

# Development and Application of a Random Lung Model for Dose Calculations in Radiotherapy

by  
Liang Liang

A dissertation submitted in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
(Nuclear Engineering and Radiological Sciences)  
in The University of Michigan  
2007

Doctoral Committee:

Professor Edward W. Larsen, Chair  
Professor William R. Martin  
Associate Professor James M. Balter  
Associate Professor Indrin J. Chetty, University of Nebraska

© Liang Liang 2007  
All Rights Reserved

To my family, and in particular, to my wife, Liyuan

## ACKNOWLEDGEMENTS

At this point, I don't know if there is indeed a perfect way to say my endless thanks to the people who have helped me along this long journey. My words will always look pale before what you have given me.

The very first person I want to thank, from the bottom of my heart, is my advisor, Dr. Ed Larsen. During my time at the University of Michigan, you have showed *infinite* patience, in every aspect I can imagine, in helping me get jobs done. Your clear thoughts in tackling problems amazed me every time I talked with you. Your passion in teaching and research has inspired, and will always inspire me.

I like to take this opportunity to thank my other two committee members from the Department of Radiation Oncology: Dr. James Balter, and Dr. Indrin Chetty. My pursuit in medical physics would never come true if I had not got the fruitful guidance from both of you. James, you may never know how important the first medical physics-related question you asked me: "What's the missing tissue?", has been in triggering my deep interest. And Indrin, how can I evaluate the frequent encouragement from you, especially when I doubt myself?

I would like to thank Dr. Bill Martin, one of my committee member, for the essential Monte Carlo skills I have learned from you, either in you class, or through the interesting discussions with you. Not to mention the care from you as the chair of the NERS department (what a cozy group I live in).

I am grateful to Dr. Alex Bielajew. Without your recommendation of the proper

tools, and your expertise in my research work, I would not have been able to write down these words.

My thanks go to my friends in NERS: you have made my life in the US so colorful. Thank you, Neelam, Mihaela, Dan, Yan, Haori, Troy, Hao, Weiji, . . .

I specially thank my long-term friends in China: Chongbin, Wei, Xiao, Haijin, Jinyong, Zhongxue, Guogang, Qiang, . . . I never lack your unselfish support from the other semi-sphere.

To conclude this “chapter”, my family, in particular, my gorgeous and unique wife, Liyuan, and my gorgeous and unique daughter, Catherine, please accept my deepest appreciations. Your unconditional support has been, and will always be with me on my road, wide or narrow, smooth or rough.

# TABLE OF CONTENTS

DEDICATION . . . . .	ii
ACKNOWLEDGEMENTS . . . . .	iii
LIST OF FIGURES . . . . .	vii
LIST OF TABLES . . . . .	xii
ABSTRACT . . . . .	xiii
CHAPTER	
<b>I. Introduction . . . . .</b>	<b>1</b>
1.1 The Lung Models for Heterogeneity Correction . . . . .	11
1.1.1 The Lung's Anatomy . . . . .	12
1.1.2 The Atomic Mix Model . . . . .	13
1.1.3 The CT Model . . . . .	16
1.1.4 The Validity of the Atomic Mix Approximation . . . . .	18
1.2 The Goals of This Thesis . . . . .	19
1.3 Organization of this thesis . . . . .	21
<b>II. The Lung's Anatomy . . . . .</b>	<b>23</b>
2.1 The Lung's Morphology . . . . .	23
2.1.1 The Horsfield Orders . . . . .	24
2.1.2 The Morphometric Data . . . . .	25
2.2 Other Relevant Parameters . . . . .	30
2.2.1 Lung Density . . . . .	30
2.2.2 $T/D$ Ratio . . . . .	30
2.2.3 Bronchoarterial Ratio . . . . .	30
<b>III. The Atomic Mix Approximation for Charged Particle Transport . . . . .</b>	<b>31</b>
3.1 Introduction . . . . .	31
3.2 The 1-D Linear Boltzmann Equation . . . . .	32
3.3 The Fokker-Planck Approximation to $L_r$ . . . . .	35
3.4 The Atomic Mix Approximation to the BFP Equation . . . . .	40
3.5 Numerical Results and Discussion . . . . .	46
3.5.1 "Droplet" model and Monte Carlo simulation . . . . .	46
3.5.2 Dose Distribution for Electrons and Photons . . . . .	47
3.6 Conclusion . . . . .	52
<b>IV. A 2 1/2-D Random Lung Model . . . . .</b>	<b>54</b>

4.1	Introduction . . . . .	54
4.2	The Lung Model . . . . .	56
4.2.1	“Random” 2 1/2-D geometry . . . . .	56
4.2.2	The Threshold Size . . . . .	62
4.3	Monte Carlo Simulations . . . . .	64
4.4	Results and Discussion . . . . .	69
4.4.1	Electron Cutoff Energy . . . . .	70
4.4.2	Threshold Selection . . . . .	71
4.4.3	CAX Depth Dose . . . . .	76
4.4.4	Isodose Lines and Profiles . . . . .	81
4.4.5	Mean lung doses . . . . .	85
4.4.6	Detailed vs. voxelized lungs . . . . .	87
4.5	Conclusion . . . . .	88
<b>V.</b>	<b>CT Resolution for Lung Treatment Planning: An Application of the Random Lung Model . . . . .</b>	<b>90</b>
5.1	Introduction . . . . .	90
5.2	A Three-beam Treatment Planning Using the Random Lung Model . . . . .	92
5.2.1	The Lung Phantom . . . . .	92
5.2.2	Photon Beams . . . . .	95
5.2.3	Monte Carlo Simulations . . . . .	100
5.2.4	Treatment Plan Evaluation Metrics . . . . .	100
5.2.5	Reference CT Resolution for the Lung Phantom . . . . .	102
5.3	Results and Discussion . . . . .	102
5.3.1	Reference CT Resolution for the Lung Phantom . . . . .	102
5.3.2	Isodose lines . . . . .	104
5.3.3	DVHs . . . . .	114
5.3.4	Mean Lung Dose . . . . .	118
5.4	Conclusion . . . . .	119
<b>VI.</b>	<b>Conclusions . . . . .</b>	<b>121</b>
	<b>APPENDICES . . . . .</b>	<b>130</b>
	<b>BIBLIOGRAPHY . . . . .</b>	<b>231</b>

## LIST OF FIGURES

<u>Figure</u>		
1.1	Example multi-beam treatment planning setup, extracted from [2]. . . . .	3
1.2	Example isodose lines, extracted from [2]. . . . .	10
1.3	Example DVHs, extracted from [2]. . . . .	10
1.4	Correction factors for: (a), correction factors for three different field sizes for a 4 MV photon beam; (b), correction factors for three different lung densities for a 15 MV photon beam of $5 \times 5 \text{ cm}^2$ field size; (c), correction factors for three different tumor widths for a 15 MV photon beam of $5 \times 5 \text{ cm}^2$ field size. All figures are extracted from [97]. . . . .	14
2.1	(a) Horsfield orders; (b) Weibel’s generations . . . . .	25
2.2	Lung’s airways . . . . .	27
3.1	<i>R-Z</i> plane view of the “droplet” model. The dark cells are water and the blank cells are air. The droplet cell size $ds = 0.1 \text{ cm}$ in this figure. . . . .	46
3.2	The relevant length scales in the electron transport within a range of energy of interest for radiotherapy: (i) transport MFP, the dash line; (ii) CSDA range, the dotted line; (iii) hard MFP, the dash-dotted line; and (iv) MFP, the solid line. . . .	48
3.3	Dose contours for 2 MeV electron beam: (a) is for the atomic mix case. The other three are for the droplet model with a cell size at (b) 0.1 cm, (c) 0.05 cm, and (d) 0.01 cm, respectively. . . . .	50
3.4	Dose contours for 3.4 MeV photon beam: (a) is for the atomic mix case. The other three are for the droplet model with a cell size at (b) 0.1 cm, (c) 0.05 cm, and (d) 0.01 cm, respectively. . . . .	51
4.1	Three-dimensional view of the simulated phantom geometry and corresponding dimensions (cm) with the 2 1/2-D lung model embedded inside the surrounding water. The coordinate system is shown in the upper left corner; the photon beams are incident in the positive <i>y</i> direction and perpendicular to the <i>x-z</i> plane; the upper legend illustrates modeled airways (concentric cylinders), arteries (single cylinders attached to the airways in the lung) and veins (independent single cylinders in the lung) inside the lung. . . . .	57



4.2	Top view of the random heterogeneous lung model in which the concentric circles represent bronchi, the single circles attached to the bronchi are arteries and the independent single circles are veins: (a–b) two realizations of the modeled “whole” lung, with four selected $10.2 \times 10 \text{ cm}^2$ regions to fit in the lung block in the simulated phantom (see also Fig. 4.1) respectively. (This figure is continued on Page 60) . . .	59
4.2	(continued from Fig. 4.2 on Page 59) (c) realization $L_1$ , indicated in (a) by the lower right framed region, with one large bronchus, artery and vein close to the CAX. (Lines (i) and (ii) are used in Fig. 4.3); (d) realization $L_2$ , indicated in (b) by the right framed region, with two large bronchi, arteries and veins off the CAX; (e) realization $S_1$ , indicated in (a) by the upper left framed region, with no large structures; (f) realization $S_2$ , indicated in (b) by the lower left framed region, with no large structures. . . . .	60
4.3	Line density change in $y$ -direction in the lung. The thick lines depict the uniform mean density (MD) case. The thin lines are for the heterogeneous realization $L_1$ in Fig. 4.2c: (a) corresponds to dashed line (i) and (b) corresponds to dashed line (ii).	62
4.4	Top view of a voxelized version of realization $L_1$ as shown in Fig. 4.2c to mimic the CT-scan. The resolution is set to be $0.4 \times 0.4 \times 0.4 \text{ cm}^3$ . . . . .	66
4.5	Top view of the random heterogeneous lung model used for threshold selection: (a) and (b) are a large and a small realization with a threshold level at the Horsfield order 6, 5 order higher than the terminal bronchiole. (This figure is continued on page 68) . . . . .	67
4.5	(continued from page 67.) (c) and (d) are a large and a small realization with a threshold level at the Horsfield order 1, i.e., the terminal bronchiole. Figs. (e) and (f) are a large and a small realization with a threshold level at the Horsfield order -3, the last respiratory bronchiole level. . . . .	68
4.6	The CAX depth dose for (a): the atomic mix model, and (b): the “droplet” model	70
4.7	The percent difference between the two $E_{cut}$ 's. It is calculated by normalizing the absolute dose difference by the $D_{max}$ of the 100 keV cutoff case, for (a): the atomic mix, and (b): the “droplet” realization. . . . .	71
4.8	The normalized CAX PDD for the large and the small realizations for three threshold levels: the terminal bronchiole (TB, the solid lines), the small bronchiole at a level of 5 orders more than the TB (TB+5, the dash lines) and the respiratory bronchiole at a level of 3 orders less than the TB (TB-3, the dotted lines). (a) is for the large realizations and (b) is for the small ones. In both figures the solid lines and the dotted lines are essentially on top of each other. . . . .	72
4.9	Isodose lines for large realizations: (a) represents the TB+5 level, (b) the TB level and (c) the TB-3 level. . . . .	73
4.10	Isodose lines for small realizations: (a) represents the TB+5 level, (b) the TB level and (c) the TB-3 level. . . . .	74
4.11	The CAX percent depth doses for: $1 \times 1 \text{ cm}^2$ field size and for (a) 18 MV, (b) 6 MV beams; $5 \times 5 \text{ cm}^2$ field size and for (c) 18 MV, (d) 6 MV beams. (to be continued on page 78) . . . . .	77

4.11	(cont'd from page 77.) The CAX percent depth doses for: $10 \times 10$ cm <sup>2</sup> field size and for (e) 18 MV, (f) 6 MV beams; $20 \times 20$ cm <sup>2</sup> field size and for (g) 18 MV, (h) 6 MV beams. The thick solid lines are for the mean density (MD) lung model. The thin solid and the dashed lines are for two large realizations, respectively. The dash-dot and the dotted lines are for two small realizations. All curves for the same field size are normalized to the $D_{max}$ along the CAX of the corresponding mean density (MD) case. Also indicated in (a) and (b) are the large structure locations (an airway, an artery and a vein) on the CAX for realization $L_1$ . . . . .	78
4.12	Isodose lines on $x$ - $y$ plane ( $z = 0$ ) for $1 \times 1$ cm <sup>2</sup> field size for (a) mean density model, 6 MV; (b) realization $L_1$ , 6 MV. The abscissa is in $y$ -direction and the ordinate is in $x$ -direction. . . . .	82
4.13	CAX dose profiles for $1 \times 1$ cm <sup>2</sup> field size for 6 MV at (a) $y=6.1$ cm; (b) $y=10.9$ cm. The thick solid lines are for the mean density (MD) lung model. The thin solid and the dashed lines are for two large realizations, respectively. The dash-dot and the dotted lines are for two small realizations. All curves for the same field size are normalized to the $D_{max}$ along the CAX of the corresponding mean density (MD) case. . . . .	83
4.14	CAX dose profiles for $10 \times 10$ cm <sup>2</sup> field size for 6 MV at (a) $y=9.1$ cm (6.1 cm deep in the lung); (b) $y=10.5$ cm (7.5 cm deep in the lung). The thick solid lines are for the mean density (MD) lung model. The dashed lines are for large realization $L_2$ . All curves for the same field size are normalized to the $D_{max}$ along the CAX of the corresponding mean density (MD) case. . . . .	86
4.15	The CAX percent depth doses for a $1 \times 1$ cm <sup>2</sup> field size and 6 MV beam. The solid line is for the detailed realization $L_1$ (Fig. 4.2c) and the dash line is for its voxelized version (Fig. 4.4), respectively. Both curves are normalized to the $D_{max}$ along the CAX of the detailed $L_1$ case. . . . .	88
5.1	The lung phantom (the outer clear box) for three-beam (one right lateral (RL) beam and two opposing AP-PA beams) treatment planning. The phantom is $24 \times 30 \times 36$ cm <sup>3</sup> with two lungs (light gray box, right and left) of the same size of $16 \times 8 \times 28$ cm <sup>3</sup> embedded inside. The water layers surrounding the lung have uniform thickness of 4 cm while the lateral distance between the two lungs is 3 cm. The cubical tumor (dark gray) of various sizes resides in the middle of the right lung. . . . .	93
5.2	Top view of the lung phantom (Fig. 5.1) with a realization of the random lung model in the right lung across the isocenter. Three beams (RL, AP and PA) are indicated. A tumor is represented by the black square in the middle of the right lung. . . . .	94
5.3	Large realizations for simulations with a tumor size of $1 \times 1 \times 1$ cm <sup>3</sup> : (a), $T1L01$ , at a CT resolution of $1 \times 1$ mm <sup>2</sup> ; (b), $T1L02$ , at $2 \times 2$ mm <sup>2</sup> ; (c), $T1L04$ , at $4 \times 4$ mm <sup>2</sup> . . . . .	96
5.4	Small realizations for simulations with a tumor size of $1 \times 1 \times 1$ cm <sup>3</sup> : (a), $T1S01$ , at a CT resolution of $1 \times 1$ mm <sup>2</sup> ; (b), $T1S02$ , at $2 \times 2$ mm <sup>2</sup> ; (c), $T1S04$ , at $4 \times 4$ mm <sup>2</sup> . . . . .	97
5.5	Large realizations for simulations with a tumor size of $4 \times 4 \times 4$ cm <sup>3</sup> : (a), $T4L01$ , at a CT resolution of $1 \times 1$ mm <sup>2</sup> ; (b), $T4L02$ , at $2 \times 2$ mm <sup>2</sup> ; (c), $T4L04$ , at $4 \times 4$ mm <sup>2</sup> . . . . .	98

5.6	Small realizations for simulations with a tumor size of $4 \times 4 \times 4 \text{ cm}^3$ : (a), <i>T4S01</i> , at a CT resolution of $1 \times 1 \text{ mm}^2$ ; (b), <i>T4S02</i> , at $2 \times 2 \text{ mm}^2$ ; (c), <i>T4S04</i> , at $4 \times 4 \text{ mm}^2$ . . . . .	99
5.7	CAX depth dose curves along (a): <i>x</i> -direction and (b): <i>y</i> -direction, of a three-beam simulation in a phantom depicted in Fig. 5.3a. The dotted lines are calculated with DPM while the solid lines are calculated with PENELOPE. . . . .	103
5.8	Isodose lines at <i>x-y</i> plane across the isocenter for simulations for the large realizations with a tumor size of $1 \times 1 \text{ cm}^3$ at various CT resolutions. Only the lung region is shown. The tumor is indicated by the shaded gray square in the lung center. The isodose levels are 20, 50, 70, 90, 95 and 100% of the dose at the isocenter ( $D_{iso}$ ) for the realizations at the reference CT resolution of $1 \times 1 \text{ mm}^2$ . Realizations: (a), <i>T1L01</i> , at a CT resolution of $1 \times 1 \text{ mm}^2$ ; (b), <i>T1L02</i> , at $2 \times 2 \text{ mm}^2$ ; (c), <i>T1L04</i> , at $4 \times 4 \text{ mm}^2$ . . . . .	105
5.9	Isodose lines at <i>x-y</i> plane across the isocenter for simulations for the small realizations with a tumor size of $1 \times 1 \text{ cm}^3$ at various CT resolutions. Only the lung region is shown. The tumor is indicated by the shaded gray square in the lung center. The isodose levels are 20, 50, 70, 90, 95 and 100% of the dose at the isocenter ( $D_{iso}$ ) for the realizations at the reference CT resolution of $1 \times 1 \text{ mm}^2$ . Realizations: (a), <i>T1S01</i> , at a CT resolution of $1 \times 1 \text{ mm}^2$ ; (b), <i>T1S02</i> , at $2 \times 2 \text{ mm}^2$ ; (c), <i>T1S04</i> , at $4 \times 4 \text{ mm}^2$ . . . . .	106
5.10	Isodose lines at <i>x-y</i> plane across the isocenter for simulations for the large realizations with a tumor size of $4 \times 4 \times 4 \text{ cm}^3$ at various CT resolutions. Only the lung region is shown. The tumor is indicated by the shaded gray square in the lung center. The isodose levels are 20, 50, 70, 90, 95 and 100% of the dose at the isocenter ( $D_{iso}$ ) for the realizations at the reference CT resolution of $1 \times 1 \text{ mm}^2$ . Realizations: (a), <i>T4L01</i> , at a CT resolution of $1 \times 1 \text{ mm}^2$ ; (b), <i>T4L02</i> , at $2 \times 2 \text{ mm}^2$ ; (c), <i>T4L04</i> , at $4 \times 4 \text{ mm}^2$ . . . . .	107
5.11	Isodose lines at <i>x-y</i> plane across the isocenter for simulations for the small realizations with a tumor size of $4 \times 4 \times 4 \text{ cm}^3$ at various CT resolutions. Only the lung region is shown. The tumor is indicated by the shaded gray square in the lung center. The isodose levels are 20, 50, 70, 90, 95 and 100% of the dose at the isocenter ( $D_{iso}$ ) for the realizations at the reference CT resolution of $1 \times 1 \text{ mm}^2$ . Realizations: (a), <i>T4S01</i> , at a CT resolution of $1 \times 1 \text{ mm}^2$ ; (b), <i>T4S02</i> , at $2 \times 2 \text{ mm}^2$ ; (c), <i>T4S04</i> , at $4 \times 4 \text{ mm}^2$ . . . . .	108
5.12	Isodose lines near the tumor at the <i>x-y</i> plane across the isocenter for simulations for the large realizations with a tumor size of $1 \times 1 \times 1 \text{ cm}^3$ at various CT resolutions. The tumor is indicated by the shaded gray square in the lung center. The isodose levels are expressed as the percentage of the dose at the isocenter ( $D_{iso}$ ) for the realizations at the reference $1 \times 1 \text{ mm}^2$ CT resolution. Realizations: (a), <i>T1L01</i> , at a CT resolution of $1 \times 1 \text{ mm}^2$ ; (b), <i>T1L02</i> , at $2 \times 2 \text{ mm}^2$ ; (c), <i>T1L04</i> , at $4 \times 4 \text{ mm}^2$ . . . . .	110
5.13	Isodose lines near the tumor at the <i>x-y</i> plane across the isocenter for simulations for the small realizations with a tumor size of $1 \times 1 \times 1 \text{ cm}^3$ at various CT resolutions. The tumor is indicated by the shaded gray square in the lung center. The isodose levels are expressed as the percentage of the dose at the isocenter ( $D_{iso}$ ) for the realizations at the reference $1 \times 1 \text{ mm}^2$ CT resolution. Realizations: (a), <i>T1S01</i> , at a CT resolution of $1 \times 1 \text{ mm}^2$ ; (b), <i>T1S02</i> , at $2 \times 2 \text{ mm}^2$ ; (c), <i>T1S04</i> , at $4 \times 4 \text{ mm}^2$ . . . . .	111

- 5.14 Percent depth dose curves along the  $x$ -direction at different  $y$ -values on the  $x$ - $y$  isocenter plane. (a) and (b): large and small realizations with a tumor size of  $1 \times 1 \times 1 \text{ cm}^3$ , at  $y = 7.7$  and  $8.0 \text{ cm}$ , respectively; (c) and (d): large and small realizations with a tumor size of  $4 \times 4 \times 4 \text{ cm}^3$ , at  $y = 7.5$  and  $8.0 \text{ cm}$ , respectively. In all figures, the solid lines are for a CT resolution of  $1 \times 1 \text{ mm}^2$ , the dashed lines  $2 \times 2 \text{ mm}^2$  and the dotted lines  $4 \times 4 \text{ mm}^2$  . . . . . 113
- 5.15 DVHs for the tumor and the right lung for all simulations. The dose is expressed as a percentage of  $D_{iso}$  of the cases at a CT resolution of  $1 \times 1 \text{ mm}^2$ . (a) and (b): large and small realizations with a tumor size of  $1 \times 1 \times 1 \text{ cm}^3$ , respectively; (c) and (d): large and small realizations with a tumor size of  $4 \times 4 \times 4 \text{ cm}^3$ , respectively. In all figures, the solid lines are for a CT resolution of  $1 \times 1 \text{ mm}^2$ , the dashed lines are for  $2 \times 2 \text{ mm}^2$  and the dotted lines are for  $4 \times 4 \text{ mm}^2$ . The differences between the lines for different resolutions are small so that most of them overlap each other. 115

## LIST OF TABLES

**Table**

2.1	The conducting airways and the accompanying vessels' morphometry [53] . . . . .	28
2.2	The acinus morphometry [105] . . . . .	29
4.1	Background density . . . . .	76
4.2	Mean lung dose for each field size ( $\text{cm}^2$ ) and beam energy for the mean density lung model (MD) and one of the four random heterogeneous realizations as a percentage normalized to the $D_{max}$ (MeV/g) along the central axis of the corresponding mean density lung case. (The mean lung density ( $\text{g}/\text{cm}^3$ ) for the corresponding case is also listed in the parentheses.) . . . . .	87
5.1	Relative absolute differential dose ( $\text{ADD}_{rel}$ ) between larger CT resolutions and the reference $1 \times 1 \text{ mm}^2$ CT resolution for simulations with tumor size of $1 \times 1 \times 1 \text{ cm}^3$ . The relative absolute differential dose is defined by Eq. 5.2, and is expressed as a percentage. . . . .	116
5.2	Relative absolute differential dose ( $\text{ADD}_{rel}$ ) between larger CT resolutions and the reference $1 \times 1 \text{ mm}^2$ CT resolution for simulations with tumor size of $4 \times 4 \times 4 \text{ cm}^3$ . The relative absolute differential dose is defined by Eq. 5.2, and is expressed as a percentage. . . . .	117
5.3	Mean lung dose (MLD) and mean tumor dose (MTD) for simulations with tumor size of $1 \times 1 \times 1 \text{ cm}^3$ . Also included are the absolute dose at the isocenter $D_{iso}$ . The MLD and MTD are expressed as a percentage of $D_{iso}$ of the cases at a CT resolution of $1 \times 1 \text{ mm}^2$ . . . . .	118
5.4	Mean lung dose (MLD) and mean tumor dose (MTD) for simulations with tumor size of $4 \times 4 \times 4 \text{ cm}^3$ . Also included are the absolute dose at the isocenter $D_{iso}$ . The MLD and MTD are expressed as a percentage of $D_{iso}$ . . . . .	118

## ABSTRACT

Radiotherapy requires accurate dose calculations in the human body, especially in disease sites with large variations of electron density in neighboring tissues, such as the lung. Currently, the lung is modeled by a voxelized geometry interpolated from computed tomography (CT) scans to various resolutions. The simplest such voxelized lung, the atomic mix model, is a homogenized whole lung with a volume-averaged bulk density. However, according traditional transport theory, even the relatively fine CT voxelization of the lung is not valid, due to the extremely small mean free path (MFP) of the electrons.

The purpose of this thesis is to study the impact of the lung's heterogeneities on dose calculations in lung treatment planning. We first extend the traditional atomic mix theory for charged particles by approximating the Boltzmann equation for electrons to its Fokker-Planck (FP) limit, and then applying a formal asymptotic analysis to the BFP equation. This analysis raises the length scale for homogenizing a heterogeneous medium from the electron mean free path (MFP) to the much larger electron transport MFP. Then, using the lung's anatomical data and our new atomic mix theory, we build a realistic  $2\frac{1}{2}$ -D random lung model. The dose distributions for representative realizations of the random lung model are compared to those from the atomic mix approximation of the random lung model, showing that significant perturbations may occur with small field sizes and large lung structures. We also apply our random lung model to a more realistic lung phantom and investigate the

effect of CT resolutions on lung treatment planning. We show that, compared to the reference  $1 \times 1 \text{ mm}^2$  CT resolution, a  $2 \times 2 \text{ mm}^2$  CT resolution is sufficient to voxelize the lung, while significant deviations in dose can be observed with a larger  $4 \times 4 \text{ mm}^2$  CT resolution. We use the Monte Carlo method extensively in this thesis, to avoid systematic errors caused by inaccurate heterogeneity corrections that occur in approximate clinical dose calculation methods.

Finally, we address potential improvements for our random lung model and some possible future applications.

## CHAPTER I

### Introduction

Cancer counts as the second largest cause of death in the US, following only heart disease. The US 2007 statistics by the American Cancer Society [111] indicates that (i) the death rates of lung cancer are the largest in both men and women, with a 5-year survival rate of only 16%, and (ii) in 2007, about 29% of all cancer deaths will come from lung cancer.

Radiation therapy (radiotherapy) uses beams of energetic particles (photons, electrons, protons, etc.) to kill the tumor cells by depositing energy from charged particles, which are either present directly in the primary beam (such as an electron beam), or are emitted as secondary particles through the interactions of the primary beam (such as a photon beam) with matter. For lung cancers, the most often-used particle beams are external photon beams, which come from a treatment machine outside the patient body, as compared with the brachytherapy, which implants radioactive seeds in patient's body. Tyldesley *et al.* [117] estimated that 61% of patients with lung cancer will need radiotherapy at some point in their illness.

The goal of radiotherapy [42], from the earliest time to the modern era with its sophisticated treatment technology, has always been to tailor a high dose to the tumor, while sparing as much normal tissue as possible. This goal is simple to state,



yet “so little practical radiotherapy completely achieves this goal [123].” To achieve the goal, each link in the “radiotherapy chain [123]” (the calibration of the treatment machine, the acquisition of patient’s data, the dose calculation, the dose delivery, the final quality assurance (QA), etc.) must be carefully and optimally performed. Each of the “links” in this chain has been extensively studied and has painstakingly evolved to its current status [42].

The goal of this thesis is to examine the *dose calculation* link in this “chain,” as it currently exists for lung cancers. The special difficulty associated with estimating dose in the lung is the extraordinarily complex physical structure of the lung, which is only crudely approximated in current dose calculation algorithms. In this thesis, we develop an accurate computer model of the human lung, and use this model to assess current computational methods for estimating dose in the lung. To explain this more fully, we must first discuss in more detail the procedures for radiotherapy, and the physical structure of the lung. We turn to these issues next.

When a patient is diagnosed of cancer, the treatment planning process [45, 46], which generates a patient-specific radiotherapy plan before the the actual dose delivery, is initiated. Treatment planning includes the following major steps:

1. Acquire the disease-related anatomy of the patient, mainly by (but not limited to) computed tomography (CT) scans, in the treatment position.
2. Delineate the external patient contour, the target (the tumor), the organs at risk (OARs), and the normal tissues on the patient’s anatomy scans.
3. Prescribe the beam type, and a dose to the target along with the dose tolerance to the normal tissues and the OARs.
4. Determine the desired number, orientations, and shapes of the beams with the

aid of modern 3-D visual tools (e.g. the beam's-eye-view [123]). Usually, a multi-beam scheme is used because the dose is continuously deposited along the beam's pathway. Therefore, the deposition of dose in the normal tissue is unavoidable, so to minimize this, multiple beams from different angles are used to treat the target at the intersection of these beams.

5. Calculate the dose distribution from the chosen beam setup, using the algorithm available with the treatment planning system (TPS) and the geometry information from the patient's CT scans as input. Then perform optimizations, based on a preset objective function, which is used to evaluate the quality of a particular plan, to obtain the final treatment plan for the patient.

Fig. 1.1 [2] gives an example of a modern multi-beam treatment planning setup.

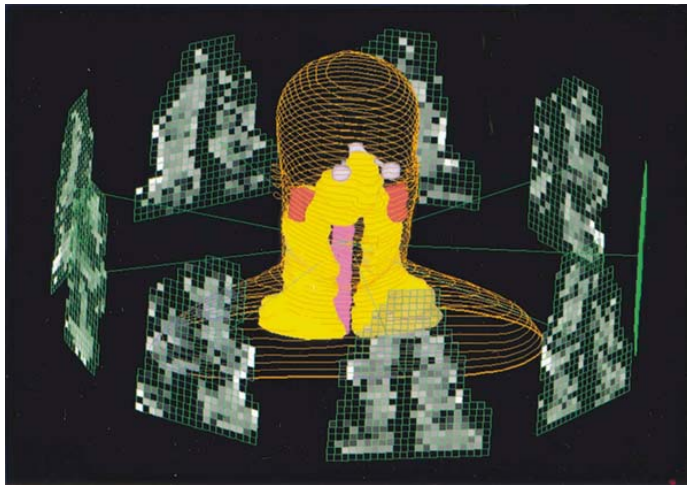


Figure 1.1: Example multi-beam treatment planning setup, extracted from [2].

The accuracy of dose calculations depends on many factors. Two key ones are (i) an accurate description of the patient's geometry, and (ii) an accurate dose calculation algorithm. This is particularly important with cancer sites having many heterogeneities, such as the lung. After many years of clinical use of the *homogeneous* assumption, in which the human body is regarded as a block of uniform

water-equivalent tissue at unit density, the *heterogeneous* correction became possible following the advent of CT. This was because CT scans are capable of describing the patient's anatomy at each location, and with its particular material composition.

To generate a treatment plan for a lung cancer patient, the anatomical information of the patient's thorax is acquired, mainly by CT scans, and is complimented by other imaging modalities, such as magnetic resonance imaging (MRI), positron emission tomography (PET), and single photon emission computed tomography (SPECT), if necessary. Then a matrix of uniform geometrical voxels, which represents the patient's thorax, is interpolated from the CT scans at various resolutions. Each geometrical voxel has a unique electron density, which is obtained from a calibrated CT number-electron density relationship curve, and a type of material which has a density range that brackets the voxel's density. Therefore, the finer the CT resolution is, and the more accurate the conversion from CT numbers to electron density is, the better this geometrical matrix will represent the heterogeneities inside the lung.

The dose distribution in the patient is then calculated, based on the geometrical matrix generated from the CT scans, and using the dose calculation algorithms available in the treatment planning system. A wide spectrum of dose calculation algorithms exist today, including the analytic pencil beam (PB) methods, the kernel-based convolution/superposition (CV/SP) method, and more recently, the Monte Carlo (MC) method. Different heterogeneity correction methods are implemented in these dose calculation methods, and differ by the sophistication of handling scattered photon contributions and lateral electron transport [5, 27]. Therefore, they show different accuracies in predicting dose when spatial heterogeneities are present. We describe these three different dose calculation methods, along with their heterogeneity correction methods as follows:

The representative correction-based analytical methods for the pencil beam method include:

1. The ratio of tissue-air ratios (RTAR), which essentially uses the primary beam effective pathlength method (EPL) [5]. This simple method can yield large errors for dose in or near heterogeneities [112].
2. The power-law (Batho) method [8], which requires lateral charge particle equilibrium (CPE), and thus was found to cause very large errors in the lung dose calculations involving small fields of high-energy beams [39].
3. The equivalent tissue air ratio (ETAR) method [113], which is still used widely in modern systems [27]. Although ETAR is more accurate than the EPL and the Batho methods, due to its 3-D capability, in a lung phantom study, Engelsman *et al.* [40] found that ETAR did not correctly predict the penumbra broadening in the low-density lung because of the method's lack of lateral electron transport.

Overall, the pencil beam algorithms with various heterogeneity corrections share one common constraint: they do not transport scattered electrons, hence they implicitly assume the presence of CPE, which makes these methods limited in accurately handling heterogeneities in the lung.

The convolution/superposition (CV/SP) method is a kernel-based, widely-used dose calculation algorithm with a good heterogeneity correction. Equation 1.1 illustrates the principle of the CV/SP method and indicates the two essential steps in the dose calculations:

$$D(\mathbf{r}) = \int_V T(\mathbf{r}')h(\mathbf{r} - \mathbf{r}')d\mathbf{r}' \quad , \quad (1.1)$$

where  $D(\mathbf{r})$  is the dose at point  $\mathbf{r}$  to be calculated,  $T(\mathbf{r}')d\mathbf{r}'$  is the terma (total energy released per mass [6]) in an infinitesimal volume  $d\mathbf{r}'$  near  $\mathbf{r}'$  from the primary photon

fluence,  $h(\mathbf{r})$  is a normalized point kernel [usually generated by the Monte Carlo (MC) method], and describes the fractional dose distribution in an infinite water phantom from a monoenergetic and monodirectional photon point source. The two basic steps are: (i) calculating the primary photon fluence and the *terma* in the patient body, using the ray-tracing method, while taking into account the density variation along the pathways of the primary photons, and (ii) for each dose point  $\mathbf{r}$ , accumulating the contribution from the *terma* released near  $\mathbf{r}'$ , by looking up the fractional dose at the relative position  $\mathbf{r} - \mathbf{r}'$  in the kernel and weighting it by the *terma*. The CV/SP method was introduced independently by several investigators in the mid 1980s [4, 13, 76, 84], and has since been extensively studied and used in treatment planning systems [5].

The heterogeneity corrections implemented in CV/SP includes: (i) the *density scaling* method, which applies to the *terma* “lookup” process described above, and correctly accounts for the contribution from the first-scattered photons along the spatial heterogeneities; and (ii) the *kernel tilting* method, which tilts  $h(\mathbf{r})$  according the direction of the primary photons, and thus accounts for the geometrical divergence of the incident beams. These methods, along with the feature that the kernel can implicitly transport scattered electrons, the CV/SP method is superior to the pencil beam methods with correction-based heterogeneity corrections [27]. However, the kernel  $h(\mathbf{r})$  is generated in an infinite, uniform medium. Therefore, it is expected that inherent errors will occur when it is applied to a finite, heterogeneous geometry, such as the human body. Also, the heterogeneity corrections for the CV/SP method can not correctly account for the contributions from multiple-scattered photons. Therefore, compared to the Monte Carlo method, which is described next, the CV/SP still shows significant errors in regions that lacks CPE [64]. Also, differences

between different CV/SP algorithms implemented in various commercial treatment planning systems could be significant when applied to the dose calculations in a lung treatment planning case [120].

The Monte Carlo method is a totally different dose calculation algorithm. It transports the particles (mostly photons and electrons in radiotherapy) explicitly by using the first principles of physics and follows exactly the statistical nature of the interactions between particles and the background matter by using the knowledge of microscopic cross sections that govern different interactions. The simulation of photon transport in the patient's body can be briefly described as follows:

1. Generate a "source" photon with a particular initial energy and direction by sampling the energy spectrum and the directional distribution of the source.
2. Transport this photon along its direction of flight to the next interaction point by a pathlength sampled from a probability distribution function (pdf) describing the exponential attenuation of the photons in matter, which is governed by the mean free path (MFP) of the photon.
3. Determine the type of the interaction statistically, and obtain possible secondary electrons.
4. Continue to transport the photon until it leaves the region of interest or is absorbed.
5. Transport the secondary electron, either in the same analog way as for the photon, or using the approximate condensed history method [11], in which the electron travels a greater distance between collisions (the *step size*) than the actual MFP for each individual collisions.

6. Deposit energy from the secondary electron along its track between two consecutive “hard” collisions, by sampling the corresponding pdf for energy dissipation.
7. Determine the type of “hard” collision statistically, as is done with the photon (possible secondary electron(s) can be emitted in this “hard” collision). And
8. Continue to transport all the secondary electrons generated by the primary photon until they all stop or leave the region of interest.

During this process, the interfaces between different materials (the heterogeneities) are crossed appropriately, and the energy deposited by the electrons is scored locally. The distribution of dose (energy deposited per unit mass) is then obtained within the region of interest. Because of the statistical nature of the Monte Carlo method, the dose tallied is subject to statistical fluctuations, which are governed by the central limit theorem. Therefore, a sufficient number of photons must be simulated before obtaining a dose distribution with good statistics. In general, the Monte Carlo method is the slowest of the available dose calculation methods, although several variance reduction techniques, such as splitting, Russian roulette, interaction forcing, etc., can be used to increase the calculation speed. However, the Monte Carlo method, due to its first-principles nature in explicitly transporting both photons and electrons in the patient’s geometry, and its sound interface crossing mechanisms, is by far the most sophisticated and accurate dose calculation algorithm [37,64,99,120]. The expensive calculating time is still an limiting issue for the use of a general purpose Monte Carlo code, such as EGSnrc [58], PENELOPE [104], and GEANT [48], directly in the clinic. However, several Monte Carlo codes, such as DPM [108], VMC++ [1], XVMC [41], MCDOSE [74], PEREGRINE [52], which are optimized in particular for the radiotherapy, have been developed and have shown a large increase

of calculation speed. In spite of the increased accuracy of Monte Carlo for geometrically complex problems, its extreme slowness (compared to pencil beam methods and CV/SP) has made Monte Carlo favored only for specialized research in the radiotherapy community. For practical clinical treatment planning simulations, CV/SP with inhomogeneity corrections is the principal dose calculation algorithm, and the pencil beam method still is used by many treatment planning systems.

The calculated dose distribution can then be evaluated by several common tools, such as the isodose lines [9], the dose volume histogram (DVH) [36,73], the tumor control probability (TCP) [116,129], and the normal tissue complication probability (NTCP) [65,73]. We describe the isodose lines and the DVH next, since these are the two methods we use throughout our analysis.

(i) Isodose lines. The spatial dose distribution in the patient can be viewed in the form of isodose lines, which are generated as a series of contours of various dose values (either in relative or absolute dose values), by interpolating from the dose matrix calculated for a treatment plan. Isodose lines, which are usually superimposed on the patient's anatomical images, help the dosimetrist to visually inspect the treatment plan. Fig. 1.2 [2] shows an example.

(ii) Dose Volume Histogram (DVH). The DVH is another commonly-used graphical tool to assess treatment plans. The most-seen DVHs are in a cumulative form, which specifies the fraction of the volume of a region of interest (e.g., the target, the normal tissue or some OARs) exceeding a given dose level. DVHs are very useful to evaluate the uniformity of dose distributions in the region of interest, and in spotting potential hot or cold spots. Fig. 1.3 [2] illustrates an example of the cumulative DVHs. However, due to its integral nature, the DVH cannot reveal any detailed spatial information about the dose distribution in a region of interest.



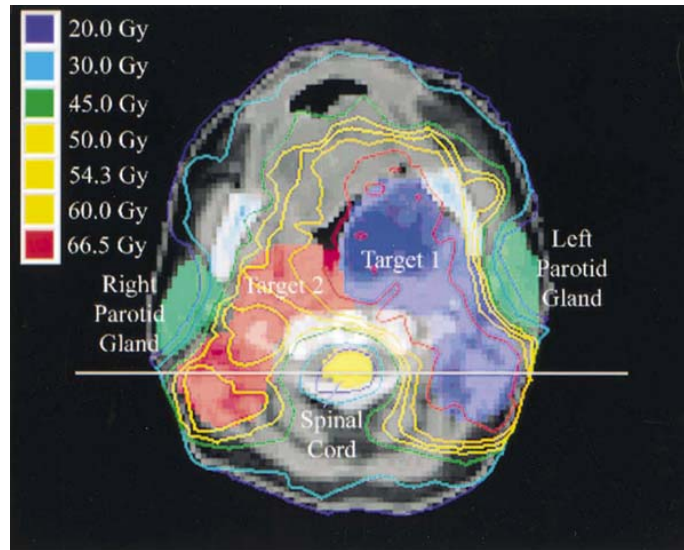


Figure 1.2: Example isodose lines, extracted from [2].

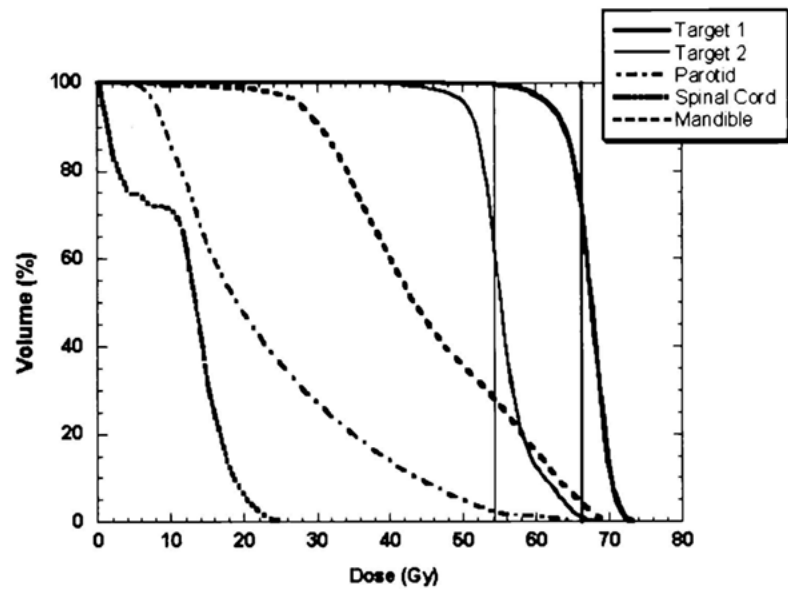


Figure 1.3: Example DVHs, extracted from [2].

Therefore, to minimize the systematic errors caused by the different capabilities of handling heterogeneity correction for different calculation methods, in this thesis, we adopt the Monte Carlo method for our dose calculations.

### **1.1 The Lung Models for Heterogeneity Correction**

As radiotherapy technology advances, and the high-dose region conforms more tightly to the target, the need for more accuracy in each stage of dose delivery increases. Webb [123] specified six points for an ideal dose calculation method, the first two being: (i) include all 3-D geometry information; and (ii) use an accurate 3-D map of the electron density. Before the advent of CT in the early 1970's, dose calculations were performed by assuming the human body to be composed of uniform water, due to the lack of information of the exact anatomy. However, the human body in nature is complicated and contains many heterogeneities, including various tissues of different compositions and air cavities, of different sizes. Any deviation from the real anatomy by ignoring these heterogeneities could lead to inaccurate dose calculations, even if one had a perfect calculation algorithms. The lung is an organ that severely challenges in the accurate prediction of dose distributions: (i) it is highly heterogeneous, with millions of randomly located air-tissue interfaces of various sizes; and (ii) its shape, size, and location change continuously during breathing, which results in a continuous change of the already-present heterogeneities. Many articles [23, 30, 101, 102] have been published addressing the effect of the lung motion on dose calculations. However, in this thesis, we study only the effect of the spatial heterogeneity of the lung by assuming that the lung is static and rigid. Therefore, no motion-related changes will be considered.

Until recently, debates [90] still existed about how to incorporate spatial hetero-

geneities in treatment planning for lung cancer. “Many cancer centers still do not use patient-specific tissue density corrections. [27]” However, the medical physics community [27] has gradually realized the importance of heterogeneity corrections, including but not limited to the lung. To provide background and motivate our work, we next describe the lung’s highly heterogeneous anatomy and the current status of heterogeneity corrections associated with the lung.

### 1.1.1 The Lung’s Anatomy

From the perspective of particle transport, we need to know two quantitative properties of the lung: (i) the materials that composed the lung; and (ii) the spatial location of these materials. The materials composing the lung are relatively simple: air and tissue. However, the locations of these two materials show a “random” character, due to the highly heterogeneous structures inside the lung: three main bifurcating “trees”, the bronchi, the arteries, and the veins, are embedded in the parenchyma, which occupies most of the lung, and contains hundred of millions of alveoli [31]. The quantitative measurements of the lung’s internal structures [51, 53–55, 91, 115, 124] show a size range covering several orders of magnitude: from the order of 1 cm for the principle bronchi and the main vessels, to about  $10^{-4}$  cm for the alveoli in the lung’s parenchyma. The bifurcating “trees” can be quantitatively described by an *order* system introduced by Horsfield [53]. In this numbering system, the largest structures have the highest order numbers, while the smaller structures have smaller order numbers. Using this system, the characteristics of structures, such as their numbers in the whole lung, their diameters, and their lengths, can be counted or measured in an order-wise manner. For example, the 25th order of bronchi, the lower lobe bronchi, count 2 in two lungs, and measure a typical diameter of 0.7 cm and a typical length of 1.2 cm. In comparison, the 1st order of bronchi,

the terminal bronchioles, count 25000 in two lungs, and measure a typical diameter of 0.051 cm and a typical length of 0.11 cm. In addition, the lung's structures have some unique features: (i) the arteries accompany the bronchi, while the veins are separate from them [49]. The size of the vessels is roughly proportional to that of the bronchus of the same order [79]. (ii) Beyond the terminal bronchioles, the morphology of the lung changes largely due to the transit from the purely-conducting airway to the respiratory region, where the basic parenchyma units, the acini, are alveolated. Simultaneously, the relatively scattered blood vessels change into a network of capillaries, which forms the part of the thin wall of the alveoli. (iv) Since the lung consists of about 75% of air, which is either contained in the airways' lumen, or in the hundreds of millions of small pouch-like alveoli, the number of the heterogeneous interfaces between the two materials, tissue and air, of great density difference, is huge. These features are reflected in the lung model soon to be discussed in Section 1.1.4.

Overall, the heterogeneous anatomy of the lung has the following unique characteristics:

- (i) Chunk sizes occur in a range of several orders of magnitude;
- (ii) A huge number of structures scatter “randomly” inside the lung;
- (iii) Basically two materials, tissue and air, alternate inside the lung, and thus a huge number of material interfaces exist.

### 1.1.2 The Atomic Mix Model

Since most of the lung is occupied by the relatively uniform parenchyma (compared to the more heterogeneous large structures with higher order numbers), the natural way to introduce a heterogeneity correction to the previous water-equivalent assumption is to model the lung as a whole “homogenized” organ at its mean den-

sity. This model is often called the *atomic mix* or *mean density* lung model. The atomic mix lung model has been widely used [12, 14, 16, 20, 22, 24, 40, 62, 75, 83, 88, 97, 103, 114, 118, 125, 126, 128] to study the effect of the low-density lung on dose distributions with a variety of combinations of geometry setups and beam arrangements, and either by measurements or by calculations.

Rice *et al.* [97] devised four clinically-relevant phantoms, which used homogeneous materials of different densities (0.015, 0.18 and 0.31 g/cm<sup>3</sup>, respectively) to represent the atomic mix lung, and did benchmark measurements of the dose along the central axis (CAX) of the photon beams of various field sizes (5 × 5 to 20 × 20 cm<sup>2</sup>) and two different energies (4 MV and 15 MV, to bracket the most used energies). Fig. 1.4 contains three representative plots extracted from the same reference, which show some common features when introducing a low-density atomic mix lung as

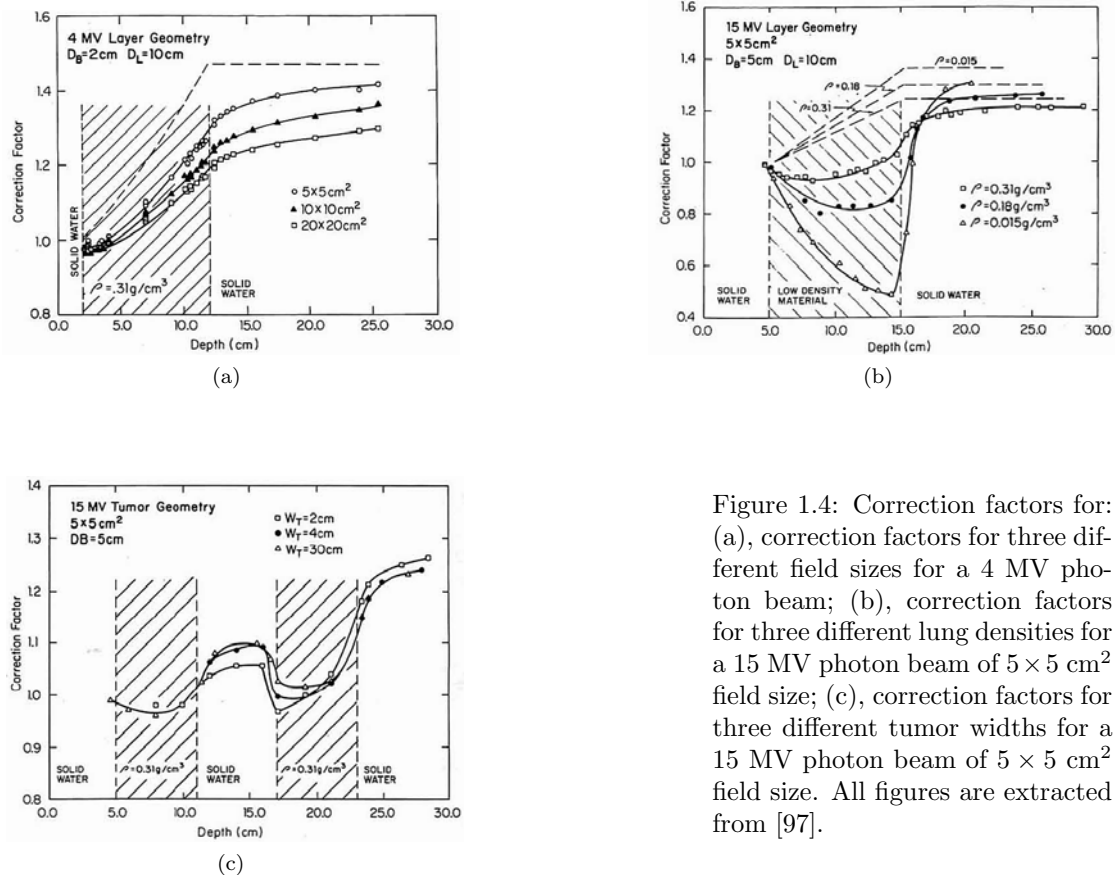


Figure 1.4: Correction factors for: (a), correction factors for three different field sizes for a 4 MV photon beam; (b), correction factors for three different lung densities for a 15 MV photon beam of 5 × 5 cm<sup>2</sup> field size; (c), correction factors for three different tumor widths for a 15 MV photon beam of 5 × 5 cm<sup>2</sup> field size. All figures are extracted from [97].

the heterogeneity correction. The correction factor (CF) in the figures is defined as  $CF = D(h)/D(w)$ , where  $D(h)$  is the dose in the heterogeneous atomic mix lung and  $D(w)$  is the dose at the same point if the lung were replaced by water. Fig. 1.4a shows, for a low energy photon beam, where lateral charged particle equilibrium (CPE) is more readily established on the central axis (CAX), than a higher energy beam, that: (i) a lower dose occurs in the atomic mix lung in the first few cm, and then a higher dose occurs for the remaining part of the lung for all field sizes, which is a combined result of the reduced scattering and increased primary fluence; (ii) a higher dose occurs in the downstream water layer for all field sizes due to the reduced attenuation of the primary photons; (iii) a shallow gradient occurs after the lung, due to the shorter electron range in water than in the atomic mix lung; and (iv) a smaller difference occurs with a larger field size, due to the effect of the lateral CPE. Fig. 1.4b shows the density effect for a beam with high energy and a small field size, where the loss of lateral CPE exists on the CAX. Here we see two other common features: (i) a larger difference with a lower density of the lung; and (ii) a clear “buildup” region upon entering the water layer after the lung, with a density-dependent slope. These two characteristics are due to the loss of lateral CPE, which is more severe with the lung with a lower density. Fig. 1.4c shows a clinically relevant situation with a tumor of various sizes embedded inside the lung. Also, one new feature occurs here: beside the “buildup” region on the upstream side of the tumor, a “buildddown” region occurs on the downstream side of the tumor, due to increased forward-scattering from the high-density tumor and decreased back-scattering from the low-density lung. Both of these effects yield a nonuniform dose distribution inside the tumor.

Because of these “new” features (compared to the “water” lung), the low-density atomic mix lung model has been used extensively, especially in comparing the dose

distributions calculated by different dose calculation algorithms with measurement. For example, Tang *et al.* [114] compared different pencil-beam dose calculation methods using a phantom containing two layers of atomic mix lung. Mackie *et al.* [75] quantified the errors of the pencil-beam algorithms using also a layered lung phantom. The accuracy of heterogeneity effects for eight calculation methods were recently examined by comparing the calculated CAX depth dose to the measurement [126]. More recently, the atomic mix lung model was used for validating the capability of the Monte Carlo method in dealing with tissue heterogeneity [20–22].

The atomic mix model has also been applied to evaluate other clinically relevant issues. Chetty *et al.* [24] investigated the influence of different beam models on dose calculations using a life-size thoracic phantom with two lung regions containing a homogeneous lung-equivalent material. Klein *et al.* [62] performed treatment planning for lung cancer on an anthropomorphic phantom with two atomic mix lung regions and provided cautions of using simplistic heterogeneity correction algorithms.

### 1.1.3 The CT Model

Since the invention of computed tomography (CT), not only could the positions of the target, the organs at risk (OARs), and the normal tissue for a specific patient be precisely contoured, but the density of the anatomies could be also used as the input for dose calculations, and thus the practical importance of heterogeneity correction methods, such as those discussed previously, could be truly evaluated [42].

Although the results coming from the atomic mix lung model “should prove helpful in understanding the different physical processes contributing to dose distributions in and near the regions with lungs” [97], the direct application of the atomic mix model in the lung treatment planning is too crude to be realistic. This is indicated in Fig. 1.4b, where different densities of the atomic mix lung result in significant

changes in dose, while the mean density of the lung among people were shown to vary widely with ages, respiration phases of the patient [119]. Van Dyk [118] analyzed the relationship between the mean lung density and the heterogeneity correction factor and applied this to study the difference of the dose between a CT-based lung geometry and a homogenized atomic mix version of the same lung. He concluded that for a majority of patients, the detailed CT-based anatomic information was needed to achieve an accuracy of 5% in the dose delivered to the lung. A collective work also recommended the voxel-by-voxel CT-based dose calculations [93].

Currently, the CT-based treatment planning is widely used [7, 28, 33, 37, 60, 80, 92, 98, 100, 122, 127]. Therefore, the accuracy of dose calculations depends on the extent to which the matrix of the CT voxels represents the patient's real geometry. The effect of CT numbers, which determine the electron densities for each CT voxel, has been well-investigated [25, 44, 50, 57, 61, 85, 106]. However, publications on the effect of CT resolutions, or alternatively, the sizes of the CT voxels, are limited. Cygler *et al.* [32] suggested a 1.9 mm CT voxel size over a coarser 3.9 mm one in the proximity of air-tissue interface, due to an error of more than 5% near this heterogeneity. Chung *et al.* [26] used a series of voxel sizes, ranging from 1.5 mm to 6 mm, and found differences up to 5.6%. They then concluded that a 2 mm size was required for accurate dose calculations in heterogeneous regions. De Smedt *et al.* [34] generated an geometrical grid and an dose scoring grid, which are independent of each other, and performed treatment planning for a lung case and a head and neck case, for a series of combinations of different sizes of the geometrical grid and different sizes of the dose scoring grid. They showed that for the lung case, the geometrical resolution was more important than the dose scoring resolution. This was consistent with an early article [44], in which Geise and McCullough indicated that it was more



important to know the accurate distribution of the spatial heterogeneities than the accurate electron density. De Smedt *et al.* [34] also recommended a CT resolution of 2 mm for the studies lung cancer cases.

#### 1.1.4 The Validity of the Atomic Mix Approximation

As stated in Section 1.1.1, the lung is a highly heterogeneous organ consisting of a hierarchy of structures with a dramatic range of sizes. The locations of these many structures are spatially random and can change with time. According to traditional transport theory [94], a heterogeneous spatial system can be accurately homogenized into an atomic mix counterpart only when the chunk sizes in the system are small compared to the mean free path (MFP) of particles traveling inside. This “atomic mix” approximation has been successfully used in the nuclear engineering and physics communities for years. However, the rigorous mathematical proof of the atomic mix approximation was not achieved until recently by Dumas and Golse [38].

The dose deposition in matter by a photon beam consists of two processes [6]: (i) the photons travel through the background matter and emit secondary electrons through various types of interactions, including the photoelectric effect, Compton scattering, and pair-production; (ii) the secondary electrons slow down in the background matter through the Coulomb force interacting with the electric field of the atoms, and the energy of the electrons is dissipated along their tracks. Even the largest structure of the lung is small compared to the MFP for the photons, which is in the order of cm within the energy range of interest in radiotherapy. Therefore, the atomic mix model is valid for photons. If electrons deposit their energy locally (no transport), the atomic mix lung model would then be valid for the dose calculations. In effect, this situation holds true when CPE exists. However, when CPE does not exist, for example, in the proximity of geometrical heterogeneities, or near

the edge of a beam, the validity of the atomic mix approximation for electrons is open to question. Since the smallest structures inside the lung are much larger than the electron MFP, which is in the order of  $\mu m$ , the atomic mix lung model is not automatically valid in the absence of CPE.

The CT-based lung model, although providing a more detailed description of the geometry of the lung than the crude atomic mix model for the whole lung, is still essentially the atomic mix model, which is now applied to each individual voxel. According to the discussion above, the CT-based voxelization is not automatically valid for electrons, and therefore, the calculated dose distribution may deviate from the “true” one, unless the CT resolution is small enough to disclose every detail of the lung structure.

## 1.2 The Goals of This Thesis

As stated above, the knowledge about the impact of the lung’s highly heterogeneous structures on dose calculations for lung has been limited by the CT resolutions present, which are larger than most of the lung’s internal structures. Our primary goal is then to investigate this impact, by explicitly building a detailed lung model, based on the lung’s morphological data.

To our knowledge, such a detailed lung model has not previously been developed and used to assess dose calculations, due to the huge number of structures, which makes a literal rendering of the lung impractical as the geometry input for any practical dose calculation method.

Since modeling every detail of lung is required, according to traditional transport theory (because the MFP of the electrons is much less than the finest structure of the lung and thus no atomic mix at any level is allowed), we first theoretically extend

the traditional atomic mix theory and raise the MFP limit, in order to explicitly model only a reasonably small number of the larger lung’s structures. By applying a formal asymptotic analysis, which is based on the recent work by Larsen and Liang [68], our new atomic mix theory states that a heterogeneous spatial system can be approximated by the atomic mix model if the chunk sizes in this system are small compared to the *transport* MFP of the particles, instead of the MFP. The transport MFP is, roughly, the path length that an particle will have to travel to be deflected by an  $O(1)$  angle. Due to the highly forward-peaked nature of the dominant “soft” (elastic) collisions of electrons with the background matter, the transport MFP for electrons is orders of magnitude larger than the MFP. Most importantly, the transport MFP of the electrons, within the energy range of interest in radiotherapy, is larger than a certain orders of structures inside the lung, and this makes a detailed lung model possible.

Using the new atomic mix theory, we build a detailed “random” lung model, and we assess the effect of the detailed and randomly-located lung’s structures on dose calculations, compared to the atomic mix lung model. This random lung model should be realistic, yet simple enough to implement the current Monte Carlo codes available for simulations. Therefore, in our random lung model, all physical structures are not explicitly modeled. Instead, we set a reasonable threshold size, guided by our new atomic mix theory and the lung’s morphology, and we homogenize all the structures in the lung smaller than this threshold size into an atomic mix “background” with a density less than the mean density of the whole lung. We then explicitly embed the remaining structures larger than the threshold size into the background. We then compare doses obtained from different *realizations* of the random lung model and the atomic mix lung model; significant perturbations in dose due to the random

structures of the lung will be analyzed.

To illustrate the application of the random lung model, we apply our random lung model, by voxelizing it for different geometrical realizations and CT resolutions, to assess the effect of CT resolutions for lung dose calculations in a more realistic treatment-like phantom setup with a tumor embedded inside the lung. The differences between the dose distributions from different CT resolutions will be examined. The result of this process is an estimated value of the CT resolutions at which Monte Carlo dose calculations are accurate. (We do not assess the accuracy of pencil beam or convolution methods for these problems; this is beyond the scope of our project and there is no theoretical reason why these methods, with their crude inhomogeneity corrections, will be accurate for such highly heterogeneous systems.)

### 1.3 Organization of this thesis

The remainder of this thesis is organized as follows:

Chapter II describes the relevant detailed morphology of the human lung, which is a highly heterogeneous organ consisting of a hierarchy of structures in a dramatic range of sizes. Using this published information on the lung’s structures, we will (in Chapter IV) build a “random lung model,” which is a simplified  $2\frac{1}{2}$ -D version of the 3-D lung, preserving essential features of the realistic 3-D lung.

Chapter III, which is partly based on a paper accepted for publication in *SIAM J. Appl. Math.*, deals with the atomic mix approximation for charged particle transport in a heterogeneous random system [68]. In this chapter, a one-dimensional Boltzmann equation is approximated by its Boltzmann Fokker-Planck (BFP) limit by applying the theoretical approach of Pomraning on the “soft” collision operator, using the highly forward-peaked nature of the “soft” (elastic) electron scattering in matter.

Then, by introducing two independent dimensionless variables to express the “fast” and “slow” component of the angular flux, a formal asymptotic analysis is applied to obtain the atomic mix approximation of the BFP equation. The purpose of this analysis is to show that when charged particle transport occurs in a random heterogeneous system and the chunk sizes of the materials are small compared to the transport mean free path, then the solution of the Boltzmann equation is very well approximated by the solution of the atomic mix Boltzmann equation (in which all the cross sections are replaced by their volume averages). We then build a random “droplet” model to numerically demonstrate the validity of this asymptotic theory.

Chapter IV is partly based on a paper published on *Medical Physics* [70]. In this chapter, we use the anatomical data discussed in Chapter II and apply the atomic mix approximation theory discussed in Chapter III to build a realistic “mixed”  $2\frac{1}{2}$ -D random lung model. In this model, structures larger than a threshold size (which is carefully selected at the order of the terminal bronchioles) are explicitly modeled and embedded in the homogenized “background” with an adjusted density, which represents the structures smaller than the same threshold size. We then use the Monte Carlo method to compare the dose distributions between different realizations of our random lung model and their atomic mix equivalent, using a one-beam phantom setup.

In Chapter V, we present an application of our random lung model: we determine an optimal CT resolution for lung treatment planning, in a realistic treatment planning phantom setup. The effect of different sizes of geometrical voxels used to describe the lung’s geometry on dose calculations is analyzed, for various tumor and field sizes.

We summarize our conclusions and suggest future work in Chapter VI.

## CHAPTER II

### The Lung's Anatomy

The lungs [49] are the air exchange part of the respiratory system. They are spongy and look roughly like a half-cone. Down from the end of the trachea, the conducting airway consists of tubular structures with concentric layers of different tissues [105], which divides continuously to the edge of the lung. As the sizes of these structures decrease, the purely conducting airway transitions from the large principal bronchi, through the medium lobar bronchi and small intrasegmental bronchi, to the terminal bronchioles. Each terminal bronchiole then leads to a parenchymal unit: the acinus, which contains alveolated respiratory bronchioles, alveolar ducts and numerous pouch-like alveoli that serve as the place of air exchange. These small structures are separated by thin walls that consist of the capillary network and the interalveolar septa of epithelium. There are two kinds of pulmonary blood vessels: the arteries and the veins. They bifurcate into binary vessel trees like the bronchi from the hila of the lungs, and the two vessel trees meet at the capillaries in the walls of the alveoli.

#### 2.1 The Lung's Morphology

In order to model the transport of particles (photons/electrons) in the lung, both the material composition and the geometrical arrangement of the internal structures of the lung are needed.

The lung is made up of three basic components: air, blood and tissue [124]. The blood and the tissue can be well-approximated as water.

However, compared to the relatively simple material composition of the lung, the hierarchical information, such as the variation of diameter and length of branches of the lung's internal structures with successive levels, are more complicated. There have been systematic, quantitative morphologic analyses of the structures of the human lung over the last several decades, either by measuring prepared casts of the airways and vessels [51, 53–55, 91, 124], or by visualizing and analyzing them on in vivo CT scans [115]. Our realistic lung model (see Chapter IV) is based on these quantitative works, which are briefly described next.

### 2.1.1 The Horsfield Orders

Two different methods have been developed to describe the hierarchy of the branches of the airways and the arterial and venous trees in the lung: *generation* [124], which numbers from principal bronchi down to the periphery, and *order* [53], which numbers in the other way from the periphery toward the stem.

In the *generation* system, the trachea is generation 0, the principal bronchi generation 1, etc., with the generation number increasing by 1 at each division. In the *order* system, the farthest (smallest) branches are defined as order 1. Two small branches join together and become a larger branch, which is one order greater than the higher one of the two small branches. This yields the so-called Horsfield orders. In the Horsfield method, order numbers are continuous along the longest pathway, where they are equal to the generation numbers. While the generation method is natural for an asymmetric binary tree such as that in the lung, the order system is better in grouping similar size of levels of branches together. The difference and relationship between these two numbering systems are illustrated in Fig. 2.1. We use

the Horsfield order throughout this thesis.

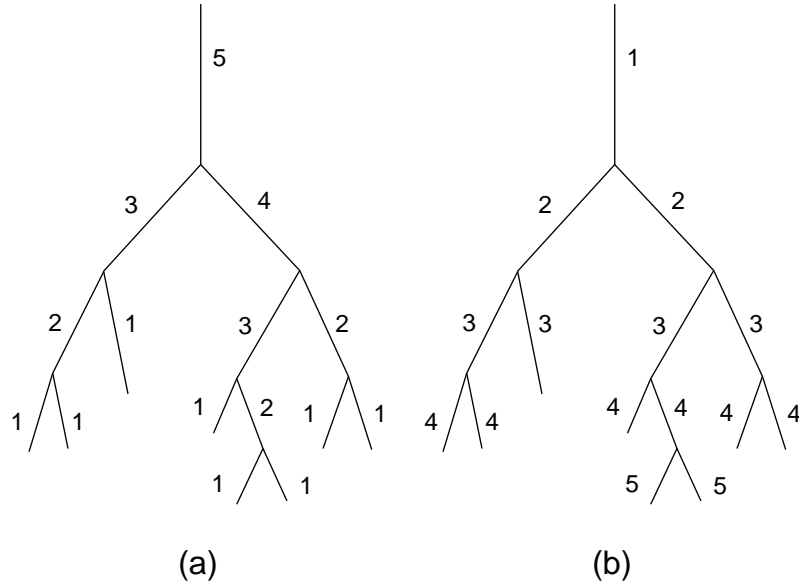


Figure 2.1: (a) Horsfield orders; (b) Weibel's generations

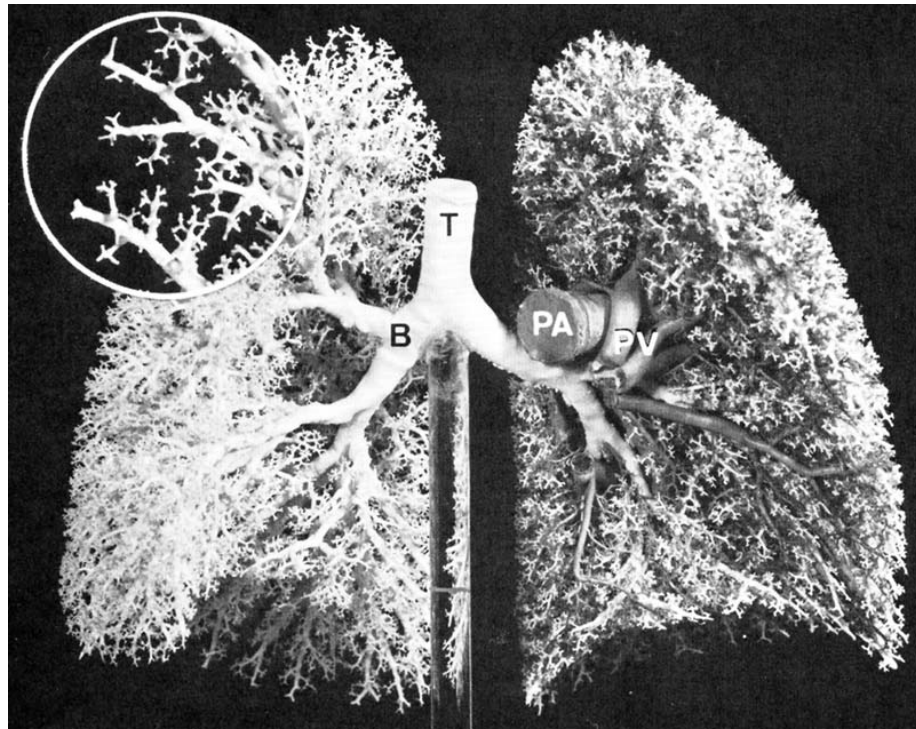
### 2.1.2 The Morphometric Data

Within each order, the lengths and diameters of the branches have a typical range of sizes [54]. The mean size of the bronchi decreases continuously. The purely conducting airways of the bronchial tree start from the right/left principal bronchus, with a typical luminal diameter of 1.2 cm, and end at terminal bronchioles, with an internal diameter in the range of 0.03 cm to 0.1 cm [31]. After each terminal bronchiole is the acinus: the complex of alveolated airways and the largest parenchymal unit, which contains three generations (on average) of respiratory bronchioles and numerous alveoli, where gas exchange mainly occurs. The acinus has an average volume of 187 ml and numbers 26,000–32,000 in both lungs, assuming a total lung capacity of 5–6 liters [51]. The internal airway diameter inside an acinus falls from 0.05 cm to 0.027 cm [51]. The end structure containing air is the thin-walled bubble-like alveolus, with a mean diameter of about 0.025 cm and a membrane thickness on the

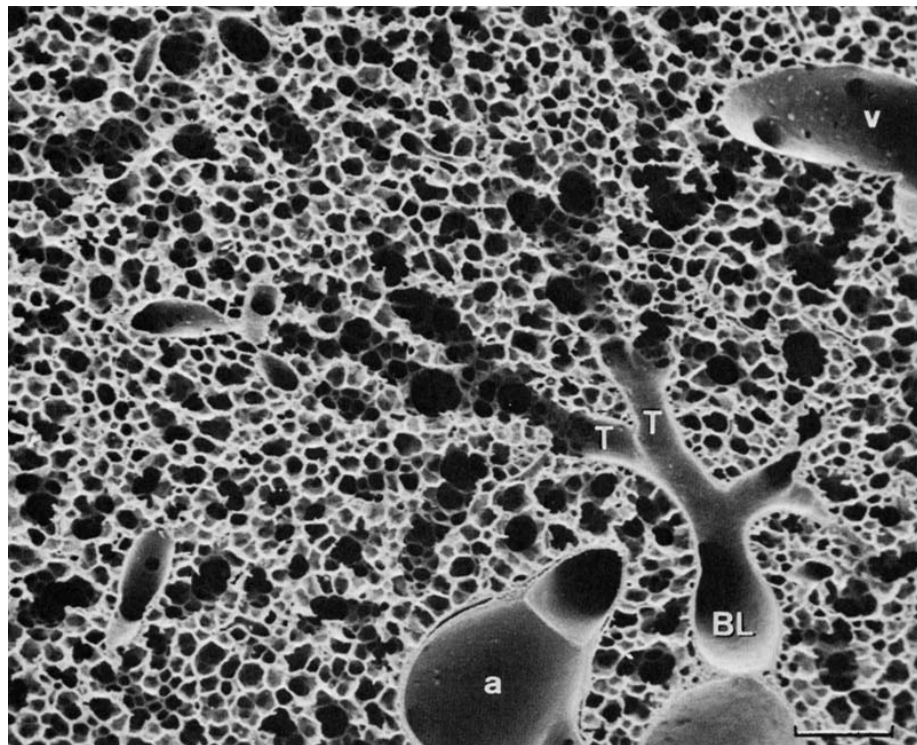


order of  $\mu m$  [31]. The total number of alveoli in each adult lung ranges from  $2 \times 10^8$  to  $6 \times 10^8$ , depending on body size [31]. Fig. 2.2a [19] is a resin cast of a human lung which shows the bifurcating style of the bronchi (B) that originate from the trachea (T). The pulmonary arteries (PA) and veins (PV) are also indicated in this figure. Fig. 2.2b [19] is an scanning electron micrograph of the distal airways, which include the small bronchioles (BL), the terminal bronchioles (T) and the parenchymal respiratory bronchioles and alveolar ducts. A pulmonary artery (a) and a vein (v) are also shown. The scale marker included is  $200 \mu m$ .

Table 2.1 shows the characteristic numbers and sizes for each order of structures in the lung's conducting airways and blood vessels. In this table, the terminal bronchioles have order 1 and the main bronchi have order 28. The lengths and diameters are from reference [53] and [105], but the diameters for the accompanying arteries and veins are calculated according to the method in Section 2.2. Also,  $n$  is the number of structures in the corresponding order and  $m$  the number of the same structures in the model (see Section 4.2). Table 2.2 provides the morphometric data for the parenchymal unit, the acinus. In this table, the terminal bronchioles are order 0 as a reference point and all the smaller structures have a negative order accordingly. All the parameters have the same meaning as in Table 2.1, but no blood vessels are listed here because down from the terminal bronchiole, the blood vessels begin to develop into the capillary network surrounding the acini. This morphological transition will be reflected in our lung model used for the numerical validation of the theory in Chapter IV.



(a) Bronchial tree



(b) Distal airways

Figure 2.2: Lung's airways

Table 2.1: The conducting airways and the accompanying vessels' morphometry [53]

Horsfield order	Structures	Length (mm)	Lumen diameter (mm)	Bronchus diameter (mm)	Artery/vein diameter (mm)	$n$	$m$
28	Large bronchi	100	16	26.67	22.86	0	0
27		40	12	20.00	17.14	0	0
26		26	10.3	17.17	14.71	2	1.86
25		18	8.9	14.83	12.71	2	1.29
24	Medium bronchi	14	7.7	12.83	11.00	2	1.00
23		11	6.6	11.00	9.43	3	1.18
22		10	5.7	9.50	8.14	6	2.14
21		10	4.9	8.17	7.00	8	2.86
20		10	4.2	7.00	6.00	12	4.29
19	Small bronchi	10	3.5	5.83	5.00	14	5.00
18		9.6	3.3	5.50	4.71	20	6.86
17		9.1	3.1	5.17	4.43	30	9.75
16		8.6	2.9	4.83	4.14	37	11.36
15		8.2	2.8	4.67	4.00	46	13.47
14		7.8	2.6	4.33	3.71	64	17.83
13		7.4	2.4	4.00	3.43	85	22.46
12		7	2.3	3.83	3.29	114	28.50
11		6.7	2.2	3.67	3.14	158	37.81
10		6.3	2	3.33	2.86	221	49.72
9	5.7	1.78	2.97	2.54	341	69.42	
8	5	1.51	2.52	2.16	499	89.11	
7	4.4	1.29	2.15	1.84	760	119.43	
6	3.9	1.1	1.83	1.57	1104	153.77	
5	3.5	0.93	1.55	1.33	1675	209.38	
4	Bronchioles	3.1	0.79	1.32	1.13	2843	314.76
3		1.1	0.64	1.07	0.91	5651	222.00
2		1.3	0.56	0.93	0.80	11300	524.64
1		1.1	0.51	0.85	0.73	25000	982.14

Table 2.2: The acinus morphometry [105]

Horsfield order	Structures	Number in acinus	Length (mm)	Diameter (mm)	$n$	$m$
0	Terminal Bronchiole	1	1.10	0.51	25000	982.14
-1	Respiratory bronchiole 1	2	0.97	0.47	50000	1732.14
-2	Respiratory bronchiole 2	4	0.97	0.47	100000	3464.29
-3	Respiratory bronchiole 3	8	0.88	0.49	200000	6285.71
-4	Alveolar duct 1	19	0.66	0.50	475000	11196.43
-5	Alveolar duct 2	45	0.51	0.49	1125000	20491.07
-6	Alveolar duct 3	108	0.58	0.51	2700000	55928.57
-7	Alveolar duct 4	254	0.43	0.40	6350000	97517.86
-8	Alveolar duct 5	374	0.41	0.38	9350000	136910.71
-9	Alveolar duct 6	366	0.30	0.30	9150000	98035.71
-10	Alveolar duct 7	146	0.28	0.27	3650000	36500.00
-11	Alveolar duct 8	58	0.22	0.24	1450000	11392.86

## 2.2 Other Relevant Parameters

### 2.2.1 Lung Density

The mean bulk density of the lung can vary dramatically among different people [119]. Depending on the specific conditions such as age, respiration phase, body position, disease, etc., the mean lung density can range from less than  $0.2 \text{ g/cm}^3$  to greater than  $0.4 \text{ g/cm}^3$ . We will adopt the ICRU [87] value  $0.26 \text{ g/cm}^3$  for the whole lung density in all our calculations.

### 2.2.2 $T/D$ Ratio

Table 2.1 only gives the luminal diameters of the airways. The wall thickness of the bronchi is approximately proportional to the luminal diameter and can be determined by the “ $T/D$  ratio” [47,78,79], which is defined as the wall thickness ( $T$ ) divided by the total diameter of the bronchus ( $D$ ). The  $T/D$  ratio has no statistically significant difference between segments, lobes and lungs; and furthermore, no significant correlation was shown between  $T/D$  ratio and age [79]. A  $T/D$  ratio of 0.2 is used in our lung model.

### 2.2.3 Bronchoarterial Ratio

The pulmonary arteries run parallel to the bronchi, while the veins are separate from these. In healthy individuals, the diameters of the pulmonary artery is approximately equal to that of its accompanying bronchus [79]. The “bronchoarterial ratio” is the diameter of the bronchial lumen ( $D-2T$ ) divided by its accompanying pulmonary artery. The bronchoarterial ratio shows a significant correlation with age. A mean value of 0.695 [79] is adopted in our lung model.

## CHAPTER III

# The Atomic Mix Approximation for Charged Particle Transport

### 3.1 Introduction

Particle transport in a physical system can be described by a linear Boltzmann equation [10,18,56,94]. For a complicated heterogeneous system consisting of two or more materials with varying chunk sizes, a general linear Boltzmann equation with both space- and energy-dependent cross sections is usually difficult and expensive to solve. If the typical chunk sizes of different materials in such a system are small compared to the mean free path (MFP) of the particles transporting inside, an accurate solution of the original Boltzmann equation can be obtained by solving an equivalent “atomic mix” Boltzmann equation in which the spatially-varying cross sections are replaced by their volume-averaged counterparts over the physical system [94]. This atomic mix Boltzmann equation is simpler and much easier to solve than the original one.

The atomic mix approximation has recently been proved mathematically by Dumas and Golse [38]. The more recent work by Larsen, Vasques, and Vilhena [66,67] also showed, using a formal asymptotic analysis, that the atomic mix approximation is valid in a 1-D diffusive stochastic system when the chunks are comparable in size to a mean free path.

In this chapter, we show that to accurately homogenize a heterogeneous system, the typical chunk sizes should be small compared to the *transport* MFP of the particles, which for charged particle is much larger than their MFP, because the scattering is very forward-peaked.

For simplicity, we consider the 3-D Boltzmann equation for electrons and specialize this equation to planar geometry. We apply Pomraning's method [95] to approximate the "soft" collision operator by its Fokker-Planck limit [17] (valid because of the forward-peaked scattering nature of these collisions) and obtain the Boltzmann-Fokker-Planck (BFP) equation [15, 96]. Finally, we express the solution of the BFP equation using two independent dimensionless spatial variables and employ a formal asymptotic analysis to obtain the atomic mix limit of the BFP equation. Numerical results obtained by employing Monte Carlo method are provided to validate this new theory. A thorough analysis using a 3-D, energy-dependent transport equation is presented in a recent work by Larsen and Liang [68].

### 3.2 The 1-D Linear Boltzmann Equation

The 3-D energy-independent particle transport equation is as follows:

$$\begin{aligned} & \boldsymbol{\Omega} \cdot \nabla \Psi(\mathbf{x}, \boldsymbol{\Omega}) + \Sigma_s(\mathbf{x}) \Psi(\mathbf{x}, \boldsymbol{\Omega}) \\ &= \int_{4\pi} \Sigma_s(\mathbf{x}, \boldsymbol{\Omega} \cdot \boldsymbol{\Omega}') \Psi(\mathbf{x}, \boldsymbol{\Omega}') d\Omega' \quad , \quad \mathbf{x} \in V \quad , \end{aligned} \quad (3.1)$$

where:

$$\begin{aligned}\Sigma_s(\mathbf{x}, \boldsymbol{\Omega} \cdot \boldsymbol{\Omega}') &= \Sigma_s(\mathbf{x}, \mu_0) \quad (\mu_0 = \boldsymbol{\Omega} \cdot \boldsymbol{\Omega}' = \text{scattering cosine}) \\ &= \sum_{n=0}^{\infty} \frac{2n+1}{2} \Sigma_{sn}(\mathbf{x}) P_n(\mu_0) \quad (P_n(\mu_0) = \text{Legendre polynomials}) \\ &= \text{differential scattering cross-section} \quad ,\end{aligned}\tag{3.2}$$

$$\Sigma_{sn} = \int_{-1}^1 \Sigma_s(\mathbf{x}, \mu_0) P_n(\mu_0) d\mu_0 \quad , \quad 0 \leq n < \infty \quad ,\tag{3.3}$$

$$\Sigma_s(\mathbf{x}) = \Sigma_{s0} \quad .\tag{3.4}$$

For simplicity, we ignore the boundary conditions and source for Eq. (3.1). These can be included without difficulty. Since electrons are not absorbed in their interactions with matter, the macroscopic total cross section  $\Sigma_t$  is equal to the macroscopic scattering cross section  $\Sigma_s$ .

We can group the electron collisions with matter into two types [6,104]: (i) “soft” collisions, which include both elastic and inelastic scattering occurring when the impact parameter  $b$  is far greater than the atomic radius  $a$  (and thus the electrons mainly interact with the Coulomb field of the atom); and (ii) “hard” collisions, which occur when  $b$  is comparable to or less than  $a$  and include large-angle elastic scattering, bremsstrahlung emission and  $\delta$ -ray emission. Typically, the number of “soft” collisions during the slowing down process is around  $10^5 \sim 10^6$ , depending on the initial energy and the matter, while the number of “hard” collisions is small and is on the order of 10 [11]. An individual soft collision causes little loss of energy and very small deflections of the angle of flight (hence it is very forward-peaked), while a hard collision (termed “catastrophic”) causes large changes in both the angle of flight and the electron energy [11].

We then split the differential scattering cross section into a “hard” and a “soft”



component:

$$\Sigma_s(\mathbf{x}, \mu_0) = \Sigma_h(\mathbf{x}, \mu_0) + \Sigma_r(\mathbf{x}, \mu_0) \quad , \quad (3.5)$$

where  $\Sigma_h$  is the differential scattering cross section for hard collisions and  $\Sigma_r$  is the differential scattering cross section for soft collisions (differential “restricted” cross section).

As with Eqs. (3.2) through (3.4), we have:

$$\begin{aligned} \Sigma_h(\mathbf{x}, \mu_0) &= \sum_{n=0}^{\infty} \frac{2n+1}{2} \Sigma_{hn}(\mathbf{x}) P_n(\mu_0) \\ &= \text{differential scattering cross section for hard collisions,} \end{aligned} \quad (3.6)$$

$$\Sigma_{hn}(\mathbf{x}) = \int_{-1}^1 \Sigma_h(\mathbf{x}, \mu_0) P_n(\mu_0) d\mu_0 \quad , \quad 0 \leq n < \infty \quad , \quad (3.7)$$

$$\begin{aligned} \Sigma_r(\mathbf{x}, \mu_0) &= \sum_{n=0}^{\infty} \frac{2n+1}{2} \Sigma_{rn}(\mathbf{x}) P_n(\mu_0) \\ &= \text{differential scattering cross section for soft collisions,} \end{aligned} \quad (3.8)$$

$$\Sigma_{rn}(\mathbf{x}) = \int_{-1}^1 \Sigma_r(\mathbf{x}, \mu_0) P_n(\mu_0) d\mu_0 \quad , \quad 0 \leq n < \infty \quad . \quad (3.9)$$

Also, Eqs. (3.4), (3.3), (3.5), (3.7) and (3.9) give:

$$\begin{aligned} \Sigma_s(\mathbf{x}) &= \Sigma_{s0}(\mathbf{x}) \\ &= \int_{-1}^1 \Sigma_s(\mathbf{x}, \mu_0) d\mu_0 \\ &= \int_{-1}^1 [\Sigma_h(\mathbf{x}, \mu_0) + \Sigma_r(\mathbf{x}, \mu_0)] d\mu_0 \\ &= \left[ \int_{-1}^1 \Sigma_h(\mathbf{x}, \mu_0) d\mu_0 \right] + \left[ \int_{-1}^1 \Sigma_r(\mathbf{x}, \mu_0) d\mu_0 \right] \\ &= \Sigma_{h0}(\mathbf{x}) + \Sigma_{r0}(\mathbf{x}) \quad . \end{aligned} \quad (3.10)$$

By using Eq. (3.5) and Eq. (3.10), Eq. (3.1) can be written in terms of the hard

and soft (restricted) cross sections:

$$\boldsymbol{\Omega} \cdot \boldsymbol{\nabla} \Psi(\mathbf{x}, \boldsymbol{\Omega}) + \Sigma_{h0}(\mathbf{x}) \Psi(\mathbf{x}, \boldsymbol{\Omega}) = \int_{4\pi} \Sigma_h(\mathbf{x}, \boldsymbol{\Omega} \cdot \boldsymbol{\Omega}') \Psi(\mathbf{x}, \boldsymbol{\Omega}') d\Omega' + L_r \Psi(\mathbf{x}, \boldsymbol{\Omega}) \quad , \quad (3.11)$$

where  $L_r$  is the restricted scattering operator:

$$L_r \Psi(\mathbf{x}, \boldsymbol{\Omega}) = \int_{4\pi} \Sigma_r(\mathbf{x}, \boldsymbol{\Omega} \cdot \boldsymbol{\Omega}') \Psi(\mathbf{x}, \boldsymbol{\Omega}') d\Omega' - \Sigma_{r0}(\mathbf{x}) \Psi(\mathbf{x}, \boldsymbol{\Omega}) \quad . \quad (3.12)$$

For the case of problems with 1-D planar symmetry, the above equations reduce to:

$$\mu \frac{\partial \Psi}{\partial x}(x, \mu) + \Sigma_h(x) \Psi(x, \mu) = \int_{-1}^1 \Sigma_h(x, \mu, \mu') \Psi(x, \mu') d\mu' + L_r \Psi(x, \mu) \quad , \quad (3.13)$$

where:

$$L_r \Psi(x, \mu) = \int_{-1}^1 \Sigma_r(x, \mu, \mu') \Psi(x, \mu') d\mu' - \Sigma_r \Psi(x, \mu) \quad . \quad (3.14)$$

In Eqs. (3.13) and (3.14),

$$\Sigma_h(x, \mu, \mu') = \sum_{n=0}^{\infty} \frac{2n+1}{2} \Sigma_{hn}(x) P_n(\mu) P_n(\mu') \quad , \quad (3.15)$$

$$\Sigma_{hn} = \text{defined by Eq. (3.7)} \quad , \quad (3.16)$$

$$\Sigma_r(x, \mu, \mu') = \sum_{n=0}^{\infty} \frac{2n+1}{2} \Sigma_{rn}(x) P_n(\mu) P_n(\mu') \quad , \quad (3.17)$$

$$\Sigma_{rn} = \text{defined by Eq. (3.9)} \quad , \quad (3.18)$$

and

$$\Sigma_h(x) = \Sigma_{h0}(x) \quad , \quad (3.19)$$

$$\Sigma_r(x) = \Sigma_{r0}(x) \quad . \quad (3.20)$$

### 3.3 The Fokker-Planck Approximation to $L_r$

As defined in Eq. (3.12),  $L_r$  describes the restricted collisions of electrons with matter, which includes scattering with very small change in direction of flight, and

which indicates that the differential scattering cross section  $\Sigma_r(x, \mu, \mu')$  is strongly peaked around  $\mu' \approx \mu$ . Up to this point, we have not introduced any approximation to Eq. (3.1). However, in this section, we use the method by Pomraning [95] to derive the Fokker-Planck approximation of  $L_r$ . This yields the Boltzmann-Fokker-Planck (BFP) approximation to Eq. (3.13).

We have from Eq. (3.14) (temporarily ignoring the spatial variable  $x$ ):

$$L_r \Psi(\mu) = \int_{-1}^1 \Sigma_r(\mu, \mu') \Psi(\mu') d\mu' - \Sigma_r \Psi(\mu) \quad , \quad (3.21)$$

and from Eq. (3.17):

$$\Sigma_r(\mu, \mu') = \sum_{n=0}^{\infty} \frac{2n+1}{2} \Sigma_{rn} P_n(\mu) P_n(\mu') \quad . \quad (3.22)$$

From Eq. (3.18) and (3.9), we also have:

$$\Sigma_{rn} = \int_{-1}^1 \Sigma_r(\mu_0) P_n(\mu_0) d\mu_0 \quad , \quad (3.23)$$

where (Eq. (3.8))

$\Sigma_r(\mu_0)$  = 3-D differential cross section for soft collisions,

and  $\mu_0 = \mathbf{\Omega} \cdot \mathbf{\Omega}' =$  scattering cosine.

The differential scattering cross section  $\Sigma_r(\mu_0)$  is strongly-peaked near  $\mu_0 = 1$  ( $\mathbf{\Omega} \approx \mathbf{\Omega}'$ ). Therefore, a Taylor expansion of  $P_n(\mu_0)$  around  $\mu_0 = 1$  in Eq. (3.23) gives:

$$\begin{aligned} \Sigma_{rn} &= \int_{-1}^1 \Sigma_r(\mu_0) \left[ P_n(1) + (\mu_0 - 1) \frac{dP_n}{d\mu_0}(1) + \dots \right] d\mu_0 \\ &\approx \int_{-1}^1 \Sigma_r(\mu_0) P_n(1) d\mu_0 + \int_{-1}^1 \Sigma_r(\mu_0) (\mu_0 - 1) \frac{dP_n}{d\mu_0}(1) d\mu_0 \\ &= \left[ \int_{-1}^1 \Sigma_r(\mu_0) d\mu_0 \right] P_n(1) - \left[ \int_{-1}^1 (1 - \mu_0) \Sigma_r(\mu_0) d\mu_0 \right] \frac{dP_n}{d\mu_0}(1) \\ &= [\Sigma_{r0}] P_n(1) - [\Sigma_{r0} - \Sigma_{r1}] \frac{dP_n}{d\mu_0}(1) \quad . \end{aligned} \quad (3.24)$$

But, we have the identities [3]:

$$P_n(1) = 1 \quad , \quad 0 \leq n < \infty \quad , \quad (3.25a)$$

$$\frac{dP_n}{d\mu_0}(1) = \frac{n(n+1)}{2} \quad , \quad 0 \leq n < \infty \quad , \quad (3.25b)$$

and the definition:

$$\begin{aligned} \Sigma_{r,tr} &= \Sigma_{r0} - \Sigma_{r1} \\ &= \text{restricted transport cross section} \quad . \end{aligned} \quad (3.26)$$

Then, by introducing Eqs. (3.25) and Eq. (3.26) into Eq. (3.24), we get

$$\Sigma_{rn} \approx \Sigma_{r0} - \Sigma_{r,tr} \frac{n(n+1)}{2} \quad . \quad (3.27)$$

Combining Eqs. (3.21), (3.22), and (3.27), we obtain

$$\begin{aligned} L_r \Psi(\mu) &= \sum_{n=0}^{\infty} \frac{2n+1}{2} \Sigma_{rn} P_n(\mu) \int_{-1}^1 P_n(\mu') \Psi(\mu') d\mu' - \Sigma_{r0} \Psi(\mu) \\ &= \sum_{n=0}^{\infty} \frac{2n+1}{2} \left[ \Sigma_{r0} - \Sigma_{r,tr} \frac{n(n+1)}{2} \right] P_n(\mu) \int_{-1}^1 P_n(\mu') \Psi(\mu') d\mu' - \Sigma_{r0} \Psi(\mu) \\ &= \Sigma_{r0} \left[ \sum_{n=0}^{\infty} \frac{2n+1}{2} P_n(\mu) \int_{-1}^1 P_n(\mu') \Psi(\mu') d\mu' - \Psi(\mu) \right] \\ &\quad + \frac{\Sigma_{r,tr}}{2} \sum_{n=0}^{\infty} \frac{2n+1}{2} [-n(n+1)] P_n(\mu) \int_{-1}^1 P_n(\mu') \Psi(\mu') d\mu' \quad . \end{aligned} \quad (3.28)$$

Now we use two other identities of the Legendre polynomials [3]. First, for any function  $\Psi(\mu)$ ,

$$\begin{aligned} \Psi(\mu) &= \sum_{n=0}^{\infty} \frac{2n+1}{2} \Psi_n P_n(\mu) \quad , \\ \Psi_n &= \int_{-1}^1 P_n(\mu') \Psi(\mu') d\mu' \quad . \end{aligned}$$

Therefore,

$$(i) \quad \Psi(\mu) = \sum_{n=0}^{\infty} \frac{2n+1}{2} P_n(\mu) \int_{-1}^1 P_n(\mu') \Psi(\mu') d\mu' \quad . \quad (3.29)$$

Also,

$$(ii) \quad \frac{\partial}{\partial \mu}(1 - \mu^2) \frac{\partial}{\partial \mu} P_n(\mu) = -n(n+1)P_n(\mu) \quad . \quad (3.30)$$

Using these identities, Eq. (3.28) becomes:

$$\begin{aligned} L_r \Psi(\mu) &\approx \frac{\Sigma_{r,tr}}{2} \sum_{n=0}^{\infty} \frac{2n+1}{2} \left[ \frac{\partial}{\partial \mu}(1 - \mu^2) \frac{\partial}{\partial \mu} P_n(\mu) \right] \int_{-1}^1 P_n(\mu') \Psi(\mu') d\mu' \\ &= \frac{\Sigma_{r,tr}}{2} \left[ \frac{\partial}{\partial \mu}(1 - \mu^2) \frac{\partial}{\partial \mu} \right] \left[ \sum_{n=0}^{\infty} \frac{2n+1}{2} P_n(\mu) \int_{-1}^1 P_n(\mu') \Psi(\mu') d\mu' \right] \\ &= \frac{\Sigma_{r,tr}}{2} \frac{\partial}{\partial \mu}(1 - \mu^2) \frac{\partial}{\partial \mu} \Psi(\mu) \\ &\equiv L_{r,FP} \Psi(\mu) \quad . \end{aligned} \quad (3.31)$$

Thus, Eq. (3.13) becomes the well-known Boltzmann-Fokker-Planck (BFP) equation for  $\Psi$ :

$$\begin{aligned} \mu \frac{\partial \Psi}{\partial x}(x, \mu) + \Sigma_h(x) \Psi(x, \mu) &= \int_{-1}^1 \Sigma_h(x, \mu, \mu') \Psi(x, \mu') d\mu' \\ &+ \frac{\Sigma_{r,tr}(x)}{2} \frac{\partial}{\partial \mu}(1 - \mu^2) \frac{\partial \Psi}{\partial \mu}(x, \mu) \quad , \end{aligned} \quad (3.32)$$

where

$$\begin{aligned} \Sigma_{r,tr}(x) &= \Sigma_{r0}(x) - \Sigma_{r1}(x) \\ &= \int_{-1}^1 (1 - \mu_0) \Sigma_r(x, \mu_0) d\mu_0 \quad . \end{aligned} \quad (3.33)$$

The Fokker-Planck approximation developed here depends only on the fact that  $\Sigma_r(x, \mu_0)$  is strongly peaked near  $\mu_0 = 1$ . This approximation is valid independently of any assumptions about the space-dependence of  $\Sigma_r$  and  $\Sigma_h$ . As can be seen from the derivation, the BFP equation applies to transport problems in which the dominant scattering is highly forward-peaked. The Fokker-Planck approximation eliminates this highly forward-peaked scattering kernel, which makes the resulting BFP equation easier to solve.

Before proceeding, we note from Eqs. (3.33) that

$$\begin{aligned}
\Sigma_{r,tr} &= 2\pi \int_{-1}^1 (1 - \mu_0) \Sigma_r(\mu_0) d\mu_0 \\
&= \Sigma_{r0} - \Sigma_{r1} \\
&= \Sigma_{r0} \left( 1 - \frac{\Sigma_{r1}}{\Sigma_{r0}} \right) \\
&= \Sigma_{r0} \left[ 1 - \frac{\int_{-1}^1 \mu_0 \Sigma_r(\mu_0) d\mu_0}{\int_{-1}^1 \Sigma_r(\mu_0) d\mu_0} \right] \\
&= \Sigma_{r0} [1 - \langle \mu_0 \rangle] \quad ,
\end{aligned}$$

where

$$\begin{aligned}
\langle \mu_0 \rangle &= \frac{\Sigma_{r1}}{\Sigma_{r0}} \\
&= \text{mean scattering cosine} \quad .
\end{aligned} \tag{3.34}$$

Thus,

$$\Sigma_{r,tr} = \Sigma_{r0} [1 - \langle \mu_0 \rangle] \tag{3.35}$$

is, for electrons, much smaller than  $\Sigma_{r0}$ . Hence,

$$\lambda_{r0} = \frac{1}{\Sigma_{r0}} = \text{restricted MFP} \quad , \tag{3.36a}$$

and

$$\lambda_{r,tr} = \frac{1}{\Sigma_{r,tr}} = \text{restricted transport MFP} \quad , \tag{3.36b}$$

satisfy

$$\lambda_{r,tr} = \frac{\lambda_{r0}}{1 - \langle \mu_0 \rangle} \quad , \tag{3.37}$$

and the restricted transport MFP is much greater than the restricted MFP.

### 3.4 The Atomic Mix Approximation to the BFP Equation

For a heterogeneous medium consisting of “chunks” of two materials, we define the three length scales:

$$\frac{1}{\lambda_h} = \text{typical value of } \Sigma_h(x) \quad , \quad (3.38a)$$

$$\frac{1}{\lambda_{r,tr}} = \text{typical value of } \Sigma_{r,tr}(x) \quad , \quad (3.38b)$$

$$\frac{1}{\lambda_{ch}} = \text{typical value of } \Sigma_{ch}(x) \quad , \quad (3.38c)$$

where  $\lambda_h$  is the typical distance an electron will travel between two consecutive hard collisions.  $\lambda_{r,tr}$  is the typical distance an electron will travel for its direction of flight to be altered an  $O(1)$  amount through soft collisions only.  $\lambda_{ch}$  is the typical width of a chunk in such a heterogeneous medium.

We make the following assumption:

$$\lambda_{ch} \ll \lambda_{r,tr} \approx \lambda_h \quad . \quad (3.39)$$

Thus,

$$\frac{\lambda_h}{\lambda_{r,tr}} \approx 1 \quad \text{and} \quad \epsilon \equiv \frac{\lambda_{ch}}{\lambda_{r,tr}} \ll 1 \quad . \quad (3.40)$$

Eqs. (3.39) and (3.40) state that a typical chunk size is small compared to  $\lambda_h$  and  $\lambda_{r,tr}$ , while  $\lambda_h$  and  $\lambda_{r,tr}$  are comparable.

We then introduce a “fast” spatial dimensionless variable  $y$  and three functions in terms of  $y$ :

$$y \equiv \frac{x}{\lambda_{ch}} \quad , \quad (3.41a)$$

$$\sigma_h(y) \equiv \lambda_{r,tr} \Sigma_h(\lambda_{ch}y) = \lambda_{r,tr} \Sigma_h(x) \quad , \quad (3.41b)$$

$$\sigma_h(y, \mu, \mu') \equiv \lambda_{r,tr} \Sigma_h(\lambda_{ch}y, \mu, \mu') = \lambda_{r,tr} \Sigma_h(x, \mu, \mu') \quad , \quad (3.41c)$$

$$\sigma_{r,tr}(y) \equiv \lambda_{r,tr} \Sigma_{r,tr}(\lambda_{ch}y) = \lambda_{r,tr} \Sigma_{r,tr}(x) \quad . \quad (3.41d)$$

Since  $\lambda_h$  and  $\lambda_{r,tr}$  are comparable to each other,  $\sigma_h(y)$ ,  $\sigma_h(y, \mu, \mu')$ , and  $\sigma_{r,tr}(y)$  are dimensionless and  $O(1)$  in magnitude. The BFP equation (Eq. (3.32)) becomes:

$$\begin{aligned} \mu \frac{\partial \Psi}{\partial x}(x, \mu) + \frac{1}{\lambda_{r,tr}} \sigma_h(y) \Psi(x, \mu) &= \frac{1}{\lambda_{r,tr}} \int_{-1}^1 \sigma_h(x, \mu, \mu') \Psi(x, \mu') d\mu' \\ &+ \frac{\sigma_{r,tr}(y)}{2\lambda_{r,tr}} \frac{\partial}{\partial \mu} (1 - \mu^2) \frac{\partial \Psi}{\partial \mu}(x, \mu) \quad . \end{aligned} \quad (3.42)$$

Here the scaled cross sections ( $\sigma_h$  and  $\sigma_{r,tr}$ ) are expressed in terms of the “fast” variable  $y$ , which means that they change by an  $O(1)$  amount over a typical chunk size  $\lambda_{ch}$ .

Similarly, we also introduce a “slow” spatial dimensionless variable  $z$ :

$$z = \frac{x}{\lambda_{r,tr}} \quad , \quad (3.43)$$

which is used to express an  $O(1)$  varying component in  $\Psi$  over a typical distance where the direction of flight has an  $O(1)$  change.

Mathematically, we assume that  $\Psi$  can be expressed in terms of  $y$  and  $z$ :

$$\Psi(x, \mu) = \psi(y, z, \mu) \quad . \quad (3.44)$$

Applying the chain rule, we obtain

$$\frac{\partial \Psi}{\partial x} = \frac{\partial \psi}{\partial y} \frac{1}{\lambda_{ch}} + \frac{\partial \psi}{\partial z} \frac{1}{\lambda_{r,tr}} \quad . \quad (3.45)$$

Introducing Eqs. (3.43) – (3.45) into (3.42), we have:

$$\begin{aligned} \mu \left( \frac{1}{\lambda_{ch}} \frac{\partial \psi}{\partial y} + \frac{1}{\lambda_{r,tr}} \frac{\partial \psi}{\partial z} \right) + \frac{1}{\lambda_{r,tr}} \sigma_h(y) \psi(y, z, \mu) \\ = \frac{1}{\lambda_{r,tr}} \int_{-1}^1 \sigma_h(y, \mu, \mu') \psi(y, z, \mu') d\mu' \\ + \frac{\sigma_{r,tr}(y)}{2\lambda_{r,tr}} \frac{\partial}{\partial \mu} (1 - \mu^2) \frac{\partial \psi}{\partial \mu}(y, z, \mu) \quad . \end{aligned}$$



Multiplying by  $\lambda_{r,tr}$  and using  $\epsilon \equiv \frac{\lambda_{ch}}{\lambda_{r,tr}}$ , we get

$$\begin{aligned} \frac{\mu}{\epsilon} \frac{\partial \psi}{\partial y}(y, z, \mu) + \mu \frac{\partial \psi}{\partial z}(y, z, \mu) + \sigma_h(y) \psi(y, z, \mu) &= \int_{-1}^1 \sigma_h(y, \mu, \mu') \psi(y, z, \mu') d\mu' \\ &+ \frac{\sigma_{r,tr}(y)}{2} \frac{\partial}{\partial \mu} (1 - \mu^2) \frac{\partial \psi}{\partial \mu}(y, z, \mu) \quad . \end{aligned} \quad (3.46)$$

Now we asymptotically expand the solution of (3.46) in terms of  $\epsilon$ :

$$\psi = \psi_0(y, z, \mu) + \epsilon \psi_1(y, z, \mu) + \dots \quad , \quad (3.47)$$

where  $y$  and  $z$  are independent.

By introducing Eq. (3.47) into Eq. (3.46) and equating the  $O(\frac{1}{\epsilon})$  terms, we obtain:

$$\mu \frac{\partial \psi_0}{\partial y}(y, z, \mu) = 0 \quad ,$$

which implies

$$\psi_0(y, z, \mu) = \hat{\psi}_0(z, \mu) \quad , \quad (3.48)$$

where  $\hat{\psi}_0$  is independent of  $y$  but is otherwise arbitrary.

Next the  $O(1)$  terms in Eq. (3.46) give:

$$\begin{aligned} \mu \frac{\partial \psi_1}{\partial y}(y, z, \mu) + \mu \frac{\partial \hat{\psi}_0}{\partial z}(z, \mu) + \sigma_h(y) \hat{\psi}_0(z, \mu) &= \int_{-1}^1 \sigma_h(y, \mu, \mu') \hat{\psi}_0(z, \mu) d\mu' \\ &+ \frac{\sigma_{r,tr}(y)}{2} \frac{\partial}{\partial \mu} (1 - \mu^2) \frac{\partial \hat{\psi}_0}{\partial \mu}(z, \mu) \quad . \end{aligned}$$

Now, we operate on this equation by the spatial averaging operator:

$$\frac{1}{2Y} \int_{-Y}^Y (\cdot) dy$$

and get:

$$\begin{aligned} \frac{\mu}{2Y} [\psi_1(Y, z, \mu) - \psi_1(-Y, z, \mu)] + \mu \frac{\partial \hat{\psi}_0}{\partial z}(z, \mu) + \left[ \frac{1}{2Y} \int_{-Y}^Y \sigma_h(y) dy \right] \hat{\psi}_0(z, \mu) \\ = \int_{-1}^1 \left[ \frac{1}{2Y} \int_{-Y}^Y \sigma_h(y, \mu, \mu') dy \right] \hat{\psi}_0(z, \mu') d\mu' \\ + \frac{1}{2} \left[ \frac{1}{2Y} \int_{-Y}^Y \sigma_{r,tr}(y) dy \right] \frac{\partial}{\partial \mu} (1 - \mu^2) \frac{\partial \hat{\psi}_0}{\partial \mu}(z, \mu) \quad . \end{aligned}$$

Letting  $Y \rightarrow \infty$ , and using

$$\begin{aligned} \psi_1 & \text{ is a bounded [O(1)] function of } y \\ \langle \cdot \rangle &= \lim_{Y \rightarrow \infty} \frac{1}{2Y} \int_{-Y}^Y (\cdot) dy \\ &= \text{averaging operator in } y \end{aligned} \quad (3.49)$$

we obtain the following equation for  $\hat{\psi}_0(z, \mu)$ :

$$\begin{aligned} \mu \frac{\partial \hat{\psi}_0}{\partial z}(z, \mu) + \langle \sigma_h \rangle \hat{\psi}_0(z, \mu) &= \int_{-1}^1 \langle \sigma_h(\mu, \mu') \rangle \hat{\psi}_0(z, \mu') d\mu' \\ &+ \frac{\langle \sigma_{r,tr} \rangle}{2} \frac{\partial}{\partial \mu} (1 - \mu^2) \frac{\partial \hat{\psi}_0}{\partial \mu}(z, \mu) \quad , \end{aligned} \quad (3.50)$$

in which all the cross sections are volume-averaged.

Next, we convert Eq. (3.50) back to the original dimensional dependent and independent variables. From Eqs. (3.44), (3.47), (3.48), and (3.43), we have:

$$\begin{aligned} \Psi(x, \mu) &= \psi(y, z, \mu) \\ &= \psi_0(y, z, \mu) + O(\epsilon) \\ &= \hat{\psi}_0(z, \mu) + O(\epsilon) \\ &= \hat{\psi}_0\left(\frac{x}{\lambda_{r,tr}}, \mu\right) + O(\epsilon) \quad . \end{aligned} \quad (3.51)$$

Let us define

$$\hat{\Psi}(x, \mu) = \hat{\psi}_0(z, \mu) \quad , \quad (3.52)$$

where, by Eq. (3.43),

$$z = \frac{x}{\lambda_{r,tr}} \quad .$$

Then

$$\frac{\partial \hat{\psi}_0}{\partial z} = \frac{\partial \hat{\Psi}}{\partial x} \frac{dx}{dz} = \frac{\partial \hat{\Psi}}{\partial x} \lambda_{r,tr} \quad .$$

Eqs. (3.52) and (3.50) give

$$\begin{aligned} \mu \lambda_{r,tr} \frac{\partial \hat{\Psi}}{\partial x}(x, \mu) + \langle \sigma_h \rangle \hat{\Psi}(x, \mu) &= \int_{-1}^1 \langle \sigma_h(\mu, \mu') \rangle \hat{\Psi}(x, \mu') d\mu' \\ &+ \frac{\langle \sigma_{r,tr} \rangle}{2} \frac{\partial}{\partial \mu} (1 - \mu^2) \frac{\partial \hat{\Psi}}{\partial \mu}(x, \mu) \quad . \end{aligned} \quad (3.53)$$

By using the averaging operator in Eq. (3.49) on Eqs. (3.41), we obtain:

$$\begin{aligned} \langle \sigma_h \rangle &= \lambda_{r,tr} \langle \Sigma_h \rangle \quad , \\ \langle \sigma_h(\mu, \mu') \rangle &= \lambda_{r,tr} \langle \Sigma_h(\mu, \mu') \rangle \quad , \\ \langle \sigma_{r,tr} \rangle &= \lambda_{r,tr} \langle \Sigma_{r,tr} \rangle \quad . \end{aligned} \quad (3.54)$$

Introducing Eqs. (3.54) into (3.53), we obtain:

$$\begin{aligned} \mu \frac{\partial \hat{\Psi}}{\partial x}(x, \mu) + \langle \Sigma_h \rangle \hat{\Psi}(x, \mu) &= \int_{-1}^1 \langle \Sigma_h(\mu, \mu') \rangle \hat{\Psi}(x, \mu') d\mu' \\ &+ \frac{\langle \Sigma_{r,tr} \rangle}{2} \frac{\partial}{\partial \mu} (1 - \mu^2) \frac{\partial \hat{\Psi}}{\partial \mu}(x, \mu) \quad . \end{aligned} \quad (3.55)$$

Thus, we have shown that the solution  $\Psi(x, \mu)$  of the BFP equation (3.32) satisfies

$$\Psi(x, \mu) = \hat{\Psi}(x, \mu) + O(\epsilon) \quad ,$$

where  $\hat{\Psi}(x, \mu)$  satisfies Eq. (3.55). Eq. (3.55) is the atomic mix approximation to the BFP equation (3.32).

To summarize, we have:

(i) Assumed that  $L_r$  has a forward-peaked differential scattering kernel to derive the approximate Fokker-Planck operator  $L_{r,FP}$  to  $L_r$ .

(ii) Assumed that  $\frac{\lambda_h}{\lambda_{r,tr}} = O(1)$  and  $\frac{\lambda_{ch}}{\lambda_{r,tr}} \equiv \epsilon \ll 1$  to derive the atomic mix approximation to the BFP operator.

Under these assumptions, we have shown that the solution  $\Psi(x, \mu)$  of Eqs. (3.13) and (3.14) is well-approximated by  $\hat{\Psi}(x, \mu)$ , the solution of the atomic mix BFP equation (3.55).

Now, if we consider the atomic mix approximation to Eqs. (3.13) and (3.14):

$$\mu \frac{\partial \Psi}{\partial x}(x, \mu) + \langle \Sigma_h \rangle \Psi(x, \mu) = \int_{-1}^1 \langle \Sigma_h(\mu, \mu') \rangle \Psi(x, \mu') d\mu' - \langle L_r \rangle \Psi(x, \mu) \quad , \quad (3.56a)$$

$$\langle L_r \rangle \Psi(x, \mu) = \int_{-1}^1 \langle \Sigma_r(\mu, \mu') \rangle \Psi(x, \mu') d\mu' - \langle \Sigma_r \rangle \Psi(x, \mu) \quad , \quad (3.56b)$$

and if we apply the Fokker-Planck approximation to  $\langle L_r \rangle$

$$\langle L_r \rangle \Psi(x, \mu) \approx -\frac{\langle \Sigma_{r,tr} \rangle}{2} \frac{\partial}{\partial \mu} (1 - \mu^2) \frac{\partial \Psi}{\partial \mu}(x, \mu) \quad , \quad (3.56c)$$

we obtain Eq. (3.55). Therefore, under the assumptions used in this section, the atomic mix transport equation for electrons [Eqs. (3.56)] is a good approximation to the original transport equation for electrons [Eqs. (3.13) and (3.14)]. In these circumstances, the original Boltzmann equation is well-approximated by the atomic mix Boltzmann equation. In particular, Monte Carlo simulations of the original Boltzmann Eqs. (3.13) and (3.14) are well-approximated by Monte Carlo simulations of the atomic mix Boltzmann Equation (Eq. (3.56)).

The asymptotic analysis presented here is based on a simplified 1-D Boltzmann equation with no energy dependence. The work by Larsen and Liang [68] gives a similar but more complicated analysis on a physically realistic 3-D, energy-dependent Boltzmann equation. The results for that analysis are essentially the same as here: if charged particles in a random medium are slowing down over a specified range of energies, and if the sizes of the material “chunks” are small compared to the transport mean free path of the charged particles over this energy range, then the slowing-down of the charged particles in this energy range is well-approximated by the atomic mix Boltzmann equation.

### 3.5 Numerical Results and Discussion

In order to test the asymptotic theory, which states that it is the transport MFP that should be used to determine the validity of homogenizing a heterogeneous medium, we next devise a random “droplet” model (Fig. 3.1) and use the Monte

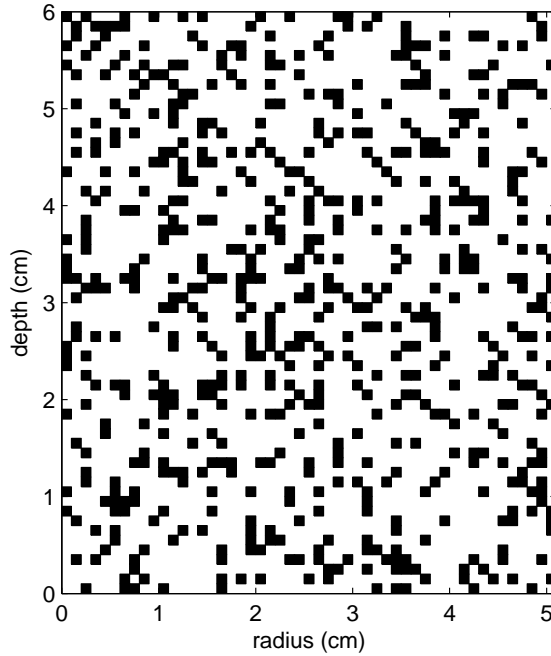


Figure 3.1:  $R$ - $Z$  plane view of the “droplet” model. The dark cells are water and the blank cells are air. The droplet cell size  $ds = 0.1$  cm in this figure.

Carlo method to conduct a series of comparison simulations between this model and its atomic mix counterpart.

#### 3.5.1 “Droplet” model and Monte Carlo simulation

As depicted in Fig. 3.1, this “droplet” model is a cylinder with a radius of 5.1 cm and a depth of 6 cm. It consists of a mesh of small square cells of the same size in  $R$ - $Z$  plane (they are concentric rings in the radial plane). We randomly fill the mesh with water (dark) and air (blank) cells, which combine to form an average density of  $0.201 \text{ g/cm}^3$  (a typical lung’s parenchymal density (see Section 4.2.1)). The side

length  $ds$  of these uniform square cells can be set at a series of decreasing values, which then enable us to explore a hierarchy of “droplet” sizes. We not only change the droplet size, but we also generate various realizations for a specified droplet size.

We use the Monte Carlo code PENELOPE [104] to do the simulations. Two types of monoenergetic particle beams are employed: electrons and photons. The incident electron energy is 2 MeV and the incident photon energy is 3.4 MeV. (The photon energy of 3.4 MeV is selected to generate secondary electrons with a mean energy of about 2 MeV.) The beams are circular, with a radius of 1 cm, and are incident parallel to the axis of the cylinder. The cutoff energies are:  $E_{cut} = 100$  keV for electrons and  $P_{cut} = 10$  keV for photons. The dose tally grid has a size of 0.1 cm in both the radial and axial directions and forms concentric rings throughout the cylinder. Thus, the dose distribution is essentially 2-dimensional in the  $R$ - $Z$  plane. The  $1\sigma$  statistical deviation at the maximum dose  $D_{max}$  is less than 0.1 % for electrons and 0.3 % for photons. The dose is normalized to  $D_{max}$  and in subsequent figures is shown as the percentage of  $D_{max}$ .

### 3.5.2 Dose Distribution for Electrons and Photons

Our asymptotic analysis predicts, essentially, that a heterogeneous medium limits to its atomic mix counterpart when the material chunk size is sufficiently small. Furthermore, this theory predicts that when the typical material chunk size in this heterogeneous medium is small compared to the transport MFP of the particle traveling inside, the results can be accurately represented by that from a homogenized version of this medium. The dose is deposited by electrons through their interactions with matter (the photons, however, deposit the dose indirectly by generating secondary electrons in matter). Therefore Fig. 3.2 (data from PENELOPE), containing the relevant length scales for electrons as a function of energy, is used to estimate

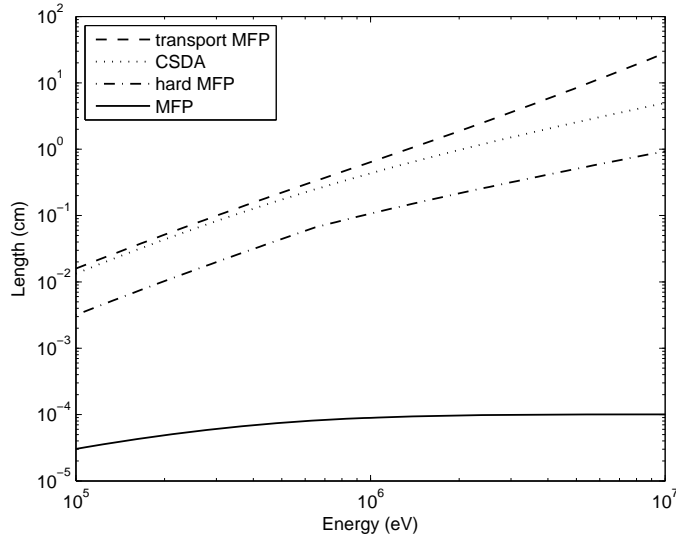


Figure 3.2: The relevant length scales in the electron transport within a range of energy of interest for radiotherapy: (i) transport MFP, the dash line; (ii) CSDA range, the dotted line; (iii) hard MFP, the dash-dotted line; and (iv) MFP, the solid line.

the valid length range of the asymptotic theory. In this figure, the energy ranges from 100 keV, which is the cutoff energy for electrons ( $E_{cut}$ ) in our Monte Carlo simulations, to 10 MeV. This energy range covers most of the energy of interest in radiotherapy. The dashed line is  $\lambda_{tr}(E)$ , the transport MFP. The dash-dotted line is  $\lambda_h(E)$ , the MFP between consecutive hard collisions. The solid line is  $\lambda_r(E)$ , the restricted MFP between consecutive soft collisions. Also included is the dotted line for the *continuous slowing down approximation (CSDA) range*  $R_{CSDA}(E)$ , which can be calculated by Eq. (3.57)

$$R_{CSDA}(E) = \int_0^E \frac{dE'}{S(E')} \quad , \quad (3.57)$$

where  $S(E')$  is the stopping power for electrons as a function of energy.  $R_{CSDA}(E)$  is the range that an electron with energy  $E$  will travel before completely being stopped by soft collisions only.  $R_{CSDA}(E)$  shows the length scale over which an  $O(1)$  amount of change in energy is expected. We can also see from this figure that the transport MFP's are orders of magnitude larger than the MFP's over the whole energy range.

Fig. 3.3 shows the dose distribution for the atomic mix case (a) and the droplet model with one realization at each cell size: (b) at  $ds = 0.1$  cm, (c) at  $ds = 0.05$  cm and (b) at  $ds = 0.01$  cm. It is clear that when the cell size decreases to 0.01 cm, the dose distribution in the heterogeneous medium limits very well to that in the atomic mix case. This is because the typical chunk size  $\lambda_{ch}$  at this cell size approaches the transport MFP,  $\lambda_{r,tr}$ . This result shows the validity of the previous asymptotic analysis. We can also see that in Fig. 3.3d, the difference between the heterogeneous and the atomic mix cases are negligible, except for the 90% and 95% contour lines. All realizations at  $ds = 0.01$  cm level show a similar behavior. Two reasons may explain this discrepancy: (i) the strong electron flux gradients near the boundary; and (ii)  $\lambda_{ch}$  is still not small enough compared to  $\lambda_{r,tr}$ , which can be seen in Fig. 3.2: in the energy range we transport the electrons, the lowest  $\lambda_{r,tr}$  is around 0.01 cm. With a droplet size  $ds = 0.01$  cm, however, the typical chunk size  $\lambda_{ch}$  is about 0.013 cm for water and 0.05 cm for air.

Similar to that from an electron beam (Fig. 3.3), the dose distribution from a photon beam is depicted in Fig. 3.4: the dose distribution for the atomic mix case (a) and the droplet model with one realization at each cell size: (b) at  $ds = 0.1$  cm, (c) at  $ds = 0.05$  cm and (d) at  $ds = 0.01$  cm. The same trend of convergence of the dose distribution to the atomic mix case occurs here, with less significant differences between the heterogeneous and the atomic mix results for case (d). This is due to the coupled transport of both photons and electrons. Two transport equations need to be solved in this photon beam simulation: one for external photon beam and one for the secondary electrons generated by interactions of photons with matter. Since the MFP for photons is on the order of tens of cm in water, which is much larger than even the largest chunk size in our test system, the atomic mix approximation is automatically



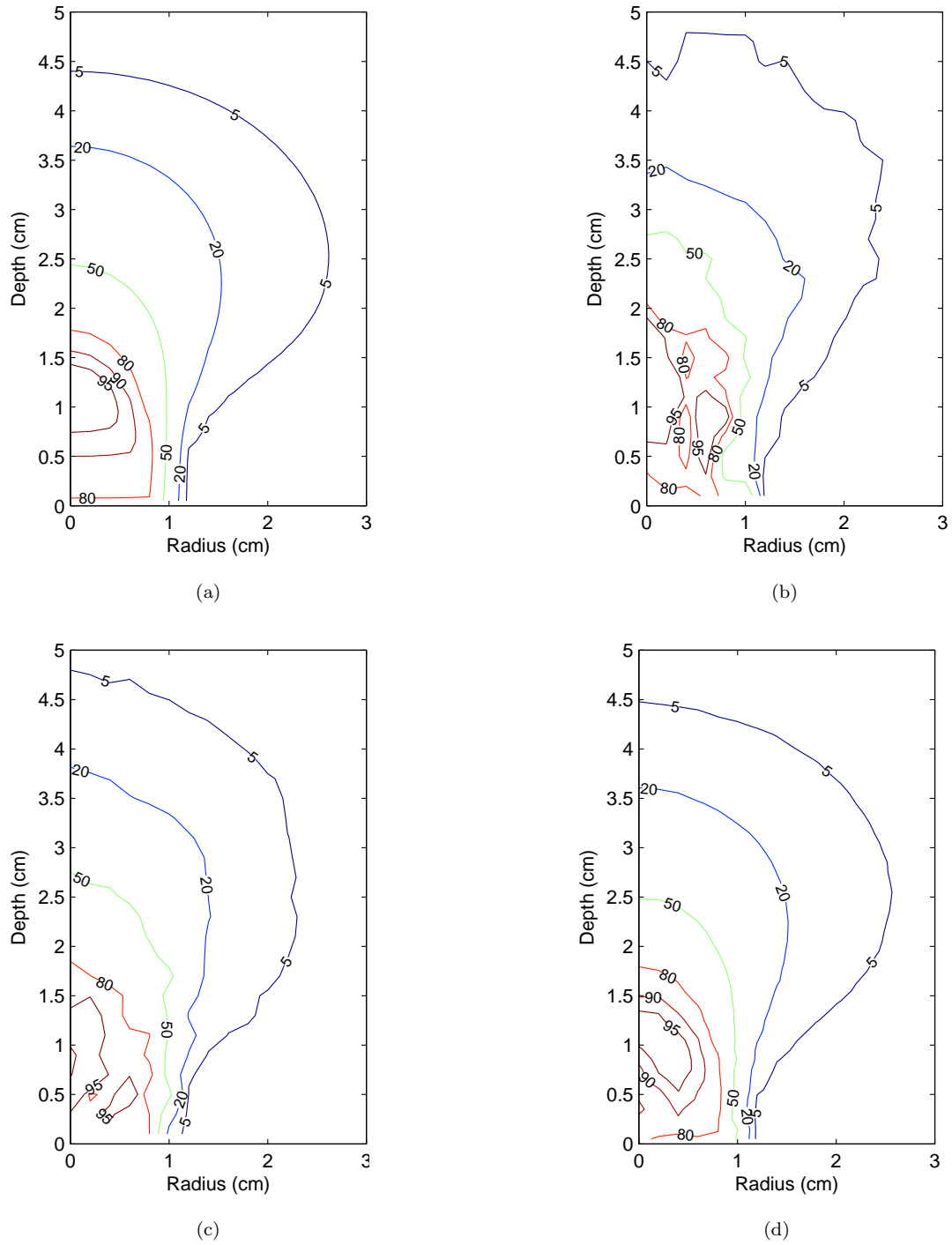
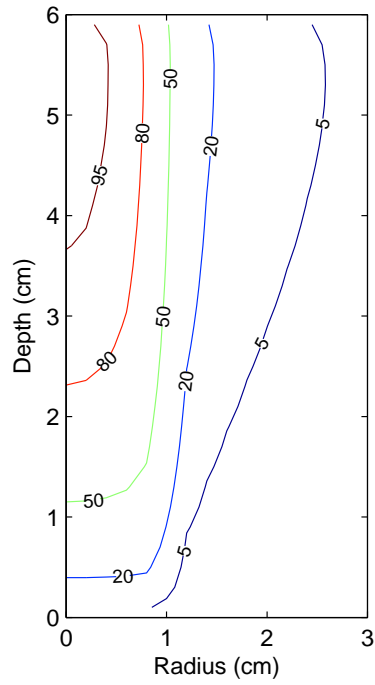
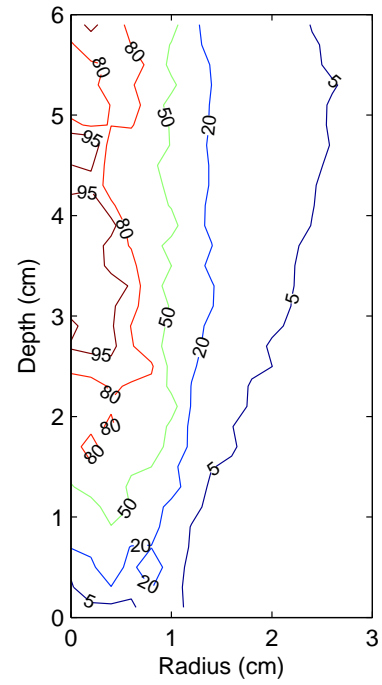


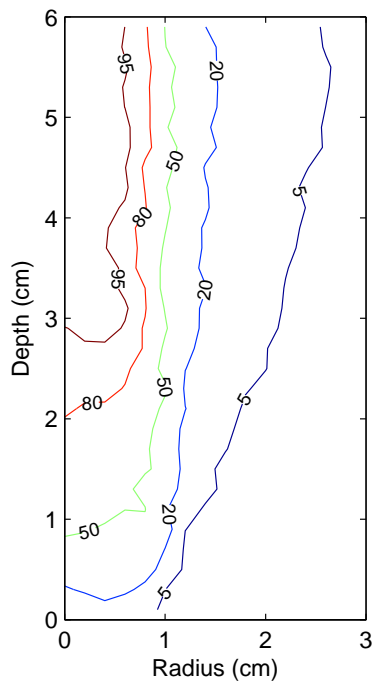
Figure 3.3: Dose contours for 2 MeV electron beam: (a) is for the atomic mix case. The other three are for the droplet model with a cell size at (b) 0.1 cm, (c) 0.05 cm, and (d) 0.01 cm, respectively.



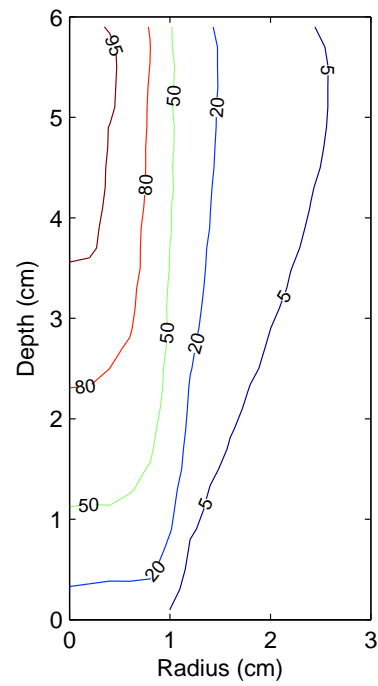
(a)



(b)



(c)



(d)

Figure 3.4: Dose contours for 3.4 MeV photon beam: (a) is for the atomic mix case. The other three are for the droplet model with a cell size at (b) 0.1 cm, (c) 0.05 cm, and (d) 0.01 cm, respectively.

valid for the photon Boltzmann equation. Although now the source term for the electron Boltzmann equation is volumetric inside the medium and different from that of an external beam, the asymptotic analysis remains the same. Thus, it is expected that when the cell size decreases from 0.1 cm to 0.01 cm, the dose distribution limits well to that of the atomic mix case, with some possible difference in the regions with deep flux gradient and/or near the boundary. However, the more uniform source term for electrons helps reduce the significance of such differences.

Fig. 3.3 and 3.4 illustrate the typical results from all the realizations we simulated at different droplet sizes. This shows that for charged particle transport in a heterogeneous medium, when the typical chunk size are small compared to the transport MFP of the charged particles, the atomic mix approximation can replace the original transport equation and accurately predict the results.

### 3.6 Conclusion

We have used a formal asymptotic analysis to show that for transport problems inside a heterogeneous medium involving highly forward-peaked scattering charged particles, when the typical chunk sizes of different materials in the heterogeneous medium are small compared to the transport MFP of the charged particles, the transport equation limits to its atomic mix approximation. In other words, the atomic mix approximation can accurately predict the results of the more complicated transport equations with space-dependent cross sections. The numerical results from Monte Carlo simulations using the PENELOPE code [104] show a very good agreement with the theory. This theory increases the length scale of the chunks for the validity of the atomic mix approximation for electrons by more than two orders of magnitude, from the traditional MFP to the transport MFP.

This asymptotic theory is useful in modeling a heterogeneous system such as the human lung, which consists of a hierarchy of structures ranging in size from 1.0 cm to  $10^{-4}$  cm. According to the traditional atomic mix theory, even the smallest structure in the lung is not small compared to the electron MFP, and hence the atomic mix approximation for electrons should not be assumed to be valid in the lung. However, the theory discussed in this chapter reveals that the atomic mix approximation for electrons is valid for structures which are small compared to the electron transport MFP, and the vast majority of structures in the lung are indeed small compared to the transport MFP. This suggests a practical way to build a model of the heterogeneous lung which will be realistic and accurate for electrons. We develop this model in the next chapter.

## CHAPTER IV

### A 2 1/2-D Random Lung Model

#### 4.1 Introduction

Tissue inhomogeneity corrections are necessary for treatment planning in sites such as the lung [43,77,82,88,89,93,121]. Previous work [14,16,24,40,63,75,83,97,125] modeled the lung as a homogenized mixture of tissue and air, at a lower density than the surrounding tissue, in order to gain understanding of certain inhomogeneity effects between the lung and surrounding tissue. This homogeneous model is also called the *atomic mix* [94] or *mean density* model. However, as shown in Chapter II, the lung is a highly complex organ, consisting of “chunks” of tissue and air ranging in diameter from about  $10^{-4}$  to 1.0 cm, with millions of air-tissue interfaces [31,81], and it is not obvious that the mean density model should be acceptable for treatment planning. In fact, modern treatment planning uses a CT-based patient geometry, in which the voxels are relatively small *local* homogenized volumes with varying densities and compositions. However, the resolution at which one can adequately represent the lung remains an open question. In this chapter, we (i) propose a realistic heterogeneous model of the lung and (ii) present some Monte Carlo (MC) calculations that compare this model to the mean density model and a single voxelized version of the original random lung. We find that in some important situations, dose is not

well-predicted by the mean density or CT models. In Chapter V, we use our random lung model to systematically assess the adequacy of current methods for treatment planning in the lung.

As stated before (Chapter II), the human lung is a spongy, heterogeneous organ consisting of two materials of great density variation: air and tissue. The relative positions and local composition of these two materials are patient-specific and time-dependent, showing a feature of unpredictable “randomness.” In the traditional atomic mix approximation, a heterogeneous particle transport region may be accurately treated by the mean density model if a typical “chunk” size in the region is smaller than the MFPs of the particle [38, 94](also see Chapter III). For electrons, however, we show in Chapter III that instead of the MFPs, the chunk size should be compared to the *transport* MFPs, which are orders of magnitude larger than the MFP. For megavoltage photons with a MFP of tens of cm, the lung’s structure is sufficiently fine to be treated by the mean density model. However, dose deposition is a two-step process: (i) charged particles are generated by interactions between incident photons and irradiated matter; and (ii) these charged particles deposit their kinetic energy along their flight path. The charged particles set in motion by megavoltage photons have a range on the order of centimeters, with a MFP on the order of microns and a transport MFP as low as 0.01 cm in the range of energy relevant to radiotherapy. Under charged particle equilibrium (CPE), the charged particles can be thought to deposit all their energy locally; only the MFPs of photons matter, and the mean density approximation is valid. However, for situations where CPE does not exist, such as within a small beam, or near a beam’s edge or a material interface, it becomes an important consideration that there are still a considerable amount of lung’s structures whose sizes are greater than the charged particles’ trans-

port MFPs. In these conditions, the homogenization in the mean density (either in a single volume or in the CT subvolumes) approximation is no longer guaranteed to be valid, and the “random” lung structure could lead to perturbations in the dose distribution. The actual dose would then deviate from that obtained from the mean density or the CT voxelized lung model, in which every order of lung’s structure is homogenized.

In this chapter we develop a simplified but geometrically sound heterogeneous *random lung model*, based on morphological data of the human lung. We use the Monte Carlo method to perform dose calculations for the “random” and “mean density” lung models, because Monte Carlo is capable of yielding highly accurate dose distributions for generally heterogeneous systems. We also use the Monte Carlo method to compare, in a preliminary simulation, the random lung model and one of its “voxelized” versions. We find that the mean density and voxelized approximations to the random lung model can be inadequate, particularly for small field sizes.

## 4.2 The Lung Model

### 4.2.1 “Random” 2 1/2-D geometry

Due to the extreme geometrical complexity of the lung, it is not practical to build a real lung model down to the smallest order of the hierarchical structures and simulate this model in Monte Carlo calculations. Fortunately, a theoretical part of our work [68] (also see Chapter III) indicates that we can employ a simplified model that (i) retains structures of sizes larger than a threshold size, and (ii) homogenizes all structures finer than the threshold size into a homogeneous mean density *background*. The threshold size should be (i) sufficiently small that in regions with no structures larger than this threshold size, the dose distribution is nearly the same as the dose distribution in the homogenized model, and (ii) as large as possible, to minimize the

complexity of the geometry and the cost of the Monte Carlo simulations.

Furthermore, in our model, we do not duplicate the bronchial/arterial/venous trees in their real 3-D form. Instead, to make our model as simple as possible for MC simulations, yet geometrically sound, a “random” 2 1/2-D model is proposed, which is essentially a 2-D ( $x$ - $y$  plane in the simulation coordinates) geometry extending a finite distance in the third dimension ( $z$ -direction in the simulation coordinates) and cut to fit into the simulated lung region (see Fig. 4.1). Due to the fact that the

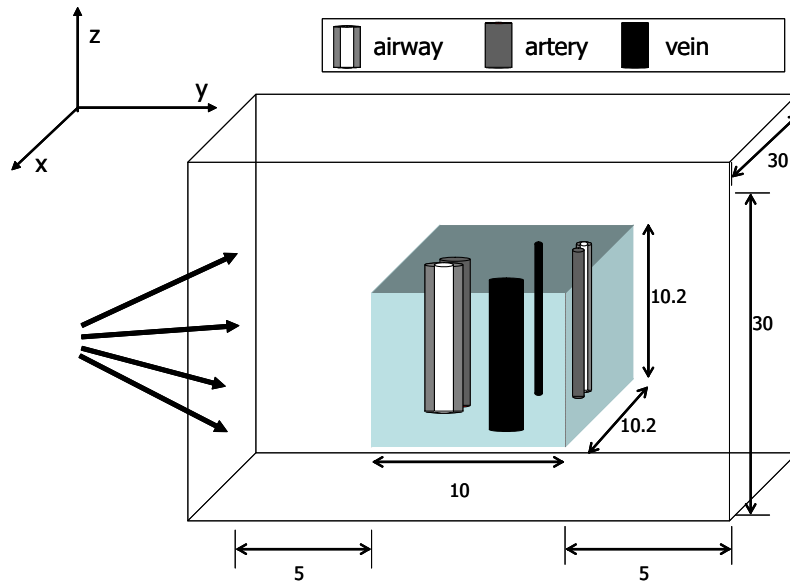


Figure 4.1: Three-dimensional view of the simulated phantom geometry and corresponding dimensions (cm) with the 2 1/2-D lung model embedded inside the surrounding water. The coordinate system is shown in the upper left corner; the photon beams are incident in the positive  $y$  direction and perpendicular to the  $x$ - $z$  plane; the upper legend illustrates modeled airways (concentric cylinders), arteries (single cylinders attached to the airways in the lung) and veins (independent single cylinders in the lung) inside the lung.

airway element has the approximate shape of a hollow cylinder [124], in this model, the “airways” and the “arteries/veins” are modeled as randomly-positioned cylinders with axes parallel to the  $z$ -direction (Fig. 4.1). Each airway consists of two concentric cylinders with the outer cylinder being the wall. The radius of the airway lumen for each order is taken directly from the available morphological data (Table 2.1 and



Table 2.2), and the wall thickness is calculated from the T/D ratio (Section 2.2.2). An artery/vein is taken to be a solid cylinder. Based on the information in Chapter II, an airway is always attached to the artery at a randomly chosen position, while the vein of the same order has a positive distance to the airway/artery bundle. Because of the lack of data on this distance, we assume that the veins of the last seven simulated orders (including all lobar and broncho-pulmonary segment bronchi) should stay in close proximity to the airways/arteries of the same order before entering the broncho-pulmonary segments. In our model, the distance between the center of the vein and the center of the smallest circle containing the airway/artery bundle (*the circumcircle*) is set to double the radius of this circumcircle. The position is randomly selected around the bundle. The remaining smaller orders of veins have no such restriction on position and are uniformly distributed within the model. Arteries and veins are assumed to have the same number of orders as the airways, and to have equal radii, which are computed from the bronchoarterial ratio for the same order. Values computed under this assumption are consistent with the results of a morphological study [55]. Representative top views in the  $x$ - $y$  plane are given in Fig. 4.2.

An important feature of our model is that it conserves the volume ratio of each order of structure, in such a way that the mass for the entire lung (not local regions such as the four particular realizations which are shown in Figs. 4.2 and will also be discussed in Section 4.3.) is the same for both the homogeneous (mean density) and the heterogeneous (random 2 1/2-D) models. To achieve this goal, two quantities need to be computed correctly. One is the adjusted number of structures for each order in the 2 1/2-D geometry. The other is the adjusted mass density for the homogenized part (parenchyma) of the 2 1/2-D model.

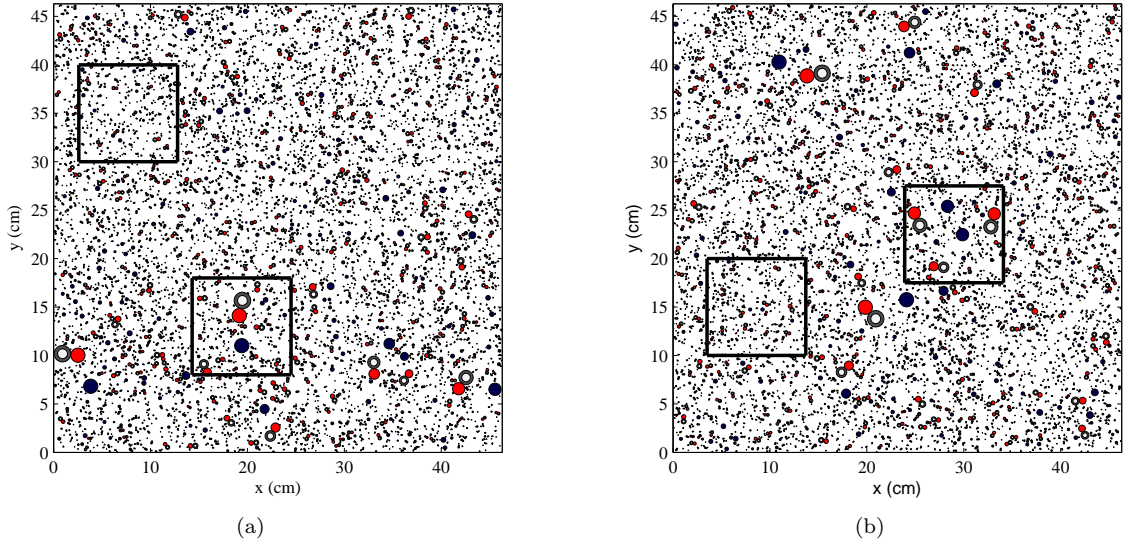


Figure 4.2: Top view of the random heterogeneous lung model in which the concentric circles represent bronchi, the single circles attached to the bronchi are arteries and the independent single circles are veins: (a–b) two realizations of the modeled “whole” lung, with four selected  $10.2 \times 10.2$  cm<sup>2</sup> regions to fit in the lung block in the simulated phantom (see also Fig. 4.1) respectively. (This figure is continued on Page 60)

We used the following equation to map the number  $n$  of the structures in a specific order from a real lung to the number  $m$  of the same order in our 2 1/2-D model, based on the morphometric data in Tables 2.1 and 2.2:

$$\frac{m_{structure}^{model} \cdot a_{structure}}{S_{model}} = \frac{n_{structure}^{lung} \cdot v_{structure}}{V_{lung}} \quad (4.1)$$

Here we have defined:

$m_{structure}^{model}$  = number of structures of a specific order in the model,

$n_{structure}^{lung}$  = number of structures of the same order in the real lung,

$a_{structure}$  = cross-section of the structure,

$v_{structure}$  = volume of the structure,

$S_{model}$  = area chosen to be able to generate all orders of structures,

$V_{lung}$  = volume of a lung.

In Eq. (4.1),  $S_{model}$  is determined in such a way that the smallest calculated  $m$  is greater or equal to 1, i.e., this order of structures must appear at least once in the

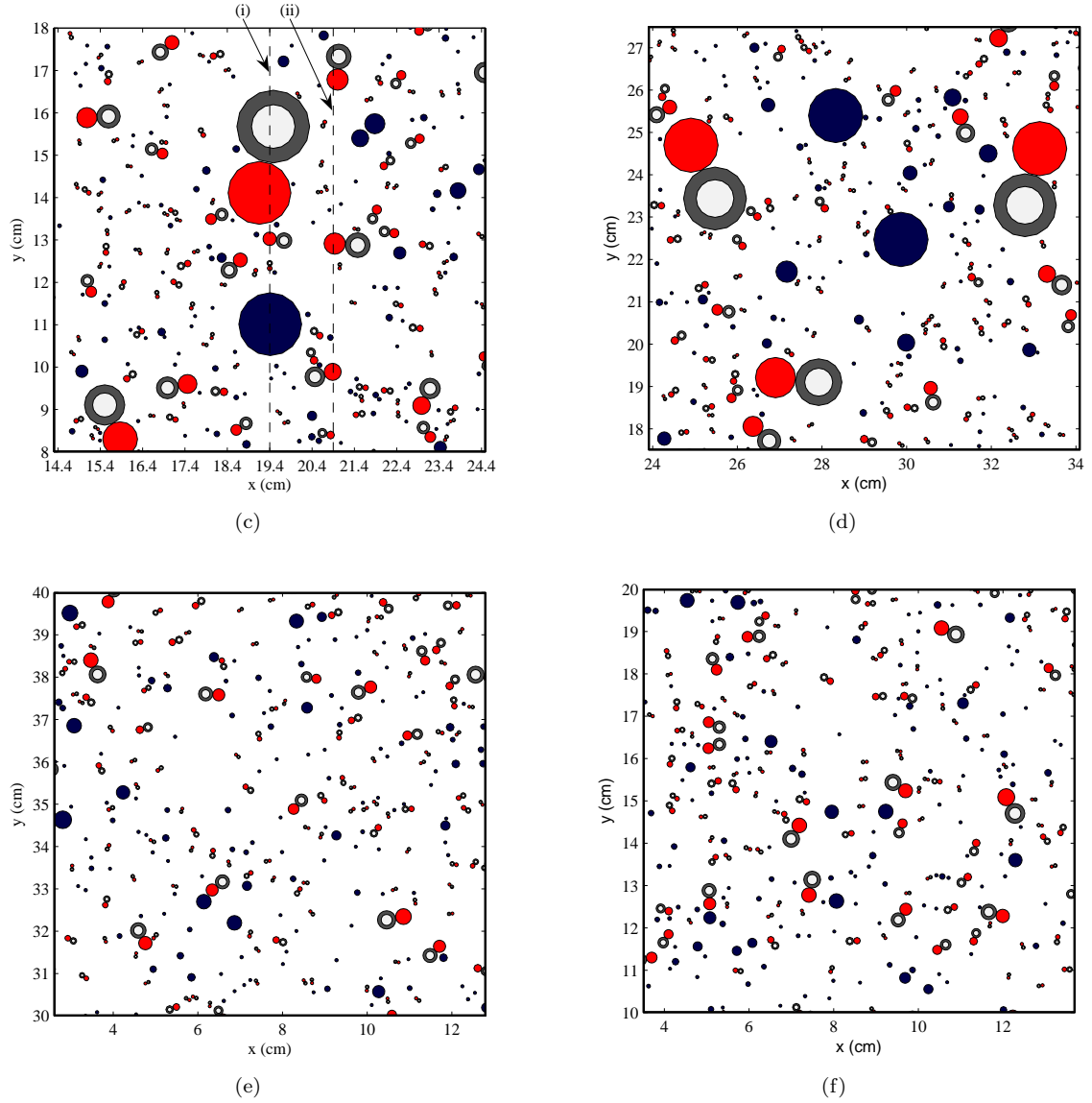


Figure 4.2: (continued from Fig. 4.2 on Page 59) (c) realization  $L_1$ , indicated in (a) by the lower right framed region, with one large bronchus, artery and vein close to the CAX. (Lines (i) and (ii) are used in Fig. 4.3); (d) realization  $L_2$ , indicated in (b) by the right framed region, with two large bronchi, arteries and veins off the CAX; (e) realization  $S_1$ , indicated in (a) by the upper left framed region, with no large structures; (f) realization  $S_2$ , indicated in (b) by the lower left framed region, with no large structures.

model geometry.  $V_{lung}$  is set to a typical 3000 ml in this study. The calculated  $m$ 's are presented in Table 2.1 and 2.2 in which most of the  $m$ 's are not whole numbers. Since the structures in the geometry cannot be fractional, a new  $m$  is recalculated by generating a random number  $\xi$ . If  $\xi < m - [m]$ ,  $m = m + 1$ ; otherwise,  $m = m$ . We note that the last two orders of largest structures are not included in the model because these two are the trachea and the principal bronchus, which are not part of the lung. The minimum value of  $S_{model}$  which allows at least one structure from each order is  $2143 \text{ cm}^2$  ( $46.3 \times 46.3 \text{ cm}^2$ ). We construct “whole lung realizations” using this value of  $S_{model}$  but then select  $10.2 \times 10 \text{ cm}^2$  subregions to represent an actual lung. This is illustrated in Fig. 4.2.

The density of the background homogeneous parenchyma is computed using Eq. (4.2):

$$\rho_{background} = \frac{\rho_{lung} V_{lung} - \sum_{order} n_{order} [\rho_{air} v_{order}^{lumen} + \rho_{water} (v_{order}^{wall} + v_{order}^{artery} + v_{order}^{vein})]}{V_{lung} - \sum_{order} n_{order} [v_{order}^{lumen} + v_{order}^{wall} + v_{order}^{artery} + v_{order}^{vein}]}, \quad (4.2)$$

where  $\rho$  = density, and other parameters are defined in Eq. (4.1).

As stated in Section 2.2, we use a mean density of the lung  $\rho_{lung} = 0.26 \text{ g/cm}^3$ ;  $\rho_{parenchyma}$  is then  $0.201 \text{ g/cm}^3$  accordingly for a threshold set at the terminal bronchiole level (see Section 4.2.2); and a T/D ratio 0.2 and a bronchoarterial ratio 0.695. Specifically, we use four different densities of water as four materials appearing in the geometry: (i) water of density  $1.0 \text{ g/cm}^3$  as the matter of the airway wall, artery and vein, as well as that of the phantom outside the lung; (ii) water of density  $0.26 \text{ g/cm}^3$  as the homogenized (mean density) lung; (iii) water of density  $0.201 \text{ g/cm}^3$  as the lung parenchyma (the background); and (iv) water of density  $0.00120479 \text{ g/cm}^3$  as the air inside the airways. We used only water composition for all different tissues in order to eliminate any factors that might affect the dose calculation other than the

random geometry itself. The line density change in the lung along the  $y$ -direction at two different widths in Fig. 4.2c are depicted in Fig. 4.3. These figures show the major differences in local densities between the mean density and the heterogeneous lung models.

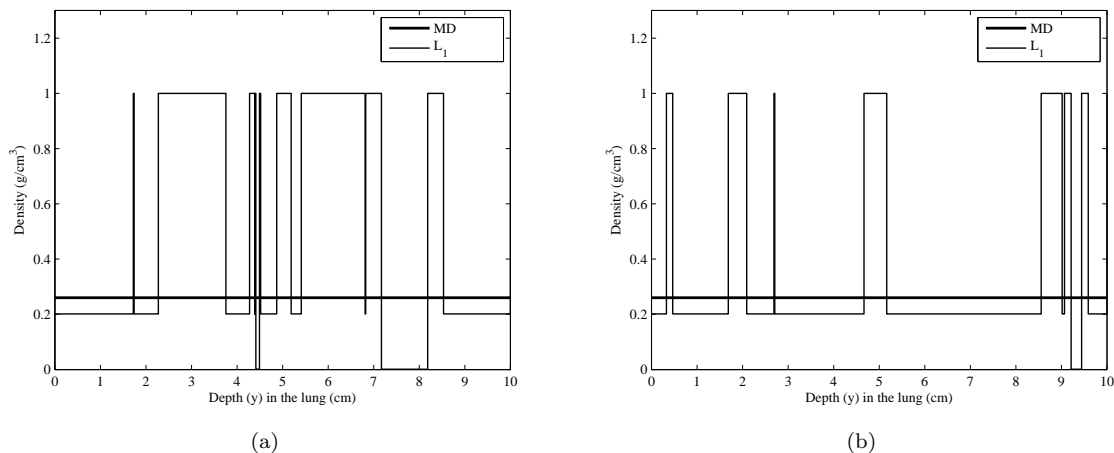


Figure 4.3: Line density change in  $y$ -direction in the lung. The thick lines depict the uniform mean density (MD) case. The thin lines are for the heterogeneous realization  $L_1$  in Fig. 4.2c: (a) corresponds to dashed line (i) and (b) corresponds to dashed line (ii).

#### 4.2.2 The Threshold Size

The classic atomic mix (mean density) approximation states that in a geometrically random system in which the chunk sizes are small compared to a mean free path, one can replace the geometrically random system by the homogenized mean-density system, and the resulting dose will accurately match the dose for the original system [38, 94]. To apply this classic approximation, the threshold size should be on the order of a mean free path for the radiation delivering the dose (the electrons). Unfortunately, the electron mean free path is so small that this would require almost the entire geometrical structure of the lung to be explicitly modeled; doing this would be prohibitively costly.

However, our theoretical work has shown that the atomic mix approximation is

valid for a random system in which the chunk sizes are small compared to a *transport mean free path*  $\lambda_{tr}$ , which is defined as [71, 72]:  $\lambda_{tr} = \frac{\lambda}{1-\langle\mu\rangle}$ , where  $\lambda$  is the mean free path and  $\langle\mu\rangle$  is the mean scattering cosine. This result implies that it is acceptable to choose a threshold size on the order of an electron *transport* mean free path, which because  $\langle\mu\rangle \approx 1$ , can be orders of magnitude greater than an electron mean free path. This extension of the classic atomic mix approximation makes it feasible to construct a practical model of the lung for accurately assessing dose deposited by photon and electron beams.

For the lowest electron energies treated (electron transport cutoff energy = 100 eV, see Section 4.3), the electron transport MFP in water is about 0.02 cm, which is comparable to the size of the lowest order of alveolar ducts (Table 2.2). According our theory, we can select a threshold size at this order of alveolar ducts, and homogenize all structures with lower order numbers to a uniform “background” with an adjusted mean density. However, it is almost impossible to include such a huge number of small structures the Monte Carlo simulation (even in the 2-D form) due to the very slow speed to transport the particles. Alternatively, we choose to select the threshold size at the level of the terminal bronchiole around 0.05 cm, which is comparable to the transport MFP of electrons of energy 200 keV, an electron cutoff energy often used in Monte Carlo simulations. Coincidentally, this part of the lung beyond the terminal bronchioles is the parenchyma (Chapter II). It contains about 90% of the total lung volume (with structures typically 0.01 cm in diameter) and about 70% of the lung mass.

Since we use a fine dose grid (Section 4.3) and thus a low electron cutoff energy at 100 keV, we will show that practically, the threshold selected at the terminal bronchiole level is accurate enough, by comparing the dose distributions in a se-

ries of random lung realizations which contain increasing numbers of orders of lung structures.

### 4.3 Monte Carlo Simulations

The Monte Carlo code PENELOPE [104] is employed for most of the calculations in this study. For the comparison between the atomic mix mean density lung model and the random lung model, we simulate the open field X-ray from a point source, with two clinical photon beam spectra (6 and 18 MV, which are calculated by Sheikh-Bagheri and Rogers [109] for the Varian Clinac) and four field sizes ( $1 \times 1$ ,  $5 \times 5$ ,  $10 \times 10$ , and  $20 \times 20$  cm<sup>2</sup>) at an  $SSD = 100$  cm. For the threshold selection, however, only one beam with a field size of  $2 \times 2$  cm<sup>2</sup> and the energy spectrum of 6 MV is used. The photon fluence at the entrance surface of the phantom is uniform across the field. Cutoff energies of 100 keV for electrons/positrons ( $E_{cut}$ ) and 20 keV for photons ( $P_{cut}$ ) are used throughout. The photon transport is performed with analog Monte Carlo. The electron transport is performed with condensed history method, using step sizes sufficiently small that at least 5 steps are required to transport electrons through each chunk. Dose scoring voxel sizes are 1 mm in the lung region and 2 mm in surrounding water in the  $y$ -direction (beam's direction). In the  $x$ -direction, a 2 mm voxel size is adopted (except for a 1 mm voxel size used for the  $1 \times 1$  cm<sup>2</sup> field size). In the  $z$ -direction (the modeled airway/vessel axes' direction), a 2 mm voxel size is used between -3.1 cm and 3.1 cm, and a 4 mm voxel size is used for the remaining lung region. No variance reduction options are used. For the threshold selection, a uniform dose grid size of  $1 \times 1 \times 2$  cm<sup>3</sup> is applied. The  $1\sigma$  statistical uncertainties at the  $D_{max}$  point along the CAX are  $< 0.5\%$  for all field sizes.

A water phantom of  $30 \times 30 \times 20$  cm<sup>3</sup> with a lung region of  $10.2 \times 10.2 \times 10$  cm<sup>3</sup>

embedded is used for simulations (Fig. 4.1). The front buildup water layer is 5 cm for the 18 MV beams and 3 cm for the 6 MV beams. The lung region extends from 5 to 15 cm for the 18 MV beams and 3 to 13 cm for the 6 MV beams in the  $y$ - direction and from -5.1 cm to 5.1 cm in both the  $x$ - and  $z$ -directions. The surrounding water extends from -15 cm to -5.1 cm and 5.1 cm to 15 cm in both the  $x$ - and  $z$ -directions. For the threshold selection, the phantom setup is similar to Fig. 4.1. However, this phantom and the corresponding embedded lung portion are smaller because the increasing number of bodies as a result of a lower order of structures being selected as the threshold would significantly decrease the calculation speed if we still used the same phantom. Hence a smaller water phantom of  $11 \times 11 \times 11 \text{ cm}^3$  with a lung region of  $5 \times 5 \times 5 \text{ cm}^3$  embedded in the center is used.

In the lung region, we first simulate a homogeneous mean density lung and then four heterogeneous lung realizations representing different parts of a real lung. Fig. 4.2 illustrates how a partial realization is selected from a whole lung realization, as described in Section 4.2.1. Realization large #1 ( $L_1$ ) is depicted in Fig. 4.2c. This is a magnified view of the lower right framed region in Fig. 4.2a. It contains one bronchus, one accompanying artery, and one vein of the same order with diameters larger than 1 cm (in the range of the orders of “large” bronchi, which include main bronchi and lower lobe bronchus), all close to the central axis region. This is intended as a representative situation in which large structures are encountered in the middle of the beam’s pathway. Realization large #2 ( $L_2$ ; Fig. 4.2d, the upper right framed region in Fig. 4.2b) consists of two large bronchi, arteries, and veins and is intended as a representative situation in which large structures occur off the CAX. Realization small #1 ( $S_1$ ; Fig. 4.2e, the upper left framed region in Fig. 4.2a) and small #2 ( $S_2$ ; Fig. 4.2f, the lower left framed region in Fig. 4.2b) are two variants



including only small structures (belonging to the orders of intrasegmental bronchi to terminal bronchioles), which may represent intrasegmental lung regions free of large structures. The results from the four heterogeneous realizations are then compared against the homogenized mean density lung.

To mimic the CT scan, we also “voxelize” this detailed random lung by superimposing a rectilinear grid on it and calculating the mass/density accordingly for each voxel, and homogenizing the material with each spatial cell (voxel). In this way, we obtain a “voxelized” random lung phantom in which the density within each voxel is uniform, but the density generally varies from one voxel to the next. The resulting voxelized random lung model is analogous to the lung models obtained from CT data. Fig. 4.4 shows the voxelized version (top view in the  $x$ - $y$  plane) of Fig. 4.2c at a resolution of  $0.4 \times 0.4 \times 0.4 \text{ cm}^3$ . We used a voxel-based MC code, DPM [108]

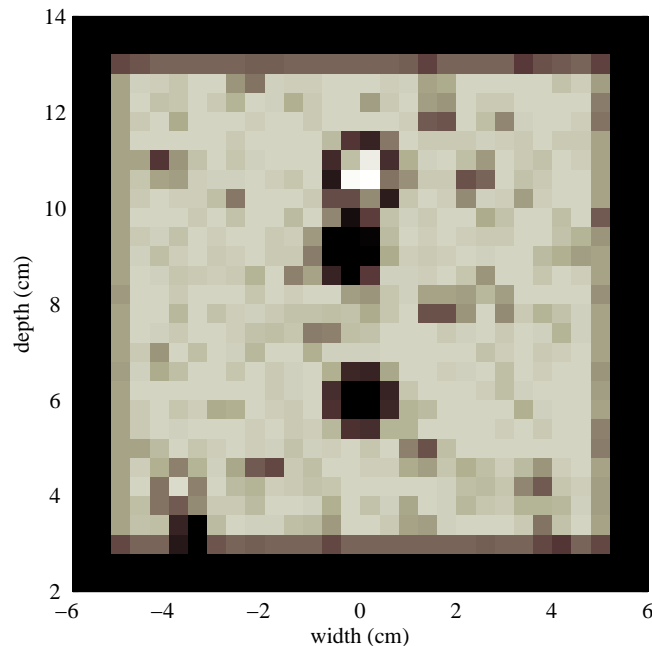


Figure 4.4: Top view of a voxelized version of realization  $L_1$  as shown in Fig. 4.2c to mimic the CT-scan. The resolution is set to be  $0.4 \times 0.4 \times 0.4 \text{ cm}^3$ .

to perform a preliminary calculation for a voxelized version of one detailed random lung realization,  $L_1$ , for the 6 MV  $1 \times 1$  photon beam. DPM has the same cross section libraries as PENELOPE and is optimized for medical physics applications [108]. In our problems, DPM runs about 40 times faster than PENELOPE. For voxelized problems in which the two codes can both be run, they give virtually the same results, and PENELOPE and DPM have both been shown to yield excellent results when compared to experiments [20–22, 108]. The DPM cutoff energies are the same as PENELOPE, while the DPM step sizes for electrons at 0.5 cm above 5 MeV and 0.1 cm otherwise.

On the other hand, we use the same strategy to generate a series of random lung realizations which include three threshold levels at the Horsfield orders of 6, 1, and -3 from Table 2.1 and Table 2.2, to fit in the smaller phantom setup. Fig. 4.5 shows the top views of all the six random lung realizations for the purpose of threshold selection. Figs. 4.5a and 4.5b are a large and a small realization with a threshold level at the

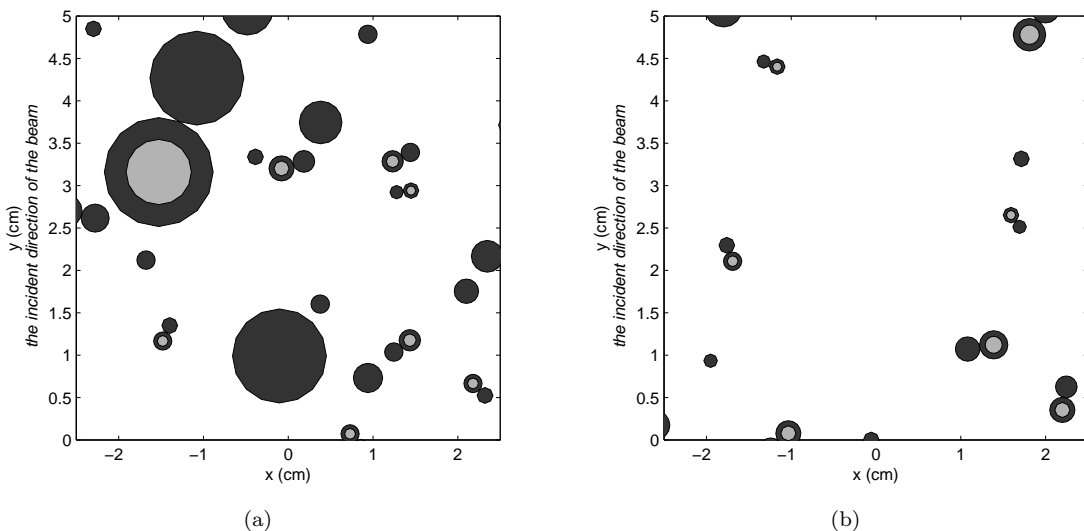


Figure 4.5: Top view of the random heterogeneous lung model used for threshold selection: (a) and (b) are a large and a small realization with a threshold level at the Horsfield order 6, 5 order higher than the terminal bronchiole. (This figure is continued on page 68)

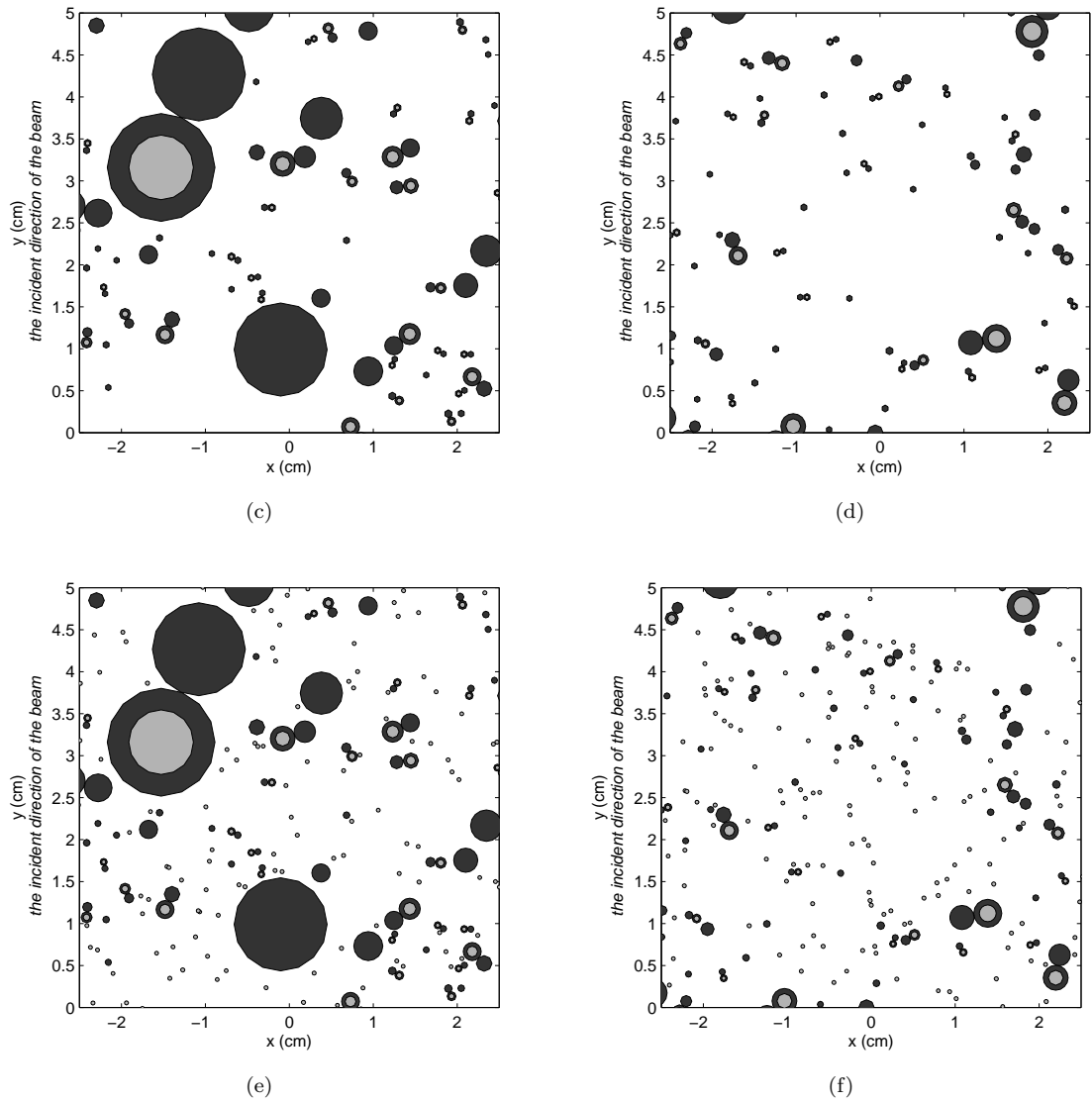


Figure 4.5: (continued from page 67.) (c) and (d) are a large and a small realization with a threshold level at the Horsfield order 1, i.e., the terminal bronchiole. Figs. (e) and (f) are a large and a small realization with a threshold level at the Horsfield order -3, the last respiratory bronchiole level.

Horsfield order 6, 5 order higher than the terminal bronchiole. Figs. 4.5c and 4.5d are a large and a small realization with a threshold level at the Horsfield order 1, i.e., the terminal bronchiole. Figs. 4.5e and 4.5f are a large and a small realization with a threshold level at the Horsfield order -3, the last respiratory bronchiole level. Because the morphology changes in the acinus (Section 2.1.2), only the air cylinders are generated as the respiratory bronchioles' lumen, while the alveolated wall and the blood capillaries are homogenized together with the numerous alveoli into the background. This is different from the method used to build structures with Horsfield orders larger or equal to that of the terminal bronchiole, where the full airway and its accompanying vessels are generated as a bundle. The background density varies among the three levels of realizations according to Eq. (4.2).

#### 4.4 Results and Discussion

We report the simulation results in the forms of the central axis (CAX) percent depth dose (PDD), isodose lines/central dose profiles and the mean lung doses (MLD, which is calculated by dividing the total energy deposited to the lung by the total mass of the lung). All numbers are relative dose normalized to the  $D_{max}$  along the CAX of the mean density case for each field size, respectively. The difference  $\Delta(x, y, z)$  between a certain realization and the mean density model is calculated using  $\Delta(x, y, z) = \frac{D_{hetero}(x,y,z) - D_{mean\ density}(x,y,z)}{D_{max, mean\ density}}$ .

Due the existence of a large number of small structures and the fine scoring voxels used in the MC simulations, we have investigated the validity of using a 100 keV electron cut-off energy. The results show no significant differences between a much lower 10 keV and the 100 keV we adopted. We also show the results for the threshold selection. Besides the main focus on comparisons between the mean density and the

random lungs, we show a comparison between realization  $L_1$  and its voxelized version in terms of the CAX depth dose.

#### 4.4.1 Electron Cutoff Energy

We use a homogeneous atomic mix and a heterogeneous “droplet” realization (see Section 3.5) with a cell size at 0.01 cm for this electron cutoff energy test. All the Monte Carlo simulation parameters are the same as in Section 3.5. The dose grid size is 0.1 cm in both the radial and the axial directions. Fig. 4.6a shows the CAX depth

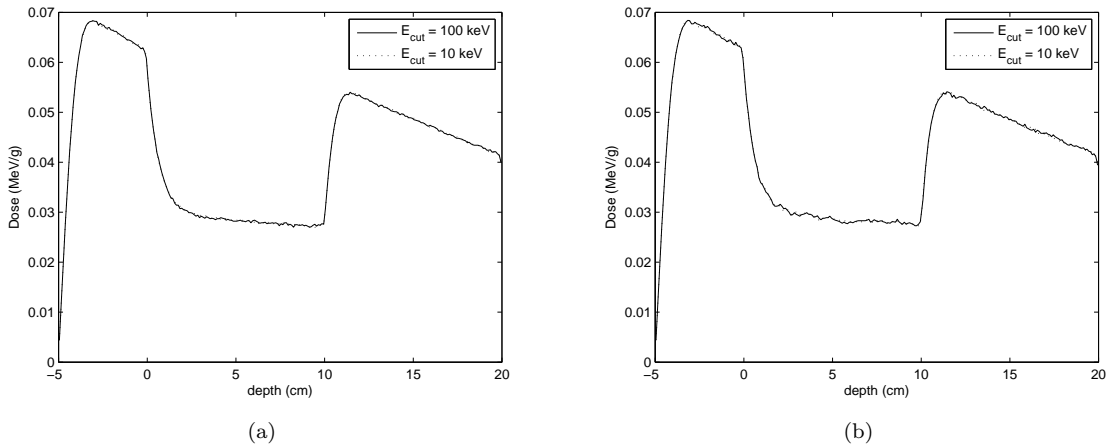


Figure 4.6: The CAX depth dose for (a): the atomic mix model, and (b): the “droplet” model

dose for the atomic mix case and Fig. 4.6b for “droplet” model. In both figures, the solid line is for the 100 keV cutoff and the dotted line for the 10 keV cutoff. The  $1\sigma$  statistical error is less than 0.6% at all depths. Fig. 4.7 shows the percent difference between the two  $E_{cut}$ ’s, which is calculated by normalizing the absolute dose difference by the  $D_{max}$  of the 100 keV case, for both the atomic mix (4.7a) and the “droplet” realization (4.7b). This illustrates that for both the homogeneous and heterogeneous cases, the CAX depth dose is virtually the same when  $E_{cut}$  is reduced from 100 keV to 10 keV, and it indicates that an  $E_{cut} = 100$  keV is adequate for the fine dose tally grid and the complex geometry such as in our random lung model in

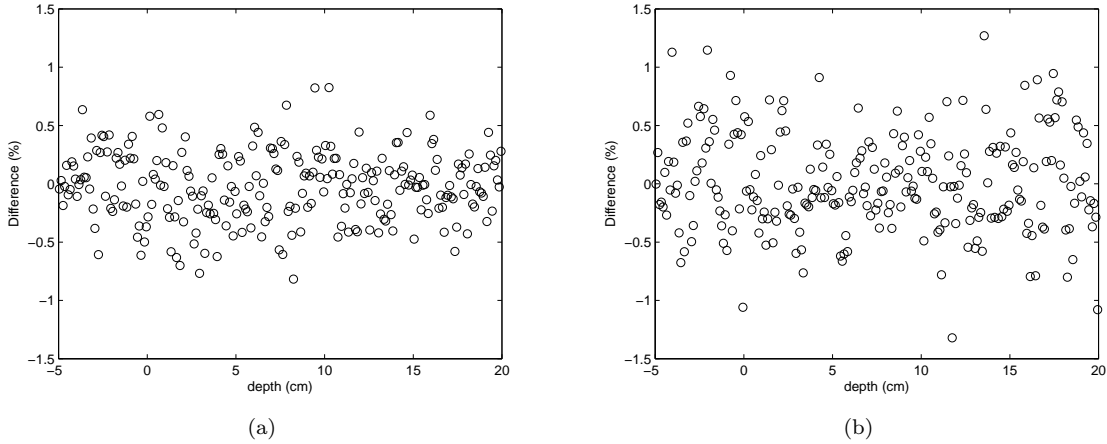


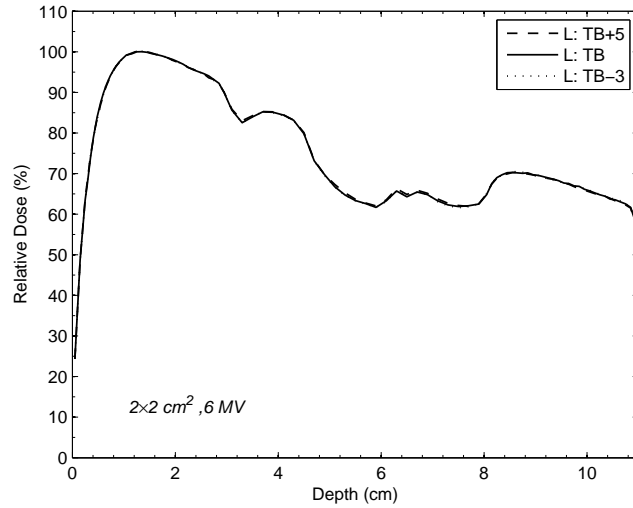
Figure 4.7: The percent difference between the two  $E_{cut}$ 's. It is calculated by normalizing the absolute dose difference by the  $D_{max}$  of the 100 keV cutoff case, for (a): the atomic mix , and (b): the “droplet” realization.

which there are numerous small structures and many boundaries for the electrons to cross.

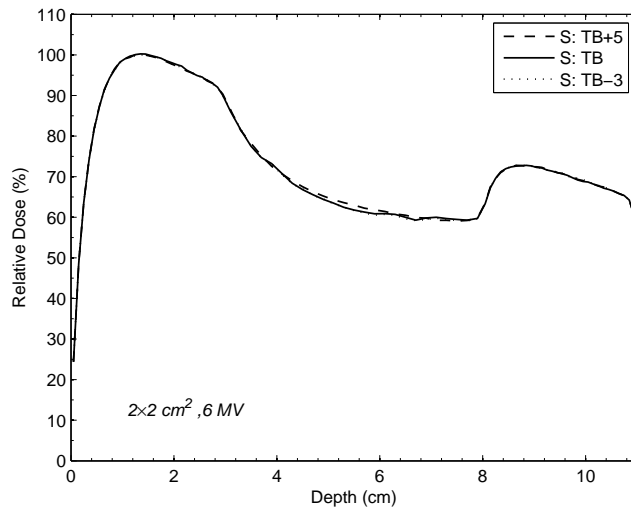
#### 4.4.2 Threshold Selection

Fig. 4.8 gives the normalized CAX PDD for the large and the small realizations for three threshold levels (Fig. 4.5): the terminal bronchiole (TB, the solid lines), the small bronchiole at a level of 5 orders more than the TB (TB+5, the dash lines) and the respiratory bronchiole at a level of 3 orders less than the TB (TB-3, the dotted lines). Fig. 4.8a is for the large realizations and Fig. 4.8b for the small realizations. The same dose distribution in terms of isodose lines is shown in Fig. 4.9 for the large realizations and Fig. 4.10 for the small realizations; in both figures, (a) represents the TB+5 level, (b) represents the TB level and (c) represents the TB-3 level.

There is almost no difference in the CAX PDD between the TB and the TB-3 levels where three more orders of airways are added: the maximum difference is less than 0.5% of  $D_{max}^{100keV}$  for both the large and the small realizations in the lung region. This is reflected in both figures: the dotted lines and the solid lines are essentially

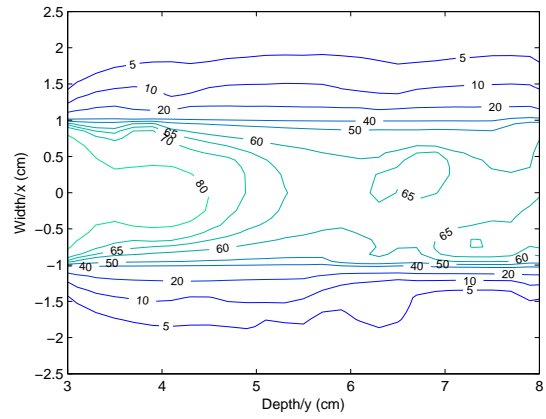


(a)

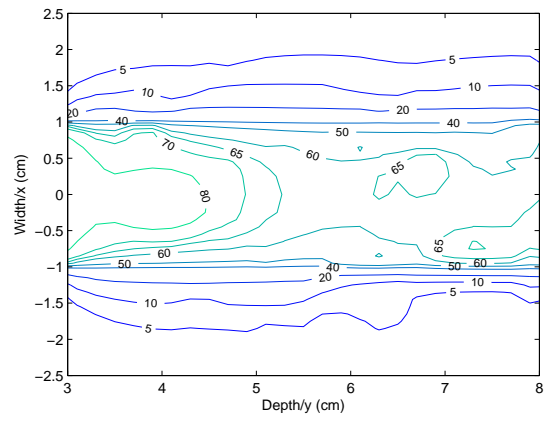


(b)

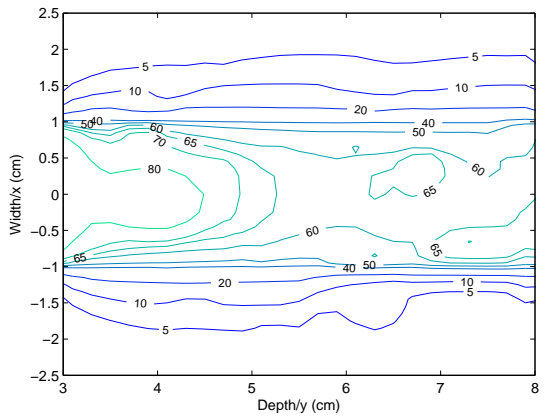
Figure 4.8: The normalized CAX PDD for the large and the small realizations for three threshold levels: the terminal bronchiole (TB, the solid lines), the small bronchiole at a level of 5 orders more than the TB (TB+5, the dash lines) and the respiratory bronchiole at a level of 3 orders less than the TB (TB-3, the dotted lines). (a) is for the large realizations and (b) is for the small ones. In both figures the solid lines and the dotted lines are essentially on top of each other.



(a)



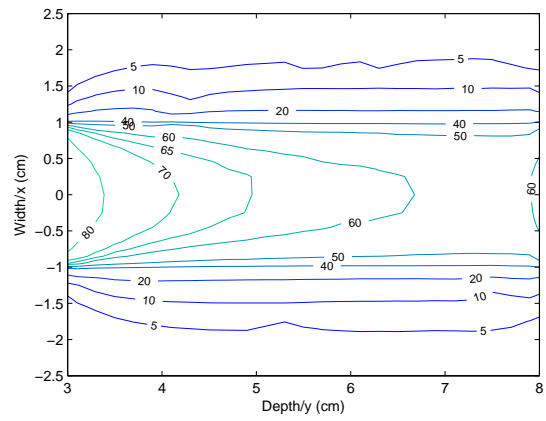
(b)



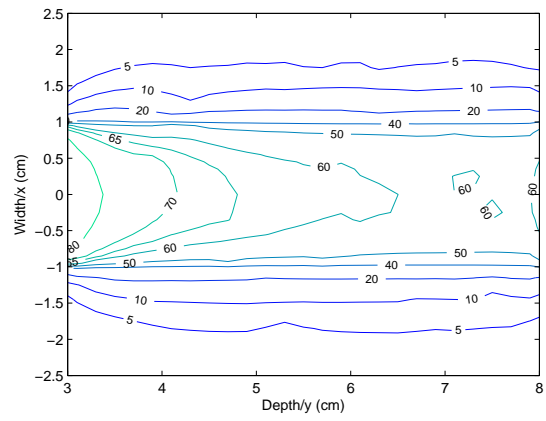
(c)

Figure 4.9: Isodose lines for large realizations: (a) represents the TB+5 level, (b) the TB level and (c) the TB-3 level.

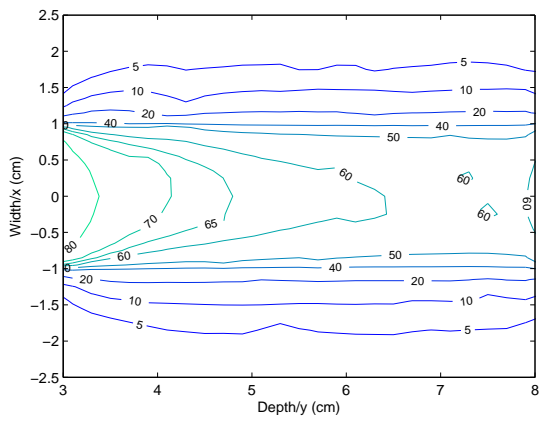




(a)



(b)



(c)

Figure 4.10: Isodose lines for small realizations: (a) represents the TB+5 level, (b) the TB level and (c) the TB-3 level.

the same. However, if we try to reduce the number of structures in the lung model by increasing the Horsfield order from the TB to TB+5, we see noticeable differences, especially in the small realizations, where the maximum difference is greater than 1.2% of  $D_{max}^{100keV}$ . Also noticed is the consistency that the dose in the TB case is lower than the TB-5 case, again, especially in the small realization (although it is difficult to tell from the figures because the difference is small). The reason for this partly lies in Table 4.1. For a small field size such as the  $2 \times 2$  beam with such a high energy, which is used in this threshold selection, charged particle equilibrium does not exist along the CAX. The dose deposited then depends on the density of the interaction sites in a complicated way. In principle, the less that CPE exists, the more heavily the dose can change with the density. For the TB+5 level, the background density is 7.8% larger than that of the TB level because when going up from the terminal bronchiole to the stem, more mass in the blood vessels and the bronchiole wall is homogenized into the background. Therefore, an increase in the dose compared to the TB level is expected. However, when going down from the terminal bronchiole to the periphery, due to different lung's morphology beyond the terminal bronchiole (see Section 4.3) and the small sizes of the structures compared to the dose scoring grid, although there is a slightly increasing density (1.07% larger), the low-density airways within each dose grid partially suppress this increase. As a result, the average density changes very slightly, so that the dose difference can be ignored. Such a trend can also be seen in Fig. 4.9 and Fig. 4.10, which are the isodose lines on the  $x$ - $y$  plane at  $z = 0$ . While there is negligible difference in the large realizations, in the small realizations, the 60% isodose line exhibits a more distorted shape in the TB and the TB-3 levels than in the TB+5 level. Furthermore, two 60% islands can be detected in this narrow beam's pathway in the TB and the

Table 4.1: Background density

	TB+5	TB	TB-3
$\rho$ ( $g/cm^3$ )	0.217186	0.201388	0.203552

TB-3 levels, while there are no such islands in the TB+5 level because the structures causing such perturbations are homogenized in the TB+5 level.

Based on the results and the discussion here, a threshold size of 0.05 cm, selected at the terminal bronchiole order, is suitable for our purpose: adding more lower orders of structures gains little extra accuracy but dramatically increases the Monte Carlo simulation time; while reducing some orders of structures up from the terminal bronchiole results in a noticeable difference in the dose distribution. Henceforth, our random lung model will use this 0.05 cm threshold.

#### 4.4.3 CAX Depth Dose

Fig. 4.11 shows the CAX percent depth doses for the 6 and 18 MV photon beams and the  $1 \times 1$ ,  $5 \times 5$ ,  $10 \times 10$  and  $20 \times 20$  cm<sup>2</sup> field sizes, respectively. When the lung is represented by the mean density model, for both energies, the basic shape of the CAX curves are well known: (i) the builddown region upon entering the lung, which is due to a longer secondary electron range in the low-density lung and the loss of charged particle equilibrium (CPE), along with a reduced photon scattering in the low-density medium; and (ii) the buildup region distal to the lung, which is due to the shorter range and the recovery of lateral CPE. These two phenomena become less pronounced and finally disappear as the field size increases because CPE is gradually recovered in the CAX region. The situation for higher energy beams with the same field size is enhanced because the range of the secondary electrons is longer, and thus more volume is needed for compensation. When the mean density lung is replaced by one of the four heterogeneous realizations simulated in this study,

deviations of different magnitudes occur, depending on conditions such as whether a large structure is in the beam's path, the size and location of the structure, the material components of the structure, the field size, and the beam's energy.

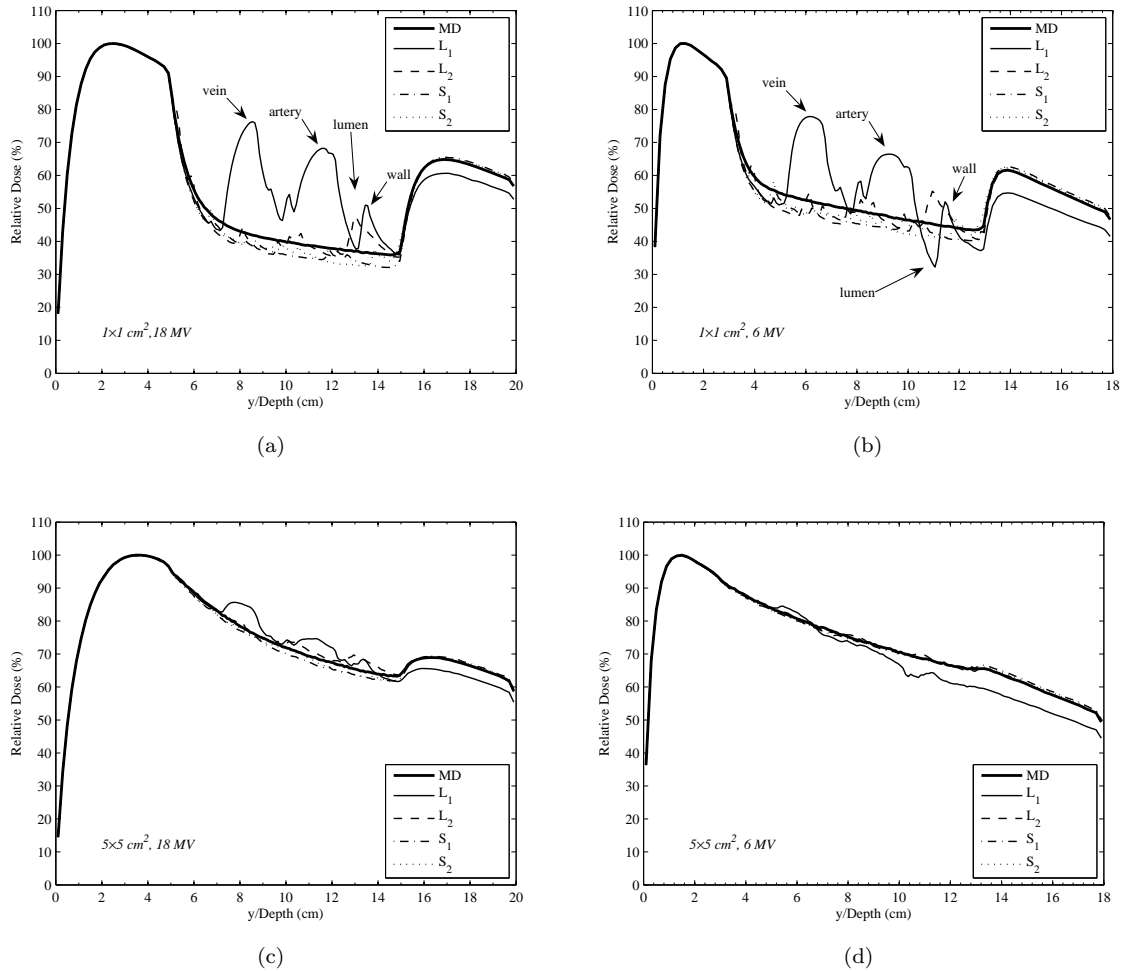


Figure 4.11: The CAX percent depth doses for:  $1 \times 1 \text{ cm}^2$  field size and for (a) 18 MV, (b) 6 MV beams;  $5 \times 5 \text{ cm}^2$  field size and for (c) 18 MV, (d) 6 MV beams. (to be continued on page 78)

### $1 \times 1$ field size

The most significant perturbations for the  $1 \times 1 \text{ cm}^2$  field size come from the three large structures in realization L<sub>1</sub>. The dose percent differences in non-air region are as high as 34% and 26% for the 18 and the 6 MV beam, respectively. The buildup and builddown regions within the large structures are clearly visible in Figs. 4.11a

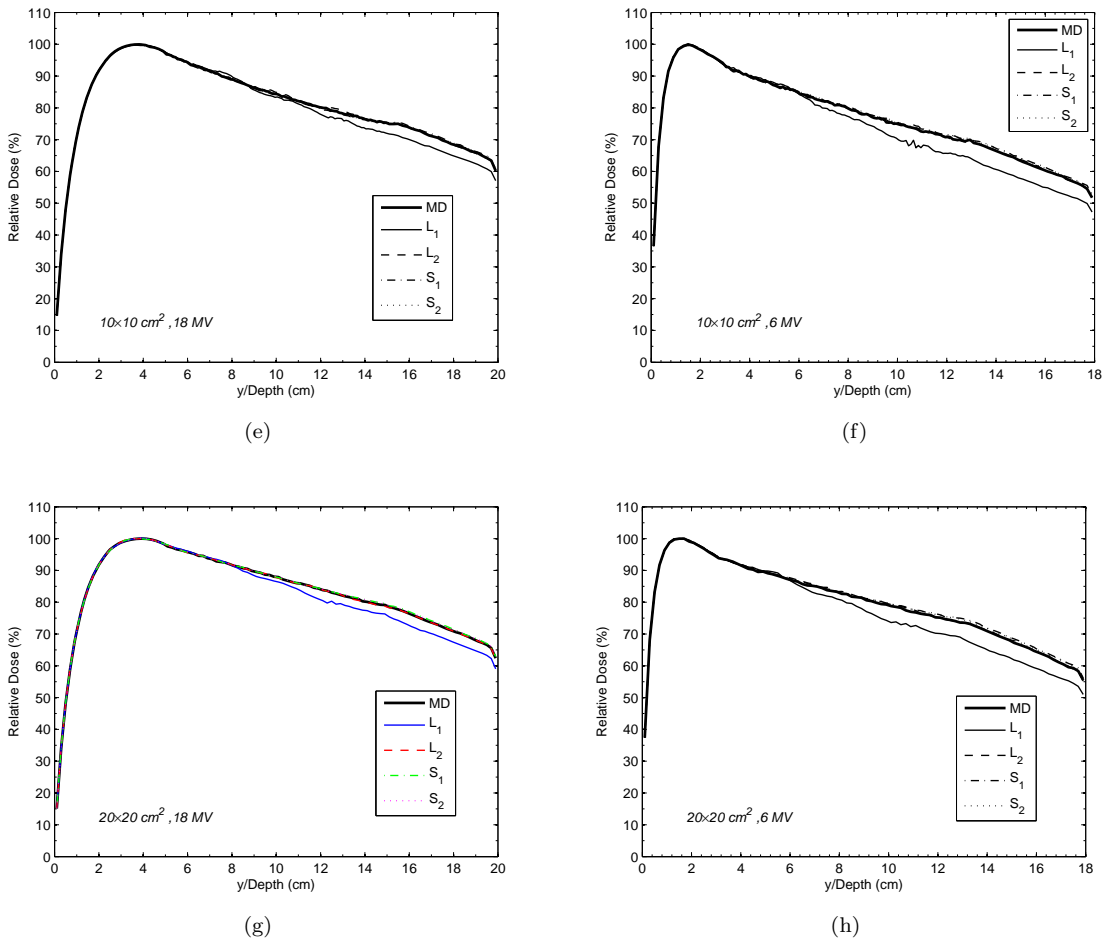


Figure 4.11: (cont'd from page 77.) The CAX percent depth doses for:  $10 \times 10 \text{ cm}^2$  field size and for (e) 18 MV, (f) 6 MV beams;  $20 \times 20 \text{ cm}^2$  field size and for (g) 18 MV, (h) 6 MV beams. The thick solid lines are for the mean density (MD) lung model. The thin solid and the dashed lines are for two large realizations, respectively. The dash-dot and the dotted lines are for two small realizations. All curves for the same field size are normalized to the  $D_{max}$  along the CAX of the corresponding mean density (MD) case. Also indicated in (a) and (b) are the large structure locations (an airway, an artery and a vein) on the CAX for realization L<sub>1</sub>.

and 4.11b. The difference is greater in the 18 MV than in the 6 MV beam, due to the longer electron range, causing an enhanced loss of lateral CPE for higher energy photons. The situation is just the opposite with regard to the dose deposited in the airway lumen. In Fig. 4.11b for the 6 MV beam, the airway lumen (air inside) is identified by the lowest dose “valley” (13% lower than the mean density model), which is less significant in Fig. 4.11a for the 18 MV beam. This is mainly a result of upstream photon scattering, since few secondary electrons originate from within the airway lumen. The same explanation applies to the region between the vein and artery, and the region after the airway, which are mostly occupied by the homogenized background tissue. For realization  $L_2$  (with large structures mostly outside the beam and only two large veins partly sliced by the beam) and the other two small realizations, smaller perturbations are observed in the beam. For the  $1 \times 1$  field size, the difference between the density of the background tissue and that of the mean density lung can cause a significant change in calculated dose. Within the beam, the dose in the small realizations is lower ( $<4.5\%$  and  $<5.4\%$  for the 18 and 6 MV beam) than the mean density case due to the lower density. As a consequence of the extra attenuation of the primary photons caused by large structures (increased local density) inside the beam, and the fact that the dose is dominated by electrons from primary photons, a “shadow” region with reduced dose in the water block distal to the lung appears. This is the case in realization  $L_1$ , where the percent difference is 4.4% for the 18 MV beam and 7% for the 6 MV beam. For the other realizations, with most regions inside the beam being low-density background (and hence lower dose), the dose in the distal water region is slightly higher ( $<1\%$  for the 18 MV and  $<1.5\%$  for the 6 MV) than that of the mean density lung. The softer spectrum of the 6 MV beam accounts for the larger differences versus the 18 MV beam.

### Larger field sizes

Figs. 4.11c through 4.11h show that with increasing field size, dose perturbations decrease, becoming 7% and 2% at the large structures for the  $5 \times 5$  cm field size and for the 18 and 6 MV beams respectively. When the field size exceeds  $10 \times 10$  cm<sup>2</sup>, the differences become  $<1.3\%$  for the  $10 \times 10$  cm<sup>2</sup>, and even smaller for the  $20 \times 20$  cm<sup>2</sup> field size. This is because for the same local density variation inside the beam, the increasing field size leads to gradual recovery of CPE. A similar trend occurs with the “shadow” region behind the lung with realization  $L_1$ . The percent differences in the region distal to the lung for the 18 and 6 MV are  $<3.9\%$  and  $<6.2\%$  for the  $5 \times 5$ ,  $<3.8\%$  and  $<6\%$  for the  $10 \times 10$  and  $<3.8\%$  and  $<5.9\%$  for the  $20 \times 20$  cm<sup>2</sup> beam, respectively. However, these changes as a function of field size are less than those within the large structures. This indicates that even though CPE exists in the CAX and local perturbations are negligible, the accumulated attenuation by the upstream structures is still present. The magnitude of the differences is not sensitive to the field size but is mainly determined by the structures in the beam’s pathway. For the cases in which the tumor is on the distal side of a large structure in the beam’s pathway, simply increasing the field size may not be an effective way to increase the dose to the tumor. For the two small realizations without significant large local density variations, the differences in the lung from the mean density model are small, even for the  $5 \times 5$  cm<sup>2</sup> field size ( $< 2.3\%$  for the 18 MV beam and  $< 1\%$  for the 6 MV).

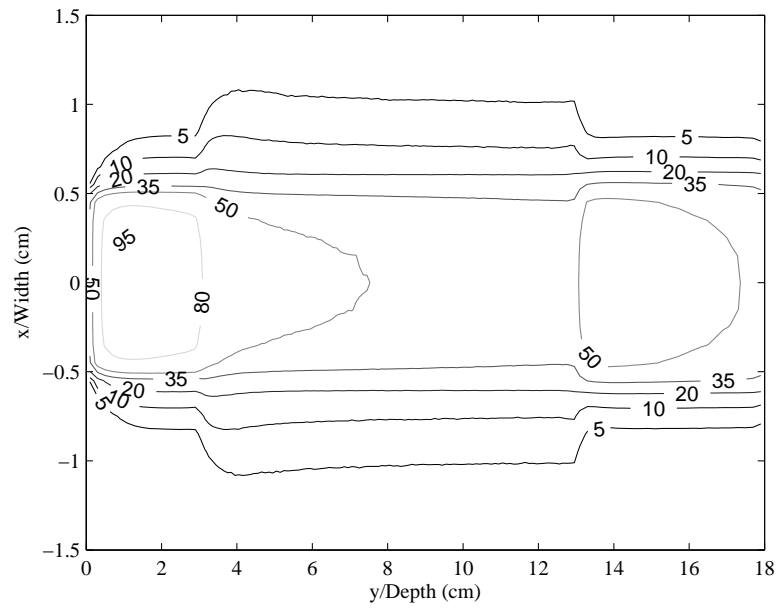
#### 4.4.4 Isodose Lines and Profiles

##### $1 \times 1$ field size

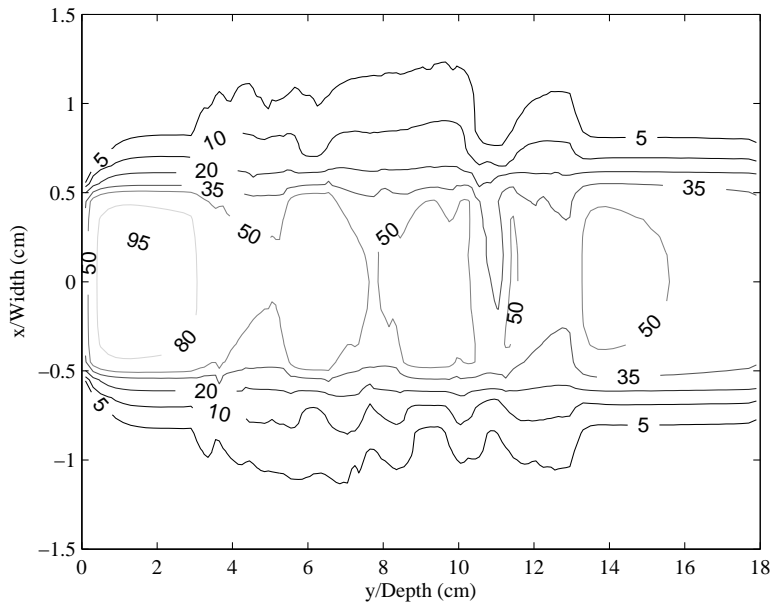
Fig. 4.12 shows the isodose lines for the mean density model and realization  $L_1$  on the  $x$ - $y$  plane at  $z = 0$  for the  $1 \times 1$  cm<sup>2</sup> field size and for the 6 MV energy. Two central dose profiles at selected depths are also shown in Fig. 4.13. The selected depths are (i) 3.1 cm depth in the lung (i.e.,  $y=6.1$  cm), crossing the large vein; (ii) 7.9 cm deep in the lung (i.e.,  $y=10.9$  cm), crossing the airway lumen. The purpose of presenting isodose lines and central dose profiles together is to provide a more complete picture of the perturbations to the dose distribution caused by the structures in the lung, while simultaneously giving typical depth information.

The deviation from the mean density model is that the smoothness of the isodose lines is altered, due to local density variations from randomly positioned structures. Apart from the mean density model, large solid structures inside the beam attenuate more primary photons and become additional local secondary particle “sources”; while at regions free of these structures, lower dose occurs due to the lower density of the background. These result in either the broadening or contracting of the penumbra region, as is clearly indicated by the 10% and 5% isodose lines in Fig. 4.12b. A similar result is also recognized with the 18 MV beam. Also, a structure can increase or decrease the local dose, depending on whether it is tissue or air, with the extent of distortion depending on the size and location of the structure. Compared to the mean density model, two hot spots are present in Fig. 4.12b for realization  $L_1$  (similar hot spots appear for realization  $L_2$ , as well). This is also the case for the 18 MV beam. These can also be seen in Fig. 4.13a, which shows at 3.1 cm depth in the lung, the large vein is almost centered on the CAX; therefore a nearly symmetric profile for realization  $L_1$  occurs. In the profile for realization  $L_2$  in Fig. 4.13a, the



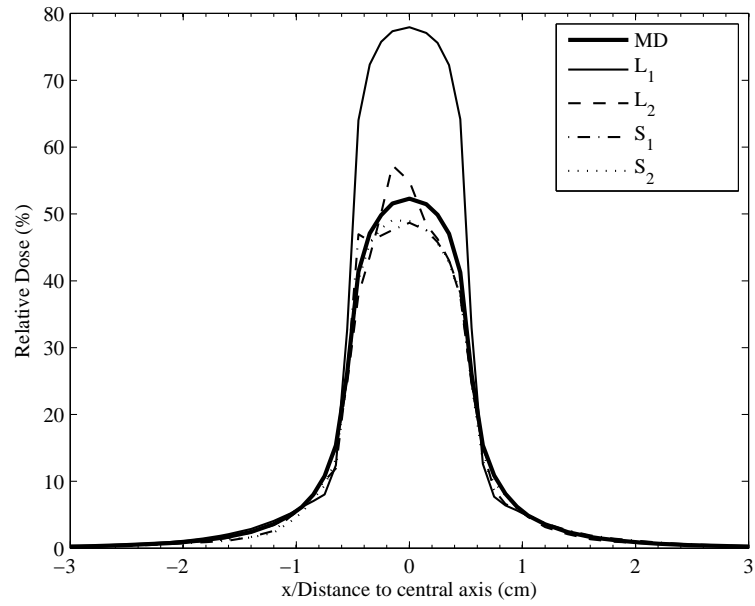


(a)

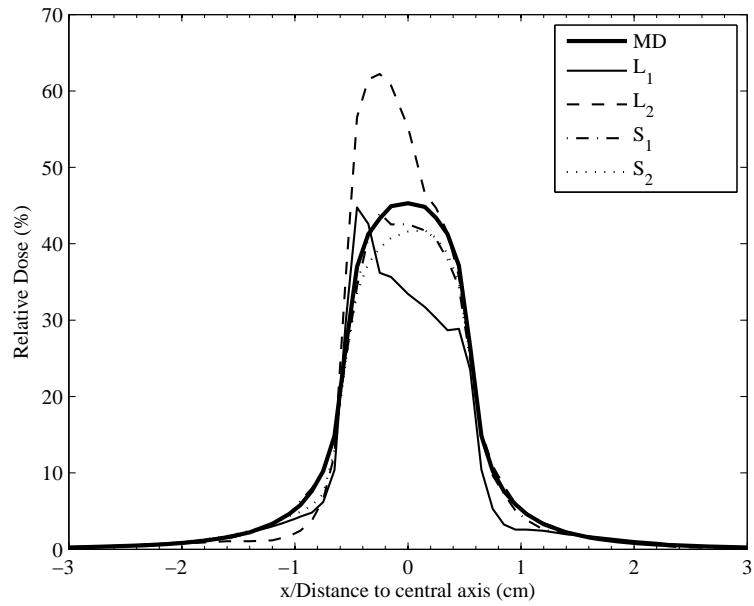


(b)

Figure 4.12: Isodose lines on  $x$ - $y$  plane ( $z = 0$ ) for  $1 \times 1$  cm<sup>2</sup> field size for (a) mean density model, 6 MV; (b) realization  $L_1$ , 6 MV. The abscissa is in  $y$ -direction and the ordinate is in  $x$ -direction.



(a)



(b)

Figure 4.13: CAX dose profiles for  $1 \times 1 \text{ cm}^2$  field size for 6 MV at (a)  $y=6.1 \text{ cm}$ ; (b)  $y=10.9 \text{ cm}$ . The thick solid lines are for the mean density (MD) lung model. The thin solid and the dashed lines are for two large realizations, respectively. The dash-dot and the dotted lines are for two small realizations. All curves for the same field size are normalized to the  $D_{max}$  along the CAX of the corresponding mean density (MD) case.

peak corresponds to a small vein with diameter about 0.22 cm, located at about  $x = -0.125$  cm and totally inside the beam. For the small  $1 \times 1$  field size, CPE does not exist inside the beam, so even a small structure perturbs the dose significantly. The same situation applies to the profiles at depth 7.9 cm deep in the lung (Fig. 4.13b, where the large airway's wall and lumen in realization  $L_1$  are indicated with the clear asymmetry).

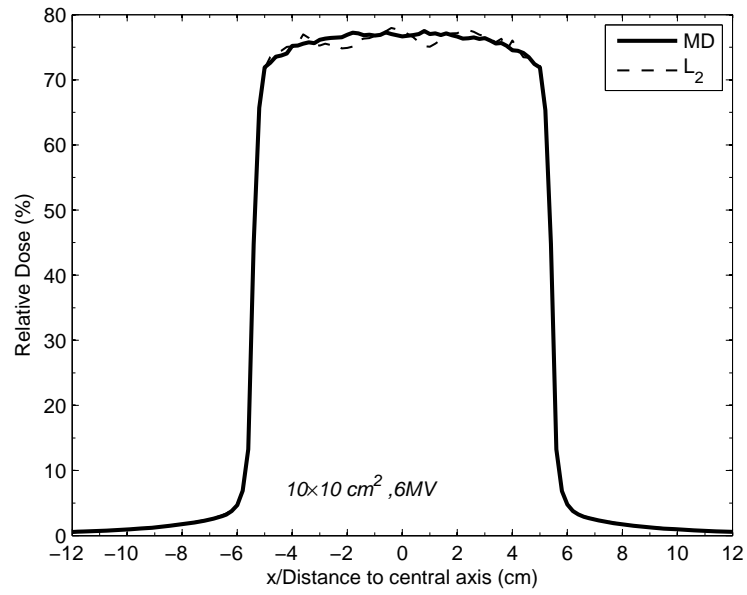
### **Larger field sizes**

As discussed in Section 4.4.3, CPE is gradually recovered inside the beam with increased field sizes. For the  $10 \times 10$  cm<sup>2</sup> field size and for both energies, the differences between the mean density model and all four realizations are negligible ( $<1.5\%$  on average) at the high dose region, except for the middle (realization  $L_1$ ), due to the extra upstream attenuation. The large structures off the CAX yet inside the  $10 \times 10$  cm<sup>2</sup> field size in realization  $L_2$  lead to a similar but smaller attenuation effect, which is clearer for the softer 6 MV beam ( $\sim 4\%$  at 1.1 cm behind the lung) than the 18 MV beam ( $\sim 2.5\%$  at the same depth). Although the lateral CPE is well established deep inside such a wide beam, in regions close to the beam's edge, lateral CPE does not exist because there is no compensation scattering from outside. Therefore, any significant local density variations occurring close to the beam's edge may possibly be of concern. To investigate this, we examined the central dose profiles in realization  $L_2$  for the  $10 \times 10$  cm<sup>2</sup> field size at two different depths (Fig. 4.14, 6.1 cm and 7.5 cm deep in the lung), in which the first depth crosses the two large airways' lumens and the second depth crosses the two accompanying arteries and a single vein near the CAX. The two airways and their arteries are close to the lung-tissue interface ( $<1$  cm). However, only slight local perturbations ( $<2\%$ ) are found to be associated with these structures, which indicates a state close to CPE. For this large field size,

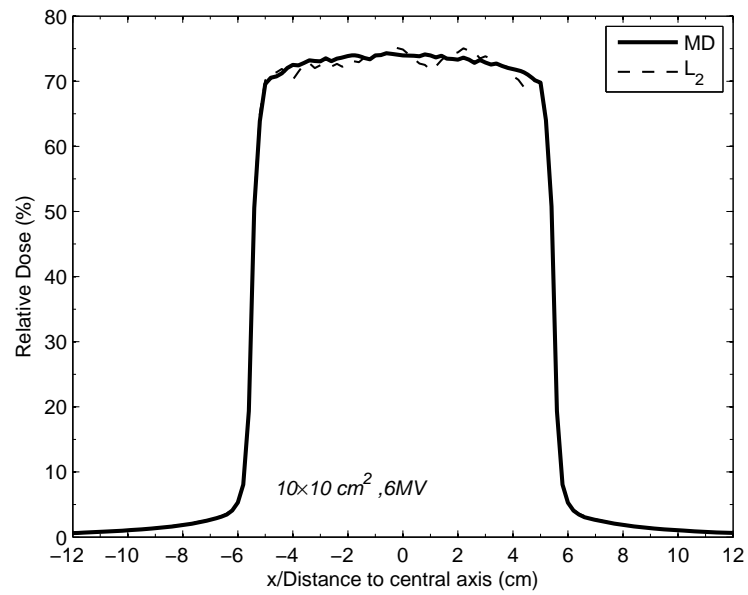
the increased scattering within water may be compensating for the dose reduction in the large structures.

#### 4.4.5 Mean lung doses

The mean lung dose (MLD = total energy deposited in the lung divided by the total mass of the lung) can illustrate from another point of view the perturbations caused by the random structures. Table 4.2 gives the MLDs for each geometry and field size, for both the 18 and 6 MV energies. We observe: (i) the MLDs are not directly related to the mean density of the whole lung. Rather, they are mainly determined by the tissue of the region through which the beam passes. For example, the  $L_1$  and  $L_2$  realizations have almost the same whole lung mean density (with a mean lung density of 0.293 and 0.306 g/cm<sup>3</sup>, respectively, both greater than the mean density model's 0.26 g/cm<sup>3</sup>) but differ much in structures in the narrow central regions covered by the  $1 \times 1$  cm field. Along the CAX, realization  $L_1$  has three large structures, but realization  $L_2$  has mostly small background structures. For the 6 MV beam, this difference results in a much higher (55% larger than the mean density model) MLD for the  $L_1$  realization and a significantly lower (9.2% smaller) MLD for the  $L_2$  realization. This can also be seen in more detail from the corresponding CAX depth doses and the central dose profiles, which show that most energy is deposited within the beam and in the high density regions (for the  $1 \times 1$  cm<sup>2</sup> field size). (ii) As the field size increases, the differences between the MLDs for the two large realizations decrease for  $5 \times 5$  cm<sup>2</sup> field size and become negligible for the  $10 \times 10$  and  $20 \times 20$  cm<sup>2</sup> field sizes. The differences between the four heterogeneous realizations and the mean density model also show a similar trend. Two reasons contribute to this (as a function of field size): (a) more structures are present in the open beam, so more energy is absorbed; and (b) the gradual recovery of CPE inside the beam. (iii) At



(a)



(b)

Figure 4.14: CAX dose profiles for  $10 \times 10 \text{ cm}^2$  field size for 6 MV at (a)  $y=9.1 \text{ cm}$  (6.1 cm deep in the lung); (b)  $y=10.5 \text{ cm}$  (7.5 cm deep in the lung). The thick solid lines are for the mean density (MD) lung model. The dashed lines are for large realization  $L_2$ . All curves for the same field size are normalized to the  $D_{max}$  along the CAX of the corresponding mean density (MD) case.

Table 4.2: Mean lung dose for each field size (cm<sup>2</sup>) and beam energy for the mean density lung model (MD) and one of the four random heterogeneous realizations as a percentage normalized to the  $D_{max}$ (MeV/g) along the central axis of the corresponding mean density lung case. (The mean lung density (g/cm<sup>3</sup>) for the corresponding case is also listed in the parentheses.)

field size	$D_{max}$ ( $\times 10^{-4}$ )	MD (0.26)	$L_1$ (0.293)	$L_2$ (0.306)	$S_1$ (0.242)	$S_2$ (0.246)
6 MV						
1 × 1	369	0.98	1.52	0.89	0.92	0.92
5 × 5	16.0	22.41	24.41	21.73	21.95	22.32
10 × 10	4.08	77.21	77.06	76.73	77.13	77.22
20 × 20	1.07	83.32	82.83	82.68	83.42	83.42
18 MV						
1 × 1	571	1.29	2.01	1.19	1.22	1.22
5 × 5	30.2	24.12	26.47	23.68	23.50	23.92
10 × 10	7.67	79.69	80.17	79.82	79.32	79.52
20 × 20	2.01	87.84	87.60	87.31	87.81	87.87

all field sizes and beam energies, the MLDs of the two small realizations (with a mean lung density of 0.242 and 0.246 g/cm<sup>3</sup>, respectively, both < the mean density model's 0.26 g/cm<sup>3</sup>) are much closer to each other and also closer to that of the mean density model than those of the two large realizations.

#### 4.4.6 Detailed vs. voxelized lungs

Fig. 4.15 is a preliminary calculation showing the difference between a detailed random lung realization  $L_1$  (Fig. 4.2c) and its voxelized version (Fig. 4.4) in terms of CAX depth dose. Basically, the two curves agree reasonably well with each other. The voxelized lung at the resolution of  $0.4 \times 0.4 \times 0.4$  cm<sup>3</sup> reveals most of the structures in the detailed one, and in particular, the magnitude of the underdosing distal to the lung is reproduced. This is as expected, because the underdosing is almost entirely affected by the amount of attenuation of the primary photons, which in turn depends on the radiological length along the beam's path. The average density in the beam's pathway is conserved, even though the homogenization in each voxel tends to smooth out the details of the structure. However, the voxelization still causes local differences up to 5% in the non-air region (up to 12% in the airway), which is due to

the smoothing of the structures in the voxels. Appropriate resolutions of voxelization is a complex issue and subject to further investigation. More comprehensive results are presented and discussed in Chapter V.

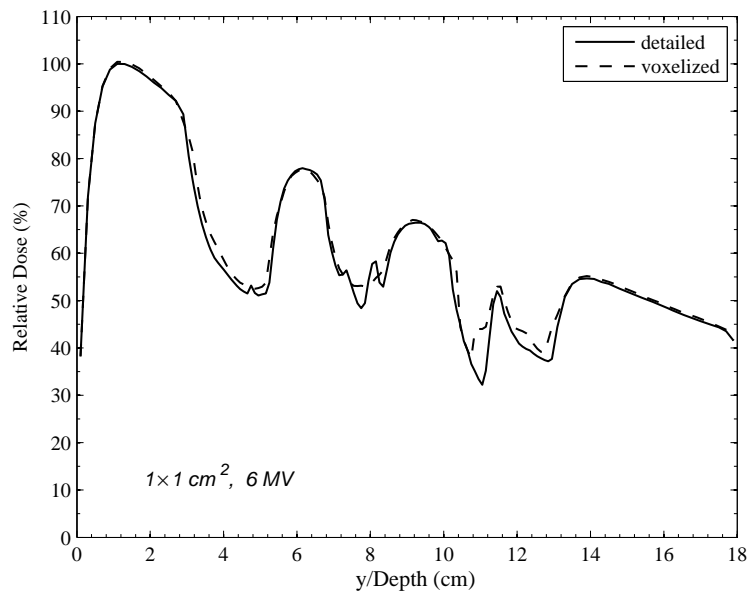


Figure 4.15: The CAX percent depth doses for a  $1 \times 1 \text{ cm}^2$  field size and 6 MV beam. The solid line is for the detailed realization  $L_1$  (Fig. 4.2c) and the dash line is for its voxelized version (Fig. 4.4), respectively. Both curves are normalized to the  $D_{max}$  along the CAX of the detailed  $L_1$  case.

## 4.5 Conclusion

We have developed a random heterogeneous 2 1/2-D lung model, based upon real lung physical data, by explicitly treating the bronchial and vessel tree structures within a homogenized tissue background with adjusted density. A threshold size of 0.05 cm, at the order of the terminal bronchiole, has been selected for this random lung model. Four realizations of this model were chosen to represent various scenarios that may be encountered in lung treatment planning. Monte Carlo simulations using the PENELOPE Monte Carlo code were performed on the homogeneous mean density lung model and the four heterogeneous realizations, for a single beam of two different energies (6 MV and 18 MV) and four field sizes ( $1 \times 1$ ,  $5 \times 5$ ,  $10 \times 10$  and

$20 \times 20 \text{ cm}^2$ ). By comparing the CAX percent depth doses, the central dose profiles, and the MLD among all the cases, we conclude that when the beam traverses a region with significantly large local structures, such as the regions close to the main and lobar bronchi and the vessels of the same order, a serious concern can exist if these structures are inside the beam. Also, significant local perturbations in dose (more than 30% of  $D_{max}$  larger than the mean density model for the 18 MV) are found for the small  $1 \times 1 \text{ cm}^2$  field size. As the field size increases, the local perturbation may finally vanish as CPE is established. However, an extra concentration of density inside the beam can lead to dose reduction as high as 7% of  $D_{max}$  in the distal “shadow” part of the beam, which is not compensated by inward scattering, even with the largest field size ( $20 \times 20 \text{ cm}^2$ ) in this study. This situation affects low-energy beams more than high-energy ones because of their softer spectra. Also, the reduction in dose in the “shadow” regions behind large structures is largely independent of the field size. On the other hand, if there are no large structures inside the beam, the results (especially the MLD) are closer to the mean density model (yet still show geometry-specific variation). For the small field sizes, such as the  $1 \times 1 \text{ cm}^2$  beam where lateral CPE is absent, even a relatively small structure ( $\sim 0.22 \text{ cm}$ ) simulated in the heterogeneous model can significantly perturb the dose.

Also, in a preliminary calculation, we compared the CAX depth dose for a random lung and a voxelized counterpart using a 0.4 cm resolution. We found a difference up to 5% of  $D_{max}$  in a non-air region.

Our results show that the mean density model for the whole lung is not generally a good approximation, especially for small field sizes, and that a voxelized model with 0.4 cm resolution can also have significant errors.



## CHAPTER V

# CT Resolution for Lung Treatment Planning: An Application of the Random Lung Model

### 5.1 Introduction

Most current treatment planning methods are CT-based [7, 28, 33, 37, 60, 80, 92, 98, 100, 122, 127], in which the patient geometry is delineated by a matrix of uniform rectangular box-like voxels of various sizes. These CT voxels are interpolated from the CT scans of the patient body, with resolutions that correspond to the number of the voxels across the CT images. Each voxel has a specific electron density obtained from the CT number of this voxel, which then maps to a specific homogenized material according to a prescribed electron density to material curve.

The better the CT voxels match the patient geometry and material composition, the more accurate the CT-based dose calculations will be. Two influencing factors exist: (i) the accuracy of the CT numbers obtained from the CT scans, and (ii) the resolution of the patient geometry represented by the CT voxels. Many previous publications [25, 44, 50, 57, 61, 85, 106] have considered the effect of CT numbers on dose calculations. In particular, Geise and McCullough [44] indicated that it was more important to know the accurate distribution of the spatial heterogeneities than the accurate electron density. Niemierko and Goitein [86] studied the error in calculated isodose lines caused by using different dose grid sizes, and recommended a dose grid

size 2.5 times larger than the acceptable maximum position error. Their work was augmented by Smith, Morrey and Gray in a letter to the editor [110], in which the authors proposed a maximum 2 mm dose grid size in a heterogeneous site such as the head and neck. In a recent work [29], Corbett *et al.* reported that a dose grid of 1 mm was sufficient to achieve accurate DVHs within  $\pm 5\%$  up to 200% of the target dose in a prostate  $^{125}\text{I}$  seed implant dose calculation. More recently, the dose grid size effect was investigated by Dempsey *et al.* [35] using a Fourier analysis, and was illustrated in a head and neck IMRT treatment planning. A 2.5 mm isotropic dose grid was concluded to be sufficient to prevent dose errors greater than 1%, and a 2-4-6 mm adaptive dose grid model was suggested for targets, structures and tissue, respectively.

However, the works cited above on the dose grid size effect emphasize the dose point sampling resolution based on a fixed uniform geometrical grid. To our knowledge, only a limited number of studies have investigated the effect of various geometrical resolutions on dose calculations. One study [32], involving electron beams and the Monte Carlo method, demonstrated that in the vicinity of the interface between an air cavity and water, a 1.9 mm geometrical voxel had better agreement with the measurement than a 3.9 mm voxel size, where the largest disagreement exceeded 5%. In another study by Chung *et al.* [26], the point dose differences between various resolutions from 1.5 mm to 6 mm were shown to be up to 5.6%. The authors concluded that although 3 mm and 4 mm grid sizes were considered acceptable to most IMRT plans, a 2 mm grid size was required to achieve accurate dose distribution in heterogeneous regions. Meanwhile, they found that these point differences did not lead to noticeable differences in the dose volume histogram because the regions with high dose differences occupy only a very small portion of the whole region of interest.

To take a step further, De Smedt *et al.* [34] investigated the combined effect of the resolutions of both the CT voxels and the dose scoring grid on the dose calculated for IMRT treatment planning for head and neck and lung cases. They confirmed that for the lung cases, the scoring grid resolution was less important than the CT resolution, and they recommended a geometrical resolution of  $2 \times 2 \times 5 \text{ mm}^3$  over a reference  $1 \times 1 \times 5 \text{ mm}^3$  resolution to save calculation time while not compromising accuracy.

In the previous chapter, we developed a random lung model and compared the dose distributions between the detailed lung model with explicit structures inside and its atomic mix counterpart. We indicated that one possible application is that we can use this detailed, highly heterogeneous lung model to investigate how the geometrical resolution (the CT resolution) affects the dose calculations. However, contrary to the one-beam calculations in Chapter IV, we use in this chapter a three-beam setup on a more realistic lung phantom with different sizes of tumor embedded inside the lung.

The remainder of this chapter is arranged as follows: in Section 5.2, we describe the detailed treatment setup of the lung phantom, the Monte Carlo simulations, and the metrics employed for evaluating dose distributions for different CT resolutions. In Section 5.3 we present the results and discussion. Section 5.4 gives our conclusions.

## **5.2 A Three-beam Treatment Planning Using the Random Lung Model**

### **5.2.1 The Lung Phantom**

The phantom set up for the three-beam treatment planning is sketched in Fig. 5.1. The posterior-anterior (PA) direction is in the  $+x$ -direction, the superior-inferior (SI) direction is in the  $-z$ -direction and the right-left direction is in the  $+y$ -direction. The phantom is a box-like uniform water (outer clear box) tank of  $24 \times 30 \times 36 \text{ cm}^3$  with

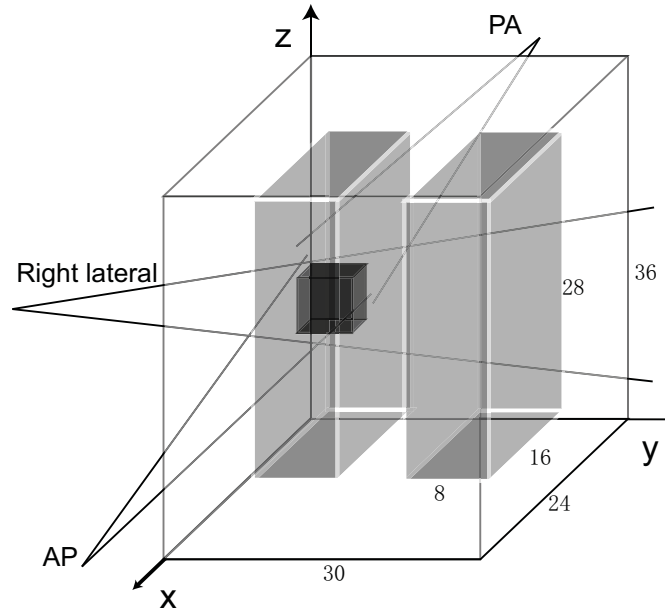


Figure 5.1: The lung phantom (the outer clear box) for three-beam (one right lateral (RL) beam and two opposing AP-PA beams) treatment planning. The phantom is  $24 \times 30 \times 36 \text{ cm}^3$  with two lungs (light gray box, right and left) of the same size of  $16 \times 8 \times 28 \text{ cm}^3$  embedded inside. The water layers surrounding the lung have uniform thickness of 4 cm while the lateral distance between the two lungs is 3 cm. The cubical tumor (dark gray) of various sizes resides in the middle of the right lung.

two box-like lung regions (light gray boxes) of the same size of  $16 \times 8 \times 28 \text{ cm}^3$  embedded inside. The water layers surrounding the lung have uniform thickness of 4 cm, while the lateral distance between the two lungs is 3 cm. The phantom dimensions are similar to those in an actual patient's geometry. A cubical tumor (dark gray box) of various sizes is located in the middle of the right lung, and its center is the iso-center for the treatment planning. The random lung model is generated (see Chapter IV) and applied to the right lung with the lung structure's cylindrical axis parallel to the  $z$ -direction, while the left lung is a homogenized atomic mix model with no explicit structures because no incident beams pass through the left lung before they strike the right lung. Therefore, the dose to the right lung from photons and electrons back-scattered from the left lung can be neglected. Fig. 5.2 shows the top view ( $x$ - $y$  plane) of the phantom setup with a typical realization of

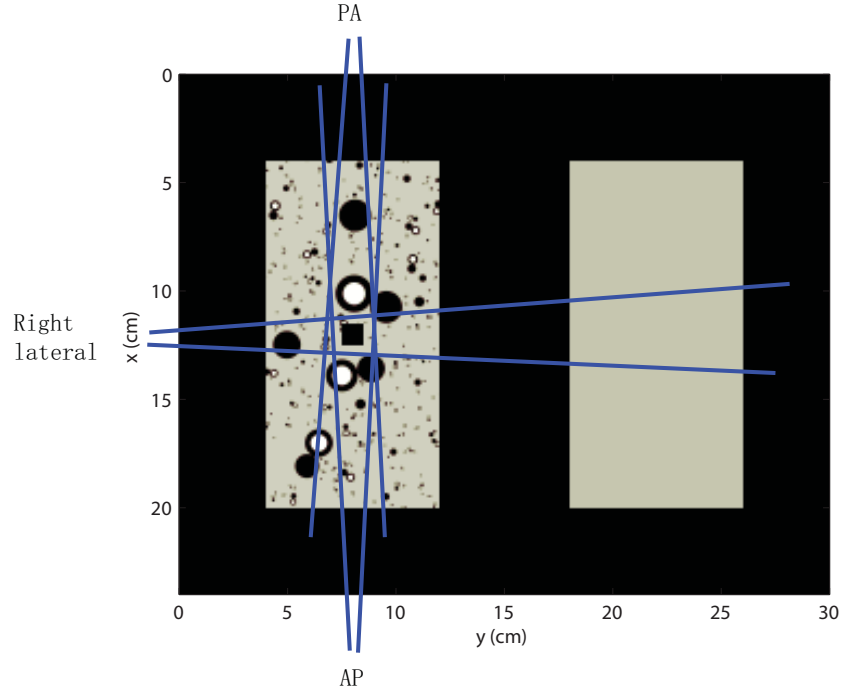


Figure 5.2: Top view of the lung phantom (Fig. 5.1) with a realization of the random lung model in the right lung across the isocenter. Three beams (RL, AP and PA) are indicated. A tumor is represented by the black square in the middle of the right lung.

the random lung model in the right lung across the isocenter plane.

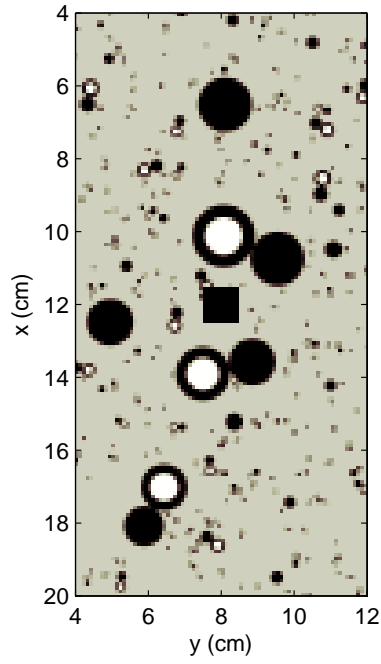
We use the method described in Chapter IV to generate a series of random lung models and fit in the right lung region. Based on the fact that for large field sizes, the difference between a random lung model and the atomic mix lung model becomes very small due to charged particle equilibrium (CPE) (Chapter IV), we choose to examine the CT resolution effect for small field sizes. Therefore two tumor sizes of  $1 \times 1 \times 1 \text{ cm}^3$  and  $4 \times 4 \times 4 \text{ cm}^3$  are picked as two representative dimensions in our treatment planning. For each tumor size, we generate (i) a large realization with large structures close to the tumor and (ii) a small realization with no large structures. For each realization, three CT resolutions on the  $x$ - $y$  plane (considering the lung model is  $2\frac{1}{2}$ -D, we use 2 mm in  $z$ -direction for all the three CT resolutions) are applied:  $1 \times 1 \text{ mm}^2$ ,  $2 \times 2 \text{ mm}^2$  and  $4 \times 4 \text{ mm}^2$ . Figs. 5.3 through 5.6 show all the

realizations that are used in the simulations. For simplicity, these figures only show the lung region. For the simulations with a tumor size of  $1 \times 1 \times 1 \text{ cm}^3$  (Figs. 5.3 and 5.4), Figs. 5.3a through 5.3c depict the large realization at a CT resolution of  $1 \times 1 \text{ mm}^2$  ( $T1L01$ ),  $2 \times 2 \text{ mm}^2$  ( $T1L02$ ),  $4 \times 4 \text{ mm}^2$  ( $T1L04$ ), respectively, while Figs. 5.4a through 5.4c are for the small realization at the same three different CT resolutions ( $T1S04$ ,  $T1S04$  and  $T1S04$ ). A similar description applies to Figs. 5.5 and 5.6 for the simulations with a tumor size of  $4 \times 4 \text{ cm}^3$ : (a) through (c) represent  $T4L01$ ,  $T4L02$  and  $T4L04$ , and (a) through (c) represent  $T4S01$ ,  $T4S02$  and  $T4S04$ , respectively.

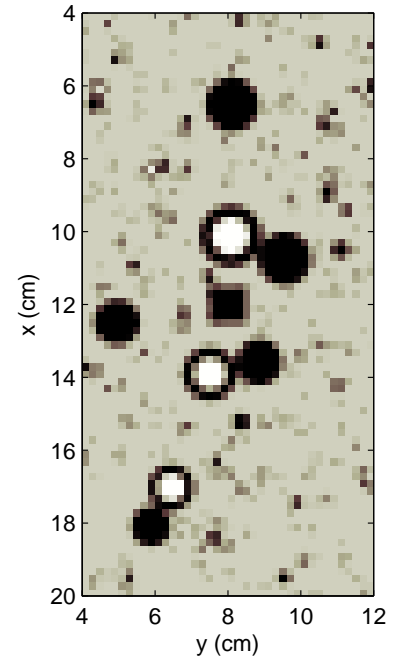
For convenience, we may denote the  $1 \times 1 \text{ mm}^2$  CT resolution as  $CT01$ , the  $2 \times 2 \text{ mm}^2$  CT resolution as  $CT02$ , and the  $4 \times 4 \text{ mm}^2$  CT resolution as  $CT04$ .

### 5.2.2 Photon Beams

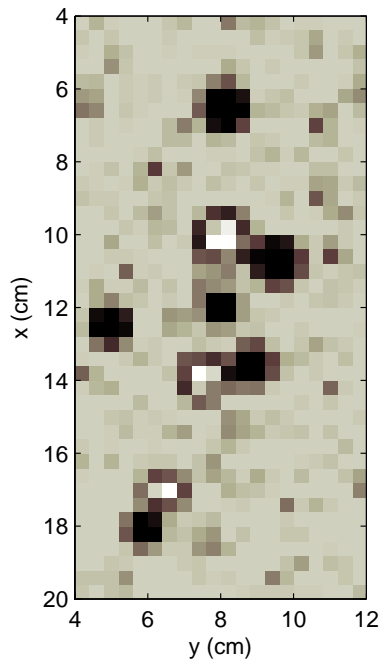
All three beams, a RL beam normally incident in the  $+y$ -direction and two opposing AP and PA beams, are open field from a point source at a source-to-axis distance (SAD) = 100 cm, and with uniform photon fluences upon entering the surface of the phantom across the field. The field size at the isocenter plane is specified as follows: a 0.5 cm margin is added onto the edge of the tumor (since we do not consider any organ/tumor movement and microscopic extension of the tumor, the tumor is both a GTV and a PTV). Therefore, the field size at the isocenter plane is  $2 \times 2 \text{ cm}^2$  for a tumor size of  $1 \times 1 \times 1 \text{ cm}^3$  and  $5 \times 5 \text{ cm}^2$  for a tumor size of  $4 \times 4 \times 4 \text{ cm}^3$ . We use an energy spectrum of 6 MV [109] for all three beams because this energy is commonly used for lung treatment in practice.



(a)

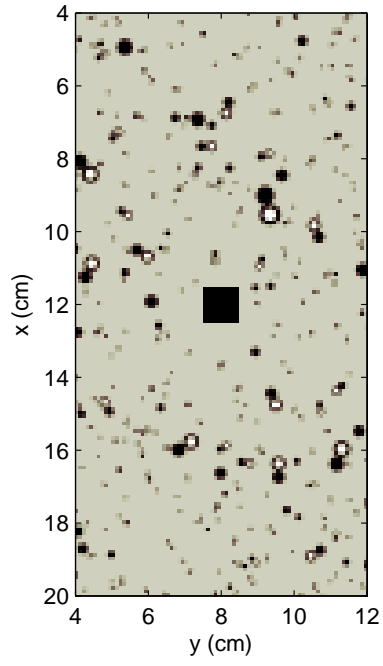


(b)

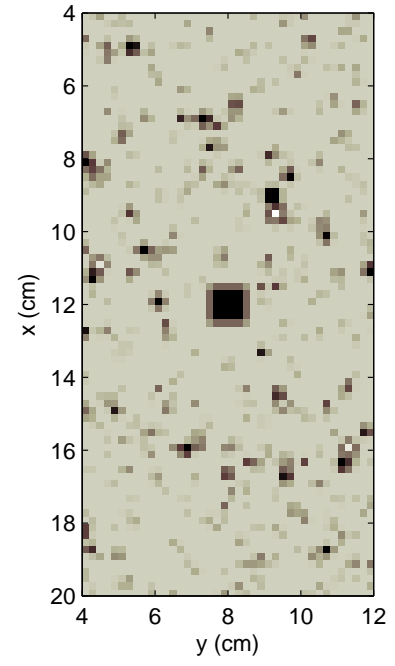


(c)

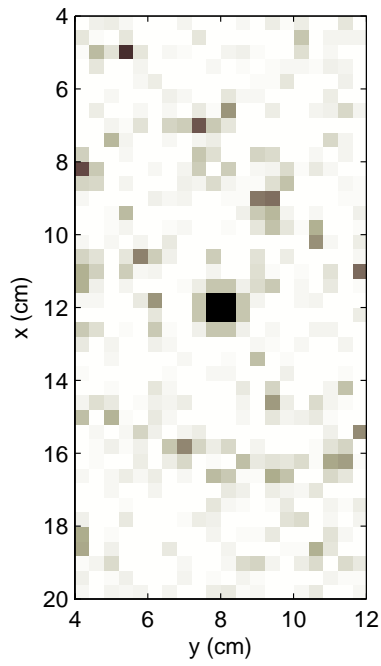
Figure 5.3: Large realizations for simulations with a tumor size of  $1 \times 1 \times 1 \text{ cm}^3$ : (a),  $T1L01$ , at a CT resolution of  $1 \times 1 \text{ mm}^2$ ; (b),  $T1L02$ , at  $2 \times 2 \text{ mm}^2$ ; (c),  $T1L04$ , at  $4 \times 4 \text{ mm}^2$ .



(a)



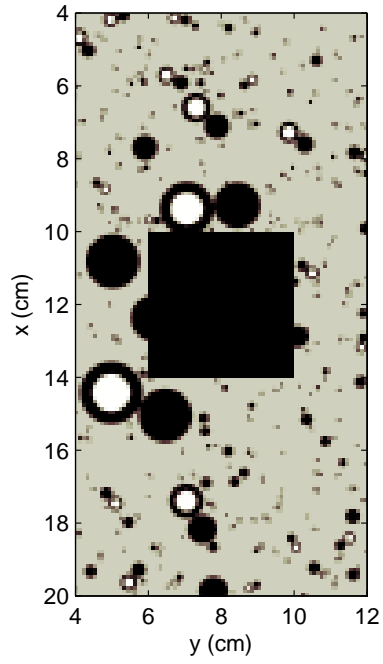
(b)



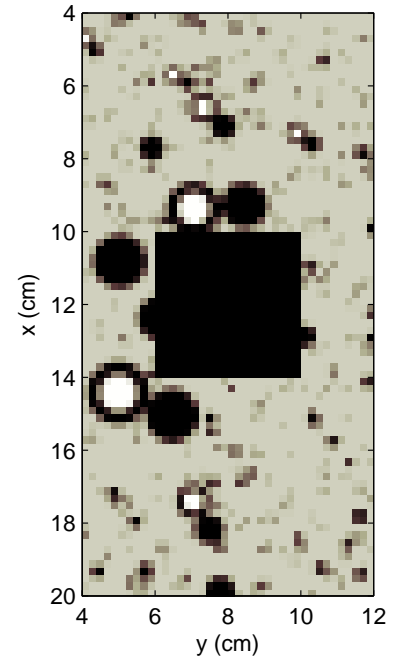
(c)

Figure 5.4: Small realizations for simulations with a tumor size of  $1 \times 1 \times 1 \text{ cm}^3$ : (a), *T1S01*, at a CT resolution of  $1 \times 1 \text{ mm}^2$ ; (b), *T1S02*, at  $2 \times 2 \text{ mm}^2$ ; (c), *T1S04*, at  $4 \times 4 \text{ mm}^2$ .

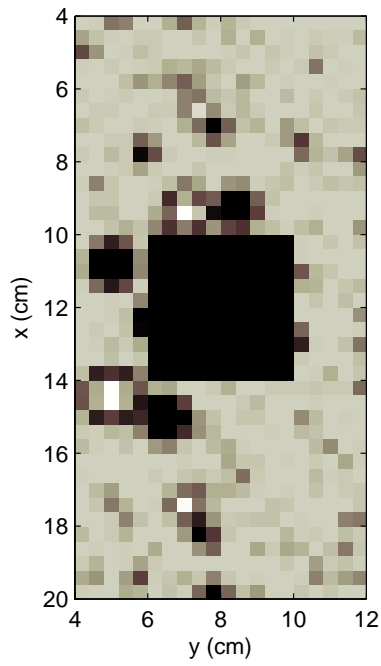




(a)

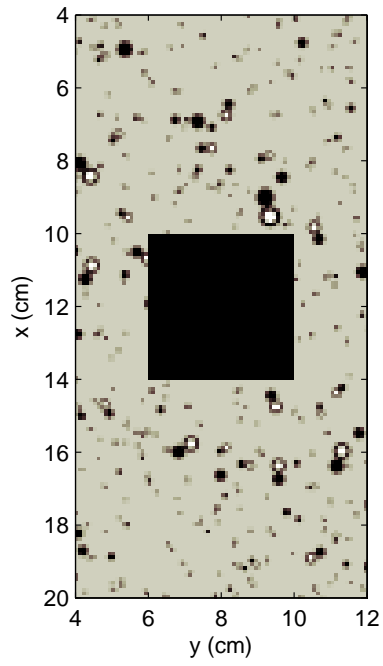


(b)

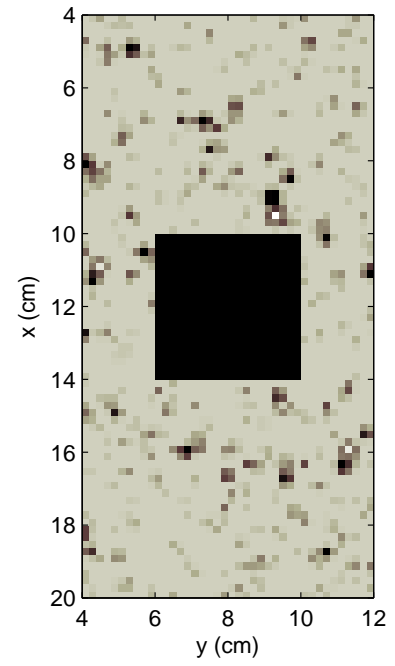


(c)

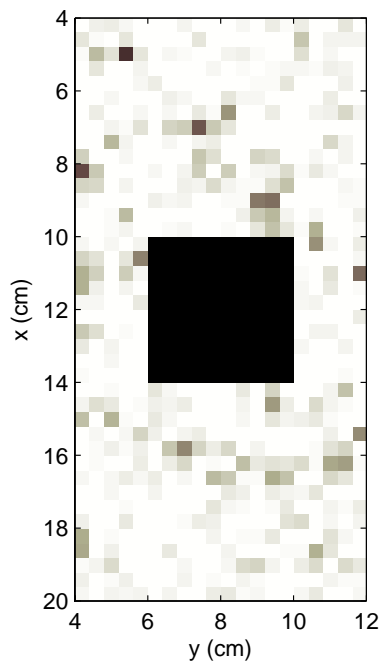
Figure 5.5: Large realizations for simulations with a tumor size of  $4 \times 4 \times 4 \text{ cm}^3$ : (a),  $T4L01$ , at a CT resolution of  $1 \times 1 \text{ mm}^2$ ; (b),  $T4L02$ , at  $2 \times 2 \text{ mm}^2$ ; (c),  $T4L04$ , at  $4 \times 4 \text{ mm}^2$ .



(a)



(b)



(c)

Figure 5.6: Small realizations for simulations with a tumor size of  $4 \times 4 \times 4 \text{ cm}^3$ : (a),  $T4S01$ , at a CT resolution of  $1 \times 1 \text{ mm}^2$ ; (b),  $T4S02$ , at  $2 \times 2 \text{ mm}^2$ ; (c),  $T4S04$ , at  $4 \times 4 \text{ mm}^2$ .

### 5.2.3 Monte Carlo Simulations

Because the random lung model is highly heterogeneous and consists of many of material boundaries, we mostly used PENELOPE [104] in Chapter IV. However, since the current PENELOPE code is body-based and cannot perform MC simulations in a voxelized way, for our purpose to compare the dose distributions between various CT resolutions, we used the voxel-based MC code, DPM [108]. Both DPM and PENELOPE are well-documented MC codes and show good agreement in the testing situations [108]. The parameters applied in all the MC simulations are the same as described in Section 4.3: cutoff energies of 100 keV for electrons/positrons ( $E_{cut}$ ) and 20 keV for photons ( $P_{cut}$ ) are used. A dose scoring grid of  $1 \times 1 \times 2 \text{ mm}^3$  is applied throughout. The dose is normalized to  $D_{iso}$  of the correspondent cases at CT01. For cases with a tumor size of  $1 \times 1 \times 1 \text{ cm}^3$ , the  $1\sigma$  statistical deviation at  $D_{iso}$  ( $\sigma_{iso}$ ) is  $< 0.2\%$ . The average  $1\sigma$  over the region with dose greater than 50% of  $D_{iso}$  ( $\sigma_{50}$ ) is  $< 0.3\%$ . For cases with a tumor size of  $4 \times 4 \times 4 \text{ cm}^3$ ,  $\sigma_{iso}$  and  $\sigma_{50}$  are less than 0.4% and 0.5%, respectively.

### 5.2.4 Treatment Plan Evaluation Metrics

We use both qualitative and quantitative methods to compare the dose distributions from different random lung model realizations at various CT resolutions.

Isodose lines are basically a qualitative graphical tool for visually inspecting the dose distribution within the regions of interest. We use isodose lines to show the differences in the dose distributions for the tumor and inside the lung when the CT resolution changes.

The mean lung dose (MLD) is a crude yet useful indicator for lung complications and is often used for treatment plan evaluation [69, 101, 102, 107]. We calculate the

MLDs for different cases by dividing the total energy deposited in the right lung (excluding the tumor) by its total mass. Also for comparison, we provide the mean tumor dose (MTD, calculated by dividing the total energy deposited in the tumor by the mass of the tumor) along with the MLDs.

The dose volume histogram (DVH) [36, 73] is another widely-used quantitative method to evaluate rival treatment plans, which can indicate the uniformity of the dose coverage in the target and show any hot spot present for the normal tissue. The most widely-used DVH is in a cumulative form, which plots, for a specific region of interest, the fraction of volume receiving a dose exceeding a given value. The cumulative DVH is essentially a cumulative distribution function (CDF), which is expressed by Eq. (5.1):

$$DVH(D) = \int_D^{D_{max}} PDF(D') dD' \quad , \quad (5.1)$$

where  $DVH(D)$  is the cumulative DVH as a function of dose  $D$ ;  $PDF(D')$  is the probability that a dose  $D'$  would fall into the range of  $[D', D' + dD']$  and satisfies  $\int_0^\infty PDF(D') dD' = 1$ . In this chapter, we calculate the cumulative DVH (for simplicity, we call it DVH in the remaining part) by binning the dose of each voxel into dose bins with equal space of 1% and ranging from 0 to  $D_{max}$ .

To quantify the differences between different DVHs, the absolute differential dose (ADD) described by Kawrakow [59] is adopted. We calculate the relative ADD ( $ADD_{rel}$ ) by Eq. (5.2) [34]:

$$\begin{aligned} ADD_{rel} &= \frac{ADD}{\int_0^{D_{max}} DVH_{ref}} \\ &= \frac{\sum_i^{N_{bin}} |h_i^{ref} - h_i^{comp}| \Delta d_i}{\sum_i^{N_{bin}} h_i^{ref} \Delta d_i} \quad , \end{aligned} \quad (5.2)$$

where  $h_i^{ref/comp}$  is the value of the reference/comparison DVH for bin  $i$ , and  $\Delta d_i$  is the width of bin  $i$ .

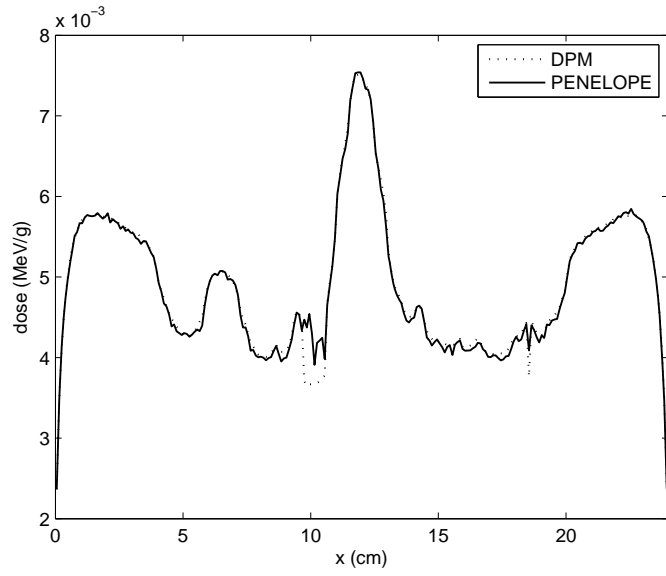
### 5.2.5 Reference CT Resolution for the Lung Phantom

A reference CT resolution (the finest one to compare with) is determined by: (i) generating a three-beam treatment plan for the lung phantom depicted in Fig. 5.3a with a small tumor size of  $1 \times 1 \times 1 \text{ cm}^3$  (hence a small field size which imposes a severe situation lacking CPE) using the detailed lung model (not voxelized) and the PENELOPE code; and (ii) voxelizing this lung phantom at a uniform fine geometrical (CT) grid and using the DPM code to do the calculations with the same three-beam setup. If the fine resolution shows no significant differences between the results from the detailed PENELOPE plan and the voxelized DPM plan, this resolution will be used as the reference. Consistent with previous publications, we choose the reference resolution at  $1 \times 1 \text{ mm}^2$  on  $x$ - $y$  plane. Because the speed of MC simulations in the detailed lung model with the PENELOPE code is much slower than that of the DPM simulations (see Chapter IV), the  $\sigma_{iso}$  and  $\sigma_{50}$  are less than 0.5% and 0.7% (larger than those with the DPM cases) for the PENELOPE cases.

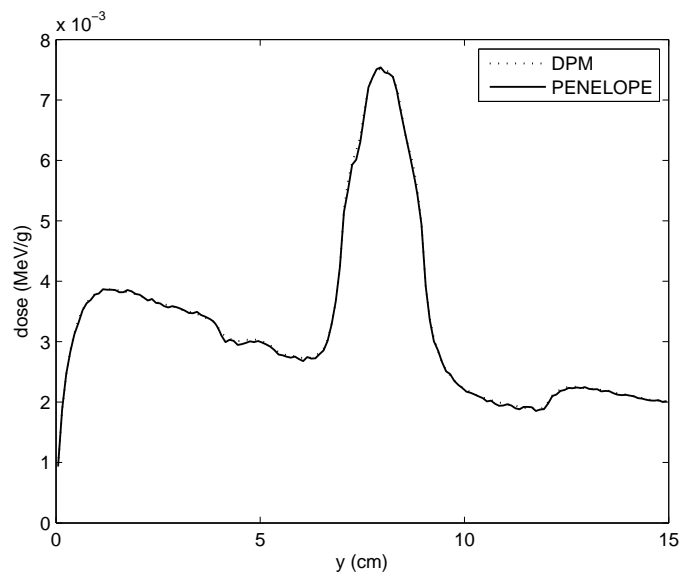
## 5.3 Results and Discussion

### 5.3.1 Reference CT Resolution for the Lung Phantom

Fig. 5.7 shows the CAX depth doses in the three-beam calculations for both the detailed lung model performed by the PENELOPE and the voxelized version by the DPM. Fig. 5.7a is the CAX depth dose along the  $x$ -direction. Fig. 5.7b is the CAX depth dose along the  $y$ -direction. In both figures, the solid lines are for the detailed lung model, while the dotted lines are for the voxelized lung model at the selected CT resolution of  $1 \times 1 \text{ mm}^2$ . The dose is expressed in absolute value. It is clear that at this resolution, the voxelized lung model agrees well with the detailed lung model. The CAX along the  $x$ -direction passes through some large and



(a)



(b)

Figure 5.7: CAX depth dose curves along (a):  $x$ -direction and (b):  $y$ -direction, of a three-beam simulation in a phantom depicted in Fig. 5.3a. The dotted lines are calculated with DPM while the solid lines are calculated with PENELOPE.

small structures, while the CAX along the  $y$ -direction passes mostly through the background region. Except for the two air regions (one is the lumen, about 1 cm in diameter, of a large airway around  $x = 10$  cm, and one is the lumen, about 2 mm in diameter, of a small airway at around  $x = 18.6$  cm), the relative differences ( $[D_{PENELOPE}(x) - D_{DPM}(x)]/D_{PENELOPE}(x)$ ) at most points are within  $-0.5\%$  to  $+1.5\%$ , which are comparable to the statistical errors. The significant discrepancy in the air regions may be caused by different mechanisms of dealing with regions with extremely low density employed in both codes. It is worth noting that the  $1\sigma$  in the air region is as large as 5%. Therefore we use the CT resolution of  $1 \times 1 \text{ mm}^2$  ( $CT01$ ) as the reference resolution.

### 5.3.2 Isodose lines

Figs. 5.8 and 5.9 show the isodose lines at  $x$ - $y$  plane across the isocenter for simulations for the large and small realizations with a tumor size of  $1 \times 1 \text{ cm}^3$  at various CT resolutions (Large realization: 5.8a,  $T1L01$ , at  $CT01$ ; 5.8b,  $T1L02$ , at  $CT02$ ; 5.8c,  $T1L04$ , at  $CT04$ . Small realization: 5.9a,  $T1S01$ , at  $CT01$ ; 5.9b,  $T1S02$ , at  $CT02$ ; 5.9c,  $T1S04$ , at  $CT04$ .) A similar arrangement in Figs. 5.10 and 5.11 is for the cases with a tumor size of  $4 \times 4 \text{ cm}^3$ , where the large realizations are: 5.10a,  $T4L01$ , at  $CT01$ ; 5.10b,  $T4L02$ , at  $CT02$ ; 5.10c,  $T4L04$ , at  $CT04$ . The small realizations are: 5.11a,  $T4S01$ , at  $CT01$ ; 5.11b,  $T4S02$ , at  $CT02$ ; 5.11c,  $T4S04$ , at  $CT04$ . Only the lung region is shown, which corresponds to Figs. 5.3 through 5.6. The tumor is indicated by the gray shaded square in the lung center. The isodose levels are 20, 50, 70, 90, 95 and 100% of the dose at the isocenter ( $D_{iso}$ ) of the realizations at the reference CT resolution of  $CT01$ .

As the CT resolution decreases (the geometrical grid size becomes larger), the fine details revealed in the high resolution are smoothed, especially when the large

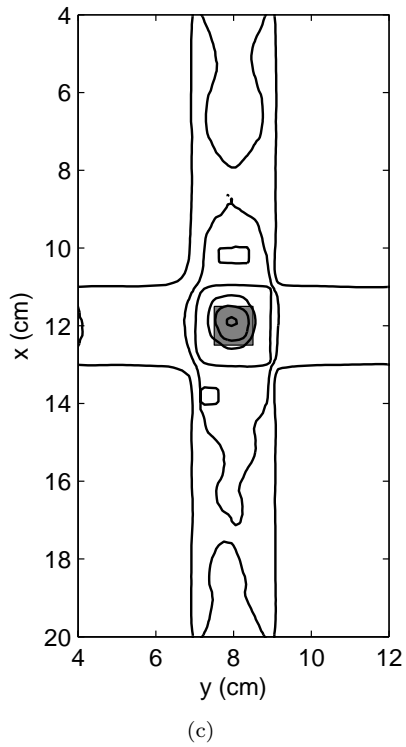
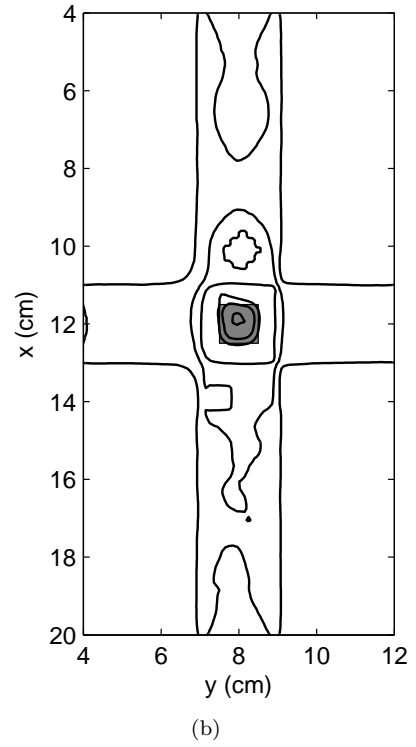
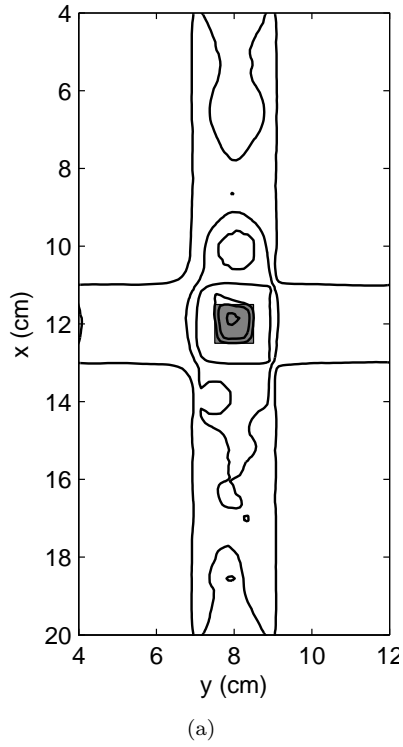


Figure 5.8: Isodose lines at  $x$ - $y$  plane across the isocenter for simulations for the large realizations with a tumor size of  $1 \times 1 \text{ cm}^3$  at various CT resolutions. Only the lung region is shown. The tumor is indicated by the shaded gray square in the lung center. The isodose levels are 20, 50, 70, 90, 95 and 100% of the dose at the isocenter ( $D_{iso}$ ) for the realizations at the reference CT resolution of  $1 \times 1 \text{ mm}^2$ . Realizations: (a),  $T1L01$ , at a CT resolution of  $1 \times 1 \text{ mm}^2$ ; (b),  $T1L02$ , at  $2 \times 2 \text{ mm}^2$ ; (c),  $T1L04$ , at  $4 \times 4 \text{ mm}^2$ .



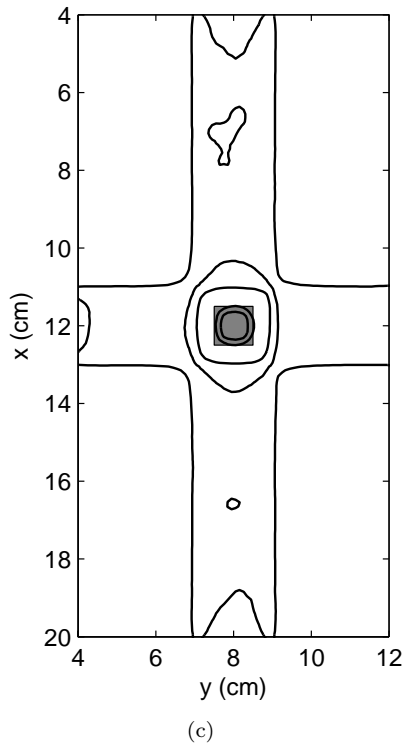
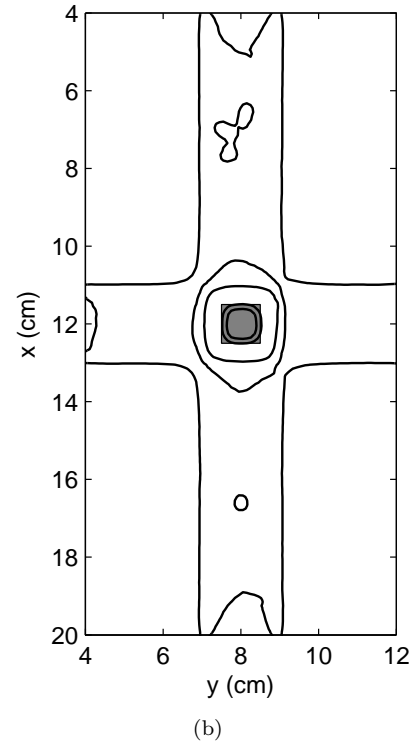
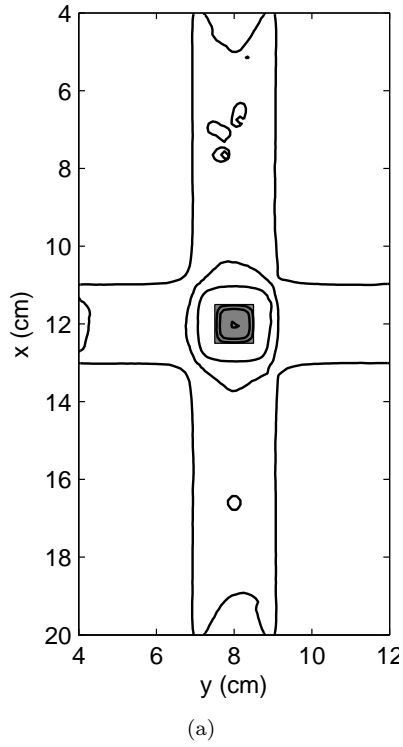


Figure 5.9: Isodose lines at  $x$ - $y$  plane across the isocenter for simulations for the small realizations with a tumor size of  $1 \times 1 \text{ cm}^3$  at various CT resolutions. Only the lung region is shown. The tumor is indicated by the shaded gray square in the lung center. The isodose levels are 20, 50, 70, 90, 95 and 100% of the dose at the isocenter ( $D_{iso}$ ) for the realizations at the reference CT resolution of  $1 \times 1 \text{ mm}^2$ . Realizations: (a),  $T1S01$ , at a CT resolution of  $1 \times 1 \text{ mm}^2$ ; (b),  $T1S02$ , at  $2 \times 2 \text{ mm}^2$ ; (c),  $T1S04$ , at  $4 \times 4 \text{ mm}^2$ .

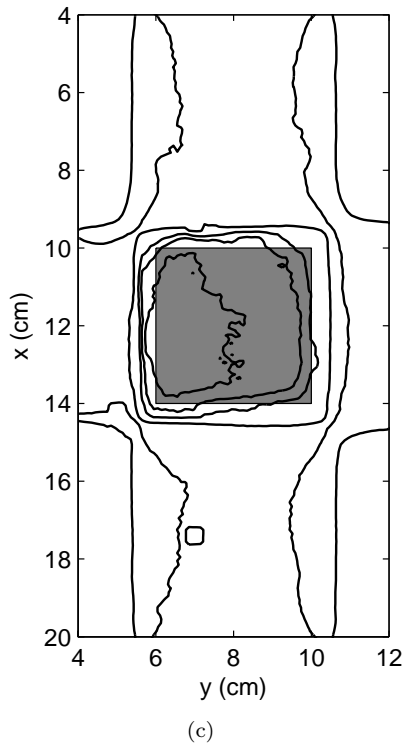
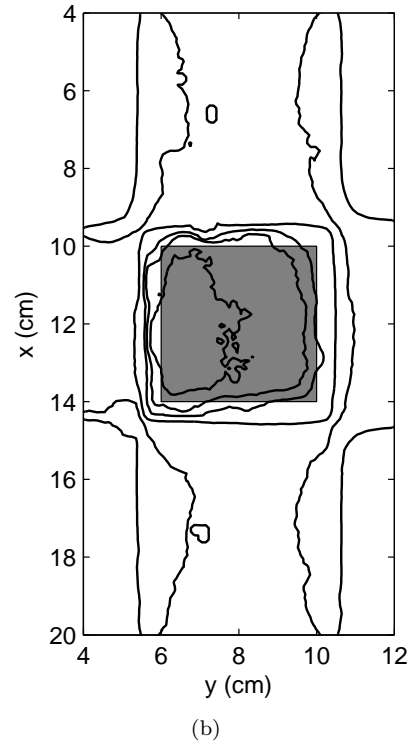
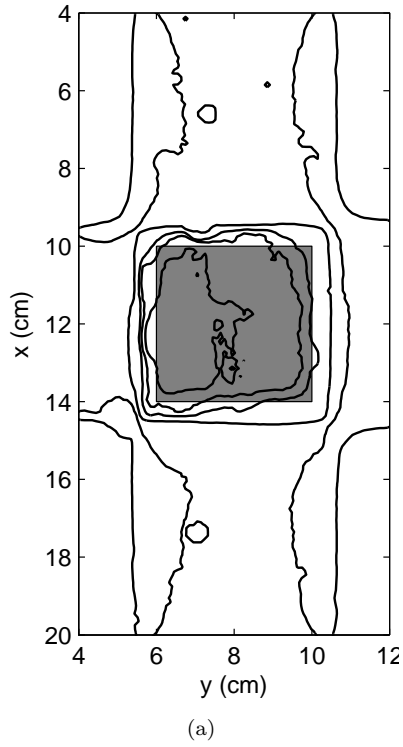


Figure 5.10: Isodose lines at  $x-y$  plane across the isocenter for simulations for the large realizations with a tumor size of  $4 \times 4 \times 4$  cm<sup>3</sup> at various CT resolutions. Only the lung region is shown. The tumor is indicated by the shaded gray square in the lung center. The isodose levels are 20, 50, 70, 90, 95 and 100% of the dose at the isocenter ( $D_{iso}$ ) for the realizations at the reference CT resolution of  $1 \times 1$  mm<sup>2</sup>. Realizations: (a),  $T4L01$ , at a CT resolution of  $1 \times 1$  mm<sup>2</sup>; (b),  $T4L02$ , at  $2 \times 2$  mm<sup>2</sup>; (c),  $T4L04$ , at  $4 \times 4$  mm<sup>2</sup>.

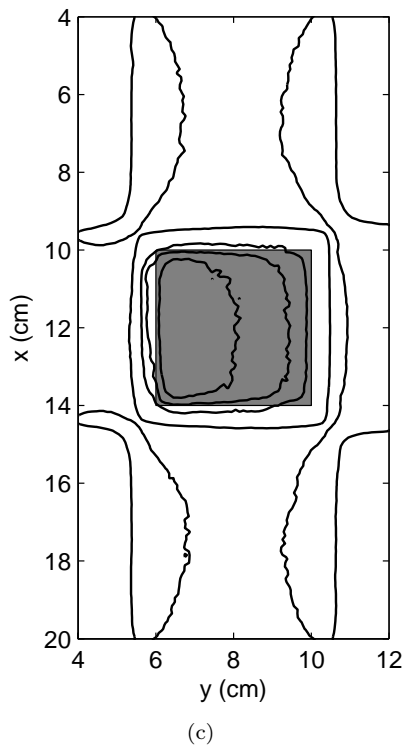
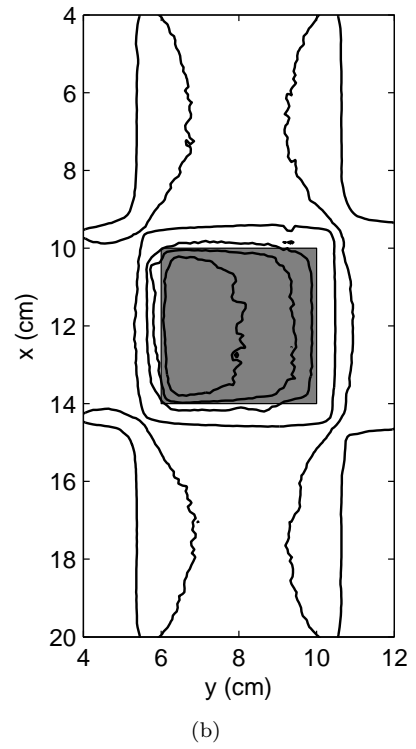
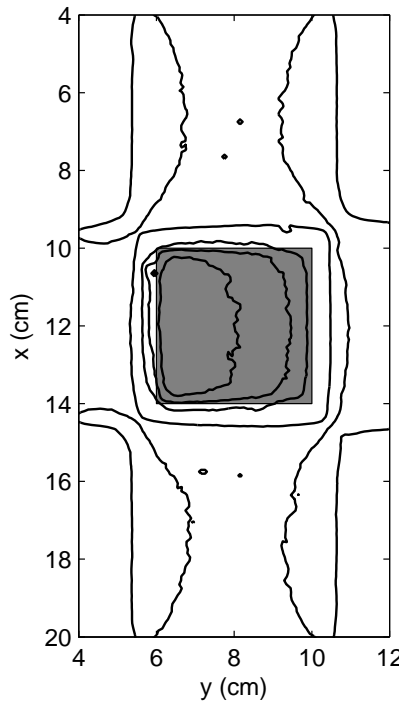


Figure 5.11: Isodose lines at  $x$ - $y$  plane across the isocenter for simulations for the small realizations with a tumor size of  $4 \times 4 \times 4 \text{ cm}^3$  at various CT resolutions. Only the lung region is shown. The tumor is indicated by the shaded gray square in the lung center. The isodose levels are 20, 50, 70, 90, 95 and 100% of the dose at the isocenter ( $D_{iso}$ ) for the realizations at the reference CT resolution of  $1 \times 1 \text{ mm}^2$ . Realizations: (a),  $T4S01$ , at a CT resolution of  $1 \times 1 \text{ mm}^2$ ; (b),  $T4S02$ , at  $2 \times 2 \text{ mm}^2$ ; (c),  $T4S04$ , at  $4 \times 4 \text{ mm}^2$ .

voxels cross the edge of a heterogeneity and regions of high dose gradient. Therefore, the dose distribution across the structures of various sizes is smoothed accordingly. This kind of difference can be viewed as a systematic error. This trend can be seen clearly in Fig. 5.8 through Fig. 5.11, where the artificial smoothing effect is more apparent with “large” realizations, which contain large structures, than with “small” realizations, which contain only small structures. The smoothing effect is also more apparent with a small tumor with a small field size. Here lack of CPE causes a more sensitive change in the dose distribution by local structures than in large field sizes.

The combinations between the different realizations and different tumor sizes (hence different field sizes) make the change of the isodose lines from the reference *CT01* to *CT04* different: while *T4S04* is almost identical to *T4S01*, *T1L04* shows some noticeable changes compared to *T1L01*, as is indicated by the 90% percent lines surrounding the tumor and the 50% percent lines surrounding the two large airway lumens. In all realizations, the differences between the *CT02* and the *CT01* resolutions are not significant.

In Chapter IV, we saw that the dose differences between different random lung realizations and the atomic mix model in the water region far behind the lung are due mainly to the attenuation of the primary photons. Therefore, the voxelization of a specific realization of the heterogeneous lung into different geometrical resolutions may not cause significant different in this far region. However, Fig. 4.15 (Page 88) shows that for a specific realization, different geometrical resolutions can result in significant changes in both the builddown and the buildup regions for a small field size. This indicates a potential dose coverage change due to different voxelizations in a treatment involving small field sizes. Figs. 5.12 and 5.13 show a close view of the

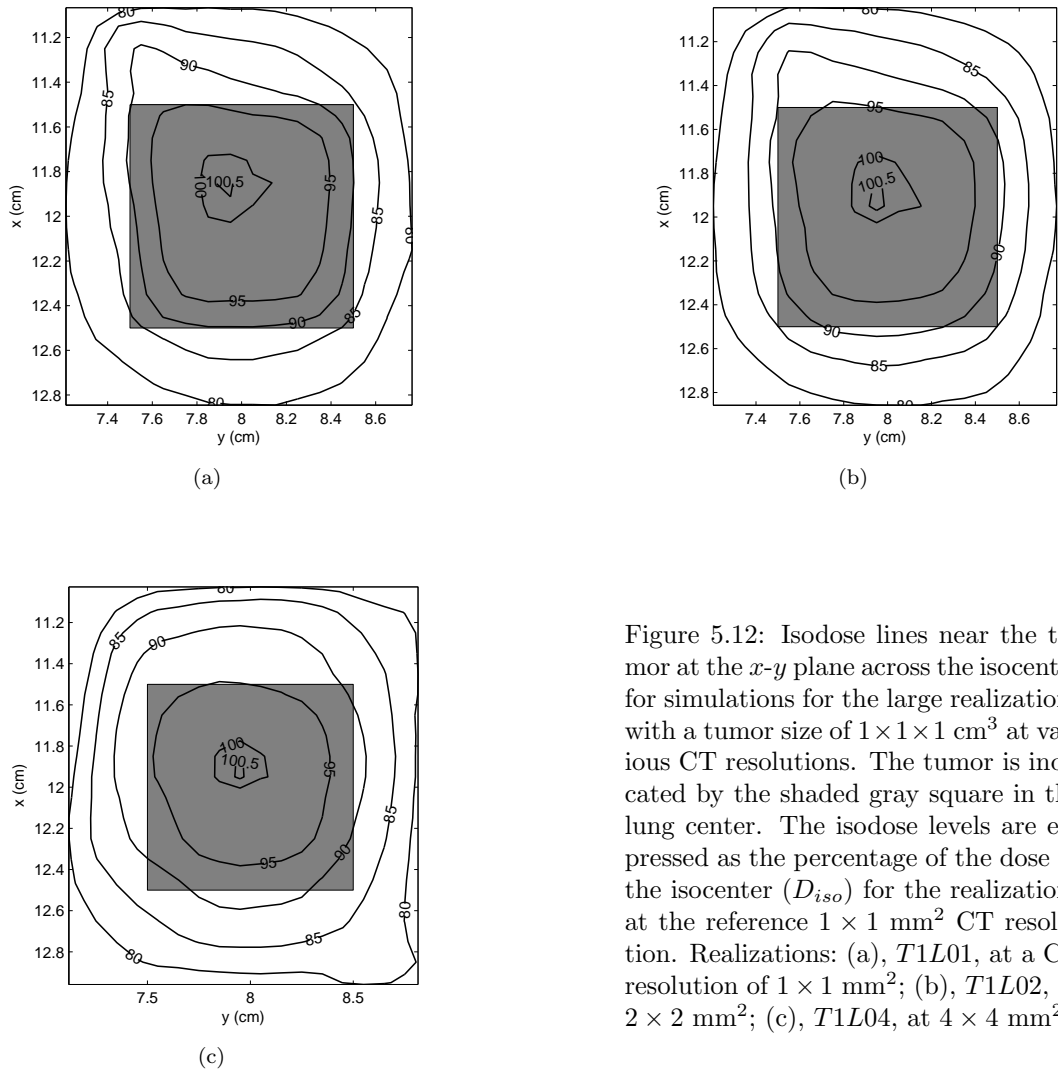
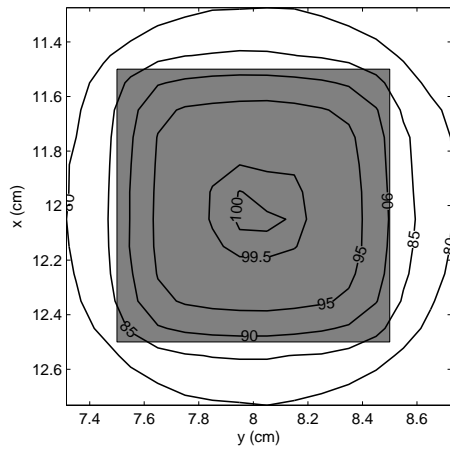
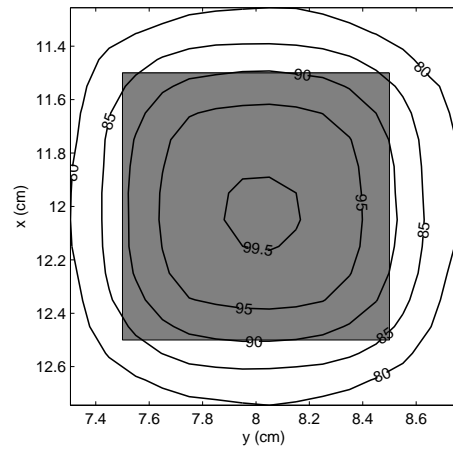


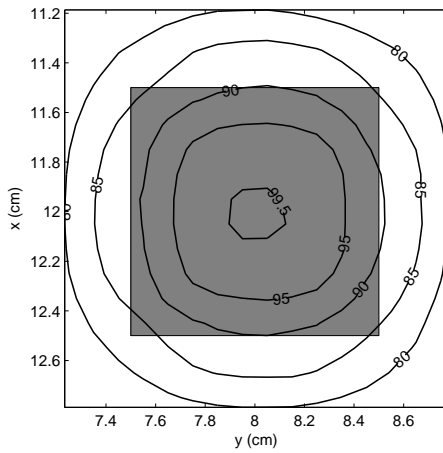
Figure 5.12: Isodose lines near the tumor at the  $x$ - $y$  plane across the isocenter for simulations for the large realizations with a tumor size of  $1 \times 1 \times 1 \text{ cm}^3$  at various CT resolutions. The tumor is indicated by the shaded gray square in the lung center. The isodose levels are expressed as the percentage of the dose at the isocenter ( $D_{iso}$ ) for the realizations at the reference  $1 \times 1 \text{ mm}^2$  CT resolution. Realizations: (a),  $T1L01$ , at a CT resolution of  $1 \times 1 \text{ mm}^2$ ; (b),  $T1L02$ , at  $2 \times 2 \text{ mm}^2$ ; (c),  $T1L04$ , at  $4 \times 4 \text{ mm}^2$ .



(a)



(b)



(c)

Figure 5.13: Isodose lines near the tumor at the  $x$ - $y$  plane across the isocenter for simulations for the small realizations with a tumor size of  $1 \times 1 \times 1 \text{ cm}^3$  at various CT resolutions. The tumor is indicated by the shaded gray square in the lung center. The isodose levels are expressed as the percentage of the dose at the isocenter ( $D_{iso}$ ) for the realizations at the reference  $1 \times 1 \text{ mm}^2$  CT resolution. Realizations: (a),  $T1S01$ , at a CT resolution of  $1 \times 1 \text{ mm}^2$ ; (b),  $T1S02$ , at  $2 \times 2 \text{ mm}^2$ ; (c),  $T1S04$ , at  $4 \times 4 \text{ mm}^2$ .

isodose lines near the tumor for the cases with a small tumor (and a small field size). In both figures, although the dose contours change little from  $CT01$  to  $CT02$ , the differences between  $CT04$  and  $CT01$  are significant for both the large and the small realizations. In Fig. 5.12c, the 85% isodose line fully covers the tumor. When the CT resolution increases to  $CT02$ : the 85% line (i) just barely covers the tumor's two lower (orientation on the paper) corners, which shows the impact of the difference in the buildup region due to different voxelizations, and (ii) shows a more asymmetric shape, which indicates a combined effect of two factors: a) the impact of the difference in the builddown region due to different voxelizations, and b) the influence from near structures resolved by the higher  $CT02$  (alternatively, hidden by the coarser  $CT04$ . See the structures close to the upper left corner of the tumor in Fig. 5.3). When the CT resolution increases to  $CT01$ , the 85% line misses the two lower corners by a small amount and shows a slightly more asymmetric shape than  $CT02$ . A same trend, with a less amplitude, occurs in the small realizations with the small tumor in Fig. 5.13. The difference in the buildup region due to different voxelizations results in a tighter dose coverage for the tumor when the CT resolution changes from  $CT04$  to  $CT01$ . However, for the cases with a large tumor with large field sizes, the changes in the dose distribution in the tumor are not significant between different CT resolutions because the buildup effect becomes small due to increased CPE.

Fig. 5.14 shows 1-D dose distributions along the  $x$ - and  $y$ -directions on the  $x$ - $y$  isocenter plane, respectively, for all realizations and CT resolutions. Fig. 5.14a and 5.14b are for the large and small realizations with a tumor size of  $1 \times 1 \times 1 \text{ cm}^3$ , at  $y = 7.7$  and  $8.0 \text{ cm}$ , respectively; while Fig. 5.14c and 5.14d are for the large and small realizations with a tumor size of  $4 \times 4 \times 4 \text{ cm}^3$ , at  $y = 7.5$  and  $8.0 \text{ cm}$ , respectively. In all figures, the solid lines are for  $CT01$ , the dashed lines for  $CT02$

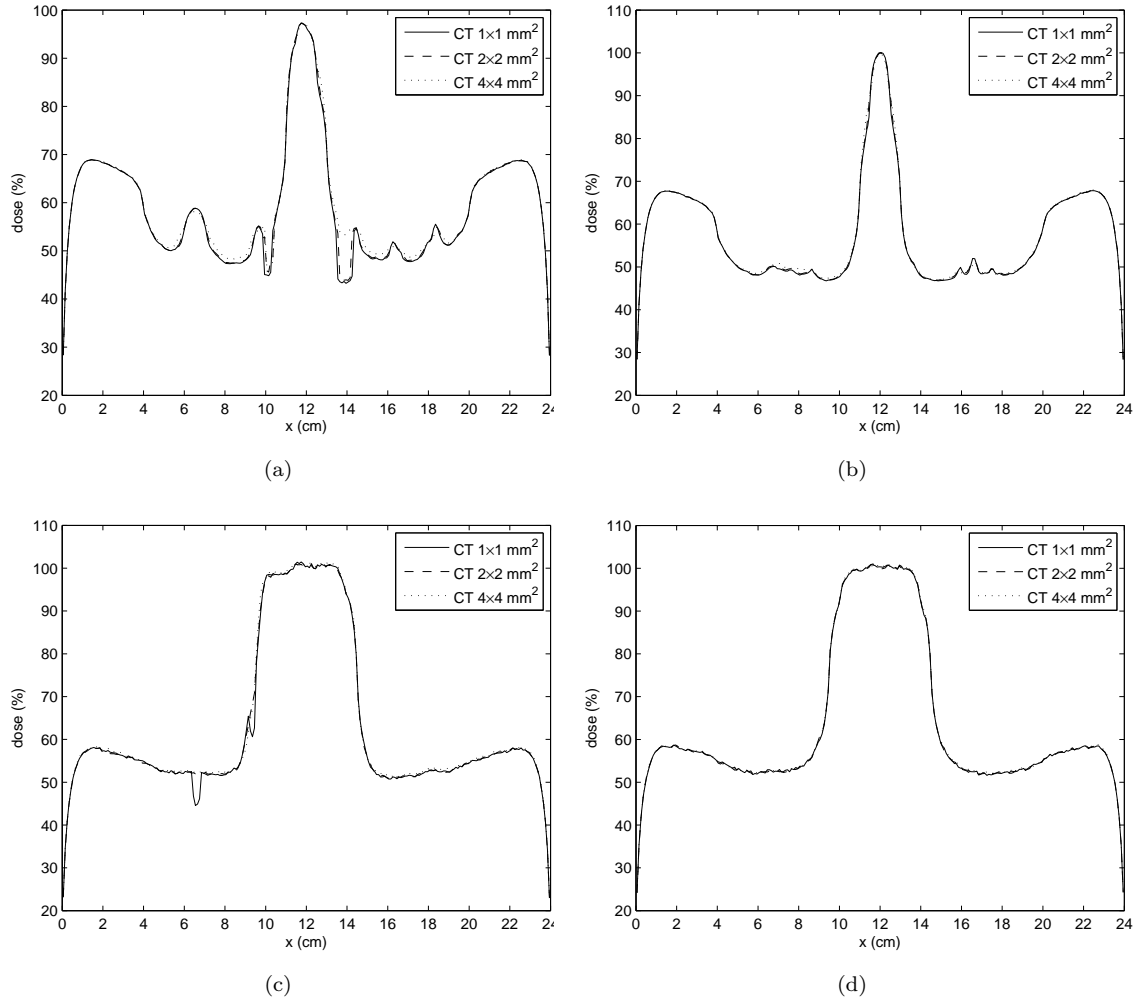


Figure 5.14: Percent depth dose curves along the  $x$ -direction at different  $y$ -values on the  $x$ - $y$  isocenter plane. (a) and (b): large and small realizations with a tumor size of  $1 \times 1 \times 1 \text{ cm}^3$ , at  $y = 7.7$  and  $8.0 \text{ cm}$ , respectively; (c) and (d): large and small realizations with a tumor size of  $4 \times 4 \times 4 \text{ cm}^3$ , at  $y = 7.5$  and  $8.0 \text{ cm}$ , respectively. In all figures, the solid lines are for a CT resolution of  $1 \times 1 \text{ mm}^2$ , the dashed lines  $2 \times 2 \text{ mm}^2$  and the dotted lines  $4 \times 4 \text{ mm}^2$



and the dotted lines for *CT04*. The locations of the depth dose curves for the large realizations are intentionally chosen so that the lines pass through as many structures as possible, while for the small realizations, we simply choose the CAX. As expected, the resolution *CT04* agrees well with *CT01* for all small realizations because the finer geometrical grid is comparable to the small structure sizes and therefore does not resolve more structures than the coarser grid. However, in large realizations, the large structure sizes are much greater in size than the fine grid. Thus, enlarging the CT voxels results in a loss of spatial resolution and an artificially smoother lung model, which in turn results in a smoother dose distribution, as can be seen from the two air regions near  $x = 10$  and  $14$  cm in Fig. 5.14a and the two air regions near  $x = 6.5$  and  $9.5$  cm in Fig. 5.14c. In the most significant cases in Fig. 5.14a, the differences in the non-air region between *T1L01* and *T1L04* are up to 2.1% of  $D_{iso}$ , while those between *T1L01* and *T1L02* are up to only 0.5%, which is comparable to statistical errors.

### 5.3.3 DVHs

Fig. 5.15 presents DVHs for the tumor and the right lung for all simulations. The dose on the abscissa is expressed as a percentage of  $D_{iso}$  of the cases at the reference CT resolution of  $1 \times 1$  mm<sup>2</sup>. Fig. 5.15a and 5.15b are for the large and small realizations with a tumor size of  $1 \times 1 \times 1$  cm<sup>3</sup>, respectively; while Fig. 5.15c and 5.15d are for the large and small realizations with a tumor size of  $4 \times 4 \times 4$  cm<sup>3</sup>, respectively. In all figures, the solid lines are for *CT01*, the dashed lines are for *CT02* and the dotted lines are for *CT04*.

The differences in DVHs for the cases with a tumor size of  $4 \times 4 \times 4$  cm<sup>3</sup> at all three CT resolutions are almost visually indiscernible for both the tumor and the right lung. While in the cases with a tumor size of  $1 \times 1 \times 1$  cm<sup>3</sup>, the DVHs for the

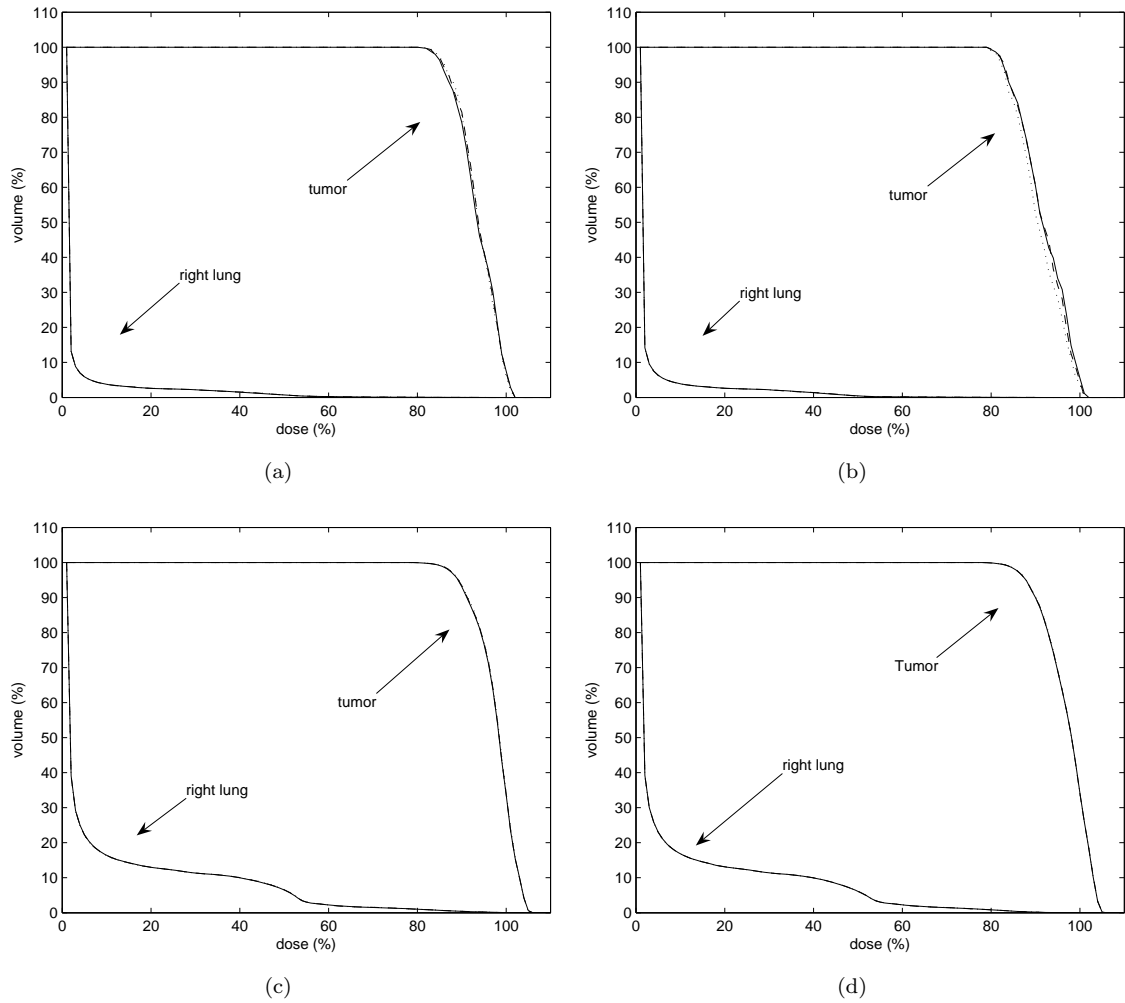


Figure 5.15: DVHs for the tumor and the right lung for all simulations. The dose is expressed as a percentage of  $D_{iso}$  of the cases at a CT resolution of  $1 \times 1 \text{ mm}^2$ . (a) and (b): large and small realizations with a tumor size of  $1 \times 1 \times 1 \text{ cm}^3$ , respectively; (c) and (d): large and small realizations with a tumor size of  $4 \times 4 \times 4 \text{ cm}^3$ , respectively. In all figures, the solid lines are for a CT resolution of  $1 \times 1 \text{ mm}^2$ , the dashed lines are for  $2 \times 2 \text{ mm}^2$  and the dotted lines are for  $4 \times 4 \text{ mm}^2$ . The differences between the lines for different resolutions are small so that most of them overlap each other.

Table 5.1: Relative absolute differential dose ( $ADD_{rel}$ ) between larger CT resolutions and the reference  $1 \times 1 \text{ mm}^2$  CT resolution for simulations with tumor size of  $1 \times 1 \times 1 \text{ cm}^3$ . The relative absolute differential dose is defined by Eq. 5.2, and is expressed as a percentage.

CT resolution ( $\text{mm}^2$ )	Tumor (%)	Right lung (%)
large realizations		
$2 \times 2$	0.181	0.125
$4 \times 4$	0.132	0.678
small realizations		
$2 \times 2$	0.155	0.046
$4 \times 4$	1.017	0.147

right lung are again on top of each other, we can see visible differences in the DVHs for the tumor when the CT resolution changes from  $CT01$  to  $CT04$ . The largest difference occurs with the small realization  $T1S04$ , where its DVH shifts from that of  $T1S01$  to the left by about 1%. This is difficult to understand initially. However, when we carefully examine the voxelization for  $T1S04$ , where the side length of the tumor is 1 cm while the CT voxel size is 4 mm, we see that the tumor cannot contain a whole number of CT voxels. The result is that the edges of the tumor are homogenized with the surrounding background tissues at a much lower density. For a small field size, where no CPE exists inside the beam, less density results in less dose. Therefore, a less dose coverage is shown for this case. The same situation could occur to the small realizations for  $T4S04$ . However, no visible difference exists for that case. By examining the voxelization for  $T4S04$ , we can see that under the voxelization we use in our simulations, the  $4 \times 4 \times 4 \text{ cm}^3$  tumor happens to cover a whole number of CT voxels. Therefore, no edge effect occurs. Should we displace the geometrical grid by one half voxel, a reduced dose coverage will occur.

The quantitative differences in terms of  $ADD_{rel}$  defined by Eq. (5.2) are shown in Tables. 5.1 and 5.2. All cases except  $T1S04$  discussed above show insignificant differences when the CT resolution changes from  $CT01$  to  $CT04$ , most of which

Table 5.2: Relative absolute differential dose ( $ADD_{rel}$ ) between larger CT resolutions and the reference  $1 \times 1 \text{ mm}^2$  CT resolution for simulations with tumor size of  $4 \times 4 \times 4 \text{ cm}^3$ . The relative absolute differential dose is defined by Eq. 5.2, and is expressed as a percentage.

CT resolution ( $\text{mm}^2$ )	Tumor (%)	Right lung (%)
large realizations		
$2 \times 2$	0.041	0.022
$4 \times 4$	0.104	0.032
small realizations		
$2 \times 2$	0.009	0.015
$4 \times 4$	0.039	0.013

are comparable to the statistical fluctuations. The reason is: for the cases with a tumor size of  $1 \times 1 \times 1 \text{ cm}^3$ , although we see local differences inside the beam, the volume with significant dose differences occupies only a small portion of the whole volume of the tumor or the lung. Therefore, the relative difference is small. This is a disadvantage of DVH: it lacks the positional information of the dose distribution due to its integral form [36]. Also, for the cases with a tumor size of  $4 \times 4 \times 4 \text{ cm}^3$ , the volume with significant dose differences now occupies a larger portion, but a larger field size provides more CPE, and thus the difference between different resolutions is smaller.

It is notable that the  $ADD_{rel}$ 's we obtain are smaller than the reported ones (see Table 4 in [34]) for a lung treatment plan. The reason is partly due to the fact that, in our lung model, we use only one material, water (see Chapter IV). Thus, the microscopic cross sections are not changed when the density changes due to voxelization. However, the reported values in [34] are from a lung treatment plan with a real patient's CT data, where the voxelization at various resolutions may not only change the density, but it may also alter the composition of the material inside the voxel thus the microscopic cross sections.

Table 5.3: Mean lung dose (MLD) and mean tumor dose (MTD) for simulations with tumor size of  $1 \times 1 \times 1 \text{ cm}^3$ . Also included are the absolute dose at the isocenter  $D_{iso}$ . The MLD and MTD are expressed as a percentage of  $D_{iso}$  of the cases at a CT resolution of  $1 \times 1 \text{ mm}^2$

CT resolution ( $\text{mm}^2$ )	$D_{iso}$ ( $\times 10^{-3} \text{ MeV/g}$ )	MLD	MTD
large realizations			
$1 \times 1$	7.504	1.899	93.18
$2 \times 2$	7.506	1.910	93.78
$4 \times 4$	7.494	1.934	93.95
small realizations			
$1 \times 1$	7.615	1.625	91.67
$2 \times 2$	7.605	1.639	92.11
$4 \times 4$	7.600	1.648	91.74

Table 5.4: Mean lung dose (MLD) and mean tumor dose (MTD) for simulations with tumor size of  $4 \times 4 \times 4 \text{ cm}^3$ . Also included are the absolute dose at the isocenter  $D_{iso}$ . The MLD and MTD are expressed as a percentage of  $D_{iso}$

CT resolution ( $\text{mm}^2$ )	$D_{iso}$ ( $\times 10^{-3} \text{ MeV/g}$ )	MLD	MTD
large realizations			
$1 \times 1$	1.303	8.314	96.71
$2 \times 2$	1.303	8.310	96.75
$4 \times 4$	1.301	8.308	96.81
small realizations			
$1 \times 1$	1.321	7.729	96.17
$2 \times 2$	1.318	7.731	96.18
$4 \times 4$	1.323	7.737	96.21

### 5.3.4 Mean Lung Dose

Tables 5.3 and 5.4 give the mean lung dose (MLD) and mean tumor dose (MTD) for simulations with tumor sizes of  $1 \times 1 \times 1 \text{ cm}^3$ ,  $4 \times 4 \times 4 \text{ cm}^3$ , respectively. Also included are the absolute dose at the isocenter  $D_{iso}$ . The MLD and MTD are expressed as a percentage of  $D_{iso}$ . The same trend can be seen: overall, the differences between  $CT01$  and  $CT02$  are minimal. Even for the cases at  $CT04$  with a small field size, the largest difference of MLD is less than 2%. For most cases, the differences are well below 1% compared with the reference CT resolution. This indicates that in terms of this crude quantity, a CT resolution of  $4 \times 4 \text{ mm}^2$  is acceptable.

## 5.4 Conclusion

We have devised a realistic lung phantom and filled the right lung with large and small realizations of the random lung model, which is developed in Chapter IV, and applied on it a three-beam (one right lateral and two opposing AP-PA beams) treatment planning for two tumor sizes ( $1 \times 1 \times 1 \text{ cm}^3$  and  $4 \times 4 \times 4 \text{ cm}^3$ ) and three CT resolutions (the reference  $1 \times 1 \text{ mm}^2$ , and two larger  $2 \times 2 \text{ mm}^2$  and  $4 \times 4 \text{ mm}^2$ ), using the code DPM. The reference CT resolution was selected based on the excellent agreement between the dose distributions calculated by PENELOPE, for a “large” realization of the detailed lung model with a tumor of  $1 \times 1 \times 1 \text{ cm}^3$ , and by DPM, for a voxelized version of this realization at the reference CT resolution, respectively. Alternatively, the detailed random lung model is very well-represented by its voxelized version at a  $1 \times 1 \text{ mm}^2$  CT resolution.

The isodose lines and the CAX depth dose curves for all cases show an increasing smoothing effect when the CT resolution changes from the reference  $1 \times 1 \text{ mm}^2$  to  $4 \times 4 \text{ mm}^2$ . The dose distributions between the cases at  $1 \times 1 \text{ mm}^2$  and  $2 \times 2 \text{ mm}^2$  are almost identical, especially for the cases with a larger tumor of  $4 \times 4 \times 4 \text{ cm}^3$ . However, for the large realization with a small tumor of  $1 \times 1 \times 1 \text{ cm}^3$ , noticeable differences occur in the 90% and 50% isodose lines between the reference CT resolution and the  $4 \times 4 \text{ mm}^2$  resolution. Also, differences of up to 2.1% of  $D_{iso}$  can be observed in the non-air regions along the CAX. The difference in the buildup region in the tumor due to different geometrical resolutions causes less dose coverage for small tumors when the CT resolution decreases from  $4 \times 4 \text{ mm}^2$  to  $1 \times 1 \text{ mm}^2$ . This may result in cold spots if the treatment planning for the lung with small tumors are based on a patient’s geometry using large CT voxels.

The DVHs and the mean doses for both the tumor and the lung show little differences between the reference and the  $2 \times 2 \text{ mm}^2$  CT resolutions for all cases. This is almost the case for the differences between the reference and the  $4 \times 4 \text{ mm}^2$  CT resolutions. However, the visible shift (corresponding to a slightly  $> 1\%$  relative difference in DVH) in the DVH for a small realization with a small tumor size reveals a potential error caused by the homogenization of the edge of a heterogeneity by a large voxel, such as the interface between the tumor and the surrounding tissue. This may be especially problematic for small tumors.

These results suggest that accurate dose calculations may be obtained for the lung in a multi-beam setup by using a CT resolution of  $2 \times 2 \text{ mm}^2$ , which is consistent with the resolutions suggested to obtain good accuracy published previously [26, 32, 34]. Using a finer geometrical grid may not gain extra accuracy, while costing extensive memory storage and calculation time. Our results show that for most cases, the  $4 \times 4 \text{ mm}^2$  CT resolution does not introduce significant deviations from the reference  $1 \times 1 \text{ mm}^2$  resolution, but one should be cautious about possible situations where large geometrical voxels could lead to significant systematic errors.

Also, we notice the possible artifact caused by the square box-like shape of the tumor, which, in particular with large tumor sizes, can happen to be aligned exactly even with large geometrical voxels. This will result in “artificially” weakened differences between various CT resolutions at the interface between the tumor and the tissue, and hence introduce bias into our conclusions.

## CHAPTER VI

### Conclusions

The goal of this thesis is to assess the impact of the detailed, highly heterogeneous structures of the human lung on dose calculations, by building and utilizing a realistic “random” lung model suitable for computer simulations. Here we summarize the major results of our work, and we propose some potential future work.

To achieve our goal, we proceeded in this thesis as follows:

1. Chapter I provided the motivation and detailed background of our work and outlined the strategy of our research;
2. Chapter II provided the quantitative description of the lung’s anatomy and several relevant parameters used to construct the lung model.
3. We developed in Chapter III a new “atomic mix” theory for particles with a strongly forward-peaked scattering differential cross section. In this new theory, the length scale for homogenizing a heterogeneous medium is raised from the mean free path (MFP) to the *transport* MFP of the particle, which for electrons is orders of magnitude larger than the MFP.
4. In Chapter IV we developed a new  $2\frac{1}{2}$ -D “random” lung model, based on the lung’s anatomical data and the new “atomic mix” theory, and we compared the



dose distributions within different realizations of our “random” lung model to the dose distributions for an “atomic mix” lung model in a one-beam phantom setup using the Monte Carlo method.

5. In Chapter V, we applied the random lung model in a more realistic lung phantom, with a treatment planning-like multiple beam setup, in order to provide an optimal CT resolution for the Monte Carlo lung dose calculations.

The new “atomic mix” theory for charged particles with highly forward-peaked scattering, was discussed in Chapter III.

We started with a simplified one-dimensional, energy-independent Boltzmann transport equation for the angular flux  $\Psi(x, \mu)$ . We separated the differential scattering cross section  $\Sigma_s(x)$  into (i) a “hard” component  $\Sigma_h(x)$ , which physically accounts for the less-frequent “catastrophic” inelastic collisions, in which the energy and the direction of flight of the charged particles have an  $O(1)$  change in a single collision; and (ii) a “soft” component  $\Sigma_r(x)$ , which physically accounts for the much more-frequent elastic interactions through the Coulomb force, in which the charged particles have small changes in energy and the direction of flight in a single collision. We made the first assumption that the soft collision operator  $L_r$  has a forward-peaked differential scattering kernel  $\Sigma_r(x, \mu, \mu')$  around  $\mu' \approx \mu$ . The approach of Pomraning was then applied to approximate  $L_r$  by its Fokker-Planck limit  $L_{r,FP}$ , which contains the key parameter  $\Sigma_{r,tr}$ , the *transport* differential cross section, or alternatively, the reciprocal of the *transport* MFP. The more forward-peaked the differential scattering kernel is, the better  $L_{r,FP}$  approximates  $L_r$ . In this way, we obtained the well-known Boltzmann-Fokker-Planck (BFP) equation.

Then we made a different assumption concerning the properties of the spatially heterogeneous medium consisting of “chunks” of two materials:  $\frac{\lambda_h}{\lambda_{r,tr}} = O(1)$  and

$\frac{\lambda_{ch}}{\lambda_{r,tr}} \equiv \epsilon \ll 1$ . Here  $\lambda_h = \text{typical value of } \frac{1}{\Sigma_h(x)}$ , the MFP between consecutive hard collisions;  $\lambda_{r,tr} = \text{typical value of } \frac{1}{\Sigma_{r,tr}(x)}$ , the transport MFP for soft collisions, over which the electron's direction of flight can vary an  $O(1)$  amount; and  $\lambda_{ch} = \text{typical width of a chunk in such a heterogeneous medium}$ . This assumption states that the MFP of the fewer hard collisions is comparable to the transport MFP of the dominant soft collisions, while the chunk sizes of the medium is small compared to this transport MFP. By introducing two dimensionless spatial variables: (i)  $y \equiv \frac{x}{\lambda_{ch}}$ , which is used to express the “fast” component of  $\Psi$  which varies an  $O(1)$  amount over a typical chunk size  $\lambda_{ch}$ ; and (ii)  $z \equiv \frac{x}{\lambda_{r,tr}}$ , which is used to express the “slow” component of  $\Psi$  which varies an  $O(1)$  amount over a typical transport MFP  $\lambda_{r,tr}$ ,  $\Psi(x, \mu)$  was mathematically expressed in terms of  $y$  and  $z$  as  $\psi(y, z, \mu)$ . We then applied a formal asymptotic analysis to the BFP equation, which is now expressed in terms of  $\psi(y, z, \mu)$ , and obtained its atomic mix approximation, in which the cross sections are volume-averaged over the whole heterogeneous medium. The solution of the BFP equation,  $\Psi(x, \mu)$ , and the solution of its atomic mix approximation,  $\hat{\Psi}(x, \mu)$  (the leading order term, in terms of  $x$ , of the asymptotic expansion of  $\psi(y, z, \mu)$ ), satisfy:  $\Psi(x, \mu) = \hat{\Psi}(x, \mu) + O(\epsilon)$ . Therefore, we theoretically demonstrated that for charged particle transport in a heterogeneous medium, if the collisions between the charged particles and the background medium are dominated by forward-peaked scattering, and the chunk sizes of the medium are small compared to the transport MFP of the charged particles, the atomic mix approximation will accurately predict the behavior of a charged particles in a heterogeneous medium.

The transport of electrons in materials encountered in radiotherapy is dominated by soft collisions, which are highly forward-peaked. To numerically verify our new atomic mix theory, we constructed a “droplet” model, which consists of “chunks” of

two materials, air and water, with random locations and various sizes. We then used the Monte Carlo code PENELOPE to transport electrons and photons in different realizations of the droplet model, with different sizes of “droplets”, and compared the dose distributions to the atomic mix approximation of this droplet model. The beams are monodirectional, and monoenergetic (2 MeV for electrons and 3.4 MeV for photons), with a small field size (radius = 1 cm), and in both beams charged particle equilibrium (CPE) does not exist. Our results show that as the chunk size of the droplet model decreases, the dose distributions limit to that of the atomic mix model. When the chunk size is small compared to the transport MFP of the electrons, the differences between different realizations of the heterogeneous droplet model and the corresponding atomic mix case are insignificant. This shows an excellent agreement with the new atomic mix theory.

Thus, the new atomic mix theory raises the length scale of homogenizing a heterogeneous medium by orders of magnitude, from the MFP to the transport MFP. This suggested a practical way to build a realistic model for the highly heterogeneous lung, which, according to traditional transport theory, is practically impossible due to the huge number of structures greater in size than a MFP.

In Chapter IV, we developed a new random heterogeneous 2  $\frac{1}{2}$ -D lung model. This model has some unique features:

1. It is “mixed”. Structures smaller than a chosen (guided by our theory and the lung’s anatomical characteristics) threshold size of 0.05 cm, are homogenized into an atomic mix background, with an adjusted density, while structures larger than the threshold size are explicitly modeled.
2. The explicitly modeled structures are not in 3-D form. Instead, they are modeled as 2-D cylinders in order to make this model realistic yet simple.

3. The mean density of the random lung model is preserved.

We then used PENELOPE to perform Monte Carlo dose calculations in a lung phantom, for a single photon beam of two energies (6 and 18 MV), and four different field sizes ( $1 \times 1$ ,  $5 \times 5$ ,  $10 \times 10$  and  $20 \times 20$  cm<sup>2</sup>). The lung region of the phantom was filled with different realizations (two “large” and two “small”) of the random lung model, which represent different scenarios that might be encountered in a lung. We found that the dose perturbations (compared to the atomic mix lung) caused by the random structures of the lung depend on various factors, including the beam’s energy, the field size, and the locations of the structures relative to the beam’s pathway. A significant local perturbation exceeding 30% of the  $D_{max}$  was observed with an 18 MV beam of  $1 \times 1$  cm<sup>2</sup> field size, at a local large blood vessel. As the field size increases, the perturbations at the same location decrease, due to the gradual recovery of CPE. However, the dose reduction in the downstream tissue layers (the “shadow” region behind the lung) is mainly determined by the attenuation of the primary photons along the beam’s pathway, which in turn is determined by the density concentration along the beam’s pathway and thus, by the locations and sizes of the random structures of the lung. This dose reduction is not fully compensated by the increased in-scattering as the field size increases, so the shadow regions also occurs with large field size. A dose reduction as high as 7% of  $D_{max}$  was found.

Since the modern treatment planning is mostly CT-based, we also compared the CAX depth dose between a large realization of the detailed lung model and its voxelized version at various CT resolutions. A 5% of  $D_{max}$  maximum difference was found in the non-air region at a 4 mm CT resolution. This indicates a need to investigate the effect of various CT resolutions on the dose calculations in the lung, and this was done in Chapter V.

In Chapter V, we devised a realistic lung phantom with a tumor of two sizes embedded inside the lung, for a treatment-like multiple-beam dose calculations, using the Monte Carlo code DPM. The lung region with the tumor was filled with representative realizations of our random lung model and then voxelized to various geometrical resolutions. A reference CT resolution of  $1 \times 1 \text{ mm}^2$  was determined to be sufficiently fine to represent the heterogeneous lung model, based on the excellent agreement between the dose distributions calculated for a detailed realization of the random lung model and its voxelized version at this selected reference resolution.

The Monte Carlo dose calculations were then performed for two realizations (a large and a small one) of the random lung model, with two tumor sizes ( $1 \times 1 \times 1 \text{ cm}^3$  and  $4 \times 4 \times 4 \text{ cm}^3$ ) and three CT resolutions ( $1 \times 1 \text{ mm}^2$ ,  $2 \times 2 \text{ mm}^2$ , and  $4 \times 4 \text{ mm}^2$ ). Various dose evaluation tools, including the CAX depth doses, the isodose lines, the DVHs, and the mean doses, were used to assess the dose distributions among the different combinations of realization, tumor size, and CT resolution.

In all combinations, the differences in dose distributions between the CT resolutions of the  $2 \times 2 \text{ mm}^2$  and the reference  $1 \times 1 \text{ mm}^2$  were insignificant. This was almost the case for the differences between the CT resolutions of the  $4 \times 4 \text{ mm}^2$  and the reference  $1 \times 1 \text{ mm}^2$ . However, in some situations, noticeable differences were found. The 90% and 50% isodose lines showed a significant difference for the large realization with a small tumor of  $1 \times 1 \times 1 \text{ cm}^3$ . Also, in the CAX depth dose, differences up to 2.1% of  $D_{iso}$  were observed in the same case. For small tumor sizes, the dose differences in the tumor buildup region, resulting from different voxelizations, caused a tighter dose coverage as the CT resolution decreases. Thus, treatment planning for small tumors based on large CT voxels may overestimate the dose coverage for the tumor and hence lead to possible cold spots. The slightly  $> 1\%$  relative difference in

DVH between the  $4 \times 4 \text{ mm}^2$  and the reference  $1 \times 1 \text{ mm}^2$  CT resolution for a small realization with a small tumor indicated a possible error introduced by the inexact delineation of the tumor boundaries by box-like geometrical voxels, which becomes more severe with increasing CT voxel sizes. Also, the unrealistic box-like shape of the tumor used in our calculations may result in “artificially” less difference between different CT resolutions, due to the possible exact alignment of the boundary of the CT voxels with the tumor.

Therefore, based on our random lung model, the results from Chapter V suggest that for the lung treatment planning, a  $2 \times 2 \text{ mm}^2$  CT resolution could be optimal: (i) it generates no significant differences from the reference  $1 \times 1 \text{ mm}^2$  resolution, compared to the larger  $4 \times 4 \text{ mm}^2$  one, where significant differences were found in certain situations, while (ii) it is more cost effective than the  $1 \times 1 \text{ mm}^2$  resolution in terms of the calculation time and the memory consumption.

Our realistic  $2 \frac{1}{2}$ -D random lung model, developed based on the lung’s anatomy and our new atomic mix theory, has been used in this thesis as a tool to evaluate the impact of the highly heterogeneous lung’s structures on dose calculations in the lung. However, the current lung model is a static, rigid-body model. In the future, our model could be improved by incorporating breathing-related motions and deformations, and/or evolving it into a real 3-D form, with the structures modeled as their spatially correlated 3-D “tree” shapes.

Our dose calculations were exclusively performed with the Monte Carlo method, which is the most accurate dose calculation method for electron/photon transport. As stated in Chapter I, however, for practical reasons, the most widely used dose calculation algorithm in treatment planning is convolution/superposition (CV/SP), while the pencil beam methods, e.g., the ETAR method, are still used in many

treatment planning systems. It is likely that these commonly-used algorithms cannot accurately deal with the lung's detailed spatial heterogeneities resolved by a fine CT grid. Therefore, possible additional systematic errors caused by the less sophisticated heterogeneity correction methods in these algorithms can be foreseen. A natural future extension of our work would be to investigate the effect of different approximate dose calculation methods on the dose distributions in the heterogeneous lung, using our random lung model. There has been some previously-published work investigating aspects of these questions [64, 120].

Also, our simulations used only uniform open fields. We observed that for the heterogeneous lung, the largest dose perturbations are associated with small field sizes, in which lateral CPE is not present. As the intensity modulated radiation therapy (IMRT), in which the beam's intensity varies across the field, and hence dose calculations are usually conducted beamlet by beamlet, gains more popularity in treatment planning, it will be valuable to study the effect of the lung's heterogeneity on the IMRT lung treatment planning, using our realistic random lung model.

We have seen that when the geometrical grid became finer, and thus more structures are resolved, the dose coverage for the tumor became tighter, and cold spots could occur. This indicates that a possible remedy for the effect of unresolved detailed structures at a certain CT resolution could be to add an extra margin to the target (alternatively, increase the conformal field size) to account for this effect. This is another possible application of our lung model that needs to be investigated in the future.

In this thesis, all our estimates of dose were obtained from calculations using the Monte Carlo method. No measurements were involved. Although the Monte Carlo codes that we used have been benchmarked against measurement for a wide range of

situations and have shown very good agreement, it would strengthen our conclusions if dosimetric measurements could be performed to validate them. However, accurate experiments for physical phantoms containing small and numerous random heterogeneities are extraordinarily difficult. Because the Monte Carlo codes PENELOPE and DPM are based on the first-principle physics of photon and electron transport, we are confident that the results obtained by these codes accurately represent the results that would be seen in measurements if it were feasible to perform them.



## APPENDICES

## APPENDIX A

## Computer Codes Used in This Thesis

The computer codes used in this thesis, which are listed below, can be obtained from the author (lliang@umich.edu) upon request.

### A.1 Fortran 90 codes to generate the geometry of the “droplet” model in Chapter III

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! Generate cylindrical "droplet" geometry for NewPencyl.
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
program Main

use InpGen_Globals

implicit none

character (len=1) :: yn1

!!!

! -- Title
write(*,100)
100 format(/,72('-'))
write(*,'(a)') 'The title for this job ( <= 128 characters )'
write(*,'(a,$)') '-> '
read(*,'(a128)') title_input

! -- Front buildup layer
write(*,100)
write(*,'(a,$)') 'Want a front water layer? (y/n) '
read(*,*) yesno
if(yesno=='y'.or.yesno=='Y') then
front = .true.
1 write(*,'(/,2x,a)') 'ZF: depth of the front water layer'
write(*,'(4x,a,f6.2,a)') 'The default ZF is',ZF,' cm'
write(*,'(6x,a,$)') 'Want to change? (y/n) '
read(*,*) yn1
select case(yn1)
case ('y', 'Y')
write(*,'(4x,a,$)') 'Then input ZF (cm) -> '
read(*,*) ZF
if(ZF<=0d0) then

```

```

        write(*,'(a)') '>>> ZF should greater than 0!'
        go to 1
    end if
end select
2 write(*,'(/,2x,a)') 'Choose a matter (C1, C2 defined in PENELOPE)'
write(*,'(4x,a)') '1: water (C1=C2=0.05)'
write(*,'(4x,a)') '2: water (C1=C2=0.1)'
    write(*,'(4x,a,$)') '-> '
read(*,*) AorW
if(AorW==1) then
    mfrontlyr = water
else if(AorW==2) then
    mfrontlyr = water_max
else
    write(*,'(a)') '>>> Input 1 or 2, please'
    go to 2
end if
end if

! -- Rear backscatter layer
write(*,100)
write(*,'(a,$)') 'Want a rear water layer? (y/n) '
read(*,*) yesno
if(yesno=='y'.or.yesno=='Y') then
    rear = .true.
3 write(*,'(/,2x,a)') 'ZR: depth of the rear water layer'
write(*,'(4x,a,f6.2,a)') 'The default ZR is',ZR,' cm'
    write(*,'(6x,a,$)') 'Want to change? (y/n) '
read(*,*) yn1
select case(yn1)
case ('y', 'Y')
    write(*,'(4x,a,$)') 'Then input ZR (cm) -> '
    read(*,*) ZR
    if(ZR<=0d0) then
        write(*,'(a)') '>>> ZR should greater than 0!'
        go to 3
    end if
end select
4 write(*,'(/,2x,a)') 'Choose a matter (C1, C2 defined in PENELOPE)'
write(*,'(4x,a)') '1: water (C1=C2=0.05)'
write(*,'(4x,a)') '2: water (C1=C2=0.1)'
    write(*,'(4x,a,$)') '-> '
read(*,*) AorW
if(AorW==1) then
    mrearilyr = water
else if(AorW==2) then
    mrearilyr = water_max
else
    write(*,'(a)') '>>> Input 1 or 2, please'
    go to 4
end if
end if

! -- Middle layer
write(*,100)
write(*,'(a)') 'The middle layer: [central tube] + [lung] + [surrounding matter]'

! -- length of the lung (i.e. middle layer)
write(*,'(/,2x,a,f6.2,a3)') 'The default length of the lung is: ',Z,' cm'
write(*,'(4x,a,$)') 'Want to change? (y/n) '
read(*,*) yesno
if(yesno=='y'.or.yesno=='Y') then
    write(*,'(4x,a,$)') '-> '
    read(*,*) Z
end if

! -- radius of the whole phantom
write(*,'(2x,a,f6.2,a3)') 'The default radius of the whole phantom is: ',R,' cm'

```

```

write(*,'(4x,a,$)') 'Want to change? (y/n) '
read(*,*) yesno
if(yesno=='y'.or.yesno=='Y') then
  write(*,'(4x,a,$)') '-> '
read(*,*) R
end if

! -- radius of the lung
write(*,'(2x,a,f6.2,a3)') 'The default radius of the lung is: ',RB, ' cm'
write(*,'(4x,a,$)') 'Want to change? (y/n) '
read(*,*) yesno
if(yesno=='y'.or.yesno=='Y') then
49 write(*,'(2x,a,f6.2,a,$)') 'New value (cm; <=',R,') -> '
read(*,*) RB
  if(RB>R) then
    write(*,'(a,1x,g12.6,a5)') '>>> RB not allowed to >', R, 'cm'
  go to 49
  end if
end if
if(RB<=0d0) then
  RB = 0d0
dr = 0d0
print_RB = 0
else if(RB==R) then
  print_RB = 1
else
  print_RB = 2
end if

! -- lung composition
5 write(*,'(/,2x,a)') 'The random part is made of'
write(*,'(4x,a)') '1: water+air'
write(*,'(4x,a)') '2: water+void'
write(*,'(4x,a,$)') '-> '
read(*,*) AorV
if(AorV==2) then
  air = void
air_max = void
else if(AorV>2) then
  write(*,'(a)') '>>> Input 1 or 2, please'
go to 5
end if

! -- lung density
51 write(*,'(/,2x,a)') 'The mass density of the lung (g/cm^3)'
read(*,*) rho(am)
rho(am_max) = rho(am)
fw = (rho(am)-rho(air))/(rho(water)-rho(air))

! -- Surrounding scatter matter
6 write(*,'(/,2x,a)') 'The matter surrounding the random part?'
write(*,'(4x,a)') '1: water (C1=C2=0.05)'
write(*,'(4x,a)') '2: water (C1=C2=0.1)'
write(*,'(4x,a,$)') '-> '
read(*,*) AorW
if(AorW==1) then
mouterring = water
else if(AorW==2) then
mouterring = water_max
else
  write(*,'(a)') '>>> Input 1 or 2, please'
go to 6
end if

call ZR_Layers

end program Main

```

```

!*****
! 2-D layers along both longitudinal and radial direction
!*****
subroutine ZR_Layers

use InpGen_Globals

implicit none

integer :: i, j, k

! z direction
real (8) :: ztmp, zstart
integer :: nz, nend
! r direction
real (8) :: r1, r2, rtmp, rstart
!
logical :: iki=.false., ksi_again=.false.
real (8) :: ksi
!
integer :: i3_10th, i7_10th, Cell_count
real (8) :: v_lung, v_tissue, v_air, v_count
!
integer :: NB, nzstart, nzend, nrstart, nrend
!
character (len=64) :: buffer
integer (1) :: ibuffer

!!!

100 format(/,74('-'))

29 continue

19 write(*,'(/,2x,a,$)') 'The cell depth (cm): -> '
read(*,*) zl

if(RB<=0d0) go to 69

59 write(*,100)
write(*,'(a,$)') 'dr of each binary cell (cm) -> '
read(*,*) dr
if(dr>RB) then
write(*,'(a,1x,g12.6,a5)') '>>> dr not allowed to >', RB, 'cm'
go to 59
else if(RB>0.and.dr<=0) then
write(*,'(a)') '>>> dr should > 0 cm when RB > 0'
go to 59
end if

69 write(*,100)
write(*,'(/,a,$)') 'ZR- realization #-> '
read(*,*) rln_zr

! -- determine the layers' z-coordinates

nk = ceiling(Z/zl)
allocate(ZC(0:nk))
ZC(0) = 0d0
do i=1, nk-1
ZC(i) = i*zl
end do
ZC(nk) = Z

! -- determine the rings within each layer

nr = ceiling(RB/max(dr,1d-35))

```

```

if(print_RB==1) then
  allocate(RC(nk,1:nr+1))
  RC(:,1) = 0; RC(:,nr+1) = R
else if(print_RB==2) then
  allocate(RC(nk,1:nr+2))
  RC(:,1) = 0; RC(:,nr+1) = RB; RC(:,nr+2) = R
end if
if(print_RB==1) then
  allocate(RC(nk,0:nr+1))
  RC(:,0) = 0; RC(:,1) = 0; RC(:,nr+1) = R
else if(print_RB==2) then
  allocate(RC(nk,0:nr+2))
  RC(:,0) = 0; RC(:,1) = 0; RC(:,nr+1) = RB; RC(:,nr+2) = R
end if

do i=2, nr
  RC(:,i) = (i-1)*dr
end do

! -- fill the phantom
allocate(matter(nk,nr))

v_lung = Z*(RB)**2
v_tissue = v_lung*fw
v_air = v_lung - v_tissue

call InpGen_Droplet(water, air, v_air, rlnz_zr, v_count)
write(*,999) 'The extra air is ', (v_count-v_air)/v_air*100, &
  '% of the specified volume'
write(*,999) 'The lung density of this case is ', &
  (v_count*rho(air)+(v_lung-v_count)*rho(water))/v_lung/rho(am)*100, &
  '% of the specified one'
999 format(/,a, f8.5, a)

call Output

write(*,'(/,a,$)') 'Another realization? (y/n) '
read(*,*) yesno
if(yesno=='y'.or.yesno=='Y') go to 69

end subroutine ZR_Layers

!*****
! Subroutine for filling the matter with droplets
!*****

subroutine InpGen_Droplet(Mbase, Mfill, VFill, Rlztm, VCount)

use InpGen_Globals

implicit none

integer (1), intent(in) :: MBase, MFill
real (8), intent(in) :: VFill
integer (8), intent(in) :: Rlztm

real (8), intent(out) :: VCount

integer :: i, j

!!! instructions

matter = Mbase
Vcount = 0d0

do while (VCount < VFill)

```

```

i = ceiling(nk*rang())
j = ceiling(nr*rang())
if(i==0.or.j==0) cycle
if(matter(i,j)==Mbase) then
  matter(i,j) = Mfill
  VCount = VCount + (ZC(i)-ZC(i-1))*(RC(i,j+1)**2-RC(i,j)**2)
end if
end do

end subroutine InpGen_Droplet

subroutine Output

use InpGen_Globals

implicit none

integer :: i, j, k
integer, allocatable :: jm(:), nwater_z(:), nwater_r(:)
integer :: jmax, jmin, jtmp

!!!

100 format(/,72('-'))

write(*,100)

! -- allocate the actual radial grid
allocate(RCM(nk,jmax+1), matterM(nk,jmax))

! -- merge the identical neighbors
RCM(:,1) = RC(:,1)
do i=1, nk
  jtmp = 2
  matterM(i,1) = matter(i,1)
  do j=2, nr
    if(matter(i,j)/=matter(i,j-1)) then
      RCM(i,jtmp) = RC(i,j)
      matterM(i,jtmp) = matter(i,j)
      jtmp = jtmp + 1
    end if
  end do
  RCM(i,jm(i)+1) = RC(i,nr+1)
end do

open(unit=10, file='merged.txt', action='WRITE', status='REPLACE')

write(10,'(a6,1x,a128)') title, title_input
write(10,'(a6)') gstart

if(front) then
  write(10,'(a6,6x,2g21.13,2x,i5)') layer, ZC(0)-ZF, ZC(0), 0
  write(10,'(a6,6x,2g21.3)') center, 0d0, 0d0
  write(10,'(a6,3x,i1,2x,2g21.13)') cylind, mfrontlyr, 0d0, R
end if

do k=1, nk
  write(10,'(a6,6x,2g21.13,2x,i5)') layer, ZC(k-1), ZC(k), k
  write(10,'(a6,6x,2g21.3)') center, 0d0, 0d0
  do j=1, jm(k)
    write(10,'(a6,3x,i1,2x,2g21.13)') cylind, matterM(k,j), RCM(k,j), RCM(k,j+1)
  end do
  if(print_RB==2) then
    write(10,'(a6,3x,i1,2x,2g21.13)') cylind, moutering, RC(k,nr+1), RC(k,nr+2)
  end if
end do

if(rear) then

```

```

write(10,'(a6,6x,2g21.13,2x,i5)') layer, ZC(nk), ZC(nk)+ZR, nk+1
write(10,'(a6,6x,2g21.3)') center, Od0, Od0
write(10,'(a6,3x,i1,2x,2g21.13)') cylind, mrearlyr, Od0, R
end if

write(10,'(a6)') gend

close(10)

deallocate(ZC, RC, RCM, matter, matterM, jm, nwater_z, nwater_r)

end subroutine Output

!
! global variables
!
module InpGen_Globals

use InpGen_Pars

character*128 :: title_input

real (8) :: ZF=ZF_default, ZR=ZR_default, &
           Z=10d0, ZO, &
           R=R1510, RB=RL510, RBO, &
           fw ! fw (volume fraction of water)

integer (8) :: realization, rlzn_zr
integer :: AorV, AorW
integer (1) :: void=0, water=1, air=2, am=3, water_max=4, air_max=5, am_max=6
real (8) :: Rho(0:6), DMean
data Rho/Od0, 1d0, 1.20479d-3, 0.201388d0, 1d0, 1.20479d-3, 0.201388d0/
integer (1) :: mfrontlyr, mrearlyr, mouterring
character (len=1) :: yesno, yesno2
logical :: front=.false., rear=.false., wfilled=.true.
integer (1), allocatable :: matter(:,,:), matterM(:,,:)
! binary geometry
integer :: nk, nk0, nr, nr0, nbuff
real (8) :: dz, dr, dr0, delZ, delR
real (8) :: z1
real (8), allocatable :: RCO(:,), RC(:,,:), RCM(:,,:), RMap(:)
real (8), allocatable :: ZC(:,), ZMap(:)
! flags
integer :: print_RB=0

end module InpGen_Globals

module InpGen_Pars

implicit none

! i/o unit numbers:
integer,parameter :: io_i = 31 ! = I/O unit for the problem setup(input) file.
integer,parameter :: io_o = 33 ! = I/O unit for the output file.

!
real(8), parameter :: ONE = 1d0
real(8), parameter :: ZERO = 0d0
real(8), parameter :: huge = 1.0d+36 ! = A very large number.
real(8), parameter :: PI = 3.1415926535897932d+0

!
character*6, parameter :: title='TITLE', gstart='GSTART', gend='GEND', &
                        layer='LAYER', center='CENTRE', cylind='CYLIND'
!
real (8), parameter :: ZF_default=3d0
real (8), parameter :: ZR_default=5d0

```



```

real (8), parameter :: R1510=15.1d0, RL510= 5.1d0
end module InpGen_Pars

```

The random number generator (RNG) “rang()” in the codes should be supplied by the users.

## A.2 Fortran 77 PENELOPE main program, based on “Pencil” from the PENELOPE distribution, and modified by Liang for the “droplet” geometry in this thesis

```

C *****
C                               MAIN PROGRAM
C      modified by L. Liang based on Pencil                               C
C *****

      implicit DOUBLE precision (A-H,O-Z), integer*4 (I-N)

C  -- command line arguments relevant
integer narg
character*32 argi, argo, argsfx
logical exists
integer len_sfx
C
character*2 LIT
character*32 PFILE,PFILED,PFILER
character*128 BUFFER
character*6 KWORD,
1  KWTITL,KWKPARG,KWSENE,KWSPEC, KWSEXT,KWSHEI,KWSRAD,KWSPOS,
2  KWSDIR,KWSAPE,KWNMAT,KWSIMP, KWPFNA,KWNBE ,KWNBTH,KWNBPH,
3  KWNBZ ,KWNBR ,KWABSE,KWNBTL, KWDO2D,KWIFOR,KWRESU,KWDUMP,
4  KWNSIM,KWTIME,KWRSEE,KWCOMM,

C  -- ADDED KEYWORDS BY LIANG
1  KWFOUT, KWNITR, KWDZDO, KWDRDO
parameter(
1  KWTITL='TITLE ',KWKPARG='SKPAR ',KWSENE='SENERG',KWSPEC='SPECTR',
2  KWSEXT='SEXTND',KWSHEI='STHICK',KWSRAD='SRADII',KWSPOS='SPOSIT',
3  KWSDIR='SDIREC',KWSAPE='SAPERT',KWNMAT='NMAT ',KWSIMP='SIMPARG',
4  KWPFNA='PFNAME',KWNBE ='NBE ',KWNBTH='NBTH ',KWNBPH='NBPH ',
5  KWNBZ ='NBZ ',KWNBR ='NBR ',KWNBTL='NBTL ',KWABSE='ABSEN ',
6  KWDO2D='DOSE2D',KWIFOR='IFORCE',KWRESU='RESUME',KWDUMP='DUMPTO',
7  KWNSIM='NSIMSH',KWTIME='TIME ',KWRSEE='RSEED ',KWCOMM=' ')

C  -- ADDED KEYWORDS' PARAMETERS BY LIANG
parameter (KWFOUT='FULOUT') ! full output control
parameter (KWNITR='NINTER') ! timing output interval
parameter (KWDZDO='DZDOSE') ! dz for dose grid
parameter (KWDRDO='DRDOSE') ! dr for dose grid
C
parameter (REV=5.10998902D5) ! Electron rest energy (eV)
parameter (TREV=REV+REV)
parameter (PI=3.1415926535897932D0, TWOPI=2.0D0*PI,
1  RA2DE=180.0D0/PI, DE2RA=PI/180.0D0)

C  -- Main-PENELOPE commons.
parameter (MAXMAT=10)
common/CSIMPA/EABS(3,MAXMAT),C1(MAXMAT),C2(MAXMAT),WCC(MAXMAT),
1  WCR(MAXMAT)
common/TRACK/E,X,Y,Z,U,V,W,WGHT,KPAR,IBODY,MAT,ILB(5)
common/RSEED/ISEED1,ISEED2
C  -- Composition data.

```

```

common/COMPOS/STF(MAXMAT,30),ZT(MAXMAT),AT(MAXMAT),RHO(MAXMAT),
1 VMOL(MAXMAT),IZ(MAXMAT,30),NELEM(MAXMAT)
dimension RHOI(MAXMAT)

C -- Cylindrical geometry.

parameter (NLAM=2003,NCYM=1023,NBDM=NLAM*NCYM)
common/CYLGED/XG(NLAM),YG(NLAM),ZG(NLAM),RG(NLAM,NCYM),
1 RG2(NLAM,NCYM),RMAX,RMAX2,IBOD(NLAM,NCYM),MATER(NLAM,NCYM),
2 ILAY(NBDM),ICYL(NBDM),NLAY,NCYL(NLAM),NBOD
common/CYLAUX/INOUT,KLAY,KCYL
dimension DSMAX(NBDM)

C -- Source.
C -- Source energy spectrum.
parameter (NSEBM=100)
dimension ES(NSEBM),PTS(NSEBM),IAS(NSEBM),FS(NSEBM),NPRIM(NSEBM)
data NPRIM/NSEBM*0/

C -- Continuous distributions (selected by the user).

dimension kzlay(4000), dzlay(4000)

parameter (NDZM=200, NDRM=150)
parameter (NDZRT=NDZM*NDRM, NDMTT=4*NDZM*NDRM)

common/dosegrid/
1 ZDose(NDZM+1), ZDLim(NDZM+1), dzDose(NDZM),
2 RDose(NDRM+1), RDLim(NDRM+1), drDose(NDRM),
3 NDZ(NDZM), NDR(NDRM), NDZT, NDRT

dimension DRMass(NDZM, NDRM)
data DRMass/NDZRT*0.0d0/

dimension DAV(NDRM), DErr(NDRM), DRel(NDRM)

dimension Dose(NDZM,NDRM), Dose2(NDZM,NDRM),
1 DoseP(NDZM,NDRM),LDose(NDZM,NDRM)
data Dose,Dose2,DoseP,LDose/NDMTT*0.0D0/

dimension DosePr(NDZM,NDRM), DosePr2(NDZM,NDRM),
1 DosePrP(NDZM,NDRM),LDosePr(NDZM,NDRM)
data DosePr,DosePr2,DosePrP,LDosePr/NDMTT*0.0D0/

C
external RAND

C =====
C Instructions
C =====
C -- Time counter initialization.
call TIMEO

C -- Read in input file.
narg = iargc()
if(narg.gt.0) then
if(narg.ne.3) then
write(*,*) '** Exactly 3 input arguments: input & output'//
1 ' files and output suffix, please **'
stop
else
call getarg(1, argi)
call getarg(2, argo)
call getarg(3, argsfx)
end if
else
argi = 'newpencyl.in'
argo = 'newpencyl.out'
argsfx = 'dft'

```

```

end if

    inquire(FILE = argi, EXIST = exists)
    if(.not. exists) then
        write(*,'(2A/)') ' >> Cannot find file ', argi
        stop
    end if

len_sfx = len_trim(argsfx)
C len_sfx = lnblnk(argsfx)

    open(5,FILE=argi)
    open(6,FILE=argo)
    write(6,1100)
1100 format(/3X,43('*'),/3X,'** PROGRAM PENCYL. Input data file. ',
1 ' **',/3X,43('*'))

C -- Title.
    read(5,'(A6,1X,A128)') KWORD,BUFFER
    write(6,'(/3X,A128)') BUFFER

C -- Geometry definition and initialization of tracking routines.
    call GEOINC(NMATG,5,6)
    do KL=1,NLAY
        do KC=1,NCYL(KL)
            KB=IBOD(KL,KC)
            DSMAX(KB)=min(ZG(KL+1)-ZG(KL),RG(KL,KC+1)-RG(KL,KC))/5.0DO
            DSMAX(KB)=max(DSMAX(KB),1.0D-8)
        end do
    end do

C -- Source description.
    write(6,1200)
1200 format(/3X,70('-'),/3X,'>>>>>> Source description.')
```

21 continue

```

    read(5,'(A6,1X,A128)') KWORD,BUFFER
    if(KWORD.EQ.KWCOMM) go to 21

C -- primary particle type.
    if(KWORD.EQ.KWKPAR) then
        read(BUFFER,*) KPARP
    22 continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER
        if(KWORD.EQ.KWCOMM) go to 22
    else
        KPARP=1
    end if
    if(KPARP.LT.1.OR.KPARP.GT.3) KPARP=1
    if(KPARP.EQ.1) write(6,1210)
1210 format(3X,'Primary particles: electrons')
    if(KPARP.EQ.2) write(6,1211)
1211 format(3X,'Primary particles: photons')
    if(KPARP.EQ.3) write(6,1212)
1212 format(3X,'Primary particles: positrons')
```

C -- Initial energy of primary particles.

```

ISPEC=0
    if(KWORD.EQ.KWSENE) then
        NSEB=1
        read(BUFFER,*) E0
        write(6,1220) E0
1220 format(3X,'Initial energy = ',1P,E13.6,' eV')
    23 continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER
        if(KWORD.EQ.KWCOMM) go to 23
    else if(KWORD.EQ.KWSPEC) then
        ISPEC=1
```

```

NSEB=0
24  continue
    NSEB=NSEB+1
    read(BUFFER,*) ES(NSEB),PTS(NSEB)
25  continue
    read(5,'(A6,1X,A128)') KWORD,BUFFER
    if(KWORD.EQ.KWCOMM) go to 25
    if(KWORD.EQ.KWSPEC) go to 24
    else
        EO=1.0D6
        write(6,1220) EO
    end if
    if(ISPEC.EQ.1) then
        if(NSEB.GT.NSEEM) then
            write(6,*) 'NSEEM is too small.'
            stop 'NSEEM is too small.'
        else if(NSEB.LE.1) then
            write(6,*) 'The source energy spectrum is not defined.'
            stop 'The source energy spectrum is not defined.'
        else
            call SORT2(ES,PTS,NSEB)
            write(6,1221)
1221  format(/3X,'Spectrum:',7X,'I',4X,'E_low(eV)',4x,'E_high(eV)',
1      5X,'P_sum(E)',/16X,45('-'))
            do I=1,NSEB-1
                write(6,'(16X,I4,1P,5E14.6)') I,ES(I),ES(I+1),PTS(I)
            end do
            EO=ES(NSEB)
            NSEB=NSEB-1
            call IRNDO(PTS,FS,IAS,NSEB)
        end if
    end if
    if(EO.LT.100.0D0) then
        write(6,*) 'The initial energy E0 is too small.'
        stop 'The initial energy E0 is too small.'
    end if
    EPMAX=EO
C      -- Positrons eventually give annihilation gamma-rays. The maximum
C      energy of annihilation photons is .lt. 1.21*(EO+me*c**2).
    if(KPARP.EQ.3) EPMAX=1.21D0*(EO+5.12D5)

    KSOURC=1

C      -- External (cylindrical) source body.
    if(KWORD.EQ.KWSHEI) then
        if(KSOURC.EQ.2) then
            write(6,*) 'An extended source has already been defined.'
            stop 'An extended source has already been defined.'
        end if
        read(BUFFER,*) STHICK
26  continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER
        if(KWORD.EQ.KWCOMM) go to 26
        else
            STHICK=0.0D0
        end if
        if(KSOURC.EQ.1) write(6,1230) STHICK
1230 format(/3X,'Active volume:          height =',1P,E13.6,' cm')
        if(STHICK.LT.-1.0D-16) then
            write(6,*) 'Negative thickness.'
            stop 'Negative thickness.'
        end if
    end if
C
    if(KWORD.EQ.KWSRAD) then
        if(KSOURC.EQ.2) then
            write(6,*) 'An extended source has already been defined.'
            stop 'An extended source has already been defined.'
        end if
    end if

```

```

        read(BUFFER,*) SRIN,SROUT
        KSOURC=1
27  continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER
        if(KWORD.EQ.KWCOMM) go to 27
    else
        SRIN=0.0D0
        SROUT=0.0D0
    end if
    if(KSOURC.EQ.1) write(6,1231) SRIN,SROUT
1231 format(21X,'inner radius =',1P,E13.6,' cm',/
1      21X,'outer radius =',E13.6,' cm' )
        SRIN2=SRIN**2
        SROUT2=SROUT**2-SRIN**2
        if(SROUT2.LT.-1.0D-35) then
            write(6,*) 'The source radii are inconsistent.'
            stop 'The source radii are inconsistent.'
        end if
C
        if(KWORD.EQ.KWSPDS) then
            if(KSOURC.EQ.2) then
                write(6,*) 'An extended source has already been defined.'
                stop 'An extended source has already been defined.'
            end if
            read(BUFFER,*) SX0,SY0,SZ0
28  continue
            read(5,'(A6,1X,A128)') KWORD,BUFFER
            if(KWORD.EQ.KWCOMM) go to 28
        else
            SX0=0.0D0
            SY0=0.0D0
            SZ0=-1.0D15
        end if
        if(KSOURC.EQ.1) write(6,1232) SX0,SY0,SZ0
1232 format(3X,'Coordinates of centre:   SX0 =',1P,E13.6,
1  ' cm',/30X,'SY0 =',E13.6,' cm',/30X,'SZ0 =',E13.6,' cm')
C
        if(KWORD.EQ.KWSDIR) then
            read(BUFFER,*) STHETA,SPHI
29  continue
            read(5,'(A6,1X,A128)') KWORD,BUFFER
            if(KWORD.EQ.KWCOMM) go to 29
        else
            STHETA=0.0D0
            SPHI=0.0D0
        end if
        write(6,1233) STHETA,SPHI
1233 format(3X,'Beam direction angles:  THETA =',1P,E13.6,' deg',/
1  30X,'PHI =',E13.6,' deg')
C
        if(KWORD.EQ.KWSAPE) then
            read(BUFFER,*) SALPHA
30  continue
            read(5,'(A6,1X,A128)') KWORD,BUFFER
            if(KWORD.EQ.KWCOMM) go to 30
        else
            SALPHA=0.0D0
        end if
        write(6,1234) SALPHA
1234 format(3X,'Beam aperture:',11X,'ALPHA =',1P,E13.6,' deg')
        call GCONEO(STHETA*DE2RA,SPHI*DE2RA,SALPHA*DE2RA)

C  -- Material data and simulation parameters.
        write(6,1300)
1300 format(/3X,70(' '),/
1  3X,'>>>>> Material data and simulation parameters.')
```

```

        if(KWORD.EQ.KWNMAT) then
```

```

    read(BUFFER,*) NMAT
31  continue
    read(5,'(A6,1X,A128)') KWORD,BUFFER
    if(KWORD.EQ.KWCOMM) go to 31
    else
        write(6,*) 'You have to specify the number of materials.'
        stop 'You have to specify the number of materials.'
    end if
    write(6,1310) NMAT
1310 format(3X,'Number of different materials = ',I2)
    if(NMAT.LT.1.OR.NMAT.GT.MAXMAT) then
        write(6,*) 'Wrong number of materials.'
        stop 'Wrong number of materials.'
    end if

C  -- Simulation parameters.
    do M=1,NMAT
        EABS(1,M)=0.010D0*EPMAX
        EABS(2,M)=0.001D0*EPMAX
        EABS(3,M)=0.010D0*EPMAX
        C1(M)=0.10D0
        C2(M)=0.10D0
        WCC(M)=EABS(1,M)
        WCR(M)=EABS(2,M)
    end do

C
    if(KWORD.EQ.KWSIMP) then
        read(BUFFER,*) M
        if(M.LT.1.OR.M.GT.NMAT) then
            write(6,'(A6,1X,A65)') KWORD,BUFFER
            write(6,*) 'Incorrect material number.'
            stop 'Incorrect material number.'
        end if
        read(BUFFER,*) M,EABS(1,M),EABS(2,M),EABS(3,M),C1(M),C2(M),
1      WCC(M),WCR(M)
32  continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER
        if(KWORD.EQ.KWCOMM) go to 32
        if(KWORD.EQ.KWSIMP) then
            read(BUFFER,*) M
            if(M.LT.1.OR.M.GT.NMAT) then
                write(6,'(A6,1X,A65)') KWORD,BUFFER
                write(6,*) 'Incorrect material number.'
                stop 'Incorrect material number.'
            end if
            read(BUFFER,*) M,EABS(1,M),EABS(2,M),EABS(3,M),C1(M),C2(M),
1      WCC(M),WCR(M)
            go to 32
        end if
    end if

C
    do M=1,NMAT
        if(M.EQ.1) LIT='st'
        if(M.EQ.2) LIT='nd'
        if(M.EQ.3) LIT='rd'
        if(M.GT.3) LIT='th'
        write(6,1320) M,LIT
1320 format(/3X,'**** ',I2,A2,' material')
        if(EABS(1,M).LT.1.0D2) EABS(1,M)=1.0D2
        if(EABS(2,M).LT.1.0D2) EABS(2,M)=1.0D2
        if(EABS(3,M).LT.1.0D2) EABS(3,M)=1.0D2
        write(6,1321) EABS(1,M)
1321 format(3X,'Electron absorption energy = ',1P,E13.6,' eV')
        write(6,1322) EABS(2,M)
1322 format(3X,' Photon absorption energy = ',1P,E13.6,' eV')
        write(6,1323) EABS(3,M)
1323 format(3X,'Positron absorption energy = ',1P,E13.6,' eV')
        write(6,1324) C1(M),C2(M),WCC(M),WCR(M)

```

```

1324 format(3X,'Electron-positron simulation parameters:',
1 /4X,'C1 =',1P,E13.6,',      C2 =',E13.6,/3X,'Wcc =',E13.6,
2 ' eV, Wcr =',E13.6,' eV')
end do

C -- Initialization of PENELOPE.
if(KWORD.EQ.KWPFNA) then
  read(BUFFER,'(A32)') PFILE
  write(6,1330) PFILE
1330 format(/3X,'PENELOPE's material definition file: ',A18)
33 continue
  read(5,'(A6,1X,A128)') KWORD,BUFFER
  if(KWORD.EQ.KWCOMM) go to 33
else
  write(6,*) 'You have to specify a material file.'
  stop 'You have to specify a material file.'
end if

C
open(15,FILE=PFILE)
open(16,FILE='material.'//argsfx(1:len_sfx)//'.dat')
  INFO=5
  call PEINIT(EPMAX,NMAT,15,16,INFO)
  CLOSE(UNIT=15)
  CLOSE(UNIT=16)
  if(NMATG.LT.1.OR.NMATG.GT.NMAT) then
    write(6,*) 'Conflicting material numbers.'
    stop 'Conflicting material numbers.'
  end if

C -- Inverse densities are used to score the local dose.
do M=1,NMAT
  RHOI(M)=1.0DO/RHO(M)
end do

C -- Tallied distributions (selected by the user).
if(KWORD.EQ.KWDZDO.OR.KWORD.EQ.KWDRDO) then
  write(6,1500)
1500 format(/3X,70('-'),/
1 3X,'>>>>> User distributions to be tallied.')
end if

C -- set up the dose grid.
NZDLim = 0
NDZT = 0
if(KWORD.EQ.KWDZDO) then
  341 NZDLim = NZDLim + 1
  read(BUFFER,*) dzDose(NZDLim), ZLim1, ZLim2
  if(ZLim1.ge.ZLim2) then
    write(6,'(a)') 'Dose grid z-bounds should be increasing.'
    stop 'Dose grid z-bounds should be increasing.'
  end if
  if(NZDLim.gt.1.AND.ZLim1.lt.ZDLim(NZDLim)) then
    write(6,'(a)') 'The adjacent dose grid z-bounds overlapped.'
    stop 'The adjacent dose grid z-bounds overlapped.'
  end if
  ZDLim(NZDLim) = ZLim1
  ZDLim(NZDLim+1) = ZLim2
  NDZ(NZDLim) = nint(
1 (ZDLim(NZDLim+1)-ZDLim(NZDLim))/dzDose(NZDLim))
  NDZT = NDZT + NDZ(NZDLim)
  34 continue
  read(5,'(A6,1X,A128)') KWORD,BUFFER
  if(KWORD.EQ.KWDZDO) go to 341
  if(KWORD.EQ.KWCOMM) go to 34
else
  NZDLim = 1
  dzDose(NZDLim) = 0.2
  ZDLim(NZDLim) = ZG(1)
  ZDLim(NZDLim+1) = ZG(NLay+1)

```

```

NDZ(NZDLim) = nint(
  1      (ZDLim(NZDLim+1)-ZDLim(NZDLim))/dzDose(NZDLim))
NDZT = NDZ(NZDLim)
end if
write(6,1450) dzDose(1), ZDLim(1), ZDLim(2)
do i=2, NZDLim
  write(6,1451) dzDose(i), ZDLim(i), ZDLim(i+1)
end do
1450 format(3x,'dz:',4x,f8.3,2x,'in [',f8.3,',',f8.3,']')
1451 format(
  10x,f8.3,2x,'in [',f8.3,',',f8.3,']')
  if (NDZT.GT.NDZM) then
    write(6,'(a)') 'NDZM is not big enough.'
    stop 'NDZM is not big enough.'
  end if
  do i=1, NZDLim
    if(dzDose(i).LT.0.0) then
      write(6,'(a,i3,a)') 'dzDose(',i,') could not be negative.'
      write(6,'(a,i3,a)') 'dzDose(',i,') could not be negative.'
      stop
    else if (ZDLim(NZDLim+1)-ZDLim(1).GT.ZG(NLay+1)-ZG(1)) then
      write(6,'(a)')
      1      'Dose grid depth is larger than the phantom thickness'
      stop 'Dose grid depth is larger than the phantom thickness'
    end if
  end do
end do
k = 0
do i=1, NZDLim
  do j=1, NDZ(i)
    k = k + 1
    ZDose(k) = ZDLim(i) + dzDose(i)*(j-1)
  end do
end do
ZDose(NDZT+1) = ZDLim(NZDLim+1)

NRDLim = 0
NDRT = 0
  if(KWORD.EQ.KWDRDO) then
    351 NRDLim = NRDLim + 1
      read(BUFFER,*) drDose(NRDLim), RLim1, RLim2
    if(RLim1.ge.RLim2) then
      write(6,'(a)') 'Dose grid r-bounds should be increasing.'
      stop 'Dose grid r-bounds should be increasing.'
    end if
    if(NRDLim.gt.1.AND.RLim1.lt.RDLim(NRDLim)) then
      write(6,'(a)') 'The adjacent dose grid r-bounds overlapped.'
      stop 'The adjacent dose grid r-bounds overlapped.'
    end if
    RDLim(NRDLim) = RLim1
    RDLim(NRDLim+1) = RLim2
    NDR(NRDLim) = nint(
      1      (RDLim(NRDLim+1)-RDLim(NRDLim))/drDose(NRDLim))
    NDRT = NDRT + NDR(NRDLim)
    35  continue
      read(5,'(A6,1X,A128)') KWORD,BUFFER
      if(KWORD.EQ.KWDRDO) go to 351
      if(KWORD.EQ.KWCOMM) go to 35
  else
    drDose(1) = 0.2
    RDLim(1) = RG(1,1)
    RDLim(2) = RG(1,NCyl(1)+1)
    NDR(1) = nint((RDLim(2)-RDLim(1))/drDose(1))
    NDRT = 1
  end if
  write(6,1452) drDose(1), RDLim(1), RDLim(2)
do i=2, NRDLim
  write(6,1451) drDose(i), RDLim(i), RDLim(i+1)
end do
1452 format(3x,'dr:',4x,f8.3,2x,'in [',f8.3,',',f8.3,']')

```



```

    if(NDRT.LT.1) then
      write(6,*) 'Wrong number of profile dose rings'
      stop 'Wrong number of profile dose rings.'
    else if (NDRT.GT.NDRM) then
      write(6,*) 'NDRM is not big enough.'
      stop 'NDRM is not big enough.'
    end if
  do i=1, NRDLim
    if(drDose(i).lt.0.0) then
      write(6,'(a,i3,a)') 'drDose(',i,') could not be negative.'
      write(*, '(a,i3,a)') 'drDose(',i,') could not be negative.'
      stop
    else if
      1 (RDLim(NRDLim+1)-RDLim(1).GT.RG(1,NCyl(1)+1)-RG(1,1)) then
        write(6,'(a)')
      1 'Dose grid depth is larger than the phantom thickness'
        stop 'Dose grid depth is larger than the phantom thickness'
      end if
  end do
k = 0
do i=1, NRDLim
  do j=1, NDR(i)
    k = k + 1
    RDose(k) = RDLim(i) + drDose(i)*(j-1)
  end do
end do
RDose(NDRT+1) = RDLim(NRDLim+1)

C -- Calculate the mass for the dose grid
kzd= 0
lzlay = 1 ! index of current layer
do i=1, NZDLim
  do k=1, NDZ(i)
    kzm = 1 ! kzm: # of layers current dose grid contains
    kzd = kzd + 1 ! kzd: index of current ZDose element
    zDose1 = ZDose(kzd) ! lower z bound of current dose voxel
    zDose2 = ZDose(kzd+1) ! upper z bound of current dose voxel
    zzmass = zDose1 ! temporary reference point
1454 continue
    dzg = ZG(lzlay+1) - zDose2 ! distance from current dose grid's upper bound
                                ! to that of current layer
    if(abs(dzg).lt.1d-10) then ! they coincide with each other
      kzlay(kzm) = lzlay ! remember what layer is inside
      dzlay(kzm) = zDose2 - zzmass ! remember how much is inside
      lzlay = lzlay + 1 ! advance to next layer
      if(lzlay.gt.Nlay) lzlay = Nlay ! meet the end
    else if(dzg.lt.0d0) then ! haven't crossed current dose voxel's upper bound
      kzlay(kzm) = lzlay ! remember what layer is inside
      dzlay(kzm) = ZG(lzlay+1) - zzmass ! remember how much is inside
      zzmass = ZG(lzlay+1) ! advance to current layer's upper bound
      lzlay = lzlay + 1 ! advance to next layer
      kzm = kzm + 1 ! increase the # of layers inside
      go to 1454 ! go until meet/cross current dose voxel's upper bound
    else if(dzg.gt.0d0) then ! crossed current dose voxel's upper bound
      kzlay(kzm) = lzlay ! remember what layer is inside
      dzlay(kzm) = zDose2 - zzmass ! remember how much is inside
    end if

    do j=1, kzm ! now do it radially in current dose grid's layer
      rmass = rg(kzlay(j),1) ! temporary reference point
      lrlay = findRingNum(kzlay(j),RDose(1)) ! index of current ring
      ldos = 1 ! index of current dose voxel
1455 continue
      drg = rg(kzlay(j),lrlay+1) - rmass ! distance from the reference point to
                                          ! current ring's upper bound
      drd = RDose(ldos+1) - rmass ! distance from the reference point to
                                  ! current dose voxel's upper bound
      if(Mater(kzlay(j),lrlay).eq.0) then

```

```

    RhoRho = 0d0
  else
    RhoRho = Rho(mater(kzlay(j),lrlay))
  end if
  if(abs(drg-drd).lt.1d-10) then ! they coincide with each other
    DRMass(kzd,ldos) = DRMass(kzd,ldos) + RhoRho*PI*
      1 (rg(kzlay(j),lrlay+1)**2-rmass**2)*dzlay(j)
    lrlay = lrlay + 1 ! advance to next ring
    ldos = ldos + 1 ! advance to next dose voxel
    rmass = rg(kzlay(j),lrlay) ! advance to next reference point
  else if(drg.lt.drd) then ! haven't crossed current dose voxel's
    ! upper bound
    DRMass(kzd,ldos) = DRMass(kzd,ldos) + RhoRho*PI*
      1 (rg(kzlay(j),lrlay+1)**2-rmass**2)*dzlay(j)
    lrlay = lrlay + 1 ! advance to next ring
    rmass = rg(kzlay(j),lrlay) ! advance to next reference point:
    !next ring's lower bound
  else ! crossed current dose voxel's upper bound
    DRMass(kzd,ldos) = DRMass(kzd,ldos) + RhoRho*PI*
      1 (RDose(ldos+1)**2-rmass**2)*dzlay(j)
    ldos = ldos + 1 ! advance to next dose voxel
    rmass = RDose(ldos) ! advance to next reference point: next
    ! dose voxel's lower bound
  end if
  if(abs(rmass - RDose(NDRT+1)).gt.1d-10) then
    go to 1455
  end if
end do

end do ! within a z- dose grid region
end do ! i: loop over dose grid regions in z-direction

C -- Job characteristics.
  write(6,1700)
  1700 format(/3X,70('-'),/
    1 3X,'>>>>> Job characteristics.')
```

```

C
  IRESUM=0
  if(KWORD.EQ.KWRESU) then
    read(BUFFER,'(A32)') PFILER
    write(6,1710) PFILER
  1710 format(3X,'Resume simulation from previous dump file: ',A32)
    IRESUM=1
  71 continue
    read(5,'(A6,1X,A128)') KWORD,BUFFER
    if(KWORD.EQ.KWCOMM) go to 71
  end if
C
  IDUMP=0
  if(KWORD.EQ.KWDUMP) then
    read(BUFFER,'(A32)') PFILED
    write(6,1720) PFILED
  1720 format(3X,'Write final counter values on the dump file: ',A32)
    IDUMP=1
  72 continue
    read(5,'(A6,1X,A128)') KWORD,BUFFER
    if(KWORD.EQ.KWCOMM) go to 72
  end if
C
  if(KWORD.EQ.KWFOUT) then
    read(BUFFER,*) IFullOutp
    75 continue
    read(5,'(A6,1X,A128)') KWORD,BUFFER
    if(KWORD.EQ.KWCOMM) go to 75
  else
    IFullOutp = 0
  end if
C
```

```

        if(KWORD.EQ.KWNSIM) then
            read(BUFFER,*) NTOT
            if(NTOT.LT.1) NTOT=2147483647
73      continue
            read(5,'(A6,1X,A128)') KWORD,BUFFER
            if(KWORD.EQ.KWCOMM) go to 73
        else
            NTOT=2147483647
        end if
        write(6,1730) NTOT
1730 format(3X,'Number of showers to be simulated = ',I11)
C
        if(KWORD.EQ.KWNITR) then
            read(BUFFER,*) NINTER
            if(NINTER.LT.1) NINTER = 100000
99      continue
            read(5,'(A6,1X,A128)') KWORD,BUFFER
            if(KWORD.EQ.KWCOMM) go to 99
        else
            NINTER = 100000
        end if
        write(6,1735) NINTER
1735 format(3X,'Number of showers to be output per interval = ',I11)
C
        if(KWORD.EQ.KWRSEE) then
            read(BUFFER,*) ISEED1,ISEED2
74      continue
            read(5,'(A6,1X,A128)') KWORD,BUFFER
            if(KWORD.EQ.KWCOMM) go to 74
        else
            ISEED1=12345
            ISEED2=54321
        end if
        if(IRESUM.EQ.0) write(6,1740) ISEED1,ISEED2
1740 format(3X,'Random number generator seeds = ',I10,', ',I10)
C
        if(KWORD.EQ.KWTIME) then
            read(BUFFER,*) ITIME
        else
            ITIME=100
        end if
        write(6,1750) ITIME
1750 format(3X,'Computation time available = ',I12,' sec')
        if(ITIME.LT.1) ITIME=100
C
        call TIMER(TSEC)
        TSECIN=TSEC
        TSECA=ITIME+TSEC
        write(6,1760)
1760 format(/3X,70('-'))

C      -- If 'RESUME' is active, read previously generated counters...
        NA=0
        TIMEA=0.0D0
        if(IRESUM.EQ.1) then
            open(9,FILE=PFILER)
            read (9,*,ERR=1800,END=1800) NAA,TIMEAA
            NA=NAA
            TIMEA=TIMEAA
            read (9,*) ISEED1,ISEED2
            read (9,*) NDZTt, NDRTt
            if(NDZTt.ne.NDZT.or.NDRTt.ne.NDRT) then
                write(6,*) '>>>Dose grid not consistent with the resume file.'
                stop      '>>>Dose grid not consistent with the resume file.'
            end if
            read(9,999) (( Dose(Kz,Kr), Kz=1,NDZT),
1          Kr=1,NDRT),
2          (( Dose2(Kz,Kr), Kz=1,NDZT),

```

```

3           Kr=1,NDRT)
  read(9,999) (( DosePr(Kz,Kr), Kz=1,NDZT),
1           Kr=1,NDRT),
2           (( DosePr2(Kz,Kr), Kz=1,NDZT),
3           Kr=1,NDRT)
  close(9)
  go to 1802
1800  continue
  write(6,1801)
1801  format(/3X,'WARNING: Could not resume from dump file...'/)
  end if
1802  continue

C  -- Initialize constants.
  WGHT0=1.0D0  ! Primary particle weight.

  N=NA  ! Shower counter, including the dump file.
  if(NTOT.LT.0) then
    write(6,*) '  WARNING: NTOT is too large. INTEGER*4 overflow.'
    if(NTOT.LT.1) NTOT=2147483647
  end if
  if(N.GE.NTOT) go to 106

C  =====
C  Shower simulation starts here.
C  =====
101  continue  ! The simulation loop starts here.

C  -- Set the initial state of the primary particle.
  N=N+1
  KPAR=KPARP
  WGHT=WGHT0

C  ---- Initial position ...
  if(KSOURC.EQ.1) then
    Z=SZO+(RAND(1.0D0)-0.5D0)*STHICK
    SR=SQRT(SRIN2+RAND(2.0D0)*SROI2)
    PHIR=RAND(3.0D0)*TWOPI
    X=SX0+SR*COS(PHIR)
    Y=SY0+SR*SIN(PHIR)
  end if

C  ---- Initial direction ...
  call GCONE(U,V,W)

C  ---- initial energy ...
  if(ISPEC.EQ.0) then
    E=EO  ! Monoenergetic source.
    NPRIM(1)=NPRIM(1)+1
  else
    ! Continuous spectrum. E sampled by Walker's method.
    RN=RAND(4.0D0)*NSEB+1
    K=INT(RN)
    RNF=RN-K
    if(RNF.GT.FS(K)) then
      KE=IAS(K)
    else
      KE=K
    end if
    E=ES(KE)+RAND(5.0D0)*(ES(KE+1)-ES(KE))
    NPRIM(KE)=NPRIM(KE)+1
  end if

C  -- Check if the trajectory intersects the material system.
  call LOCATC
  if(MAT.EQ.0) then
    call STEPC(1.0D30,DSEF,NCROSS)
    if(MAT.EQ.0) then
      go to 105  ! The particle does not enter the system.
    end if
  end if
end if

```

```

C  -- Initialization of primary particle counters.
    ILB(1)=1          ! Identifies primary particles.
    ILB(2)=0
    ILB(3)=0
    ILB(4)=0
    ILB(5)=0

C  -----
C  -- Track simulation begins here.
    call CLEANS       ! Cleans secondary stack.

102 continue
    call START        ! Starts simulation in current medium.

103 continue
    call JUMP(DSMAX(IBODY),DS)    ! Analogue simulation.
    call STEPC(DS,DSEF,NCROSS)   ! Determines step end position.

C  -- Check whether the particle is outside the enclosure.
    if(MAT.EQ.0) then
        if(Z.GE.ZG(NLAY+1)) then
            if(W.LT.0) stop 'Transmitted with negative W?'
            go to 104
        else if(Z.LE.ZG(1)) then
            if(W.GT.0) stop 'Backscattered with positive W?'
            go to 104
        end if
    end if

C  -- If the particle has crossed an interface, restart the track in the new material.
    if(NCROSS.GT.0) go to 102

C  -- Simulate the interaction event
    call KNOCK(DE,ICOL)          ! Analogue simulation.

C  -- Dose distributions
C  -- Tally it if DE>0 and inside the dose grid
    if(DE.gt.1e-35) then
        KLAY=ILAY(IBODY)
        rD = sqrt((x-XG(KLAY))**2+(y-YG(KLAY))**2)
        if(RDose(1).le.rD.AND.rD.le.RDose(NDRT+1)) then
            if(ZDose(1).le.z.AND.z.le.ZDose(NDZT+1)) then
                Kz = Get_Idx('z',z) ! depth channel
                Kr = Get_Idx('r',rD) ! radial channel
            C  -- total dose
                if(N.ne.LDose(Kz,Kr)) then
                    Dose(Kz,Kr) = Dose(Kz,Kr) + DoseP(Kz,Kr)
                    Dose2(Kz,Kr) = Dose2(Kz,Kr) + DoseP(Kz,Kr)**2
                    DoseP(Kz,Kr) = DE*WGHT
                    LDose(Kz,Kr) = N
                else
                    DoseP(Kz,Kr) = DoseP(Kz,Kr) + DE*WGHT
                end if
            C  if(
                1      (ILB(1).eq.1) .OR.
                1      (ILB(1).eq.2.and.kpar.eq.2.and.icol.eq.2) .OR.
                2      (ILB(1).eq.2.and.ILB(2).eq.2.and.
                2      kpar.eq.1.and.(icol.eq.1.or.icol.eq.3)) .OR.
                2      (ILB(1).eq.2.and.ILB(2).eq.2.and.
                2      kpar.eq.3.and.
                2      (icol.eq.1.or.icol.eq.3.or.icol.eq.6)) .OR.
                3      (ILB(1).eq.3.and.ILB(2).ne.2.and.ILB(3).eq.3
                3      .and.kpar.eq.1.and.icol.eq.1)
                4      ) then
                    if(N.ne.LDosePr(Kz,Kr)) then
                        DosePr(Kz,Kr) = DosePr(Kz,Kr) + DosePrP(Kz,Kr)
                        DosePr2(Kz,Kr) = DosePr2(Kz,Kr) + DosePrP(Kz,Kr)**2
                        DosePrP(Kz,Kr) = DE*WGHT
                    end if
                end if
            end if

```

```

        LDosePr(Kz,Kr) = N
        else
            DosePrP(Kz,Kr) = DosePrP(Kz,Kr) + DE*WGHT
        end if
    end if
end if ! z
end if ! r
end if ! DE

C  -- Check if the particle has been absorbed .
    if(E.GT.EABS(KPAR,MAT)) go to 103

C  -- The simulation of the track ends here.
C  -----

104 continue

C  -- Any secondary left?
    call SECPAR(LEFT)
    if(LEFT.GT.0) then
        INOUT=1
        KLAY=ILAY(IBODY)
        KCYL=ICYL(IBODY)
C  -- Subtract E and charge from the tallied distributions to avoid double-counting.
        rD = sqrt((x-XG(KLAY))**2+(y-YG(KLAY))**2)
        if(RDose(1).le.rD.AND.rD.le.RDose(NDRT+1)) then
            if(ZDose(1).le.z.AND.z.le.ZDose(NDZT+1)) then
                Kz = Get_Idx('z',z) ! depth channel
                Kr = Get_Idx('r',rD) ! radial channel
C  -- total dose
                if(N.NE.LDOSE(Kz,Kr)) then
                    Dose(Kz,Kr) = Dose(Kz,Kr) + DoseP(Kz,Kr)
                    Dose2(Kz,Kr) = Dose2(Kz,Kr) + DoseP(Kz,Kr)**2
                    DoseP(Kz,Kr) = -E*WGHT
                    LDose(Kz,Kr) = N
                else
                    DoseP(Kz,Kr) = DoseP(Kz,Kr) - E*WGHT
                end if
                if(
1         (ILB(1).eq.2.and.ILB(2).eq.2.and.kpar.ne.2) .OR.
2         (ILB(1).eq.3.and.ILB(2).eq.3.and.ILB(3).eq.6.and.kpar.eq.2) .OR.
3         (ILB(1).eq.3.and.ILB(2).ne.2.and.ILB(3).eq.3.and.kpar.eq.1)
                ) then
                    if(N.ne.LDosePr(Kz,Kr)) then
                        DosePr(Kz,Kr) = DosePr(Kz,Kr) + DosePrP(Kz,Kr)
                        DosePr2(Kz,Kr) = DosePr2(Kz,Kr) + DosePrP(Kz,Kr)**2
                        DosePrP(Kz,Kr) = -E*WGHT
                        LDosePr(Kz,Kr) = N
                    else
                        DosePrP(Kz,Kr) = DosePrP(Kz,Kr) - E*WGHT
                    end if
                end if
            end if ! z
        end if ! rD

        go to 102
    end if ! left
C  -- The simulation of the shower ends here.
C  -----

105 continue
    call TIMER(TSEC)

if(N.LT.NTOT.AND.mod(N, NINTER).eq.0) then
    Dt1 = Tsec-Tsecin
    IHours = INT(Dt1/3600)
    IMinutes = INT((Dt1-IHours*3600)/60)
    Seconds = Dt1 - IHours*3600 - IMinutes*60

```

```

write(6,9999) 'N =', N, 'Elapsed', Dt1, "(s)",
1      "-->", IHours, ":", IMinutes, ":", Seconds
write(*,9999) 'N =', N, 'Elapsed', Dt1, "(s)",
1      "-->", IHours, ":", IMinutes, ":", Seconds
9999  format(1x,A3,1x,I10,3x,A7,2x,f15.2,2x,A3,3X,A3,I6,A1,I6,A1,F6.2)
C      -- dump intermediate results in case that the job could not be finished as scheduled
      TSIM=MAX(1.0D0,Dt1)+TIMEA
      if(IDUMP.EQ.1) then
open(9,FILE=PFILED)
write(9,*) N,TSIM
write(9,*) ISEED1,ISEED2
      write(9,*) NDZT, NDRT
write(9,999) (( (Dose(Kz,Kr)+DoseP(Kz,Kr)),
1             Kz=1,NDZT),Kr=1,NDRT),
2             (( (Dose2(Kz,Kr)+DoseP(Kz,Kr)**2),
3             Kz=1,NDZT),Kr=1,NDRT)
write(9,999) (( (DosePr(Kz,Kr)+DosePrP(Kz,Kr)),
1             Kz=1,NDZT),Kr=1,NDRT),
2             (( (DosePr2(Kz,Kr)+DosePrP(Kz,Kr)**2),
3             Kz=1,NDZT),Kr=1,NDRT)
close(9)
      end if
end if
999  format(g25.16)
C      -- end of intermediate dump
C      -----

      if(TSEC.LT.TSECA.AND.N.LT.NTOT) go to 101
C      =====
C      End the simulation after the allotted time or after completing NTOT showers.
C      =====

C      -- Transfer contents of partial counters of the last one shower to global counters.
do Kz=1,NDZT
  do Kr=1,NDRT
    Dose(Kz,Kr) = Dose(Kz,Kr) + DoseP(Kz,Kr)
    Dose2(Kz,Kr) = Dose2(Kz,Kr) + DoseP(Kz,Kr)**2
    DosePr(Kz,Kr) = DosePr(Kz,Kr) + DosePrP(Kz,Kr)
    DosePr2(Kz,Kr) = DosePr2(Kz,Kr) + DosePrP(Kz,Kr)**2
  end do
end do
C
      TSIM=MAX(1.0D0,TSEC-TSECIN)+TIMEA

C      -- If 'DUMPTO' is active, write counters to a dump file.
if(IDUMP.EQ.1) then
  open(9,FILE=PFILED)
  write(9,*) N,TSIM
  write(9,*) ISEED1,ISEED2
  write(9,*) NDZT, NDRT
  write(9,999) (( Dose(Kz,Kr), Kz=1,NDZT),
1             Kr=1,NDRT),
2             (( Dose2(Kz,Kr), Kz=1,NDZT),
3             Kr=1,NDRT)
  write(9,999) (( DosePr(Kz,Kr), Kz=1,NDZT),
1             Kr=1,NDRT),
2             (( DosePr2(Kz,Kr), Kz=1,NDZT),
3             Kr=1,NDRT)
  close(9)
end if

C      -- Print simulation results.
106  continue

      TOTN=N
      write(6,3000)
3000  format(///3X,34(' ')/3X,'** Program NewPENCYL. Results. **',
1 /3X,34(' '))

```

```

        IFNT = N - NA
        TSIM=MAX(1.0D0,TSEC-TSECIN)
        write(6,3001) TSIM
3001 format(/3X,'Simulation time ..... ',
1 1P,E13.6,' sec')
        TAVS=IFNT/TSIM
        write(6,3002) TAVS
3002 format(3X,'Simulation speed ..... ',
1 1P,E13.6,' showers/sec')
        write(6,3003) IFNT
3003 format(/3X,
1 'Simulated primary particles this time ..... ',1P,I13)
        write(6,3004) N
3004 format(/3X,
1 'Total simulated primary particles ..... ',1P,I13)

        write(6,3099) ISEED1,ISEED2
3099 format(/3X,'Random seeds = ',I10,' , ',I10)
        write(6,'(/3X,'*** END ***')')
        close(6)

C -- Print tallied distributions.
  if(IFullOutp.gt.0) then ! output control

        open(9,FILE='Dose.'//argsfx(1:len_sfx)//'.dat')
        DF=1.0D0/TOTN

C
        write(9,'(A,/)') 'Dose.'//argsfx(1:len_sfx)//'.dat'
        write(9,'(A,I8)') 'Num_Dose_Grid_Z=', NDZT
        write(9,'(A,I8)') 'Num_Dose_Grid_R=', NDRT
        write(9,'(/,A)') 'ZDose(...)'
        write(9,'(1000f8.3)') (ZDose(i), i=1,NDZT+1)
        write(9,'(/,A)') 'RDose(...)'
        write(9,'(1000f8.3)') (RDose(i), i=1,NDRT+1)

C
        write(9,'(/,1x,A)') 'Dose (MeV/g)'
        call Write_zr_header(9)
        do Kz=1, NDZT
            xx = (ZDose(kz) + ZDose(kz+1))/2
            do Kr=1,NDRT
                DAV(Kr) = Dose(Kz,Kr)*DF/1d6/max(1.0d-35,DRMass(Kz,Kr))
            end do
            write(9,'(f8.3,1000g18.8)') xx,(DAV(i),i=1,NDRT)
        end do

C
        write(9,'(/,1x,A)') 'RelErr'
        call Write_zr_header(9)
        do Kz=1, NDZT
            xx = (ZDose(kz) + ZDose(kz+1))/2
            do Kr=1,NDRT
                DErr(Kr) = sqrt(abs(Dose2(Kz,Kr)-Dose(Kz,Kr)**2*DF))
                DAV(Kr) = abs(Dose(Kz,Kr))
                DRel(Kr) = DErr(Kr)/max(1.0d-35,DAV(Kr))
            end do
            write(9,'(f8.3,1000f18.8)') xx,(DRel(i),i=1,NDRT)
        end do

C
        write(9,'(/, 1x,A)') 'Err of Dose (MeV/g)'
        call Write_zr_header(9)
        do Kz=1, NDZT
            xx = (ZDose(kz) + ZDose(kz+1))/2
            do Kr=1,NDRT
                DErr(Kr) = sqrt(abs(Dose2(Kz,Kr)-Dose(Kz,Kr)**2*DF))
                DErr(Kr) = DErr(Kr)*DF/1d6/max(1.0d-35,DRMass(Kz,Kr))
            end do
            write(9,'(f8.3,1000g18.8)') xx,(DErr(i),i=1,NDRT)

```



```

        end do
C
        write(9,'(//, 1x,A)') 'mass (g)  '
call Write_zr_header(9)
        do Kz=1, NDZT
            xx = (ZDose(kz) + ZDose(kz+1))/2
            write(9,'(f8.3,1000g18.8)') xx,(DRMass(Kz,Kr), Kr=1,NDRT)
        end do

        close(9)

end if ! full output control

stop
end
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                     End of main program                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

function findRingNum(kLay,r)

IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER*4 (I-N)
PARAMETER (NLAM=2003,NCYM=1023,NBDM=NLAM*NCYM)
COMMON/CYLGeo/XG(NLAM),YG(NLAM),ZG(NLAM),RG(NLAM,NCYM),
1  RG2(NLAM,NCYM),RMAX,RMAX2,IBOD(NLAM,NCYM),MATER(NLAM,NCYM),
2  ILAY(NBDM),ICYL(NBDM),NLAY,NCYL(NLAM),NBOD

do i=1, NCyl(kLay)
    if((RG(kLay,i).le.r).AND.(r.lt.RG(kLay,i+1))) then
        findRingNum = i
        return
    end if
end do

end

C*****
C function for finding the index of ZDose/RDose,
C given z/r.
C   input:
C       value --> value;
C       c --> 'z' or 'r'
C*****
function Get_Idx(c, value)

IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER*4 (I-N)

character*1 c
parameter (NDZM=200, NDRM=150)
common/dosegrid/
1  ZDose(NDZM+1), ZDLim(NDZM+1), dzDose(NDZM),
2  RDose(NDRM+1), RDLim(NDRM+1), drDose(NDRM),
3  NDZ(NDZM), NDR(NDRM), NDZT, NDRT

!!!

if(c.eq.'z') then
    n1 = 1
    n2 = NDZT
    do while (n1.ne.n2)
        nm = (n2+n1)/2
        if(value.le.ZDose(nm+1)) then
            n2 = nm
        else
            n1 = nm + 1
        end if
    end do
end do

```

```

else if(c.eq.'r') then
  n1 = 1
  n2 = NDRT
  do while (n1.ne.n2)
    nm = (n2+n1)/2
    if(value.le.RDose(nm+1)) then
      n2 = nm
    else
      n1 = nm + 1
    end if
  end do
end if
Get_Idx = n1

end

C*****
C Subroutine for writing the common header
C given file unit id.
C   input:
C     fid --> file id;
C*****
subroutine Write_zr_header(fid)
IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER*4 (I-N)

integer fid
parameter (NDZM=200, NDRM=150)
common/dosegrid/
  1  ZDose(NDZM+1), ZDLim(NDZM+1), dzDose(NDZM),
  2  RDose(NDRM+1), RDLim(NDRM+1), drDose(NDRM),
  3  NDZ(NDZM), NDR(NDRM), NDZT, NDRT

if(RDose(1).lt.1d-35) then
  write(fid,'(3x,a5,1000g18.5)')
  1  'z--r', 0, ((RDose(i)+RDose(i+1))/2, i=2,NDRT)
else
  write(fid,'(3x,a5,1000g18.5)')
  1  'z--r', ((RDose(i)+RDose(i+1))/2, i=1,NDRT)
end if

end

```

### A.3 Fortran 90 codes to generate the $2\frac{1}{2}$ -D random lung model in Chapter IV

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                                                                                   !
! Generate box + airway/vessel geometry for newpendoses.                         !
!                                                                                   !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
program main

use GeoGen_CreatAVC
use GeoGen_Output

implicit none

integer (i8) :: Rlztm
character (i1) :: flag_yn
logical :: another_cut
real (r8) :: rv

!!!

write(*,'(/a)') 'Cutoff level relative to the terminal bronchiole)'
write(*,'(2x,a)') '*note* -- negative for less orders and positive for more orders'

```

```

write(*,'(4x,a,i3,a1,i2,a1)') 'The range of input is: [' , -NAVGroup_TB+1,':',11,']'
write(*,'(4x,a,i2)') 'The default is: ', 0
write(*,'(4x,a,$)') 'Want to change? (y/n) '
read(*,'(a)') flag_yn
select case(flag_yn)
case ('y','Y')
  write(*,'(6x,a,$)') 'delta orders --> '
  read(*,*) del_NAVG_TB
case default
  del_NAVG_TB = 0
end select
NAVG = NAVGroup_TB + del_NAVG_TB

write(*,'(/a)') 'Size of the lung (cm)'
write(*,'(2x,a)') '*note* -- symmetric about x- & z-axes'
write(*,'(/2x,a, 3f8.3)') 'The default LX, LY, LZ are: ',LXdft, LYdft, LZdft
write(*,'(4x,a,$)') 'Want to change? (y/n) '
read(*,'(a)') flag_yn
select case(flag_yn)
case ('y','Y')
  write(*,'(6x,a,$)') 'LX, LY, LZ --> '
  read(*,*) LX, LY, LZ
case default
  LX = LXdft
  LY = LYdft
  LZ = LZdft
end select

write(*,'(/a)') 'Size of the phantom (cm)'
write(*,'(2x,a)') '*note* -- symmetric about x- & z-axes'
write(*,'(/2x,a, 5f8.3)') 'The default PX, PZ, PYpre, PYpos are: ',PXdft, PZdft, PYpredft, PYposdft
write(*,'(4x,a,$)') 'Want to change? (y/n) '
read(*,'(a)') flag_yn
select case(flag_yn)
case ('y','Y')
  write(*,'(6x,a,$)') 'PX, PZ, PYpre, PYpos --> '
  read(*,*) PX, PZ, PYpre, PYpos
case default
  PX = PXdft
  PZ = PZdft
  PYpre = PYpredft
  PYpos = PYposdft
end select
PY = PYpre + LY + PYpos

TofL = 0
do while (TofL<1.or.TofL>4)
  write(*,'(/a)') 'Types of lung construction:'
  write(*,'(2x,a)') '(1) Atomic mix;'
  write(*,'(2x,a)') '(2) Arbitrary cylinders input manually + background atomic mix;'
  write(*,'(2x,a)') '(3) Actual lung: airway/vessel bundles + background atomic mix;'
  write(*,'(2x,a)') '(4) Arbitrary cylinders input manually + background (water straws + air);'
  write(*,'(2x,a,$)') '--> '
  read(*,*) TofL
  write(*,'(2x,a,$)') 'realization --> '
  read(*,*) Rlztm
  !
  if(TofL>4) then
    write(*,'(a)') '>>> No such choice. Choose it again.'
  end if
end do

20 write(*,'(/a,$)') 'The suffix to all output file (<64 characters) --> '
read(*,'(a)') suffix
len_sfx = len_trim(suffix)

! -- initialize the basic planes and bodies
call Init_bPlnBod

```

```

! -- initialize the tumor if present
write(*,'(/a,$)') 'Need a tumor? (y/n)'
read(*,'(a)') flag_yn
select case(flag_yn)
case ('y', 'Y')
  flag_tumor = .true.
30 write(*,'(2x,a)') 'Shape of the tumor:'
   write(*,'(4x,a)') '(1) box;'
   write(*,'(4x,a)') '(2) sphere'
   write(*,'(4x,a)') '--> '
   read(*,'(i)') SofT
   select case(SofT)
   case (1)
write(*,'(4x,a)') 'The tumor box size and location (cm):'
write(*,'(6x,a20,2(f5.1,a4),f5.1,a1)') '*note* -- within [0,', LX, ', 0,', LY, ', 0,', LZ, ']'
  write(*,'(6x,a,$)') 'xtmr, ytmr, ztmr, dxtmr, dytmr, dztmr --> '
read(*,*) BTumor%x, BTumor%y, BTumor%z, BTumor%dx, BTumor%dy, BTumor%dz
  case (2)
write(*,'(4x,a)') 'The tumor sphere size and location (cm):'
write(*,'(6x,a20,2(f5.1,a4),f5.1,a1)') '*note* -- within [0,', LX, ', 0,', LY, ', 0,', LZ, ']'
  write(*,'(6x,a,$)') 'xtmr, ytmr, ztmr, rtmr (cm) --> '
read(*,*) STumor%x, STumor%y, STumor%z, STumor%r
  case default
    write(*,'(a)') 'No such choice. Choose it again.'
    go to 30
  end select
case default
end select

select case(TofL)
case (2)
  call CreatAVCircle_Manually(NBdle)
case (3)
  call CreatAVCircle(Rlztzn)
!
  write(*,'(/A)') 'Cut out to (2,2,2)'
write(*,'(2x,A,f8.3,A)') 'note: LXB = ', LXB, ' cm'
write(*,'(2x,A,f8.3,A)') '  LYB = ', LYB, ' cm'
write(*,'(2x,A,f8.3,A)') '  LX = ', LX, ' cm'
write(*,'(2x,A,f8.3,A)') '  LY = ', LY, ' cm'
write(*,'(2x,A,f8.3,A)') '  LZ = ', LZ, ' cm'
write(*,'(/2x,a)') 'Origin of the cut-out region (cm)'
write(*,'(4x,a,$)') 'Lox, Loy --> '
read(*,*) Lox, Loy
case (4)
  call CreatAVCircle_Manually(NBdle)
  write(*,'(/A,$)') 'Radius for the veins (cm) --> '
  read(*,*) rv
  allocate(Straw(ceiling(LX*LY/(PI*rv**2))))
  call CreatWaterStraws(Rlztzn,NBdle,rv)
case default
end select
!
another_cut = .false.
40 if(another_cut) then
  write(*,'(/a,$)') 'The suffix to all output file (<64 characters) --> '
  read(*,'(a)') suffix
  len_sfx = len_trim(suffix)
  call Init_bPlnBod
  write(*,'(/A)') 'Cut out to (2,2,2)'
write(*,'(2x,A,f8.3,A)') 'note: LXB = ', LXB, ' cm'
write(*,'(2x,A,f8.3,A)') '  LYB = ', LYB, ' cm'
write(*,'(2x,A,f8.3,A)') '  LX = ', LX, ' cm'
write(*,'(2x,A,f8.3,A)') '  LY = ', LY, ' cm'
write(*,'(2x,A,f8.3,A)') '  LZ = ', LZ, ' cm'
write(*,'(/2x,a)') 'Origin of the cut-out region (cm)'
write(*,'(4x,a,$)') 'Lox, Loy --> '

```

```

read(*,*) Lox, Loy
end if

call Output

!
write(*, '(A$)') 'Another cut? (y/n)'
read(*,*) flag_yn
if(flag_yn=='y'.or.flag_yn=='Y') then
  another_cut = .true.
else
  another_cut = .false.
end if
if(another_cut) goto 40

end program main

module GeoGen_surfaceGen

use GeoGen_Type

implicit none

contains

!*****
!
! create planes perpendicular to one axis
!   input:
!     axis  -- the axis;
!     offset -- the coordinate of the intersection of the plane and the axis
!   output:
!     nPln  -- the plane object;
!*****
type (Reduced_Surface) function normalPln(axis, offset) result(nPln)

  use GeoGen_Global, only: rPln

  character (len=*), intent(in) :: axis
  real (r8),          intent(in) :: offset

  !!!

  nPln = rPln

  select case(axis)
  case ('x', 'X')
    nPln%typ = 'reduced x-plane'
    nPln%tht = 9.0d1
  nPln%xshft = offset
  case ('y', 'Y')
    nPln%typ = 'reduced y-plane'
    nPln%tht = 9.0d1
  nPln%phi = 9.0d1
  nPln%yshft = offset
  case ('z', 'Z')
    nPln%typ = 'reduced z-plane'
    nPln%zshft = offset
  case default
    write(*,*) 'normalPln|no such axis: '//axis
  stop
  end select

end function normalPln

!*****
! create a sperical plane
!   input:

```

```

!           Sph -- the sphere object;
!   output:
!           SSph -- the description of the surface of the input Sph
!*****
type (Reduced_Surface) function sphereSurGen(Sph) result(SSph)

use GeoGen_Global, only: rSph

   type (Sphere), intent(in) :: Sph

!!!

   SSph = rSph

   SSph%xsc1 = Sph%r
   SSph%ysc1 = Sph%r
   SSph%zsc1 = Sph%r

   SSph%xshft = Sph%x
   SSph%yshft = Sph%y
   SSph%zshft = Sph%z

end function sphereSurGen

!*****
! create a box's 6 planes
!   input:
!           Bx -- the box object;
!   output:
!           SBx -- the description of the surface of the input Bx
!*****
subroutine boxSurGen(Bx, SBx)

use GeoGen_Global, only: rPln

   type (Box),           intent(in) :: Bx
   type (Reduced_Surface), intent(out) :: SBx(6)

   integer (i4) :: i

!!!

   SBx = rPln
   do i=1, 6

   end do

end subroutine boxSurGen

end module GeoGen_surfaceGen

!*****
! Subroutine
!   Initialize the basic defining planes and bodies
!*****
subroutine Init_bPlnBod

use GeoGen_Global
use GeoGen_surfaceGen

implicit none

   integer (i4) :: i, j, k
   character (i1) :: flag_yn
   integer (i4) :: choice
   integer (i1) :: am_lung

!!!

```

```

! used as input to view the whole setup in matlab
open(unit=io_v, file='view.'//suffix(1:len_sfx)//'.geo', action='WRITE', status='REPLACE')
10 format(a1,2x,g23.16)
write(io_v,'(A/)') '[Basic planes]'

! -- x-planes
bPln(1) = normalPln('x', -PX/2)
write(io_v,10) 'x', -PX/2
bPln(2) = normalPln('x', -LX/2)
write(io_v,10) 'x', -LX/2
bPln(3) = normalPln('x', LX/2)
write(io_v,10) 'x', LX/2
bPln(4) = normalPln('x', PX/2)
write(io_v,10) 'x', PX/2

! -- y-planes
bPln(5) = normalPln('y', -PYpre)
write(io_v,10) 'y', -PYpre
bPln(6) = normalPln('y', ZERO)
write(io_v,10) 'y', ZERO
bPln(7) = normalPln('y', LY)
write(io_v,10) 'y', LY
bPln(8) = normalPln('y', LY+PYpos)
write(io_v,10) 'y', LY+PYpos

! -- z-planes
bPln(9) = normalPln('z', -PZ/2)
write(io_v,10) 'z', -PZ/2
bPln(10) = normalPln('z', -LZ/2)
write(io_v,10) 'z', -LZ/2
bPln(11) = normalPln('z', LZ/2)
write(io_v,10) 'z', LZ/2
bPln(12) = normalPln('z', PZ/2)
write(io_v,10) 'z', PZ/2

do i=1, 12
  bPln(i)%num = i
end do

! -- bodies

bBod%mat = water_max

do k=1, 3
  do j=1, 3
    do i=1, 3
      bBod(i,j,k)%num = i + (j-1)*3 + (k-1)*9
      bBod(i,j,k)%id = 'basic body'
      !
      allocate(bBod(i,j,k)%sur(6), bBod(i,j,k)%sid(6), bBod(i,j,k)%bod(0))
      ! -- limiting surfaces
      bBod(i,j,k)%sur(1) = i
      bBod(i,j,k)%sur(2) = i + 1
      bBod(i,j,k)%sur(3) = j + 4
      bBod(i,j,k)%sur(4) = j + 4 + 1
      bBod(i,j,k)%sur(5) = k + 8
      bBod(i,j,k)%sur(6) = k + 8 + 1
      ! -- side pointers of limiting surfaces
      bBod(i,j,k)%sid(1) = 1
      bBod(i,j,k)%sid(2) = -1
      bBod(i,j,k)%sid(3) = 1
      bBod(i,j,k)%sid(4) = -1
      bBod(i,j,k)%sid(5) = 1
      bBod(i,j,k)%sid(6) = -1
    end do
  end do
end do
end do

```

```

! -- adjusting matter in bodies

write(*, '(/a,$)') 'Surrounding material to be atomic mix lung? (y/n) '
read(*,*) flag_yn
select case(flag_yn)
case ('y', 'Y')
  do k=1, 3
    do i=1, 3
      bBod(i,2,k)%mat = aml_max
    end do
  end do
case default
end select

choice = 0
do while (choice<1.or.choice>2)
  write(*, '(/A)') 'Density of lung parenchyma? (g/cm^3)'
  write(*, '(2x,A)') '(1) 0.26;'
  write(*, '(2x,A)') '(2) 0.201388;'
  write(*, '(2x,a,$)') '--> '
  read(*,*) choice
  select case(choice)
  case (1)
    select case (TofL)
    case(1,2)
      am_lung = aml_max
    case (3)
      am_lung = aml
    case (4)
      am_lung = air
      Rho4 = Rho(aml)
    end select
  case (2)
    select case (TofL)
    case(1,2)
      am_lung = amlp_max
    case (3)
      am_lung = amlp
    case (4)
      am_lung = air
      Rho4 = Rho(amlp)
    end select
  case default
    write(*,*) '>>> No such choice. Choose it again'
  end select
end do
bBod(2,2,2)%mat = am_lung ! lung parenchyma

end subroutine Init_bPlnBod

module GeoGen_CreatAVC

use GeoGen_Global

implicit none

!!!

contains

!*****
! -- create airway/vessel bundles in the lung region (circles) manually
!*****
subroutine CreatAVCircle_Manually(NBB)

```



```

integer (i4), intent(out) :: NBB
integer (i4) :: i, j
real (r8) :: theta

!!!

write(*, '(a,$)') 'How many bundles? --> '
read(*,*) NBB

allocate(Away(NBB), Vein(NBB))

do i=1, NBB
write(*, '(4x,a,$)') 'xi, yi, ri, ro --> '
read(*,*) Away(i)%ci%x, Away(i)%ci%y, Away(i)%ci%r, Away(i)%co%r
  Away(i)%co%x = Away(i)%ci%x
  Away(i)%co%y = Away(i)%ci%y
  !
write(*, '(4x,a,$)') 'theta(degree, relative to +x), ra --> '
read(*,*) theta, Away(i)%ca%r
  theta = 2*PI*theta/360.0
  Away(i)%ca%x = Away(i)%ci%x + (Away(i)%co%r+Away(i)%ca%r)*cos(theta)
  Away(i)%ca%y = Away(i)%ci%y + (Away(i)%co%r+Away(i)%ca%r)*sin(theta)
  !
write(*, '(4x,a,$)') 'xv, yv, rv --> '
read(*,*) Vein(i)%x, Vein(i)%y, Vein(i)%r
end do

end subroutine CreatAVCircle_Manually

!*****
! -- fill the lung region with just veins (water pipes)
!*****
subroutine CreatWaterStraws(RlztN,NBB,rv)

integer (i8), intent(in) :: RlztN
integer (i4), intent(in) :: NBB
real (r8), intent(in) :: rv

integer (i4) :: i, ib, iv
real (r8) :: Srest, Mrest, Mtmp, xv, yv, ds, sr

!!!

Srest = ZERO
do i=1, NBB
  Srest = Srest + PI*( (Away(i)%co%r)**2 + (Away(i)%ca%r)**2 + (Vein(i)%r)**2 )
end do
Srest = LX*LY - Srest ! the lung area less the bundles'
Mrest = Rho4*Srest
Mtmp = Rho(bBod(2,2,2)%mat)*Srest

iv = 0
do while(Mtmp<Mrest)
ck: do
  xv = rang()*LX
  yv = rang()*LY
  ! should be inside the lung
  if( xv-rv>=ZERO.and.xv+rv<=LX.and.&
     yv-rv>=ZERO.and.yv+rv<=LY ) then
  ! -- check the bundles/veins
  do ib=1, NBB
    ! distance between (xv,yv) and the airway origin
    ds = sqrt((xv-Away(ib)%co%x)**2 + (yv-Away(ib)%co%y)**2)
    ! sum of rv and the airway radius
    sr = rv + Away(ib)%co%r
    if(ds<sr) cycle ck
    ! distance between (xv,yv) and the artery origin

```

```

        ds = sqrt((xv-Away(ib)%ca%x)**2 + (yv-Away(ib)%ca%y)**2)
        ! sum of rv and the artery radius
        sr = rv + Away(ib)%ca%r
        if(ds<sr) cycle ck
        ! distance between (xv,yv) and the vein origin
        ds = sqrt((xv-Vein(ib)%x)**2 + (yv-Vein(ib)%y)**2)
        ! sum of rv and the vein radius
        sr = rv + Vein(ib)%r
        if(ds<sr) cycle ck
    end do
    ! -- check the previous straws
    if(iv>0) then
        do i=1, iv
            ! distance between (xv,yv) and the vein origin
            ds = sqrt((xv-Straw(i)%x)**2 + (yv-Straw(i)%y)**2)
            ! sum of rv and the vein radius
            sr = rv + Straw(i)%r
            if(ds<sr) cycle ck
        end do
    end if
    iv = iv + 1
    Straw(iv)%x = xv
    Straw(iv)%y = yv
    Straw(iv)%r = rv
    Mtmp = Mtmp + (Rho(water)-Rho(bBod(2,2,2)%mat))*PI*(Straw(iv)%r)**2
    exit ck
end if
end do ck
end do

NStraw = iv
write(*,'(A,g16.7,A,g16.7)') &
'The ratio of the parenchyma density of this realization to the prescribed:',&
Rho4,'g/cm^3 is --> ', Mtmp/Mrest

end subroutine CreatWaterStraws

!*****
! -- create all airway/vessel bundles in the lung region (circles)
!*****
subroutine CreatAVCircle(Rlztn)

integer (i8), intent(in) :: Rlztn

integer :: i, j

!!!
! -- do the 3D to 2D mapping for all orders down to respiratory ducts
call Determine_NBundle_Num

! -- Get the bundle numbers according to input
! -- NB is the largest number of bundles down to the smallest order
NB = Get_nbd(NAVG,NBundle(NAVG))
! -- NB_TB is always the bundle number at the terminal bronchiole order
NB_TB = Get_nbd(NAVGGroup_TB,NBundle(NAVGGroup_TB))
write(*,*) 'NB = ', NB, ' NB_TB = ', NB_TB
if(NAVG>=NAVGGroup_TB) then
    allocate(Away(NB_TB), Vein(NB_TB), AAci(NB-NB_TB))
else
    allocate(Away(NB), Vein(NB), AAci(0))
end if

do i=1, min(NAVG,NAVGGroup_TB)
    do j=1, NBundle(i)
        call CreatAVC(i,j)
    end do
end do

```

```

end do

do i=NAVGroup_TB+1, NAVG
  do j=1, NBundle(i)
    call CreatAci(i,j)
    write(*,*) 'Group ',i,', bundle --> ', j, ' of ', NBundle(i)
  end do
end do

end subroutine CreatAVCircle

!*****
!
! determine the particular # of bundles of each group
!
! m = (L*W/V_lung)*(n*l)
! where
! m = # of bundles of each group after adjustment
! V_lung = typical volume of both lungs (6000 mL)
! L*W = length*width of the 2D model
! n = # of bronchi of each group
! l = length of bronchi of each group
!
!*****

subroutine Determine_NBundle_Num

integer (i4) :: i, imin
real (r8) :: lmin, smax, residual
character (i1) :: yesno1, yesno2

!!!

  lmin = NLumen(1)*LLumen(1)
  imin = 1
  do i=2, NAVGroup
    smax = NLumen(i)*LLumen(i)
    if(smax<lmin) then
      lmin = smax
    imin = i
  end if
end do

  smax = VLung/lmin
10 write(*, '(A,/2x,g23.16,A)') 'The minimal area required to construct a 2D rectangle is: ',&
    smax, ' cm^2'

  LXB = sqrt(smax)
  LYB = LXB
  write(*, '(A,/2x,g23.16,A)') 'The width of the default square is: ', LXB, ' cm'
  write(*, '(A,$)') 'Input the length (LXB) (so that the width (LYB)) manually? (y/n) '
  read(*,*) yesno1
  select case(yesno1)
  case ('y', 'Y')
    write(*, '(/2x,A,$)') 'LXB --> '
    read(*,*) LXB
    LYB = smax/LXB
    write(*, '(/2x,A,/2x,g23.16,a1,g23.16,a5)') 'The buffer region is: ', LXB, '*', LYB, ' cm^2'
    write(*, '(2x,A,$)') 'Satisfied? (y/n) '
    read(*,*) yesno2
    select case(yesno2)
    case ('y', 'Y')
      case default
        go to 10
    end select
  case default
  end select
end select

```

```

do i=1, NAVGroup
  residual = 1/lmin*NLumen(i)*LLumen(i)
  NBundle(i) = int(residual)
  residual = residual - dble(NBundle(i))
  if(rang(<residual) NBundle(i) = NBundle(i) + 1
end do
NBundle(imin) = 1

end subroutine Determine_NBundle_Num

!*****
! function for getting the absolute index of the
! nc-th bundle of group ng
!*****
integer (i4) function Get_nbd(ng,nc) result(nbb)

integer (i4), intent(in) :: ng, nc

integer (i4) :: ig, ic

!!!

nbb = 0
do ig=1, ng-1
  nbb = nbb + NBundle(ig)
end do
nbb = nbb + nc

end function Get_nbd

!*****
! create an airway/vessel bundle in the buffered lung region
! (i.e., generate the according (r,x0,y0) for each single circle in
! the buffered lung)
!
! ng -- this group #
! nc -- this bundle # in this group
!*****
subroutine CreatAVC(ng, nc)

integer (i4), intent(in) :: ng, nc

integer (i4) :: i, j, nbd
real (r8) :: x, y, ri, ro, rv, rbd, rv2b
real (r8) :: xbd, ybd, x1, y1, x2, y2, uu, vv, alpha, theta

!!!

ri = DLumen(ng)/2
ro = ri*(1+PShell)
rv = ri*PVssl
rbd = ro + rv

nbd = Get_nbd(ng,nc)

bd: do
  xbd = rang()*LXB
  ybd = rang()*LYB

  if(AVCheck(xbd,ybd,rbd,nbd,'bundle')) then
    ! bundle
    Away(nbd)%cbd%x = xbd
    Away(nbd)%cbd%y = ybd
    Away(nbd)%cbd%r = rbd
    !
    alpha = 2*PI*rang() ! angle between vector (r1-r0) and x-axis

```

```

! artery
uu = dcos(alpha)
vv = dsin(alpha)
x1 = xbd + ro*uu
y1 = ybd + ro*vv
!
Away(nbd)%ca%x = x1
Away(nbd)%ca%y = y1
Away(nbd)%ca%r = rv
! airway
uu = -dcos(alpha)
vv = -dsin(alpha)
x = xbd + rv*uu
y = ybd + rv*vv
!
  Away(nbd)%ci%x = x
  Away(nbd)%ci%y = y
  Away(nbd)%co%x = Away(nbd)%ci%x
  Away(nbd)%co%y = Away(nbd)%ci%y
  Away(nbd)%ci%r = ri
  Away(nbd)%co%r = ro
!
if(ng>CVn2Bd1) then
  exit bd
else
  rv2b = Away(nbd)%cbd%r*(1 + PVn2Bd1) + rv
  alpha = 2*PI*rang()
  uu = dcos(alpha)
  vv = dsin(alpha)
  x2 = Away(nbd)%cbd%x + rv2b*uu
  y2 = Away(nbd)%cbd%y + rv2b*vv
  if(AVCheck(x2,y2,rv,nbd,'vein')) then
! vein
Vein(nbd)%x = x2
Vein(nbd)%y = y2
Vein(nbd)%r = rv
!
return
  end if
end if
  end if
end do bd

vn: do
  x2 = rang()*LXB
  y2 = rang()*LYB
  if(AVCheck(x2,y2,rv,nbd,'vein')) then
! vein
Vein(nbd)%x = x2
Vein(nbd)%y = y2
Vein(nbd)%r = rv
!
exit vn
  end if
end do vn

end subroutine CreatAVC

!*****
! create a pure airway in the buffered lung region for the acinus part
! (i.e., generate the according (r,x0,y0) for each single circle in
! the buffered lung)
!
! ng -- this group #
! nc -- this bundle # in this group
!*****
subroutine CreatAci(ng, nc)

```

```

integer (i4), intent(in) :: ng, nc

integer (i4) :: i, j, nbd
real (r8) :: x, y, rv
real (r8) :: x2, y2

!!!

rv = DLumen(ng)/2

nbd = Get_nbd(ng,nc) - NB_TB

ac: do
  x2 = rang()*LXB
  y2 = rang()*LYB
  if(AVCheck(x2,y2,rv,nbd,'acinus')) then
    ! vein
    AAci(nbd)%x = x2
    AAci(nbd)%y = y2
    AAci(nbd)%r = rv
    !
  exit ac
end if
end do ac

end subroutine CreatAci

!*****
! function for checking the validity of a circle
!*****
logical function AVCheck(x,y,r,nbd,cha) result(flag)

real(8), intent(in) :: x, y, r
integer, intent(in) :: nbd
character (len=*), intent(in) :: cha

integer :: ib
real (8) :: ds, sr

!!!

flag = .false.

if( x-r>=ZERO.and.x+r<=LXB.and.&
y-r>=ZERO.and.y+r<=LYB      ) then
  if(cha=='acinus') then
    do ib=1, NB_TB
      ! distance between (x,y) and the airway origin
      ds = sqrt((x-Away(ib)%co%x)**2 + (y-Away(ib)%co%y)**2)
      ! sum of r and the airway radius
      sr = r + Away(ib)%co%r
      if(ds<sr) return
      ! distance between (x,y) and the artery origin
      ds = sqrt((x-Away(ib)%ca%x)**2 + (y-Away(ib)%ca%y)**2)
      ! sum of r and the artery radius
      sr = r + Away(ib)%ca%r
      if(ds<sr) return
      ! distance between (x,y) and the vein origin
      ds = sqrt((x-Vein(ib)%x)**2 + (y-Vein(ib)%y)**2)
      ! sum of r and the vein radius
      sr = r + Vein(ib)%r
      if(ds<sr) return
    end do
    do ib=NB_TB+1, nbd-1
      ! distance between (x,y) and the airway origin
      ds = sqrt((x-AAci(ib)%x)**2 + (y-AAci(ib)%y)**2)

```

```

    ! sum of r and the airway radius
    sr = r + AAci(ib)%r
    if(ds<sr) return
end do
else
! -- check the lower rank bundles/veins
do ib=1, nbd-1
    ! distance between (x,y) and the airway origin
    ds = sqrt((x-Away(ib)%co%x)**2 + (y-Away(ib)%co%y)**2)
    ! sum of r and the airway radius
    sr = r + Away(ib)%co%r
    if(ds<sr) return
    ! distance between (x,y) and the artery origin
    ds = sqrt((x-Away(ib)%ca%x)**2 + (y-Away(ib)%ca%y)**2)
    ! sum of r and the artery radius
    sr = r + Away(ib)%ca%r
    if(ds<sr) return
    ! distance between (x,y) and the vein origin
    ds = sqrt((x-Vein(ib)%x)**2 + (y-Vein(ib)%y)**2)
    ! sum of r and the vein radius
    sr = r + Vein(ib)%r
    if(ds<sr) return
end do
! -- check the same bundle of the same rank, if vein
if(cha=='vein') then
    ib = nbd
    ! distance between (x,y) and the airway origin
    ds = sqrt((x-Away(ib)%co%x)**2 + (y-Away(ib)%co%y)**2)
    ! sum of r and the airway radius
    sr = r + Away(ib)%co%r
    if(ds<sr) return
    ! distance between (x,y) and the artery origin
    ds = sqrt((x-Away(ib)%ca%x)**2 + (y-Away(ib)%ca%y)**2)
    ! sum of r and the artery radius
    sr = r + Away(ib)%ca%r
    if(ds<sr) return
end if
end if
!
flag = .true.
end if

end function AVCheck

!*****
! function for checking the validity of a circle
!*****
logical function XSphereCheck(Cir,Sph) result(flag)

    type (Circle), intent(in) :: Cir
    type (Sphere), intent(in) :: Sph

    real (r8) :: RT, O102

!!!

flag = .false.

    RT = Cir%r + Sph%r
    O102 = sqrt((Cir%x - Sph%x)**2 + (Cir%y - Sph%y)**2)

if(O102<RT) flag = .true.

end function XSphereCheck

end module GeoGen_CreatAVC

```

```

!*****
!
! output the geometry file
!
!*****
module GeoGen_Output

use GeoGen_CreatAVC
use GeoGen_surfaceGen

implicit none

integer (i4) :: i, j, k, ii
real (r8) :: xcut(2), ycut(2), dxct(2), dyct(2), dcnr(4)

contains

!*****
!
! subroutine Output -- main output
!
!*****
subroutine Output

integer (i4) :: ichange=0
integer (i4) :: nbd, sNum, bmNum, bNum, mNum, &
nci, nco, nca, ncv, ncs, &
ncim, ncom, ncam, ncvm, ncsm
integer (i4) :: ib, im, &
ici, ico, ica, icv, ics, &
icim, icom, icam, icvm, icsm
logical :: flag(6), flag_ci_in
integer (i4), allocatable :: iModule(:), iBody(:)
type (Circle), allocatable :: ci(:), co(:), ca(:), cv(:), cs(:), &
cim(:), com(:), cam(:), cvm(:), csm(:)

!!!

! used as penelope input geo file
open(unit=io_o, file='geo.'//suffix(1:len_sfx)//'.geo', action='WRITE', status='REPLACE')
! used as PenDosesMass input to calculate mass for dose grid
open(unit=io_m, file='mass.'//suffix(1:len_sfx)//'.geo', action='WRITE', status='REPLACE')

! -- begin
write(io_o,10)

! -- the basic planes
do i=1, 12
j = len_trim(bPln(i)%typ)
! -- surface #
write(io_o,20) LSUR, bPln(i)%num, bPln(i)%typ(1:j)
! -- 5 indices
write(io_o,30) LIND, (bPln(i)%idx(j), j=1,5)
! -- xscale
if(bPln(i)%xscl/=rPln%xscl) then
write(io_o,40) LXSC, bPln(i)%xscl, ichange, LOPEN
end if
! -- yscale
if(bPln(i)%yscl/=rPln%yscl) then
write(io_o,40) LYSC, bPln(i)%yscl, ichange, LOPEN
end if
! -- zscale
if(bPln(i)%zscl/=rPln%zscl) then
write(io_o,40) LZSC, bPln(i)%zscl, ichange, LOPEN
end if
! -- omega
if(bPln(i)%omg/=rPln%omg) then

```



```

    write(io_o,40) LOME, bPln(i)%omg,  ichange, LDEG
end if
! -- theta
if(bPln(i)%ttht/=rPln%ttht) then
    write(io_o,40) LTHE, bPln(i)%ttht,  ichange, LDEG
end if
! -- phi
if(bPln(i)%phi/=rPln%phi) then
    write(io_o,40) LPHI, bPln(i)%phi,  ichange, LDEG
end if
! -- xshift
if(bPln(i)%xshft/=rPln%xshft) then
    write(io_o,40) LXSH, bPln(i)%xshft,  ichange, LOPEN
end if
! -- yshift
if(bPln(i)%yshft/=rPln%yshft) then
    write(io_o,40) LYSH, bPln(i)%yshft,  ichange, LOPEN
end if
! -- zshift
if(bPln(i)%zshft/=rPln%zshft) then
    write(io_o,40) LZSH, bPln(i)%zshft,  ichange, LOPEN
end if
    write(io_o,10)
end do
10 format(64('0')) ! format label 2001 in pengeom.f
20 format(A8,'(,I4,') ',A) ! format label 1001, 2002 in pengeom.f
30 format(A8,'(,4(I2,',',I2,')')') ! format label 2003 in pengeom.f
40 format(A8,'(,1P,E22.15,',',I3,A8) ! format label 2004 in pengeom.f
50 format(A8,'(,I4,')') ! format label 1004, 2005, 2007 in pengeom.f
60 format(A8,'(,I4,')', SIDE POINTER=(,I2,')') ! format label 1005, 2006 in pengeom.f

! -- The basic bodies
do k=1, 3
    do j=1, 3
        do i=1, 3
            if(i==2.and.j==2.and.k==2) then
                if(flag_tumor.or.TofL/=1) cycle
            end if
            write(io_o,20) LBOD, bBod(i,j,k)%num, bBod(i,j,k)%id
            write(io_o,50) LMAT, bBod(i,j,k)%mat
            do ii=1, size(bBod(i,j,k)%sur)
                write(io_o,60) LSUR, bBod(i,j,k)%sur(ii), bBod(i,j,k)%sid(ii)
            end do
            do ii=1, size(bBod(i,j,k)%bod)
                write(io_o,50) LBOD, bBod(i,j,k)%bod(ii)
            end do
            write(io_o,10)
        end do ! i
    end do ! j
end do ! k

! -- Tumor
if(flag_tumor) then
    select case(SofT)
    case (2,4)
        STumor%x = STumor%x - LX/2
        STumor%z = STumor%z - LZ/2
        SurTumor = sphereSurGen(STumor)
        SurTumor%num = 13 ! follow the previous 12 basic planes
        BodyTumor%num = 28 ! follow the previous 27 basic bodies
        !
        BodyTumor%id = 'SphericalTumor'
        BodyTumor%mat = water
        allocate(BodyTumor%sur(1),BodyTumor%sid(1),BodyTumor%bod(0))
        BodyTumor%sur(1) = SurTumor%num
        BodyTumor%sid(1) = -1

        write(io_o,20) LSUR, SurTumor%num, 'tumor sphere'

```

```

write(io_o,30) LIND, (SurTumor%idx(k), k=1,5)
! -- xscale
  if(SurTumor%xsc1/=rSph%xsc1) then
    write(io_o,40) LXSC, SurTumor%xsc1, ichtange, LOPEN
  end if
! -- yscale
  if(SurTumor%ysc1/=rSph%ysc1) then
    write(io_o,40) LYSC, SurTumor%ysc1, ichtange, LOPEN
  end if
! -- zscale
  if(SurTumor%zsc1/=rSph%zsc1) then
    write(io_o,40) LZSC, SurTumor%zsc1, ichtange, LOPEN
  end if
! -- xshift
  if(SurTumor%xshft/=rSph%xshft) then
    write(io_o,40) LXSH, SurTumor%xshft, ichtange, LOPEN
  end if
! -- yshift
  if(SurTumor%yshft/=rSph%yshft) then
    write(io_o,40) LYSH, SurTumor%yshft, ichtange, LOPEN
  end if
! -- zshift
  if(SurTumor%zshft/=rSph%zshft) then
    write(io_o,40) LZSH, SurTumor%zshft, ichtange, LOPEN
  end if
write(io_o,10)
!
write(io_o,20) LBOD, BodyTumor%num, BodyTumor%id
write(io_o,50) LMAT, BodyTumor%mat
do ii=1, size(BodyTumor%sur)
  write(io_o,60) LSUR, BodyTumor%sur(ii), BodyTumor%sid(ii)
end do
do ii=1, size(BodyTumor%bod)
  write(io_o,50) LBOD, BodyTumor%bod(ii)
end do
write(io_o,10)
case default
end select
end if

! -- various types of explicit random structure

select case(TofL)
case (1) ! -- Atomic mix, just the parenchyma
  ! -- lung's parenchyma
  if(flag_tumor) then
    write(io_o,20) LBOD, bBod(2,2,2)%num, 'lung''s parenchyma'
    write(io_o,50) LMAT, bBod(2,2,2)%mat
    write(io_o,60) LSUR, 2, +1
    write(io_o,60) LSUR, 3, -1
    write(io_o,60) LSUR, 6, +1
    write(io_o,60) LSUR, 7, -1
    write(io_o,60) LSUR, 10, +1
    write(io_o,60) LSUR, 11, -1
    write(io_o,60) LSUR, SurTumor%num, +1
  write(io_o,10)
  end if
  !
  write(io_m,*) 'For atomic mix, don''t need mass.geo and use PenDosesMass directly!'
  write(*,*) 'For atomic mix, don''t need mass.geo and use PenDosesMass directly!'
case (2) ! -- Arbitrary cylinders input manually + background atomic mix
case (3) ! -- Actual lung: airway/vessel bundles + background atomic mix
  xcut(1) = Lox
  xcut(2) = Lox + LX
  ycut(1) = Loy
  ycut(2) = Loy + LY
  write(io_v,'(2/A/)' ) '[Cut-out region limits in the lung]'
70  format(a2,2x,g23.16)

```

```

write(io_v,70) 'x1', xcut(1)
write(io_v,70) 'x2', xcut(2)
write(io_v,70) 'y1', ycut(1)
write(io_v,70) 'y2', ycut(2)
sNum = 13 ! surface #
bmNum = 28 ! body/module #
bNum = 0 ! body #
mNum = 0 ! module #
nci = 0 ! for view (in the whole buffer)
nco = 0
nca = 0
ncv = 0
ncs = 0
ncim = 0 ! for mass (in the lung)
ncom = 0
ncam = 0
ncvm = 0
ncsm = 0

! -- the airway+artery+vein bundles down to terminal bronchiole
do i=1, min(NAVG, NAVGroup_TB)
  do j=1, NBundle(i)
    nbd = Get_nbd(i,j)
    flag_ci_in = .false.
    ! -- lumen
    call Location_2_Cutout(AWay(nbd)%ci,flag)
    nci = nci + 1
    if(.not.flag(2)) then
      flag_ci_in = .true.
    ncim = ncim + 1
    end if
    ! -- wall
    call Location_2_Cutout(AWay(nbd)%co,flag)
    nco = nco + 1
    if(.not.flag(2)) then
      if(flag_ci_in) then
        mNum = mNum + 1
      else
        bNum = bNum + 1
      end if
      ncom = ncom + 1
    end if
    ! -- artery
    call Location_2_Cutout(AWay(nbd)%ca,flag)
    nca = nca + 1
    if(.not.flag(2)) then
      bNum = bNum + 1
      ncam = ncam + 1
    end if
    ! -- vein
    call Location_2_Cutout(Vein(nbd),flag)
    ncv = ncv + 1
    if(.not.flag(2)) then
      bNum = bNum + 1
      ncvm = ncvm + 1
    end if
  end do ! j
end do ! i

! -- the airways in the acini
do i=NAVGroup_TB+1, NAVG
  do j=1, NBundle(i)
    nbd = Get_nbd(i,j) - NB_TB
    ! -- acinus airway lumen
    call Location_2_Cutout(AAci(nbd),flag)
    ncs = ncs + 1
    if(.not.flag(2)) then
      bNum = bNum + 1
    end if
  end do
end do

```

```

    nscm = nscm +1
  end if
end do
end do

  if(bNum>0) allocate(iBody(bNum))
if(mNum>0) allocate(iModule(mNum))
!
  if(nci>0) allocate(ci(nci))
  if(nco>0) allocate(co(nco))
  if(nca>0) allocate(ca(nca))
  if(ncv>0) allocate(cv(ncv))
  if(ncs>0) allocate(cs(ncs))
!
  if(ncim>0) allocate(cim(ncim))
  if(ncom>0) allocate(com(ncom))
  if(ncam>0) allocate(cam(ncam))
  if(ncvm>0) allocate(cvm(ncvm))
  if(ncsm>0) allocate(csm(ncsm))

  ! -- now do it again for output based on the above calculation
ib = 0      ! body # used to define the parenchyma module so the airway lumen not counted
im = 0      ! module # (the airway) used to define the parenchyma module
nci = 0
nco = 0
nca = 0
ncv = 0
ncs = 0
ncim = 0
ncom = 0
ncam = 0
ncvm = 0
ncsm = 0

! -- the airway+artery+vein bundles down to terminal bronchiole
do i=1, min(NAVG, NAVGroup_TB)
  do j=1, NBundle(i)
    nbd = Get_nbd(i,j)
    flag_ci_in = .false.
    ! -- lumen
    call Location_2_Cutout(AWay(nbd)%ci,flag)
    nci = nci + 1
    ci(nci)%x = AWay(nbd)%ci%x-(xcut(1)+xcut(2))/2
    ci(nci)%y = AWay(nbd)%ci%y- ycut(1)
    ci(nci)%r = AWay(nbd)%ci%r
    if(.not.flag(2)) then
      flag_ci_in = .true.
      ! -- airway inner cylinder
      sNum = sNum + 1
      write(io_o,20) LSUR, sNum, 'Airway inner cylinder x^2 + y^2 = ri^2'
      write(io_o,30) LIND, (rCyl%idx(k), k=1,5)
      write(io_o,40) LXSC, AWay(nbd)%ci%r, ichange, LOPEN
      write(io_o,40) LYSC, AWay(nbd)%ci%r, ichange, LOPEN
      write(io_o,40) LXSH, AWay(nbd)%ci%x-(xcut(1)+xcut(2))/2, ichange, LOPEN
      write(io_o,40) LYSH, AWay(nbd)%ci%y- ycut(1), ichange, LOPEN
      write(io_o,10)
    end if
    ncim = ncim + 1
    cim(ncim)%x = AWay(nbd)%ci%x-(xcut(1)+xcut(2))/2
    cim(ncim)%y = AWay(nbd)%ci%y- ycut(1)
    cim(ncim)%r = AWay(nbd)%ci%r
    !
    bmNum = bmNum + 1
    write(io_o,20) LBOD, bmNum, 'airway lumen'
    write(io_o,50) LMAT, air
    write(io_o,60) LSUR, 10, +1
    write(io_o,60) LSUR, 11, -1
    write(io_o,60) LSUR, sNum, -1
  end do
  if(flag_tumor.and.XSphereCheck(cim(ncim),STumor)) then

```

```

    write(io_o,60) LSUR, SurTumor%num, +1
end if
if(flag(3)) write(io_o,60) LSUR, 2, +1
if(flag(4)) write(io_o,60) LSUR, 3, -1
if(flag(5)) write(io_o,60) LSUR, 6, +1
if(flag(6)) write(io_o,60) LSUR, 7, -1
    write(io_o,10)
end if ! flag(2)
! -- wall
call Location_2_Cutout(AWay(nbd)%co,flag)
nco = nco + 1
co(nco)%x = AWay(nbd)%co%x-(xcut(1)+xcut(2))/2
co(nco)%y = AWay(nbd)%co%y- ycut(1)
co(nco)%r = AWay(nbd)%co%r
if(.not.flag(2)) then
    ! -- airway outer cylinder= ro^2
    sNum = sNum + 1
write(io_o,20) LSUR, sNum, 'Airway outer cylinder x^2 + y^2 = ro^2'
    write(io_o,30) LIND, (rCyl%idx(k), k=1,5)
    write(io_o,40) LXSC, AWay(nbd)%co%r, ichange, LOPEN
    write(io_o,40) LYSC, AWay(nbd)%co%r, ichange, LOPEN
write(io_o,40) LXSH, AWay(nbd)%co%x-(xcut(1)+xcut(2))/2, ichange, LOPEN
write(io_o,40) LYSH, AWay(nbd)%co%y- ycut(1), ichange, LOPEN
    write(io_o,10)
ncom = ncom + 1
com(ncom)%x = AWay(nbd)%co%x-(xcut(1)+xcut(2))/2
com(ncom)%y = AWay(nbd)%co%y- ycut(1)
com(ncom)%r = AWay(nbd)%co%r
    !
    bmNum = bmNum + 1
if(flag_ci_in) then
    write(io_o,20) LMOD, bmNum, 'airway wall'
    im = im + 1
    iModule(im) = bmNum
else
    write(io_o,20) LBOD, bmNum, 'airway wall'
    ib = ib + 1
    iBody(ib) = bmNum
end if
    write(io_o,20) LMAT, water
    write(io_o,60) LSUR, 10, +1
    write(io_o,60) LSUR, 11, -1
    write(io_o,60) LSUR, sNum, -1
if(flag_tumor.and.XSphereCheck(com(ncom),STumor)) then
    write(io_o,60) LSUR, SurTumor%num, +1
end if
if(flag(3)) write(io_o,60) LSUR, 2, +1
if(flag(4)) write(io_o,60) LSUR, 3, -1
if(flag(5)) write(io_o,60) LSUR, 6, +1
if(flag(6)) write(io_o,60) LSUR, 7, -1
if(flag_ci_in) write(io_o,50) LBOD, bmNum-1
    write(io_o,10)
end if ! flag(2)
! -- artery
call Location_2_Cutout(AWay(nbd)%ca,flag)
nca = nca + 1
ca(nca)%x = AWay(nbd)%ca%x-(xcut(1)+xcut(2))/2
ca(nca)%y = AWay(nbd)%ca%y- ycut(1)
ca(nca)%r = AWay(nbd)%ca%r
if(.not.flag(2)) then
    ! -- airway's accompanying artery
    sNum = sNum + 1
write(io_o,20) LSUR, sNum, 'artery cylinder x^2 + y^2 = ra^2'
    write(io_o,30) LIND, (rCyl%idx(k), k=1,5)
    write(io_o,40) LXSC, AWay(nbd)%ca%r, ichange, LOPEN
    write(io_o,40) LYSC, AWay(nbd)%ca%r, ichange, LOPEN
write(io_o,40) LXSH, AWay(nbd)%ca%x-(xcut(1)+xcut(2))/2, ichange, LOPEN
write(io_o,40) LYSH, AWay(nbd)%ca%y- ycut(1), ichange, LOPEN

```

```

    write(io_o,10)
ncam = ncam + 1
cam(ncam)%x = AWay(nbd)%ca%x-(xcut(1)+xcut(2))/2
cam(ncam)%y = AWay(nbd)%ca%y- ycut(1)
cam(ncam)%r = AWay(nbd)%ca%r
!
    bmNum = bmNum + 1
ib = ib + 1
iBody(ib) = bmNum
    write(io_o,20) LBOD, bmNum, 'artery'
    write(io_o,50) LMAT, water
    write(io_o,60) LSUR, 10, +1
    write(io_o,60) LSUR, 11, -1
write(io_o,60) LSUR, sNum, -1
if(flag_tumor.and.XSphereCheck(cam(ncam),STumor)) then
    write(io_o,60) LSUR, SurTumor%num, +1
end if
if(flag(3)) write(io_o,60) LSUR, 2, +1
if(flag(4)) write(io_o,60) LSUR, 3, -1
if(flag(5)) write(io_o,60) LSUR, 6, +1
if(flag(6)) write(io_o,60) LSUR, 7, -1
    write(io_o,10)
    end if ! flag(2)
! -- vein
call Location_2_Cutout(Vein(nbd),flag)
ncv = ncv + 1
cv(ncv)%x = Vein(nbd)%x-(xcut(1)+xcut(2))/2
cv(ncv)%y = Vein(nbd)%y- ycut(1)
cv(ncv)%r = Vein(nbd)%r
if(.not.flag(2)) then
    ! -- Vein
    sNum = sNum + 1
write(io_o,20) LSUR, sNum, 'vein cylinder x^2 + y^2 = rv^2'
    write(io_o,30) LIND, (rCyl%idx(k), k=1,5)
    write(io_o,40) LXSC, Vein(nbd)%r, ichange, LOPEN
    write(io_o,40) LYSC, Vein(nbd)%r, ichange, LOPEN
write(io_o,40) LXSH, Vein(nbd)%x-(xcut(1)+xcut(2))/2, ichange, LOPEN
write(io_o,40) LYSH, Vein(nbd)%y- ycut(1), ichange, LOPEN
    write(io_o,10)
ncvm = ncvm + 1
cvm(ncvm)%x = Vein(nbd)%x-(xcut(1)+xcut(2))/2
cvm(ncvm)%y = Vein(nbd)%y- ycut(1)
cvm(ncvm)%r = Vein(nbd)%r
!
    bmNum = bmNum + 1
ib = ib + 1
iBody(ib) = bmNum
    write(io_o,20) LBOD, bmNum, 'vein'
    write(io_o,50) LMAT, water
    write(io_o,60) LSUR, 10, +1
    write(io_o,60) LSUR, 11, -1
write(io_o,60) LSUR, sNum, -1
if(flag_tumor.and.XSphereCheck(cvm(ncvm),STumor)) then
    write(io_o,60) LSUR, SurTumor%num, +1
end if
if(flag(3)) write(io_o,60) LSUR, 2, +1
if(flag(4)) write(io_o,60) LSUR, 3, -1
if(flag(5)) write(io_o,60) LSUR, 6, +1
if(flag(6)) write(io_o,60) LSUR, 7, -1
    write(io_o,10)
end if
end do ! NBundle(i)
end do ! NAVGroup

! -- the airways in the acini
do i=NAVGroup_TB+1, NAVG
do j=1, NBundle(i)
    nbd = Get_nbd(i,j) - NB_TB

```

```

! -- acinus airway lumen
call Location_2_Cutout(AAci(nbd),flag)
ncs = ncs + 1
cs(ncs)%x = AAci(nbd)%x-(xcut(1)+xcut(2))/2
cs(ncs)%y = AAci(nbd)%y- ycut(1)
cs(ncs)%r = AAci(nbd)%r
if(.not.flag(2)) then
! -- AAci
sNum = sNum + 1
write(io_o,20) LSur, sNum, 'AALumen cylinder x^2 + y^2 = rv^2'
write(io_o,30) LIND, (rCyl%idx(k), k=1,5)
write(io_o,40) LXSC, AAci(nbd)%r,  ichange, LOPEN
write(io_o,40) LYSC, AAci(nbd)%r,  ichange, LOPEN
write(io_o,40) LXSH, AAci(nbd)%x-(xcut(1)+xcut(2))/2,  ichange, LOPEN
write(io_o,40) LYSH, AAci(nbd)%y- ycut(1),          ichange, LOPEN
write(io_o,10)
ncsm = ncsm + 1
csm(ncsm)%x = AAci(nbd)%x-(xcut(1)+xcut(2))/2
csm(ncsm)%y = AAci(nbd)%y- ycut(1)
csm(ncsm)%r = AAci(nbd)%r
!
bmNum = bmNum + 1
ib = ib + 1
iBody(ib) = bmNum
write(io_o,20) LBOD, bmNum, 'AALumen'
write(io_o,50) LMAT, air
write(io_o,60) LSur, 10, +1
write(io_o,60) LSur, 11, -1
write(io_o,60) LSur, sNum, -1
if(flag_tumor.and.XSphereCheck(csm(ncsm),STumor)) then
write(io_o,60) LSur, SurTumor%num, +1
end if
if(flag(3)) write(io_o,60) LSur, 2, +1
if(flag(4)) write(io_o,60) LSur, 3, -1
if(flag(5)) write(io_o,60) LSur, 6, +1
if(flag(6)) write(io_o,60) LSur, 7, -1
write(io_o,10)
end if
end do ! NBundle(i)
end do ! NAVGroup

case (4)
write(io_v,'(2/A/)' ) '[Cut-out region limits in the lung]'
write(io_v,70) 'x1', 0
write(io_v,70) 'x2', LX
write(io_v,70) 'y1', 0
write(io_v,70) 'y2', LY
sNum = 13 ! surface #
bmNum = 28 ! body/module #
bNum = 0 ! body #
mNum = 0 ! module #
im = 0
ib = 0
nci = 0 ! for view (in the whole buffer)
nco = 0
nca = 0
ncv = 0
ncim = 0 ! for view (in the whole buffer)
ncom = 0
ncam = 0
ncvm = 0

allocate(ci(NBdle), co(NBdle), ca(NBdle), cv(NBdle+NStraw), &
cim(NBdle),com(NBdle),cam(NBdle),cvm(NBdle+NStraw), &
iBody(3*NBdle+NStraw),iModule(NBdle))

do i=1, NBdle
! -- lumen

```

```

    nci = nci + 1
    ci(nci)%x = AWay(i)%ci%x-LX/2
    ci(nci)%y = AWay(i)%ci%y
    ci(nci)%r = AWay(i)%ci%r
    ! -- airway inner cylinder
    sNum = sNum + 1
    write(io_o,20) LSUR, sNum, 'Airway inner cylinder x^2 + y^2 = ri^2'
    write(io_o,30) LIND, (rCyl%idx(k), k=1,5)
    write(io_o,40) LXSC, ci(nci)%r, ichange, LOPEN
    write(io_o,40) LYSC, ci(nci)%r, ichange, LOPEN
    write(io_o,40) LXSH, ci(nci)%x, ichange, LOPEN
    write(io_o,40) LYSH, ci(nci)%y, ichange, LOPEN
    write(io_o,10)
    ncim = ncim + 1
    cim(ncim) = ci(nci)
    !
    bmNum = bmNum + 1
    write(io_o,20) LBOD, bmNum, 'airway lumen'
    write(io_o,50) LMAT, air
    write(io_o,60) LSUR, 10, +1
    write(io_o,60) LSUR, 11, -1
    write(io_o,60) LSUR, sNum, -1
    if(flag_tumor.and.XSphereCheck(cim(ncim),STumor)) then
        write(io_o,60) LSUR, SurTumor%num, +1
    end if
    write(io_o,10)
    ! -- wall
    nco = nco + 1
    co(nco)%x = AWay(i)%co%x-LX/2
    co(nco)%y = AWay(i)%co%y
    co(nco)%r = AWay(i)%co%r
    ! -- airway outer cylinder= ro^2
    sNum = sNum + 1
    write(io_o,20) LSUR, sNum, 'Airway outer cylinder x^2 + y^2 = ro^2'
    write(io_o,30) LIND, (rCyl%idx(k), k=1,5)
    write(io_o,40) LXSC, co(nco)%r, ichange, LOPEN
    write(io_o,40) LYSC, co(nco)%r, ichange, LOPEN
    write(io_o,40) LXSH, co(nco)%x, ichange, LOPEN
    write(io_o,40) LYSH, co(nco)%y, ichange, LOPEN
    write(io_o,10)
    ncom = ncom + 1
    com(ncom) = co(nco)
    !
    bmNum = bmNum + 1
    write(io_o,20) LMOD, bmNum, 'airway wall'
    im = im + 1
    iModule(im) = bmNum
    write(io_o,20) LMAT, water
    write(io_o,60) LSUR, 10, +1
    write(io_o,60) LSUR, 11, -1
    write(io_o,60) LSUR, sNum, -1
    if(flag_tumor.and.XSphereCheck(com(ncom),STumor)) then
        write(io_o,60) LSUR, SurTumor%num, +1
    end if
    write(io_o,50) LBOD, bmNum-1
    write(io_o,10)
    ! -- artery
    nca = nca + 1
    ca(nca)%x = AWay(i)%ca%x-LX/2
    ca(nca)%y = AWay(i)%ca%y
    ca(nca)%r = AWay(i)%ca%r
    ! -- airway's accompanying artery
    sNum = sNum + 1
    write(io_o,20) LSUR, sNum, 'artery cylinder x^2 + y^2 = ra^2'
    write(io_o,30) LIND, (rCyl%idx(k), k=1,5)
    write(io_o,40) LXSC, ca(nca)%r, ichange, LOPEN
    write(io_o,40) LYSC, ca(nca)%r, ichange, LOPEN
    write(io_o,40) LXSH, ca(nca)%x, ichange, LOPEN

```



```

write(io_o,40) LYSH, ca(nca)%y, icheange, LOPEN
write(io_o,10)
ncam = ncam + 1
cam(ncam) = ca(nca)
!
bmNum = bmNum + 1
ib = ib + 1
iBody(ib) = bmNum
write(io_o,20) LBOD, bmNum, 'artery'
write(io_o,50) LMAT, water
write(io_o,60) LSUR, 10, +1
write(io_o,60) LSUR, 11, -1
write(io_o,60) LSUR, sNum, -1
if(flag_tumor.and.XSphereCheck(cam(ncam),STumor)) then
write(io_o,60) LSUR, SurTumor%num, +1
end if
write(io_o,10)
!-- vein
ncv = ncv + 1
cv(ncv)%x = Vein(i)%x-LX/2
cv(ncv)%y = Vein(i)%y
cv(ncv)%r = Vein(i)%r
!
sNum = sNum + 1
write(io_o,20) LSUR, sNum, 'vein cylinder x^2 + y^2 = rv^2'
write(io_o,30) LIND, (rCyl%idx(k), k=1,5)
write(io_o,40) LXSC, cv(ncv)%r, icheange, LOPEN
write(io_o,40) LYSC, cv(ncv)%r, icheange, LOPEN
write(io_o,40) LXSH, cv(ncv)%x, icheange, LOPEN
write(io_o,40) LYSH, cv(ncv)%y, icheange, LOPEN
write(io_o,10)
ncvm = ncvm + 1
cvm(ncvm) = cv(ncv)
!
bmNum = bmNum + 1
ib = ib + 1
iBody(ib) = bmNum
write(io_o,20) LBOD, bmNum, 'vein'
write(io_o,50) LMAT, water
write(io_o,60) LSUR, 10, +1
write(io_o,60) LSUR, 11, -1
write(io_o,60) LSUR, sNum, -1
if(flag_tumor.and.XSphereCheck(cvm(ncvm),STumor)) then
write(io_o,60) LSUR, SurTumor%num, +1
end if
write(io_o,10)
end do ! NBdle
do i=1, NStraw
ncv = ncv + 1
cv(ncv)%x = Straw(i)%x-LX/2
cv(ncv)%y = Straw(i)%y
cv(ncv)%r = Straw(i)%r
!
sNum = sNum + 1
write(io_o,20) LSUR, sNum, 'Straw cylinder x^2 + y^2 = rv^2'
write(io_o,30) LIND, (rCyl%idx(k), k=1,5)
write(io_o,40) LXSC, cv(ncv)%r, icheange, LOPEN
write(io_o,40) LYSC, cv(ncv)%r, icheange, LOPEN
write(io_o,40) LXSH, cv(ncv)%x, icheange, LOPEN
write(io_o,40) LYSH, cv(ncv)%y, icheange, LOPEN
write(io_o,10)
ncvm = ncvm + 1
cvm(ncvm) = cv(ncv)
!
bmNum = bmNum + 1
ib = ib + 1
iBody(ib) = bmNum
write(io_o,20) LBOD, bmNum, 'Straw'

```

```

write(io_o,50) LMAT, water
write(io_o,60) LSUR, 10, +1
write(io_o,60) LSUR, 11, -1
write(io_o,60) LSUR, sNum, -1
if(flag_tumor.and.XSphereCheck(cvm(ncvm),STumor)) then
  write(io_o,60) LSUR, SurTumor%num, +1
end if
write(io_o,10)
end do
  bNum = ib
  mNum = im
case default
end select

! -- end the input geo file by including the lung parenchyma module and
!   write geo files for viewing and mass calculation
select case(TofL)
case(3,4)
  ! -- lung's parenchyma
write(io_o,20) LMOD, bBod(2,2,2)%num, 'lung's parenchyma'
write(io_o,50) LMAT, bBod(2,2,2)%mat
write(io_o,60) LSUR, 2, +1
write(io_o,60) LSUR, 3, -1
write(io_o,60) LSUR, 6, +1
write(io_o,60) LSUR, 7, -1
write(io_o,60) LSUR, 10, +1
write(io_o,60) LSUR, 11, -1
if(flag_tumor) then
  write(io_o,60) LSUR, SurTumor%num, +1
end if
do ib=1, bNum
  write(io_o,50) LBOD, iBody(ib)
end do
do im=1, mNum
  write(io_o,50) LMOD, iModule(im)
end do
  write(io_o,10)

  ! -- geo file for viewing in matlab
  write(io_v,'(2/A/)') '[Circles within the Lung buffer]'
if(nci>0) then
  write(io_v,97) KWLumen, nci
  write(io_v,98) 'xi', 'yi', 'ri'
  do i=1, nci
    write(io_v,99) ci(i)%x, ci(i)%y, ci(i)%r
  end do
end if
!
if(nco>0) then
  write(io_v,'(/)')
  write(io_v,97) KWall, nco
  write(io_v,98) 'xo', 'yo', 'ro'
  do i=1, nco
    write(io_v,99) co(i)%x, co(i)%y, co(i)%r
  end do
end if
!
if(nca>0) then
  write(io_v,'(/)')
  write(io_v,97) KWArtery, nca
  write(io_v,98) 'xa', 'ya', 'ra'
  do i=1, nca
    write(io_v,99) ca(i)%x, ca(i)%y, ca(i)%r
  end do
end if
!
if(ncv>0) then
  write(io_v,'(/)')

```

```

write(io_v,97) KWVein, ncv
  write(io_v,98) 'xv', 'yv', 'rv'
do i=1, ncv
  write(io_v,99) cv(i)%x, cv(i)%y, cv(i)%r
end do
end if
!
if(ncs>0) then
write(io_v,'(/)')
write(io_v,97) KWAALmn, ncs
  write(io_v,98) 'xs', 'ys', 'rs'
do i=1, ncs
  write(io_v,99) cs(i)%x, cs(i)%y, cs(i)%r
end do
end if
97  format(a6,1x,i8/)
971 format(a6,1x,i8)
98  format(3(13x,A2,11x))
99  format(3g26.16)

! -- geo file for calculating the mass grid by PenDosesMass
write(io_m,'(A)') '[Circles within the Lung]'
if(ncim>0) then
write(io_m,971) KWLumen, ncim
write(io_m,*) KWLine
  write(io_m,98) 'xi', 'yi', 'ri'
do i=1, ncim
  write(io_m,99) cim(i)%x, cim(i)%y, cim(i)%r
end do
end if
!
if(ncom>0) then
write(io_m,'(/)')
write(io_m,971) KWWall, ncom
write(io_m,*) KWLine
  write(io_m,98) 'xo', 'yo', 'ro'
do i=1, ncom
  write(io_m,99) com(i)%x, com(i)%y, com(i)%r
end do
end if
!
if(ncam>0) then
write(io_m,'(/)')
write(io_m,971) KWAartery, ncam
write(io_m,*) KWLine
  write(io_m,98) 'xa', 'ya', 'ra'
do i=1, ncam
  write(io_m,99) cam(i)%x, cam(i)%y, cam(i)%r
end do
end if
!
if(ncvm>0) then
write(io_m,'(/)')
write(io_m,971) KWVein, ncvm
write(io_m,*) KWLine
  write(io_m,98) 'xv', 'yv', 'rv'
do i=1, ncvm
  write(io_m,99) cvm(i)%x, cvm(i)%y, cvm(i)%r
end do
end if
!
if(ncsm>0) then
write(io_m,'(/)')
write(io_m,971) KWAALmn, ncsm
write(io_m,*) KWLine
  write(io_m,98) 'xs', 'ys', 'rs'
do i=1, ncsm
  write(io_m,99) csm(i)%x, csm(i)%y, csm(i)%r

```

```

    end do
end if
!
  if(flag_tumor) then
    write(io_m,'(/A)') '[Tumor within the Lung]'
    write(io_m,'(/)')
    write(io_m,971) 'Tumor', 1
  write(io_m,*) KWLine
    write(io_m,'(4(13x,A2,11x))') 'xt', 'yt', 'zt', 'rt'
    do i=1, 1
      write(io_m,'(4g26.16)') STumor%x, STumor%y, STumor%z, STumor%r
    end do
  end if

if(bNum>0) deallocate(iBody)
if(mNum>0) deallocate(iModule)
  if(nci>0) deallocate(ci)
  if(nco>0) deallocate(co)
  if(nca>0) deallocate(ca)
  if(ncv>0) deallocate(cv)
  if(ncim>0) deallocate(cim)
  if(ncom>0) deallocate(com)
  if(ncam>0) deallocate(cam)
  if(ncvm>0) deallocate(cvm)

! deallocate(Away, Vein, AAci)

  do k=1, 3
    do j=1, 3
      do i=1, 3
        deallocate(bBod(i,j,k)%sur, bBod(i,j,k)%sid, bBod(i,j,k)%bod)
      end do
    end do
  end do

  if(flag_tumor) deallocate(BodyTumor%sur, BodyTumor%sid)

  case default
end select

! -- end
write(io_o,2009) LEND
2009 format(A8,1X,55('0'))

  close(io_o)
  close(io_v)
close(io_m)

end subroutine Output

!*****
!
! subroutine Location_2_Cutout:
!   Check if a given circle is inside the cutout region defined by
!   xcut and ycut or not.
!
! input:
!   Cir --> a circle type object
! output:
!   flag(1) --> .true. if fully inside
!   flag(2) --> .true. if fully outside
!   flag(3) --> .true. if crossing xcut(1)
!   flag(4) --> .true. if crossing xcut(2)
!   flag(5) --> .true. if crossing ycut(1)
!   flag(6) --> .true. if crossing ycut(2)
!
!*****
subroutine Location_2_Cutout(Cir, flag)

```

```

    type (Circle), intent(in) :: Cir
    logical, intent(out) :: flag(6)

    real (r8) :: root(2)
    integer (i4) :: nrt

    !!!

    flag = .false.

    if(xcut(1)<Cir%x.and.Cir%x<xcut(2).and. &
       ycut(1)<Cir%y.and.Cir%y<ycut(2)) then
        ! -- origin inside
        call get_roots(Cir,'x',xcut(1),nrt,root)      ! flag(3)
        if(nrt==2) flag(3) = .true.
        call get_roots(Cir,'x',xcut(2),nrt,root)      ! flag(4)
        if(nrt==2) flag(4) = .true.
        call get_roots(Cir,'y',ycut(1),nrt,root)      ! flag(5)
        if(nrt==2) flag(5) = .true.
        call get_roots(Cir,'y',ycut(2),nrt,root)      ! flag(6)
        if(nrt==2) flag(6) = .true.
        if(flag(3).or.flag(4).or.flag(5).or.flag(6)) then ! flag(1)
            else
                flag(1) = .true.
            end if
        else
            ! -- origin outside
            call get_roots(Cir,'x',xcut(1),nrt,root)      ! flag(3)
            if(nrt==2) then
                if((ycut(1)<root(1).and.root(1)<ycut(2)).or. &
                   (ycut(1)<root(2).and.root(2)<ycut(2))) flag(3) = .true.
            end if
            call get_roots(Cir,'x',xcut(2),nrt,root)      ! flag(4)
            if(nrt==2) then
                if((ycut(1)<root(1).and.root(1)<ycut(2)).or. &
                   (ycut(1)<root(2).and.root(2)<ycut(2))) flag(4) = .true.
            end if
            call get_roots(Cir,'y',ycut(1),nrt,root)      ! flag(5)
            if(nrt==2) then
                if((xcut(1)<root(1).and.root(1)<xcut(2)).or. &
                   (xcut(1)<root(2).and.root(2)<xcut(2))) flag(5) = .true.
            end if
            call get_roots(Cir,'y',ycut(2),nrt,root)      ! flag(6)
            if(nrt==2) then
                if((xcut(1)<root(1).and.root(1)<xcut(2)).or. &
                   (xcut(1)<root(2).and.root(2)<xcut(2))) flag(6) = .true.
            end if
            if(flag(3).or.flag(4).or.flag(5).or.flag(6)) then ! flag(2)
                else
                    flag(2) = .true.
                end if
            end if
        end if
    end subroutine Location_2_Cutout

    !*****
    !
    ! subroutine get_roots(Cir,sgn,Val,nrt,root):
    !     Get the distances to the 4 sides and 4 corners of the cutout region.
    !
    ! input:
    !     Cir --> a circle type object
    !     sgn --> 'x' or 'y'
    !     val --> value of the coordinate of sgn
    ! output:
    !     nrt    --> # of distinct roots
    !     root(2) --> 2 roots, root(1)<=root(2)

```

```

!*****
subroutine get_roots(Cir,sgn,Val,nrt,root)

  type (Circle), intent(in) :: Cir
  character (len=1), intent(in) :: sgn
  real (r8), intent(in) :: Val
  integer (i4), intent(out) :: nrt
  real (r8), intent(out) :: root(2)

  real (r8) :: dist, rtt

  !!!

  nrt = 0
  select case(sgn)
  case ('x', 'X')
    dist = abs(Cir%x - Val)
    if(Cir%r>dist) then
      nrt = 2
      rtt = sqrt(Cir%r**2 - (Val-Cir%x)**2)
      root(1) = Cir%y - rtt
      root(2) = Cir%y + rtt
    end if
  case ('y', 'Y')
    dist = abs(Cir%y - Val)
    if(Cir%r>=dist) then
      nrt = 2
      rtt = sqrt(Cir%r**2 - (Val-Cir%y)**2)
      root(1) = Cir%x - rtt
      root(2) = Cir%x + rtt
    end if
  case default
    write(*,'(A)') 'get_roots | '//sgn//' is NOT a correct axis!'
    stop
  end select

end subroutine get_roots

end module GeoGen_Output

!
! global variables
!
module GeoGen_Global

use GeoGen_Par
use GeoGen_Type

implicit none

!!!

integer (i1) :: am
character (len=64) :: suffix
integer (i4) :: len_sfx

!
! -- predefined reduced surfaces' indices
!
  type (Reduced_Surface), parameter :: &
rPln = Reduced_Surface( 0,&
'Plane',&
(/0, 0, 0, 1, 0/),&
ONE, ONE, ONE,&
ZERO, ZERO, ZERO,&
ZERO, ZERO, ZERO),&
rSph = Reduced_Surface( 0,&

```

```

'Sphere',&
(/1, 1, 1, 0, -1/),&
ONE, ONE, ONE,&
ZERO, ZERO, ZERO,&
ZERO, ZERO, ZERO),&
rCyl = Reduced_Surface( 0,&
'Cylinder',&
(/1, 1, 0, 0, -1/),&
ONE, ONE, ONE,&
ZERO, ZERO, ZERO,&
ZERO, ZERO, ZERO)

!
! -- geometry of the lung and the enclosure phantom
!
integer (i1) :: TofL ! type of lung construction
real (r8) :: LX, LY, LZ ! side length of the box lung
real (r8) :: PX, PY, PZ, PYpre, PYpos ! side length of the box phantom
type (Reduced_Surface) :: bPln(12) ! basic planes defining the lung and the phantom
type (MyBody) :: bBod(3,3,3) ! basic bodies for the whole phantom, bBod(2,2,2) is the lung
real (r8) :: LXB, LYB ! side length of the buffered box lung (for creating AVC)
real (r8) :: Lox, Loy ! origin of the cut-out region from [0:LXB,0:LYB]
logical :: flag_full_view=.false. ! output full view of the whole buffered lung
!
! -- densities
!
real (8) :: Rho(0:8), Rho4 ! Rho4 for TofL=4
data Rho/0d0, 1d0, 1.20479d-3, 0.26d0, 1d0, 1.20479d-3, 0.26d0, 0.201388d0, 0.201388d0/
!
! -- airway + vessel
!

integer (i4) :: del_NAVG_TB, NAVG
integer (i4) :: NB, NB_TB
type (Airway), allocatable :: Away(:)
type (Circle), allocatable :: Vein(:), AAci(:), StrawP(:), Straw(:)
real (r8) :: DLumen(NAVGroup), LLumen(NAVGroup)
integer (i4) :: NLumen(NAVGroup), NBundle(NAVGroup)
!
! -- tumor
!
integer (i4) :: SofT
type (Sphere) :: STumor
type (Box) :: BTumor
type (MyBody) :: BodyTumor
type (Reduced_Surface) :: SurTumor
logical :: flag_tumor
!
! -- independent "straws"
!
integer (i4) :: NStraw, NBdle

!-----!
! Data source:
! (1) Horsfield, et al. Morphology of the bronchial tree in man. Journal of !
! Applied Physiology: Vol. 24, No. 3, March 1968, 373-383. And !
! (2) Horsfield. The structure of the tracheobronchial tree. In !
! Scientific Foundations of Respiratory Medicine, edited by J.G.Scadding!
! and Gordon Cumming. 1981 !
! (3) Scientific foundations of respiratory medicine, P58 talble 3.
!----- in cm -----!

data DLumen/
10.3d-1, 8.9d-1, 7.7d-1, &
6.6d-1, 5.7d-1, 4.9d-1, 4.2d-1, 3.5d-1, &
3.3d-1, 3.1d-1, 2.9d-1, 2.8d-1, 2.6d-1, &
2.4d-1, 2.3d-1, 2.2d-1, 2.0d-1, 1.78d-1, &
1.51d-1, 1.29d-1, 1.1d-1, 0.93d-1, 0.79d-1, &

```

```

0.64d-1, 0.56d-1, 0.51d-1, 0.47d-1, 0.47d-1, &
0.49d-1, 0.50d-1, 0.49d-1, 0.51d-1, 0.40d-1, &
0.38d-1, 0.30d-1, 0.27d-1, 0.24d-1/

data LLumen/                26.0d-1, 18.0d-1, 14.0d-1, &
11.0d-1, 10.0d-1, 10.0d-1, 10.0d-1, 10.0d-1, &
9.6d-1, 9.1d-1, 8.6d-1, 8.2d-1, 7.8d-1, &
7.4d-1, 7.0d-1, 6.7d-1, 6.3d-1, 5.7d-1, &
5.0d-1, 4.4d-1, 3.9d-1, 3.5d-1, 3.1d-1, &
1.1d-1, 1.3d-1, 1.1d-1, 0.97d-1, 0.97d-1, &
0.88d-1, 0.66d-1, 0.51d-1, 0.58d-1, 0.43d-1, &
0.41d-1, 0.30d-1, 0.28d-1, 0.22d-1/

data NLumen/                2, 2, 2, &
3, 6, 8, 12, 14, &
20, 30, 37, 46, 64, &
85, 114, 158, 221, 341, &
499, 760, 1104, 1675, 2843, &
5651, 11300, 25000, 50000, 100000, &
200000, 475000, 1125000, 2700000, 6350000, &
9350000, 9150000, 3650000, 1450000/

end module GeoGen_Global

module GeoGen_Par

implicit none

! Kind parameters:
integer,parameter :: i1 = selected_int_kind( 2)      != 1-byte integer kind
integer,parameter :: i2 = selected_int_kind( 4)      != 2-byte integer kind
integer,parameter :: i4 = selected_int_kind( 9)      != 4-byte integer kind
integer,parameter :: i8 = selected_int_kind(18)      != 8-byte integer kind
integer,parameter :: r4 = selected_real_kind( 6, 37) != 4-byte real kind
integer,parameter :: r8 = selected_real_kind(15,307) != 8-byte real kind

! i/o unit numbers:
integer,parameter :: io_i = 31 != I/O unit of the problem setup(input) file.
integer,parameter :: io_o = 32 != I/O unit of the output file.
integer,parameter :: io_v = 33 != I/O unit of the output geometry grid for viewing in matlab.
integer,parameter :: io_m = 34 != I/O unit of the output circles for setting up mass grid.

!
real (r8), parameter :: ONE = 1d0
real (r8), parameter :: ZERO = 0d0
real (r8), parameter :: HUGE = 1.0d+36             != A very large number.
real (r8), parameter :: PI = 3.141592653589793238462643383279502884197d+0

! keywords used by PENELOPE geometric file
character (len=8), parameter :: &
LNUL = '00000000', &

LSUR = 'SURFACE ', &
LIND = 'INDICES=', &
LBOD = 'BODY ', &
LMAT = 'MATERIAL', &
LMOD = 'MODULE ', &
LOPEN = ') ', &
LEND = 'END ', &
LONE = '11111111', &
LXSC = 'X-SCALE=', &
LYSC = 'Y-SCALE=', &
LZSC = 'Z-SCALE=', &
LTHE = ' THETA=', &
LPHI = ' PHI=', &
LOME = ' OMEGA=', &
LXSH = 'X-SHIFT=', &
LYSH = 'Y-SHIFT=', &

```



```

LZSH = 'Z-SHIFT=', &
LAXX = '  AXX=', &
LAXY = '  AXZ=', &
LAXZ = '  AYZ=', &
LAYY = '  AYY=', &
LAYZ = '  AYZ=', &
LAZZ = '  AZZ=', &
LAX  = '  AX=', &
LAY  = '  AY=', &
LAZ  = '  AZ=', &
LAO  = '  AO=', &
LDEG = ') DEG ', &
LRAD = ') RAD ', &
LINC = 'INCLUDE ', &
LFIL='FILE= '

integer (i1), parameter :: void=0, &
                        water=1, water_max=4,&
                        air=2, air_max=5, &
                        aml=3, aml_max=6, &
                        amlp=7, amlp_max=8

! default lung and phantom sizes
real (r8), parameter :: LXdft = 10.2d0, &
                        LYdft = 10.0d0, &
                        LZdft = 10.2d0 ! side length of the box lung
real (r8) :: PXdft = 30.0d0, &
            PZdft = 30.0d0, &
            PYpredft = 5.0d0, &
            PYposdft = 5.0d0 ! side length of the box phantom

! lung airway/artery/vein parameters:
integer, parameter :: NAVGroup_TB=26, NAVGroup=NAVGroup_TB + 11;
real (r8), parameter :: VLung=6000 ! 6000 mL for both lungs
real (r8), parameter :: pVssl=1.43884892086331d0 ! Bronchoarterial Ratio = 0.695 [1]
real (r8), parameter :: PShell=0.666666666666667d0 ! T/D = 0.2 [1]
integer, parameter :: CVn2Bdl=6 ! # of top generations with vein correlated with
real (r8), parameter :: pVn2Bdl=1d0 ! correlation range as the ratio of the bundle radius
character (len=6), parameter :: &
                        KWLumen = 'Lumen ', &
                        KWWall = 'Wall ', &
                        KWArtery = 'Artery', &
                        KWVein = 'Vein ', &
                        KWAALmn = 'AALumn'
character (len=25), parameter :: KWLine = '-- must have this line --'
end module GeoGen_Par

!-----
! references:
! [1] S. Matsuoka, et al. Bronchoarterial ratio and bronchial wall thickness
! on high resolution CT in asymptomatic subjects: correlation with age and smoking.
! AJR 2003;180: 513-518
!-----

!
! type definitions
!
module GeoGen_Type

use GeoGen_Par

type :: Reduced_Surface
integer (i4) :: num
character (len=64) :: typ
integer (i1) :: idx(5)
real (r8) :: xscl, yscl, zscl,&
            omg, tht, phi,&
            xshft, yshft, zshft

```

```

end type reduced_surface

type :: MyBody
integer (i4) :: num
character (len=64) :: id
integer (i1) :: mat
integer (i4), allocatable :: sur(:)
integer (i1), allocatable :: sid(:)
integer (i4), allocatable :: bod(:)
end type MyBody

type :: Circle
real (r8) :: x
real (r8) :: y
real (r8) :: r
end type Circle

type :: Sphere
real (r8) :: x, &
           y, &
           z, &
           r
end type Sphere

type :: Box
real (r8) :: x, &
           y, &
           z
real (r8) :: dx, &
           dy, &
           dz
end type Box

type :: Airway
type (Circle) :: ci, &
               co, &
               ca, &
               cbd
end type Airway

end module GeoGen_Type

```

#### A.4 Fortran 90 codes to generate the mass for the dose grid used for the lung phantom the $2\frac{1}{2}$ -D random lung model in Chapter V

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!      |
!      |
!      |
!      |-----> y
!      |(0,0)
!      |
!      |
!      V
!      x
!!!!!!!!!!!!!!
!a      d!
!      !
!b      c!
!!!!!!!!!!!!!!

program main

```

```

use NPDM_Get_Partial_Mass
use NPDM_GetIndex
use NPDM_SetGrid

implicit none

real (r8) :: x, y, z, dz
real (r8) :: ntmp, dxext, dzext
real (r8) :: PA
real (r8) :: rw
real (r8) :: AFy, ADy ! cross section perpendicular to y-axis
integer (i4) :: MBndX(2), MBndY(2) ! mass bounds for caculating partial mass contribution
integer (i4) :: i, j, k, ii, jj, kk, il, iw, ia, iv, iaa, im, jm, km
integer (i4) :: ic(2), kc(2)
integer (i4) :: lorlp
integer (i4) :: amaw
integer (i4) :: len_fmlass
character (len=6) :: KWWord
character (len=128) :: buffer

real (r8) :: DxLim1, DxLim2, DyLim1, DyLim2, DzLim1, DzLim2
type (Box) :: btmp, btmp2
integer (i4) :: region
real (r8) :: SubV
real (r8) :: RhoTmp
real (r8) :: msum
logical :: exists, exists2
integer (i4) :: ismat
type (Index) :: ihead, itail
character (i1) :: yesno

!!!

do
! -- read in dose grid info
write(*, '(a)') 'This is an auxiliary program to generate a mass file for'
write(*, '(a)') 'the PENELOPE used to simulate the lung model.'

write(*, '(a)') '*Note* -- prepare first the dose grid file from the dose grid section of '
write(*, '(a)') ' the corresponding PENELOPE input file. Usually, Copy & Past'
write(*, '(a)') ' would do the job.'
write(*, '(a$)') 'dose grid file name --> '
read(*, *) FDGrid
! -- inquire about file's existence:
inquire(file=FDGrid, exist=exists)
if(exists) then
else
write(*, '(2a)') ' >> Cannot find file ', FDGrid
end if
end do

! -- prepare the output mass file
do
write(*, '(a$)') 'Output mass file name --> '
read(*, *) FMass
! -- inquire about file's existence:
inquire(file=FMass, exist=exists)
if(exists) then
write(*, '(2x,2a)') 'File '//trim(FMass)//' already exists'
write(*, '(2x,a$)') 'Replace it? (y/n) '
read(*, '(a)') yesno
select case(yesno)
case ('y', 'Y')
open(io_m, FILE=FMass, action='WRITE', status='OLD')
write(*, '(2x,2a)') 'Replaced file ', trim(FMass)
exit

```

```

        case default
        end select
    else
    open(io_m,FILE=FMass,action='WRITE',status='NEW')
        write(*,'(2a)') 'Created file ', trim(FMass)
        exit
    end if
end do

open(unit=io_d, file=FDGrid, action='READ', status='OLD')

NDxRgn = 0; NDyRgn = 0; NDzRgn = 0
NxDT = 0; NyDT = 0; NzDT = 0

do while(.not.eof(io_d))
    read(io_d,30) KWord,buffer
    select case(KWord)
    case (KWDXDO)
        call ReadGrid('x',buffer,NxD,NDxRgn,NxD,DT,DBnDX,dxD)
    case (KWDYDO)
        call ReadGrid('y',buffer,NyD,NDyRgn,NyDT,DT,DBnDY,dyD)
    case (KWDZDO)
        call ReadGrid('z',buffer,NzD,NDzRgn,NzDT,DT,DBnDZ,dzD)
    case default
    end select
end do
close(io_d)

allocate(XDose(NxDT+1), YDose(NyDT+1), ZDose(NzDT+1))
call SetGrid('x',NxD,NDxRgn,NxD,DT,DBnDX,dxD,XDose)
call SetGrid('y',NyD,NDyRgn,NyDT,DT,DBnDY,dyD,YDose)
call SetGrid('z',NzD,NDzRgn,NzDT,DT,DBnDZ,dzD,ZDose)

write(*,'(/a)') 'Current dose grid boundaries:'
write(*,10) 'X1 = ', DBnDX(1), 'X2 = ', DBnDX(NDxRgn+1)
write(*,10) 'Y1 = ', DBnDY(1), 'Y2 = ', DBnDY(NDyRgn+1)
write(*,10) 'Z1 = ', DBnDZ(1), 'Z2 = ', DBnDZ(NDzRgn+1)

10 format(2x,2(a5,g13.5))

    amaw = 0
    do while(amaw<1.or.amaw>3)
        write(*,'(/a)') '(1) Atomic Mix;'
        write(*,'(a)') '(2) Airway in parenchyma;'
        write(*,'(a)') '(3) Airway in air;'
        write(*,'(a$)') '--> '
        read(*,*) amaw
        if(amaw<1.or.amaw>3) write(*,'(a)') '>>> No such choice. Choose it again.'
    end do

    if(amaw/=1) then
        do
            ! -- read in dose grid info
            write(*,'(a$)') 'Mass-geo file name --> '
            read(*,*) FMassGeo
            ! -- inquire about file's existence:
            inquire(file=FMassGeo, exist=exists)
            if(exists) then
                exit
            else
                write(*,'(2a)') ' >> Cannot find file ', FMassGeo
            end if
        end do
    end if

    if(amaw==1) then
        lorlp = 0
        do while(lorlp<1.or.lorlp>4)

```

```

write(*,'(/2x,a)') '(1) Lung          (0.26      g/cm^3);'
write(*,'(2x,a)') '(2) Lung Parenchyma (0.201388  g/cm^3);'
write(*,'(2x,a)') '(3) Air           (1.20479e-3 g/cm^3);'
write(*,'(2x,a)') '(4) Other         (your input g/cm^3);'
write(*,'(2x,a$)') '--> '
read(*,*) lorlp
select case(lorlp)
case (1,2,3,4)
case default
write(*,*) 'no such choice'
end select
end do
end if

write(*,'(/a)') 'Current lung boundries:'
!
write(*,10) 'X1 = ', LBndX(1), 'X2 = ', LBndX(2)
write(*,20) 'Want to change? (y/n) '
yesno = 'n'
read(*,*) yesno
select case(yesno)
case ('y', 'Y')
write(*,20) 'new X1, X2? '
read(*,*) LBndX(1), LBndX(2)
case default
end select
!
write(*,10) 'Y1 = ', LBndY(1), 'Y2 = ', LBndY(2)
write(*,20) 'Want to change? (y/n) '
yesno = 'n'
read(*,*) yesno
select case(yesno)
case ('y', 'Y')
write(*,20) 'new Y1, Y2? '
read(*,*) LBndY(1), LBndY(2)
case default
end select
!
write(*,10) 'Z1 = ', LBndZ(1), 'Z2 = ', LBndZ(2)
write(*,20) 'Want to change? (y/n) '
yesno = 'n'
read(*,*) yesno
select case(yesno)
case ('y', 'Y')
write(*,20) 'new Z1, Z2? '
read(*,*) LBndZ(1), LBndZ(2)
case default
end select

11 format(2x,3(a9,g11.3))
20 format(4x,a,$)

! -- prescribed mass (atomic mix case) for the whole dose grid
allocate(Mass(NxDT,NyDT,NzDT))

if(amaw==1) then
if(lorlp==1) then
RhoTmp = rho(aml)
else if(lorlp==2) then
RhoTmp = rho(amlp)
else if(lorlp==3) then
RhoTmp = rho(air)
else if(lorlp==4) then
write(*,'(/A$)') 'The density for the lung part --> '
read(*,*) RhoTmp
end if
end if

```

```

else if(amaw==2) then
  write(*,'(2x, a,f11.8,a)') &
    'Current density of parenchyma: ',rho(amlp), 'g/cm^3, want to change? (y/n) '
  read(*,*) yesno
  select case(yesno)
  case ('y', 'Y')
    write(*,'(4x,a$)') 'New density (g/cm^3) --> '
    read(*,*) rho(amlp)
  case default
    end select
  RhoTmp = rho(amlp)
else if(amaw==3) then
  RhoTmp = rho(air)
end if

! Fill the phantom with water outside and atomic mix inside the lung
do k=1, NzDT
  do j=1, NyDT
    do i=1, NxDT
      btmp%head%x = XDose(i); btmp%head%y = YDose(j); btmp%head%z = ZDose(k);
      btmp%tail%x = XDose(i+1); btmp%tail%y = YDose(j+1); btmp%tail%z = ZDose(k+1);
      btmp2%head%x = LBndX(1); btmp2%head%y = LBndY(1); btmp2%head%z = LBndZ(1);
      btmp2%tail%x = LBndX(2); btmp2%tail%y = LBndY(2); btmp2%tail%z = LBndZ(2);
      call FracVol(btmp, btmp2, SubV) ! outer water or the lung
      Mass(i,j,k) = (VBox(btmp) - SubV)*rho(water) + SubV*RhoTmp
    end do
  end do
end do

NDLx(1) = getidx(LBndX(1)+TINY,XDose)
NDLx(2) = getidx(LBndX(2)-TINY,XDose)
NDLy(1) = getidx(LBndY(1)+TINY,YDose)
NDLy(2) = getidx(LBndY(2)-TINY,YDose)
NDLz(1) = getidx(LBndZ(1)+TINY,ZDose)
NDLz(2) = getidx(LBndZ(2)-TINY,ZDose)

if(amaw==1) go to 3 ! if homogenized, detour!

! -- get all the airway/vein bundles' locations
open(unit=io_mg, file=FMassGeo, action='READ', status='OLD')
do while(.not.eof(io_mg))
  read(io_mg,30) KWWord,buffer
  select case(KWWord)
  case (KWLumen)
    read(buffer,*) NLumen
    allocate(Lumen(NLumen))
    read(io_mg,'(a)') buffer
    read(io_mg,'(a)') buffer
    do i=1, NLumen
      read(io_mg,*) Lumen(i)%x, Lumen(i)%y, Lumen(i)%r
    end do
  case (KWWall)
    read(buffer,*) NWall
    allocate(Wall(NWall))
    read(io_mg,'(a)') buffer
    read(io_mg,'(a)') buffer
    do i=1, NWall
      read(io_mg,*) Wall(i)%x, Wall(i)%y, Wall(i)%r
    end do
  case (KWArtery)
    read(buffer,*) NArtery
    allocate(Artery(NArtery))
    read(io_mg,'(a)') buffer
    read(io_mg,'(a)') buffer
    do i=1, NArtery
      read(io_mg,*) Artery(i)%x, Artery(i)%y, Artery(i)%r
    end do
  case (KWVein)

```

```

read(buffer,*) NVein
allocate(Vein(NVein))
read(io_mg,'(a)') buffer
read(io_mg,'(a)') buffer
do i=1, NVein
  read(io_mg,*) Vein(i)%x, Vein(i)%y, Vein(i)%r
end do
case default
end select
end do
close(io_mg)
30 format(a6,1x,a128)

! -- the correct density to replace
select case(amaw)
case (2)
  RhoTmp = Rho(amlp)
case (3)
  RhoTmp = Rho(air)
case default
end select

! -- artery
if(NArtery>0) then
a: do ia=1, NArtery
  ! -- get the enclosure boundary MBndX and MBndY
  call GetMassBoundary(Artery(ia), MBndX, MBndY)
  MBndX(1) = max(MBndX(1),NDLx(1))
  MBndX(2) = min(MBndX(2),NDLx(2))
  MBndY(1) = max(MBndY(1),NDLy(1))
  MBndY(2) = min(MBndY(2),NDLy(2))
  ! -- get the partial mass for each swept voxel
  do im=MBndX(1), MBndX(2)
    do jm=MBndY(1), MBndY(2)
      call Get_abcd(im,jm)
      PA = Partial_Area(Artery(ia), xyVox)
      do k=NDLz(1), NDLz(2)
        dz = min(ZDose(k+1), LBndZ(2)) - max(ZDose(k),LBndZ(1))
        Mass(im,jm,k) = Mass(im,jm,k) + dz*PA*(Rho(water)-RhoTmp)
      end do
    end do
  end do
end do a
end if

! -- vein
if(NVein>0) then
v: do iv=1, NVein
  ! -- get the enclosure boundary MBndX and MBndY
  call GetMassBoundary(Vein(iv), MBndX, MBndY)
  MBndX(1) = max(MBndX(1),NDLx(1))
  MBndX(2) = min(MBndX(2),NDLx(2))
  MBndY(1) = max(MBndY(1),NDLy(1))
  MBndY(2) = min(MBndY(2),NDLy(2))
  ! -- get the partial mass for each swept voxel
  do im=MBndX(1), MBndX(2)
    do jm=MBndY(1), MBndY(2)
      call Get_abcd(im,jm)
      PA = Partial_Area(Vein(iv), xyVox)
      do k=NDLz(1), NDLz(2)
        dz = min(ZDose(k+1), LBndZ(2)) - max(ZDose(k),LBndZ(1))
        Mass(im,jm,k) = Mass(im,jm,k) + dz*PA*(Rho(water)-RhoTmp)
      end do
    end do
  end do
end do v
end if

```

```

! -- Wall
if(NWall>0) then
w: do iw=1, NWall
! -- get the enclosure boundary MBndX and MBndY
call GetMassBoundary(Wall(iw), MBndX, MBndY)
MBndX(1) = max(MBndX(1),NDLx(1))
MBndX(2) = min(MBndX(2),NDLx(2))
MBndY(1) = max(MBndY(1),NDLy(1))
MBndY(2) = min(MBndY(2),NDLy(2))
! -- get the partial mass for each swept voxel
do im=MBndX(1), MBndX(2)
do jm=MBndY(1), MBndY(2)
call Get_abcd(im,jm)
PA = Partial_Area(Wall(iw), xyVox)
do k=NDLz(1), NDLz(2)
dz = min(ZDose(k+1), LBndZ(2)) - max(ZDose(k),LBndZ(1))
Mass(im,jm,k) = Mass(im,jm,k) + dz*PA*(Rho(water)-RhoTmp)
end do
end do
end do
end do w
end if

! -- Lumen
if(NLumen>0) then
l: do il=1, NLumen
! -- get the enclosure boundary MBndX and MBndY
call GetMassBoundary(Lumen(il), MBndX, MBndY)
MBndX(1) = max(MBndX(1),NDLx(1))
MBndX(2) = min(MBndX(2),NDLx(2))
MBndY(1) = max(MBndY(1),NDLy(1))
MBndY(2) = min(MBndY(2),NDLy(2))
! -- get the partial mass for each swept voxel
do im=MBndX(1), MBndX(2)
do jm=MBndY(1), MBndY(2)
call Get_abcd(im,jm)
PA = Partial_Area(Lumen(il), xyVox)
do k=NDLz(1), NDLz(2)
dz = min(ZDose(k+1), LBndZ(2)) - max(ZDose(k),LBndZ(1))
Mass(im,jm,k) = Mass(im,jm,k) + dz*PA*(Rho(air)-Rho(water))
end do
end do
end do
end do l
end if

3 continue

! -- output mass file
write(io_m,'(3I8)') NxDT, NyDT, NzDT
write(io_m,'(A,I8)') 'Num_Dose_Grid_X=', NxDT
write(io_m,'(A,I8)') 'Num_Dose_Grid_Y=', NyDT
write(io_m,'(A,I8)') 'Num_Dose_Grid_Z=', NzDT
write(io_m,'(/,A)') 'XDose(...)'
write(io_m,'(1000f8.3)') (XDose(i), i=1,NxDT+1)
write(io_m,'(/,A)') 'YDose(...)'
write(io_m,'(1000f8.3)') (YDose(i), i=1,NyDT+1)
write(io_m,'(/,A)') 'ZDose(...)'
write(io_m,'(1000f8.3)') (ZDose(i), i=1,NzDT+1)
write(io_m,'(/a)') 'Mass...'
write(io_m,*) ((( Mass(i,j,k),i=1,NxDT),j=1,NyDT),k=1,NzDT)
write(io_m,'(/,a,g25.16)') 'Total mass =', sum(Mass)

close(io_m)
write(*,*) ' end'

end program main

```



```

Module NPDM_GetIndex

use NPDM_Globals

implicit none

contains

!-----
! function GetIdx(x,XA)
!   given a value x, give the region index in the bound array XA
! comments:
!       for x < XA(1),      ix = 1
!       x > XA(size(XA)), ix = size(XA) - 1
!       so x always in XA in output, which is only true for here
!-----
integer (i4) function GetIdx(x,XA) result(ix)

  real (r8), intent(in) :: x, XA(:)

  integer (i4) :: n1, n2, nm

  !!!

  n1 = 1                ! first region
  n2 = size(XA) - 1    ! last region
  do while(n1/=n2)
    nm = (n1 + n2)/2   ! a middle region
    if(x<=XA(nm+1)) then
      n2 = nm
    else
      n1 = nm + 1
    end if
  end do
  ix = n1

end function GetIdx

!-----
! subroutine GetMassBoundary(Cir, BX, BY)
!   given a circle, give the dose(mass) grid bounds enclosing it.
!-----
subroutine GetMassBoundary(Cir, Bx, By)

  type (Circle), intent(in) :: Cir
  integer (i4), intent(inout) :: Bx(2), By(2)

  !!!

  Bx(1) = GetIdx(Cir%x - Cir%r, XDose)
  Bx(2) = GetIdx(Cir%x + Cir%r, XDose)
  By(1) = GetIdx(Cir%y - Cir%r, YDose)
  By(2) = GetIdx(Cir%y + Cir%r, YDose)

end subroutine GetMassBoundary

!-----
! subroutine GetVoxBoundary(bx)
!   given a circle, give the dose(mass) grid bounds enclosing it.
!-----
subroutine GetVoxBoundary(bx, ihead, itail)

  type (Box), intent(in) :: bx
  type (Index), intent(inout) :: ihead, itail

```

```

!!!

ihead%i = GetIdx(bx%head%x, XDose)
ihead%j = GetIdx(bx%head%y, YDose)
ihead%k = GetIdx(bx%head%z, ZDose)

itail%i = GetIdx(bx%tail%x, XDose)
itail%j = GetIdx(bx%tail%y, YDose)
itail%k = GetIdx(bx%tail%z, ZDose)

end subroutine GetVoxBoundary

end module NPDM_GetIndex

module NPDM_Globals

use NPDM_Types

integer (i1) :: void=0,&
                water=1, water_max=4,&
                air=2, air_max=5,&
                aml=3, aml_max=6,&
                amlp=7, amlp_max=8
real (r8) :: rho(0:8)
data Rho/0.0d0, &
    1.0d0, 1.20479d-3, 0.26d0, &
    1.0d0, 1.20479d-3, 0.26d0, &
    0.201388d0, 0.201388d0/

real (r8) :: PhBndX(2), PhBndY(2), PhBndZ(2) ! fantom boundaries
data PhBndX/-15.0d0, 15.0d0/, PhBndY/-5.0d0, 15.0d0/, PhBndZ/-15.0d0, 15.0d0/

real (r8) :: LBndX(2), LBndY(2), LBndZ(2) ! lung boundaries
data LBndX/-5.1d0, 5.1d0/, LBndY/0d0, 1.0d1/, LBndZ/-5.1d0, 5.1d0/

real (r8) :: DBndX(NDRgnM+1), DBndY(NDRgnM+1), DBndZ(NDRgnM+1) , &
    dxD(NDRgnM), dyD(NDRgnM), dzD(NDRgnM)

real (r8), allocatable :: XDose(:), YDose(:), ZDose(:), &
    XDose2(:), YDose2(:), ZDose2(:)

integer (i4) :: NxD(NDRgnM), NyD(NDRgnM), NzD(NDRgnM), &
    NDxRgn, NDyRgn, NDzRgn, &
    NxDT, NyDT, NzDT, &
    NxDT2, NyDT2, NzDT2

integer :: NDLx(2), NDly(2), NDLz(2), iDLy(2)
integer :: NLumen=0, NWall=0, NArtery=0, NVein=0, NAALm=0

type (Circle), allocatable :: Lumen(:), Wall(:), Artery(:), Vein(:)
type (Rectangle) :: xyVox

real (r8), allocatable :: Mass(:, :, :), Mass2(:, :, :)

character (len=64) :: FDGrid, FDGrid2, FMass, FMassGeo

end module NPDM_Globals

module NPDM_Parameters

! Kind parameters:
integer,parameter :: i1 = selected_int_kind( 2)      != 1-byte integer kind
integer,parameter :: i2 = selected_int_kind( 4)      != 2-byte integer kind
integer,parameter :: i4 = selected_int_kind( 9)      != 4-byte integer kind
integer,parameter :: i8 = selected_int_kind(18)      != 8-byte integer kind
integer,parameter :: r4 = selected_real_kind( 6, 37) != 4-byte real kind
integer,parameter :: r8 = selected_real_kind(15,307) != 8-byte real kind

```

```

! i/o unit numbers:
integer,parameter :: io_d  = 33  != I/O unit of the input dose grid info.
integer,parameter :: io_mg = 34  != I/O unit of the output circles for setting up mass grid.
integer,parameter :: io_m  = 35  != I/O unit of the output mass file.

!
real (r8), parameter :: ONE = 1d0
real (r8), parameter :: ZERO = 0d0
real (r8), parameter :: TINY = 1.0d-10      ! A very small number
real (r8), parameter :: HUGE = 1.0d+36      ! A very large number.
real (r8), parameter :: PI = 3.141592653589793238462643383279502884197d+0

character (len=6), parameter :: &
      KWLumen = 'Lumen ', &
      KWWall  = 'Wall ', &
      KWArtery = 'Artery', &
      KWVein  = 'Vein ', &
      KWAALm  = 'AALumn', &
      KWNull  = ' ', &
      KWDXDO  = 'DXDOSE', &
      KWDYDO  = 'DYDOSE', &
      KWDZDO  = 'DZDOSE'

integer (i4), parameter :: NDRgnM=20, &
      NDXM=10000, NDYM=10000, NDZM=10000
integer (i4), parameter :: NRegion=2 ! 2 main regions (surrounding water and lung)

end module NPDM_Parameters

Module NPDM_Get_Partial_Mass

use NPDM_Globals

implicit none

character (len=2) :: cname(4), sname(4)
data cname/'a ', 'b ', 'c ', 'd '/, &
      sname/'ab', 'bc', 'cd', 'da'/

contains

real (r8) function Partial_Area(Cir0, Vox) result(PArea)
!*****
! get the partial area that's the interception of the circle and the
! rectangle.
!*****

type (Circle), intent(in) :: Cir0
type (Rectangle), intent(in) :: Vox

real (r8) :: ACir, AVox
type (Point) :: VC
type (Index) :: Idx
type (Circle) :: Cir, Cir22
type (Circle2Voxel) :: CV
type (Point) :: T1, T2, T3, T4, T5, T6, T7, T8
integer :: iab, ibc, icd, ida

!!!

ACir = PI*Cir0%r**2
AVox = abs((Vox%a%x - Vox%c%x)*(Vox%a%y - Vox%c%y))
VC%x = (Vox%a%x + Vox%c%x)/2
VC%y = (Vox%a%y + Vox%c%y)/2
VC%z = (Vox%a%z + Vox%c%z)/2

! -- get the relative location of the origin of the circle to this voxel
Idx = Get_Index(Cir0,Vox)

```

```

! -- take advantage of the symmetry and do reflection accordingly
!
!
!           ----->y
!           *****
!           *           *           *
!           \           *           *
!           **> (1,1) | (1,2) <** | (1,3) *
!           *   -----|-----*----- *
!           *   *****|*****|***** *
!           *   *       |       * | *   *
!           *   \       |       * | *   *
!           *   (2,1) | (2,2) * | (2,3) *
!           *   -----|-----*----- *
!           *** (3,1) | (3,2) *** | (3,3) ***
!
!           \
!           x
!*****
  Cir = Cir0
  select case(Idx%i)
  case (1)
select case(Idx%j)
case (1,2)
case (3) ! -- (1,3) --> (1,1)
  Cir%x = Cir0%x
  Cir%y = 2*VC%y - Cir0%y
  Idx = Index(1,1,0)
case default
  stop 'Partial_Mass|wrong index in the 1st row.'
end select
  case (2)
select case(Idx%j)
case (1,2)
case (3) ! -- (2,3) --> (2,1)
  Cir%x = Cir0%x
  Cir%y = 2*VC%y - Cir0%y
  Idx = Index(2,1,0)
case default
  stop 'Partial_Mass|wrong index in the 2nd row.'
end select
  case (3)
select case(Idx%j)
case (1) ! -- (3,1) --> (1,1)
  Cir%x = 2*VC%x - Cir0%x
  Cir%y = Cir0%y
  Idx = Index(1,1,0)
  case (2) ! -- (3,2) --> (1,2)
  Cir%x = 2*VC%x - Cir0%x
  Cir%y = Cir0%y
  Idx = Index(1,2,0)
  case (3) ! -- (3,3) --> (1,1)
  Cir%x = 2*VC%x - Cir0%x
  Cir%y = 2*VC%y - Cir0%y
  Idx = Index(1,1,0)
case default
  stop 'Partial_Mass|wrong index in the 3rd row.'
end select
  case default
  stop 'Partial_Mass|No such Index'
end select

! -- get the distances to the 4 corners and 4 sides
  CV = C2V(Cir,Vox)

regn: select case(idx%i)
! (1,:)
  case (1)
row1:  select case(idx%j)

```

```

! - (1,1)
case (1)
  if(Cir%r <= CV%dcnr(1)) then ! all outside the circle
  PArea = 0d0
    else if(Cir%r < CV%dcnr(2)) then ! the closest corner inside
  T1 = LineCir(Cir,Vox,'ab','x+')
  T2 = LineCir(Cir,Vox,'da','y+')
  T3 = Vox%a
  T4 = T2
  PArea = PArea1(Cir,T1,T2,T3,T4,'x')
    else if(Cir%r < CV%dcnr(3)) then ! 2 corners inside
  select case(CV%cnr(2))
  case ('b ') ! corner b inside
    T1 = LineCir(Cir,Vox,'bc','y+')
    T2 = LineCir(Cir,Vox,'da','y+')
    T3 = Vox%b
    T4 = Vox%a
    PArea = PArea1(Cir,T1,T2,T3,T4,'y')
  case ('d ') ! corner d inside
    T1 = LineCir(Cir,Vox,'ab','x+')
    T2 = LineCir(Cir,Vox,'cd','x+')
    T3 = Vox%a
    T4 = Vox%d
    PArea = PArea1(Cir,T1,T2,T3,T4,'x')
  case default
    write(*,*) 'PDM_Partial_Mass|(1,1)|2 corners|wrong corner: '//CV%cnr(2)
    stop
  end select
    else if(Cir%r < CV%dcnr(4)) then ! 3 corners inside
  T1 = LineCir(Cir,Vox,'bc','y+')
  T2 = LineCir(Cir,Vox,'cd','x+')
  T4 = Vox%d
  T3%x = T4%x
  T3%y = T1%y
  PArea = PArea1(Cir,T1,T2,T3,T4,'x')
  PArea = PArea + abs((T3%x-Vox%b/x)*(T3%y-Vox%b/y))
    else ! all inside the circle
  PArea = AVox
    end if
! - (1,2)
case (2)
  ida = get_side_i(CV,'da')
  if(Cir%r <= CV%dsid(ida)) then ! all outside
  PArea = 0d0
    else if(Cir%r <= CV%dcnr(1)) then ! some but no corners inside
  T1 = LineCir(Cir,Vox,'da','y-')
  T2 = LineCir(Cir,Vox,'da','y+')
  PArea = Moon(Cir,T1,T2)
  ibc = Get_Side_i(CV,'bc')
  if(Cir%r > CV%dsid(ibc)) then ! beyond the far side
    T3 = LineCir(Cir,Vox,'bc','y-')
    T4 = LineCir(Cir,Vox,'bc','y+')
    PArea = PArea - Moon(Cir,T3,T4)
  end if
    else if(Cir%r < CV%dcnr(2)) then ! one corner inside
  ibc = Get_Side_i(CV,'bc')
  select case(CV%cnr(1))
  case ('a ')
    if(Cir%r > CV%dsid(ibc)) then ! beyond the far side
    T2 = LineCir(Cir,Vox,'da','y+')
    T3 = LineCir(Cir,Vox,'ab','x+')
    T4 = Vox%a
    T1%x = T3%x
    T1%y = TCross(T1%x,'y+',Cir)
    T5 = LineCir(Cir,Vox,'bc','y-')
    T6 = LineCir(Cir,Vox,'bc','y+')
    PArea = PArea1(Cir,T1,T2,T3,T4,'y')
    PArea = PArea + Moon(Cir,T1,T3) - Moon(Cir,T5,T6)

```

```

else
T1 = LineCir(Cir,Vox,'ab','x+')
T2 = LineCir(Cir,Vox,'da','y+')
T3 = Vox%a
T4 = T2
PArea = PArea1(Cir,T1,T2,T3,T4,'x')
end if
case ('d ')
if(Cir%r > CV%dsid(IBC)) then ! beyond the far side
T1 = LineCir(Cir,Vox,'da','y-')
T3 = Vox%d
T4 = LineCir(Cir,Vox,'cd','x+')
T2%x = T4%x
T2%y = TCross(T2%x,'y-',Cir)
T5 = LineCir(Cir,Vox,'bc','y-')
T6 = LineCir(Cir,Vox,'bc','y+')
PArea = PArea1(Cir,T1,T2,T3,T4,'y')
PArea = PArea + Moon(Cir,T2,T4) - Moon(Cir,T5,T6)
else
T1 = LineCir(Cir,Vox,'da','y-')
T2 = LineCir(Cir,Vox,'cd','x+')
T3 = T1
T4 = Vox%d
PArea = PArea1(Cir,T1,T2,T3,T4,'x')
end if
case default
write(*,*) 'PDM_Partial_Mass|(1,2)|1 corner|wrong corner: '//CV%cnr(1)
stop
end select
else if(Cir%r < CV%dcnr(3)) then ! two corners inside
select case(CV%cnr(1)//CV%cnr(2))
case ('a d ', 'd a ')
T1 = LineCir(Cir,Vox,'ab','x+')
T2 = LineCir(Cir,Vox,'cd','x+')
T3 = Vox%a
T4 = Vox%d
PArea = PArea1(Cir,T1,T2,T3,T4,'x')
IBC = Get_Side_i(CV,'bc')
if(Cir%r > CV%dsid(IBC)) then ! beyond the far side
T5 = LineCir(Cir,Vox,'bc','y-')
T6 = LineCir(Cir,Vox,'bc','y+')
PArea = PArea - Moon(Cir,T5,T6)
end if
case ('a b ')
T1 = LineCir(Cir,Vox,'bc','y+')
T2 = LineCir(Cir,Vox,'da','y+')
T3 = Vox%b
T4 = Vox%a
PArea = PArea1(Cir,T1,T2,T3,T4,'y')
case ('c d ')
T1 = LineCir(Cir,Vox,'da','y-')
T2 = LineCir(Cir,Vox,'bc','y-')
T3 = Vox%d
T4 = Vox%c
PArea = PArea1(Cir,T1,T2,T3,T4,'y')
case default
write(*,*) 'PDM_Partial_Mass|(1,2)|2 corners|wrong combination: '//CV%cnr(1)//CV%cnr(2)
stop
end select
else if(Cir%r < CV%dcnr(4)) then ! three corners inside
select case(CV%cnr(4))
case ('b ')
T1 = LineCir(Cir,Vox,'ab','x+')
T2 = LineCir(Cir,Vox,'bc','y-')
T3 = Vox%a
T4%x = T3%x
T4%y = T2%y
PArea = PArea1(Cir,T1,T2,T3,T4,'x')

```

```

PArea = PArea + abs((T4%x-Vox%c%x)*(T4%y-Vox%c%y))
case ('c ')
  T1 = LineCir(Cir,Vox,'bc','y+')
  T2 = LineCir(Cir,Vox,'cd','x+')
  T4 = Vox%d
  T3%x = T4%x
  T3%y = T1%y
  PArea = PArea1(Cir,T1,T2,T3,T4,'x')
  PArea = PArea + abs((T3%x-Vox%b%x)*(T3%y-Vox%b%y))
case default
  write(*,*) 'PDM_Partial_Mass|(1,2)|3 corners|no such corner: '//CV%cnr(4)
  stop
end select
else ! all inside
PArea = AVox
end if
case default
  stop 'Partial_Mass|Wrong mapped index into 1st row'
end select row1
! (2,:)
  case (2)
row2:  select case(idx%j)
! - (2,1)
case (1)
  iab = get_side_i(CV,'ab')
  if(Cir%r <= CV%dsid(iab)) then ! all outside
PArea = 0d0
  else if(Cir%r <= CV%dcnr(1)) then ! some but no corners inside
  T1 = LineCir(Cir,Vox,'ab','x+')
  T2 = LineCir(Cir,Vox,'ab','x-')
  PArea = Moon(Cir,T1,T2)
  icd = Get_Side_i(CV,'cd')
  if(Cir%r > CV%dsid(icd)) then ! beyond the far side
    T3 = LineCir(Cir,Vox,'cd','x+')
    T4 = LineCir(Cir,Vox,'cd','x-')
    PArea = PArea - Moon(Cir,T3,T4)
  end if
  else if(Cir%r < CV%dcnr(2)) then ! one corner inside
select case(CV%cnr(1))
case ('a ')
  T1 = LineCir(Cir,Vox,'ab','x+')
  T2 = LineCir(Cir,Vox,'da','y+')
  T3 = Vox%a
  T4 = T2
  PArea = PArea1(Cir,T1,T2,T3,T4,'x')
case ('b ')
  T1 = LineCir(Cir,Vox,'bc','y+')
  T2 = LineCir(Cir,Vox,'ab','x-')
  T3 = T1
  T4 = Vox%b
  PArea = PArea1(Cir,T1,T2,T3,T4,'x')
case default
  write(*,*) 'PDM_Partial_Mass|(1,2)|1 corner|wrong corner: '//CV%cnr(1)
  stop
end select
  else if(Cir%r < CV%dcnr(2)) then ! one corner inside
  icd = Get_Side_i(CV,'cd')
  select case(CV%cnr(1))
  case ('a ')
    if(Cir%r > CV%dsid(icd)) then ! beyond the far side
      T1 = LineCir(Cir,Vox,'ab','x+')
      T3 = Vox%a
      T4 = LineCir(Cir,Vox,'da','y+')
      T2%y = T4%y
      T2%x = TCross(T2%y,'x+',Cir)
      T5 = LineCir(Cir,Vox,'cd','x+')
      T6 = LineCir(Cir,Vox,'cd','x-')
      PArea = PArea1(Cir,T1,T2,T3,T4,'x')

```

```

PArea = PArea + Moon(Cir,T2,T4) - Moon(Cir,T5,T6)
  else
T1 = LineCir(Cir,Vox,'ab','x+')
T2 = LineCir(Cir,Vox,'da','y+')
T3 = Vox%a
T4 = T2
PArea = PArea1(Cir,T1,T2,T3,T4,'x')
  end if
case ('b ')
  if(Cir%r > CV%dsid(icd)) then ! beyond the far side
T2 = LineCir(Cir,Vox,'ab','x-')
T3 = LineCir(Cir,Vox,'bc','y+')
T4 = Vox%b
T1%y = T3%y
T1%x = TCross(T1%y,'x-',Cir)
T5 = LineCir(Cir,Vox,'cd','x+')
T6 = LineCir(Cir,Vox,'cd','x-')
PArea = PArea1(Cir,T1,T2,T3,T4,'x')
PArea = PArea + Moon(Cir,T2,T4) - Moon(Cir,T5,T6)
  else
T1 = LineCir(Cir,Vox,'bc','y+')
T2 = LineCir(Cir,Vox,'ab','x-')
T3 = T1
T4 = Vox%b
PArea = PArea1(Cir,T1,T2,T3,T4,'x')
  end if
case default
  write(*,*) 'PDM_Partial_Mass|(2,1)|1 corner|wrong corner: '//CV%cnr(1)
  stop
end select
  else if(Cir%r < CV%dcnr(3)) then ! two corners inside
select case(CV%cnr(1)//CV%cnr(2))
case ('a b ', 'b a ')
  T1 = LineCir(Cir,Vox,'bc','y+')
  T2 = LineCir(Cir,Vox,'da','y+')
  T3 = Vox%b
  T4 = Vox%a
  PArea = PArea1(Cir,T1,T2,T3,T4,'y')
  icd = Get_Side_i(CV,'cd')
  if(Cir%r > CV%dsid(icd)) then ! beyond the far side
T5 = LineCir(Cir,Vox,'cd','x+')
T6 = LineCir(Cir,Vox,'cd','x-')
PArea = PArea - Moon(Cir,T5,T6)
  end if
case ('b c ')
  T1 = LineCir(Cir,Vox,'cd','x-')
  T2 = LineCir(Cir,Vox,'ab','x-')
  T3 = Vox%c
  T4 = Vox%b
  PArea = PArea1(Cir,T1,T2,T3,T4,'x')
case ('a d ')
  T1 = LineCir(Cir,Vox,'ab','x+')
  T2 = LineCir(Cir,Vox,'cd','x+')
  T3 = Vox%a
  T4 = Vox%d
  PArea = PArea1(Cir,T1,T2,T3,T4,'x')
case default
  write(*,*) 'PDM_Partial_Mass|(2,1)|2 corners|wrong combination: '//CV%cnr(1)//CV%cnr(2)
  stop
end select
  else if(Cir%r < CV%dcnr(4)) then ! three corners inside
select case(CV%cnr(4))
case ('c ')
  T1 = LineCir(Cir,Vox,'bc','y+')
  T2 = LineCir(Cir,Vox,'cd','x+')
  T4 = Vox%d
  T3%x = T4%x
  T3%y = T1%y

```



```

PArea = PArea1(Cir,T1,T2,T3,T4,'x')
PArea = PArea + abs((T1%x-Vox%a%x)*(T1%y-Vox%a%y))
case ('d ')
T1 = LineCir(Cir,Vox,'cd','x-')
T2 = LineCir(Cir,Vox,'da','y+')
T3 = Vox%c
T4%x = T3%x
T4%y = T2%y
PArea = PArea1(Cir,T1,T2,T3,T4,'x')
PArea = PArea + abs((T2%x-Vox%b%x)*(T2%y-Vox%b%y))
case default
write(*,*) 'PDM_Partial_Mass|(2,1)|3 corners|no such corner: '//CV%cnr(4)
stop
end select
else ! all inside
PArea = AVox
end if
! - (2,2)
case (2)
cnr: if(Cir%r <= CV%dcnr(1)) then ! no corner inside
sid1: if(Cir%r <= CV%dsid(1)) then ! all inside
PArea = ACir
else if(Cir%r <= CV%dsid(2)) then ! cross 1 side
select case(CV%sid(1))
case ('ab')
T1 = LineCir(Cir,Vox,'ab','x-')
T2 = LineCir(Cir,Vox,'ab','x+')
case ('bc')
T1 = LineCir(Cir,Vox,'bc','y-')
T2 = LineCir(Cir,Vox,'bc','y+')
case ('cd')
T1 = LineCir(Cir,Vox,'cd','x+')
T2 = LineCir(Cir,Vox,'cd','x-')
case ('da')
T1 = LineCir(Cir,Vox,'da','y+')
T2 = LineCir(Cir,Vox,'da','y-')
case default
write(*,*) 'Partial_Mass|(2,2)|cross 1 side|no this side: '//CV%sid(1)
stop
end select
PArea = ACir - Moon(Cir,T1,T2)
else if(Cir%r <= CV%dsid(3)) then ! cross 2 sides
select case(CV%sid(1)//CV%sid(2))
case ('abcd', 'cdab')
T1 = LineCir(Cir,Vox,'ab','x-')
T2 = LineCir(Cir,Vox,'ab','x+')
T3 = LineCir(Cir,Vox,'cd','x+')
T4 = LineCir(Cir,Vox,'cd','x-')
case ('abbc', 'bcab')
T1 = LineCir(Cir,Vox,'ab','x-')
T2 = LineCir(Cir,Vox,'ab','x+')
T3 = LineCir(Cir,Vox,'bc','y-')
T4 = LineCir(Cir,Vox,'bc','y+')
case ('bccd', 'cdbc')
T1 = LineCir(Cir,Vox,'bc','y-')
T2 = LineCir(Cir,Vox,'bc','y+')
T3 = LineCir(Cir,Vox,'cd','x+')
T4 = LineCir(Cir,Vox,'cd','x-')
case ('cdda', 'dacd')
T1 = LineCir(Cir,Vox,'cd','x+')
T2 = LineCir(Cir,Vox,'cd','x-')
T3 = LineCir(Cir,Vox,'da','y+')
T4 = LineCir(Cir,Vox,'da','y-')
case ('daab', 'abda')
T1 = LineCir(Cir,Vox,'da','y+')
T2 = LineCir(Cir,Vox,'da','y-')
T3 = LineCir(Cir,Vox,'ab','x-')
T4 = LineCir(Cir,Vox,'ab','x+')

```

```

    case ('bcda', 'dabc')
T1 = LineCir(Cir,Vox,'bc','y-')
T2 = LineCir(Cir,Vox,'bc','y+')
T3 = LineCir(Cir,Vox,'da','y+')
T4 = LineCir(Cir,Vox,'da','y-')
    case default
write(*,*) 'Partial_Mass|(2,2)|cross 2 sides|no such side combination: '//CV%sid(1)//CV%sid(2)
stop
    end select
    PArea = ACir - Moon(Cir,T1,T2) - Moon(Cir,T3,T4)
else if(Cir%r <= CV%dsid(4)) then ! cross 3 sides
    select case(CV%sid(4))
    case ('ab')
T1 = LineCir(Cir,Vox,'bc','y-')
T2 = LineCir(Cir,Vox,'bc','y+')
T3 = LineCir(Cir,Vox,'cd','x+')
T4 = LineCir(Cir,Vox,'cd','x-')
T5 = LineCir(Cir,Vox,'da','y+')
T6 = LineCir(Cir,Vox,'da','y-')
    case ('bc')
T1 = LineCir(Cir,Vox,'cd','x+')
T2 = LineCir(Cir,Vox,'cd','x-')
T3 = LineCir(Cir,Vox,'da','y+')
T4 = LineCir(Cir,Vox,'da','y-')
T5 = LineCir(Cir,Vox,'ab','x-')
T6 = LineCir(Cir,Vox,'ab','x+')
    case ('cd')
T1 = LineCir(Cir,Vox,'da','y+')
T2 = LineCir(Cir,Vox,'da','y-')
T3 = LineCir(Cir,Vox,'ab','x-')
T4 = LineCir(Cir,Vox,'ab','x+')
T5 = LineCir(Cir,Vox,'bc','y-')
T6 = LineCir(Cir,Vox,'bc','y+')
    case ('da')
T1 = LineCir(Cir,Vox,'ab','x-')
T2 = LineCir(Cir,Vox,'ab','x+')
T3 = LineCir(Cir,Vox,'bc','y-')
T4 = LineCir(Cir,Vox,'bc','y+')
T5 = LineCir(Cir,Vox,'cd','x+')
T6 = LineCir(Cir,Vox,'cd','x-')
    case default
write(*,*) 'Partial_Mass|(2,2)|cross 3 sides|no such side left: '//CV%sid(4)
stop
    end select
    PArea = ACir - Moon(Cir,T1,T2) - Moon(Cir,T3,T4) - Moon(Cir,T5,T6)
else ! cross all 4 sides
    T1 = LineCir(Cir,Vox,'ab','x-')
    T2 = LineCir(Cir,Vox,'ab','x+')
    T3 = LineCir(Cir,Vox,'bc','y-')
    T4 = LineCir(Cir,Vox,'bc','y+')
    T5 = LineCir(Cir,Vox,'cd','x+')
    T6 = LineCir(Cir,Vox,'cd','x-')
    T7 = LineCir(Cir,Vox,'da','y+')
    T8 = LineCir(Cir,Vox,'da','y-')
    PArea = ACir - Moon(Cir,T1,T2) - Moon(Cir,T3,T4) - Moon(Cir,T5,T6) - Moon(Cir,T7,T8)
end if sid1
    else if(Cir%r <= CV%dcnr(2)) then ! 1 corner inside
select case(CV%cnr(1))
case ('a ')
    PArea = PA22OneCorner(Cir, Vox)
case ('b ')
    Cir22 = Cir
    Cir22%x = 2*VC%x - Cir%x
    PArea = PA22OneCorner(Cir22, Vox)
case ('c ')
    Cir22 = Cir
    Cir22%x = 2*VC%x - Cir%x
    Cir22%y = 2*VC%y - Cir%y

```

```

    PArea = PA22OneCorner(Cir22, Vox)
case ('d ')
    Cir22 = Cir
    Cir22%y = 2*VC%y - Cir%y
    PArea = PA22OneCorner(Cir22, Vox)
case default
    write(*,*) 'PDM_Partial_Mass|(2,2)|1 corner|no such corner: '//CV%cnr(1)
    stop
end select
    else if(Cir%r <= CV%dcnr(3)) then ! 2 corners inside
select case(CV%cnr(1)//CV%cnr(2))
case ('a b ', 'b a ')
    PArea = PA22TwoCorner(Cir, Vox, 'ab')
case ('a d ', 'd a ')
    PArea = PA22TwoCorner(Cir, Vox, 'da')
case ('c d ', 'd c ')
    Cir22 = Cir
    Cir22%y = 2*VC%y - Cir%y
    PArea = PA22TwoCorner(Cir22, Vox, 'ab')
case ('b c ', 'c b ')
    Cir22 = Cir
    Cir22%x = 2*VC%x - Cir%x
    PArea = PA22TwoCorner(Cir22, Vox, 'da')
case default
    write(*,*) 'PDM_Partial_Mass|(2,2)|2 corners|no such corner combination: '//CV%cnr(1)//CV%cnr(2)
    stop
end select
    else if(Cir%r <= CV%dcnr(4)) then ! 3 corners inside
select case(CV%cnr(4))
case ('a ')
    PArea = PA22ThreeCorner(Cir, Vox)
case ('b ')
    Cir22 = Cir
    Cir22%x = 2*VC%x - Cir%x
    PArea = PA22ThreeCorner(Cir22, Vox)
case ('c ')
    Cir22 = Cir
    Cir22%x = 2*VC%x - Cir%x
    Cir22%y = 2*VC%y - Cir%y
    PArea = PA22ThreeCorner(Cir22, Vox)
case ('d ')
    Cir22 = Cir
    Cir22%y = 2*VC%y - Cir%y
    PArea = PA22ThreeCorner(Cir22, Vox)
case default
    write(*,*) 'PDM_Partial_Mass|(2,2)|3 corners|no such corner left: '//CV%cnr(4)
    stop
end select
    else ! 4 corners inside
PArea = AVox
    end if  cnr
case default
    stop 'Partial_Mass|Wrong mapped index into 2nd row'
end select  row2
    case default
stop 'Partial_Mass|no such mapped index'
    end select  regn

end function Partial_Area

integer function get_side_i(CV, sgn) result(isgn)
!*****
! See what's the index i for side name sn.
!*****
    type (Circle2Voxel), intent(in) :: CV
    character (len=2), intent(in) :: sgn

    integer :: i

```

```

!!!

do i=1, 4
if(CV%sid(i)==sgn) then
  isgn = i
  exit
end if
end do

end function get_side_i

type (Point) function LineCir(Cir,Vox,Side,sgn) result(T)
!*****
! Calculate the interception point T of Cir and Side
!*****
type (Circle), intent(in) :: Cir
type (Rectangle), intent(in) :: Vox
character (len=2), intent(in) :: Side, sgn

!!!

select case (Side)
case ('ab')
  T%y = Vox%a%y
T%x = TCross(T%y,sgn,Cir)
case ('bc')
  T%x = Vox%b%x
T%y = TCross(T%x,sgn,Cir)
case ('cd')
  T%y = Vox%c%y
T%x = TCross(T%y,sgn,Cir)
case ('da')
  T%x = Vox%d%x
T%y = TCross(T%x,sgn,Cir)
case default
  write(*,*) 'LineCir: no such side: '//Side
stop
end select

end function LineCir

real(8) function TCross(v1, sgn, Cir) result(v2)
!*****
! Solve for the other coordinate v2, given v1
!*****

real(8), intent(in) :: v1
character (len=2), intent(in) :: sgn
type (Circle), intent(in) :: Cir

!!!

if(sgn=='x-') then
v2 = Cir%x - sqrt(Cir%r**2 - (v1-Cir%y)**2)
else if(sgn=='x+') then
v2 = Cir%x + sqrt(Cir%r**2 - (v1-Cir%y)**2)
else if(sgn=='y-') then
v2 = Cir%y - sqrt(Cir%r**2 - (v1-Cir%x)**2)
else if(sgn=='y+') then
v2 = Cir%y + sqrt(Cir%r**2 - (v1-Cir%x)**2)
else
  write(*,*) 'TCross: no such sign: '//sgn
stop
end if

end function TCross

```

```

real(8) function Moon(Cir,T1,T2) result(AA)
!*****
! Calculate the area between segment T1T2 and small arc T1T2
!*****
type (Circle), intent(in) :: Cir
type (Point), intent(in) :: T1, T2

real(8) :: LStrg ! half of the segment T1T2
real(8) :: Theta ! half of the angle extended by segment T1T2
real(8) :: Height

!!!

LStrg = 0.5*sqrt((T1%x - T2%x)**2 + (T1%y - T2%y)**2)
Theta = dasin(Lstrg/Cir%r)
Height = sqrt(Cir%r**2 - LStrg**2)

AA = Theta*Cir%r**2 - LStrg*Height

end function Moon

real(8) function PArea1(Cir, T1, T2, T3, T4, sgn) result(AA)
!*****
! Calculate the intercepted area  $S = 1/2*(a+b)*h + \text{moon}(Cir,T1,T2)$ .
! Note: T1T3 // T2T4, T3T4 perpendicular to T2T4 or T1T3;
!       a = abs(T1 - T3); b = abs(T2 - T4); h = abs(T3 - T4)
!*****
type (Circle), intent(in) :: Cir
type (Point), intent(in) :: T1, T2, T3, T4
character (len=1), intent(in) :: sgn ! direction in which parallel sides are

real(8) :: ds1, ds2, dh
real(8) :: LStrg ! half of the segment T1T2
real(8) :: Theta ! half of the angle extended by segment T1T2
real(8) :: Height

!!!

select case(sgn)
case('x')
ds1 = abs(T1%x - T3%x)
ds2 = abs(T2%x - T4%x)
dh = abs(T3%y - T4%y)
case('y')
ds1 = abs(T1%y - T3%y)
ds2 = abs(T2%y - T4%y)
dh = abs(T3%x - T4%x)
end select
AA = 0.5*(ds1 + ds2)*dh

AA = AA + Moon(Cir,T1,T2)

end function PArea1

real(8) function PA22OneCorner(Cir, Vox) result(AA)
!*****
! Calculate the intercepted area given the circle is in region (2,2) and
! only corner a is inside.
! * 'ab' and 'da' already crossed so only need to check
! crossing 'bc' and 'cd' or not
!*****
type (Circle), intent(in) :: Cir
type (Rectangle), intent(in) :: Vox

type (Circle2Voxel) :: CV
type (Point) :: T1, T2, T3, T4, T5, T6
integer (i4) :: isid

```

```

!!!

CV = C2V(Cir,Vox)

T1 = LineCir(Cir,Vox,'ab','x+')
T2 = LineCir(Cir,Vox,'da','y+')
T3 = Vox%a
T4 = T2
AA = PArea1(Cir,T1,T2,T3,T4,'x')
! -- side 'bc'
! isid = get_side_i(CV, 'bc')
! if(CV%dsid(isid)<Cir%r) then
T5 = LineCir(Cir,Vox,'bc','y-')
T6 = LineCir(Cir,Vox,'bc','y+')
AA = AA - Moon(Cir,T5,T6)
! end if
! -- side 'cd'
! isid = get_side_i(CV, 'cd')
! if(CV%dsid(isid)<Cir%r) then
T5 = LineCir(Cir,Vox,'cd','x+')
T6 = LineCir(Cir,Vox,'cd','x-')
AA = AA - Moon(Cir,T5,T6)
! end if

end function PA22OneCorner

real (r8) function PA22TwoCorner(Cir, Vox, sgn) result(AA)
!*****
! Calculate the intercepted area given the circle is in region (2,2) and
! 2 corners are inside
!*****
type (Circle), intent(in) :: Cir
type (Rectangle), intent(in) :: Vox
character (len=2), intent(in) :: sgn

type (Circle2Voxel) :: CV
type (Point) :: T1, T2, T3, T4, T5, T6
integer :: icd, ibc

!!!

CV = C2V(Cir,Vox)

select case(sgn)
case ('ab')
T1 = LineCir(Cir,Vox,'bc','y+')
T2 = LineCir(Cir,Vox,'da','y+')
T3 = Vox%b
T4 = Vox%a
AA = PArea1(Cir,T1,T2,T3,T4,'y')
icd = get_side_i(CV,'cd')
if(Cir%r > CV%dsid(icd)) then
T5 = LineCir(Cir,Vox,'cd','x+')
T6 = LineCir(Cir,Vox,'cd','x-')
AA = AA - Moon(Cir,T5,T6)
end if
case ('da')
T1 = LineCir(Cir,Vox,'ab','x+')
T2 = LineCir(Cir,Vox,'cd','x+')
T3 = Vox%a
T4 = Vox%d
AA = PArea1(Cir,T1,T2,T3,T4,'x')
ibc = get_side_i(CV,'bc')
if(Cir%r > CV%dsid(ibc)) then
T5 = LineCir(Cir,Vox,'bc','y-')
T6 = LineCir(Cir,Vox,'bc','y+')

```

```

AA = AA - Moon(Cir,T5,T6)
end if
case default
write(*,*) 'PA22TwoCorner|2 corners|no such corner combination: '//sgn
end select

end function PA22TwoCorner

real(8) function PA22ThreeCorner(Cir, Vox) result(AA)
!*****
! Calculate the intercepted area given the circle is in region (2,2) and
! only corner a is outside
!*****
type (Circle), intent(in) :: Cir
type (Rectangle), intent(in) :: Vox

type (Circle2Voxel) :: CV
type (Point) :: T1, T2, T3, T4

!!!

CV = C2V(Cir,Vox)

T1 = LineCir(Cir,Vox,'da','y-')
T2 = LineCir(Cir,Vox,'ab','x-')
T4 = Vox%b
T3%x = T4%x
T3%y = T1%y
AA = PAreal(Cir,T1,T2,T3,T4,'x')
AA = AA + abs((T1%x-Vox%c%x)*(T1%y-Vox%c%y))
end function PA22ThreeCorner

type (Index) function Get_Index(Cir,Vox) result(Idx)
!*****
! get the relative location of the circle's origin to the rectangle.
!
!
!          ----->y
!          | (1,1) | (1,2) | (1,3) |
!          -----a-----d-----
!          | (2,1) | (2,2) | (2,3) |
!          -----b-----c-----
!          | (3,1) | (3,2) | (3,3) |
!          -----
!          \
!          x
!*****
type (Circle), intent(in) :: Cir
type (Rectangle), intent(in) :: Vox

!!!

Idx%k = 0

if(Cir%x <= Vox%a%x) then
  if(Cir%y <= Vox%a%y) then ! - (1,1)
    Idx%i = 1
  Idx%j = 1
  else if(Cir%y >= Vox%c%y) then ! - (1,3)
    Idx%i = 1
  Idx%j = 3
  else ! - (1,2)
    Idx%i = 1
  Idx%j = 2
  end if
else if(Cir%x >= Vox%c%x) then

```

```

    if(Cir%y <= Vox%a%y) then ! - (3,1)
      Idx%i = 3
Idx%j = 1
    else if(Cir%y >= Vox%c%y) then ! - (3,3)
      Idx%i = 3
Idx%j = 3
    else ! - (3,2)
      Idx%i = 3
Idx%j = 2
    end if
else
  if(Cir%y <= Vox%a%y) then ! - (2,1)
    Idx%i = 2
Idx%j = 1
  else if(Cir%y >= Vox%c%y) then ! - (2,3)
    Idx%i = 2
Idx%j = 3
  else ! - (2,2)
    Idx%i = 2
Idx%j = 2
  end if
end if

end function Get_Index

type (Circle2Voxel) function C2V(Cir,Vox) result(CV)
!*****
! get the distance of the circle's origin to 4 corners: a,b,c and d of
! a voxel(im,jm)
!*****

  type (Circle),    intent(in) :: Cir
  type (Rectangle), intent(in) :: Vox

  real(8) :: dp(4), dsd(4)
  integer :: i, j

!!!

  dp(1) = sqrt((Cir%x - Vox%a%x)**2 + (Cir%y - Vox%a%y)**2)
  dp(2) = sqrt((Cir%x - Vox%b%x)**2 + (Cir%y - Vox%b%y)**2)
  dp(3) = sqrt((Cir%x - Vox%c%x)**2 + (Cir%y - Vox%c%y)**2)
  dp(4) = sqrt((Cir%x - Vox%d%x)**2 + (Cir%y - Vox%d%y)**2)

  dsd(1) = abs(Cir%y - Vox%a%y)
  dsd(2) = abs(Cir%x - Vox%c%x)
  dsd(3) = abs(Cir%y - Vox%c%y)
  dsd(4) = abs(Cir%x - Vox%a%x)

! -- sort the distances to the corner while remembering their correspondances
CV%dcnr = dp
CV%cnr = cname
call sort(CV%dcnr,CV%cnr)

! -- sort the distances to the side while remembering their correspondances
CV%dsid = dsd
CV%sid = sname
call sort(CV%dsid,CV%sid)

end function C2V

SUBROUTINE sort(arr,carr)
!*****
! Insertion sort
!*****

  real(r8), intent(inout) :: arr(:)
  character (len=2), intent(inout) :: carr(:)

```



```

REAL(r8) :: a
character (len=2) :: ca
INTEGER(i4) :: n, i, j

n = size(arr)

do j=2, n
  a = arr(j)
ca = carr(j)
  do i=j-1, 1, -1
    if (arr(i) <= a) exit
    arr(i+1) = arr(i)
  carr(i+1) = carr(i)
  end do
  arr(i+1) = a
carr(i+1) = ca
end do

end subroutine sort

subroutine Get_abcd(im,jm)
!*****
! get the corners -- a,b,c and d's coordinates of voxel(im,jm)
!
! comments:
!   this is used in part to calculate the partial area by
!   airway/artery/vein in lung. so limit to lung's boundaries
!*****

integer, intent(in) :: im, jm

!!!

! -- a
xyVox%a%x = max(XDose(im), LBndX(1))
xyVox%a%y = max(YDose(jm), LBndY(1))
! -- c
xyVox%c%x = min(XDose(im+1), LBndX(2))
xyVox%c%y = min(YDose(jm+1), LBndY(2))
! -- b
xyVox%b%x = xyVox%c%x
xyVox%b%y = xyVox%a%y
! -- d
xyVox%d%x = xyVox%a%x
xyVox%d%y = xyVox%c%y

  end subroutine Get_abcd

subroutine FracVol(bx1, bx2, vol)
!*****
! get the volume of intersection of box1 and box2
! region: 1 --> surrounding water; 2 -- lung; 3 -- mix
!*****

type (Box), intent(in) :: bx1, bx2
real (r8), intent(out) :: vol

real (r8) :: a(2), b(2), ds

!!!

vol = 1.d0

! -- x direction
a(1) = bx1%head%x
a(2) = bx1%tail%x
b(1) = bx2%head%x

```

```

b(2) = bx2%tail%x

call Overlap(a, b, ds)

vol = vol*ds

! -- y direction
a(1) = bx1%head%y
a(2) = bx1%tail%y
b(1) = bx2%head%y
b(2) = bx2%tail%y

call Overlap(a, b, ds)

vol = vol*ds

! -- z direction
a(1) = bx1%head%z
a(2) = bx1%tail%z
b(1) = bx2%head%z
b(2) = bx2%tail%z

call Overlap(a, b, ds)

vol = vol*ds

if(vol<ZERO) then
  write(*,*) "Error in FracVol(), check it"
  stop
end if
end subroutine FracVol

```

```

subroutine Switch(x,y)
!-----
! switch x and y
!-----

real (8), intent(inout) :: x, y

real (8) :: tmp

tmp = x; x = y; y = tmp

end subroutine Switch

```

```

subroutine Overlap(a, b, ds)
!-----
! given two 1D segments, determine the overlap
!-----

real (r8), intent(inout) :: a(2), b(2)
real (r8), intent(out) :: ds

real (r8) :: La, Lb

!!!

! -- always let La <= Lb
La = a(2) - a(1)
Lb = b(2) - b(1)
if(La>Lb) then
  call Switch(La, Lb)
  call Switch(a(1), b(1))
  call Switch(a(2), b(2))
end if

```

```

    if(a(2)<=b(1)) then          ! (1) a1____a2 b1____b2
      ds = ZERO
    else if(a(2)<b(2)) then
      if(a(1)<b(1)) then        ! (2) a1__b1_a2____b2
        ds = a(2) - b(1)
      else                      ! (3) b1__a1____a2____b2
        ds = La
      end if
    else
      if(a(1)<b(2)) then        ! (4) b1____a1__b2__a2
        ds = b(2) - a(1)
      else                      ! (5) b1____b2 a1____a2
        ds = ZERO
      end if
    end if

end subroutine Overlap

real (r8) function VBox(bx) result(vol)
!*****
! get its volume given a box
!*****

type (Box), intent(in) :: bx

!!!

vol = abs((bx%tail%x - bx%head%x)* &
          (bx%tail%y - bx%head%y)* &
          (bx%tail%z - bx%head%z))
end function VBox

end Module NPDM_Get_Partial_Mass

module NPDM_SetGrid

use NPDM_Parameters

implicit none

contains

subroutine ReadGrid(axis,buffer,ND,NDRgn,NDT,DBnD,dD)

character (len=1), intent(in) :: axis
character (len=128), intent(in) :: buffer
integer (i4), intent(inout) :: ND(:), NDRgn, NDT
real (r8), intent(inout) :: DBnD(:), dD(:)

real (r8) :: DLim1, DLim2
integer (i4) :: i, j, k

!!!

NDRgn = NDRgn + 1
read(buffer,*) dD(NDRgn), DLim1, DLim2
if(DLim1>=DLim2) then
  write(*,'(a)') 'Dose grid '//axis//'-bounds should be increasing.'
  stop
end if
if(NDRgn.gt.1.AND.DLim1.lt.DBnD(NDRgn)) then
  write(6,'(a)') 'The adjacent dose grid '//axis//'-bounds overlapped.'
  stop
end if
DBnD(NDRgn) = DLim1
DBnD(NDRgn+1) = DLim2
ND(NDRgn) = nint((DBnD(NDRgn+1)-DBnD(NDRgn))/dD(NDRgn))

```

```

    NDT = NDT + ND(NDRgn)

end subroutine ReadGrid

subroutine SetGrid(axis,ND,NDRgn,NDT,DBnD,dD,Dose)

    character (len=1), intent(in) :: axis
    integer (i4), intent(in) :: ND(:), NDRgn, NDT
    real (r8), intent(in) :: DBnd(:), dD(:)
    real (r8), intent(inout) :: Dose(:)
    integer (i4) :: i, j, k

    !!!

    k = 0
    do i=1, NDRgn
        if(dD(i).LT.0.0) then
            write(*,'(a,i3,a)') 'd'//axis//'D'(i,') could not be negative.'
            stop
        end if
        do j=1, ND(i)
            k = k + 1
            Dose(k) = DBnd(i) + dD(i)*(j-1)
        end do
    end do
    Dose(NDT+1) = DBnd(NDRgn+1)

end subroutine SetGrid

end module NPDM_SetGrid

!
! type definitions
!
module NPDM_Types

use NPDM_Parameters

type :: Point
real (r8) :: x, &
                y, &
                z
end type Point

type :: Circle
real (r8) :: x, &
                y, &
                r
end type Circle

type :: Airway
type (Circle) :: ci, &
                co, &
                ca, &
                cbd
end type Airway

type :: Rectangle
type (Point) :: a, b, c, d
!!!!!!!!!!!!y+
!a      d!
!      !
!b      c!
!!!!!!!!!!!!
!x+
end type Rectangle

```

```

type :: Box
  type (Point) :: head, tail
end type Box

type :: Circle2Voxel
real (r8) :: dcnr(4), &
          dsid(4)
character (len=2) :: cnr(4), &
                 sid(4)
end type Circle2Voxel

type :: Index
integer (i4) :: i, &
              j, &
              k
end type Index

end module NPDM_Types

```

## A.5 Fortran 77 PENELOPE main program, based on “Pendoses” from the PENELOPE distribution, and modified by Liang for the $2\frac{1}{2}$ -D random lung model

```

C *****
C                               main program
C   modified by L. Liang
C *****
  implicit DOUBLE precision (A-H,O-Z), integer*4 (I-N)
C
  parameter (PI=3.1415926535897932D0, TWOPI=2.0D0*PI,
1  RA2DE=180.0D0/PI, DE2RA=PI/180.0D0)
C
C   -- command line argument relative
integer narg
  character*32 argi, argo, argsfx
  logical exists
integer len_sfx
C
  character*2 LIT
  character*128 BUFFER, MBuffer
  character*32 PFILE,GFILE, MFILE, DUMPF,RESUMEF
C
  character*6 KWORD,
1  KWTITL,KWKPAR,KWSENE,KWSPEC, KWSEXT,KWSHEI,KWSRAD,KWSPOS,
2  KWSDIR,KWSAPE,KWNMAT,KWSIMP, KWPFNA,KWNBE ,KWNBTH,KWNBPH,
3  KWNBZ ,KWNBR ,KWABSE,KWNBTL, KWDO2D,KWIFOR,KWRESU,KWDUMP,
4  KWNSIM,KWTIME,KWRSEE,KWCOMM,
C   -- ADDED KEYWORDS BY LIANG
5  KWSCIR, KWSSQU, KWGFNM, KWMFNM, KWFOUT, KWNITR,
6  KWDXDO, KWDYDO, KWDZDO
  parameter(
1  KWTITL='TITLE ',KWKPAR='SKPAR ',KWSENE='SENERG',KWSPEC='SPECTR',
2  KWSEXT='SEXTND',KWSHEI='STHICK',KWSRAD='SRADII',KWSPOS='SPOSIT',
3  KWSDIR='SDIREC',KWSAPE='SAPERT',KWNMAT='NMAT ',KWSIMP='SIMP',
4  KWPFNA='PFNAME',KWNBE ='NBE ',KWNBTH='NBTH ',KWNBPH='NBPH ',
5  KWNBZ ='NBZ ',KWNBR ='NBR ',KWNBTL='NBTL ',KWABSE='ABSEN ',
6  KWDO2D='DOSE2D',KWIFOR='IFORCE',KWRESU='RESUME',KWDUMP='DUMPTO',
7  KWNSIM='NSIMSH',KWTIME='TIME ',KWRSEE='RSEED ',KWCOMM=' ')
C   -- ADDED KEYWORDS BY LIANG
parameter (KWSCIR='SCIRCU') ! circular field
parameter (KWSSQU='SSQURE') ! square field
parameter (KWGFNM='GFNAME') ! geometry file name
parameter (KWMFNM='MFNAME') ! mass file name
parameter (KWFOUT='FULOUT') ! full output control

```

```

parameter (KWNITR='NINTER') ! timing output interval
parameter (KWDXDO='DXDOSE') !
parameter (KWDYDO='DYDOSE') !
parameter (KWDZDO='DZDOSE') !

C   -- Main-PENELOPE commons.
      parameter (MAXMAT=10)
      common/CSIMPA/EABS(3,MAXMAT),C1(MAXMAT),C2(MAXMAT),WCC(MAXMAT),
1     WCR(MAXMAT)
      common/TRACK/E,X,Y,Z,U,V,W,WGHT,KPAR,IBODY,MAT,ILB(5)
      common/RSEED/ISEED1,ISEED2

C   -- Geometry.
      dimension PARINP(20)
      dimension DSMAX(MAXMAT)

C   -- Source.
C   -- Source energy spectrum.
      parameter (NSEBM=100)
      dimension ES(NSEBM),PTS(NSEBM),IAS(NSEBM),FS(NSEBM)

C   -- Dose grid and tallies.
      parameter (NDXM=240,NDYM=300,NDZM=140)
      parameter (NDRgnM=20)
      parameter (NDM1=NDXM*NDYM*NDZM, NDM3=3*NDXM*NDYM*NDZM)
      dimension DBndX(NDRgnM+1), DBndY(NDRgnM+1), DBndZ(NDRgnM+1)
      dimension dxD(NDRgnM+1), dyD(NDRgnM+1), dzD(NDRgnM+1)
      dimension NxD(NDRgnM+1), NyD(NDRgnM+1), NzD(NDRgnM+1)
      dimension Dose(NDXM,NDYM,NDZM),
1         VDose(NDXM,NDYM,NDZM),
1         DoseP(NDXM,NDYM,NDZM),
3         Mass(NDXM,NDYM,NDZM),
1         LDose(NDXM,NDYM,NDZM)
      double precision Mass
      data Dose, VDose, DoseP/NDM3*0d0/
      data LDose/NDM1*0/
Common/DoseGrid/YDose(NDYM+1),XDose(NDXM+1),ZDose(NDZM+1),
1     NyDT,NxDT,NzDT
      dimension DAV(NDXM), DErr(NDXM), DRel(NDXM)

C
      external RAND

C
=====
C   Instructions
C
=====

      do M=1,MAXMAT
        C1(M)=0.0DO
        C2(M)=0.0DO
        WCC(M)=0.0DO
        WCR(M)=0.0DO
        EABS(1,M)=0.0DO
        EABS(2,M)=0.0DO
        EABS(3,M)=0.0DO
        DSMAX(M)=1.0E20
      end do

C   -- Time counter initiation.
      call TIMEO

C   -- Read in the input file.
narg = iargc()
if(narg.gt.0) then
  if(narg.ne.3) then
    write(*,*) '** Exactly 3 input arguments: input & output'//
1      ' files and output suffix, please **'
    stop
  else

```

```

    call getarg(1, argi)
    call getarg(2, argo)
    call getarg(3, argsfx)
    end if
else
    argi = 'newpendoses.in'
    argo = 'newpendoses.out'
    argsfx = 'dft'
end if

    inquire(FILE = argi, EXIST = exists)
    if(.not. exists) then
        write(*,'(2A/)') ' >> Cannot find file ', argi
        stop
    end if

    len_sfx = len_trim(argsfx)
C    len_sfx = lnblnk(argsfx)

    open(5,FILE=argi)
    open(6,FILE=argo)
    write(6,1000)
1000 format(/3X,48('*')/,3X,'** Program NEWPENDOSES. ',
1 'Source and geometry. **'/3X,48('*'))

C -- Title.
20 continue
read(5,'(A6,1X,A128)') KWORD,BUFFER
if(KWORD.EQ.KWCOMM) go to 20
if(KWORD.NE.KWTITL) then
    write(6,*) 'The title line does not exist.'
    stop 'The title line does not exist.'
end if
write(6,1010) BUFFER
1010 format(/2X,A128)

C -- Source data.
C -- Source description.
write(6,1200)
1200 format(/3X,70('-')/,3X,'>>>>> Source description.

21 continue
read(5,'(A6,1X,A128)') KWORD,BUFFER
if(KWORD.EQ.KWCOMM) go to 21

C -- primary particle type.
if(KWORD.EQ.KWKPAR) then
    read(BUFFER,*) KPARP
22 continue
read(5,'(A6,1X,A128)') KWORD,BUFFER
if(KWORD.EQ.KWCOMM) go to 22
else
    KPARP=1
end if
if(KPARP.LT.1.OR.KPARP.GT.3) KPARP=1
if(KPARP.EQ.1) write(6,1021)
1021 format(/3X,'Primary particles: electrons')
if(KPARP.EQ.2) write(6,1022)
1022 format(/3X,'Primary particles: photons')
if(KPARP.EQ.3) write(6,1023)
1023 format(/3X,'Primary particles: positrons')

C -- Initial energy of primary particles.
ISPEC=0
if(KWORD.EQ.KWSENE) then
    NSEB=1
    read(BUFFER,*) EO
    write(6,1030) EO

```

```

1030 format(3X,'Initial energy = ',1P,E13.6,' eV')
23  continue
    read(5,'(A6,1X,A128)') KWORD,BUFFER
    if(KWORD.EQ.KWCOMM) go to 23
    else if(KWORD.EQ.KWSPEC) then
        ISPEC=1
        NSEB=0
24  continue
        NSEB=NSEB+1
        read(BUFFER,*) ES(NSEB),PTS(NSEB)
25  continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER
        if(KWORD.EQ.KWCOMM) go to 25
        if(KWORD.EQ.KWSPEC) go to 24
    else
        E0=1.0D6
        write(6,1030) E0
    end if
    if(ISPEC.EQ.1) THEN
        if(NSEB.GT.NSEBM) THEN
            write(6,*) 'NSEBM is too small.'
            stop 'NSEBM is too small.'
        else if(NSEB.le.1) then
            write(6,*) 'The source energy spectrum is not defined.'
            stop 'The source energy spectrum is not defined.'
        ELSE
            call SORT2(ES,PTS,NSEB)
            write(6,1031)
1031  format(/3X,'Spectrum:',7X,'I',4X,'E_low(eV)',4x,'E_high(eV)',
1      5X,'P_sum(E)',/16X,45('-'))
            do I=1,NSEB-1
                write(6,'(16X,I4,1P,5E14.6)') I,ES(I),ES(I+1),PTS(I)
            end do
            E0=ES(NSEB)
            NSEB=NSEB-1
            call IRNDO(PTS,FS,IAS,NSEB)
        ENDIF
    ENDIF
    if(E0.LT.100.0D0) then
        write(6,*) 'The initial energy E0 is too small.'
        stop 'The initial energy E0 is too small.'
    end if
    EPMAX=E0

C      -- Positrons eventually give annihilation gamma-rays. The maximum
C      energy of annihilation photons is .lt. 1.21*(E0+me*c**2).
    if(KPARP.EQ.3) EPMAX=1.21D0*(E0+5.12D5)

C      -- Source position.
    if(KWORD.EQ.KWSPOS) then
        read(BUFFER,*) X0,Y0,Z0
241  continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER
        if(KWORD.EQ.KWCOMM) go to 241
    else
        X0 = 0.0D0
        Y0 = -1.0D15
        Z0 = 0.0D0
    end if
    write(6,1040) X0,Y0,Z0
1040 format(3X,'Coordinates of centre:   X0 =',1P,E13.6,
1      ' cm',/30X,'Y0 =',E13.6,' cm',/30X,'Z0 =',E13.6,' cm')

C      -- Beam direction.
    if(KWORD.EQ.KWSDIR) then
        read(BUFFER,*) BTHETA,BPHI
252  continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER

```



```

        if(KWORD.EQ.KWCOMM) go to 252
    else
        BTHETA = 90.0D0
        BPHI   = 90.0D0
    end if
    write(6,1050) BTHETA,BPHI
1050 format(3X,'Beam direction angles:  THETA =',1P,E13.6,' deg',/
1 30X,'PHI =',E13.6,' deg')

C    -- Beam's shape

    if(KWORD.EQ.KWSAPE) then                ! 'SAPERT': circular field defined by aperture
        ISShp = 0 ! circular field
        read(BUFFER,*) BALPHA
    30  continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER
        if(KWORD.EQ.KWCOMM) go to 30
        write(6,1060) BALPHA
    else if(KWORD.EQ.KWSCIR) then           ! 'SCIRCU': circular field defined by diameter at ssd
        ISShp = 0 ! circular field
        read(BUFFER,*) d1S, ssd
    253 continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER
        if(KWORD.EQ.KWCOMM) go to 253
        if(d1S.gt.1d-35) then
            BALPHA = atan((d1S/2)/ssd)*RA2DE
            write(6,1060) BALPHA
            write(6,1041) d1S, ssd
        else
            BALPHA = 0.0d0
            write(6,1060) BALPHA
        end if
    else if(KWORD.EQ.KWSSQU) then          ! 'SSQURE': square field defined by side length at ssd
        ISShp = 1 ! square field
        read(BUFFER,*) d1S, ssd
    251 continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER
        if(KWORD.EQ.KWCOMM) go to 251
        if(d1S.gt.1d-35) then
            BALPHA = atan(sqrt(2d0)*(d1S/2)/ssd)*RA2DE
            write(6,1042) d1S, ssd
        else
            ISShp = 0 ! pencil beam (circular of course)
            BALPHA = 0.0d0
            write(6,1060) BALPHA
        end if
    else                                    ! pencil beam by default
        ISShp = 0
        BALPHA=0.0D0
        write(6,1060) BALPHA
    end if
1060 format(3X,'Beam aperture:',11X,'ALPHA =',1P,E13.6,' deg')
1041 format(3X,'diameter of the circle:    d = ',
1 1P,E13.6,' cm at distance = ',1P,E13.6,' cm')
1042 format(3X,'side length of the square: dl = ',
1 1P,E13.6,' cm at distance = ',1P,E13.6,' cm')

    call GCONEO(BTHETA*DE2RA,BPHI*DE2RA,BALPHA*DE2RA)

C    -- Material data and Simulation parameters.
    write(6,1300)
1300 format(//3X,70('-'),/
1 3X,'>>>>> Material data and simulation parameters.')

C
    if(KWORD.EQ.KWNMAT) then
        read(BUFFER,*) NMAT
    31  continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER

```

```

        if(KWORD.EQ.KWCOMM) go to 31
    else
        write(6,*) 'You have to specify the number of materials.'
        stop 'You have to specify the number of materials.'
    end if
    write(6,1070) NMAT
1070 format(3X,'Number of different materials = ',I2)
    if(NMAT.LT.1.OR.NMAT.GT.MAXMAT) then
        write(6,*) 'Wrong number of materials.'
        stop 'Wrong number of materials.'
    end if
C
    do M=1,NMAT
        EABS(1,M) = 0.010D0*EPMAX
        EABS(2,M) = 0.001D0*EPMAX
        EABS(3,M) = 0.010D0*EPMAX
        C1(M)      = 0.10D0
        C2(M)      = 0.10D0
        WCC(M)     = EABS(1,M)
        WCR(M)     = EABS(2,M)
        DSMAX(M)  = 1.0D20
    end do
C
    if(KWORD.EQ.KWSIMP) then
        read(BUFFER,*) M
        if(M.LT.1.OR.M.GT.NMAT) then
            write(6,'(A6,1X,A128)') KWORD,BUFFER
            write(6,*) 'Incorrect material number.'
            stop 'Incorrect material number.'
        end if
        read(BUFFER,*) M,EABS(1,M),EABS(2,M),EABS(3,M),C1(M),C2(M),
1      WCC(M),WCR(M),DSMAX(M)
32 continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER
        if(KWORD.EQ.KWCOMM) go to 32
        if(KWORD.EQ.KWSIMP) then
            read(BUFFER,*) M
            if(M.LT.1.OR.M.GT.NMAT) then
                write(6,'(A6,1X,A128)') KWORD,BUFFER
                write(6,*) 'Incorrect material number.'
                stop 'Incorrect material number.'
            end if
            read(BUFFER,*) M,EABS(1,M),EABS(2,M),EABS(3,M),C1(M),C2(M),
1      WCC(M),WCR(M),DSMAX(M)
            go to 32
        end if
    end if
C
    do M=1,NMAT
        if(M.EQ.1) LIT='st'
        if(M.EQ.2) LIT='nd'
        if(M.EQ.3) LIT='rd'
        if(M.GT.3) LIT='th'
        write(6,1080) M,LIT
1080 format(/3X,'**** ',I2,A2,' Material')
        if(EABS(1,M).LT.1.0D2) EABS(1,M)=1.0D2
        if(EABS(2,M).LT.1.0D2) EABS(2,M)=1.0D2
        if(EABS(3,M).LT.1.0D2) EABS(3,M)=1.0D2
        write(6,1081) EABS(1,M)
1081 format(3X,'Electron absorption energy = ',1P,E13.6,' eV')
        write(6,1082) EABS(2,M)
1082 format(3X,' Photon absorption energy = ',1P,E13.6,' eV')
        write(6,1083) EABS(3,M)
1083 format(3X,'Positron absorption energy = ',1P,E13.6,' eV')
        write(6,1084) C1(M),C2(M),WCC(M),WCR(M)
1084 format(3X,'Electron-positron transport parameters:',
1 /4X,'C1 =',1P,E13.6,',      C2 =',E13.6,/3X,'Wcc =',E13.6,
2 ' eV, Wcr =',E13.6,' eV')

```

```

        if(DSMAX(M).LT.1.0D-7) DSMAX(M)=1.0D20
        write(6,1085) DSMAX(M)
1085 format(3X,'Maximum allowed step length =',1P,E13.6,' cm')
        end do

C      -- Initialization of PENELOPE.
C      -- Material file
        if(KWORD.EQ.KWPFNA) then
            read(BUFFER,'(A32)') PFILE
            write(6,1090) PFILE
1090 format(/3X,'PENELOPE's material definition file: ',A32)
        33 continue
            read(5,'(A6,1X,A128)') KWORD,BUFFER
            if(KWORD.EQ.KWCOMM) go to 33
        else
            write(6,*) 'You have to specify a material file.'
            stop      'You have to specify a material file.'
        end if

        inquire(FILE = PFILE, EXIST = exists)
        if(.not. exists) then
            write(6,'(2A/)') ' >> Cannot find file ', PFILE
            write(*,'(2A/)') ' >> Cannot find file ', PFILE
            stop
        end if

        open(15,FILE=PFILE)
        open(16,FILE='material.'//argsfx(1:len_sfx)//'.dat')
        INFO=0
        call PEINIT(EPMAX,NMAT,15,16,INFO)
        CLOSE(UNIT=15)
        CLOSE(UNIT=16)

C      -- Geometry definition.
C
C      Define here the geometry parameters that are to be altered, if any.
C      PARINP(1)=
C      PARINP(2)= ...
        NPINP=0
C      -- Geometry file
        if(KWORD.EQ.KWGFNM) then
            read(BUFFER,'(A32)') GFILE
            write(6,1100) GFILE
1100 format(/3X,'PENGEOM's geometry definition file: ',A32,/)
        26 continue
            read(5,'(A6,1X,A128)') KWORD,BUFFER
            if(KWORD.EQ.KWCOMM) go to 26
            inquire(FILE = GFILE, EXIST = exists)
            if(.not. exists) then
                write(6,'(2A/)') ' >> Cannot find file ', GFILE
                write(*,'(2A/)') ' >> Cannot find file ', GFILE
                stop
            end if
            open(15,FILE=GFILE)
            call GEOMIN(PARINP,NPINP,NMATG,NBOD,15,6)
            close(UNIT=15)
            if(NMATG.GT.NMAT) THEN
                write(6,1101)
1101 format(/6X,'Too many different materials.')
                stop      'Too many different materials.'
            ENDIF
        else
            write(6,*) 'You have to specify a geometry file.'
            stop      'You have to specify a geometry file.'
        end if
C      -- Mass file correspondent to the geo file above
        if(KWORD.EQ.KWFMNM) then
            read(BUFFER,'(A32)') MFILE

```

```

        write(6,1102) MFILE
1102  format(/3X,'PENGEOM''s mass file: ',A32/)
    34  continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER
        if(KWORD.EQ.KWCOMM) go to 34
    else
        write(6,*) 'You have to specify a mass file name although
1 it may not exist.'
        stop      'You have to specify a mass file name although
1 it may not exist.'
    end if
C    -- Output control
if(KWORD.EQ.KWFOUT) then
    read(BUFFER,*) IFullOutp
    35  continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER
        if(KWORD.EQ.KWCOMM) go to 35
else
    IFullOutp = 0
end if

C    -- Dose grid setup.
    if(KWORD.eq.KWDXDO.OR.KWORD.eq.KWDYDO.OR.KWORD.eq.KWDZDO) then
        write(6,1500)
1500  format(/3X,70('-'),/
    1   3X,'>>>>> User distributions to be tallied.')
```

```

    else
        write(6,*) 'Dose grid missing.'
        stop      'Dose grid missing.'
    end if
C    -- x- dose grid
NDxRgn = 0
NxDT = 0
    if(KWORD.EQ.KWDXDO) then
271  NDxRgn = NDxRgn + 1
        read(buffer,*) dxD(NDxRgn), DxLim1, DxLim2
        if(DxLim1.ge.DxLim2) then
            write(6,'(a)') 'Dose grid x-bounds should be increasing.'
            stop 'Dose grid x-bounds should be increasing.'
        end if
        if(NDxRgn.gt.1.AND.DxLim1.lt.DBndX(NDxRgn)) then
            write(6,'(a)') 'The adjacent dose grid x-bounds overlapped.'
            stop 'The adjacent dose grid x-bounds overlapped.'
        end if
        DBndX(NDxRgn) = DxLim1
        DBndX(NDxRgn+1) = DxLim2
        NxDT(NDxRgn) = nint(
            1 (DBndX(NDxRgn+1)-DBndX(NDxRgn))/dxD(NDxRgn))
        NxDT = NxDT + NxDT(NDxRgn)
    27  continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER
        if(KWORD.EQ.KWCOMM) go to 27
    if(KWORD.EQ.KWDXDO) go to 271
    if(NxDT.gt.NDXM) then
        write(6,'(a)') 'NDXM is too small.'
        stop 'NDXM is too small..'
    end if
else
    write(6,*) 'Dose grid info in x-direction missing.'
    stop      'Dose grid info in x-direction missing.'
end if

        write(6,1451) dxD(1), DBndX(1), DBndX(2)
        do i=2, NDxRgn
            write(6,1450) dxD(i), DBndX(i), DBndX(i+1)
        end do
k = 0
        do i=1, NDxRgn

```

```

if(dxD(i).LT.0.0) then
    write(6,'(a,i3,a)') 'dxD(' ,i,') could not be negative.'
    write(6,'(a,i3,a)') 'dxD(' ,i,') could not be negative.'
    stop
end if
do j=1, NxDT(i)
    k = k + 1
    XDose(k) = DBndX(i) + dxD(i)*(j-1)
end do
end do
XDose(NxDT+1) = DBndX(NxRgn+1)

C    -- y- dose grid
NDyRgn = 0
NyDT = 0
    if(KWORD.EQ.KWDYDO) then
281  NDyRgn = NDyRgn + 1
    read(buffer,*) dyD(NDyRgn), DyLim1, DyLim2
    if(DyLim1.ge.DyLim2) then
        write(6,'(a)') 'Dose grid y-bounds should be increasing.'
        stop 'Dose grid y-bounds should be increasing.'
    end if
    if(NDyRgn.gt.1.AND.DyLim1.lt.DBndY(NDyRgn)) then
        write(6,'(a)') 'The adjacent dose grid y-bounds overlapped.'
        stop 'The adjacent dose grid y-bounds overlapped.'
    end if
    DBndY(NDyRgn) = DyLim1
    DBndY(NDyRgn+1) = DyLim2
    NyD(NDyRgn) = nint(
        1 (DBndY(NDyRgn+1)-DBndY(NDyRgn))/dyD(NDyRgn))
    NyDT = NyDT + NyD(NDyRgn)
28  continue
    read(5,'(A6,1X,A128)') KWORD,BUFFER
    if(KWORD.EQ.KWCOMM) go to 28
    if(KWORD.EQ.KWDYDO) go to 281
    if(NyDT.gt.NDYM) then
        write(6,'(a)') 'NDYM is too small.'
        stop 'NDYM is too small..'
    end if
else
    write(6,*) 'Dose grid info in y-direction missing.'
    stop 'Dose grid info in y-direction missing.'
end if

    write(6,1452) dyD(1), DBndY(1), DBndY(2)
    do i=2, NDyRgn
        write(6,1450) dyD(i), DBndY(i), DBndY(i+1)
    end do
k = 0
    do i=1, NDyRgn
    if(dyD(i).LT.0.0) then
        write(6,'(a,i3,a)') 'dyD(' ,i,') could not be negative.'
        write(6,'(a,i3,a)') 'dyD(' ,i,') could not be negative.'
        stop
    end if
    do j=1, NyD(i)
        k = k + 1
        YDose(k) = DBndY(i) + dyD(i)*(j-1)
    end do
    end do
YDose(NyDT+1) = DBndY(NDyRgn+1)

C    -- z- dose grid
NDzRgn = 0
NzDT = 0
    if(KWORD.EQ.KWDZDO) then
291  NDzRgn = NDzRgn + 1
    read(buffer,*) dzD(NDzRgn), DzLim1, DzLim2

```

```

if(DzLim1.ge.DzLim2) then
  write(6,'(a)') 'Dose grid z-bounds should be increasing.'
  stop 'Dose grid z-bounds should be increasing.'
end if
if(NDzRgn.gt.1.AND.DzLim1.lt.DBndZ(NDzRgn)) then
  write(6,'(a)') 'The adjacent dose grid z-bounds overlapped.'
  stop 'The adjacent dose grid z-bounds overlapped.'
end if
DBndZ(NDzRgn) = DzLim1
DBndZ(NDzRgn+1) = DzLim2
NzD(NDzRgn) = nint(
  1 (DBndZ(NDzRgn+1)-DBndZ(NDzRgn))/dzD(NDzRgn))
NzDT = NzDT + NzD(NDzRgn)
29 continue
  read(5,'(A6,1X,A128)') KWORD,BUFFER
  if(KWORD.EQ.KWCOMM) go to 29
if(KWORD.EQ.KWZDO) go to 291
if(NzDT.gt.NDZM) then
  write(6,'(a)') 'NDZM is too small.'
  stop 'NDZM is too small..'
end if
else
  write(6,*) 'Dose grid info in z-direction missing.'
  stop 'Dose grid info in z-direction missing.'
end if

  write(6,1453) dzD(1), DBndZ(1), DBndZ(2)
  do i=2, NDzRgn
    write(6,1450) dzD(i), DBndZ(i), DBndZ(i+1)
  end do
k = 0
  do i=1, NDzRgn
    if(dzD(i).LT.0.0) then
      write(6,'(a,i3,a)') 'dzD('i,') could not be negative.'
      write(6,'(a,i3,a)') 'dzD('i,') could not be negative.'
      stop
    end if
    do j=1, NzD(i)
      k = k + 1
      ZDose(k) = DBndZ(i) + dzD(i)*(j-1)
    end do
  end do
ZDose(NzDT+1) = DBndZ(NDzRgn+1)
1450 format( 10x,f8.3,2x,'in [',f8.3,',',f8.3,']')
1451 format(3x,'dx:',4x,f8.3,2x,'in [',f8.3,',',f8.3,']')
1452 format(3x,'dy:',4x,f8.3,2x,'in [',f8.3,',',f8.3,']')
1453 format(3x,'dz:',4x,f8.3,2x,'in [',f8.3,',',f8.3,']')

if(IFullOutp.gt.0) then
  inquire(FILE = MFILE, EXIST = exists)
  if(.not. exists) then
    write(6,'(2A/)') ' >> Cannot find file ', MFILE
    write(*,'(2A/)') ' >> Cannot find file ', MFILE
    stop
  end if
  open(15,FILE=MFILE)
  read(15,*) NDX, NDY, NDZ
  if(NDX.ne.NxDT.or.NDY.lt.NyDT.or.NDZ.ne.NzDT) then
    write(6,*) 'The mass grid is not consistent with
      1 the dose grid.'
    stop 'The mass grid is not consistent with
      1 the dose grid.'
  end if
  read(15,*) Mbuffer
  do while(trim(Mbuffer).ne.'Mass...')
    read(15,*) Mbuffer
  end do
  read(15,*) ((( Mass(i,j,k),i=1,NDX),j=1,NDY),k=1,NDZ)

```

```

        close(15)
    end if

C   -- Job characteristics.
    write(6,1700)
1700 format(/3X,70('-'),/
    1 3X,'>>>>> Job characteristics.')
C   -- resume file
    IRESUM=0
    if(KWORD.EQ.KWRESU) then
        read(BUFFER,'(A32)') RESUMEF
        write(6,1710) RESUMEF
1710 format(3X,'Resume simulation from previous dump file: ',A32)
        IRESUM=1
    71 continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER
        if(KWORD.EQ.KWCOMM) go to 71
    end if
C   -- dump file
    IDUMP=0
    if(KWORD.EQ.KWDUMP) then
        read(BUFFER,'(A32)') DUMPF
        write(6,1720) DUMPF
1720 format(3X,'Write final counter values on the dump file: ',A32)
        IDUMP=1
    72 continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER
        if(KWORD.EQ.KWCOMM) go to 72
    end if
C   -- simulation shower number
    if(KWORD.EQ.KWNSIM) then
        read(BUFFER,*) NTOT
        if(NTOT.LT.1) NTOT=2147483647
    73 continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER
        if(KWORD.EQ.KWCOMM) go to 73
    else
        NTOT=2147483647
    end if
    write(6,1730) NTOT
1730 format(3X,'Number of showers to be simulated = ',I11)
C   -- dump interval
    if(KWORD.EQ.KWNITR) then
        read(BUFFER,*) NINTER
        if(NINTER.LT.1) NINTER = 100000
    99 continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER
        if(KWORD.EQ.KWCOMM) go to 99
    else
        NINTER = 100000
    end if
    write(6,1735) NINTER
1735 format(3X,'Number of showers to be output per interval = ',I11)
C   -- random seeds
    if(KWORD.EQ.KWRSEE) then
        read(BUFFER,*) ISEED1,ISEED2
    74 continue
        read(5,'(A6,1X,A128)') KWORD,BUFFER
        if(KWORD.EQ.KWCOMM) go to 74
    else
        ISEED1 = 12345
        ISEED2 = 54321
    end if
    if(IRESUM.EQ.0) write(6,1740) ISEED1,ISEED2
1740 format(3X,'Random number generator seeds = ',I10,', ',I10)
C   -- simulation time
    if(KWORD.EQ.KWTIME) then

```

```

        read(BUFFER,*) ITIME
    else
        ITIME = 100
    end if
    write(6,1750) ITIME
1750 format(3X,'Computation time available = ',I12,' sec')
    if(ITIME.LT.1) ITIME=100
C
    call TIMER(TSEC)
    TSECIN = TSEC
    TSECA = TSEC + ITIME
    write(6,1760)
1760 format(/3X,70('-'))

C  -- If 'RESUME' is active, read previously generated counters...
    NA=0
    TIMEA=0.0D0
    if(IRESUM.EQ.1) then
        open(9,FILE=RESUMEF)
        read (9,*,ERR=1800,END=1800) NAA,TIMEAA
        NA=NAA
        TIMEA=TIMEAA
        read(9,*) ISEED1,ISEED2
        read(9,*) NxDTt, NyDTt, NzDTt
        if(NxDTt.ne.NxDT.or.NyDTt.ne.NyDT.or.NzDTt.ne.NzDT) then
            write(6,*) '>>Dose grid not consistent with the resume file.'
            stop      '>>Dose grid not consistent with the resume file.'
        end if
        read(9,999) ((( Dose(i,j,k),
            1          i=1,NxDT),j=1,NyDT),k=1,NzDT),
            2          ((VDose(i,j,k),
            3          i=1,NxDT),j=1,NyDT),k=1,NzDT)
        close(9)
        go to 1802
1800 continue
        write(6,1801)
1801 format(/3X,'WARNING: No resume file...',/)
    end if
1802 continue

    WGHT=1.0D0
C
    N=NA ! Shower counter.
    if(NTOT.LT.0) then
        write(6,*) ' WARNING: NTOT is too large. INTEGER*4 overflow.'
        if(NTOT.LT.1) NTOT=2147483647
    end if
    if(N.GE.NTOT) go to 106

C  =====
C  Shower simulation starts here.
C  =====
101 continue
    N = N + 1

C  -- Set the initial state of the primary particle.
    KPAR = KPARP
C  -- initial position ...
    X = XO
    Y = YO
    Z = ZO
C  -- initial direction.
    if(ISShp.eq.0) then          ! circular field
        call GCONE(U,V,W)
    else if(ISShp.eq.1) then    ! square field
        call GCONES(U,V,W,d1S,ssd)
    end if
C  -- initial energy ...

```



```

if(ISPEC.EQ.0) then ! Monoenergetic source.
  E=EO
ELSE ! Continuous spectrum. E sampled by Walker's method.
  RN=RAND(4.0D0)*NSEB+1
  K=INT(RN)
  RNF=RN-K
  if(RNF.GT.FS(K)) THEN
    KE=IAS(K)
  ELSE
    KE=K
  ENDIF
  E=ES(KE)+RAND(5.0D0)*(ES(KE+1)-ES(KE))
ENDIF

C  -- Check if the trajectory intersects the material system.
call LOCATE
if(MAT.EQ.0) THEN
  call STEP(1.0D30,DSEF,NCROSS)
  if(MAT.EQ.0) THEN ! The particle does not enter the system.
    go to 105
  ENDIF
ENDIF

C  -- Initialization of primary particle counters.
ILB(1)=1 ! Identifies primary particles.
ILB(2)=0
ILB(3)=0
ILB(4)=0
ILB(5)=0

C  -----
C  -- Track simulation begins here.
C
call CLEANS ! Cleans the secondary stack.
102 continue
call START ! Starts simulation in current medium.

103 continue
call JUMP(DSMAX(MAT),DS) ! Determines segment length.
call STEP(DS,DSEF,NCROSS) ! Moves particle to end of step.

C  -- Check whether the particle is outside the enclosure.
if(MAT.EQ.0) THEN
  go to 104
ENDIF

C  -- If the particle has crossed an interface, restart the track in the new material.
if(NCROSS.GT.0) go to 102

C  -- Simulate the interaction event
call KNOCK(DE,ICOL)

C  -- Dose distributions
C  -- Tally it if DE>0 and inside the dose grid
if(DE.gt.1e-35) then
  if(XDose(1).lt.x.AND.x.lt.XDose(NxDT+1).AND.
1  YDose(1).lt.y.AND.y.lt.YDose(NyDT+1).AND.
2  ZDose(1).lt.z.AND.z.lt.ZDose(NzDT+1)) then
    call get_dose_index(x,y,z,i,j,k)
  C  -- total dose
  if(N.NE.LDose(i,j,k)) then
    Dose(i,j,k) = Dose(i,j,k) + DoseP(i,j,k)
    VDose(i,j,k) = VDose(i,j,k) + DoseP(i,j,k)**2
    DoseP(i,j,k) = DE*WGHT
    LDose(i,j,k) = N
  else
    DoseP(i,j,k) = DoseP(i,j,k) + DE*WGHT
  end if

```

```

        end if ! (i,j,k)
        end if ! DE

C   -- Check if the particle has been absorbed .
        if(E.gt.EABS(KPAR,MAT)) go to 103

C   -- The simulation of the track ends here.
C   -----

104 continue

C   -- Any secondary left?
        call SECPAR(LEFT)
        if(LEFT.GT.0) THEN
C   -- Subtract E and charge from the tallied distributions to avoid double-counting.
        if(XDose(1).lt.x.AND.x.lt.XDose(NxDT+1).AND.
           1   YDose(1).lt.y.AND.y.lt.YDose(NyDT+1).AND.
           2   ZDose(1).lt.z.AND.z.lt.ZDose(NzDT+1)) then
        call get_dose_index(x,y,z,i,j,k)
C   -- total dose
        if(N.NE.LDOSE(i,j,k)) then
            Dose(i,j,k) = Dose(i,j,k) + DoseP(i,j,k)
            VDose(i,j,k) = VDose(i,j,k) + DoseP(i,j,k)**2
            DoseP(i,j,k) = -E*WGHT
            LDose(i,j,k) = N
        else
            DoseP(i,j,k) = DoseP(i,j,k) - E*WGHT
        end if
    end if

        go to 102
    ENDIF
C   -- The simulation of the shower ends here.
C   -----

105 continue
    call TIMER(TSEC)

    if(N.LT.NTOT.AND.mod(N, NINTER).eq.0) then
        Dt1 = Tsec-Tsecin
        IHours = INT(Dt1/3600)
        IMinutes = INT((Dt1-IHours*3600)/60)
        Seconds = Dt1 - IHours*3600 - IMinutes*60
        write(6,9999) 'N =', N,
           1   'Elapsed', Dt1, "(s)", "-->", IHours, ":",
           2   IMinutes, ":", Seconds
        write(*,9999) 'N =', N,
           1   'Elapsed', Dt1, "(s)", "-->", IHours, ":",
           2   IMinutes, ":", Seconds
9999   format(1x,A3,1x,I10,3x,A7,2x,f15.2,2x,A3,3X,A3,I6,A1,I6,A1,F6.2)
C   -- dump intermediate results in case that the job could not be finished as scheduled
        if(IDUMP.EQ.1) then
            TSIM=MAX(1.0D0,Dt1)+TIMEA
            open(9,FILE=DUMPF)
            write(9,*) N,TSIM
            write(9,*) ISEED1,ISEED2
            write(9,*) NxDT, NyDT, NzDT
            write(9,999) ((( (Dose(i,j,k)+DoseP(i,j,k)),
                1           i=1,NxDT),j=1,NyDT),k=1,NzDT),
                2           ((( (VDose(i,j,k)+DoseP(i,j,k)**2),
                3           i=1,NxDT),j=1,NyDT),k=1,NzDT)
        close(9)
        end if
    end if
999   format(g25.16)
C   -- end of intermediate dump
C   -----

```

```

      if(TSEC.LT.TSECA.AND.N.LT.NTOT) go to 101
C =====
C      End the simulation after the allotted time or after completing NTOT showers.
C =====

C      -- Transfer contents of partial counters of the last one shower to global counters.
      do k=1, NzDT
        do j=1, NyDT
          do i=1, NxDT
            Dose(i,j,k) = Dose(i,j,k) + DoseP(i,j,k)
            VDose(i,j,k) = VDose(i,j,k) + DoseP(i,j,k)**2
          end do
        end do
      end do
end do
C
      TSIM=MAX(1.0D0,TSEC-TSECIN)+TIMEA

C      -- If 'DUMPTO' is active, write counters to a dump file.
      if(IDUMP.EQ.1) then
        open(9,FILE=DUMPF)
        write(9,*) N,TSIM
        write(9,*) ISEED1,ISEED2
        write(9,*) NxDT, NyDT, NzDT
        write(9,999) ((( Dose(i,j,k),
          1           i=1,NxDT),j=1,NyDT),k=1,NzDT),
          2           ((VDose(i,j,k),
          3           i=1,NxDT),j=1,NyDT),k=1,NzDT)
        close(9)
      end if

C      -- Print simulation results.
      106 continue

      TOTN=N
      write(6,3000)
3000 format(//3X,36('*')/3X,'**  Program NewPENDOSES. Results.  **',
          1 /3X,36('*'))

      IFNT = N - NA
      TSIM=MAX(1.0D0,TSEC-TSECIN)
      write(6,3010) TSIM
3010 format(/3X,'Calculation time ..... ',
          1 1P,E13.6,' sec')
      TAVS=IFNT/TSIM
      write(6,3011) TAVS
3011 format(3X,'Simulation speed ..... ',
          1 1P,E13.6,' showers/sec')
      write(6,3012) IFNT
3012 format(/3X,
          1 'Simulated primary particles this time ..... ',1P,I13)
      write(6,3013) N
3013 format(/3X,
          1 'Total simulated primary particles ..... ',1P,I13)

      write(6,3099) ISEED1,ISEED2
3099 format(/3X,'Random seeds = ',I10,' , ',I10)
      write(6,'(//3X,'*** END ***')')
      close(6)

C      -- Print tallied distributions.
      IF(IFullOutp.gt.0) THEN ! output control

C      -- Total dose
      open(9,FILE='Dose.'//argsfx(1:len_sfx)//'.dat')
      DF=1.0D0/TOTN

      write(9,'(A,/)') 'Dose.'//argsfx(1:len_sfx)//'.dat'
      write(9,'(A,I8)') 'Num_Dose_Grid_X=', NxDT

```

```

write(9,'(A,I8)') 'Num_Dose_Grid_Y=', NyDT
write(9,'(A,I8)') 'Num_Dose_Grid_Z=', NzDT
write(9,'(/,A)') 'XDose(...)'
write(9,'(1000f8.3)') (XDose(i), i=1,NxDT+1)
write(9,'(/,A)') 'YDose(...)'
write(9,'(1000f8.3)') (YDose(i), i=1,NyDT+1)
write(9,'(/,A)') 'ZDose(...)'
write(9,'(1000f8.3)') (ZDose(i), i=1,NzDT+1)

do j=1, NyDT
  write(9,'(//a2,1x,f8.3,1x,a2)')
  1 'y=', (YDose(j)+YDose(j+1))/2, 'cm'
C -- value
  write(9,'(/1x,A)') 'Dose(MeV/g)'
  call Write_zx_header(9)
  do k=1, NzDT
    xx = (ZDose(k)+ZDose(k+1))/2
    do i=1, NxDT
      DAV(i) = Dose(i,j,k)*DF/1d6/max(1d-35,Mass(i,j,k))
    end do
    write(9,'(f8.3,500(1pg18.8))') xx,
  1 (DAV(i), i=1,NxDT)
  end do
C -- relative 1 sigma
  write(9,'(/1x,A)') '1-Sigma'
  call Write_zx_header(9)
  do k=1, NzDT
    xx = (ZDose(k)+ZDose(k+1))/2
    do i=1, NxDT
      DAV(i) = abs(Dose(i,j,k))
      DErr(i) = sqrt(abs(YDose(i,j,k)-Dose(i,j,k)**2*DF))
    DRel(i) = DErr(i)/max(1.0d-35,DAV(i))
    end do
    write(9,'(f8.3,500(1pg18.8))') xx,
  1 (DRel(i), i=1,NxDT)
  end do
end do

END IF ! output control

stop
END

C *****
C SUBROUTINE get_dose_index
C subroutine for finding out the indexes of (x,y,z) in the dose grid
C input:
C xx, yy, zz --> the coordinates;
C output:
C i, j, k --> the indexes in the dose grid
C *****
SUBROUTINE get_dose_index(xx,yy,zz,i,j,k)
IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER*4 (I-N)

parameter (NDXM=240,NDYM=300,NDZM=140)
Common/DoseGrid/YDose(NDYM+1),XDose(NDXM+1),ZDose(NDZM+1),
1 NyDT,NxDT,NzDT

C -- i
n1 = 1
n2 = NxDT
do while (n1.ne.n2)
  nm = (n2+n1)/2
  if(xx.le.XDose(nm+1)) then
    n2 = nm
  else
    n1 = nm + 1
  end if
end if

```

```

end do
i = n1
C -- j
n1 = 1
n2 = NyDT
do while (n1.ne.n2)
  nm = (n2+n1)/2
  if(yy.le.YDose(nm+1)) then
    n2 = nm
  else
    n1 = nm + 1
  end if
end do
j = n1
C -- k
n1 = 1
n2 = NzDT
do while (n1.ne.n2)
  nm = (n2+n1)/2
  if(zz.le.ZDose(nm+1)) then
    n2 = nm
  else
    n1 = nm + 1
  end if
end do
k = n1

end

C*****
C Subroutine for writing the common header
C given file unit id.
C   input:
C       fid --> file id;
C*****
subroutine Write_zx_header(fid)
IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER*4 (I-N)

integer fid

      parameter (NDXM=240,NDYM=300,NDZM=140)
Common/DoseGrid/YDose(NDYM+1),XDose(NDXM+1),ZDose(NDZM+1),
1      NyDT,NxDT,NzDT

write(fid,'(3x,a5,500(1pg18.4))')
1      'z-x', ((XDose(i)+XDose(i+1))/2, i=1,NxDT)

end

```

## BIBLIOGRAPHY

## BIBLIOGRAPHY

- [1] Vmc++, electron and photon monte carlo calculations optimized for radiation treatment planning. In A. Kling, F. Barao, M. Nakagawa, L. Tavora, and P. Vaz, editors, *Advanced Monte Carlo for Radiation Physics, Particle Transport Simulation and Applications: Proceedings of the Monte Carlo Meeting, Lisbon*, Berlin. Springer.
- [2] Intensity-modulated radiotherapy: current status and issues of interest. *Int J Radiat Oncol Biol Phys*, 51(4):880–914, 2001. Intensity Modulated Radiation Therapy Collaborative Working Group Consensus Development Conference Consensus Development Conference, NIH Journal Article Review United States.
- [3] Milton Abramowitz and Irene A. Stegun. *Handbook of mathematical functions, with formulas, graphs, and mathematical tables*. Dover Publications, New York,, [9th dover printing with corrections] edition, 1973. edited by Milton Abramowitz and Irene A. Stegun. ill. ; 27 cm. "This ninth Dover printing conforms to the tenth (December 1972) printing by the Government Printing Office, except that additional corrections have been made on pages 18, 79, 80, 82, 408, 450, 786, 825 and 934." Includes bibliographical references and index.
- [4] A. Ahnesjo, P. Andreo, and A. Brahme. Calculation and application of point spread functions for treatment planning with high energy photon beams. *Acta Oncol*, 26(1):49–56, 1987. Journal Article Research Support, Non-U.S. Gov't Sweden.
- [5] A. Ahnesjo and M. M. Aspradakis. Dose calculations for external photon beams in radiotherapy. *Phys Med Biol*, 44(11):R99–155, 1999. Journal Article Review England.
- [6] Frank H. Attix. *Introduction to radiological physics and radiation dosimetry*. Wiley, New York, 1986. Frank Herbert Attix. ill. ; 24 cm. "A Wiley-Interscience publication." Includes index.
- [7] J. M. Balter and M. L. Kessler. Imaging and alignment for image-guided radiation therapy. *J Clin Oncol*, 25(8):931–7, 2007. Journal Article Review United States official journal of the American Society of Clinical Oncology.
- [8] H. F. Batho. Lung corrections in cobalt 60 beam therapy. *J Can Assoc Radiol*, 15:79–83, 1964. Journal Article Canada.
- [9] B. Bauer-Kirpes, W. Schlegel, R. Boesecke, and W. J. Lorenz. Display of organs and isodoses as shaded 3-d objects for 3-d therapy planning. *Int J Radiat Oncol Biol Phys*, 13(1):135–40, 1987. Journal Article United states.
- [10] George I. Bell and Samuel Glasstone. *Nuclear reactor theory*. R. E. Krieger Pub. Co., Huntington, N.Y., 1979. 78022102 George I. Bell, Samuel Glasstone. ill. ; 24 cm. Reprint of the ed. published under auspices of the Division of Technical Information, United States Atomic Energy Commission, by Van Nostrand Reinhold Co., New York. Includes bibliographical references and indexes.
- [11] M. J. Berger. *Monte Carlo calculation of the penetration and diffusion of fast charged particles*, volume 1, pages 135–215. Academic Press, New York, 1963.

- [12] M. Blomquist and M. Karlsson. Measured lung dose correction factors for 50 mv photons. *Phys Med Biol*, 43(11):3225–3234, 1998.
- [13] A. Boyer and E. Mok. A photon dose distribution model employing convolution calculations. *Med Phys*, 12(2):169–77, 1985. Journal Article United states.
- [14] M. J. Brugmans, A. van der Horst, J. V. Lebesque, and B. J. Mijnheer. Beam intensity modulation to reduce the field sizes for conformal irradiation of lung tumors: a dosimetric study. *Int J Radiat Oncol Biol Phys*, 43(4):893–904, 1999.
- [15] M. Caro and J. Ligou. Treatment of scattering anisotropy of neutrons through the boltzmann-fokker-planck equation. *Nucl Sci Eng*, 83:242–250, 1983.
- [16] P. Carrasco, N. Jornet, M. A. Duch, L. Weber, M. Ginjaume, T. Eudaldo, D. Jurado, A. Ruiz, and M. Ribas. Comparison of dose calculation algorithms in phantoms with lung equivalent heterogeneities under conditions of lateral electronic disequilibrium. *Med Phys*, 31(10):2899–2911, 2004.
- [17] S. Chandrasekhar. Stochastic problems in physics and astronomy. *Rev Mod Phys*, 15:1, 1943.
- [18] S. Chandrasekhar. *Radiative transfer*. Dover Publications, New York, 1960. 60003117 /L/r85 illus. 22 cm. "Unabridged and slightly revised version of the work first published in 1950."
- [19] H. K. Chang and Manuel Paiva. *Respiratory physiology : an analytical approach*. Lung biology in health and disease ; v. 40. Dekker, New York, 1989. 88033432 edited by H.K. Chang, Manuel Paiva. ill. ; 24 cm. Includes bibliographies and indexes.
- [20] I. J. Chetty, P. M. Charland, N. Tyagi, D. L. McShan, B. A. Fraass, and A. F. Bielajew. Photon beam relative dose validation of the dpm monte carlo code in lung-equivalent media. *Med Phys*, 30(4):563–73, 2003. 0094-2405 (Print) Evaluation Studies Journal Article Validation Studies.
- [21] I. J. Chetty, J. M. Moran, D. L. McShan, B. A. Fraass, S. J. Wilderman, and A. F. Bielajew. Benchmarking of the dose planning method (dpm) monte carlo code using electron beams from a racetrack microtron. *Med Phys*, 29(6):1035–41, 2002. 0094-2405 (Print) Journal Article.
- [22] I. J. Chetty, J. M. Moran, T. S. Nurushev, D. L. McShan, B. A. Fraass, S. J. Wilderman, and A. F. Bielajew. Experimental validation of the dpm monte carlo code using minimally scattered electron beams in heterogeneous media. *Phys Med Biol*, 47(11):1837–51, 2002. 0031-9155 (Print) Journal Article.
- [23] I. J. Chetty, M. Rosu, D. L. McShan, B. A. Fraass, J. M. Balter, and R. K. Ten Haken. Accounting for center-of-mass target motion using convolution methods in monte carlo-based dose calculations of the lung. *Med Phys*, 31(4):925–32, 2004. P01-ca59827/ca/nci Clinical Trial Comparative Study Journal Article Research Support, Non-U.S. Gov't Research Support, U.S. Gov't, P.H.S. Validation Studies United States.
- [24] I. J. Chetty, M. Rosu, D. L. McShan, B. A. Fraass, and R. K. Ten Haken. The influence of beam model differences in the comparison of dose calculation algorithms for lung cancer treatment planning. *Phys Med Biol*, 50(5):801–815, 2005.
- [25] J. C. Chu, B. Ni, R. Kriz, and V. Amod Saxena. Applications of simulator computed tomography number for photon dose calculations during radiotherapy treatment planning. *Radiother Oncol*, 55(1):65–73, 2000.
- [26] H. Chung, H. Jin, J. Palta, T. S. Suh, and S. Kim. Dose variations with varying calculation grid size in head and neck imrt. *Phys Med Biol*, 51(19):4841–56, 2006.
- [27] AAPM Task Group #65 Radiation Therapy Committee. Tissue inhomogeneity corrections for megavoltage photon beams. Technical Report 85, July 2004.



- [28] C. Constantinou, J. C. Harrington, and L. A. DeWerd. An electron density calibration phantom for ct-based treatment planning computers. *Med Phys*, 19(2):325–7, 1992. Journal Article United states.
- [29] J. F. Corbett, J. Jezioranski, J. Crook, and I. Yeung. The effect of voxel size on the accuracy of dose-volume histograms of prostate 125i seed implants. *Med Phys*, 29(6):1003–6, 2002. Journal Article United States.
- [30] M. M. Coselmon, J. M. Balter, D. L. McShan, and M. L. Kessler. Mutual information based ct registration of the lung at exhale and inhale breathing states using thin-plate splines. *Med Phys*, 31(11):2942–8, 2004. P01-ca59827/ca/nci Clinical Trial Comparative Study Journal Article Research Support, U.S. Gov’t, P.H.S. United States.
- [31] J. E. Cotes and G. L. Leathart. *Lung function: assessment and application in medicine*. Blackwell Scientific Publications, Oxford, 5th edition, 1993. J.E. Cotes with the editorial collaboration of G.L. Leathart. ill. ; 25 cm.
- [32] J. E. Cygler, G. M. Daskalov, G. H. Chan, and G. X. Ding. Evaluation of the first commercial monte carlo dose calculation engine for electron beam treatment planning. *Med Phys*, 31(1):142–53, 2004. Journal Article Research Support, Non-U.S. Gov’t United States.
- [33] R. K. Das, R. Patel, H. Shah, H. Odau, and R. R. Kuske. 3d ct-based high-dose-rate breast brachytherapy implants: treatment planning and quality assurance. *Int J Radiat Oncol Biol Phys*, 59(4):1224–8, 2004.
- [34] B. De Smedt, B. Vanderstraeten, N. Reynaert, W. De Neve, and H. Thierens. Investigation of geometrical and scoring grid resolution for monte carlo dose calculations for imrt. *Phys Med Biol*, 50(17):4005–4019, 2005.
- [35] J. F. Dempsey, H. E. Romeijn, J. G. Li, D. A. Low, and J. R. Palta. A fourier analysis of the dose grid resolution required for accurate imrt fluence map optimization. *Med Phys*, 32(2):380–8, 2005.
- [36] R. E. Drzymala, R. Mohan, L. Brewster, J. Chu, M. Goitein, W. Harms, and M. Urie. Dose-volume histograms. *Int J Radiat Oncol Biol Phys*, 21(1):71–8, 1991. N01 cm-47316/cm/nci N01 cm-47695/cm/nci N01 cm-47696/cm/nci etc. Journal Article Research Support, U.S. Gov’t, P.H.S. United states.
- [37] F. C. du Plessis, C. A. Willemse, M. G. Lotter, and L. Goedhals. Comparison of the batho, etar and monte carlo dose calculation methods in ct based patient models. *Med Phys*, 28(4):582–9, 2001.
- [38] L. Dumas and F. Golse. Homogenization of transport equations. *Siam Journal on Applied Mathematics*, 60(4):1447–1470, 2000.
- [39] E. El-Khatib and J. J. Battista. Improved lung dose calculation using tissue-maximum ratios in the batho correction. *Med Phys*, 11(3):279–86, 1984. Journal Article Research Support, Non-U.S. Gov’t United states.
- [40] M. Engelsman, E. M. Damen, P. W. Koken, A. A. van ’t Veld, K. M. van Ingen, and B. J. Mijnheer. Impact of simple tissue inhomogeneity correction algorithms on conformal radiotherapy of lung tumours. *Radiother Oncol*, 60(3):299–309, 2001.
- [41] M. Fippel. Fast monte carlo dose calculation for photon beams based on the vmc electron algorithm. *Med Phys*, 26(8):1466–75, 1999. Comparative Study Journal Article Research Support, Non-U.S. Gov’t United states.
- [42] B. A. Fraass. The development of conformal radiation therapy. *Med Phys*, 22(11 Pt 2):1911–21, 1995. Nci-p01-ca59827/ca/nci Historical Article Journal Article Research Support, U.S. Gov’t, P.H.S. United states.

- [43] S. J. Frank, K. M. Forster, C. W. Stevens, J. D. Cox, R. Komaki, Z. Liao, S. Tucker, X. Wang, R. E. Steadham, C. Brooks, and G. Starkschall. Treatment planning for lung cancer: traditional homogeneous point-dose prescription compared with heterogeneity-corrected dose-volume prescription. *Int J Radiat Oncol Biol Phys*, 56(5):1308–1318, 2003.
- [44] R. A. Geise and E. C. McCullough. The use of ct scanners in megavoltage photon-beam therapy planning. *Radiology*, 124(1):133–41, 1977.
- [45] M. Goitein and M. Abrams. Multi-dimensional treatment planning: I. delineation of anatomy. *Int J Radiat Oncol Biol Phys*, 9(6):777–87, 1983. Ca-00251/ca/nci Ca-21239/ca/nci Journal Article Research Support, U.S. Gov't, P.H.S. United states.
- [46] M. Goitein, M. Abrams, D. Rowell, H. Pollari, and J. Wiles. Multi-dimensional treatment planning: Ii. beam's eye-view, back projection, and projection through ct sections. *Int J Radiat Oncol Biol Phys*, 9(6):789–97, 1983. Ca-00251/ca/nci Ca-21239/ca/nci Journal Article Research Support, Non-U.S. Gov't Research Support, U.S. Gov't, P.H.S. United states.
- [47] H. Gono, K. Fujimoto, S. Kawakami, and K. Kubo. Evaluation of airway wall thickness and air trapping by hrct in asymptomatic asthma. *Eur Respir J*, 22(6):965–71, 2003. 0903-1936 (Print) Journal Article.
- [48] M. Goosens, S. Giani, and S. Ravndal. Geant: detector description and simulation tool. Technical report, Geneva, Switzerland, 1993. CERN Program Library, long writeup W5013 CERN, Geneva, Switzerland.
- [49] Henry Gray, Peter L. Williams, and Lawrence H. Bannister. *Gray's anatomy : the anatomical basis of medicine and surgery*. Churchill Livingstone, New York, 38th edition, 1995. 95005806 Anatomy chairman of the editorial board, Peter L. Williams ; editorial board, Lawrence H. Bannister ... [et al.]. ill. (some col.) ; 31 cm. Includes bibliographical references (p. 1937-2044) and index.
- [50] H. Guan, F. F. Yin, and J. H. Kim. Accuracy of inhomogeneity correction in photon radiotherapy from ct scans with different settings. *Phys Med Biol*, 47(17):N223–31, 2002.
- [51] B. Haefeli-Bleuer and E. R. Weibel. Morphometry of the human pulmonary acinus. *Anat Rec*, 220(4):401–414, 1988.
- [52] C. L. Hartmann Siantar, R. S. Walling, T. P. Daly, B. Faddegon, N. Albright, P. Bergstrom, A. F. Bielajew, C. Chuang, D. Garrett, R. K. House, D. Knapp, D. J. Wiczcerek, and L. J. Verhey. Description and dosimetric verification of the peregrine monte carlo dose calculation system for photon beams incident on a water phantom. *Med Phys*, 28(7):1322–37, 2001. Journal Article Research Support, U.S. Gov't, Non-P.H.S. United States.
- [53] K. Horsfield and G. Cumming. Morphology of the bronchial tree in man. *J Appl Physiol*, 24(3):373–383, 1968.
- [54] K. Horsfield, G. Dart, D. E. Olson, G. F. Filley, and G. Cumming. Models of the human bronchial tree. *J Appl Physiol*, 31(2):207–217, 1971.
- [55] W. Huang, R. T. Yen, M. McLaurine, and G. Bledsoe. Morphometry of the human pulmonary vasculature. *J Appl Physiol*, 81(5):2123–2133, 1996.
- [56] Theodore M. Jenkins, Walter R. Nelson, and Alessandro Rindi. *Monte Carlo transport of electrons and photons*. Ettore Majorana international science series. Physical sciences ; v. 38. Plenum Press, New York, 1989. 88031147 International School of Radiation Damage and Protection (8th : 1987 : Erice, Italy) edited by Theodore M. Jenkins, Walter R. Nelson, and Alessandro Rindi. ill. ; 26 cm. "Proceedings of the International School of Radiation Damage and Protection, eighth course ... held September 24-October 3, 1987, in Erice, Sicily, Italy"—T.p. verso. Includes bibliographies and index.

- [57] H. Jiang, J. Seco, and H. Paganetti. Effects of hounsfield number conversion on ct based proton monte carlo dose calculations. *Med Phys*, 34(4):1439–49, 2007.
- [58] I. Kawrakow. Accurate condensed history monte carlo simulation of electron transport. i. egsnrc, the new egs4 version. *Med Phys*, 27(3):485–98, 2000. Journal Article Research Support, Non-U.S. Gov't United states.
- [59] I. Kawrakow. On the de-noising of monte carlo calculated dose distributions. *Phys Med Biol*, 47(17):3087–103, 2002.
- [60] M. L. Kessler, S. Pitluck, P. Petti, and J. R. Castro. Integration of multimodality imaging data for radiotherapy treatment planning. *Int J Radiat Oncol Biol Phys*, 21(6):1653–67, 1991.
- [61] W. Kilby, J. Sage, and V. Rabett. Tolerance levels for quality assurance of electron density values generated from ct in radiotherapy treatment planning. *Phys Med Biol*, 47(9):1485–92, 2002.
- [62] E. E. Klein, A. Morrison, J. A. Purdy, M. V. Graham, and J. Matthews. A volumetric study of measurements and calculations of lung density corrections for 6 and 18 mv photons. *Int J Radiat Oncol Biol Phys*, 37(5):1163–1170, 1997.
- [63] T. Knöös, A. Ahnesjö, P. Nilsson, and L. Weber. Limitations of a pencil beam approach to photon dose calculations in lung tissue. *Phys Med Biol*, 40(9):1411–1420, 1995.
- [64] T. Krieger and O. A. Sauer. Monte carlo- versus pencil-beam-/collapsed-cone-dose calculation in a heterogeneous multi-layer phantom. *Phys Med Biol*, 50(5):859–68, 2005. Journal Article England.
- [65] G. J. Kutcher and C. Burman. Calculation of complication probability factors for non-uniform normal tissue irradiation: the effective volume method. *Int J Radiat Oncol Biol Phys*, 16(6):1623–30, 1989. Nci-cm-37616-21/cm/nci Journal Article Research Support, U.S. Gov't, P.H.S. United states.
- [66] E. Larsen. Asymptotic derivation of the atomic-mix diffusion model for 1-d random media. *Trans Am Nucl Soc*, 89:296, 2003.
- [67] E. Larsen, R. Vasques, and M. Vilhena. Particle transport in the 1-d diffusive atomic mix limit. In *Proc. Conf. on Mathematics and Computation, Supercomputing, Reactor Physics and Nuclear and Biological Applications*, Avignon, France, 9 2005. American Nuclear Society. On CD-ROM.
- [68] E. W. Larsen and L. Liang. The atomic mix approximation for charged particle transport. *SIAM Journal on Appl Math*, 2007. Accepted for publication.
- [69] X. Li, P. Zhang, D. Mah, R. Gewanter, and G. Kutcher. Novel lung imrt planning algorithms with nonuniform dose delivery strategy to account for respiratory motion. *Med Phys*, 33(9):3390–8, 2006.
- [70] L. Liang, E. W. Larsen, and I. J. Chetty. An anatomically realistic lung model for monte carlo-based dose calculations. *Med Phys*, 34(3):1013–25, 2007. Journal Article Research Support, U.S. Gov't, Non-P.H.S. United States.
- [71] D. Liljequist and M. Ismail. Transport mean free-path related to trajectory patterns - comparison of nonrelativistic and highly relativistic electron penetration through matter. *Journal of Applied Physics*, 62(2):342–350, 1987.
- [72] D. Liljequist, M. Ismail, F. Salvat, R. Mayol, and J. D. Martinez. Transport mean free-path tabulated for the multiple elastic-scattering of electrons and positrons at energies less-than-or-equal-to-20 mev. *Journal of Applied Physics*, 68(7):3061–3065, 1990.

- [73] J. T. Lyman. Complication probability as assessed from dose-volume histograms. *Radiat Res Suppl*, 8:S13–9, 1985.
- [74] C.M. Ma, J.S. Li, T. Pawlicki, and *et al.* Mcdose - a monte carlo dose calculation tool for radiation therapy treatment planning. In *Proceedings of the XIII International Conference on the Use of Computer in Radiation Therapy*, Heidelberg. Springer-Verlag.
- [75] T. R. Mackie, E. el Khatib, J. Battista, J. Scrimger, J. Van Dyk, and J. R. Cunningham. Lung dose corrections for 6- and 15-mv x rays. *Med Phys*, 12(3):327–332, 1985.
- [76] T. R. Mackie, J. W. Scrimger, and J. J. Battista. A convolution method of calculating dose for 15-mv x rays. *Med Phys*, 12(2):188–96, 1985. Journal Article Research Support, Non-U.S. Gov't United states.
- [77] K. Mah and J. Van Dyk. On the impact of tissue inhomogeneity corrections in clinical thoracic radiation therapy. *Int J Radiat Oncol Biol Phys*, 21(5):1257–1267, 1991.
- [78] S. Matsuoka, Y. Kurihara, Y. Nakajima, H. Niimi, H. Ashida, and K. Kaneoya. Serial change in airway lumen and wall thickness at thin-section ct in asymptomatic subjects. *Radiology*, 234(2):595–603, 2005. 0033-8419 (Print) Journal Article.
- [79] S. Matsuoka, K. Uchiyama, H. Shima, N. Ueno, S. Oish, and Y. Nojiri. Bronchoarterial ratio and bronchial wall thickness on high-resolution ct in asymptomatic subjects: correlation with age and smoking. *AJR Am J Roentgenol*, 180(2):513–518, 2003.
- [80] E. C. McCullough. Potentials of computed tomography in radiation therapy treatment planning. *Radiology*, 129(3):765–8, 1978. Journal Article United states.
- [81] John A. McDonald. *Lung growth and development*. Lung biology in health and disease ; v. 100. M. Dekker, New York, 1997.
- [82] W. G. McKenna, K. Yeakel, A. Klink, B. A. Fraass, J. van de Geijn, E. Glatstein, and A. S. Lichter. Is correction for lung density in radiotherapy treatment planning necessary? *Int J Radiat Oncol Biol Phys*, 13(2):273–278, 1987.
- [83] R. C. Miller, J. A. Bonner, and R. W. Kline. Impact of beam energy and field margin on penumbra at lung tumor-lung parenchyma interfaces. *Int J Radiat Oncol Biol Phys*, 41(3):707–713, 1998.
- [84] R. Mohan, C. Chui, and L. Lidofsky. Differential pencil beam dose computation model for photons. *Med Phys*, 13(1):64–73, 1986. Journal Article United states.
- [85] D. L. Newman, G. Dougherty, A. al Obaid, and H. al Hajrasy. Limitations of clinical ct in assessing cortical thickness and density. *Phys Med Biol*, 43(3):619–26, 1998.
- [86] A. Niemierko and M. Goitein. The influence of the size of the grid used for dose calculation on the accuracy of dose estimation. *Med Phys*, 16(2):239–47, 1989.
- [87] International Commission on Radiation Units and Measurements. *Tissue substitutes in radiation dosimetry and measurement*. ICRU report ; 44. Washington,, 1989.
- [88] C. G. Orton, S. Chungbin, E. E. Klein, M. T. Gillin, T. E. Schultheiss, and W. T. Sause. Study of lung density corrections in a clinical trial (rtog 88-08). radiation therapy oncology group. *Int J Radiat Oncol Biol Phys*, 41(4):787–794, 1998.
- [89] C. G. Orton, P. M. Mondalek, J. T. Spicka, D. S. Herron, and L. I. Andres. Lung corrections in photon beam treatment planning: are we ready? *Int J Radiat Oncol Biol Phys*, 10(12):2191–2199, 1984.

- [90] N. Papanikolaou and E. E. Klein. Heterogeneity corrections should be used in treatment planning for lung cancer. *Med Phys*, 27(8):1702–4, 2000. Journal Article United states.
- [91] H. Parker, K. Horsfield, and G. Cumming. Morphology of distal airways in the human lung. *J Appl Physiol*, 31(3):386–391, 1971.
- [92] C. E. Pelloski, M. Palmer, G. M. Chronowski, A. Jhingran, J. Horton, and P. J. Eifel. Comparison between ct-based volumetric calculations and icru reference-point estimates of radiation doses delivered to bladder and rectum during intracavitary radiotherapy for cervical cancer. *Int J Radiat Oncol Biol Phys*, 62(1):131–7, 2005.
- [93] Photon Treatment Planning Collaborative Working Group. Role of inhomogeneity corrections in three-dimensional photon treatment planning. *Int J Radiat Oncol Biol Phys*, 21(1):59–69, 1991.
- [94] G. C. Pomraning. *Linear kinetic theory and particle transport in stochastic mixtures*. Series on advances in mathematics for applied sciences ; vol. 7. World Scientific, Singapore ; New Jersey, 1991. 91041290 Gerald C. Pomraning. 23 cm. Includes bibliographical references.
- [95] G.C. Pomraning. The fokker-planck operator as an asymptotic limit. *Math Models and Methods Appl Sci*, (2):21, 1992.
- [96] K. Przybylski and J. Lizou. Numerical analysis of the boltzmann equation including fokker-planck terms. *Nucl Sci Eng*, 81(1):92–109, 1982.
- [97] R. K. Rice, B. J. Mijnheer, and L. M. Chin. Benchmark measurements for lung dose corrections for x-ray beams. *Int J Radiat Oncol Biol Phys*, 15(2):399–409, 1988.
- [98] R. Roberts. How accurate is a ct-based dose calculation on a pencil beam tps for a patient with a metallic prosthesis? *Phys Med Biol*, 46(9):N227–34, 2001.
- [99] D. W. Rogers. Fifty years of monte carlo simulations for medical physics. *Phys Med Biol*, 51(13):R287–301, 2006. Journal Article Review England.
- [100] M. Rosu, J. M. Balter, I. J. Chetty, M. L. Kessler, D. L. McShan, P. Balter, and R. K. Ten Haken. How extensive of a 4d dataset is needed to estimate cumulative dose distribution plan evaluation metrics in conformal lung therapy? *Med Phys*, 34(1):233–45, 2007. P01 ca59872/ca/nci R01 ca106770/ca/nci Evaluation Studies Journal Article Research Support, N.I.H., Extramural United States.
- [101] M. Rosu, I. J. Chetty, J. M. Balter, M. L. Kessler, D. L. McShan, and R. K. Ten Haken. Dose reconstruction in deforming lung anatomy: dose grid size effects and clinical implications. *Med Phys*, 32(8):2487–95, 2005.
- [102] M. Rosu, I. J. Chetty, D. S. Tatro, and R. K. Ten Haken. The impact of breathing motion versus heterogeneity effects in lung cancer treatment planning. *Med Phys*, 34(4):1462–73, 2007.
- [103] H. Saitoh, T. Fujisaki, R. Sakai, and E. Kunieda. Dose distribution of narrow beam irradiation for small lung tumor. *Int J Radiat Oncol Biol Phys*, 53(5):1380–7, 2002. 0360-3016 (Print) Journal Article.
- [104] F. Salvat, J. M. Fernandez-Varea, and J. Sempau. *PENELOPE, a code system for Monte Carlo simulation of electron and photon transport*. OECD Nuclear Energy Agency, 2003.
- [105] J. G. Scadding, Gordon Cumming, and William M. Thurlbeck. *Scientific foundations of respiratory medicine*. Heinemann Medical Books, London, 1981. edited by J.G. Scadding and Gordon Cumming ; associate editor, W.M. Thurlbeck. ill. ; 29 cm. Includes bibliographies and index.

- [106] U. Schneider, E. Pedroni, and A. Lomax. The calibration of ct hounsfield units for radiotherapy treatment planning. *Phys Med Biol*, 41(1):111–24, 1996.
- [107] C. Scholz, S. Nill, and U. Oelfke. Comparison of imrt optimization based on a pencil beam and a superposition algorithm. *Med Phys*, 30(7):1909–13, 2003.
- [108] J. Sempau, S. J. Wilderman, and A. F. Bielajew. Dpm, a fast, accurate monte carlo code optimized for photon and electron radiotherapy treatment planning dose calculations. *Phys Med Biol*, 45(8):2263–91, 2000. 0031-9155 (Print) Journal Article.
- [109] D. Sheikh-Bagheri and D. W. Rogers. Monte carlo calculation of nine megavoltage photon beam spectra using the beam code. *Med Phys*, 29(3):391–402, 2002.
- [110] C. W. Smith, D. Morrey, and K. Gray. The influence of grid size on accuracy in radiotherapy dose plotting. *Med Phys*, 17(1):135–6, 1990.
- [111] American Cancer Society. Cancer facts & figures 2007. Atlanta:American Cancer Society; 2007.
- [112] M. R. Sontag and J. R. Cunningham. Corrections to absorbed dose calculations for tissue inhomogeneities. *Med Phys*, 4(5):431–6, 1977. Journal Article United states.
- [113] M. R. Sontag and J. R. Cunningham. The equivalent tissue-air ratio method for making absorbed dose calculations in a heterogeneous medium. *Radiology*, 129(3):787–94, 1978. Journal Article United states.
- [114] W. L. Tang, F. M. Khan, and B. J. Gerbi. Validity of lung correction algorithms. *Med Phys*, 13(5):683–686, 1986.
- [115] M. H. Tawhai, P. Hunter, J. Tschirren, J. Reinhardt, G. McLennan, and E. A. Hoffman. Ct-based geometry analysis and finite element models of the human and ovine bronchial tree. *J Appl Physiol*, 97(6):2310–2321, 2004.
- [116] S. L. Tucker and J. M. Taylor. Improved models of tumour cure. *Int J Radiat Biol*, 70(5):539–53, 1996. Journal Article England.
- [117] S. Tyldesley, C. Boyd, K. Schulze, H. Walker, and W. J. Mackillop. Estimating the need for radiotherapy for lung cancer: an evidence-based, epidemiologic approach. *Int J Radiat Oncol Biol Phys*, 49(4):973–85, 2001. Journal Article Research Support, Non-U.S. Gov’t Review United States.
- [118] J. Van Dyk. Lung dose calculations using computerized tomography: is there a need for pixel based procedures? *Int J Radiat Oncol Biol Phys*, 9(7):1035–1041, 1983.
- [119] J. Van Dyk, T. J. Keane, and W. D. Rider. Lung density as measured by computerized tomography: implications for radiotherapy. *Int J Radiat Oncol Biol Phys*, 8(8):1363–1372, 1982.
- [120] B. Vanderstraeten, N. Reynaert, L. Paelinck, I. Madani, C. De Wagter, W. De Gersem, W. De Neve, and H. Thierens. Accuracy of patient dose calculation for lung imrt: A comparison of monte carlo, convolution/superposition, and pencil beam computations. *Medical physics*, 33(9):3149–58, 2006. Vanderstraeten, Barbara Reynaert, Nick Paelinck, Leen Madani, Indira De Wagter, Carlos De Gersem, Werner De Neve, Wilfried Thierens, Hubert Journal Article United States.
- [121] A. van’t Riet, H. C. Stam, A. C. Mak, and F. H. van Slooten. Implications of lung corrections for dose specification in radiotherapy. *Int J Radiat Oncol Biol Phys*, 11(3):621–625, 1985.
- [122] L. Wang, C. S. Chui, and M. Lovelock. A patient-specific monte carlo dose-calculation method for photon beams. *Med Phys*, 25(6):867–78, 1998.

- [123] Steve Webb. *The physics of three-dimensional radiation therapy : conformal radiotherapy, radiosurgery, and treatment planning*. Institute of Physics Pub., Bristol ; Philadelphia, 1993. 93006665 GB93-23570 Steve Webb. ill. ; 24 cm. Medical science series Includes bibliographical references and index.
- [124] Ewald R. Weibel. *Morphometry of the human lung*. Springer, Berlin,, 1963.
- [125] P. J. White, R. D. Zwicker, and D. T. Huang. Comparison of dose homogeneity effects due to electron equilibrium loss in lung for 6 mv and 18 mv photons. *Int J Radiat Oncol Biol Phys*, 34(5):1141–1146, 1996.
- [126] J. W. Wong and J. A. Purdy. On methods of inhomogeneity corrections for photon transport. *Med Phys*, 17(5):807–14, 1990. 0094-2405 (Print) Journal Article Review.
- [127] S. Yoo and F. F. Yin. Dosimetric feasibility of cone-beam ct-based treatment planning compared to ct-based treatment planning. *Int J Radiat Oncol Biol Phys*, 66(5):1553–61, 2006.
- [128] M. E. Young and R. O. Kornelsen. Dose corrections for low-density tissue inhomogeneities and air channels for 10-mv x rays. *Med Phys*, 10(4):450–5, 1983. 0094-2405 (Print) Journal Article.
- [129] M. Zaider and G. N. Minerbo. Tumour control probability: a formulation applicable to any temporal protocol of dose delivery. *Phys Med Biol*, 45(2):279–93, 2000. Journal Article England.