Masters Theses   Graduate School

5-1999

Use of Positron Emission Tomography (PET) to improve radiation dosimetry for Boron Neutron Capture Therapy (BNCT)

Mohammad Khurram Khan

Recommended Citation

To the Graduate Council:

I am submitting herewith a thesis written by Mohammad Khurram Khan entitled "Use of Positron Emission Tomography (PET) to improve radiation dosimetry for Boron Neutron Capture Therapy (BNCT)." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Nuclear Engineering.

Laurence F. Miller, Major Professor

We have read this thesis and recommend its acceptance:

Lawrence W. Townsend, Peter G. Groer
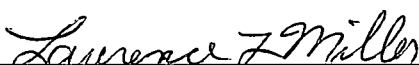
Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

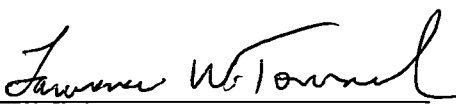(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Mohammad Khurram Khan entitled "Use of Positron Emission Tomography (PET) to Improve Radiation Dosimetry for Boron Neutron Capture Therapy (BNCT)." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Masters of Science, with a major in Nuclear Engineering.
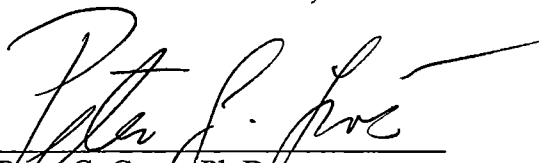
Laurence F. Miller
Laurence F. Miller, Ph.D., Major Professor

We have read this thesis
and recommend its acceptance:

Lawrence W. Townsend
Lawrence W. Townsend, Ph.D.

Peter G. Groer
Peter G. Groer, Ph.D.

Accepted for the Council:

Lew Minkel
Associate Vice Chancellor and
Dean of The Graduate School

Use of Positron Emission Tomography (PET) to Improve Radiation Dosimetry for Boron Neutron Capture Therapy (BNCT)

A Thesis
Presented for the
Master of Science
Degree
The University Of Tennessee, Knoxville

Mohammad Khurram Khan
May, 1999

DEDICATION

This thesis is dedicated to my parents:

Mr. Muhammad Nafees Khan

And

Mrs. Maryum Un-Nisa Khan

who have instilled in me a desire to learn, and have given me more love and support than could be deemed possible.

ACKNOWLEDGMENTS

Behind any major effort are the many people who contribute significantly to the understanding of your research. Without them, the task at hand would be very difficult. First, I would like to thank god for giving me the ability to learn, the opportunity to challenge myself, and for guiding me in the right direction. Next, I would like to thank my parents for their love and support. Their words of encouragement have always kept me motivated.

Next, I would like to thank my advisor, Dr. Laurence F. Miller whose technical support, words of encouragement, and guidance has been invaluable. I will always cherish the many afternoons when we sat down in his office and talked about many issues, some research oriented and some not so research oriented. He gave me the opportunity to explore my research interests, work independently, and develop confidence in my work.

In addition, I would like to thank Dr. George W. Kabalka and Dr. Trent F. Nichols. Dr. Kabalka provided not only the funds, he also helped me gain insight to other current issues and research developments in BNCT. Throughout the research, I had the opportunity to work as a volunteer directly under Dr. Nichols. Dr. Nichols always provided words of encouragement. While working for Dr. Nichols as a researcher and as a volunteer in his clinic, I had the opportunity to meet directly with some of the patients whose data is presented in this research. For this, I am grateful.

Perhaps nothing else can be as conducive to one's own research than personally meeting the patients and the families who are directly affected by your research. I will never forget Dr. Nichols words, "[jokingly] are you sure you want to do this?" whenever we talked about my interests in pursuing medical school and medical research as a future career. He has been very helpful in providing me with the patient history and keeping me updated on the patients' well-being pre and post BNCT.

I would also like to thank Hedley Bond and Shannon Campbell. Hedley and Shannon have always taken time out of their busy schedule to explain me the IPL software necessary for my research.

I would also like to thank Dr. Karl Hubner, Dr. Garry T. Smith, and Dr. Joe Thei. I would also like to thank Dr. Jon Dyke, Dr. Dan Wessol, and Chuck Wimperle. Dr. Jon Dyke has been very helpful in showing me "the ropes" on how to setup treatment planning using BNCT_RTPE. Dr. Dan Wessol and Chuck Wimperle have contributed to my understanding of BNCT_RTPE outputs that are not so well documented in the user's manual. I would also like to thank Winston Holmes, Mark Wyatt, and Abul-Rahim for all of their help. In addition I would like to thank Dr. Jacek Kapala for sharing the BNL protocol information. Finally I would like to recognize my committee members, Dr. Lawrence Townsend and Dr. Peter G. Groer for reviewing my work.

v

ABSTRACT

The purpose of this work is to enhance dosimetry for BNCT by incorporating positron emission tomography (PET) into treatment planning. Currently, the INEEL code, BNCT Radiation Treatment Planning Environment (BNCT_RTPE) is used by some research institutions to determine patient specific dose to and around the tumor. Non-uniform rational B-spline (NURBS) geometry with Monte Carlo techniques are used to generate the three dimensional geometry and to determine dosimetry based on magnetic resonance imaging (MRI) and/or computed tomography (CT) [9, 13]. During treatment planning homogeneous distribution of boron concentration in tumor cells and normal brain cells is assumed to be constant throughout the brain in BNCT_RTPE.

A technique has been developed that allows for the incorporation of detailed patient specific boron biodistribution provided by PET to post process the dose based on the homogeneous distribution used in BNCT_RTPE. The method allows for three dimensional geometrical reconstruction with PET in BNCT_RTPE followed by peak to average scaling of the boron dose. Differences between the BNCT_RTPE boron dose and PET adjusted boron dose are presented. Results of the dose adjustment indicate remarkable differences between the conventional way of determining the dose and the PET adjusted boron dose, especially around the boundaries of the tumor. Based on data in 3 patients and the analysis done using BNCT_RTPE, the differences between conventional approach of radiation treatment planning and PET based treatment planning are as much as 20 – 80 %. Much of the differences indicate an

over prediction by conventional methods in the amount of dose predicted in and around the tumor boundary regions. In conclusion, the use of PET data is shown to enhance existing methods for predicting the dosimetry for BNCT in existing codes. Improvements in dosimetry and the pharmacokinetic information provided by PET taken together with better boron compounds can significantly improve the efficacy of BNCT.

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

1. Boron Neutron Capture Therapy Fundamentals

Each year, about 7000 Americans are diagnosed with a deadly brain cancer, glioblastoma multiforme, with only about 3% surviving past 5 years [1, 2]. Diagnosis for glioblastoma multiforme is like an immediate death sentence with very little hope for survival. However, recent developments in radiation therapy have provided some hope for those patients. One of the tools currently seen as promising for fighting the aggressive multiforme glioblastoma as well as other forms of cancer is referred to as Boron Neutron Capture Therapy (BNCT). Boron Neutron Capture Therapy (BNCT) is a novel form of targeted radiotherapy that uses epithermal or low energy neutrons to activate boron containing pharmaceuticals specifically delivered to the tumor site. Four years after the initial discovery of the neutron in 1932 by Cambridge's J. Chadwick, G.L. Locher of Franklin Institute of Pennsylvania introduced the concept of neutron capture therapy (NCT) [1, 2]. The beauty of NCT lies in its simple binary modality. The objective is simple: to get B-10 localized in the tumor in sufficient concentration and then radiate the tumor with a low energy neutron beam [1, 2]. Since B-10 has a high probability of capturing the neutron at low energies, capturing of the neutron by the B-10 compound results in an excited B-10 compound (B-11). The excited B-10 compound then decays with a very short half-life by emitting a .84 MeV Li-7 particle, a 1.47 MeV alpha particle, and a .48 MeV photon, as shown in the following reaction [http://www.mit.edu:8001/people/flavor/intro.html]:

$$^{10}\text{B} + {^1}\text{n} \text{ -------------} \rightarrow {^{11}}\text{B} \rightarrow \text{--------------} \rightarrow {^7}\text{Li} + {^4}\text{He} + 2.34 \text{ Mev} + 0.48 \text{ MeV } \gamma$$
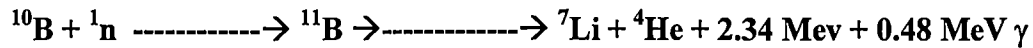
Most of the ionizing events are known to occur within about 7 microns range, or within cells' diameter of the fission event. Given the large mass and energy of the ions, they primarily travel in a linear fashion causing highly ionizing events at closely spaced intervals along their tracks. The resulting ionizing events and energy released during fission can severely damage the tumor cell's DNA via direct and indirect interactions [2]. Shown in Figure 1 is a simple representation of the fissioning event as it is expected to occur in a cell. With the tumor cell's DNA severely damaged, the chances of the cell surviving and reproducing after BNCT is slim to none. Some cells will exhibit immediate death from the radiation injury, others will exhibit a "reproductive" death. The selectivity, simplicity, and elegance of BNCT are what make BNCT a possible candidate for therapy.

In 1951, Sweet and his colleagues first demonstrated in animals that certain B-10 compounds somewhat concentrated in human brain tumor cells relative to normal brain cells [3]. Shortly after, clinical trials in 1951 and 1952 were begun at
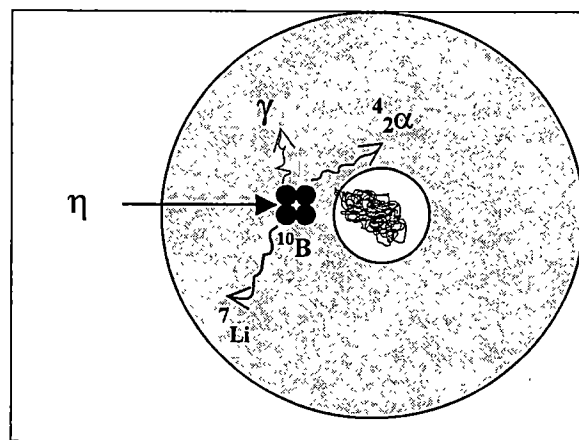
Figure 1: BNCT Cell Death Killing Mechanism

Brookhaven National Lab (BNL) and in 1961 and 1962 at Massachusetts Institute of Technology (MIT) research reactor. B-10 compound used during these early trials was sodium tetraborate (or borax, $Na_2B_4O_7 \cdot 10H_2O$). Clinical trials were also conducted in Japan in the 60's. Unfortunately, early clinical trials failed because of two reasons: (1). Thermal neutrons used in the early trials only penetrated about 3-4 cm in tissue, thus deep suited tumors were spared, and (2) Sodium Tetraborate and other early compounds were diffuseable, low-molecular weight, and those compounds that did selectively concentrate in the tumor cell had large background blood values. Hence, normal cells along with vital brain centers were radiated in larger amounts along with the tumor cells [2]. Combined with thermal beam impurities, deep situated tumors, and diffuseable and low selectivity of the compounds used in the earlier trials, the early clinical trials failed miserably. Patient health may have been jeopardized. Since 1968, over 150 patients with high-grade glioblastomas have been treated via BNCT in Japan. Reports of long-term survival in selected patients with confirmed high-grade glioblastomas in Japan has spurred a keen interest in BNCT in the US and elsewhere around the world.

Currently, BNCT is in a PhaseI/II trial period in the U.S. The objectives of the trials are: to evaluate the safety of a stepwise increase in BNCT dose to the normal brain using p-Boronophenylalanine (BPA) and epithermal neutrons, to evaluate any diverse effects of BNCT at each dose level, and to evaluate the therapeutic effectiveness of each BNCT dose level in patients with glioblastoma multiforme.

3

Shown in Figure 2 is the molecular structure of *para*-Boronophenylalanine, currently

used for some patients undergoing BNCT PhaseI/II clinical trials in the U.S.

Elsewhere around the world in places like Japan and Europe, researchers are

investigating BSH as a possible candidate for BNCT [4,5].

$$\overset{\oplus}{N}H_3$$
$$HC\text{-}COO^{\ominus}$$
$$CH_2$$

$$B(OH)_2$$

BPA
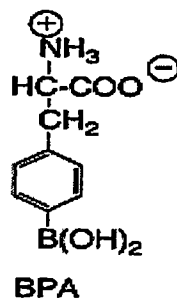
Figure 2: Molecular Structure of Boronophenylalanine

As can be seen from the molecular nature of BPA, BPA is an amino acid

analog of a naturally occurring essential amino acid in the body, phenylalanine. The

thought process behind the uptake of Boronophenylalanine is that since actively

growing tumor need amino acids for protein synthesis, Boronophenylalanine is more

likely to concentrate in those cells [5,6]. Actual patient debulking data during BNCT

have shown Boronophenylalanine to concentrate around 3.5-4 times more in tumor

cells than in normal brain cells and the surrounding blood [5,6].

Results from initial dose escalation studies within the US have demonstrated that

BNCT doses used to date are safe, patient survival after a single BNCT dose is

slightly better than survival after six weeks of standard radiation therapy with or without chemotherapy, and patient's quality of life after BNCT is better than with standard therapy [4, 6]. Current results from BNCT have also shown that the average survival time with BNCT to be around 13.5 months with average recurrence time of 6 months post debulking and BNCT [5]. However, even though progress has already been shown in some cases, much is yet to be understood. Causes of recurrence are many. Perhaps better beam filter designs, purification of the neutron beam, better beam alignment, better understanding of the dosimetry, and further development of better pharmaceuticals and the kinetics associated with the pharmaceuticals can prolong the mean survival time past the 13.5 months currently seen in the early clinical trials. The objective of this work is to address better beam alignment methods and to improve current methods of predicting the dose delivered to the tumor during irradiation.

2. Current BNL protocol

Currently, when an individual is diagnosed for glioblastoma and recognized as a candidate for BNCT, the patient is sent to either the MIT or the BNL research reactors for treatment. A cross sectional view of the BNL reactor is shown in Figure 3.

According to the BNL protocols, the patient must be 18 years or older, the tumor must not be deeper than 6.0 cm from the surface of scalp, have no history of phenylketoneuria, Karnofsky performance must be 70 or greater, and the

5
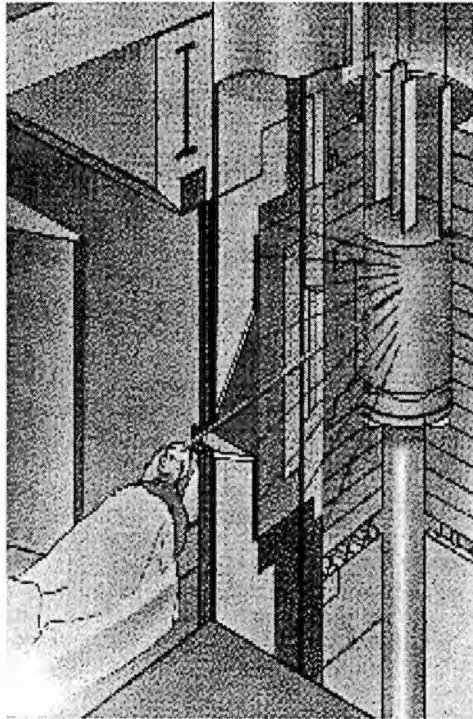
Figure 3: Cross Sectional View of BNL Research Reactor in Upton, N.Y.
(* Picture reprinted from BNL website: http://www.bnct.bnl.gov/clinical.htm).

patient must be a U.S citizen [http://www.bnct.bnl.gov/clinical.htm].

Phenylketoneuria is a disease of the kidneys in which normal breakdown of the amino

acid phenylalanine is dysfunctional due to a mutation in a specific enzyme.

Karnofsky index in a measure of patients' well being and the patients' ability to

survive the aggressive treatment. The BNL procedure involves a CT scan or an MRI

scan within one week of BNCT followed by a 2-hour intra-venous (i.v) infusion of

290 mg/kg of body weight of BPA. Following infusion, the patient is placed in front

of the BNL reactor and exposed to an epithermal neutron beam for about 30-40

minutes for each of the two fields (for deep situated tumors only). However, for

shallow tumors, the BNL protocol allows for only one field lasting for about 40-50

minutes [http://www.bnct.bnl.gov/clinical.htm]. The BNL reactor operates at a

maximum power level of 3-megawatt (thermal). The two main difficulties

encountered in the earlier1954 BNCT clinical trials were lack of neutron beam purity

and an inability to localize the boron compound at the tumor site [2,4].

The beam from a nuclear reactor is composed primarily of photons (gamma rays),

fast neutrons, and epithermal neutrons. The boron at the tumor site does not absorb

many of the fast neutrons because the cross section of the boron nucleus is inversely

proportional to the velocity of the neutron. However, fast neutrons induce tissue

damage due to direct collisions and interactions with lighter nuclei. Since the

reaction with the boron nuclei is inversely related to the velocity of the neutron, a

high flux of lower energy epithermal neutrons is necessary to increase capture

reactions and to reduce surface irradiation. An epithermal neutron source has the

added advantage over the thermal neutrons used in earlier trials in that they can

penetrate deeper into the brain. As a result of penetration and hydrogen recoil

interactions with surface tissue, the epithermal neutrons slow down to thermal

neutron energies, hence increasing the probability of interaction with the boron

compound located in deep situated tumors. The neutron spectrum of the 8-cm disk

source used by Brookhaven National Lab for treatment planning in some patients is

shown in Figure 4. It is this neutron spectrum that is used for treatment planning in

this thesis. The source consists of 47 neutron energy groups over the range of .1 eV

to 17.3 MeV, and 20 gamma energy groups over the range of 10 keV to 14 MeV, as

shown in Figure 4. A linear interpolation is assumed within each group. Each

neutron and gamma energy group has an associated angular distribution, given by 10
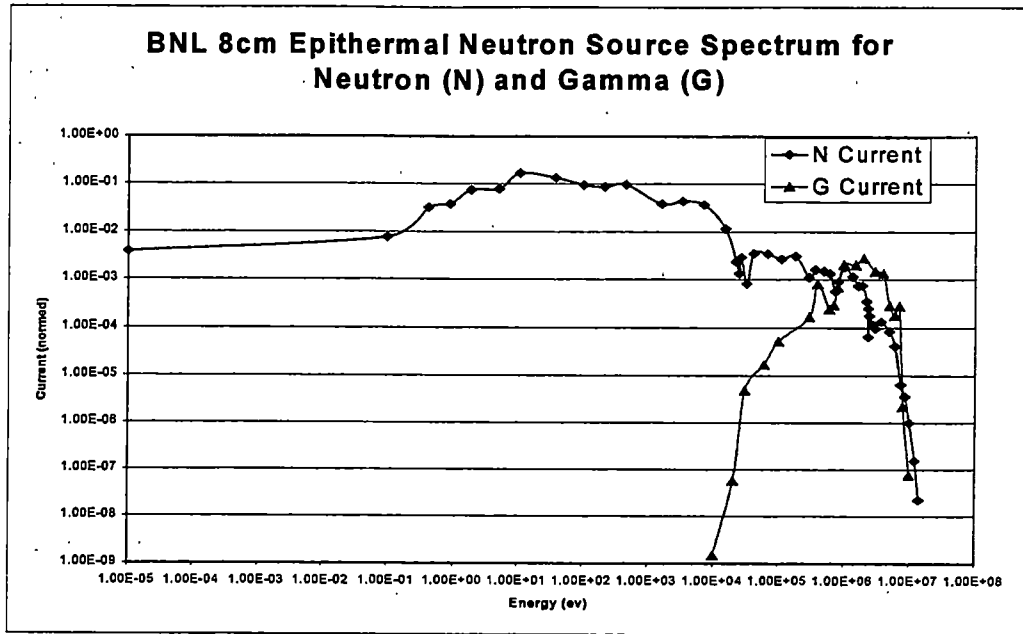
Figure 4: BNL 8 cm Epithermal Neutron Source Spectrum

equally probable cosine angles. The source is defined as an 8-cm disk, representing

the exit boundary of the filter.

Although the boron has the highest capture cross section in a BNCT therapy

situation, the concentration of nitrogen and hydrogen in the body is very high and can

also contribute significantly to the total dose delivered to the tumor. The total dose is

the sum of the individual components contributing to the dose adjusted by a

weighting factor, as shown in the following equation:

$$\text{TotalDose} = A(x)*^{10}B+B(x)*^{14}N+C(x)*\gamma+D(x)*^{1}n \text{ (fast neutron).} \qquad (1)$$

where,

$A(x)$, $B(x)$, $C(x)$, and $D(x)$ are the weighting factors and ^{10}B, ^{14}N, γ, and fast neutron,

^{1}n, are the principal components of the dose that contribute to the total dose during

therapy. Table 1 contains a summary of other possible reactions that might occur

within the body during therapy [9, 10]:

Table 1: Expected Reactions Caused by Thermal Neutrons in Tissue

THERMAL CROSS SECTION

Element	Natural Abundance (%)	Reaction	Microscopic σ (Barns)	Macroscopic Σ (cm^-1)	Released Engergy per Reaction (MeV)
H-1	99.985	1H(n,γ)2H	0.332	1.99 X 10^-2	2.22
H-2	0.0145	2H(n,γ)3H	0.0005	4.34 X 10^-9	6.25
C-12	98.892	12C(n,γ)13C	0.0034	2.52 X 10^-5	4.95
C-13	1.11	13C(n,γ)14C	0.0009	7.48 X 10^-8	8.15
N-14	99.635	14N(n,ρ)14C	1.81	2.70X10^-3	0.63
N-15	0.365	15N(n,γ)16N	0.00024	1.31X10^-10	2.48
O-16	99.795	16O(n,γ)17O	0.0003	4.34X10^-6	4.14
O-17	0.037	17O(n,α)14C	0.24	2.39X10^-6	1.82
O-18	0.204	18O(n,γ)19O	0.00021	1.15X10^-8	3.95
B-10	19.61		3837		2.79
		10B(n,α)7Li	93.70%	4.33X10^-3	(γ = .478)
		10B(n,α)7Li	6.30%	2.91X10^-4	(no γ)

3. Monte Carlo Techniques

The equation describing the interaction of the radiation source with an arbitrary volume is known as the transport equation, or the linearized Boltzman transport equation named in honor of L. Boltzman who first derived it over a century ago [11]. For neutrons, the transport equation characterizing scattering, fissioning, and absorption reactions are shown in Figure 5.

Throughout history, several methods have been developed to solve the Boltzman transport equation describing the motion of particles through a given volume. At least two methods have been developed to solve the transport equation: Discrete Ordinates and Monte Carlo. Although Discrete Ordinates method can be used to solve the transport equation and to model the geometry, the technique is

$$\frac{1}{\bar{v}}\frac{\partial\phi}{\partial t}+\bar{\Omega}\bullet\Delta\Phi+\Sigma_t(\bar{r},E)\phi(\bar{r},E,\bar{\Omega},t)-\iint\Sigma_s(\bar{r},E'\to E,\bar{\Omega}\to\Omega)\phi(\bar{r},E',\bar{\Omega},t)\partial E'\partial\Omega'$$

$$=\frac{1}{4\pi}\chi\iint\nu(\bar{r},E')\Sigma_f(\bar{r},E')\phi(\bar{r},E',\bar{\Omega},t)\partial E'\partial\bar{\Omega}'+S(\bar{r},E,\bar{\Omega},t)$$

Where,

    E = Neutron energy (J)

    $\bar{r}$ = Position of Neutron (cm)

    t = Time (s)

    $\bar{v}$ = Neutron Velocity (cm s^{-1})

    S = External Neutron Source (cm^{-3} s^{-1})

    ν = Mean Number of Neutrons Produced Per Fission

    Σ_a = Macroscopic Fission Cross Section (cm^{-1})

    Σ_s = Macroscopic Scattering Cross Section (cm^{-1})

    Σ_f = Macroscopic Fission Cross Section (cm^{-1})

    Σ_t = Macroscopic Total Cross Section (cm^{-1})

    χ = Normalized Fission Spectrum

    ϕ = Forward Neutron Fluence Rate (cm^{-2} s^{-1})

    $\bar{\Omega}$ = Unit Direction Vector

    Δ = Gradient Operator

Figure 5: Boltzman Neutron Transport Equation for an Arbitrary Volume

cumbersome when it comes to generating the details of the three dimensional geometry needed for medical purposes, details of which are easier to implement in Monte Carlo. Monte Carlo is also recognized to obtain correct results when statistical criteria are satisfied and the input correctly represents the problem under study.

The Monte Carlo technique is a statistical approach. It was first used in the early 18th century. The development of the method is contributed to the French historian and mathematician, Georges Louis Leclerc Comte de Buffon, who in 1768 performed an experiment in his house. In the experiment he took loaves of bread of length L and threw them over his shoulder. He then looked to see if they crossed the lines on his tile floor, which had a spacing D. For L less than or equal to D, he obtained that the probability that the loaf intersected the tile is given by:

$$P = 2L/(P_i * D) \qquad (2)$$

If the loaves are dropped N times and counted R intersections then the total probability would be:

$$P = R/N; \qquad (3)$$

With this, one can estimate P_i through statistical methods:

$$P_i = 2LN/(RD) \qquad (4)$$

A key asset of this method is the fact that the absolute error of the estimate decreases as $N^{-1/2}$, where N is the number of times that the procedure is repeated. Thus a good estimate for P_i can be estimated if one repeats the experiment enough times. Perhaps the real use of Monte Carlo technique was shown in 1948 when

11

Fermi, Metropolis, and Ulam obtained a solution of the Shroedinger wave equation with this method. Monte Carlo technique is also used for macrodosimetry simulations and patient treatment planning in this work.

4. PET Imaging

The problem of in-vivo localization of ^{10}B is not trivial and current medical diagnostic tools are required. Positron Emission Tomography (PET) is a medical diagnostic tool that uses a radioactive tracer and filtered back projection image reconstruction to determine where the tracer localizes. For BNCT purposes, a positron emitting radioisotope like ^{18}F is attached to para-Boronophenylalanine and imaged using a dual detector response system. The digital data is then reconstructed using filtered backprojection to generate an analog image of the tracer. What lights up with the F-18 tagged BPA are the active areas of the brain that require the amino acid phenylalanine. An ^{18}F-BPA PET image for a patient having undergone debulking prior to BNCT treatment at BNL is shown in Figure 6.

As can be seen from Figure 6, that which lights up most intensely is that part of the brain that is actively undergoing proliferation, as is the case for actively metabolizing tumors. As can also be seen from the post surgery PET, not all of the tumor was completely removed during surgery. It should also be pointed out that the PET study shown in Figure 6 was taken shortly after a surgical procedure based on MRI. It is this part of the tumor that is not clearly evident to the surgeon based on the MRI, hence it resulted in failure to resect all of the tumor. According to Dr. Nichols at University of Tennessee Knoxville Medical Center (UTMCK), it is this remaining

12

portion of the tumor that may be causing regrowth in post BNCT patients. Thus the advantages provided by PET are not just limited to the quantitative measurements that it provides to the physicist, it can also be used to greatly optimize surgical procedures as is clearly evident in this case.
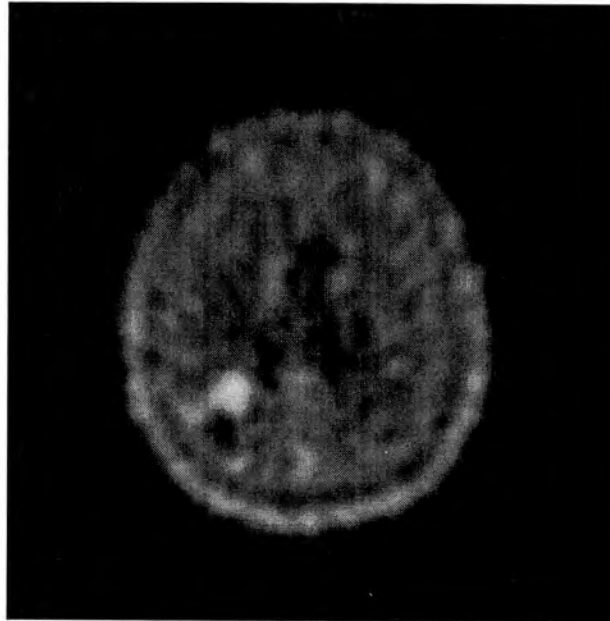
Figure 6: ^{18}F-BPA labeled PET for a Patient Having Undergone Surgery for BNCT

Essentially PET utilizes a positron emitting isotope that is administered intra-venously. The tracer, ^{18}F-BPA, localizes to certain regions of the brain and emits a positron. The positron travels within a few microns before an annihilation event occurs in which the positron collides with an electron and causes production of two photons. The annihilation event results in the emission of two 511 keV photons emitted at roughly 180 degrees apart. Coincidence detectors are used to detect the emitted photons. The coincident data obtained allows for the determination of the position of the annihilation event. The data is then back-projected to reconstruct the

image. The process involves normalization, back projection, decay correction, scatter correction, and filtering to produce the standard image, as shown in Figure 6 above. The filtered back projection method, as derived by Radon in 1917, can be described as:

$$f_{bp}(x, y) = \int_0^\pi m(x\cos\theta, y\sin\theta)\,d\theta \qquad (5)$$

where

x and y are the location of the annihilation event

θ is the angle at which the detector is located along the line of the image

and m is the mapping function that maps the annihilation event to the location on the image.

5. Problem Definition

In current BNL clinical trials, tissue samples are taken in minute amounts at possibly two or more time points. These samples are then analyzed via direct emission spectroscopy (DES) to provide the quantitative concentration levels of B-10 localized in the tumor. This information is then used in dosimetry planning to make a homogeneous assumption about the concentration throughout the tumor volume [http://www.bnct.bnl.gov/clinical.htm].

Measurements of boron concentration in the blood are easily taken, and an accurate time activity curve (TAC) is provided using direct venous sampling. The information about the B-10 concentration in the different regions of the brain during treatment planning is derived from a few closely spaced time points gained from

14

invasive tissue sampling of the deep-seated tumor itself during debulking [http://www.bnct.bnl.gov/clinical.htm]. The invasive procedure is extremely expensive and provides very minimal amount of data in exchange for the risks associated with intra-cranial biopsy. The risks associated with intra-cranial biopsy can include infections (meningitis and/or encephalitis), cerebra-vascular accident due to CSF leakage, reduced endo-cranial pressure, stroke, focal neuro-logical deficit, and other serious complications [12]. In addition to the associated risks, the procedure can result in non-uniform sampling, hence defeating the purpose of the sampling. The need for more accurate estimation of boron concentration in both the spatial and time domains is essential to accurate and predictive treatment planning.

Standard radiotherapy involves determination of the location of the tumor including the shape and volume in vivo. BNCT is dependent on this information as well as the specific concentration of the boron compound distribution within this volume. It has been shown that to be effective, there must be at least 10^9 boron atoms concentrated at each tumor cell given a neutron fluence of 10^{12} neutrons/cm^2 [2]. For treatment planning, it is assumed that boron has the same concentration in like tissue. This assumption is known to be incorrect based on data in numerous pathology articles addressing the composition and histological heterogeneity of cellular enviroments found in typical glioblastoma multiforme tumors [12,13]. For example the homogeneous assumption is that there is a constant 3.5:1 ratio of boron being localized inside the tumor cell relative to the normal brain cell regardless of the location of the cell within the volume and the local environment of the cell [5,6]. For

example, this assumes that if there were 500 cells with two of them being tumor cells located anywhere within this population, both of these cells would have 3.5 times the selective uptake of the boron relative to the background. Thus the dose to those two cells would be increased linearly by this factor. According to the argument presented during a conversation by a BNL physicist, Jacek Kapala, he points out that this is the case currently assumed under the BNL, and that PET is a cellular density map and is not representative of the boron within the cell. However, the author feels that this argument fails to hold based on several reasons.

For one, several studies have shown that tumor cells themselves are heterogeneously distributed throughout any volume and have a heterogeneous environment [9, 11, 12]. Figure 7 shows the glioblastoma cells as indicated by the dark spots to be heterogeneously distributed throughout the volume. Thus each cell can have various "localized" enironments. Thus, even if the 3.5:1 ratio is correct, this does not allow for the heterogeneous distribution of the cells themselves and their localized environment to be modeled into the treatment planning.
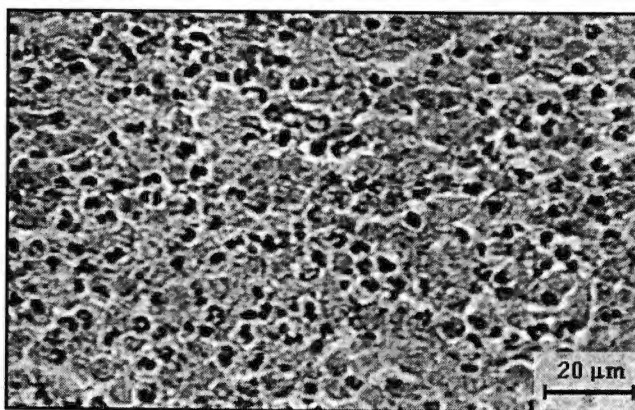
Figure 7: Characteristic Dense Cellularity of GBM.
*Image Reprinted from MIT Website
(http://www.mit.edu:8001//people/flavor/intro.html)

In addition, this assumption does not allow for disintegration in one cell to effect the killing of a neighbor cell. For example, if BPA localizes in the endoplasmic reticulum and/or the nucleus, and for a given cell diameter of 10 micron and the alpha particle track being around 7 microns in length, it is possible that the alpha particle emitted in one cell can readily kill an adjacent cell. This would be true especially if the cells' shape is rather elliptical, not spherical. Thus killing of tumor cells spaced closely together would be enhanced by nearest neighbor phenomenon. In addition, the homogeneous assumption also leads to the notion that background diffusion of blood is constant throughout the volume for each cell regardless of the location of the cell in respect to the blood supply and the localized microscopic environment of the cell. For example, if one of the tumor cells is placed close to a capillary or some other blood pool supply while another is placed much further away from the blood supply. Then it is highly improbable that both of the cells will have the same ratio of boron within the cell. In addition cells closer to the blood supply may be less in competition for the nutrients then those much further away from the blood supply. Thus it is probable that those cells much close to the blood supply may even contain higher levels of boron then those further away from the blood supply. This is not accounted for by the homogeneous assumption.

Thus, the use of the homogeneous assumption in existing codes leads to inaccurate dose contours for treatment planning in BNCT. The homogeneous distribution of boron in the tumor volume can significantly affect the boron capture

17

reaction component of the total dose, and hence jeopardize the overall treatment. It is the inhomogeneity of the tumor cells that cannot be determined by a few biopsies taken during debulking of the tumor, but can significantly affect the treatment planning and efficacy of BNCT. However, PET data provides for needed heterogeneous information along with the boron map needed for scaling of the dose.

In addition to misinterpretations about the tumor B-10 concentration, current BNL trials use MRI for geometrical reconstruction and beam position [http://www.bnct.bnl.gov/clinical.htm]. Although MRI provides for the detailed anatomical information necessary for geometrical reconstruction, it fails to provide accurate beam location positioning to allow for the peak thermal flux to be positioned at the hypocenter of the tumor, as indicated by PET to be the location with the maximum uptake of boron. For example, what is seen as the center of the tumor will not be the same as that which is seen by MRI, especially along the Z axis. Similarly, the boundaries of the tumor that are seen in PET may not be the same as those seen in MRI. Thus a pre-debulking PET could greatly aid in position of the beam during treatment planning.

The purpose of this work is to develop a method to allow the incorporation of detailed boron distribution information provided by PET to be used to adjust the boron dose based on the homogeneous assumption. In addition, use of PET for geometrical reconstruction is used to allow for the peak thermal flux to be placed along the center of the tumor. Differences between the homogeneous assumption and the detailed boron distribution adjusted dose are also calculated. Radiation dose to

the tumor is predicted by the Monte Carlo portion of BNCT_RTPE, a dose simulation software package developed via a collaborative effort by Idaho National Engineering and Environmental Lab (INEEL) and the computer science department at Montana State University. The problem of being able to incorporate a detailed geometrical description of the planar geometry is overcome by the use of Non-Uniform Rational B-Spline (NURBS) [7, 13]. NURBS generation of the volume and the solution to the transport equation using Monte Carlo technique is incorporated into the INEEL software, BNCT radiation treatment planning environment (BNCT_RTPE). The issue of in-vivo localization of the boron-10 is solved by using the filtered back projected PET image.

II. METHODOLOGY

1. Overview of BNCT_RTPE

Currently, the INEL code, BNCT_RTPE is used by several research institutions for predicting the radiation dose delivered to a patient. In accordance with the BNL protocol, BNCT_RTPE uses MRI data for three-dimensional geometrical reconstruction to allow for patient specific treatment planning and geometrical reconstruction. BNCT_RTPE uses Non-Uniform Rational B-Spline (NURBS) interpolation to generate a geometrical mesh [7,13]. The equation for the Non-Uniform Rational B-Spline interpolation is as follows:

$$\alpha\left(t\right) = \sum_{i=0}^{n} P_i B_{i,k}\left(t\right) \tag{6}$$

where,

k = order (degree +1)

t = parameter that varies between t_{min} to t_{max} along curve $\alpha(t)$

$\alpha(t)$ is the position vector along curve

P_i are n + 1 polygon vertices or control points

$B_{i,k}(t)$ is the basis weighting function

Non-Uniform Rational B-Spline (NURBS) curves used to generate the three

dimensional volume for a BNCT patient treated in a similar fashion to the BNL

protocol is shown in Figure 8.
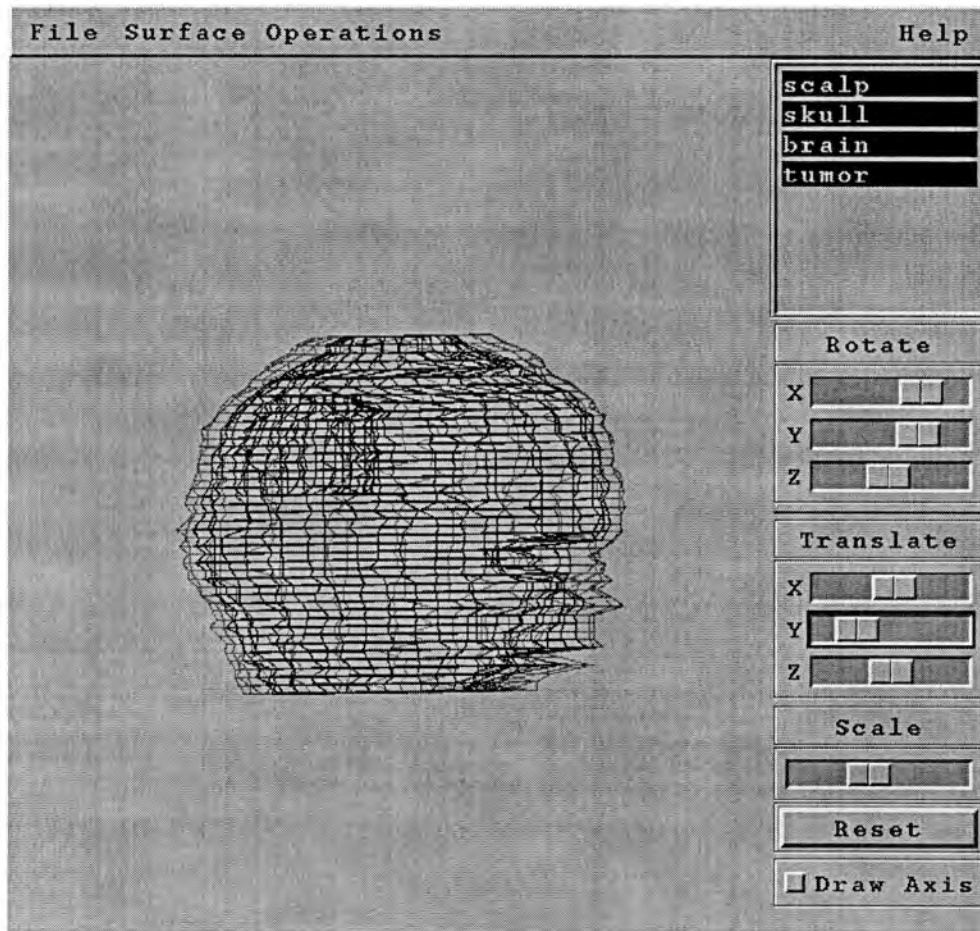
Figure 8: NURBS Geometry for Volume Reconstruction

20

After geometrical reconstruction, Monte Carlo techniques are used to predict the dose delivered to the patient during treatment. During material composition input, BNCT_RTPE makes the assumption that the boron concentration within a given region is constant. For example, in the software it is assumed that within a tumor region, boron concentration (~20 ppm) is homogeneously distributed in like tissue. Similar homogeneous assumptions are made about the brain , skull, and scalp regions. Based on PET data, tumor biopsies, and histological studies, it is known that this assumption is unjustified [9, 10, 12]. A more detail description of BNCT_RTPE and Macrodosimetry is given in Chapter III.

2. PETTOPET Dose Adjustment Algorithm Overview

Since BNCT_RTPE does not allow for the incorporation of the boron concentration data provided by PET to be input to treatment planning, a method for the post processing of the BNCT_RTPE dose has been developed. The method allows for PET images to be input into the treatment planning, and the PET pixel intensity is used to scale the boron dose. The boron dose, as calculated by BNCT_RTPE is adjusted using peak to average scaling as shown in the following equation:

$$D_{adj}(x,y,z) = D_{bnct\_rtpe}(x,y,z) \frac{I_{pet}(x,y,z)}{I_{ave}(x,y,z)} \qquad (7)$$

where,

x, y, z are the geometry coordinate system relating image and dose results

D_{adj} is the adjusted ^{10}B component of the total dose as function of x,y, and z

D_{bnct_rtpe} is the unadjusted ^{10}B component of the total dose as predicted by BNCT_RTPE

I_{pet} is the pixel intensity (i.e. the boron map) as it results from the image reconstruction

I_{ave} is the average pixel intensity over the brain region.

The average pixel intensity over the brain region is calculated as follows:

$$I_{ave}\left(x, y, z_i\right) = \frac{\sum_{i=1}^{n} I_{brain}\left(x, y, z_i\right)}{N} \tag{8}$$

where,

$I_{ave}(x,y,z)$ is the average pixel intensity over the brain region at a given axial (z) location.

$I_{brain}(x,y,z)$ is the pixel intensity as determine from PET over the entire brain region input into BNCT_RTPE.

and N is the total number of pixels that are found to make up the brain region.

The % difference between the homogeneously scaled ^{10}B dose and the PET adjusted dose is calculated as following:

$$\% \, DIFF = \frac{D_{adj} - D_{bnct\_rtpe}}{D_{bnct\_rtpe}} \tag{9}$$

Once the correct boron dose distribution is measured, the total dose is corrected according to equation 1.

III. MACRODOSIMETRY

1. Introduction

After a patient has been approved for BNCT treatment, radiation treatment planning is carried out to determine the logistics needed to administer a specific dose to a specific tumor in three dimensional space. For most patients, the tumor has already been debulked by the surgical team and has been approved for irradiation. Idaho National Environment and Engineering Lab (INEEL) in conjunction with Montana State University has developed a treatment planning package specifically designed for Boron Neutron Capture Therapy [7, 13].   Patient specific geometrical reconstruction and the solution of the neutron transport equation for patient treatment planning is accomplished via the INEEL code, Boron Neutron Capture Therapy Radiation Treatment Planning Environment (BNCT_RTPE) version 2.2. The current license for BNCT_RTPE at the University of Tennessee Medical Center in Knoxville (UTMCK) is held by Dr. George Kabalka, and is installed on a Silicon Graphics machine, Curie. A more detailed version of BNCT_RTPE documentation and documentation for the improved BNCT_RTPE now referred to as SERA (Simulation Environment for Radiotherapy Applications) can also be accessed via the internet by visiting INEEL website http://id.inel.gov/cart/bnct.html and following the links for BNCT_RTPE.   This package is in current use at Brookhaven National Laboratory to provide patient specific treatment planning in their clinical BNCT trials.

2. Problem Definition

This thesis has as a primary goal, the inclusion of Positron Emission Tomography images into the treatment planning. This goal has immense importance because unlike CT and MRI images used in current BNL protocol, PET data contains detailed ^{10}B concentration distribution on a per pixel basis. The accurate ^{10}B distribution provided by PET is different than the homogeneous assumption of boron distribution currently used in clinical cases at BNL. The homogeneous assumption combined with pharmacokinetic data and better beam positioning can significantly improve the efficacy of current clinical trials. Thus the importance of being able to allow PET images to be used during treatment planning and to fold the quantitative information of the study into the treatment planning is extremely important. The metabolic uptake information provided by PET is unparalleled by any other modality at the present time. Although PET lacks in anatomical description, it makes up for the lack in the quantitative information that it provides to the physicist. PET scan data shows the "true" center of the tumor as the point of highest uptake. Thus the goal of this work is to allow for PET images to be included into the treatment planning, and to allow for the incorporation of the detailed boron distribution provided by PET to be included in the dosimetry planning. BNCT_RTPE is used to determine the dose based on a 1ppm homogeneous input of boron in every region. The results are then used as input into the PETTOPET algorithm that allows for the incorporation of the

detailed boron distribution into the treatment planning. The results section shows the original treatment planning based on homogeneous assumption and the PET adjusted results based on actual distribution of boron. Finally a comparison of the homogeneous assumption and the adjusted dose is presented.

3. BNCT_RTPE

BNCT_RTPE consists of several modules, each designed to run independently of the others, as well as to be run together via one control panel. The main control panel to BNCT_RTPE is shown in Figure 9.
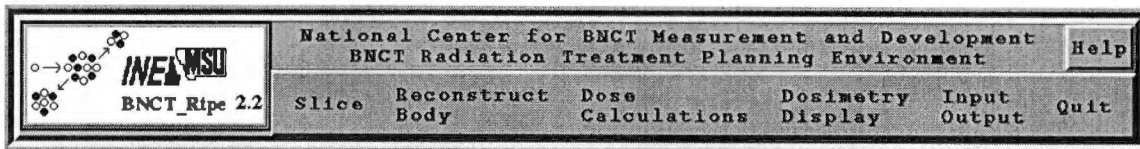
Figure 9: INEL BNCT_RTPE Main Control Panel

From the main control panel shown in Figure 9, several other utilities can be easily accessible. Some of these tools allow for various operations during treatment planning, such as loading images, drawing region of interests, creating the volume, determining the dose, and displaying the results. Figure 10 shows the diagram tree for BNCT_RTPE and all of the associated components used during treatment planning.

Before treatment planning can be discussed in a meaningful manner, a general understanding of how the images are represented and some of the terminology such as slices, body slices, and bodies, must be explained.
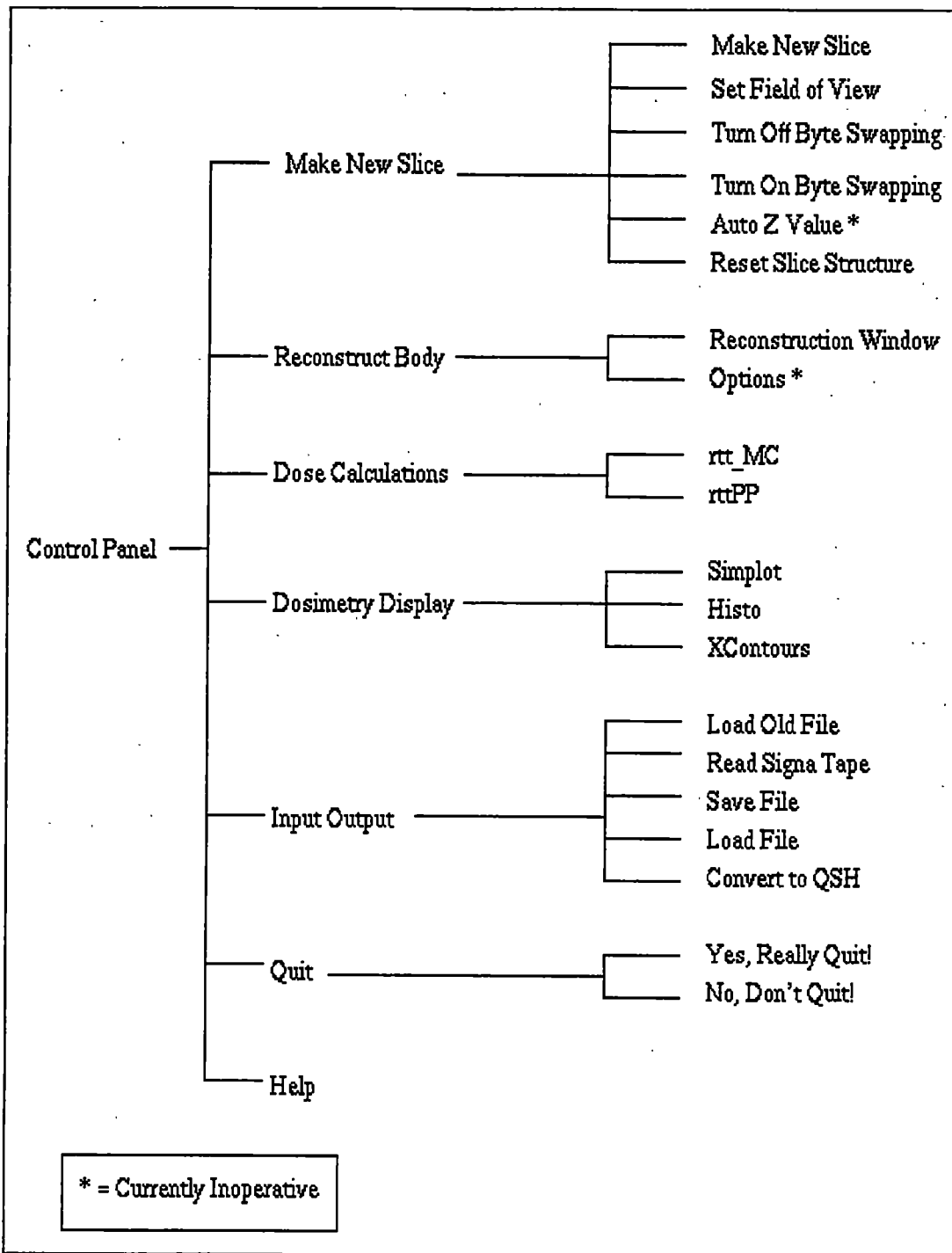
Figure 10: Treatment Planning Operations Tree Diagram

Figure 11 shows how the medical (PET) images are arranged during a
treatment planning session, and how body slices are defined on top of the image
slices. Image slices refer to the individual medical planes like those provided by PET
or MRI. Body slices refer to the individual body slices defined on the individual
planes. For example one can define the optic chiasma body by defining several points
around an MRI image plane containing the optic chiasma. Once the optic chiasma is
defined over several image slices, the individual points defining the body slices can
then be interpolated together to generate what is referred to as a body. For the
treatment planning in this work, the bodies defined as region of interests include the
scalp, the skull, the brain, and the tumor.
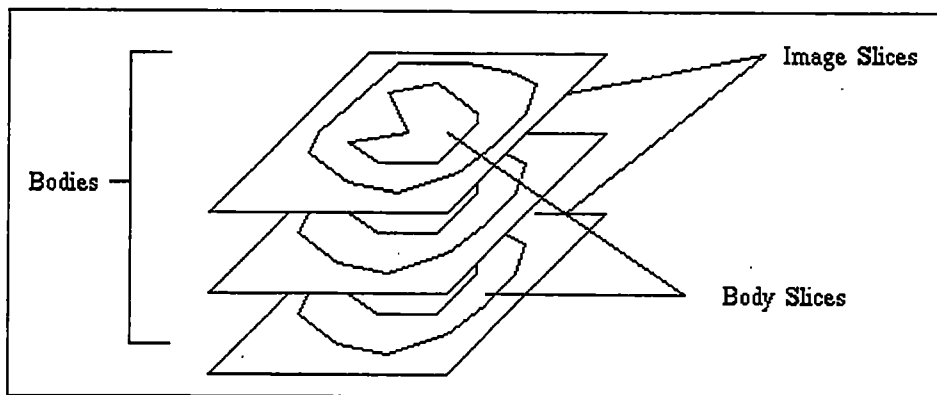
Figure 11: Image Slices, Body Slices, and Bodies
(*reprinted from INEEL documentation on BNCT_RTPE)

3. Image Series Input

BNCT_RTPE allows for medical images to be input and regions of definitions (i.e bodies) to be defined. Treatment planning with BNCT_RTPE is accomplished by either using 256X256 RAW 8-bit images or by using QSH formatted images. QSH is an image handling toolkit developed by New York University in 1990 and uses the American Association of Physicist in Medicine (AAPM) image format. A more detailed description of the QSH format is given in Appendix B. BNCT_RTPE **Input/Output** option in the main control panel (Figure 8) allows for conversion of QSH formatted images to RAW images and RAW images to QSH formatted images. A graphical user interface (GUI) utility called toQsh is provided under the **Input/Output** option of the control panel to allow for such conversion. Original PET data as obtained from the PET scanner is usually in a 16 bit 128X128 format. Hedley Bond at University of Tennessee Hospital in Knoxville (UTMCK) has developed a C code, **zoom_img**, to allow for the 16 bit 128X128 format to be zoomed to 8 bit 256X256 format which is consistent with that needed for BNCT_RTPE. Appendix E gives a screen output from a sample conversion in which the original PET data is modified such that the data results in individual RAW 256X256 8 bit slices along the z axis. Once the PET data are preprocessed into the RAW 8-bit 256X256 format, it can be convert into the QSH format using the toQsh graphical user interface built into BNCT_RTPE. The advantage of converting the raw format PET images into the QSH format is that several individual RAW PET image planes get converted into one

large file with a header that contains the date of image creation, imaging modality, field of view, bytes per pixel, number of slices, z separation, x and y dimensions, and other parameters needed to convert the N-dimensional array of pixel values to the visual representation for interpretation. However, it is recommended that one use RAW PET images as input since the final dose results are displayed on top of the RAW images. It is these RAW 8 bit 256X256 images that are used as input into the PETTOPET dose adjustment algorithm code included in Appendix A and described in more detail in Chapter IV.

4. ROI Definition, Volume Generation, and Material Composition

Treatment planning in BNCT_RTPE begins with loading the individual pre-processed RAW 256X256 8-bit PET images or the QSH formatted PET image into the **SLICE VIEWER**. An important parameter used in the initial uploading of images is the pixel size of the image. It is this pixel size that is used to relate the size of the image with the actual geometry used for treatment planning. The pixel size can be calculated as follows:

$$PixelSize = \frac{FieldofVie\ w(FOV)}{MatrixSize} = \frac{(27\,cm)*(10\,mm/cm)}{128} = 2.11\,mm/pixel \quad (10)$$

where

the Field of View (FOV) is the entire x,y space (in mm) over which the scanner scans the patient along the orthogonal axis (z).

The pixel size and z separation are output from the **zoom_img** code when pre-processing PET images and should be noted for future reference. Once the software is started by issuing "bnct_rtpe" on **Curie**, one can allow for volume definition to

take place. By using the **SLICE** button to **Make New Slice** in the main control panel, one will get a file browser window that allows for the input of the RAW or QSH formatted images to be loaded into the treatment planning software. The field of view and the z values can then be entered into the treatment planning by using the **Set Field of View** and **Auto Z Value,** respectively, under the **Make New Slice** operation of the main control panel. When loading a RAW image into BNCT_RTPE, the physicist is initially prompted for the FOV. Here the treatment planner must back calculate from the pixel size to determine the FOV (FOV = 128 pixels X 2.11 mm/pixel = 270 mm). For treatment planning, it is assumed that the FOV will not change from one z plane to the next z plane. In addition to the FOV, one is also prompted for the z location of the image. The author recommends that the "Image Location" button be set to auto-slice assuming a uniform slice thickness along the orthogonal axis (for example, 3.375 mm for PET). This allows for automatic z assignment to each plane, which eases calculation later on. After all of the images are loaded into the **SLICE VIEWER**, one should use the **Save File** option in the **Input/Output** operations in the main control panel.

Once all of the planes of the PET study are loaded into the **SLICE VIEWER** panel of BNCT_RTPE, the next step is to draw region of interests (ROI). There are two possible methods for ROI definition. The first is entered under the **Edit** option in the slice viewer as **Automatic Edge**. The other option is to manually enter the edit points clockwise around a given region. The popup options menu for Automatic Edge is shown in Figure 12.
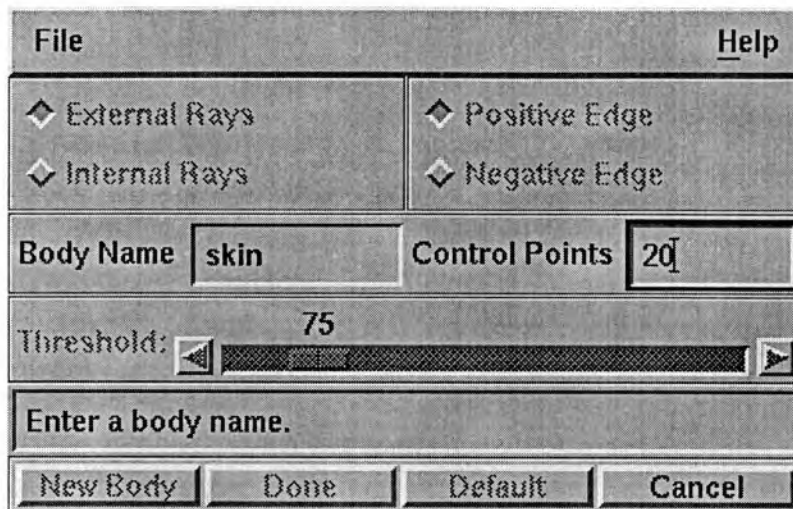
Figure 12: Automatic Edge Detection Widget

The automatic option allows for automatic edge detection of the boundary of

an object. One can use either an external or internal ray with positive or negative

edge detection for a specified intensity threshold. For example with a given threshold

value and a positive external edge detection, the code searches for a boundary from

the outside and is ideal for defining boundary regions. This process works fairly well

with Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) images.

However, due to the noise in the back projected PET data and the artifacts associated

with such method, the automatic edge detection does not work very well, especially

when defining regions inside the brain. To define ROI's for PET studies, one must

define a new body and manually enter the points for the ROI in a clockwise manner.

The standard regions of interest drawn by Brookhaven National Lab starting from the

outside are: scalp, skull, brain, target, and tumor. The target region is defined as a 2

cm boundary around the tumor. The regions of interest can include as many edit

31

points as needed however, it has been found that 20 edit points are an ideal value to minimize computational time as well as to insure accurate regional definition. Once defined, the ROI (body slices) can be copied onto other planes, limiting the amount of time required to generate new ROI's. The points can be moved around and re-positioned in corresponding image planes with the **edit points** button in the **SLICE VIEWER** operation. Once all ROI's are defined, one should exercise the **SAVE** option under the **Input/Output** option of the main control panel. This exercise will result in a file with a .bs extension and contains the location of the ROI points (in cm) with ROI names, the FOV in mm, z locations, the pixel size, and corresponding images so that it can be re-loaded at a later date. An example of the image loading operation with manual definitions of ROI on top of the PET images is shown in Figure 13.

Once all ROIs (i.e. bodies) are defined, one should use the **Reconstruction Body** option in the main control panel to generate the Non-Uniform Rational B-Spline (NURBS) geometry and to define the materials associated with each of the bodies in the volume. It is recommended that the material, "Fairchild_tissue + 20 ppm," be used for the scalp region since the 20 ppm B-10 is the closest value of boron-10 distribution seen in some clinical cases. The initial boron concentration values like 20ppm and 10 ppm are already pre-defined in BNCT_RTPE and saves the user the problem of having to define new material files. Similarly, one should use "Duck_cranium" for the scalp, "brain_MIT+10ppm" for the brain, and
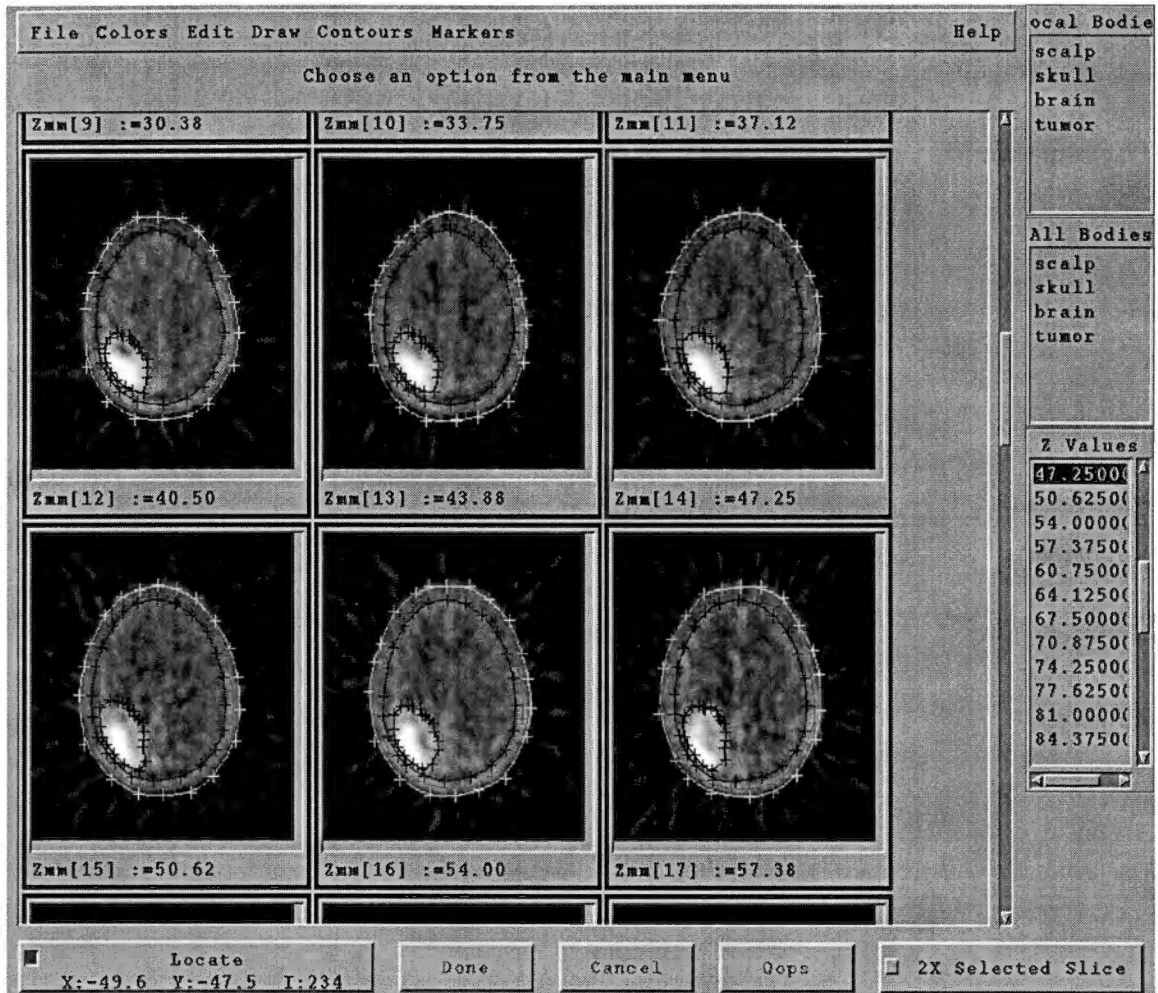
Figure 13: Defining ROIs for Volume Generation

"brain_MIT+20ppm" for the tumor. If one is not satisfied with the materials pre-defined in the code, BNCT_RTPE allows for additional generation of materials and mixing of the cross sections to correspond with the new material. However, it must be noted that the boron placed in a specific region is assumed to be homogeneously distributed throughout the volume. Based on email discussions with Dr. David Niggs at INEEL, it is assumed that if the boron concentration is less than 50 ppm in a region, then the effect on the neutron thermal flux depression becomes negligible. Figure 14 shows the popup menu for defining the body and the material associated with the body. It is this window panel that is used to define which material should be placed in what body. For example, one can select the scalp body and define a material for it. One can choose the provided materials in the window panel or generate new material definitions as mentioned earlier. For generation of new materials, one must consult INEEL on how this is accomplished. Also, one must be careful in that the bodies of interest must be predefined in an earlier step in the **Slice Viewer**. One must also be careful that the regions are defined in and **OUTSIDE-IN** fashion or else run time errors will occur. By the OUTSIDE-IN fashion, it is meant that the bodies located on the outside like scalp must be loaded first and material definitions defined for the outside region before one can define inner bodies like skull, brain, and tumor.
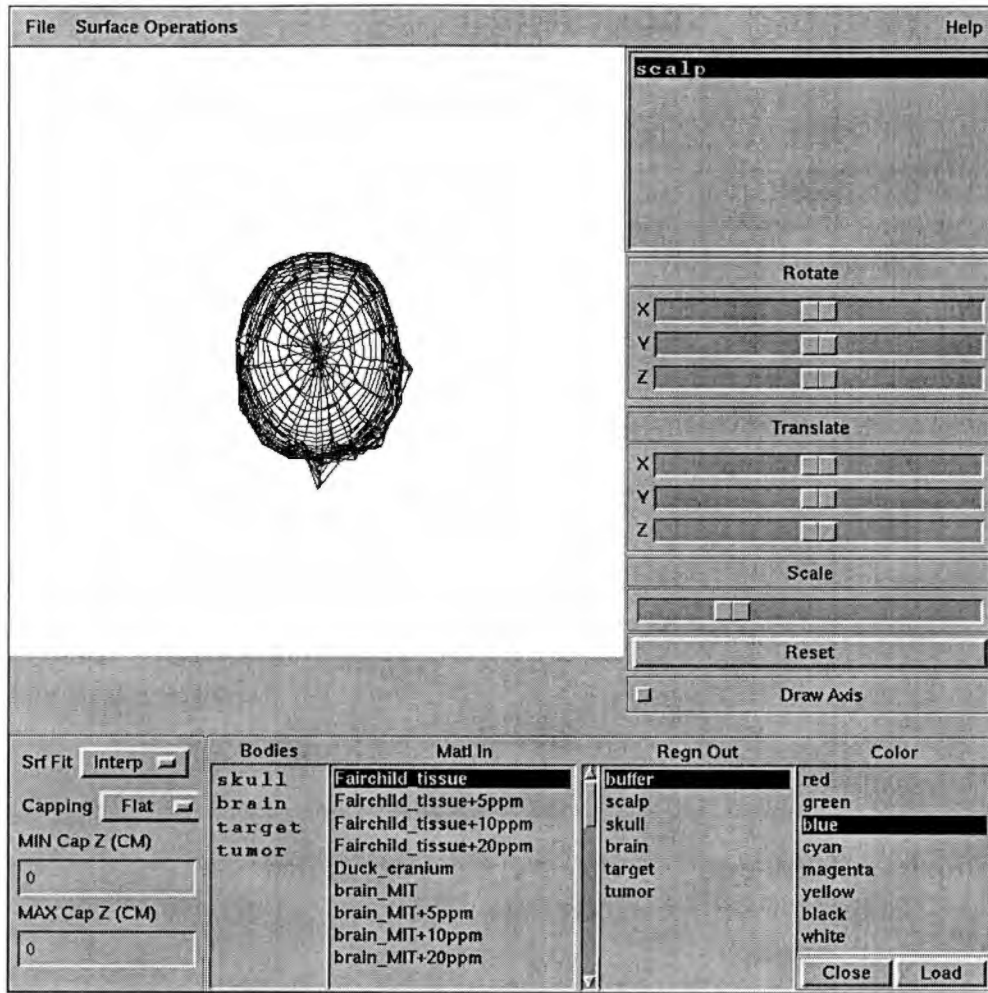
Figure 14: Material Definitions and Body Reconstruction Window

Figure 15 show a typical body volume reconstruction in 3 dimensions and the

materials defined for the scalp, skull, brain, and tumor bodies. Although

BNCT_RTPE allows for any naming convention for the bodies, it is imperative that

the words scalp, skull, brain, and tumor be used to define the corresponding bodies,

else the PETTOPET algorithm used to adjust the dosimetry will fail since it

specifically looks for these region names. This is described in more detail in the next

section.
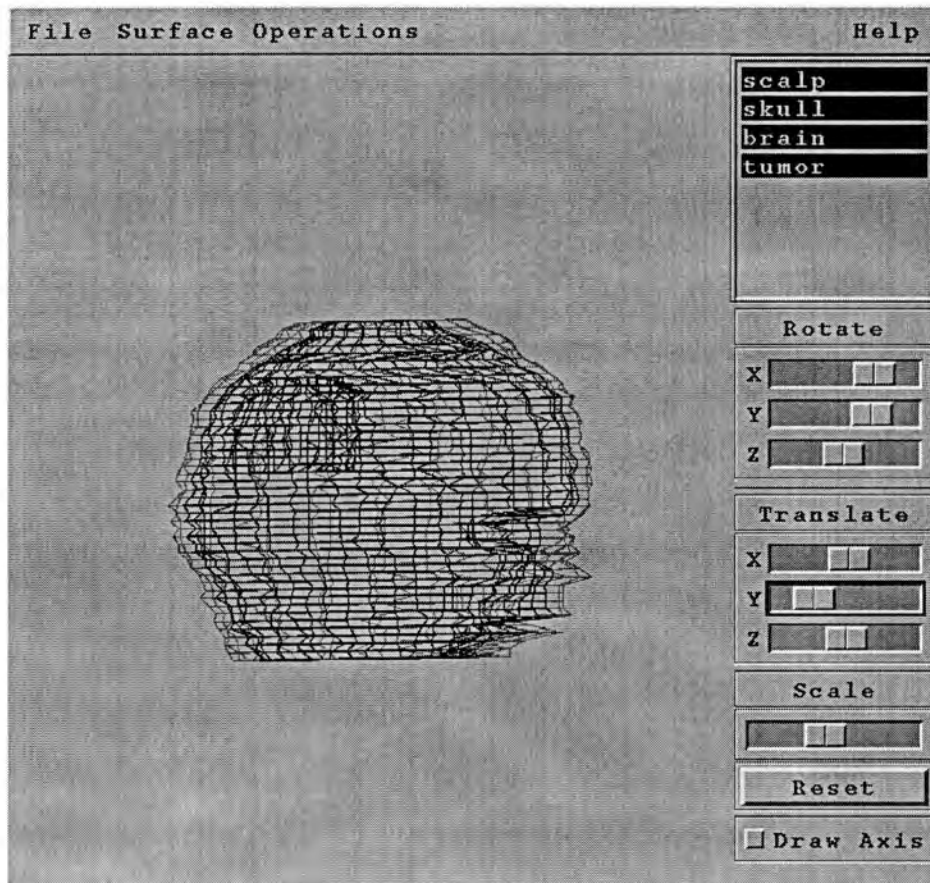
Figure 15: 3 Dimensional View of Scalp, Skull, Brain, and Tumor Bodies

Once the bodies are created and materials defined, one should exercise the

Make RTT_MC Interface file option under the **Reconstruction Window** of the

Reconstruction operation in the main panel. The **Make RTT_MC Interface** option

results in a file with a .rs extension and a .rm file. The .rm files created in the

reconstruction menu contains the region number, the region name, the material

associated with the region, and other parameters used by the Monte Carlo code during

dosimetry calculation. One must also note that it is important to place ' and ' around
fov located in line one of the .rs file, else no contour files will be written out. This
was a bug found in the code and is not reported in any of the user's manuals. One the
.rm file is created, it is this .rm file that the PETTOPET adjustment algorithm
described in the code description section uses to determine the region numbers with
the corresponding region names. An example of the .rm file is shown in Figure 16:

```
regn_in "regn name" "matl name" regn_out cap_style min_z max_z type
color
1 "buffer" "buffer" 0
2 "scalp" "Fairchild_tissue+20ppm" 1 1 0.0000 0.0000 interp  1
3 "skull" "Duck_cranium" 2 1 0.0000 0.0000 interp  4
4 "brain" "brain_MIT+10ppm" 3 1 0.0000 0.0000 interp  2
5 "tumor" "brain_MIT+20ppm" 4 1 0.0000 0.0000 interp  6
-1 "eof" "eof" -1 -1 -1.0 -1.0 "eof" "eof"
```

Figure 16: Example of a .rm File

Once the .rm file containing material definitions and the .rs file containing the
NURBS geometry are created, one is ready to use the Monte Carlo radiation transport
module (**rtt_MC_106**) of BNCT_RTPE.

5. Neutron Transport Calculation

The neutron transport calculation is accomplished with the Monte Carlo
approach. The basic assumption of this approach is that it is possible to determine the
solution to the Boltzman transport equation for a 3 dimensional volume by randomly
selecting neutrons from an incident source with a specific spectrum. The neutrons are
tracked from birth until the end when they either become captured or exit the volume
defined by the calculation. The process randomly selects a particle from the flux

37

distribution in the source spectrum and assigns it a position and velocity vector. Another pseudo-random number determines whether the particle interacts with the medium via one of several mechanisms or if it passes through one of the boundaries. If it is captured, then the history is finished. If scattered, then new velocity and position vectors are defined and the cycle starts over. For this technique to be successful, one must obtain enough particles to statistically determine the particle flux in each region of interest.

6. Input File Description

Shown in appendix C is a sample input file for a patient, and is very similar to the current BNL protocol for BNCT. It is this file that contains the needed parameters for successfully incorporating the volume component of BNCT_RTPE with the Monte Carlo component (**rtt_MC_106**). A brief description of the input file and a discussion of only the important lines are presented in this section. A more detailed description can be found on the INEEL website [http://id.inel.gov/cart/bnct.html].

The first line of the input file (rtt.input) is a text line up to 80 characters and is not used for anything other than for descriptive purposes. The second line contains the number of histories (*nhist*) per batch, the number of batches (*nbatch*), and the Russian roulette cut of weight (*rrcut*) used for weight reduction biasing. The total number of particles processed is equal to the number of batches times the number of histories per batch (*nbatch X nhist*). This line must retain the FORMAT 3E20.8 structure. The third line is used to define the target point in centimeters. The target point is the point about which the beam is rotated, and the center of the beam passes

38

through this point. The target point is usually chosen to be at the center of the tumor along the Z axis and to be the deepest point of the tumor at that Z level. The target point (in mm) is located by using the "locate" button in the lower right hand corner of the **SLICE VIEWER** menu. Upon discussions with Dr. Jacek Kapala, a physicist at BNL, this method of approach is very similar to the approach currently used by clinicians at BNL during actual treatment planning. However, it must be noted that the slice viewer reverses the x and y axis in it's orientation. Thus during defining the target point, one must be careful to reverse the x and y points to correspond with the BNL protocol.

7. Beam Positioning

Line 4 contains the beam orientation with regard to patient position. This is the most critical line since it can have a significant impact on the overall efficacy of the treatment. A diagram reprinted from INEEL BNCT website is shown to explain how beam positioning is accomplished. Beam positioning involves the understanding of figures 17, 18, and 19. Figure 17 is a frontal view of a patient during beam alignment and is related to the coronal view. Zb, the first parameter in line 4 is used to define the separation distance (in cm) from the defined target point (Xp, Yp, Zp) to the source center (Xp, Yp, $Zp+Zb$) and is shown in figures 17, 18, and 19.

source center $X_p, Y_p, Z_p + Z_b$

Z_b

X_p, Y_p, Z_p

z

y

x

0
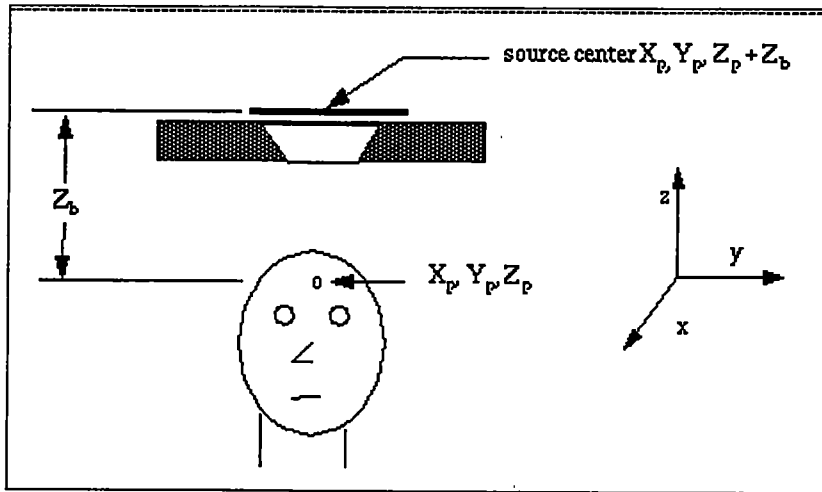
Figure 17: The Separation Distance, *Zb*

Z_b

$X_p, Y_p - Z_b, Z_p$

ϕ

?

X_p, Y_p, Z_p

0
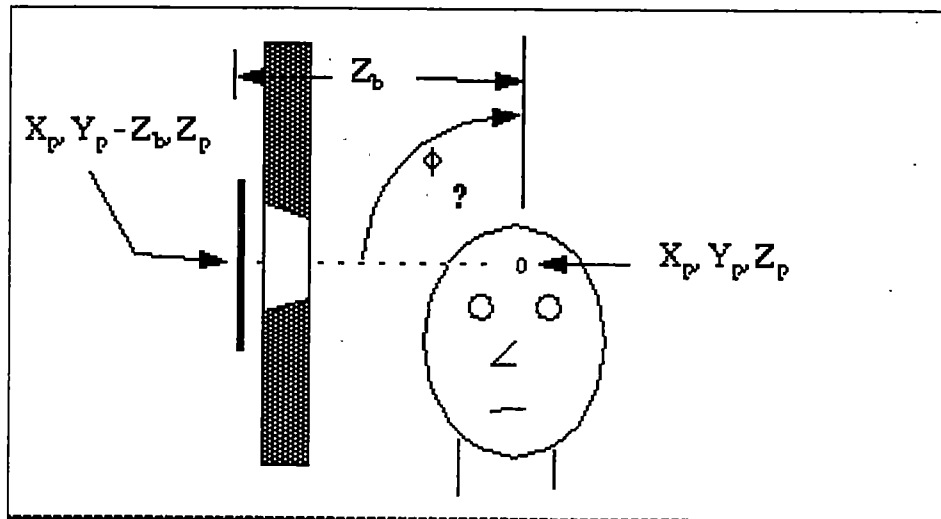
Figure 18: The Polar Angle, ϕ
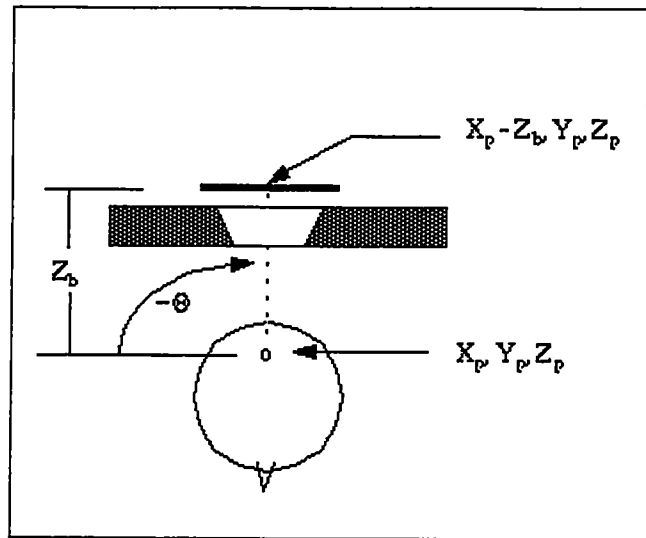
Figure 19: The Azimuthal Angle, θ

ϕ, the second parameter in line 4, is the polar angle from the reference beam location at top of the head to the current beam location. ϕ is shown to be 90 degrees in figure 18 as the beam is rotate from the superior axis to the right side of the head. This view is related to the saggital view.

θ, the third parameter in line 4, is the azimuthal angle from the right side of the head to current beam location. In figure 19, θ is shown to be 90 degrees and the beam is rotate from the right axis of the head to the back of the head. This view is related to the axial view.

Figure 20 shows the coronal, saggital, and axial views respectively for one of our patients that underwent treatment at BNL. These view files are .gif files generated during run time and can be used in **XCONTOURS** to view them. These views should be carefully analyzed to confirm beam positioning.

41
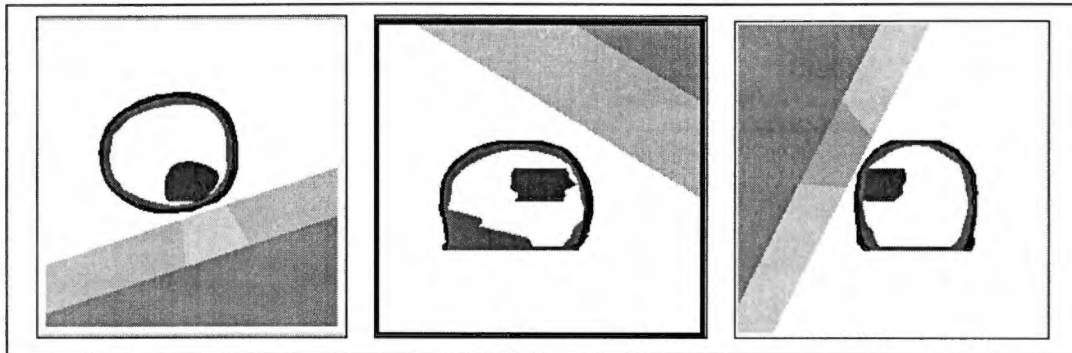
Figure 20: Axial, Saggital, and Coronal Views

Although one can use figures 17, 18, and 19 to manually calculate the placement of the beam, the author recommends a better approach. The author, as is consistent with the treatment planning at BNL, recommends that one use the edit directive, **edit_dir**, command placed in block 5 of the input and use the automatic beam locator options (*iop*) to search for the optimal beam alignment position. One should use *iop=3* or *iop=4* to determine the minimum distance from the target point to the surface of the scalp. According to BNL protocol, as discussed with Dr. Jacek Kapala, it is recommended that a Z separation of 1 mm be used since this represents a more realistic representation of the actual treatment planning as carried out at BNL. After an *iop* of 3 is used, one should execute the Monte Carlo portion (rtt_MC_106) of BNCT_RTPE with a very low number of histories and generate an output file. Dr. Jacek Kapala at BNL also pointed out that the following command be issued on **Curie** in order to run the input file (rtt.input) in the background and to generate an output file:

Curie% nohup run_MC_106 rtt> & sys &

This initial run results in a rtt.out file that contains optimal beam location

42

parameters, Zb, ϕ, and θ. ϕ and θ can then be manipulated slightly to allow for the center of the dose contours to correspond with the center of the tumor. This allows for maximum dose to be delivered to the center tumor volume, and is consistent with the BNL treatment planning. Once beam alignment is finalized, one should increase the number of histories.

Another important line during initial treatment planning is the ninth line of the input file. This line defines the origin of the mesh used for treatment planning. The standard defaults are 30X30X30 cubes with a user defined origin and voxel dimension (*delw*). For example, if the model was centered about 0,0,0 and the origin placed at –15, -15, -15 the volume would enclose a 27 cm^3 box assuming 1 cm^3 voxels. It is important that this volume enclose the entire geometry, or else this will result in run time errors and the contours will not be produced.

8. *Edit Directive*

Other important lines include the edit directive options used to control the output of the results. An example of the rtt.input with the edit directives is shown in Appendix C. The minimum edit directives include *mw_min*, **b10_blood**, and **ottocon**. *Mw_min* represents the exposure in megawatts/min from the reactor during treatment planning. *B10_blood* represents the boron-10 concentration in a region and *ottocon* represents the option to display radiation dose contours overlying the original images. The recommended options of mw_min, b10_blood, and ottocon are shown below:

```
mw_min      1.0
b10_blood   1.0
ottocon     40 40 scalp
```

43

In addition to these edit directives, BNCT_RTPE allows for the following edit directives:

```
in_reg      target tumor
b10_blood   55.65
rbe         3.8  1.0  3.2  3.2 1.0 1.0
DVbs
```

in_reg allows for the edit directive to be focused on a specific region (s) like target and tumor. **b10_blood** allows for the b_10 blood concentration (in ppm) to be defined in the specific region. **rbe** allows for the relative biological effectiveness to be defined. In the above example, as consistent with the BNL protocol, the b10_blood concentration is assumed to be 55.65 ppm in the target and the tumor regions. For the target and the tumor region, the relative biological effectiveness is assumed to be 3.8 for ^{10}B capture dose, 1.0 for gamma dose, 3.2 for ^{14}N dose (due to $^{14}N(n,p)^{14}C$ reaction), 3.2 for recoil-proton dose, 1.0 for other dose (not used by the code), and 1.0 for the gamma production dose. One can also specify **rbe** for fast neutron flux, epithermal neutron flux, and thermal neutron flux, but is not used in this example. For the physical dose to be generated, one should use a **b10_blood** ratio of 1 in all the regions and 1 megawatt minute for the exposure.

IV. CODE DESCRIPTION

1.0 Archiving and Image Pre-Processing

For PET images to be used into treatment planning, an independent effort has been make to take all of the PET data (and MRI) for each of the BNCT patients and to save the data on CD-ROM. Currently most of the PET and MRI data for five of the patients are stored on three CD-ROMs labeled BNCT#1, BNCT#2, and BNCT#3 currently in the custody of Dr. Laurence F. Miller. Patient data can also be found on

Bodrum in /ne5/bnct directory. In addition, a separate effort was made by Hedley

Bond at the image processing laboratory (IPL) at the University of Tennessee

Medical Center, Knoxville (UTMCK) hospital to store the data on optical disks. Data

can be retrieved from the optical disks as described in Appendix D.

For incorporation of PET images into BNCT_RTPE, a C program has been

written by Hedley Bond at the image processing lab in the University of Tennessee,

Knoxville. The program, **zoom_img**, allows for 16 bit 128X128 PET images to be

zoomed and converted into various formats. It is recommended that the **zoom_img** be

used to convert the 16 bit 128X128 images into 8 bit 256X256 format. When

specifying the output file, the author recommends that one uses a # sign in the name.

This will result in the pound sign being replaced by an increasing integer. For

example if one uses the following name as the output file: testp#.img. Then the

resulting output files will be testp#1.img, testp#2.img, testp#3.img, etc. The

numbering convention corresponds with the plane slice number that is converted. An

example of this conversion is shown in Appendix E.

Although MRI images are not needed for the dose adjustment subroutine and

for the work in this thesis, one can use MRI images for treatment planning. For

processing of MRI images, an IDL software, **GEVIEW** is available at the UTMCK

IPL to convert the IDL formatted MRI images into RAW 256X256 8 bit format

needed for BNCT_RTPE. In addition, a Matlab™ subroutine has been developed,

MRI2RAW.m, that allows the MRI images to be converted from the IDL format into

the format needed for BNCT_RTPE. This program is attached in appendix F. Figure

21 shows MRI images that have been converted into raw format by the Matlab™

subroutine. These images are then used in BNCT_RTPE as input.

2.0 PETTOPET

At the heart of this work is the development of an algorithm that allows for

PET images to be input into the treatment planning and to allow for the detailed

boron distribution to be incorporated into the treatment planning. This task has been

accomplished by developing a Matlab™ program, PETTOPET.m, which interacts

with the outputs of the BNCT_RTPE and allows for the adjustment of the dose based

on the detailed boron distribution. This section describes the details of the

PETTOPET.m program and its subroutines.

Figure 21: Converted MRI Images for Use With BNCT_RTPE

Shown in Figure 22 is an overview of the PETTOPET algorithm, its subroutines, and how it interacts with BNCT_RTPE input and output files to adjust the boron component of the total dose.

Once BNCT_RTPE is run successfully with the appropriate edit directives described in the previous section, one should see dose contour files with *contour* extension as a result of the BNCT_RTPE calculation. It is these contour files that are used as input into the PETTOPET algorithm. As shown in Figure 22, other files that are needed include the BNCT_RTPE input files, *.rm file*, *rtt.input*, original PET
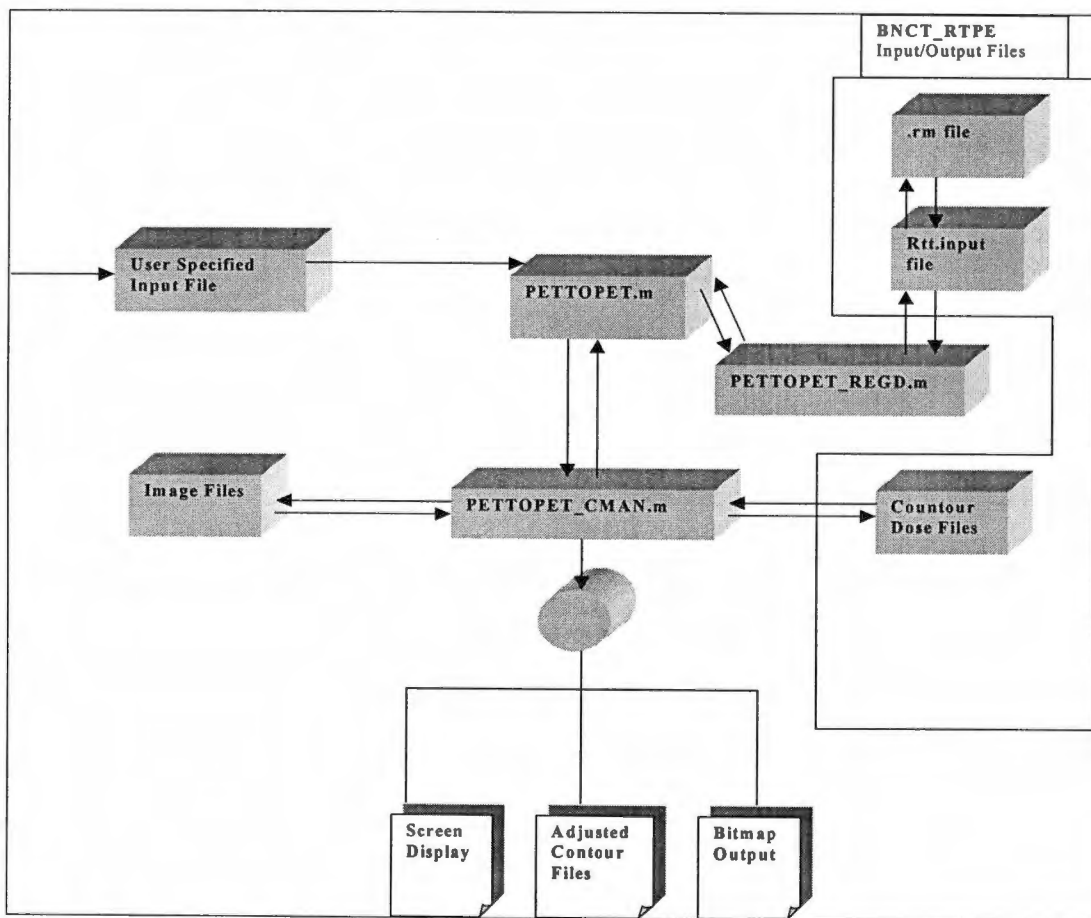
Figure 22: PETTOPET Algorithm Flow Diagram

image files, and a user specified input file that contains pointers to each of these files and certain other parameters like pixel size needed during the adjustment.

PETTOPET relies on two other subroutines written to accomplish specific tasks. The two essential subroutines needed by the main program, PETTOPET.m, include PETTOPET_REGD.m and PETTOPET_CMAN.m. These subroutines should be placed in the same directory as the main program.

3.0 Input File Description

For PETTOPET to run successfully, several key lines must be included in the input file. An example of the input file is included in Appendix A. The first line of the input file to PETTOPET.m must contain the PET image pixel size. This is a very important parameter and must be correct since it allows for the one to one correspondence between the image pixel data and the physical geometry. The pixel size for the original 128X128 image is output from **zoom_img** during the pre-processing step. The pixel size for the 256X256 image is equal to one half of that for the 128X128 image since the overall geometry does not change during the pre-processing step. For example if the PET pixel size during pre-processing the 128X128 matrix size is 2.11 mm/pixel then the resulting pixel size for the 256X256 matrix will be 1.055 mm/pixel (.5*2.11 mm/pixel). This ensures that the coordinate geometry of the matrix is not altered and the Field of View (FOV) for both images is the same.

The second line of the input file contains a flag to control the plot size over which

the results are to be displayed. This line basically controls how the output results are displayed. The default setup is on a 256X256 grid size. In the **ottocon** edit directive of BNCT_RTPE, one has the option of controlling how many x and y points are to be used over which the doses are to be written out for each Z plane containing the ROI of interest. The suggested value is 40 X 40 for the **xdim** and **ydim.** This allows for 1600 dosimetric data points to be shown over the original image. The more data points that are used, the slower the BNCT_RTPE will run. For PETTOPET dose adjustment purposes, the user can specify the **xdim, ydim** option used during **ottocon** in BNCT_RTPE or to have the results plotted over the entire 256X256 image grid size. An option of 1 allows for the flag to be on. An option of 0 turns the flag off. If the 256X256 option is 1, then the PETTOPET_CMAN subroutine takes the original x and y points as written out in the normalized coordinate system ranging from {−1,1} in the **contour** files and maps them onto the 256X256 geometrical coordinate system ranging from {1:255*pixelsize}. The recommended value for this is 1. This allows for the results to be displayed over a 256X256 grid size that is consistent with the gird size of the original PET image. This prevents from having to scale the 256X256 image grid size to a 40X40 (or **xdim X ydim**) grid size. The downward scaling of the image to a smaller size results in loss of resolution of the original image.

Line three of the input file allows for control of which figures to be displayed. Selection of the 1st option in input will allow for a two dimensional view of the original boron 10 dose contour as calculated by BNCT_RTPE. Similarly selection of the 2nd option allows for a 3D view of the dose. Option 3 option allows for a three

dimensional view of the PET image and a point by point corresponding view of the 3

dimensional PET pixel intensity. Option 4 allows for the PETTOPET adjusted 3

dimensional B-10 dose to be displayed. Option 5 in line three allows for the 2

dimensional PET adjusted boron to be displayed. Option 6 allows for a 2 dimensional

view of the INEEL homogeneously scaled boron dose to be displayed. Option 7

allows for a 3 dimensional view of Figure 6. Option 8 allows for the 3 dimensional

difference between the PETTOPET adjusted dose and the INEEL homogeneously

scaled boron dose to be displayed. The results are shown in the result section.

Line 4 allows for the INEEL/BNL homogeneous scaling factor to be used. This

scaling factor is used to scale the boron in the corresponding ROI's to be scaled

linearly, as is assumed in the current BNL protocol. For example one can specify a

boron concentration (in ppm) of the tumor to be 45.5 ppm, 13 ppm in brain, and 19

ppm in the scalp. This results in placing a homogeneous distribution of the boron

concentration in each of the ROIs and the results are scaled based on this

homogeneous assumption.

The fifth line is used for descriptive purposes only. The sixth line specifies the

directory in which the BNCT_RTPE dose contour files, rtt.input, and the .rs files are

located. Lines following the sixth line include the names of the contour files that are

to be used for adjustment. If a (–)sign is placed in front of the name of the contour

file, then PETTOPET will write out bitmap images of the selected figures. Otherwise,

the figures will be displayed onto the screen. At the moment, the printer option used

to print directly to the printer is not functional.

It must also be noted that the first four lines of the input file must have the flags placed after the 40th character space since the codes assumes text for the first 40 characters in the line and then assumes numerical data like 0 or 1 for the later characters. An example of the input file is provided in Appendix A. In addition, one can use flags (placed after the 40 characters) in the listing of the contour files after the 6th line to specifically turn the display option for any Z plane on and off. This will prevent the code from displaying results for each Z slice that is listed. The default value for this is on or 1.

4.0 General Overview of the Subroutines

This section provides a general understanding of the dose adjustment algorithm. A brief description of the purpose of the subroutines and how they interact with the output files of BNCT_RTPE is provided. In addition, a description of the details of how the data is handled so that the doses can be adjusted based on the PET pixel intensity is also provided. This basically involves getting all of the dosimetric data and the PET image to correspond pixel by pixel and scaling the data as outlined in chapter 2 of this thesis. Finally, the results as calculated by BNCT_RTPE, the homogeneously scaled results, and the PETTOPET adjusted results are presented in the next chapter.

PETTOPET.m is the main program that performs all of the data handling and referencing to other files and subroutines. PETTOPET_REGD.m is a subroutine that uses rtt.input (the original input file to BNCT_RTPE) to find the name of the .rs file which contains the region definitions as defined during treatment planning. The .rs

51

file must be in the same directory as that specified for the contour files in the input file. This file is very important since it contains numerical ROI definitions for tumor, brain, skull, and scalp. It is these region numbers assigned to the body names that are defined by the user during treatment planning, and it is these same region numbers that are used to separate the various regions during the adjustment. PETTOPET_CMAN.m is a subroutine that handles all of the data processing and scaling based on user defined input in the input file and the region definitions that it finds in the .rs file described in Chapter III of this thesis.

Data handling depends on the flag used in line 2 of the user input file to PETTOPET.m. If one chooses the data to be displayed on the PET image size, then all of the **xdim, ydim** data (40X40) must be scaled upward to match the original PET image size. Since the data are two dimensional, resizing of the data is not an easy task. For this, the bilinear interpolation in Matlab™ is used to take the original two-dimensional data and interpolate it to generate a 256X256 vector data matrix so that it corresponds with the PET image matrix data. The same approach is applied with the various components of the dosimetric data (like B-10, Hydrogen recoil, N-14, Fast Neutron, Photon, etc.) in the contour files. The bilinear interpolation function is also used to scale the adjusted doses down to the original grid size, **xdim X ydim** if the **xdim ydim** option is chosen.

For multi-dimensional interpolation, it is necessary to get an estimate, $z(x',y')$, from a grid that is 2 dimensional $(z(x,y))$ and is represented by one-dimensional position vector arrays xa(1:j) and ya(1:j). To calculate this new estimate,

z(x',y'), bilinear interpolation is used and is quoted as being "close enough for government work." (13 p124). If one imagines a box like grid of 4 data points numbered 1 to 4 starting clockwise from the lower left hand corner, bilinear interpolation for the new data point, *z(x',y')*, can be described mathematically as [14]:

$$z(x',y') = (1-t)(1-u)*z_1 + t(1-u)z_2 + t*u*z_3 + (1-t)*u*z_4$$

where,

$t = (x' - xa[j])/(xa[j+1] - xa[j])$, for $0 < t < 1$

$u = (y' - ya[j])/(ya[j+1] - ya[j])$, for $0 < u < 1$

x' is the new x position location

y' is the new y position location

and z1, z2, z3, z4 are the 4 data points surrounding the new data, *z(x', y')*.

Bilinear interpolation is also used to scale the data downward onto an **xdim** by **ydim** grid size. However, this is not recommended since it causes loss or resolution in the image caused by averaging of the pixel intensity. Once the dosimetric data and the PET image are the same size, this allows for a point to point correspondence between the data and the intensity. Thus, dose scaling can be accomplished at the individual intensity level as described in chapter 2 of this thesis. Also based on the region definitions retrieved from the .rs file, if the tumor region is found to occur in the slice, then adjustment to that particular slice is done. If the tumor region is not defined over that slice, then no adjustment is done.

5.0 Output Files

The output files from PETTOPET include the adjusted contour file with a

PTP.out extension appended to it, screen displays of the data, and bitmap Figures of selected planes. The PTP.out appended file has an additional column containing results of the adjusted B_10 dose next to the original B-10 column containing the unadjusted dose and an updated total dose column.

V. RESULTS

In total, four patients have been analyzed with the PETTOPET dose adjustment subroutine. Patients 1 and 2 were treated under the BNL protocol in Upton, NY and results are presented. Patient 3 was denied treatment since the size of the tumor was large, and there were two separate places where tumor growth was evident based on PET and MRI. This is inconsistent with the Harvard/MIT and BNL protocol. However, results for patient 3 is still presented since the PET study of patient 3 does enforce the convincing argument for the need of PET in treatment planning. Only the PET intensity map, a side by side comparison of the PETTOPET adjusted dose contours, the homogeneously scaled boron dose contours, and the difference is presented for patient 3. As for patient 4, no pre-surgery PET was available. However, the post surgery PET is analyzed for PETTOPET adjustment just out of curiosity, but the results were not included in this thesis since they did not contribute anything significantly new that has not already been pointed out by the first three patient studies. A discussion of the data is also provided. It must be noted that only the results of the plane along the center of the tumor for each patient are presented. Similar results, sometimes more dramatic are encountered at the upper and lower boundaries of the tumor, but are not included in the analysis. Results at the

boundaries of the tumor has been left out because it appears that the same points can be made by analyzing the data along the center plane of the tumor just as well as that for the boundary planes. In addition, the center point of the tumor is also the most important in terms of treatment planning, because it is along this plane that the beam is positioned.

1.0 Patient 1

Shown in Figure 23 is a 3 dimensional pixel intensity view of the PET image take along the center of the tumor (PET plane # 16). The pixel intensity map is representative of the boron concentration and has been scaled to represent 256 varying shades of gray per pixel. As can be seen from Figure 23, *para-*boronophenylalanine is preferentially taken up by the tumor.
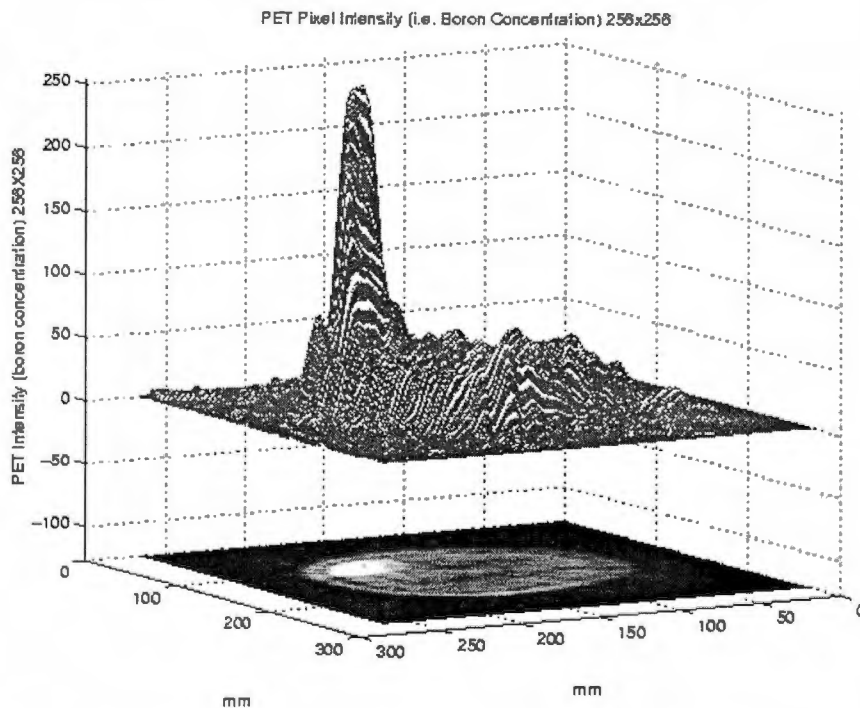
Figure 23: B-10 Pixel Intensity Map For Patient 1

As is also evident from Figure 23, boron is clearly not homogeneously distributed throughout the tumor volume, thus invalidating the homogeneous assumption currently applied in clinical trials.

Shown in Figure 24 are B-10 radiation dose contour lines predicted by BNCT_RTPE for 1 ppm or boron-10 placed in each region of interest. Figure 24 shows the contour lines with the center representing 90% line and then gradually decreasing by increments of 10% as one moves from the center in an outward direction. Shown in Figure 25 is a three dimensional view of the contour lines shown in Figure 24. The units of the dose axis shown in Figure 25 are in cGy/mw_min/ppm and it is referred to as the boron dose axis. It is these units that are implied when referring to dose in other planes, unless otherwise explicitly stated. The contour lines can be better centered over the tumor in the figure below by adjusting ϕ and θ used in defining the beam. The edit automatic beam location edit directive of BNCT_RTPE was used to place the beam in the figure below. No additional adjustments to ϕ and θ were made. In addition, the target point was defined to be the point of maximum tumor uptake indicated by the pixel intensity.
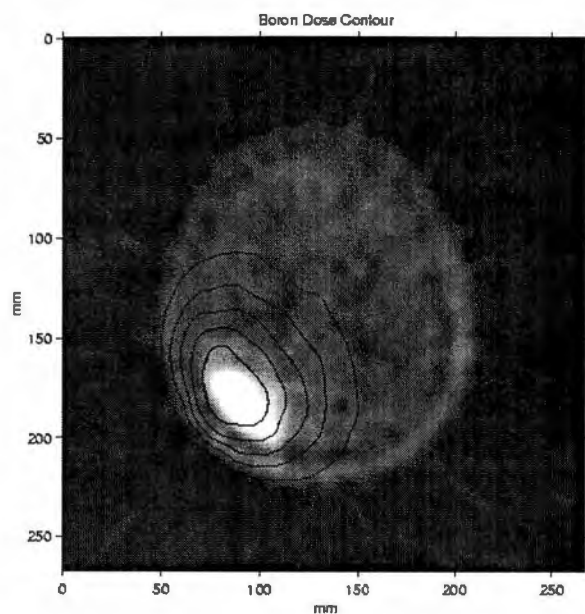
Figure 24: Radiation Dose Contour Lines for 1ppm B-10 Homogeneously Distributed Throughout Brain
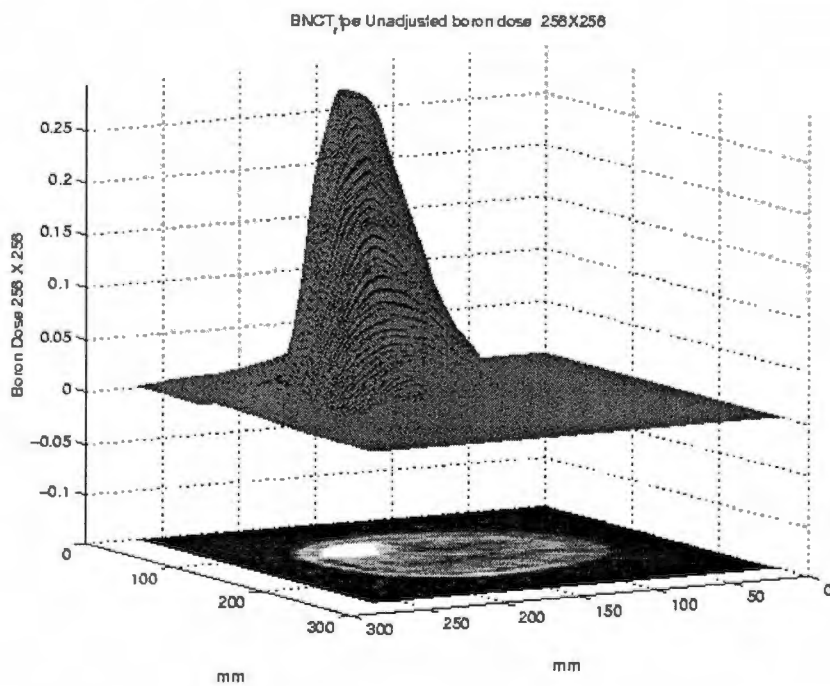
Figure 25: Three Dimensional View of the Radiation Dose for 1ppm B-10 Homogeneously Distributed Throughout Brain

57

Shown in Figure 26 are the resulting Radiation Dose Contour lines from scaling the dose based on a homogeneous assumption (45.5 ppm in tumor, 13 ppm in brain, and 19 ppm in scalp), as is used in current clinical trials. Shown in Figure 27 is a three dimensional representation of the homogeneously scaled boron dose shown in Figure 27.

As can be inferred from Figures 26 and 27, radiation dose due to the boron is localized much more preferentially to the tumor location since a relatively large B-10 concentration is assumed in the tumor region. Also note that the contour lines do not seem to be indicative of the heterogeneous distribution of boron within the tumor volume as is evident from the PET pixel intensity plot, which is made clear from
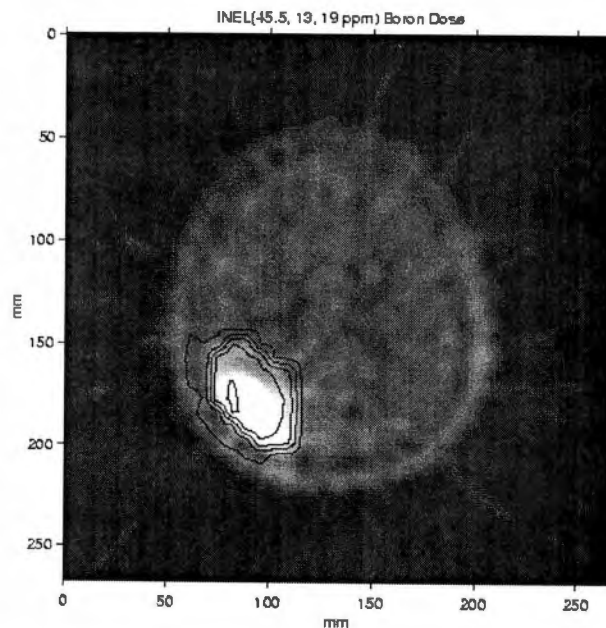
Figure 26: B-10 Homogeneously Scaled Dose by a Factor of Tumor Uptake Relative to Normal Brain Ratio being 3.5:1

58
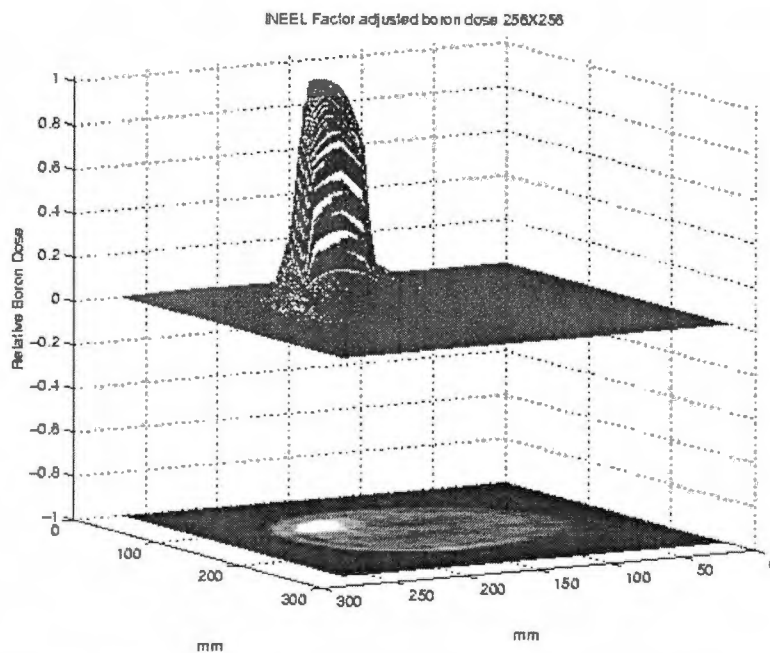
Figure 27: Three Dimensional Representation of B-10 Homogeneously Scaled Dose by a Factor of Tumor Uptake Relative to Normal Brain Ratio being 3.5:1

Discussions regarding Figures 23 and 28.

Shown in Figure 28 is the PETTOPET adjusted boron dose based on the heterogeneous distribution of boron shown in Figures 24 and 25 and described in Chapter II of this thesis. Figure 29 shows a three dimensional view of Figure 28. As can be inferred from Figures 28 and 29, the boron dose based on PETTOPET adjusted dose closely represents the expected heterogeneous distribution within the tumor volume. Also note that the contour lines are shifted more towards the center of the tumor, and the PETTOPET adjusted dose seems to be much more peaked at the center then the homogeneously assumed boron dose.
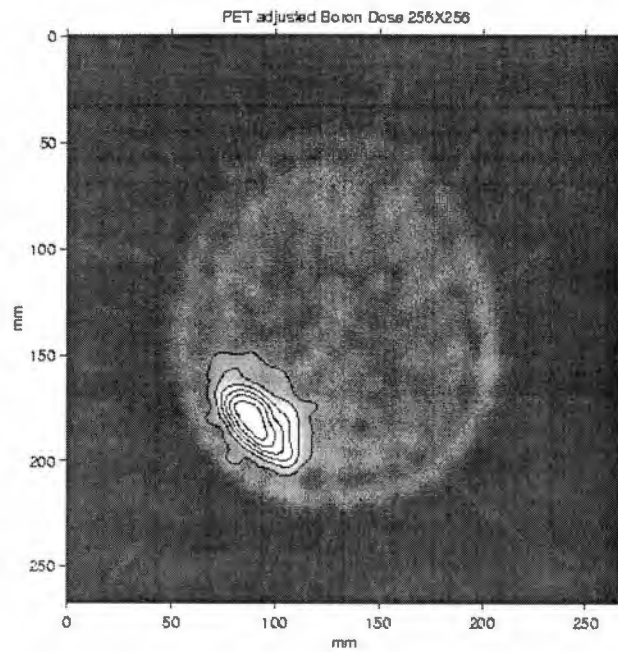
Figure 28: PETTOPET Adjusted B-10 Dose Contour Lines
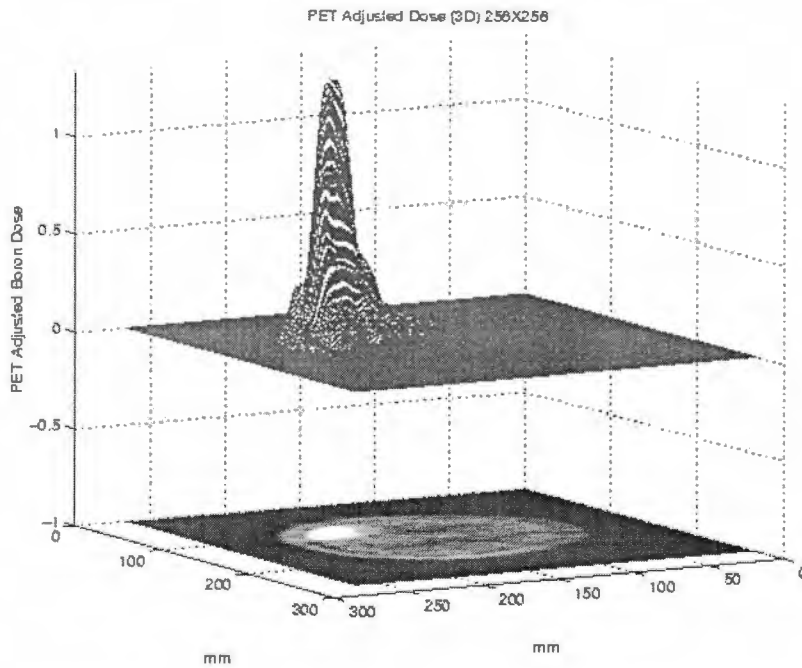
Figure 29: Three Dimensional View of PETTOPET Adjusted B-10 Dose

Shown in Figure 30 is the resulting difference between the homogeneously

assumed boron dose and the PETTOPET adjusted boron dose. The difference is

shown only around the tumor region. As is clearly evident from the figure, there is a

significant difference between the homogeneously assumed boron dose and the

PETTOPET adjusted boron dose. Results for patient 1 seem to indicate that the

radiation dosimetry predicted by the conventional method currently used by BNL

seems to over-predict the actual dose being delivered to the tumor and its surrounding

tissue.
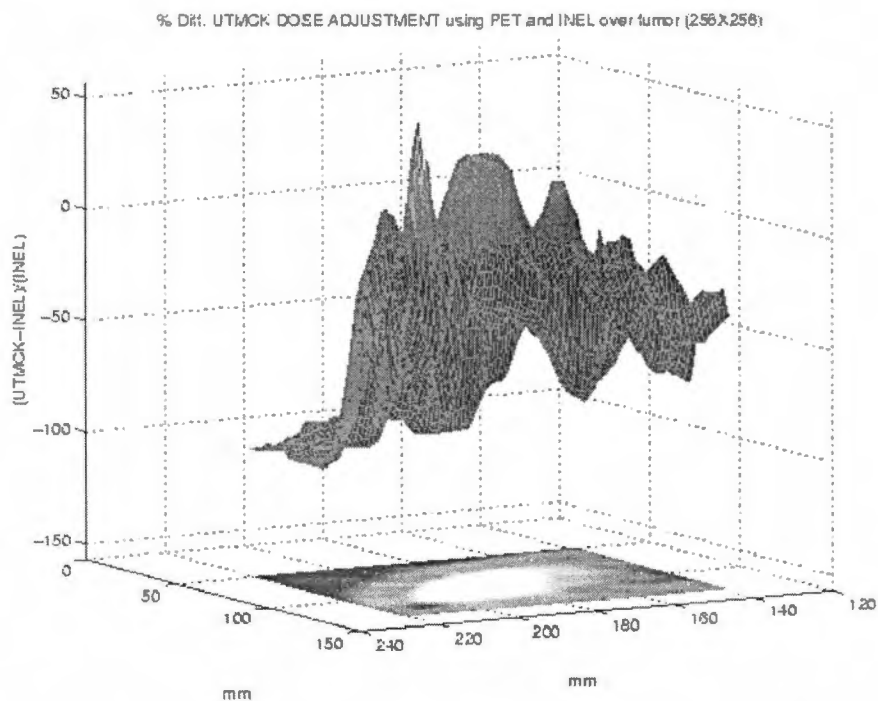
Figure 30: Three Dimensional View of the Difference Between PETTOPET Adjusted Boron Dose and the Homogeneously Assumed Boron Dose Over The Tumor Region

Based on clinical studies and email discussion with Dr. Nichols, the follow-up physician for patient 1, this result may be representative of the clinical findings. Based on post-BNCT follow-up and post mortem reports, patient 1 had re-growth of the tumor. There were no visible signs of radiation burns on the scalp. There was mild redness and loss of hair in the region where the beam was positioned. Re-growth of the tumor was diffuse and did not indicate any sort of random pattern. Much of the re-growth of the tumor seemed to have occurred around the edges of the tumor where the localization of the boron might not have been as high as that predicted in current clinical studies. Also one must note that based on the PET pixel intensity plot, the boron concentration in the center of the tumor is shown to be at much higher concentrations than that around the edges. Also one must keep in mind that by making the homogeneous distribution assumption, one is spreading the boron concentration at equal concentrations to the edges where tumor recurrence is seen, thus the homogeneous assumption may not be justified based on clinical findings alone.

2.0 Patient 2

Shown in Figure 31 is a 3 dimensional pixel intensity view of the PET image take along the center of the tumor for patient 2 pre-debulking (PET plane #12). This Figure is conceptually similar to that of Figure 23 for patient 1. However, note that the tumor for this patient does not take up the boron as preferentially as that of

PET Pixel Intensity (i.e. Boron Concentration) 256x256
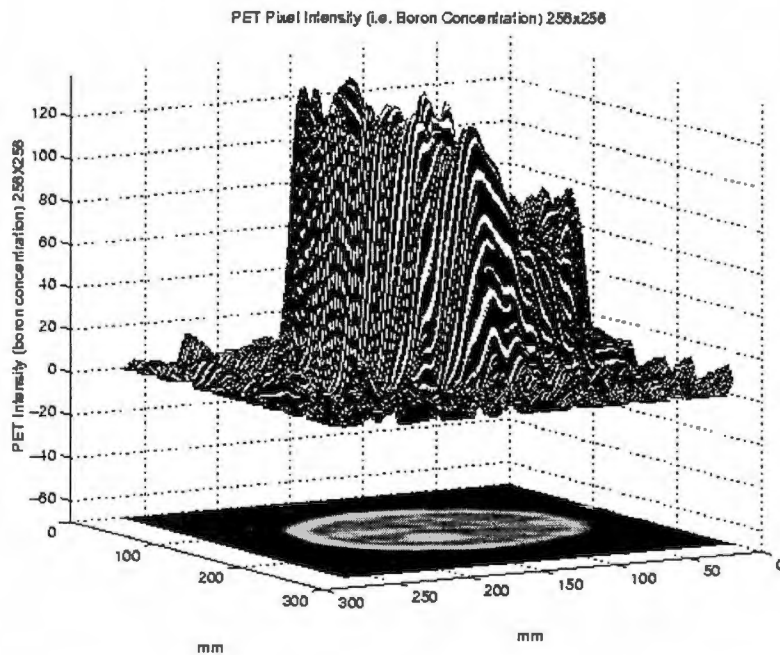
Figure 31: B-10 Pixel Intensity Map For Patient 2

patient 1. Also notice that the boron is localizing much more to the bottom scalp region to almost the same extent as the tumor itself. Thus this confirms the notion that not all PET studies will be the same. Not all of the tumors in the patient are going to behave in the same fashion during treatment. Thus, it makes the incorporation of PET data into treatment planning even a greater priority.

Shown in Figure 32 are B-10 radiation dose contour lines predicted by BNCT_RTPE for 1 ppm of boron-10 placed in each region of interest. The contour lines shown in Figure 32 are similar to those shown in Figure 24 for patient 1. In addition, a three dimensional representation of the radiation dose is shown in Figure 33.
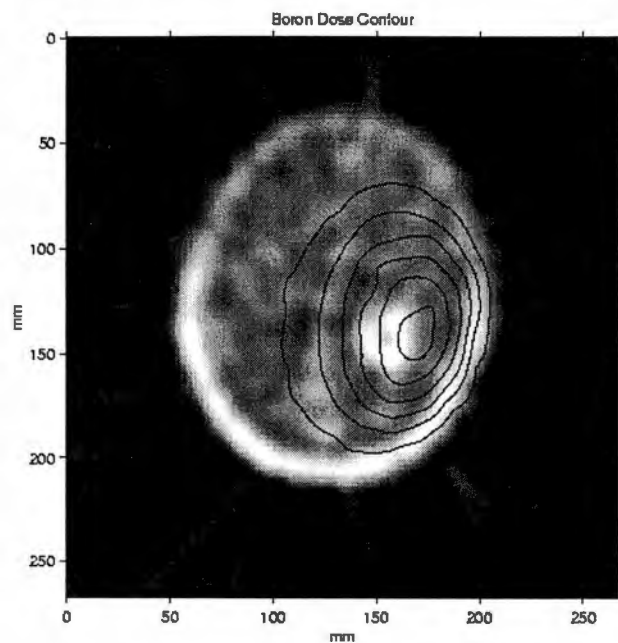
Figure 32: Radiation Dose Contour Lines for 1ppm B-10 Homogeneously Distributed Throughout Brain
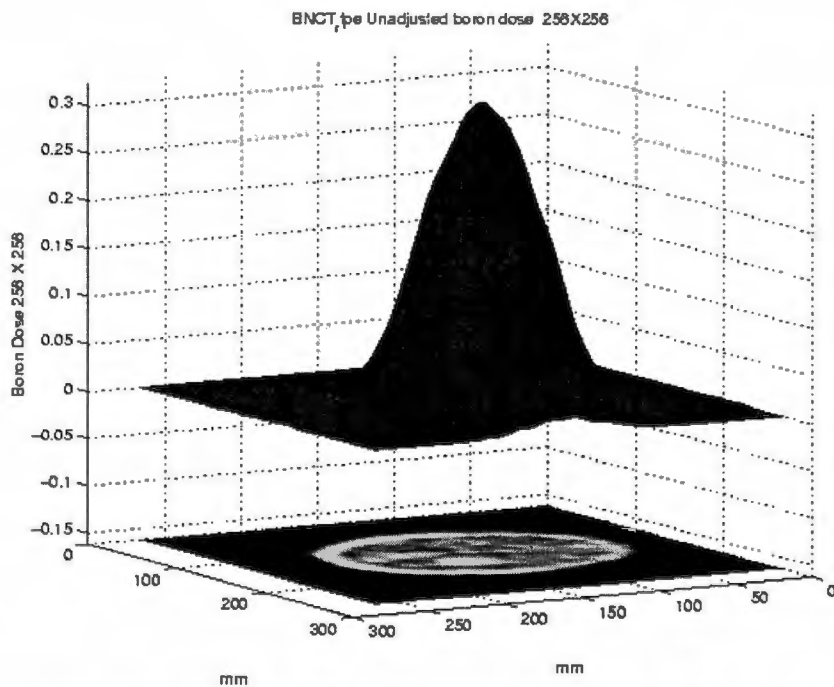
Figure 33: Three Dimensional View of the Radiation Dose for 1ppm B-10 Homogeneously Distributed Throughout Brain

Shown in Figures 34 and 35 are the homogeneously scaled boron doses. The

scaling factor is kept the same as that for Figures 26 and 27. As can be inferred from

Figures 34 and 35, radiation dose due to the boron is localized much more

preferentially to the tumor location. This is consistent with that of Figures 26 and 27

since the same large B-10 concentration is assumed in the tumor region. Similarly,

note that the contour lines do not seem to be indicative of the heterogeneous

distribution of boron within the tumor volume for patient 2. This is consistent with

that seen in patient 1.
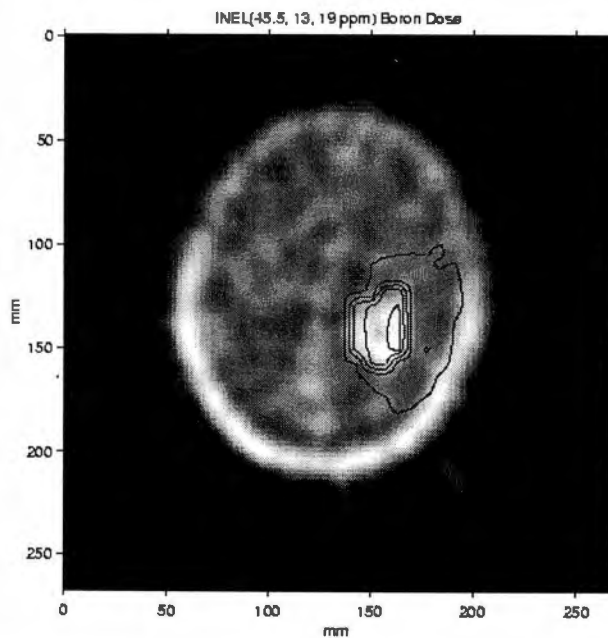
Figure 34: B-10 Homogeneously Scaled Dose by a Factor of Tumor Uptake Relative to Normal Brain Ratio being 3.5:1
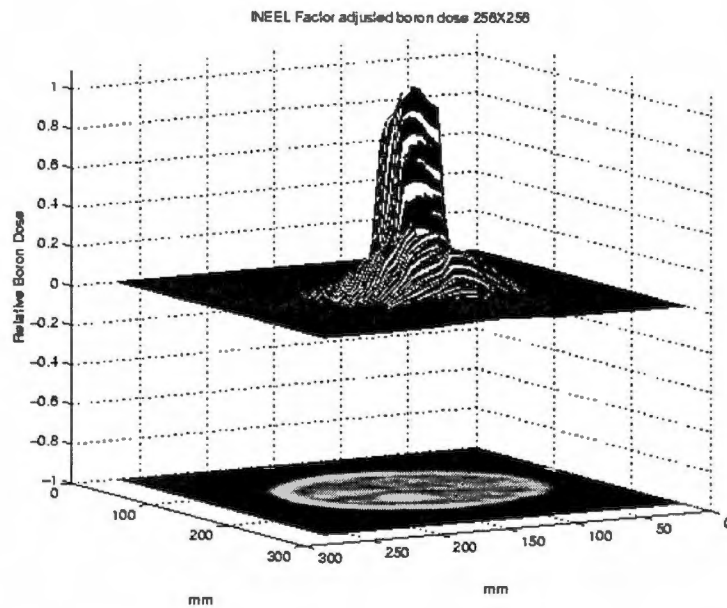
Figure 35: Three Dimensional Representation of B-10 Homogeneously Scaled Dose by a Factor of Tumor Uptake Relative to Normal Brain Ratio being 3.5:1

Shown in Figures 36 and 37 is the PETTOPET adjusted boron dose based on the heterogeneous distribution of boron. This is similar to that shown in Figures 24 and 25 and described in chapter 2 of this thesis.

As can be seen from Figures 36 and 37, the boron dose based on PETTOPET adjusted dose closely represents the expected heterogeneous distribution within the tumor volume. This too is consistent with that seen in Figures 28 and 29. Also notice that the contour lines are shifted slightly away from the center of the tumor. This is much different from that seen in Figures 28 and 29 for patient 1. However, this is consistent with the expectation that if the boron does not localize preferentially in large amounts to the tumor site when compared to the normal brain, one would expect a spreading of the dose away from the tumor location.
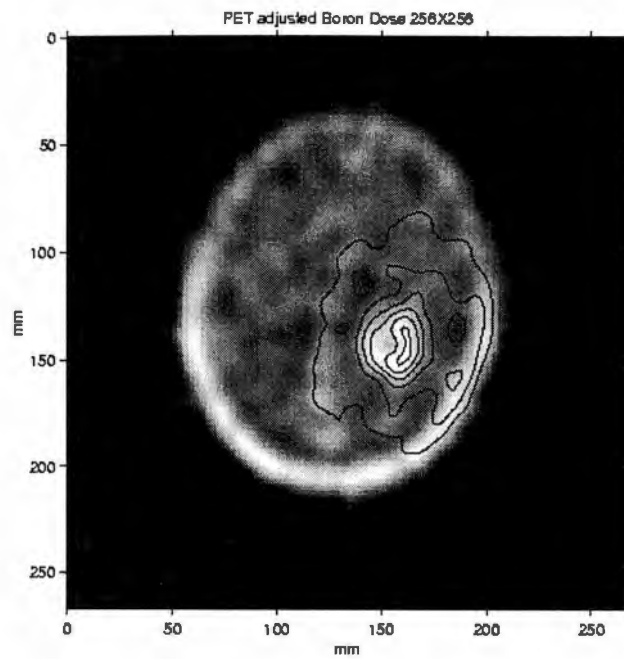
Figure 36: PETTOPET Adjusted B-10 Dose Contour Lines
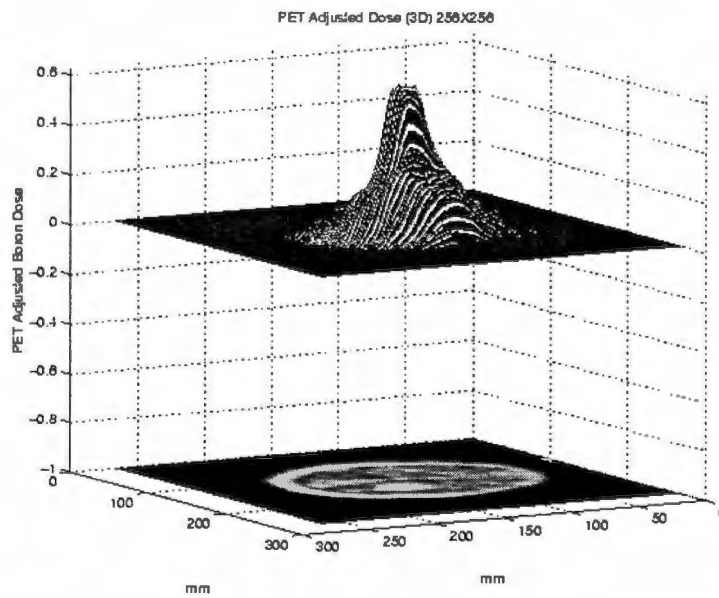
Figure 37: Three Dimensional View of PETTOPET Adjusted B-10 Dose

Shown in Figure 38 are the differences between the homogeneously assumed boron dose and the PETTOPET boron dose based on the dose being controlled by the local boron concentration for patient 2 along the center tumor plane. This would be the difference if the homogeneous assumption is wrong. Note that the greatest difference occurs near the scalp region where BPA seems to localize much more than it did for patient 1. Thus based on PET scaling, the difference plot indicates a higer dose to this region. Results for this patient also indicate an over prediction of the dose to the tumor and its surrounding tissue by conventional methods. Based on clinical findings and discussions with Dr. Nichols, the post-BNCT follow up physician, patient 2 had similar diffusive re-growth of the tumor. No clear pattern of re-growth was seen in the patient.
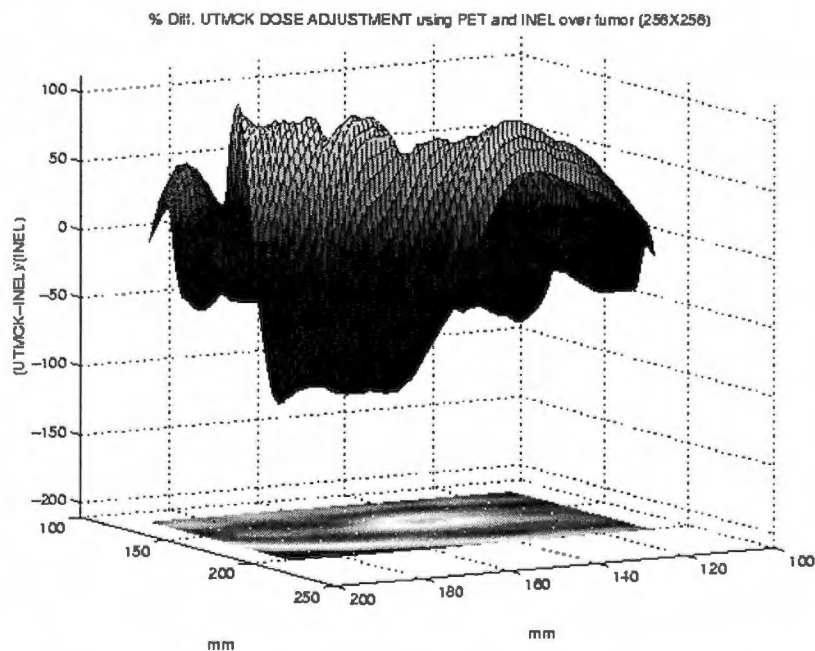
Figure 38: Three Dimensional View of the Difference Between PETTOPET Adjusted Boron Dose and the Homogeneously Assumed Boron Dose Near The Tumor Region

Post follow-up MRI and PET data seemed to indicate recurrence at the boundaries of the tumor. In addition, there might have been signs of slightly elevated dose to the scalp in this patient than as compared to patient 1, but by no means were there any radiation burn signs seen in this patient. The clinical findings for patient 2 seem to be similar to those seen for patient 1.

3.0 Patient 3

Shown in Figure 39 is a 3 dimensional plot of the PET pixel intensity for patient 3 (PET plane #24). Once again, this is conceptually similar to those plots shown for patients 1 and 2 however, note that the tumor has two foci. Because there is a presence of two foci, there tends to be double peaking of the boron concentration since it is concentrated in the two different foci.
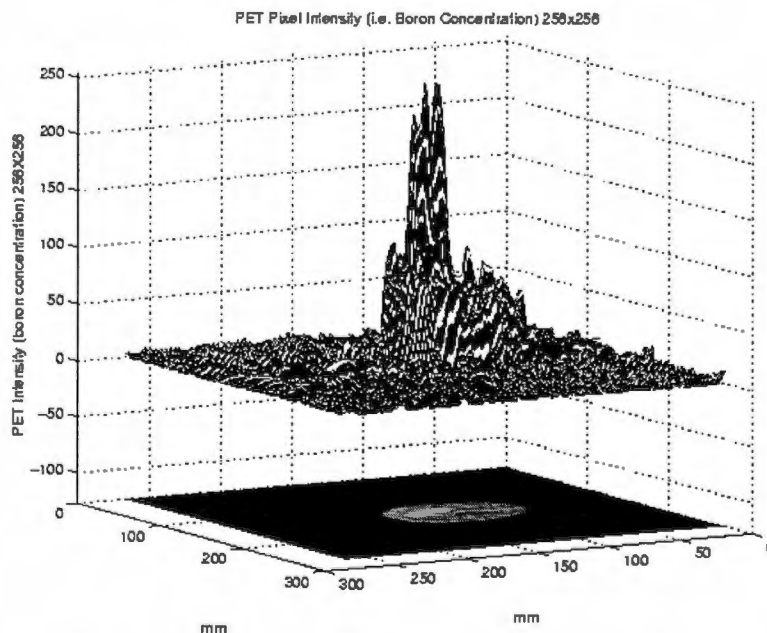
Figure 39: B-10 Pixel Intensity Map For Patient 3

This again validates the notion that tumors are heterogeneous and will behave differently from patient to patient.

Figure 40 shows a side by side comparison of the PETTOPET adjusted dose and the homogeneously adjusted boron dose for patient 3. Finally a comparison of the difference is shown in Figure 41.

Since patient 3 was not treated under the BNL or MIT/Harvard protocol, a correlation between the findings and the outcome of the patient could not be made. However, the need for PET data can be argued given the fact that there were two foci of tumor with each one taking the boron tracer up differently than the other in the same patient. In addition, findings of the adjusted dose seem to be similar to those seen in patients 1 and 2 tumor boundaries, that the conventional method may be over-predicting the dose to those regions.
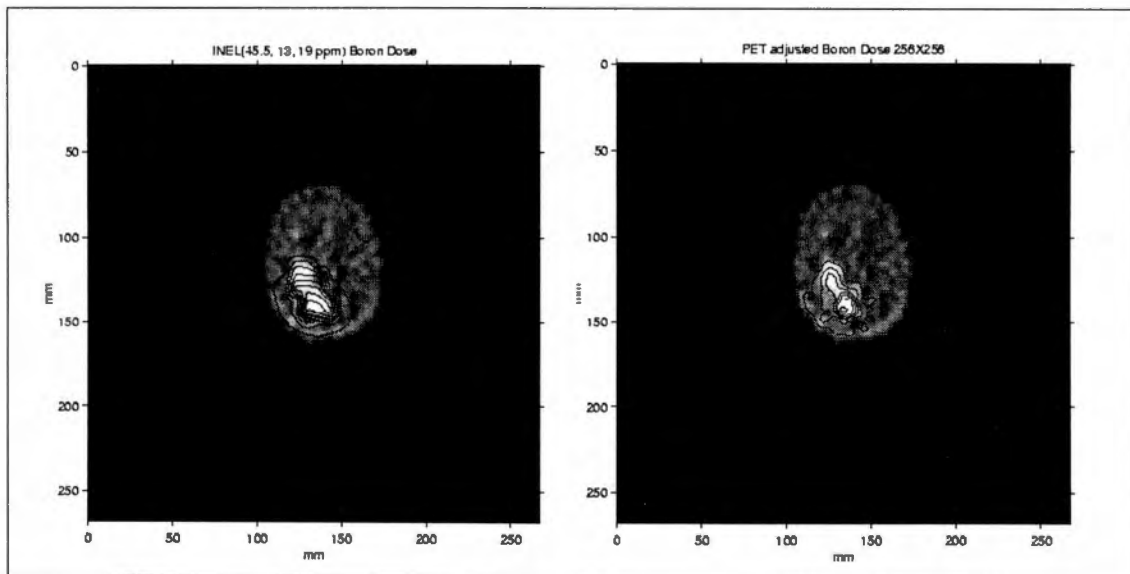
Figure 40: A Side By Side Comparison of PET Adjusted Dosimetry and Homogeneously Scaled Dosimetry Based on 3.5:1 Ratio of Boron Uptake by The Tumor Relative to Normal Brain

70
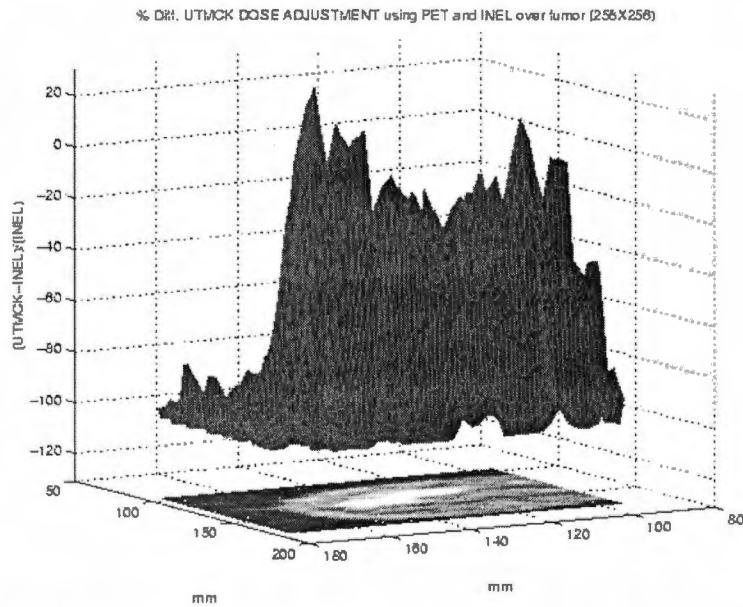
Figure 41: Three Dimensional View of the Difference Between PETTOPET Adjusted Boron Dose and the Homogeneously Assumed Boron Dose

4.0 Patient 4

No pre-surgery data was available for patient 4, thus the results of the PETTOPET adjustment may not be justified since the 4:1 ratio is for the tumor region, which has been surgically removed in this patient. However, it must be pointed out that the PET data did show an increased activity in the faux region immediately after MRI based surgery for patient 4. It is this same region that was seen to re-grow in patient 4 after BNCT treatment. Approximations of the differences seen in the patients are presented in table 2.

71

Table 2: Summary of the Differences in Patient Dosimetry Based on Homogeneous Distribution Vs. PET Adjusted Dosimetry

% Difference	Patient 1	Patient 2	Patient 3
Center of Tumor	~ +/- 10 %	~ +/- 10%	~ -20%
Boundaries of Tumor	~ -50 %	~ -50-80%	~ -80%

- (-) Represents an Over Prediction by Current Methods. (+) Represents an Under Estimate by Current Methods.

VI CONCLUSIONS AND FUTURE WORK

1.0 Conclusions

Based on the data presented, it is clear that one should use PET data in treatment planning. There are remarkable differences seen when one uses the homogeneous assumption about the B-10 distribution as opposed to the heterogeneous boron distribution provided by PET. Based on the data in the four patients the differences varied by as much as 20 – 80 % in some cases with much of the differences showing an overprediction of the dose in some regions of the tumor volume. Also based on patient 2 data, PET data seems to suggest a higher dose to the scalp than that which is predicted by the current protocol.

2.0 Future Work

Although MRI and CT images provide good anatomical descriptions, they fail to provide the quantitative boron-10 distribution that is at the heart of the treatment planning. Thus, the need for PET is very important because it provides the detailed boron distribution needed for accurate treatment planning. In addition, the use of PET data allows for peak thermal flux to be more accurately positioned on the tumor

72

hypocenter as indicated by the maximum uptake value of the specific BNCT patient. When using MRI/CT data, there is no direct way of knowing where the center of the tumor within the tumor volume lies. Another advantage un-realized at the moment, is the pharmacokinetic information that PET provides and is left for future work. In retrospect, improvements in beam positioning taken together with better dosimetry and the pharmacokinetic information provided by PET can significantly improve results and efficacy of the boron agent.

For future work, it is recommend that a utility be developed that allows for co-registration of PET/MRI images and a comparison of the treatment based on MRI images and adjusted via PET be made. Some work has been carried out in regards to this. But due to the complex nature of registering two medical images without external fissures, this has been postponed. In addition, Dan Wessol of Montana State University has pointed out that the new version of BNCT_RTPE, SERA, is being enhanced to allow for the image registration capability. In addition, it is recommend that pharmacokinetic information from PET dynamic study be used to predict the optimal time for treatment. Currently some progress is being made in regards to this. Development of new chemical compounds that preferentially localize at the tumor site at a greater ratio than that seen for current BPA patients would also prove to be invaluable to BNCT. New and improved compounds like ACBC which localize much more preferentially to the tumor site, taken along with improved dosimetry and pharmacokinetic information provided with PET and a better purification of the neutron beam can significantly improve the current status of patients undergoing

73

treatment planning. To summarize, future work remaining to be accomplished include:

a. Incorporation of PET boron distribution in actual treatment.

b. Use of pharmacokinetic information provided by PET in treatment planning.

c. MRI/CT/PET co-registration capability.

d. Voxalized approach to material definition.

e. Development of better localizing pharmaceuticals.

f. Improved beam designs to allow for pure beam with a high neutron flux.

REFERENCES

REFERENCES

1. R. F. Barth, A. H. Soloway, R. G., Fairchild, "Boron Neutron Capture Therapy for Cancer," *Scientif. Am.,* 100 (1990).

2. M. Frederick Hawthorne, "New Horizons for Therapy Based on the Boron Neutron Capture Reaction," Molecular Medicine Today, 1998, 4:4:174-181

3. Sweet, W.H., Javid, M., "The possible Use of Neutron-capturing Isotopes such as boron-10 in the treatment of neoplasms," I. Intracranial Tumors, J. Neurosurg., 9:200-209, (1952)

4. Barth R.F., Soloway A.H. and Brugger R.M. (1996) Boron neutron capture therapy of brain tumors: past history, current status, and future potential. Cancer Invest., 14:534-550

5. Coderre JA, Chanana AD, Joel DD, Elowitz EH, Micca PL, Nawrocky MM, Chadha M, Gebbers JO, Shady M, Peress NS, Slatkin DN. "Biodistribution of boronophenylalanine in patients with glioblastoma multiforme boron concentration correlates with tumor cellularity," Radiation Research. 1998 Feb 149:2 163-70

6. Chadha M, Capala J, Coderre JA, Elowitz EH, Iwai J, Joel DD, Liu HB, Wielopolski L, Chanana AD. Boron neutron-capture therapy (BNCT) for glioblastoma multiforme (GBM) using the epithermal neutron beam at the Brookhaven National Laboratory. Int J Radiat Oncol Biol Phys 1998 Mar 1 40:4 829-34.

7. D.E. Wessol and F.J. Wheeler, "Creating and Using a Type of Free-Form Geometry in Monte Carlo Particle Transport," *Nucl. Sci. Eng.,* 113, 1993, pp. 314-323.

8. Kobayashi, T., Kanda, K., Analytical Calculation of Boron-10 Dosage in Cell Nucleus for Neutron Capture Therapy. *Radiation Research* 1982; 91: 77-94.

9. Dyke, Jon Paul. Dosimetry and Stability Studies of The Boron Neutron Capture Therapy Agent F-BPA-Fr Using PET and MRI. A Dissertation Presented for the Degree of Philosophy. The University of Tennessee, Knoxville. Dec. 1997.

10. Burger, P.C., Kleihues, P., Cytological Composition of the Untreated Glioblastoma With Implications for Evaluation of Needle Biopsies. *Cancer.* 1989; 63:2014-2023.

11. Chilton, Arthur B., Shultis, J. Kenneth., Faw, Richard, E. Principles of Radiation Shielding. Prentice Hall, New Jersey. 1984.

12. McKeever, P.E., Hood, T.W., Varani, J. et.al, Products of Cells Cultured from Gliomas. Cytology and Morhpometry of Two Cell Types cultured from Glioma, *JNCI* 1987: 78, 1:75-84

13. Nigg, D.W., Methods for Radiation Dose Distribution Analysis and Treatment Planning in Boron Neutron Capture Therapy. *Int. J. Rad. Oncol. Biol. Phys.* 1994; 34, 5:1081-1086.

14. Press, William H., Vetterling, William T., Teukolsky, Saul A., Brian P. Flannery. Numerical Recipes in C: The Art of Scientific Computing. Second Edition. Cambridge Press, NY. 1992.

APPENDICES

A.1: PETTOPET_README.txt

The following files are needed to successfully run PETTOPET dose
adjustment subroutine.

(1) PETTOPET.m   - Main program and will call the other
Subroutines

(2) PETTOPET.INPUT - Input file containing a list of tumor planes
to
                     adjust with certain output options
(3) PETTOPET_REGD.m - Subroutine to get region definitions from
                     BNCT_rtpe

(4) PETTOPET_CMAN.m - Subroutine to do the actual contour file
                     manipulation and adjustment

To execute the algorithm, please give the following command within
the Matlab™ working environment:

>>PETTOPET

A.2: PETTOPET.INPUT

```
petimagepixelsize                        =        1.045
plotsizeflag[xdim*ydim 256X256]          =        0 1
FigureSelect[1 2 3 4 5 6 7 8]            =        1 1 1 1 1 1 1 1
INEELFactor [tumor brain scalp] (ppm)    =         45.5 13 19

List of Contour Files :                       [screen    printer]
/ne4/mkhan/programs/pettopet/God
god0926bpa_f24_10.img,contour
-god0926bpa_f24_11.img,contour
-god0926bpa_f24_12.img,contour
-god0926bpa_f24_13.img,contour
-god0926bpa_f24_14.img,contour
-god0926bpa_f24_15.img,contour
god0926bpa_f24_16.img,contour
god0926bpa_f24_17.img,contour
god0926bpa_f24_18.img,contour
god0926bpa_f24_19.img,contour
god0926bpa_f24_20.img,contour
god0926bpa_f24_21.img,contour
god0926bpa_f24_22.img,contour
god0926bpa_f24_23.img,contour
```

A.3: PETTOPET.m

```
% PETTOPET.m: Code to adjust Boron Dose based on PET Pixel Intensity
Scaling
clear all;

sprintf('%s', '********************************')
sprintf('%s', 'This is written by Mohammad Khan')
sprintf('%s', '          Copy righted   1998      ')
sprintf('%s', '********************************')


      Directory = pwd;
%1.   Get pettopet.input file that contains listing of contourfiles
to adjust and the pixel size of corresponding image
      nn=input(' \n Enter the name of the input file
(PETTOPET.INPUT) that you would like to use? \n', 's');
      %nn='pettopet.input';
      inputfid = fopen(nn, 'r');

          %CHECK for the existence of file in the current
directory
          while (inputfid == -1)
              sprintf('Attempt to find the input file failed...
Please try again')
              nn=input(' \n Enter the path to input
(PETTOPET.INPUT) file ? \n', 's');
              inputfid = fopen(nn, 'r');
          end

      hh = 0;    %this is a counter that will count how many contour
files to adjust
              % and get pixel size
      % Read the contour file names from the input file
      global pixelsize plotsize FigureSelect INELfactor
      %n=fgetl(inputfid);
      %default =    str2num(nt(42:size(nt,2)));
      nt=fgetl(inputfid);
      pixelsize=   str2num(nt(42:size(nt,2)));
      nt=fgetl(inputfid);
      plotsize=    str2num(nt(42:size(nt,2)));
      nt=fgetl(inputfid);
      FigureSelect=str2num(nt(42:size(nt,2)));
      nt=fgetl(inputfid);
      INELfactor=  str2num(nt(42:size(nt,2)));
      nt=fgetl(inputfid);
      nt=fgetl(inputfid);
      nt=fgetl(inputfid);
      Directory_PET = deblank(nt);


%2.   Get Region Definitions
      cd(Directory);
```

```
        PETTOPET_REGD                        %GET region definitions
        cd(Directory_PET);

while(feof(inputfid) ~= 1)
        nn='PTP.out';
        hh= hh+1;

    n1=fgetl(inputfid);
            if n1 == -1,
                    break;
            end
        if size(n1, 2) > 42
                nt=deblank(n1(1:35));   % Gets the contour file name from
the input
                printopts= str2num(n1(42:size(n1,2)));
        else
                nt = deblank(n1);
                printopts= [1 0];
        end
        if (nt(1) == '-')
                %input('stop');
                special_flag(hh) = 1;   %Flags used to selcect certain
planes from input and write out results as bitmaps
                                %Automatically
                nt=nt(size(nt,1), 2:size(nt,2));   %Remove flag from name
        else
                special_flag(hh) = 0;

        end


        contourfid=fopen(nt, 'r');
        sprintf('Working on %s file... Please be patient! ..', nt)

        while (contourfid == -1)
                sprintf('Attempt to find %s failed... Please try again',
nt)
                nt=input(' \n please enter the path to that file ? \n',
's');
                contourfid = fopen(nt, 'r');
        end
        adjustedcontour=strcat(nt, nn);
        output=fopen(adjustedcontour, 'w');

% 3.        Read the first 13 (header Lines) of the contour file
        i=0; h=zeros;
        for i=1: 13;
                line=fgetl(contourfid);
                fprintf(output, '%s \n', line);      % Write the headers
out to output file
                h=str2mat(h,line);
            end
        gridsize = str2num(h(3, :));
        xdim = gridsize(9);         % xdim is the X grid size
        ydim = gridsize(10);        % ydim is the y grid size
```

```
%4.   Now Find the image that corresponds to the that particular
contour file
      imagefilename = h(5, 32:114);
      filename= setstr(imagefilename);
      imagefid=fopen(filename, 'r');


      %Check to see if the image file exists in present directory,
if not prompt.
      while (imagefid == -1)
      sprintf('\n I Could not find "%s" imagefile in the current
directory please let me know where it is? ', filename)
      imagefile =input('(images/test16.img) ? ', 's');
      imagefid = fopen(imagefile, 'r');   % 'r' means read only
      % This part will scale the x and y values based on a pixel
size in mm
         pixelsize=input('\n please enter the pixel size of the image
in mm ?  ')
      % pixelsize= 2.01/2;
      end


%5    Now Read the rest of the file into a matrix in a sturctured
format
         x=zeros(1, xdim*ydim);
      y=zeros(1, xdim*ydim);
      z=zeros(1, xdim*ydim);
      r=zeros(1, xdim*ydim);
      t=zeros(1, xdim*ydim);
      b=zeros(1, xdim*ydim);
      g=zeros(1, xdim*ydim);
      n=zeros(1, xdim*ydim);
      f=zeros(1, xdim*ydim);
      g1=zeros(1, xdim*ydim);
      g2=zeros(1, xdim*ydim);
      th=zeros(1, xdim*ydim);
        i=1;
        while(feof(contourfid) ~=1 )
      line=fgetl(contourfid);
      x(1, i)= str2num(line(1:10));
      y(1, i)= str2num(line(11:20));
      z(1, i)= str2num(line(21:30));
      r(1, i)= str2num(line(31:34));
      t(1, i)= str2num(line(35:44));
      b(1, i)= str2num(line(45:54));
      g(1, i)= str2num(line(55:64));
      n(1, i)= str2num(line(65:74));
      f(1, i)= str2num(line(75:84));
      g1(1, i)=str2num(line(85:94));
      g2(1, i)=str2num(line(95:104));
      th(1, i)=str2num(line(105:114));
      i=i+1;
```

```
        end
        fclose(contourfid);


            if (max(max(r)) ~= tumor_region)
                sprintf(' No tumor region found in this contour')
                sprintf(' No adjustment done to this plane')

        else

                %6      Call Contour_man3 to do the adjustment and
reading of pet image file
                cd(Directory)
                PETTOPET_CMAN
                cd(Directory_PET)

        fprintf(output,'%10.3e%10.3e%10.3e%4.f%10.3e%10.3e%10.3e%10.3e
%10.3e%10.3e%10.3e%10.3e%10.3e\n', a1);
                fclose(output);


                %fclose(imagefid);
        end

end

fclose('all')
```

A.3: PETTOPET_REGD.m

```
% This file gets the region definitions from rtt.input and the .rm
file
% It reads the rtt.input file, looks for the .rm file and uses the
same
% prefix to find the .rs file that contains region defintions (3 is
added
% to each of the region definitions in that file so that it will be
consistent
% with the contour file (A suggestion by Chuck Wimperle via Email).

cd(Directory_PET);
nn ='rtt.input';
%1    Get the rtt.input file generated by BNCT_rtpe
      rtt_input_fid=fopen('rtt.input','r');
      while (rtt_input_fid == -1)
            sprintf('Attempt to find %s in the current directory
failed... Please try again', nn)
            nn=input(' \n Enter the path to rtt.input file used by
BNCT_rtpe? \n', 's');
            rtt_input_fid = fopen(nn, 'r');
      end

%2    Read the file to find the output file
      i=0; rtt_input_h=zeros; rm_line_h=zeros;
      for i=1:13,
       rtt_line=fgetl(rtt_input_fid);
       rtt_input_h=str2mat(rtt_input_h, rtt_line);
      end

      w=0;
      for i=1:80,
       space = rtt_input_h(7,:)-32;
      if space(i) ~= 0
            space(i);
       rs_file(1,i)=space(i);
      end
      end

      haha = size(rs_file);
      haha =haha(1,2);

%3    Get the .rm file that contains region definitions
      rm_file=setstr(rs_file(1,1:haha-2)+32);
      rm_file=strcat(rm_file, 'rm');
      rm_file_fid = fopen(rm_file, 'r');
      while (rm_file_fid == -1)
            sprintf('Attempt to find %s in the current directory
failed...', rm_file)
            rm_file=input('Enter the path to this file written out
by BNCT_rtpe ? \n', 's');
            rm_file_fid = fopen(nn, 'r');
      end
```

```
%4      Read the .rm file and specify the regions, tumor, scalp,
skull, brain
        while(feof(rm_file_fid) ~=1)

                rm_line=fgetl(rm_file_fid);
                rm_line_h=str2mat(rm_line_h, rm_line);
        end

        haha=size(rm_line_h);
        haha=haha(1,1)-1;
        hehe=size(rm_line_h);


        number_of_regions=haha-2;
        regions=zeros(1,number_of_regions);

        i=0;
        for i = 1:number_of_regions,
                a=sscanf(rm_line_h(i+2,:),'%d %s')';
                aa=char(a);
                if (isempty(findstr(lower(aa), lower('skull'))) ~= 1)
                        skull_region = a(1)+3;
                elseif (isempty(findstr(lower(aa), lower('brain'))) ~=1)
                        brain_region = a(1)+3;
                elseif (isempty(findstr(lower(aa), lower('tumor'))) ~=1)
                        tumor_region = a(1)+3;
                elseif (isempty(findstr(lower(aa), lower('scalp'))) ~=1)
                        scalp_region =a(1)+3;
                end
        end
```

A.4: PETTOPET_CMAN.m

```
%  ********************************************************%
%  *    The code reads the contour matrix        %
%  *    and adjusts the  contours based on the 256X256  %
%  *    Raw PET images.  The doses are linearly scaled  %
%  *  ·based on peak to average intensity of the image %
%  *    from the PET                             %
%  *              Written by              %
%  *              Mohammad K. Khan             %
%  *        University Of Tennessee, Knoxville  %
%  *              September 9, 1998           %
%  ********************************************************%

% Figure 1 = INEL BORON DOSE from BNCT_rtpe (2D)
% Figure 2 =  3D view of the INEL BORON DOSE (unadjusted)
% Figure 3 = PET image and Pixel Intesity Plot (3D view)
% Figure 4 = PET adjusted Boron Dose (3D view)
% Figure 5 = PET adjusted Boron Dose contour (2D view)
% Figure 6 = Boron Dose (Linearly Scaled) Based on INEL numbers
(45.5ppm, 13 ppm, 19 ppm)
% Figure 7 = 3D view of Figure 6
% Figure 8 = Difference over the tumor region between INEL
homogeneous
%            Scaling and Patient Specific PET.


        image=filename;
        a1= [x' y' z' r' t' b' g' n' f' g1' g2' th'];

        contourgridsize=[xdim ydim];
        boron=zeros(contourgridsize);
        nitrogen=zeros(contourgridsize);
        gamma=zeros(contourgridsize);
        region=zeros(contourgridsize);
        total=zeros(contourgridsize);
        group1=zeros(contourgridsize);
        group2=zeros(contourgridsize);


%1.   %Use Spline Interpolation to get an X and Y array to be
256X256 positions.
        Px=x(1:xdim); xx=spline([1:xdim],Px, [1:1/(255/(xdim-
1)):xdim]);
        Py=y(1:xdim+1:xdim*ydim); yy=spline([1:ydim], Py,
[1:1/(255/(xdim-1)):ydim]);
        xnewposition=zeros(xdim,1);
        ynewposition=zeros(ydim,1);
        xnewposition=(1+xx)/2*pixelsize*255;  % transform range from
[-1 1] to [1 256]*pixelsize.
        ynewposition=(1+yy)/2*pixelsize*255;
        xx=xnewposition;
        yy=ynewposition;
```

87

```matlab
% Read all of the values into a matrix (xdym * ydim) size
nn=1;
for i=1:ydim,
  for j=1:xdim,
        boron(i,j)=b(1, nn);
        region(i,j)=r(1,nn);
        total(i,j)=t(1,nn);
        gamma(i,j)=g(1,nn);
        nitrogen(i,j)=n(1,nn);
        fast(i,j)=f(1, nn);
            %group1(i,j)=g1(1, nn);
            %group2(i,j)=g2(1, nn);
            %therma(i,j)=th(1, nn);
        nn=nn+1;
    end
        if nn> (xdim*ydim)
            break
        end
end
reg= imresize(region, [256 256],'bilinear');
 b_10= imresize(boron, [256 256], 'bilinear');
tot= imresize(total, [256 256], 'bilinear');
gam= imresize(gamma, [256 256], 'bilinear');
nit= imresize(nitrogen, [256 256], 'bilinear');
fas= imresize(fast, [256 256], 'bilinear');

reg = flipud(reg);
b_10 = flipud(b_10);
tot = flipud(tot);
gam = flipud(gam);
nit = flipud(nit);
fas = flipud(fas);


%2      *************************** Read the PET image ************
        % ******************************************************
        cd(Directory_PET);
        FID=fopen(filename, 'r');
        cd(Directory);
        xi=([0:255]*pixelsize);
        yi=([0:255]*pixelsize);
        imagematrix=zeros(256);

        imagematrix=fread(FID, [256 256], 'uint8');
        fclose(FID);
          imagematrix = imagematrix';

        %Set default settings for the imageprocessing tool box;
        % imagematrix=flipud(imagematrix);
        iptsetpref('ImshowBorder', 'loose');
        iptsetpref('ImshowAxesVisible', 'on');
```

```
% Compute Average over regions

n1=0; sum1=0;
n2=0; sum2=0;
for i=1:256,
for j=1:256,

     if reg(i,j)>=tumor_region    % region
          n1=n1+1;
          sum1=sum1+imagematrix(i,j);
     end
     if reg(i,j)>= brain_region      % over brain region
          n2=n2+1;
          sum2=sum2+imagematrix(i,j);
     end
end
end
tumoraverage= sum1/n1;
brainaverage= sum2/n2;
entireregion= sum(sum(imagematrix)')/(256*256);

adjusteddose=zeros(256); adjusteddose1=zeros(256);
adjusteddose2=zeros(256);
% adjusteddose=b_10.*imagematrix/entireregion;
% adjusteddose1=b_10.*imagematrix/tumoraverage;
% for i=1:256,
% for j=1:256,

%       if reg(i,j)>= tumor_region
%             n1=n1+1;
%
adjusteddose2(i,j)=b_10(i,j)*imagematrix(i,j)/brainaverage;
%       end
%
%       if ( reg(i,j)>=tumor_region & reg(i,j) < brain_region)
%             n2=n2+1;
%
adjusteddose2(i,j)=b_10(i,j)*imagematrix(i,j)/brainaverage;
%       end
%
%       if (reg (i,j)>=skull_region & reg(i,j) < brain_region)
%
adjusteddose2(i,j)=b_10(i,j)*imagematrix(i,j)/brainaverage;
%
%       end
%end
%end


adjusteddose2=b_10.*imagematrix/brainaverage;   %Peak To
Average Scaling
```

```
            tot=adjusteddose2+gam+nit+fas;
            b_adjusted=imresize(adjusteddose2, [xdim ydim],
'bilinear');   %Scale it back to xdim by ydim
            total_adjusted=imresize(tot, [xdim ydim], 'bilinear');

            ss=0; bb = zeros(1, xdim*ydim);
            for i = 1:xdim,
                for j= 1:ydim,
                    ss=ss+1;
                    bb(ss)=b_adjusted(i,j);
                    t(ss) = total_adjusted(i,j);
                end
            end


    a1= [x' y' z' r' t' b' bb' g' n' f' g1' g2' th']';

%3     % ***********This part will plot the Figures ****************



            hhh=0;
            hhh =hh-1;
            maximum_pixel_intensity=max(max(imagematrix)); % Used
for range to show image intensity
            I=imagematrix;
            if maximum_pixel_intensity > 200

    maximum_pixel_intensity=max(max(imagematrix))/1.25;
            end


            % For xdim * Ydim option Scale Everything down to xdim X
ydim
            if plotsize(1)== 1
                imagematrix2 = imresize(imagematrix, [xdim ydim],
'bilinear');
                I2=imagematrix2;
                ad_xdimydim=boron.*imagematrix2/brainaverage;
%adjusteddose calculated on xdim X Ydim grid
                newpixelsizex=256/xdim*pixelsize;
rangex=[1:newpixelsizex:xdim*newpixelsizex];
                newpixelsizey=256/ydim*pixelsize;
rangey=[1:newpixelsizey:ydim*newpixelsizey];
                x_xdim=Px;
                y_ydim=Py;
                xnewx_xdim=zeros(xdim,1);
                ynewy_ydim=zeros(ydim,1);
                xnewx_xdim=(1+x_xdim)/2*newpixelsizex*xdim;   %
transform range from [-1 1] to [1 256]*pixelsize.
                ynewy_ydim=(1+y_ydim)/2*newpixelsizey*ydim;
                x_xdim=xnewx_xdim;
                y_ydim=ynewy_ydim;

            end
```

```
printopts;
if (printopts(2) ==1)
           print -dps adjust_pettopet_out
end

if (printopts(1) == 1 )                    % Print to screen

           if (FigureSelect(1) == 1)
                         Figure(1+hhh*8)

                         if (plotsize(2)==1 & plotsize(1) ~= 1)
       %256X256 Grid

                                imshow(xi, yi, imagematrix, [0
maximum_pixel_intensity]);
                                hold on; colormap('bone');
                                contour(xx, yy, b_10); title('Boron
Dose Contour');
                                xlabel('mm'); ylabel('mm');
                                hold off;

                         elseif (plotsize(2)~=1 & plotsize(1) ==1)
%xdim X Ydim selection
                                imshow(rangex, rangey, imagematrix2,
[0 maximum_pixel_intensity]);
                                hold on; colormap('bone');

                                contour(x_xdim, y_ydim, boron)
                                title('Boron Dose Contour');
                                xlabel('mm'); ylabel('mm');
                                hold off;

                         elseif (plotsize(2) ==1 & plotsize(1) ==1)


                                subplot(1, 2, 1);              %256 X
256 Plot
                                imshow(xi, yi, imagematrix, [0
maximum_pixel_intensity]);
                                hold on; colormap('bone');
                                contour(xx, yy, b_10); title('Boron
Dose Contour 256X256');
                                xlabel('mm'); ylabel('mm');
                                hold off;

                                subplot(1, 2, 2);
                                imshow(rangex, rangey, imagematrix2,
[0 maximum_pixel_intensity]);
                                hold on; colormap('bone');

                                contour(x_xdim, y_ydim, boron)
                                title('Boron Dose Contour Xdim * Ydim
');
                                xlabel('mm'); ylabel('mm');
```

```
                              hold off;
                      end
                      %if (printopts(2) ==1)
                      %     print -dps -append adjust_pettopet_out

                      %end
                      if special_flag(hh) == 1

                              say = setstr(num2str(1+hhh*8));

                              say2 = strcat('fig_P2PMC2D_', say);
                              say2 = strcat(say2, '.bmp');
                              num = 1+hhh*8;
                              printstring = sprintf('print -
fFigure(%d) -dpgm %s', num, say2);
                              eval(printstring)
                              %print fig_MC2D_1.bmp -fFigure(1) -
dpgm
                      end

              end


          if (FigureSelect(2)==1)
                      Figure(2+hhh*8);

                      if (plotsize(2)==1 & plotsize(1) ~= 1)
      %256X256 Grid
                              plotat=.5*((min(min(b_10)')-
max(max(b_10)')));

                              h=surf(xx, yy, b_10);
                              set(h, 'Edgecolor', 'blue');
                              hold on;colormap('bone');
                              I=imagematrix;
                              h2= warp(xi, yi,
plotat*ones(size(b_10)), I, [0 maximum_pixel_intensity]);
                              hold off; colormap('bone');
                              set(gca, 'zlim', [plotat
max(max(b_10)')]);

                              zlabel('Boron Dose 256 X 256');
                              xlabel(' mm');
                              ylabel(' mm');
                              title('BNCT_rtpe Unadjusted boron dose
256X256 ')

                              view(60, 10);


                      elseif (plotsize(2) ~=1 & plotsize(1) ==1)

                              plotat= .5*((min(min(boron)')-
max(max(boron)')));

                              h=surf(x_xdim, y_ydim, boron);
                              set(h, 'Edgecolor', 'blue');
                              hold on;colormap('bone');
                              I2=imagematrix2;
```

```
                                    h2=warp(rangex, rangey,
plotat*ones(size(boron)), I2, [0 maximum_pixel_intensity]);
                                    hold off; colormap('bone');
                                    set(gca, 'zlim', [plotat
max(max(b_10)')]);

                                    zlabel('Boron Dose Xdim * Ydim');
                                    xlabel(' mm');
                                    ylabel(' mm');
                                    title('BNCT_rtpe Unadjusted boron dose
xdim X ydim')
                                    view(60, 10);


                       elseif (plotsize(2) ==1 & plotsize(1) ==1)

                                    subplot(1,2,1)
                                    plotat=.5*((min(min(b_10)')-
max(max(b_10)')));

                                    h=surf(xx, yy, b_10);
                                    set(h, 'Edgecolor', 'blue');
                                    hold on; colormap('bone');
                                    I=imagematrix;
                                    h2= warp(xi, yi,
plotat*ones(size(b_10)), I, [0 maximum_pixel_intensity]);
                                    hold off;
                                    set(gca, 'zlim', [plotat
max(max(b_10)')]);

                                    zlabel('Boron Dose 256 X 256');
                                    xlabel(' mm');
                                    ylabel(' mm');
                                    title('BNCT_rtpe Unadjusted boron dose
256X256 ')
                                    view(60, 10);

                                    subplot(1,2,2)
                                    plotat= .5*((min(min(boron)')-
max(max(boron)')));

                                    h=surf(x_xdim, y_ydim, boron);
                                    set(h, 'Edgecolor', 'blue');
                                    hold on; colormap('bone');
                                    I2=imagematrix2;
                                    h2=warp(rangex, rangey,
plotat*ones(size(boron)), I2, [0 maximum_pixel_intensity]);
                                    hold off;
                                    set(gca, 'zlim', [plotat
max(max(b_10)')]);

                                    zlabel('Boron Dose Xdim * Ydim');
                                    xlabel(' mm');
                                    ylabel(' mm');
                                    title('BNCT_rtpe Unadjusted boron dose
xdim X ydim')
                                    view(60, 10);
                       end
                       %if (printopts(2) ==1)
                       %      print -dps -append adjust_pettopet_out
```

93

```
                              %end
                           .if special_flag(hh) == 1

                                   say = setstr(num2str(2+hhh*8));

                                   say2 = strcat('fig_MC3D_', say);
                                   say2 = strcat(say2, '.bmp');
                                   num = 2+hhh*8;
                                   printstring = sprintf('print -
fFigure(%d) -dpgm %s',.num, say2);
                                   eval(printstring);

                       ·end
               end

                   if (FigureSelect(3) ==1)
                           Figure(3+hhh*8)

                           if (plotsize(2)==1 & plotsize(1) ~= 1)
     %256X256 Grid

       plotat=floor(.5*((min(min(imagematrix)')-
max(max(imagematrix)'))));
                                   h=mesh(xi,yi,I);
                                   set(h, 'Edgecolor', 'blue');
                                   hold on; colormap('bone');
                                   h2= warp(xi, yi,
 plotat*ones(size(imagematrix)), I, [0 maximum_pixel_intensity]);
                                   hold off;
                                   set(gca, 'zlim', [plotat
max(max(imagematrix)')]);
                                   title('PET Pixel Intensity (i.e. Boron
Concentration) 256x256');
                                   xlabel(' mm');
                                   ylabel(' mm');
                                   zlabel(' PET Intensity (boron
concentration) 256X256')

                                   view(60, 10);

                           elseif (plotsize(2) ~=1 & plotsize(1) ==1) %
Xdim X Ydim Grid

       plotat=floor(.5*((min(min(imagematrix2)')-
max(max(imagematrix2)'))));
                                   h=mesh(rangex, rangey, imagematrix2);
                                   set(h, 'Edgecolor', 'Blue');
                                   hold on; colormap('bone');
                                   h2 = warp(rangex, rangey,
plotat*ones(size(imagematrix2)), I2, [0 maximum_pixel_intensity]);
                                   hold off;
                                   set(gca, 'zlim', [plotat
max(max(imagematrix2)')]);
                                   title('PET pixel Intensity (i.e. Boron
Concentration) Xdim x Ydim');
```

94

```
                                      xlabel(' mm');
                                      ylabel(' mm');
                                      zlabel(' PET Intensity (boron
concentration) xdim X Ydim');
                                      view(60, 10);


                      elseif (plotsize(2) ==1 & plotsize(1) == 1)
                                      subplot(1, 2, 1)

        plotat=floor(.5*((min(min(imagematrix)')-
max(max(imagematrix)')))));
                                      h=mesh(xi,yi,I);
                                      set(h, 'Edgecolor', 'blue');
                                      hold on; colormap('bone');
                                      h2= warp(xi, yi,
plotat*ones(size(imagematrix)), I, [0 maximum_pixel_intensity]);
                                      hold off;
                                      set(gca, 'zlim', [plotat
max(max(imagematrix)')]);
                                      title('PET Pixel Intensity (i.e. Boron
Concentration) 256x256');
                                      xlabel(' mm');
                                      ylabel(' mm');
                                      zlabel(' PET Intensity (boron
concentration) 256X256')
                                      view(60, 10);

                                      subplot(1, 2, 2)

        plotat=floor(.5*((min(min(imagematrix2)')-
max(max(imagematrix2)')))));
                                      h=mesh(rangex, rangey, imagematrix2);
                                      set(h, 'Edgecolor', 'Blue');
                                      hold on; colormap('bone');
                                      h2 = warp(rangex, rangey,
plotat*ones(size(imagematrix2)), I2, [0 maximum_pixel_intensity]);
                                      hold off;
                                      set(gca, 'zlim', [plotat
max(max(imagematrix2)')]);
                                      title('PET pixel Intensity (i.e. Boron
Concentration) Xdim x Ydim');
                                      xlabel(' mm');
                                      ylabel(' mm');
                                      zlabel(' PET Intensity (boron
concentration) xdim X Ydim');
                                      view(60, 10);

                      end
                      %if (printopts(2) ==1)
                      %      print -dps -append adjust_pettopet_out

                      %end

                      if special_flag(hh) == 1
```

```matlab
                              say = setstr(num2str(3+hhh*8));

                              say2 = strcat('fig_PETINT3D_', say);
                              say2 = strcat(say2, '.bmp');
                              num = 3+hhh*8;
                              printstring = sprintf('print -
fFigure(%d) -dpgm %s', num, say2);
                              eval(printstring);


                      end
              end

              if (FigureSelect(4) == 1)
                      Figure(4+hhh*8)

                      if (plotsize(2)==1 & plotsize(1) ~= 1)
        %256X256 Grid

        plotat=floor(.5*((min(min(adjusteddose2)')-
max(max(adjusteddose2)')))));
                              h=mesh(xx, yy, adjusteddose2);
                              set(h, 'Edgecolor', 'blue');
                              hold on; colormap('bone');
                              h2= warp(xi, yi,
plotat*ones(size(adjusteddose2)), I, [0 maximum_pixel_intensity]);
                              hold off;
                              set(gca, 'zlim', [plotat
max(max(adjusteddose2)')]);

                              zlabel('PET Adjusted Boron Dose');
                              xlabel(' mm');
                              ylabel(' mm');
                              title('PET Adjusted Dose (3D) 256X256
')

                              view(60, 10)

                      elseif (plotsize(2) ~=1 & plotsize(1) ==1) %
Xdim X Ydim Grid

        plotat=floor(.5*((min(min(ad_xdimydim)')-
max(max(ad_xdimydim)')))));
                              h=mesh(x_xdim, y_ydim, ad_xdimydim);
                              set(h, 'Edgecolor', 'blue');
                              hold on; colormap('bone');
                              h2= warp(rangex, rangey,
plotat*ones(size(ad_xdimydim)), I2, [0 maximum_pixel_intensity]);
                              hold off;
                              set(gca, 'zlim', [plotat
max(max(ad_xdimydim)')]);

                              zlabel('PET Adjusted Boron Dose');
                              xlabel(' mm');
                              ylabel(' mm');
                              title('PET Adjusted Dose (3D) xdim X
ydim ')

                              view(60, 10)
```

```
elseif (plotsize(2) ==1 & plotsize(1) == 1)

                subplot(1, 2, 1)

    plotat=floor(.5*((min(min(adjusteddose2)')-
max(max(adjusteddose2)')))));
                h=mesh(xx, yy, adjusteddose2);
                set(h, 'Edgecolor', 'blue');
                hold on; colormap('bone');
                h2= warp(xi, yi,
plotat*ones(size(adjusteddose2)), I, [0 maximum_pixel_intensity]);
                hold off;
                set(gca, 'zlim', [plotat
max(max(adjusteddose2)')]);
                zlabel('PET Adjusted Boron Dose');
                xlabel(' mm');
                ylabel(' mm');
                title('PET Adjusted Dose (3D) 256X256
')
                view(60, 10)

                subplot(1, 2, 2)

    plotat=floor(.5*((min(min(ad_xdimydim)')-
max(max(ad_xdimydim)')))));
                h=mesh(x_xdim, y_ydim, ad_xdimydim);
                set(h, 'Edgecolor', 'blue');
                hold on; colormap('bone');
                h2= warp(rangex, rangey,
plotat*ones(size(ad_xdimydim)), I2, [0 maximum_pixel_intensity]);
                hold off;
                set(gca, 'zlim', [plotat
max(max(ad_xdimydim)')]);
                zlabel('PET Adjusted Boron Dose');
                xlabel(' mm');
                ylabel(' mm');
                title('PET Adjusted Dose (3D) xdim X
ydim ')
                view(60, 10)
            end
            %if (printopts(2) ==1)
            %      print -dps -append adjust_pettopet_out

            %end
            if special_flag(hh) == 1

                say = setstr(num2str(4+hhh*8));

                say2 = strcat('fig_PETADJ3D_', say);
                say2 = strcat(say2, '.bmp');
                num = 4+hhh*8;
                printstring = sprintf('print -
fFigure(%d) -dpgm %s', num, say2);
                eval(printstring);
```

97

```
                        end
                end

        if (FigureSelect(5) == 1)
                Figure(5+hhh*8)
                        if (plotsize(2)==1 & plotsize(1) ~= 1)
    %256X256 Grid
                                imshow(xi, yi, imagematrix, [0
maximum_pixel_intensity]);

                                hold on; colormap('bone');
                                contour(xx, yy, adjusteddose2);
                                title('PET adjusted Boron Dose
256X256');

                                xlabel('mm'); ylabel('mm');
                                hold off;

                        elseif (plotsize(2) ~=1 & plotsize(1)
==1) % Xdim X Ydim Grid
                                imshow(rangex, rangey,
imagematrix2, [0 maximum_pixel_intensity]);
                                hold on; colormap('bone');
                                contour(x_xdim, y_ydim,
ad_xdimydim);
                                title('PET adjusted Boron Dose
40X40');

                                xlabel('mm'); ylabel('mm');
                                hold off;

                        elseif (plotsize(2) ==1 & plotsize(1)
== 1)
                                subplot(1, 2, 1)
                                imshow(xi, yi, imagematrix, [0
maximum_pixel_intensity]);

                                hold on; colormap('bone');
                                contour(xx, yy, adjusteddose2);
                                title('PET adjusted Boron Dose
256X256');

                                xlabel('mm'); ylabel('mm');
                                hold off;

                                subplot(1, 2, 2)
                                imshow(rangex, rangey,
imagematrix2, [0 maximum_pixel_intensity]);
                                hold on; colormap('bone');
                                contour(x_xdim, y_ydim,
ad_xdimydim);
                                title('PET adjusted Boron Dose
40X40');

                                xlabel('mm'); ylabel('mm');
                                hold off;
                        end
                        %if (printopts(2) ==1)
                        %print -dps -append
adjust_pettopet_out
                        %end
```

98

```
                    if special_flag(hh) == 1

                            say = setstr(num2str(5+hhh*8));

                            say2 = strcat('fig_PETADJ2D_', say);
                            say2 = strcat(say2, '.bmp');
                            num = 5+hhh*8;
                            printstring = sprintf('print -
fFigure(%d) -dpgm %s', num, say2);
                            eval(printstring);

                    end
            end


        %4      Calculate Difference between predicted INEL
homogeneous calculation and PET adjusted
        %       This part of code will show a comparison of what
the INEL predictions would like if the dose was linearly scaled by
45.5 ppm in the tumor
        %       region, 19 ppm in the skull/scalp region, and 13
ppm in the brain region.


        %1      Calculate B-10 Dose based on INEL numbers (tumor =
45.5 ppm; brain = 13ppm; scalp = 19ppm)
        %1      One can use whatever numbers to scale the dose by
using the input file.
            for i=1:xdim,
                for j=1:ydim,
                        if region(i,j)>=tumor_region     % region =
Tumor

      boron(i,j)=boron(i,j)*INELfactor(1)/INELfactor(2);
                        end
                        if ( region(i,j)>= brain_region & region(i,j) <
tumor_region ) % Brain Region Only
                            boron(i,j)=boron(i,j)*INELfactor(2)/INELfactor(2);

                        end
                        if(region(i,j) >= skull_region & region(i,j) <
brain_region)  % Scalp/Skull Region
                            boron(i,j) =
boron(i,j)*INELfactor(3)/INELfactor(2);
                        end

                end
            end


        %2      This part will resize the xdim*ydim boron size  to
256X256 to be consistent with image size (256X256)
            reg= imresize(region, [256 256], 'bilinear');
            b_10= imresize(boron, [256 256], 'bilinear');

            boron = flipud(boron);
```

99

```matlab
                reg = flipud(reg);
                b_10=flipud(b_10);

        %3      ***********This part will plot the Figures
*****************

                hhh=0;
                hhh=hh-1;

                if FigureSelect(6) == 1
                        Figure(6+hhh*8)
                                if (plotsize(2)==1 & plotsize(1) ~= 1)
        %256X256 Grid
                                        I=imagematrix;
                                        imshow(xi, yi, imagematrix, [0
maximum_pixel_intensity]);
                                        hold on; colormap('bone');
                                        contour(xx, yy, b_10);
title('INEL(45.5, 13, 19 ppm) Boron Dose');
                                        xlabel('mm'); ylabel('mm');
                                        hold off;

                                elseif (plotsize(2) ~=1 & plotsize(1) == 1)
                                        I2=imagematrix2;
                                        imshow(rangex, rangey, imagematrix2,
[0 maximum_pixel_intensity]);
                                        hold on; colormap('bone');
                                        contour(x_xdim, y_ydim, boron);
                                        title('INEL or other scaled factor
Boron Dose Xdim X Ydim ');
                                        xlabel('mm'); ylabel('mm');
                                        hold off;

                                elseif (plotsize(2) ==1 & plotsize(1) == 1)
                                        subplot(1, 2, 1)
                                        I=imagematrix;
                                        imshow(xi, yi, imagematrix, [0
maximum_pixel_intensity]);
                                        hold on; colormap('bone');
                                        contour(xx, yy, b_10);
title('INEL(45.5, 13, 19 ppm) Boron Dose');
                                        xlabel('mm'); ylabel('mm');
                                        hold off;

                                        subplot(1,2,2)
                                        I2=imagematrix2;
                                        imshow(rangex, rangey, imagematrix2,
[0 maximum_pixel_intensity]);
                                        hold on; colormap('bone');
                                        contour(x_xdim, y_ydim, boron);
                                        title('INEL or other scaled factor
Boron Dose Xdim X Ydim ');
                                        xlabel('mm'); ylabel('mm');
                                        hold off;
                                end
```

100

```
                %if (printopts(2) ==1)
                %       print -dps -append adjust_pettopet_out

                %end
                if special_flag(hh) == 1

                        say = setstr(num2str(6+hhh*8));

                        say2 = strcat('fig_INELADJ2D_', say);
                        say2 = strcat(say2, '.bmp');
                        num = 6+hhh*8;
                        printstring = sprintf('print -
fFigure(%d) -dpgm %s', num, say2);
                        eval(printstring);


                    end
            end

        if FigureSelect(7)== 1
                Figure(7+hhh*8);
                        if (plotsize(2)==1 & plotsize(1) ~= 1)
        %256X256 Grid
                                plotat=floor(.5*((min(min(b_10)')-
max(max(b_10)')))));
                                h=mesh(xx, yy, b_10);
                                set(h, 'Edgecolor', 'blue');
                                hold on; colormap('bone');
                                h2= warp(xi, yi,
plotat*ones(size(b_10)), imagematrix, [0 maximum_pixel_intensity]);
                                hold off;
                                set(gca, 'zlim', [plotat
max(max(b_10)')]);

                                zlabel('Relative Boron Dose');
                                xlabel(' mm');
                                ylabel(' mm');
                                title('INEEL Factor adjusted boron
dose 256X256 ')

                                view(60, 10);

                        elseif (plotsize(2) ~=1 & plotsize(1) == 1)
                                plotat=floor(.5*((min(min(boron)')-
max(max(boron)')))));

                                h=mesh(x_xdim, y_ydim, boron);
                                set(h, 'Edgecolor', 'blue');
                                hold on; colormap('bone');
                                h2= warp(rangex, rangey,
plotat*ones(size(boron)), imagematrix2, [0
maximum_pixel_intensity]);
                                hold off;
                                set(gca, 'zlim', [plotat
max(max(boron)')]);

                                zlabel('Relative Boron Dose');
                                xlabel(' mm');
                                ylabel(' mm');
                                title('INEEL Factor adjusted boron
```

```
dose xdim * ydim ')
                                        view(60, 10);

                         elseif (plotsize(2) ==1 & plotsize(1) == 1)
                              subplot(1, 2, 1)
                              plotat=floor(.5*((min(min(b_10)')-
max(max(b_10)')))));
                              h=mesh(xx, yy, b_10);
                              set(h, 'Edgecolor', 'blue');
                              hold on; colormap('bone');
                              h2= warp(xi, yi,
plotat*ones(size(b_10)), imagematrix, [0 maximum_pixel_intensity]);
                              hold off;
                              set(gca, 'zlim', [plotat
max(max(b_10)')]);

                              zlabel('Relative Boron Dose');
                              xlabel(' mm');
                              ylabel(' mm');
                              title('INEEL factor adjusted boron
dose 256X256 ')
                              view(60, 10);

                              subplot(1,2,2)
                              plotat=floor(.5*((min(min(boron)')-
max(max(boron)')))));
                              h=mesh(x_xdim, y_ydim, boron);
                              set(h, 'Edgecolor', 'blue');
                              hold on; colormap('bone');
                              h2= warp(rangex, rangey,
plotat*ones(size(boron)), imagematrix2, [0
maximum_pixel_intensity]);
                              hold off;
                              set(gca, 'zlim', [plotat
max(max(boron)')]);
                              zlabel('Relative Boron Dose');
                              xlabel(' mm');
                              ylabel(' mm');
                              title('INELs unadjusted boron dose
xdim * ydim ')
                              view(60, 10);

                    end
                    %if (printopts(2) ==1)
                    %      print -dps -append adjust_pettopet_out

                    %end
                    if special_flag(hh) == 1

                              say = setstr(num2str(7+hhh*8));

                              say2 = strcat('fig_INELADJ3D_ ', say);
                              say2 = strcat(say2, '.bmp');
                              num = 7+hhh*8;
                              printstring = sprintf('print -
fFigure(%d) -dpgm %s', num, say2);
```

```
                        eval(printstring);

                end
        end


        if (plotsize(2) == 1)  %256X256 grid plot of difference over
tumor region only

                INELVSUT=zeros(256);
                INELVSUT=(adjusteddose2-b_10)./b_10.*100;
                [ nn, m]= find(imagematrix == max(max(imagematrix)));
% this will for maximum peak (i.e. tumor in image))
                        if size(nn, 1) >1 %if you find many maximums then
take the center point of the maximum
                                nn =nn(size(nn, 1));
                                m = m(size(nn, 1));
                        end


        nop = 80;

        tumor_boron=zeros(nop); xb=zeros(1, nop); yb=zeros(1,
nop);
        tumor_image=zeros(nop); xii=zeros(1, nop); yii=zeros(1,
nop);


        k = nn-nop/2; l = nn+nop/2+1;
        kk = m-nop/2; ll = m+nop/2+1;
                if k < 1
                        k = 1; l = 80;
                elseif kk < 1
                        kk = 1; ll = 80;
                end
                for i=k:l,
                        if ((i-(k)+1) > size(yb) | i > size(yy, 2))
                                break;
                        end
                        yb(i-(k)+1)=yy(i);
                        yii(i-(k)+1)=yi(i);
                        for j=kk:ll,

                                xb(j-(kk)+1)=xx(j);
                                xii(j-(kk)+1)=xi(j);
                                tumor_boron(i-(k)+1, j-
(kk)+1)=INELVSUT(i,j);
                                tumor_image(i-(k)+1, j-
(kk)+1)=imagematrix(i,j);
                        end
                end
        end


        if (plotsize(1) == 1)  % Xdim Ydim
```

103

```
            INELVSUT2=zeros(xdim, ydim);
            INELVSUT2=(ad_xdimydim-boron)./boron.*100;
            [ nn, m]= find(imagematrix2 == max(max(imagematrix2)));
% this will for maximum peak (i.e. tumor in image))
                if size(nn, 1) >1 %if you find many maximums then take
the center point of the maximum
                        nn =nn(size(nn, 1));
                        m = m(size(nn, 1));
                end

        nop2 = 10;

        tumor_boron2=zeros(nop2); xb2=zeros(1, nop2);
yb2=zeros(1, nop2);
        tumor_image2=zeros(nop2); xii2=zeros(1, nop2);
yii2=zeros(1, nop2);


        k = nn-nop2/2; l = nn+nop2/2;
        kk = m-nop2/2; ll = m+nop2/2;
            if k < 1
                    k = 1; l = 10;
            elseif kk < 1
                    kk = 1; ll = 10;
            end

            for i=k:l,
                    if ((i-(k)+1) > size(yb2) | i >
size(y_ydim, 2))
                            break;
                    end
                    yb2(i-(k)+1)=y_ydim(i);
                    yii2(i-(k)+1)=rangey(i);
                    for j=kk:ll,

                            xb2(j-(kk)+1)=x_xdim(j);
                            xii2(j-(kk)+1)=rangex(j);
                            tumor_boron2(i-(k)+1, j-
(kk)+1)=INELVSUT2(i,j);

                            tumor_image2(i-(k)+1, j-
(kk)+1)=imagematrix2(i,j);
                    end
            end
    end


    if FigureSelect(8)==1
            Figure(8+hhh*8)
                if (plotsize(2)==1 & plotsize(1) ~= 1)      %256X256
Grid
                    plotat=floor(1*((min(min(tumor_boron)')-
max(max(tumor_boron)'))));
                    I=tumor_image;
                    h=surf(xb, yb, tumor_boron);
                    set(h, 'Edgecolor', 'blue');
```

104

```
                    hold on; colormap('bone');
                    h2= warp(xii,yii, plotat*ones(size(I)), I,
[0 maximum_pixel_intensity]);
                    hold off;
                    set(gca, 'zlim', [plotat
max(max(tumor_boron)')]);
                    xlabel(' mm');
                    ylabel(' mm');
                        zlabel('(UTMCK-INEL)/(INEL)');
                      title(' % Diff. UTMCK DOSE ADJUSTMENT
using PET and INEL over tumor (256X256) ')
                        view(60, 10);

              elseif (plotsize(2) ~=1 & plotsize(1) == 1)
                    plotat=floor(1*((min(min(tumor_boron2)')-
max(max(tumor_boron2)')))));
                    I2=tumor_image2;
                    h=mesh(xb2, yb2, tumor_boron2);
                    set(h, 'Edgecolor', 'blue');
                    hold on; colormap('bone');
                    h2= warp(xii2,yii2, plotat*ones(size(xii,
2), size(yii,2))', I2, [0 maximum_pixel_intensity]);
                    hold off;
                    set(gca, 'zlim', [plotat
max(max(tumor_boron2)')]);
                    xlabel(' mm');
                    ylabel(' mm');
                        zlabel('(UTMCK-INEL)/(INEL)');
                      title(' % Diff. UTMCK DOSE ADJUSTMENT
using PET and INEL over tumor (40X40) ')
                        view(60, 10);

              elseif (plotsize(2) ==1 & plotsize(1) == 1)

                    subplot(1, 2, 1)


      plotat=floor(1*((min(min(tumor_boron)')-
max(max(tumor_boron)')))));
                    I=tumor_image;
                    h=mesh(xb, yb, tumor_boron);
                    set(h, 'Edgecolor', 'blue');
                    hold on; colormap('bone');
                    h2= warp(xii,yii, plotat*ones(size(xii, 2),
size(yii,2))', I, [0 maximum_pixel_intensity]);
                    hold off;
                    set(gca, 'zlim', [plotat
max(max(tumor_boron)')]);
                    xlabel(' mm');
                    ylabel(' mm');
                        zlabel('(UTMCK-INEL)/(INEL)');
                      title(' % Diff. UTMCK DOSE ADJUSTMENT
using PET and INEL over tumor (256X256) ')
                        view(60, 10);
```

```
                           subplot(1,2,2)

          plotat=floor(1*((min(min(tumor_boron2)')-
max(max(tumor_boron2)')))));
                              I2=tumor_image2;
                              h=mesh(xb2, yb2, tumor_boron2);
                              set(h, 'Edgecolor', 'blue');
                              hold on; colormap('bone');
                              h2= warp(xii2,yii2, plotat*ones(size(xii,
2), size(yii,2))', I2, [0 maximum_pixel_intensity]);
                              hold off;
                              set(gca, 'zlim', [plotat
max(max(tumor_boron2)')]);
                              xlabel(' mm');
                              ylabel(' mm');
                                  zlabel('(UTMCK-INEL)/(INEL)');
                                title(' % Diff. UTMCK DOSE ADJUSTMENT
using PET and INEL over tumor (40X40) ')
                                  view(60, 10);
                              end


                              % Figure(9); imshow(tumor_image, [0 180]);
hold on; colormap('bone'); contour(tumor_boron); hold on;

                              %if (printopts(2) ==1)
                              %       print -dps -append adjust_pettopet_out

                              %end
                      if special_flag(hh) == 1

                                  say = setstr(num2str(8+hhh*8));

                                  say2 = strcat('fig_INELVSUT3D_', say);
                                  say2 = strcat(say2, '.bmp');
                                  num = 8+hhh*8;
                                  printstring = sprintf('print -
fFigure(%d) -dpgm %s', num, say2);
                                  eval(printstring);


                          end

      end


end
```

Appendix B: QSH Format Description

Qsh is a medical image handling toolkit developed at New York University, Division of Nuclear Medicine. Qsh is a copyrighted, freely distributed set of programs and subroutines for image manipulation which use the American Association of Physicists in Medicine (AAPM) Image Format Standard for image storage and handling.

Features of QSH include the following:

- Qmenu, an example Motif interface to many of the QSH shell level programs. Qmenu is designed to manage the user interaction for some experiments involving viewing and registering nuclear medicine and CT or MRI images, and doing dosimetry calculations.
- For windowing systems such as X11, a shell program to create and manage a virtual frame buffer which other QSH programs will use for image and colormap display.
- Shell programs to add and extract key-value-pairs from the QSH header files, to find the maximum and minimum pixels in images and slices, to obtain pixel histograms and do gray-scale or equalized pixel remapping, to extract various subimages, to resize images, and to generate test images.
- Shell programs to clear the whole frame buffer or selected portions of it, to create color maps, to load and change the window and level of colormaps, to display a color bar across the bottom of the frame buffer, to "white-out" a range of colormap values, to display whole or partial 3D images as a series of slices with resizing and pixel remapping, to label the frame buffer, to read colormap values and pixel coordinates from the screen, and to save the contents of the frame buffer and the colormap.
- Shell programs to allow interactive tracing of regions of interest (ROIs), to resize and translate ROIs, to display ROIs, to generate and display masks generated from the area inside an ROI, and to display total activity counts, means, standard deviations, and other statistics from pixels contained in the intersection of a mask and an image.
- Shell programs to create single oblique slices or complete 3D oblique reprojections of images, and to interactively determine the "swing pixel" and planes or angles for these reprojections.
- Shell programs to interactively position 3D landmarks on slices of 3D images, to display and edit these landmarks, to do least-squares fits of landmarks from different images and generate 3D warping transform parameters, and to do warping transforms to bring dissimilar images into registration.
- Example shell scripts showing the ways in which the shell programs can be combined to automate image manipulation tasks and provide useful, quantifiable statistics from images and user interaction.

*see http://nucmed.nyu.edu/NucMed/qsh.html for more details.

Appendix C: Example Input File (rtt.input) From BNL

```
Treatment planing using MRI scan  of Aug. 14, 1998
        500       2000        .10000000E-01
        .12200000E+01       -.2540000E+01      -.2570000E+01
        .19900000E+02       .90000000E+02      .30000000E+02
../cg.geom
../../Images/tumor.rs
        30        3
        32       72       94
      -.15000000E+02      -.15000000E+02     -.15000000E+02
        .10000000E+01       .10000000E+01
final.rst
none
$RTT/beam/BPTB_D_12cmBS.source
        1        0
D                08/19/1998
$RTT/lib/rtt_008.mat
$RTT/lib/rtt_005.sigma
   521   508   513   .60142600E-07   .63150000E-01   .79110000E-03
rtt_MC.106

#iop
#   4   2   3
#   .1

# exclude      target

edit_dir
#patient      LB
#date         Aug 19, 1998
#beam         BNL_12cm
 mw_min       1.0
 b10_blood    1.0
 refvol       1.0
 ref_reg      brain target
# point        -3.00 1.00 -3.00
 ottocon      40 40 scalp
 delta        0.25
 eps          1.0
# dose- volume histogram for brain
 in_reg       brain target
 b10_blood    15.9
 rbe          1.3   1.0   3.2   3.2 1.0 1.0
 DVbs
# dose-volume histogram for tumor
 in_reg       tumor
 b10_blood    55.65
 rbe          3.8   1.0   3.2   3.2 1.0 1.0
 DVbs
# dose-volume histogram for target
```
108

```
in_reg        target tumor
b10_blood     55.65
rbe           3.8   1.0   3.2   3.2 1.0 1.0
DVbs
```

Appendix D:
PET Image Archiving for Treatment Planning

Note: when needing to retrieve PET data for BNCT treatment planning, please read the BPA_README file located in /pet/BPA/ directory on scanner for the study date of interest. For password access and other questions regarding the optical disk, please email Hedley Bond: bond@scanner.ipl.utmck.edu.

Once the optical disk is at hand and you have access to scanner, it is recommended that you change your working directory to /eod/. Within this directory, issue the following command to load the PET image from the optical disk onto the scanner so that it can be converted using **zoom_img** or some other image processing software.

```
scanner{mkhan}73: M

ARTECON OPTICAL DISK MENU:

1.  Mount Optical Disk (RW)
2.  Mount Optical Disk (READ ONLY)
3.  Mount Optical Disk (READ ONLY) and change directory to optical
disk
4.  Unmount Optical Disk
5.  Mount Optical Disk (RW) and export the disk to another system.
6.  Mount ALPHATRONIX Optical Disk (READ ONLY).
7.  Mount ALPHATRONIX Optical Disk (READ/WRITE).
8.  Unmount ALPHATRONIX Optical Disk.
9.  Mount ALPHATRONIX Optical Disk (READ ONLY) and
        export the disk to another system.
10. Exit this menu.

    Response ? (1-10) >  2
eod0: <Artecon EOD ISO - 512 cyl 37524 alt 2 hd 1 sec 31>

  *** Optical disk mounted READ ONLY. ***


    {** Enter M to display this menu again; bye to logout.**}

scanner{mkhan}74:   *** Optical disk mounted READ ONLY. ***


    {** Enter M to display this menu again; bye to logout.**}
```

Once the disk is mounted, it will place everything in /odsk/images/ directory. You should copy the images out of this directory and place it into some other working directory.

APPENDIX E:
zoom_img RAW Image Extraction Example

This example has been captured from the screen.

scanner{mkhan}60: zoom_img

Enter name of input file > gadd1110j1d_f24.img

Can't open file gadd1110j1d_f24.img

Reenter filename > gad1110j1d_f24.img

Enter name of output file > gadd1110j1d_f24_#.img

*** Zoom_img - Image Conversion and Zoom Utility ***

Select conversion type (1-8):

1. -   CTI to Analyze - enter frames, planes, flip, zoom
2. -   CTI to Analyze - last frame, no zoom.
3. -   CTI to BNCT - one image, zoom to 256 x 256.
4. -   CTI to BNCT - loop to convert > 1 image to > 1 file.
5. -   Analyze to CTI - no zoom, 1 frame; 16 bit CTI images.
6. -   Analyze to CTI - enter planes, flip, zoom; 16 bit CTI images
7. -   Analyze to CTI - no zoom, 1 frame, with 8 bit images.
8. -   Analyze to Analyze - zoom
9. -   I don't belong here - exit zoom_img now.

    Selection (1-9) [1] > 4

Output filenames will have slice# inserted in place of #.
DATA TYPE: 2
Input file (gad1110j1d_f24.img) - 1 frames, 47 planes.

Enter frame & planes to zoom (1 only).
Zoom which plane of frame 1? [23] > 1

Zoom image 1 in frame 1. OK? (y/n) [y] > y
Data for first image selected:
 Image size: 128 x 128   Pixel size: 2.109mm
 Slice width: 3.375mm
Continue? (y/n) [y] > y
Zooming 1 images (reading front to back) 1
 Image saved to gadd1110j1d_f24_01.img.
Convert more images? (y/n) [y] > y

DATA TYPE: 2
Input file (gad1110j1d_f24.img) - 1 frames, 47 planes.

111

Enter frame & planes to zoom (1 only).
Zoom which plane of frame 1? [23] > 2

Zoom image 2 in frame 1. OK? (y/n) [y] > y
Data for first image selected:
 Image size: 128 x 128   Pixel size: 2.109mm
 Slice width: 3.375mm
Continue? (y/n) [y] > y
Zooming 1 images (reading front to back) 1
 Image saved to gadd1110j1d_f24_02.img.
Convert more images? (y/n) [y] > y

DATA TYPE: 2
Input file (gad1110j1d_f24.img) - 1 frames, 47 planes.

Enter frame & planes to zoom (1 only).
Zoom which plane of frame 1? [23] > 3

Zoom image 3 in frame 1. OK? (y/n) [y] > y
Data for first image selected:
 Image size: 128 x 128   Pixel size: 2.109mm
 Slice width: 3.375mm
Continue? (y/n) [y] > y
Zooming 1 images (reading front to back) 1
 Image saved to gadd1110j1d_f24_03.img.
Convert more images? (y/n) [y] > n

Done!

APPENDIX F:

MRI2RAW.m: A Matlab™ Subroutine to Convert IDL formatted MRI images into RAW images needed for BNCT Treatment Planning.

Below is an example of an input file needed for this program.

```
                          MRICONVERT.in (R_in.m)
THIS IS AN INPUT FILE TO MRI2RAW.m: A code to read 8 bit raw MRI/PET
images and convert a selected image into a raw format.

HERE IS A LIST OF PET/MRI IMAGES THAT I WOULD LIKE TO CONVERT:
/ne5/bnct/God/MRI/10_15_97/
E5039S3I1.MR
E5039S3I2.MR
E5039S3I3.MR
E5039S3I4.MR
E5039S3I5.MR
E5039S3I6.MR
E5039S3I7.MR
E5039S3I8.MR
E5039S3I9.MR
E5039S3I10.MR
E5039S3I11.MR
E5039S3I12.MR
E5039S3I13.MR
E5039S3I14.MR
E5039S3I15.MR
E5039S3I16.MR
E5039S3I17.MR
E5039S3I18.MR
E5039S3I19.MR
E5039S3I20.MR
E5039S3I21.MR
```

Below is the program itself:

```
close all; clear all;
%                          MRI2RAW.m
%                    Code to Read MRI image & Convert to 8 bit
RAW format image for BNCT

sprintf('%s', '********************************')
sprintf('%s', 'This is written by Mohammad K. Khan')
sprintf('%s', '         Copyright    February, 1999    ')
sprintf('%s', '********************************')

clear all
```
113

```matlab
nn='MRI2RAW.in';
nn=input('Please enter the name of input file(MRI2RAW.in) ?', 's');

        inputfid = fopen(nn, 'r');
                %CHECK for the existence of file in the current
directory
                while (inputfid == -1)
                        sprintf('Attempt to find MRI2RAW.in in the current
directory failed... Please try again')
                        nn=input(' \n Enter the path to input (MRI2RAW.in)
file ? \n', 's');
                        inputfid = fopen(nn, 'r');
                end

        hh = 0;    %this is a counter that will count how many image
files to read
                % and get pixel size
        % Read the contour file names from the input file
        global pixelsize plotsize FigureSelect INELfactor
        n=fgetl(inputfid);
        nt=fgetl(inputfid);
        nt=fgetl(inputfid);
        nt =fgetl(inputfid);
        nt =fgetl(inputfid);
        nt = fgetl(inputfid);
D = pwd;
nt=deblank(nt);
cd(nt);

while(feof(inputfid) ~= 1)
        hh= hh+1;
        n1=fgetl(inputfid);
                if n1 == -1,
                        break;
                end
        nt=deblank(n1);   % Gets the image file names from the input
file
        imagefiles(hh) = {nt};     %names of all the mri/pet image
files
        imagefid=fopen(nt, 'r');
        %sprintf('Working on %s file... Please be patient! ..', nt)

        while (imagefid == -1)
                sprintf('Attempt to find %s failed... Please try again',
nt)
                nt=input(' \n please enter the path to that file ? \n',
's');
                imagefid = fopen(nt, 'r');
        end

        say = fread(imagefid, 'int16');

        k=0;
        for i = 1:256,
                for j = 1:256,
```

```
                    k=k+1;
                    A(i,j) = say(3952+k);   % Header Lines NOT NEEDED

            end
        end
        imagematrix(:,1+(hh-1)*256:256*hh)=A;
        fclose(imagefid);
end

cd(D);
grayimages= ind2gray(imagematrix, gray(256));
        if min(min(imagematrix)') <0;
                grayimages = ind2gray(imagematrix/2, gray(256));
        end

Figure(1)
montage(grayimages, [256 256 hh]);
sprintf('Image planes are number 1..2..3..4 .. etc right to left and
into next row')
number = input('please select an image plane to convert from MRI to
BMP 8 BIT ? ');

IM = grayimages(:, 1+(number-1)*256:256*number);
IM = flipud(IM);
nt = imagefiles(number);
nt2 = strcat(nt, '.img');
nt = strcat(nt, '.bmp');
output = fopen(char(nt2), 'wb');
imwrite(IM*255, gray(255), char(nt), 'bmp');
fwrite(output, IM'*255, 'int8');   % Convert to Raw 8 bit image


Figure(2);
imshow(IM);
cd(D);
multiple= input('Would you like to convert other images Planes
(n/y)?','s');
while(multiple == 'y' | 'Y')
        number = input('\n please select an image plane to convert
from RAW to Bitmap ? ');
        IM = grayimages(:, 1+(number-1)*256:256*number);
        IM = flipud(IM);
        nt = imagefiles(number);
        nt2 = strcat(nt, '.img');
        nt = strcat(nt, '.bmp');
        output = fopen(char(nt2), 'wb');
        imwrite(IM*255, gray(255), char(nt), 'bmp');
        fwrite(output, IM'*255, 'int8');   % Convert to Raw 8 bit image

        Figure(2);
        imshow(IM);
        cd(D);
        multiple= input('\n Would you like to continue converting
other images Planes (n/y)?','s')
                if ((multiple == 'n') | (multiple == 'N'))
```

```
            break;
        end
end
```

VITA

Mohammad Khurram Khan was born in Lahore, Pakistan on May 9, 1975. He was seven years of age when he moved from Pakistan and attended the public school system in Brooklyn, N.Y. Since then he moved around several times during elementary school, middle school, and high school. He graduated from Maryville High School in May, 1993. He then entered the nuclear engineering program at The University of Tennessee, Knoxville where he graduated Summa Cum Laude. He received his Bachelor's of Science degree in Nuclear Engineering in May of 1997. After his bachelor's degree, he decided to work on his Masters Degree in Nuclear Engineering at The University of Tennessee, Knoxville. His Masters Degree was received in May of 1999.

Currently he has plans to pursue medical school after his Masters Degree. Between now and medical school, he plans to continue doing research work towards his Ph.D. at The University of Tennessee, Knoxville.