulting from
  deconvolution */

  for(j=1;j<=2*ntot-1;j=j+2)
  {
    if(deconv[j]>max_terma) {max_terma=deconv[j];}
  }

  /* Start with a fresh attenuation grid matrix for each new
  BEV */

  for(ycoord=1;ycoord<=40;ycoord++)
  {
    for(xcoord=1;xcoord<=40;xcoord++)
```

```

        {
            sum_thickness[xcoord][ycoord]=0;
            sum_ctr[xcoord][ycoord]=0;
        }
    }

/* Now loop through all voxels within boundary of
calculation volume limits */
    j=0;
    for(d=voxdim/2;d<=depthmax;d=d+voxdim)
        {
            z=ssd+d;
            for(y=-ymax;y<=ymax;y=y+voxdim)
                {
                    for(x=-xmax;x<=xmax;x=x+voxdim)
                        {

/* xref and yref locate intersection of ray line in plane of
lead attenuator.
Each ray line will pass through a portion of the grid
identified by xcoord
and ycoord.
*/

                            xref=x*refdist/z;
                            yref=y*refdist/z;

                            for(i=0;i<=19;i++)
                                {
                                    if(xref >= i*grid && xref < (i+1)*grid)
{xcoord=21+i;}
                                    if(yref >= i*grid && yref < (i+1)*grid)
{ycoord=21+i;}
                                }
                            for(i=0;i<=19;i++)
                                {
                                    if(xref <= -i*grid && xref > -(i+1)*grid)
{xcoord=20-i;}
                                    if(yref <= -i*grid && yref > -(i+1)*grid)
{ycoord=20-i;}
                                }

                            xs=x*(z-d)/z;
                            ys=y*(z-d)/z;
                            s=sqrt((z-d)*(z-d) + xs*xs + ys*ys);
                            r=sqrt(z*z + x*x + y*y);
                            invsq=(refdist/r)*(refdist/r);
                            terma=deconv[ctr];

/* A lead thickness value is calculated from each
calculation volume voxel by
reverse ray tracing towards the photon source. Also, a
count of voxels whose

```

ray lines pass through each attenuator grid square is tallied. Later, the sum of thicknesses for each grid square is divided by the summed voxel count to find the average thickness assigned to each grid square.

```

        */
        if(terma > 0.01*max_terma &&
crit_struct[view][ctr] != 1)
        {
thickness[xcoord][ycoord]=(log(terma/(eflu_ref*invsq*u_water
[e]*
u_water[e]*1.0*(r-s))))/
        1.6e-10*exp(-
        (-
u_lead[e]*11.33*(s/z)));
        if(thickness[xcoord][ycoord]>10.0)
        {
            thickness[xcoord][ycoord]=10.0;
        }
sum_thickness[xcoord][ycoord]=sum_thickness[xcoord][ycoord]+
thickness[xcoord][ycoord];
sum_ctr[xcoord][ycoord]=sum_ctr[xcoord][ycoord]+1;
        }

/* If the tissue voxel is within a critical structure or if
terma is
unrealistically less than or equal to zero, assign a
thickness value of
10.0 cm to be included in the average thickness.
*/
        if(crit_struct[view][ctr]==1)
        {
            j=j+1;
            thickness[xcoord][ycoord]=10.0*critical_wt;

sum_thickness[xcoord][ycoord]=sum_thickness[xcoord][ycoord]+
thickness[xcoord][ycoord];
sum_ctr[xcoord][ycoord]=sum_ctr[xcoord][ycoord]+1;
        }
        ctr=ctr+2;
    }
}
}
/* printf("total crit_struct present=%d \n",j); */

```

```

terma_ptr=fopen("terma", "w");
for(ctr=1;ctr<=2*ntot;ctr=ctr+2)
    {
        fprintf(terma_ptr, "%10.3e \n", 1000*deconv[ctr]);
    }

fclose(terma_ptr);
return;
}

/*****
*****
This function, OUTTHICKNESS(), Print file containing
thickness of lead (cm)
for each grid position of lead attenuator.
*****
*****/

void outthickness(void)
{
    FILE *outthickness_ptr;
    int xcoord, ycoord;
    float thin=10.0;
    char thickness_file[100];

    sprintf(thickness_file, "T%dG%d.imf", (int)table, (int)gantry);

    if ((outthickness_ptr=fopen(thickness_file, "w"))==NULL)
    {
        printf("*** UNABLE TO WRITE ATTENUATOR THICKNESS FILE
***\n");
        exit(0);
    }

    for(ycoord=1;ycoord<=40;ycoord++)
    {
        for(xcoord=1;xcoord<=40;xcoord++)
        {
            if(sum_ctr[xcoord][ycoord]==0.0)
            {
                ave_thickness[xcoord][ycoord]=10.0;
            }
            if(sum_ctr[xcoord][ycoord]!=0.0)
            {
                ave_thickness[xcoord][ycoord]=sum_thickness[xcoord][ycoord]/
sum_ctr[xcoord][ycoord];
            }
            if(ave_thickness[xcoord][ycoord]<thin)
            {
                thin=ave_thickness[xcoord][ycoord];
            }
        }
    }
}

```

```

    }
  }
}

for(ycoord=1;ycoord<=40;ycoord++)
{
  for(xcoord=1;xcoord<=40;xcoord++)
  {
    if(ave_thickness[xcoord][ycoord] != 10.0)
    {
ave_thickness[xcoord][ycoord]=ave_thickness[xcoord][ycoord]-
thin;
    }
    fprintf(outthickness_ptr,"%5.2f
",ave_thickness[xcoord][ycoord]);
    }
    fprintf(outthickness_ptr, "\n");
  }
  fclose(outthickness_ptr);
return;
}

/*****
*****
This function, TERMA3D(), calculates a three dimensional
TERMA (units of
J/kg per mu) distribution for a collimated point source of
radiation
incident on a flat, homogeneous water phantom.
*****
*****/

void terma3d(float terma[])
{
FILE *imf_ptr;
double x,y,z,xs,ys,s,r,u_water[10],u_lead[10];
double d,atten;
double thickness[50][50];
float
invsq,xedge,yedge,xmax,ymax,depthmax,x_fs,y_fs,xref,yref,tem
P;
int    check,xcoord,ycoord,ctr=0,i;
char   imf_file[100];

    depthmax=(nn[1]-0.5)*voxdim; /*z dist to center of
deepest calc voxel (cm) */
    xmax=(nn[3]/2-0.5)*voxdim; /* x dist to center of
farthest calc voxel cm)*/
    ymax=(nn[2]/2-0.5)*voxdim; /* y dist to center of
farthest calc voxel(cm)*/
    x_fs=sq_field; /* total width of field size
in x dir (cm) */

```

```

    y_fs=sq_field;                /* total width of field size
in y_dir (cm)                    */
    terma[0]=0;                   /* initialize all TERMA to
zero                               */

/* total attenuation constants in units of cm2/g
*/

    u_water[0]=0.09665,
u_lead[0]=0.1614;
    u_water[1]=0.07059,
u_lead[1]=0.0708;
    u_water[2]=0.04932,
u_lead[2]=0.0455;
    u_water[3]=0.03961,
u_lead[3]=0.0417;
    u_water[4]=0.03396,
u_lead[4]=0.0415;
    u_water[5]=0.03026,
u_lead[5]=0.0424;
    u_water[6]=0.02764,
u_lead[6]=0.0436;

/* Read file containing thickness of lead (cm) for each grid
position of
lead attenuator
*/

    sprintf(imf_file,"T%dG%d.imf", (int)table, (int)gantry);

    if ((imf_ptr=fopen(imf_file, "r"))==NULL)
    {
        printf("*** UNABLE TO OPEN IMF FILE ***\n");
        exit(0);
    }
    for(ycoord=1;ycoord<=40;ycoord++)
    {
        for(xcoord=1;xcoord<=40;xcoord++)
        {
            fscanf(imf_ptr,"%f", &temp);
            thickness[xcoord][ycoord]=(double)temp;
        }
    }
    fclose(imf_ptr);

    for(d=voxdim/2;d<=depthmax;d=d+voxdim)
    {
        z=ssd+d;
        for(y=-ymax;y<=ymax;y=y+voxdim)
        {
            for(x=-xmax;x<=xmax;x=x+voxdim)
            {
                xref=x*refdist/z;
                yref=y*refdist/z;
            }
        }
    }

```

```

        for(i=0;i<=19;i++)
            {
                if(xref >= i*grid && xref < (i+1)*grid)
{xcoord=21+i;}
                if(yref >= i*grid && yref < (i+1)*grid)
{ycoord=21+i;}
            }
        for(i=0;i<=19;i++)
            {
                if(xref <= -i*grid && xref > -(i+1)*grid)
{xcoord=20-i;}
                if(yref <= -i*grid && yref > -(i+1)*grid)
{ycoord=20-i;}
            }

        xs=x*(z-d)/z;
        ys=y*(z-d)/z;
        s=sqrt((z-d)*(z-d) + xs*xs + ys*ys);
        r=sqrt(z*z + x*x + y*y);
        invsq=(refdist/r)*(refdist/r);
        atten=exp((-u_water[e]*1.0*(r-s))+(-
u_lead[e]*11.33*
        thickness[xcoord][ycoord]*(s/z)));
        xedge=(z/(z-d))*x_fs/2;
        yedge=(z/(z-d))*y_fs/2;
        check=1;
        if(fabs(x) >= xedge || fabs(y) >= yedge)
            {
                check=0;
            }
        ctr++;

/* The units of terma are (J/kg) as opposed to Gy since the
energy is
    released, not deposited.
*/

terma[ctr]=eflu_ref*invsq*atten*check*u_water[e]*1.6e-10;

/* Provide zero padding for bottom portion of TERMA to
prevent wrap
    around effects near surface. This zero TERMA region will
contain
    contaminated dose information and will not be displayed.
*/

        if(d>depthmax-((nn[1]/4)*voxdim))
            {
                terma[ctr]=0;
            }

```

```

/* Real values of TERMA assigned to x,y,z
*/

    ctrx[ctr]=x;
    ctry[ctr]=y;
    ctrz[ctr]=z+voxdim/2;
    ctr++;

/* Imaginary values of TERMA assigned to x,y,z
*/

    terma[ctr]=0;
    ctrx[ctr]=x;
    ctry[ctr]=y;
    ctrz[ctr]=z+voxdim/2;
    }
    }
    }
return;
}

/*****
*****
This function, NDIMFFT(), performs an FFT in ndim dimensions
if isign=1 and
an inverse FFT if isign=-1. The vector nn[] contains the
dimensions of the
input data in each dimension. Each dimension must be a power
of two.
*****
*****/

void ndimfft(float data[],int isign)
{
    int
i1=0,i2=0,i3=0,i2rev=0,i3rev=0,ip1=0,ip2=0,ip3=0,ifp1=0,ifp2
=0;
    int    ibit=0,idim=0,k1=0,k2=0,n=0,nprev=0,nrem=0,i=0;
    float  tempi=0,tempr=0;
    double theta=0,wi=0,wpi=0,wpr=0,wr=0,wtemp=0;

/* Main loop over the dimensions
*/
    nprev=1;
    for(idim=ndim;idim>=1;idim--)
    {
        n=nn[idim];
        nrem=ntot/(n*nprev);
        ip1=nprev << 1;
        ip2=ip1*n;
        ip3=ip2*nrem;
        i2rev=1;

```

```
/* Bit reversal section of routine
*/
```

```
for(i2=1;i2<=ip2;i2+=ip1)
{
  if(i2 < i2rev)
  {
    for(i1=i2;i1<=i2+ip1-2;i1+=2)
    {
      for(i3=i1;i3<=ip3;i3+=ip2)
      {
        i3rev=i2rev+i3-i2;
        SWAP(data[i3],data[i3rev]);
        SWAP(data[i3+1],data[i3rev+1]);
      }
    }
  }
  ibit=ip2 >> 1;
  while(ibit >= ip1 && i2rev > ibit)
  {
    i2rev -= ibit;
    ibit >>= 1;
  }
  i2rev += ibit;
}
```

```
/* Here begins the Danielson-Lanczos section of the routine
*/
```

```
ifp1=ip1;
while(ifp1 < ip2)
{
  ifp2=ifp1 << 1;
  theta=isign*6.28318530717959/(ifp2/ip1);
  wtemp=sin(0.5*theta);
  wpr = -2.0*wtemp*wtemp;
  wpi=sin(theta);
  wr=1.0;
  wi=0.0;
  for(i3=1;i3<=ifp1;i3+=ip1)
  {
    for(i1=i3;i1<=i3+ip1-2;i1+=2)
    {
      for(i2=i1;i2<=ip3;i2+=ifp2)
      {
        k1=i2;
        k2=k1+ifp1;
        tempr=wr*data[k2]-wi*data[k2+1];
        tempi=wr*data[k2+1]+wi*data[k2];
        data[k2]=data[k1]-tempr;
        data[k2+1]=data[k1+1]-tempi;
        data[k1] += tempr;
        data[k1+1] += tempi;
      }
    }
  }
}
```

```

        }
        wr=(wtemp=wr)*wpr-wi*wpi+wr;
        wi=wi*wpr+wtemp*wpi+wi;
    }
    ifp1=ifp2;
}
nprev *= n;
}
if(isign===-1)
{
    for(i=1;i<=2*ntot;i++)
    {
        data[i]=data[i]/ntot;
    }
}
return;
}

/*****
*****
This function, CONV3D(), multiplies two complex arrays,
FFT1[] and FFT2[],
and stores the result in conv[].
*****
*****/

void conv3d(float fft1[],float fft2[],float conv[])
{
    register int i;

    conv[0]=0;
    for (i=2;i<=2*ntot;i+=2)
    {
        conv[i-1]=conv[i]=0;
        conv[i-1]=(fft1[i-1]*fft2[i-1] - fft1[i]*fft2[i]);
        conv[i]=(fft1[i]*fft2[i-1] + fft1[i-1]*fft2[i]);
    }
    return;
}

/*****
*****
This function, MONO_SUM(), sums the monoenergetic dose
distributions to
create a single polyenergetic distribution.
*****
*****/

void mono_sum(float mono_dose[])
{
    int ctr;

/* Start with fresh cumulative dose array for each BEV
*/

```

```

if(e==0)
  {
  for(ctr=1;ctr<=2*ntot-1;ctr=ctr+2)
    {
    mono_sum_dose[ctr]=0;
    mono_sum_dose[ctr+1]=0;
    }
  }

/* Sum the energy fluence weighted dose arrays to create a
single
polyenergetic dose array for each BEV
*/

ctr=0;
for(ctr=1;ctr<=2*ntot-1;ctr=ctr+2)
  {

mono_sum_dose[ctr]=mono_sum_dose[ctr]+mono_dose[ctr]*e_wt[e]
;
  mono_sum_dose[ctr+1]=0;
  }
return;
}

/*****
*****
This function, ROTATE_DOSE(), rotates the desired dose
distribution about the
Z axis (table rotation) then the X axis (gantry rotation) to
obtain a new
beam's eye view. The isocenter of rotation is identified by
iso.
*****/
*****/

void rotate_dose(double table, double gantry)
  {
  register int x, y, z;
  int      newx, newy, newz, r, c, X, Y, nbr;
  int      ctr, new_ctr, ctr_xyz[32][32][64];
  double   n, m, l, matrix[4][4];

/* Create a mapping between the 3_D XYZ voxel locations and
the 1-D ctr */

  ctr=1;
  for(z=0;z<=nn[1]-1;z++)
    {
    for(y=0;y<=nn[2]-1;y++)
      {
      for(x=0;x<=nn[3]-1;x++)
        {

```

```

        ctr_xyz[x][y][z]=ctr;
/*      printf("%5d %5d %5d %5d %5d \n", x, y, z,
ctr_xyz[x][y][z],iso); */
        if(ctr==iso)
        {
            l=x;
            m=y;
            n=z;
/*      printf("%10.3e %10.3e %10.3e %5d \n", l, m,
n, iso); */
        }
        dose[view][ctr]=dose[view][ctr+1]=0;
        ctr=ctr+2;
/*      printf("dose[%d][%d]=%10.3e \n", view, ctr,
dose[view][ctr]); */
    }
}

/* Create the transformation matrix. This matrix translates
the isocenter
    voxel to the origin, rotates the gantry, rotates the
patient table,
    and translates the isocenter voxel back to its original
position. */

table=table/57.29577951;
gantry=gantry/57.29577951;

matrix[0][0]= cos(table);
matrix[0][1]= sin(table);
matrix[0][2]= 0;
matrix[0][3]= 0;
matrix[1][0]=-cos(gantry)*sin(table);
matrix[1][1]= cos(gantry)*cos(table);
matrix[1][2]= sin(gantry);
matrix[1][3]= 0;
matrix[2][0]= sin(gantry)*sin(table);
matrix[2][1]=-sin(gantry)*cos(table);
matrix[2][2]= cos(gantry);
matrix[2][3]= 0;
matrix[3][0]=-l*cos(table)+m*(cos(gantry)*sin(table))-
n*(sin(gantry)*sin(table))+1;
matrix[3][1]=-l*sin(table)-m*(cos(gantry)*cos(table))+
n*(sin(gantry)*cos(table))+m;
matrix[3][2]=-m*sin(gantry)-n*cos(gantry)+n;
matrix[3][3]= 1;

/* for(r=0;r<=3;r++)
    {
        for(c=0;c<=3;c++)
            {
                printf("%10.3e", matrix[r][c]);
            }
    }

```

```

        printf("\n");
    }
    /*
    /* Operate on all voxels. The mapper previously determined
    is provided
    the new XYZ values and the new ctr position is
    determined. The dose
    value from the old ctr position is given to the new ctr
    position.
    */

    ctr=1;
    for(z=0;z<=nn[1]-1;z++)
        {
        for(y=0;y<=nn[2]-1;y++)
            {
            for(x=0;x<=nn[3]-1;x++)
                {

newx=(int)(x*matrix[0][0]+y*matrix[1][0]+z*matrix[2][0]
            +matrix[3][0]+0.5);

newy=(int)(x*matrix[0][1]+y*matrix[1][1]+z*matrix[2][1]
            +matrix[3][1]+0.5);

newz=(int)(x*matrix[0][2]+y*matrix[1][2]+z*matrix[2][2]
            +matrix[3][2]+0.5);

                if(newx<0 || newx>nn[3]-1 || newy<0 ||
newy>nn[2]-1 ||
                    newz<0 || newz>nn[1]-1)
                    {
/*
                    printf("limit %d %d %10.3e
\n",view,ctr,dose[view][ctr]); */
                    goto NEXT;
                    }

                    new_ctr=ctr_xyz[newx][newy][newz];
                    dose[view][new_ctr]=mono_sum_dose[ctr];
                    dose[view][new_ctr+1]=0;
                    NEXT:
                    ctr=ctr+2;
                    }
                }
            }
        }
    }

/* for(ctr=1;ctr<=2*ntot;ctr=ctr+2)
    {
    if(dose[view][ctr] != 0)
        {
        printf("nonzero at ctr #: %5d dose value=
%10.3e\n", ctr,
            dose[view][ctr]);
        }
    }

```

```

    }
*/
X=nn[3];
Y=nn[2];
for(ctr=1;ctr<=2*ntot;ctr=ctr+2)
{
  nbr=0;
  if(dose[view][ctr] == 0 &&
     ctr > 2*(nn[2]*nn[3]+nn[3]+1) &&
     ctr < 2*(nn[3]*nn[2]*nn[1]-(nn[2]*nn[3]+nn[3]+1)))
  {
    if(dose[view][ctr-2*Y*X-2*X+2] != 0) nbr++;
    if(dose[view][ctr-2*Y*X+2] != 0) nbr++;
    if(dose[view][ctr-2*Y*X+2*X+2] != 0) nbr++;
    if(dose[view][ctr-2*Y*X-2*X] != 0) nbr++;
    if(dose[view][ctr-2*Y*X] != 0) nbr++;
    if(dose[view][ctr-2*Y*X+2*X] != 0) nbr++;
    if(dose[view][ctr-2*Y*X-2*X-2] != 0) nbr++;
    if(dose[view][ctr-2*Y*X-2] != 0) nbr++;
    if(dose[view][ctr-2*Y*X+2*X-2] != 0) nbr++;

    if(dose[view][ctr-2*X+2] != 0) nbr++;
    if(dose[view][ctr+2] != 0) nbr++;
    if(dose[view][ctr+2*X+2] != 0) nbr++;
    if(dose[view][ctr-2*X] != 0) nbr++;
    if(dose[view][ctr+2*X] != 0) nbr++;
    if(dose[view][ctr-2*X-2] != 0) nbr++;
    if(dose[view][ctr-2] != 0) nbr++;
    if(dose[view][ctr+2*X-2] != 0) nbr++;

    if(dose[view][ctr+2*Y*X-2*X+2] != 0) nbr++;
    if(dose[view][ctr+2*Y*X+2] != 0) nbr++;
    if(dose[view][ctr+2*Y*X+2*X+2] != 0) nbr++;
    if(dose[view][ctr+2*Y*X-2*X] != 0) nbr++;
    if(dose[view][ctr+2*Y*X] != 0) nbr++;
    if(dose[view][ctr+2*Y*X+2*X] != 0) nbr++;
    if(dose[view][ctr+2*Y*X-2*X-2] != 0) nbr++;
    if(dose[view][ctr+2*Y*X-2] != 0) nbr++;
    if(dose[view][ctr+2*Y*X+2*X-2] != 0) nbr++;

    if(nbr >= 17)
    {
      dose[view][ctr]=(
        dose[view][ctr-2*Y*X-2*X+2] + dose[view][ctr-
2*Y*X+2]+
        dose[view][ctr-2*Y*X+2*X+2] + dose[view][ctr-
2*Y*X-2*X]+
        dose[view][ctr-2*Y*X] + dose[view][ctr-
2*Y*X+2*X]+
        dose[view][ctr-2*Y*X-2*X-2] + dose[view][ctr-
2*Y*X-2]+
        dose[view][ctr-2*Y*X+2*X-2] + dose[view][ctr-
2*X+2]+

```

```

        dose[view][ctr+2]          +
dose[view][ctr+2*X+2]+
        dose[view][ctr-2*X]      +
dose[view][ctr+2*X]+
        dose[view][ctr-2*X-2]    + dose[view][ctr-2]+
        dose[view][ctr+2*X-2]    +
dose[view][ctr+2*Y*X-2*X+2] +
        dose[view][ctr+2*Y*X+2]  +
dose[view][ctr+2*Y*X+2*X+2]+
        dose[view][ctr+2*Y*X-2*X] +
dose[view][ctr+2*Y*X]+
        dose[view][ctr+2*Y*X+2*X] +
dose[view][ctr+2*Y*X-2*X-2]+
        dose[view][ctr+2*Y*X-2]  +
dose[view][ctr+2*Y*X+2*X-2])/nbr;
    }
}

return;
}

```

```

/*****
*****
This function, OUTDOSE(), writes each rotated dose array,
dose[][] to a file
easily read by AVS. The X axis values change most rapidly,
then Y, and the Z
axis values most slowly. This is the same arrangement for
reading all 3-D
data into a 1-D array. Units are Gy per mu.
*****/

```

```

void outdose()
{
    FILE *outdose_ptr;
    int ctr;
    char outfile[100];

    sprintf(outfile, "T%dG%d.dos", (int)table, (int)gantry);

    outdose_ptr=fopen(outfile, "w");

    for(ctr=1;ctr<=2*ntot;ctr=ctr+2)
/* for(ctr=991;ctr<=2*ntot;ctr=ctr+2048) */
        {
            fprintf(outdose_ptr, "%10.3e \n", dose[view][ctr]);
        }
    fclose(outdose_ptr);
    return;
}

```

```

/*****
*****
This function, OPTIMIZE(), contains the starting values and
annealing
schedule required by AMEBSA().
*****
*****/

```

```

void optimize(void)
{
    float ftol=1.0E-5;
    int i,iiter,iter,j,jiter,nit;
    int ndim=view_max;
    float tempr,ȳb,ybb;
    float **p,*x,*y,*pb;
    static float xoff[100];

    printf("ndim=%d \n", ndim);

    p=matrix(1,ndim+1,1,ndim);
    x=vector(1,ndim);
    y=vector(1,ndim+1);
    pb=vector(1,ndim);

    for(i=1;i<=ndim;i++)
    {
        xoff[i]=unif_raw_rx_dose*mu_est*250/ndim;
    }

    for(i=1;i<=ndim+1;i++)
    {
        for(j=1;j<=ndim;j++)
        {
            p[i][j]=0.0;
        }
    }

    printf("Input temp, iiter:");
    scanf("%f %d",&tempr,&iiter);

    for(j=2;j<=ndim+1;j++)
    {
        p[j][j-1]=1.0;
    }
    for(i=1;i<=ndim+1;i++)
    {
        for(j=1;j<=ndim;j++)
        {
            x[j]=(p[i][j] = p[i][j] + xoff[j]);
/*            printf("xoff=%10.3e \n", x[j]);          */
        }
        y[i]=funkt(x);
/*        printf("objfunc(xoff)=%10.3e \n", y[i]);      */
    }
}

```

```

}
yb=1.0e30;
ybb=1.0e30;
nit=0;
printf("\n");
printf("    nit      temp      objfunc \n");
for(jiter=1;jiter<=100;jiter++)
{
    iter=iiter;
    tempr *= 0.80;
    ambsa(p,y,ndim,pb,&yb,ftol,funk,&iiter,temptr);
    nit += iiter-iter;
    if(yb < ybb)
    {
        ybb=yb;
        printf("%6d %10.3e ",nit,temptr);
/*      for(j=1;j<=ndim;j++)
            {
                printf("%10.5f ",pb[j]);
            }
            printf("%15.5e\n",yb);
        */
        if(iter > 0) break;
    }

/*    for(i=1;i<=ndim+1;i++)
        {
            printf("%3d ",i);
            for(j=1;j<=ndim;j++)
                {
                    printf("%12.6f ",p[i][j]);
                }
            printf("%15.7e\n",y[i]);
        }
        printf("\n");
        j=0;
*/

for(table=table_start;table<=table_end;table=table+table_delta)
{

for(gantry=gantry_start;gantry<=gantry_end;gantry=gantry+gantry_delta)
{
    j=j+1;
    printf("T=%3.0f  G=%3.0f  MU: %d
\n",table,gantry,(int)pb[j]);
}
}
printf("\n");
printf("Objective function =%15.7e\n\n",yb);

free_vector(pb,1,ndim);
free_vector(y,1,ndim+1);

```

```

    free_vector(x,1,ndim);
    free_matrix(p,1,ndim+1,1,ndim);
    return;
}

/*****
*****
This function, FUNK(), contains the objective function used
to measure the
difference between the desired dose distribution contained
in rx_dose[] and
the sum of the individual calculated BEV dose distributions
contained in
dose[][].
*****/
*****/

float funk(float bev_mu[])
{
    register int bev, ctr, g, t;
    double      objfunc=0, neg_penalty=1, peak_penalty=1,
norm_penalty=1, wt;
    float      peak_value=0,mu_total=0;

    for(ctr=1;ctr<=2*ntot-1;ctr=ctr+2)
    {
        sum_dose[ctr]=0;
    }

/* Award penalty if any beam weighting drops below zero
*/

    for(bev=1;bev<=view_max;bev++)
    {
        if (bev_mu[bev]<0)
        {
            neg_penalty=1000;
        }
    }

/* Find the total number of monitor units
*/

    for(bev=1;bev<=view_max;bev++)
    {
        mu_total=mu_total+bev_mu[bev];
    }

/* Penalize if beam weights drop below that level defined by
normal
tissue weighting.
*/

    for(bev=1;bev<=view_max;bev++)

```

```

    {
    if(bev_mu[bev] < mu_total*normal_wt/view_max)
        {
        norm_penalty=1000;
        }
    }

/* The contribution to dose in each voxel is summed from all
views */

    for(ctr=1;ctr<=2*ntot-1;ctr=ctr+2)
    {
    view=0;
    for(t=table_start;t<=table_end;t=t+table_delta)
        {
        for(g=gantry_start;g<=gantry_end;g=g+gantry_delta)
            {
            view=view+1;

sum_dose[ctr]=sum_dose[ctr]+bev_mu[view]*dose[view][ctr];
            }
        }
    }

/* Determine the maximum dose value
*/

    if(sum_dose[ctr]>peak_value)
    {
    peak_value=sum_dose[ctr];
    }

/* Assign a weighting to the target tissue voxels (1 by
default) */

    if(raw_rx_dose[ctr]!=0 && raw_crit_struct[ctr]!=1)
    {
    wt=target_wt/target_num;
    }

/* Assign a weighting to the normal tissue voxels relative
to target */

    if(raw_rx_dose[ctr]==0)
    {
    wt=normal_wt/normal_num;
    }

/* Assign a weighting to the critical structure voxels
relative to target */

    if(raw_crit_struct[ctr]==1)
    {
    wt=critical_wt/crit_num;
    }

```

```

/* Determine the objective function
*/
    objfunc=objfunc+pow(fabs((raw_rx_dose[ctr]*mu_est-
sum_dose[ctr])*wt),2);
    }

/* Apply penalty if maximum dose is less than prescribed for
target */

    if(peak_value<unif_raw_rx_dose*mu_est)
    {
        peak_penalty=1000;
    }

    objfunc=objfunc*neg_penalty*peak_penalty*norm_penalty;
    objfunc=sqrt(objfunc);
    return objfunc;
}

/*****
*****
This function, AMEBSA(), performs a multidimensional
optimization of the
function FUNK through a combination of simulated annealing
and downhill
simplex.
*****
*****/

void amebssa(float **p, float y[], int ndim, float pb[],
float *yb, float ftol,
float (*funk)(float []), int *iter, float tempr)
{
    float amotsa(float **p, float y[], float psum[], int
ndim, float pb[],
float *yb, float (*funk)(float []), int ihi,
float *yhi,
float fac);
    float ranl(long *idum);
    int i,ih,i,ilo,j,m,n,mpts=ndim+1;
    float rtol,sum,swap,yhi,ylo,ynhi,ysave,yt,ytry,*psum;

    psum=vector(1,ndim);
    tt = -tempr;
    GET_PSUM
    for (;) {
        ilo=1;
        ihi=2;
        ynhi=ylo=y[1]+tt*log(ranl(&idum));
        yhi=y[2]+tt*log(ranl(&idum));
        if (ylo > yhi) {
            ihi=1;

```

```

        ilo=2;
        ynhi=yhi;
        yhi=ylo;
        ylo=ynhi;
    }
    for (i=3;i<=mpts;i++) {
        yt=y[i]+tt*log(ranl(&idum));
        if (yt <= ylo) {
            ilo=i;
            ylo=yt;
        }
        if (yt > yhi) {
            ynhi=yhi;
            ihi=i;
            yhi=yt;
        } else if (yt > ynhi) {
            ynhi=yt;
        }
    }
    rtol=2.0*fabs(yhi-ylo)/(fabs(yhi)+fabs(ylo));
    if (rtol < ftol || *iter < 0) {
        swap=y[1];
        y[1]=y[ilo];
        y[ilo]=swap;
        for (n=1;n<=ndim;n++) {
            swap=p[1][n];
            p[1][n]=p[ilo][n];
            p[ilo][n]=swap;
        }
        break;
    }
    *iter -= 2;
    ytry=amotsa(p,y,psum,ndim,pb,yb,funk,ihi,&yhi,-
1.0);
    if (ytry <= ylo) {
        ytry=amotsa(p,y,psum,ndim,pb,yb,funk,ihi,&yhi,2.0);
    } else if (ytry >= ynhi) {
        ysave=yhi;
        ytry=amotsa(p,y,psum,ndim,pb,yb,funk,ihi,&yhi,0.5);
        if (ytry >= ysave) {
            for (i=1;i<=mpts;i++) {
                if (i != ilo) {
                    for (j=1;j<=ndim;j++) {
                        psum[j]=0.5*(p[i][j]+p[ilo][j]);
                        p[i][j]=psum[j];
                    }
                    y[i]=(*funk)(psum);
                }
            }
            *iter -= ndim;
            GET_PSUM

```

```

        }
        } else ++(*iter);
    }
    free_vector(psum,1,ndim);
}

/*****
*****
This function, AMOTSA(), is called by AMEBSA().
*****
*****/

float amotsa(float **p, float y[], float psum[], int ndim,
float pb[],
float *yb, float (*funk)(float []), int ihi, float
*yhi,
float fac)
{
    float ran1(long *idum);
    int j;
    float fac1,fac2,yflu,ytry,*ptry;

    ptry=vector(1,ndim);
    fac1=(1.0-fac)/ndim;
    fac2=fac1-fac;
    for (j=1;j<=ndim;j++)
        ptry[j]=psum[j]*fac1-p[ihi][j]*fac2;
    ytry=(*funk)(ptry);
    if (ytry <= *yb) {
        for (j=1;j<=ndim;j++) pb[j]=ptry[j];
        *yb=ytry;
    }
    yflu=ytry-tt*log(ran1(&idum));
    if (yflu < *yhi) {
        y[ihi]=ytry;
        *yhi=yflu;
        for (j=1;j<=ndim;j++) {
            psum[j] += ptry[j]-p[ihi][j];
            p[ihi][j]=ptry[j];
        }
    }
    free_vector(ptry,1,ndim);
    return yflu;
}

/*****
*****
This function, RAN1(), returns a uniform random deviate
between 0.0 and 1.0
(exclusive of endpoint values). It uses idum to initialize.
*****
*****/

float ran1(long *idum)

```

```

{
    int j;
    long k;
    static long iy=0;
    static long iv[NTAB];
    float temp;

    if (*idum <= 0 || !iy) {
        if (-(*idum) < 1) *idum=1;
        else *idum = -(*idum);
        for (j=NTAB+7;j>=0;j--) {
            k=(*idum)/IQ;
            *idum=IA*( *idum-k*IQ)-IR*k;
            if (*idum < 0) *idum += IM;
            if (j < NTAB) iv[j] = *idum;
        }
        iy=iv[0];
    }
    k=(*idum)/IQ;
    *idum=IA*( *idum-k*IQ)-IR*k;
    if (*idum < 0) *idum += IM;
    j=iy/NDIV;
    iy=iv[j];
    iv[j] = *idum;
    if ((temp=AM*iy) > RNMX) return RNMX;
    else return temp;
}

/*****
*****
The following functions are utilities needed by AMEBSA() and
AMOTRY().
*****
*****/

void nrerror(char error_text[])
/* Numerical Recipes standard error handler */
{
    fprintf(stderr,"Numerical Recipes run-time
error...\n");
    fprintf(stderr,"%s\n",error_text);
    fprintf(stderr,"...now exiting to system...\n");
    exit(1);
}

float *vector(long nl, long nh)
/* allocate a float vector with subscript range v[nl..nh] */
{
    float *v;

    v=(float *)malloc((size_t) ((nh-
nl+1+NR_END)*sizeof(float)));
    if (!v) nrerror("allocation failure in vector()");
    return v-nl+NR_END;
}

```

```

}

float **matrix(long nrl, long nrh, long ncl, long nch)
/* allocate a float matrix with subscript range
m[nrl..nrh][ncl..nch] */
{
    long i, nrow=nrh-nrl+1, ncol=nch-ncl+1;
    float **m;

    /* allocate pointers to rows */
    m=(float **)
malloc((size_t)((nrow+NR_END)*sizeof(float*)));
    if (!m) nrerror("allocation failure 1 in matrix()");
    m += NR_END;
    m -= nrl;

    /* allocate rows and set pointers to them */
    m[nrl]=(float *)
malloc((size_t)((nrow*ncol+NR_END)*sizeof(float)));
    if (!m[nrl]) nrerror("allocation failure 2 in
matrix()");
    m[nrl] += NR_END;
    m[nrl] -= ncl;

    for(i=nrl+1;i<=nrh;i++) m[i]=m[i-1]+ncol;

    /* return pointer to array of pointers to rows */
    return m;
}

void free_vector(float *v, long nl, long nh)
/* free a float vector allocated with vector() */
{
    free((FREE_ARG) (v+nl-NR_END));
}

void free_matrix(float **m, long nrl, long nrh, long ncl,
long nch)
/* free a float matrix allocated by matrix() */
{
    free((FREE_ARG) (m[nrl]+ncl-NR_END));
    free((FREE_ARG) (m+nrl-NR_END));
}

/*****
*****
This function, OUTFINAL(), writes the final optimized each
dose array,
sum_dose[] to a file easily read by AVS. The X axis values
change most
rapidly, then Y, and the Z axis values most slowly. This is
the same
arrangement for reading all 3-D data into a 1-D array. Units
are Gy or %.

```

```
*****
*****/
```

```
void outffinal()
{
  FILE          *outfinal_ptr;
  int           response;
  float         max_value=0, norm;
  register int  ctr;

  outffinal_ptr=fopen("doseplan", "w");

  /* Find the largest dose value in the summed dose
  distribution          */

  for(ctr=1;ctr<=2*ntot;ctr=ctr+2)
  {
    if(sum_dose[ctr]>max_value)
    {
      max_value=sum_dose[ctr];
    }
  }

  printf("max_value=%10.3e Gy \n",max_value);

  AGAIN:
  printf("Do you prefer absolute dose in Gy (1) or
  normalized dose (2):");
  scanf("%d",&response);
  if(response==1) norm=1;
  if(response==2) norm=100/max_value;
  if(response !=1 && response != 2) goto AGAIN;

  for(ctr=1;ctr<=2*ntot;ctr=ctr+2)
  {
    fprintf(outfinal_ptr, "%10.3e \n",
    sum_dose[ctr]*norm);
  }

  fclose(outfinal_ptr);
  return;
}
```

```
/*****
*****
This function, EVALUATE(), provides statistical information
for the 3-D
dose distribution resulting from optimal BEV MU weighting.
The maximum and
minimum dose values and a dose-volume frequency analysis are
determined for
the total, target, and critical structure volumes.
*****
*****/
```

```

void evaluate(void)
{
FILE *total_ptr;
FILE *critical_ptr;
FILE *target_ptr;
FILE *normal_ptr;
int ctr, bin, critical=0;
float total_max_value=0;
float total_min_value=100;
float critical_max_value=0;
float critical_min_value=100;
float target_max_value=0;
float target_min_value=100;
float normal_max_value=0;
float normal_min_value=100;
float total_vol=0, critical_vol=0, target_vol=0,
normal_vol=0;
float total_norm, critical_norm, target_norm,
normal_norm;
float total_volume[105], critical_volume[105];
float target_volume[105], normal_volume[105];
float delta_vol=voxdim*voxdim*voxdim;

for(bin=0;bin<=99;bin++)
{
total_volume[bin]=critical_volume[bin]=0;
target_volume[bin]=normal_volume[bin]=0;
}

for(ctr=1;ctr<=2*ntot-1;ctr=ctr+2)
{
if(sum_dose[ctr]>total_max_value)
{
total_max_value=sum_dose[ctr];
}
if(sum_dose[ctr]<total_min_value)
{
total_min_value=sum_dose[ctr];
}
}

for(ctr=1;ctr<=2*ntot-1;ctr=ctr+2)
{
total_norm=100*sum_dose[ctr]/total_max_value;
for(bin=0;bin<=99;bin++)
{
if(total_norm>bin && total_norm<=(bin+1))
{
total_volume[bin]=total_volume[bin]+delta_vol;
total_vol=total_vol+delta_vol;
}
}
}
}

```

```

for(ctr=1;ctr<=2*ntot-1;ctr=ctr+2)
{
  if(raw_crit_struct[ctr]==1)
  {
    critical=1;
    if(sum_dose[ctr]>critical_max_value)
    {
      critical_max_value=sum_dose[ctr];
    }
    if(sum_dose[ctr]<critical_min_value)
    {
      critical_min_value=sum_dose[ctr];
    }
  }
}

for(ctr=1;ctr<=2*ntot-1;ctr=ctr+2)
{
  if(raw_crit_struct[ctr]==1)
  {
    critical_norm=100*sum_dose[ctr]/critical_max_value;
    for(bin=0;bin<=99;bin++)
    {
      if(critical_norm>bin && critical_norm<=(bin+1))
      {
critical_volume[bin]=critical_volume[bin]+delta_vol;
        critical_vol=critical_vol+delta_vol;
      }
    }
  }
}

for(ctr=1;ctr<=2*ntot-1;ctr=ctr+2)
{
  if(raw_rx_dose[ctr]==unif_raw_rx_dose)
  {
    if(sum_dose[ctr]>target_max_value)
    {
      target_max_value=sum_dose[ctr];
    }
    if(sum_dose[ctr]<target_min_value)
    {
      target_min_value=sum_dose[ctr];
    }
  }
}

```

```

for(ctr=1;ctr<=2*ntot-1;ctr=ctr+2)
{
  if(raw_rx_dose[ctr]==unif_raw_rx_dose)
  {
    target_norm=100*sum_dose[ctr]/target_max_value;
    for(bin=0;bin<=99;bin++)
    {
      if(target_norm>bin && target_norm<=(bin+1))
      {
target_volume[bin]=target_volume[bin]+delta_vol;
        target_vol=target_vol+delta_vol;
      }
    }
  }
}

```

```

for(ctr=1;ctr<=2*ntot-1;ctr=ctr+2)
{
  if(raw_rx_dose[ctr]==0)
  {
    if(sum_dose[ctr]>normal_max_value)
    {
      normal_max_value=sum_dose[ctr];
    }
    if(sum_dose[ctr]<normal_min_value)
    {
      normal_min_value=sum_dose[ctr];
    }
  }
}

```

```

for(ctr=1;ctr<=2*ntot-1;ctr=ctr+2)
{
  if(raw_rx_dose[ctr]==0)
  {
    normal_norm=100*sum_dose[ctr]/normal_max_value;
    for(bin=0;bin<=99;bin++)
    {
      if(normal_norm>bin && normal_norm<=(bin+1))
      {
normal_volume[bin]=normal_volume[bin]+delta_vol;
        normal_vol=normal_vol+delta_vol;
      }
    }
  }
}

```

```

printf("***** RESULTS *****
\n\n");

```

```

    printf("Desired Target Dose:    %7.3e Gy
\n\n",unif_raw_rx_dose*mu_est);
    printf("Maximum Total Dose:    %7.3e Gy
\n",total_max_value);
    printf("Minimum Total Dose:    %7.3e Gy
\n",total_min_value);

    printf("Maximum Target Dose:    %7.3e Gy
\n",target_max_value);
    printf("Minimum Target Dose:    %7.3e Gy
\n",target_min_value);

    printf("Maximum Normal Dose:    %7.3e Gy
\n",normal_max_value);
    printf("Minimum Normal Dose:    %7.3e Gy
\n",normal_min_value);

    if(critical==1)
    {
        printf("Maximum Critical Dose: %7.3e Gy
\n",critical_max_value);
        printf("Minimum Critical Dose: %7.3e Gy
\n",critical_min_value);
    }

    total_ptr=fopen("total_dvh", "w");
    for(bin=0;bin<=99;bin++)
    {
        fprintf(total_ptr, "%3d %7.3e \n", bin,
100*total_volume[bin]
        /total_vol);
    }
    fclose(total_ptr);

    target_ptr=fopen("target_dvh", "w");
    for(bin=0;bin<=99;bin++)
    {
        fprintf(target_ptr, "%3d %7.3e \n", bin,
100*target_volume[bin]
        /target_vol);
    }
    fclose(target_ptr);

    normal_ptr=fopen("normal_dvh", "w");
    for(bin=0;bin<=99;bin++)
    {
        fprintf(normal_ptr, "%3d %7.3e \n", bin,
100*normal_volume[bin]
        /normal_vol);
    }
    fclose(normal_ptr);

    if(critical==1)

```

```
{
critical_ptr=fopen("critical_dvh", "w");
for(bin=0;bin<=99;bin++)
{
fprintf(critical_ptr, "%3d %7.3e \n", bin,
100*critical_volume[bin]
/critical_vol);
}
fclose(critical_ptr);
}

return;
}
```

REFERENCES

AAPM Task Group 21. A Protocol for the Determination of Absorbed Dose from High-Energy Photon and Electron Beams. *Medical Physics*. 1983;10:741-771.

Ahnesjo A, Andreo P. Determination of Effective Bremsstrahlung Spectra and Electron Contamination for Photon Dose Calculations. *Phys. Med. Biol.* 1989;34:1451-64.

Ahnesjo A, Saxner M, Trepp A. A Pencil Beam Model for Photon Dose Calculation. *Medical Physics*. 1992;19:263-73.

Archer B, Almond P, Wagner L. Application of Laplace Transform Pair Model for High-Energy X-Ray Spectral Reconstruction. *Medical Physics*. 1985;12:630.

Barth N. An Inverse Problem in Radiation Therapy. *Int. J. Radiation Oncology Biol. Phys.* 1990;18:425-31.

Bjarngard B, Cunningham J, Comments on "Validity of the Concept of Separating Primary and Scatter Dose". *Medical Physics*. 1986;13:760-61.

Bjarngard B, Petti P. Description of the Scatter Component in Photon-Beam Data. *Phys. Med. Biol.* 1988;33:21-32.

Bjarngard B, Tsai J, Rice R. Doses on the Central Axes of Narrow 6-MV X-Ray Beams. *Medical Physics*. 1990;17:794-99.

Bohachevsky I, Johnson M, Stein M. Generalized Simulated Annealing for Function Optimization. *Technometrics*. 1986;28:209-217.

Bortfeld T, Burkelbach J, Boesecke R, Schlegel W. Methods of Image Reconstruction from Projections Applied to Conformation Radiotherapy. *Phys. Med. Biol.* 1990;35:1423-34.

Bova F. Radiation Physics. *Neurosurgery Clinics of North America*. 1990a;1:909-31.

Bova F. A Film Phantom for Routine Film Dosimetry in the Clinical Environment. *Med. Dosimetry*. 1990b;15:83-85.

Boyer A. Shortening the Calculation Time of Photon Dose Distributions in an Inhomogeneous Medium. *Medical Physics*. 1984;11:552-54.

- Boyer A, Desobry G, Wells N. Potential and Limitations of Invariant Kernel Conformal Therapy. *Medical Physics*. 1991;18:703-712.
- Boyer A, Mok E. A Photon Dose Distribution Model Employing Convolution Calculations. *Medical Physics*. 1985;12:169-77.
- Boyer A, Mok E. Calculation of Photon Dose Distributions in an Inhomogeneous Medium using Convolutions. *Medical Physics*. 1986;13:503-09.
- Boyer A, Wackwitz R, Mok E. A Comparison of the Speeds of Three Convolution Algorithms. *Medical Physics*. 1988;15:224-27.
- Brahme A. Optimization of Stationary and Moving Beam Radiation Therapy Techniques. *Radiotherapy and Oncology*. 1988;12:129-40.
- Brahme A, Roos J, Lax I. Solution of an Integral Equation Encountered in Rotation Therapy. *Phys. Med. Biol*. 1982;27:1221-29.
- Cardarelli G, Rao S, Cail D. Investigation of the Relative Surface Dose from Lipowitz-Metal Tissue Compensators for 24- and 6-MV Photon Beams. *Medical Physics*. 1991;18:282-87.
- Carol M, Targovnik H, Verhey L, Sanders M. Integrated 3-D Conformal Planning/Multivane Intensity Modulating Delivery System for Radiotherapy. *Medical Physics (Abstract)*. 1993;20:904.
- Chui C, Mohan R. Extraction of Pencil Beam Kernels by the Deconvolution Method. *Medical Physics*. 1988;15:138-44.
- Cormack A. A problem in Rotation Therapy with X Rays. *Int. J. Radiation Oncology Biol. Phys*. 1987;13:623-630.
- Cormack A, Cormack R. A problem in Rotation Therapy with X Rays: Dose Distributions with an Axis of Symmetry. *Int. J. Radiation Oncology Biol. Phys*. 1987;13:1921-1925.
- Coursey B, Goodman L, Hoppes D, Loevinger R, McLaughlin W, Soares C, Weaver J. The Needs for Brachytherapy Source Calibrations in the United States. *Nuclear Instr. and Methods in Physics Research*. 1992;A312:246-250.
- Dawson J. Estimation of Medical Linear Accelerator Bremsstrahlung Spectra. Ph.D. Dissertation. University of Missouri, Columbia. 1989.
- Desobry G, Boyer A. An Analytic Calculation of the Energy Fluence Spectrum of a Linear Accelerator. *Medical Physics (Abstract)*. 1992;19:785.

Dubois D. Digitization of Film for Quality Assurance and Dosimetry. M.S. Project. University of Florida. 1993.

Eklöf A, Ahnesjö A, Brahme A. Photon Beam Energy Deposition Kernels for Inverse Radiotherapy Planning. *Acta Oncologica*. 1990;29:447-454.

Faddegon B, Ross C, Rogers D. Angular Distribution of Bremsstrahlung from 15-MeV Electrons Incident on Thick Targets of Be, Al, and Pb. *Medical Physics*. 1991;18:727-39.

Friedman W, Bova F. Stereotactic Radiosurgery. *Contemporary Neurosurgery*. 1989a;11;no. 12:1-7.

Friedman W, Bova F. The University of Florida Radiosurgery System. *Surg. Neurol*. 1989b;32:334-42.

Friedman W, Bova F, Spiegelmann R. Linear Accelerator Radiosurgery at the University of Florida. *Neurosurgery Clinics of North America*. 1992;3:141-60.

Galvin J, Chen X, Smith R. Combining Multileaf Fields to Modulate Fluence Distributions. *Int. J. Radiation Oncology Biol. Phys*. 1993;27:697-705.

Galvin J, Smith A, Lally B. Characterization of a Multileaf Collimator System. *Int. J. Radiation Oncology Biol. Phys*. 1993;25:181-192.

Goitein M, The Inverse Problem. *Int. J. Radiation Oncology Biol. Phys*. 1990;18:489-91.

Hendrickson F, Withers R. Principles of Radiation Oncology. American Cancer Society Textbook of Clinical Oncology, Atlanta, GA, 1991:35-46.

Horton J. Handbook of Radiation Therapy Physics. Prentice-Hall Inc, Englewood Cliffs, NJ, 1987.

Holmes T, Mackie T. A Filtered Backprojection Dose Calculation Method for Inverse Treatment Planning. *Medical Physics*. 1994;21:303-313.

Holmes T, Mackie T, Reckwerdt P, Deasy J. An Iterative Backprojection Inverse Treatment Planning Algorithm for Tomotherapy. AAPM Presentation. Washington, DC. 1993.

Huang P, Kase K, Bjarngard B. Reconstruction of 4 MV Bremsstrahlung Spectra From Measured Transmission Data. *Medical Physics*. 1983;10:778-785.

Johns H, Cunningham J. The Physics of Radiology. Charles C Thomas, Publisher, Springfield, IL, 1983.

Jones S. Laser TLD 2-Dimensional Dose Mapping. Radiation Protection Dosimetry. 1993;47:457-460.

Jones S, Sweet J, Fehl D, Sujka B, Vehar D, Westfall R. Comparison of Laser and Conventional Heating in Thermoluminescence Dosimetry Dosemapping. Rev. Sci. Instrum. 1992;63:4898-4900.

Khan F. The Physics of Radiation Therapy. Williams & Wilkins, Baltimore, MD, 1984.

Kirkpatrick S, Gelatt C, Vecchi M. Optimization by Simulated Annealing. Science. 1983;220:671-680.

Koch H, Motz J. Bremsstrahlung Cross-Section Formulas and Related Data. Reviews of Modern Physics. 1959;31:920-955.

Kooy H, Barth N. The Verification of an Inverse Problem in Radiation Therapy. Int. J. Radiation Oncology Biol. Phys. 1990;18:433-39.

Kusbad S, Mackie T, Gehring M, Misco D, Paliwal B, Mehta M, Kinsella T. Monte Carlo and Convolution Dosimetry For Stereotactic Radiosurgery. Int. J. Radiation Oncology Biol. Phys. 1990;19:1027-35.

Leavitt D, Gibbs F, Heilbrun M, Moeller J, Takach G. Dynamic Field Shaping to Optimize Stereotactic Radiosurgery. Int. J. Radiation Oncology Biol. Phys. 1991;1:1-9.

Leksell L. The Stereotaxic Method and Radiosurgery of the Brain. Acta. Chir. Scand. 1951;102:316-19.

Leksell L. Stereotactic Radiosurgery. Journal of Neurology, Neurosurgery, and Psychiatry. 1983;46:797-803.

Levy L, Waggener R, McDavid W, Payne W. Experimental and Calculated Bremsstrahlung Spectra from a 25-MeV Linear Accelerator and a 19-MeV Betatron. Medical Physics. 1974;1:62-67.

Levy L, Waggener R, Wright A. Measurement of Primary Bremsstrahlung Spectrum from an 8-MeV Linear Accelerator. Medical Physics. 1976;3:173-75.

Lind B, Brahme A. Photon Field Quantities and Units for Kernel Based Radiation Therapy Planning and Treatment Optimization. Phys. Med. Biol. 1992;37:891-909.

Lo Y. Albedos for 4-, 10-, and 18-MV Bremsstrahlung on X-Ray Beams on Concrete, Iron, and Lead--Normally Incident. Medical Physics. 1992;19:659-66.

Mackie T, Bielajew A, Rogers D, Battista J. Generation of Photon Energy Deposition Kernels Using the EGS Monte Carlo Code. *Phys. Med. Biol.* 1988;33:1-20.

Mackie T, Holmes T, Swerdloff S, Rockwendt P, Deasy J, Yang J, Paliwal B, Kinsella T. Tomotherapy: A New Concept for the Delivery of Dynamic Conformal Radiotherapy. *Medical Physics.* 1993;20:1709-1719.

Mackie T, Scrimger J, Battista J. A Convolution Method of Calculating Dose for 15-MV X Rays. *Medical Physics.* 1985;12:188-96.

McCormick N. Inverse Radiative Transfer Problems: A Review. *Nuclear Science and Engineering.* 1992; 112:185-198.

McLaughlin W, Yun-Dong C, Soares C, Miller A, Van Dyk J, Lewis D. Sensitometry of the Response of a New Radiochromic Film Dosimeter to Gamma Radiation and Electron Beams. *Nuclear Instr. and Methods in Physics Research.* 1991;A302:165-176.

Metropolis N, Rosenbluth A, Rosenbluth M, Teller E. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics.* 1953;21:1087-1092.

Mohan R, Chui C, Reply to Comments of Bjarngard and Cunningham. *Medical Physics.* 1986;13:761-62.

Mohan R, Chui C, Lidofsky L. Energy and Angular Distributions of Photons from Medical Accelerators. *Medical Physics.* 1985;12:592-97.

Mohan R, Chui C, Lidofsky L. Differential Pencil Beam Dose Computation Model for Photons. *Medical Physics.* 1986;13:64-73.

Morrill S, Lam K, Lane R, Langer M, Rosen I. Very Fast Simulated Reannealing in Radiation Therapy Treatment Plan Optimization. *ASTRO Annual Meeting.* 1993.

Moss D. Conformal Stereotactic Radiosurgery with Multileaf Collimation. Ph.D. Dissertation. University of Florida. 1992.

Muench P, Meigooni A, Nath R, McLaughlin W. Photon Energy Dependence of the Sensitivity of Radiochromic Film and Comparison with Silver Halide Film and LiF TLDS Used for Brachytherapy Dosimetry. *Medical Physics.* 1991;18:769-775.

Näfstadius P, Brahme A, Nordell B. Computer Assisted Dosimetry of Scanned Electron and Photon Beams for Radiation Therapy. *Radiotherapy and Oncology.* 1984;2:261-69.

Nath R, Schulz R. Determination of High-Energy X-Ray Spectra by Photoactivation. *Medical Physics*. 1976;3:133-140.

Nizin P, Chang X. Primary Dose in Photon Beams with Lateral Electron Disequilibrium. *Medical Physics*. 1991;18:744-48.

Park C. Interactive Microcomputer Graphics. Addison-Wesley Publishing Company, Reading, MA, 1985.

Podgorsak E, Olivier A, Pla J, Lefebvre P, Hazel J. Dynamic Stereotactic Radiosurgery. *Int. J. Radiation Oncology Biol. Phys.* 1988;14:115-26.

Pratt W. Digital Image Processing. John Wiley & Sons Inc., New York, NY, 1978.

Press W, Teukolsky S, Vetterling W, Flannery B. Numerical Recipes in C. The Art of Scientific Computing. Cambridge University Press, Cambridge, NY, 1992.

Sauer O, Neumann M. Reconstruction of High-energy Bremsstrahlung Spectra by Numerical Analysis of Depth-Dose Data. *Radiotherapy and Oncology*. 1990;18:39-47.

Sayeg J, Gregory R. A New Method for Characterizing Beta-Ray Ophthalmic Applicator Sources. *Medical Physics*. 1991;18:453-461.

Sharpe M, Battista J. Orientation of Dose Spread Kernels in X-Ray Beam Dose Calculations using Convolution/Superposition principles. 1992 AAPM annual meeting. *Medical Physics*. 1992;19:784.

Sixel K, Podgorsak E, Souhami L. Cylindrical Dose Distributions in Pseudodynamic Rotation Radiosurgery: An Experimental Study. *Medical Physics*. 1993;20:163-170.

Soares C. Calibration of Ophthalmic Applicators at NIST: A Revised Approach. *Medical Physics*. 1991;18:787-793.

Soderstrom S, Brahme A. Selection of Suitable Beam Orientations in Radiation Therapy Using Entropy and Fourier Transform Measures. *Phys. Med. Biol.* 1992;37:911-924.

Soderstrom S, Brahme A. Optimization of the Dose Delivery in a Few Field Techniques Using Radiobiological Objective Functions. *Medical Physics*. 1993;20:1201-1210.

Spiegelmann R, Friedman W, Bova F. Limitations of Angiographic Target Localization in Planning Radiosurgical Treatment. *Neurosurgery*. 1992;30:619-24.

Starkschall G. A Convolution Method for Constructing Primary Beam Profiles in the Presence of Beam Modifiers. Medical Physics. 1988;15:657-61.

Suh T. Optimization of Dose Distribution for the System of Linear Accelerator Based Stereotactic Radiosurgery. Ph.D. Dissertation, University of Florida. 1990.

Walker M, Puhl J, Soares C, Humphreys J, Coursey B, McLaughlin W. Precision Source Profiling Techniques for Ionizing Radiation Sources. Proceedings RadTech. 1992.

Webb S. Optimisation of Conformal Radiotherapy Dose Distributions by Simulated Annealing. Phys. Med. Biol. 1989;34:1349-1370.

Webb S. Optimization by Simulated Annealing of Three-Dimensional Conformal Treatment Planning for Radiation Fields Defined by a Multileaf Collimator. Phys. Med. Biol. 1991a;36:1201-1226.

Webb S. Optimization of Conformal Radiotherapy Dose Distributions by Simulated Annealing: 2. Inclusion of Scatter in the 2 D Technique. Phys. Med. Biol. 1991b;36:1227-1237.

Winston K, Lutz W. Linear Accelerator as a Neurosurgical Tool For Stereotactic Radiosurgery. Neurosurgery. 1988;22:454-64.

Zhu Y, Van Dyk J. Accuracy Requirements of the Primary X-ray Spectrum in Dose Calculations using FFT Convolution Techniques. AAPM Poster Presentation. Washington, DC, 1993.

BIOGRAPHICAL SKETCH

Joseph F. Harmon Jr. was born in Charleston, SC, in 1960. He received a Bachelor of Science degree in physics with honors from The Citadel in 1982. He was immediately commissioned as an officer in the U.S. Air Force and began his military career by earning a Master of Science in Nuclear Science at the Air Force Institute of Technology located at Wright Patterson Air Force Base, OH. During the next four years, he served as a Nuclear Research Officer for the Air Force Technical Applications Center at Patrick Air Force Base, FL, where he was primarily responsible for analyzing atomic energy detection system data collected worldwide. He was then selected for a teaching assignment in the Physics Department of the Air Force Academy in Colorado Springs, CO. During his three years at the Academy, he directed and taught a variety of courses and was promoted to Assistant Professor of Physics.

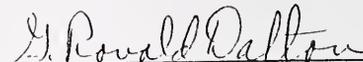
Sponsored by the Air Force Academy, he has spent the last three years earning the degree of Doctor of Philosophy in the Nuclear Engineering Sciences Department of the University of Florida exercising the Medical Physics option. Joseph Harmon is married to Rose Milligan Harmon and they have three children named Ashley, Sarah, and Taylor.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Frank J. Beva, Chair
Associate Professor of Nuclear
Engineering Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



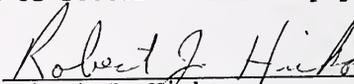
George R. Dalton
Professor of Nuclear
Engineering Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



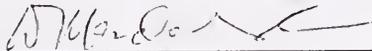
David E. Hintenlang
Assistant Professor of Nuclear
Engineering Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



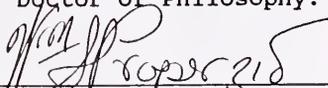
Robert J. Hirko
Associate Engineer of Aerospace
Engineering, Mechanics, and
Engineering Science

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



William M. Mendenhall
Professor of Radiation Oncology

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



William S. Properzio
Associate Professor of Nuclear
Engineering Sciences

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

April 1994


fu Winfred M. Phillips
Dean, College of Engineering

Karen A. Holbrook
Dean, Graduate School

